1

Modulation

1.1. Modulation?

1.1.1. *Main reasons for modulation*

– In the modulation process, one characteristic of a high-frequency (passband) carrier signal is modified according to the instantaneous amplitude of the signal to be processed (in its baseband).

– Why does modulation make sense for signal transmission (distance, etc.)?

– Several signals are transmitted on the same channel.

– Capacitative or inductive devices need high frequency (carrier) to operate (to ensure Stability and a good Noise Rejection).

1.1.2. *Main modulation schemas*

– Examples of application: transmitting audio and video signals.

– Mobile radio communications, such as cell phones.

Basic modulation types are as follows:

– amplitude modulation (AM): modifies the amplitude;

– frequency modulation (FM): changes the frequency;

– phase modulation (PM): changes the phase.

1.1.3. *Criteria for modulation via electronics*

Requirements for signal processing are as follows:

– speed;

– reliability and precision;

– low energy consumption.

Electronics serve to transport information at least energy cost, as quick as possible, in the greatest quantity possible, with the highest quality possible and with maximum reliability.

NOTE.– In electrical engineering, the electrical signal serves to transport the *energy* and not the *information*.

1.1.4. *Digital modulation: why do it?*

Digital modulation provides more information capacity, compatibility with digital data services, better security levels, better communication quality and wider availability than analog, for example.

1.2. Main technical constraints

– available passband;

– usable power;

– level of system background noise.

The radiofrequency (RF) spectrum is aimed at many users and communication services are only growing. Digital modulation has greater capacity to propagate large quantities of information than analog.

If all the frequencies that form the signal to be transmitted are found in the passband supporting the transmission, this signal can be applied to the line *directly*.

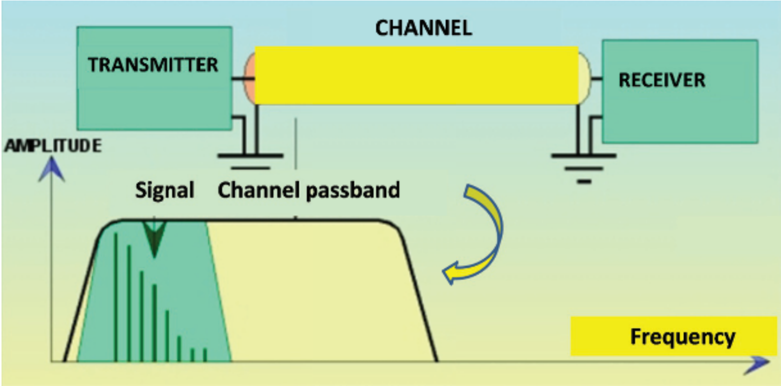


Figure 1.1. *Transmission of a signal in baseband. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

If all or some of the frequencies forming the signal to be transmitted are found outside the bandwidth of the transmission medium, a modulation will make it possible to shift all the frequencies, in *blocks,* to higher frequencies (see the so-called heterodyne process) (Figure 1.2).

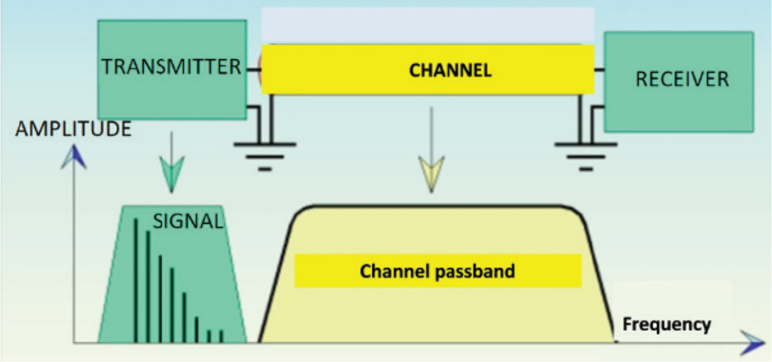


Figure 1.2. *Transmission of a signal via modulation. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

Each frequency f of the signal "becomes" $f + \Delta f$.

If Δf suffices, all frequencies of the signal thus modulated are *translated* within the passband. Once this procedure has been made by the *transmitter*, a *modulator* for translating the frequencies and a *demodulator* are needed at *reception* to regroup the frequencies, at the initial value, into blocks.

As the system is generally conversational, each station is equipped with a MODEM (MODulator-DEMdulator).

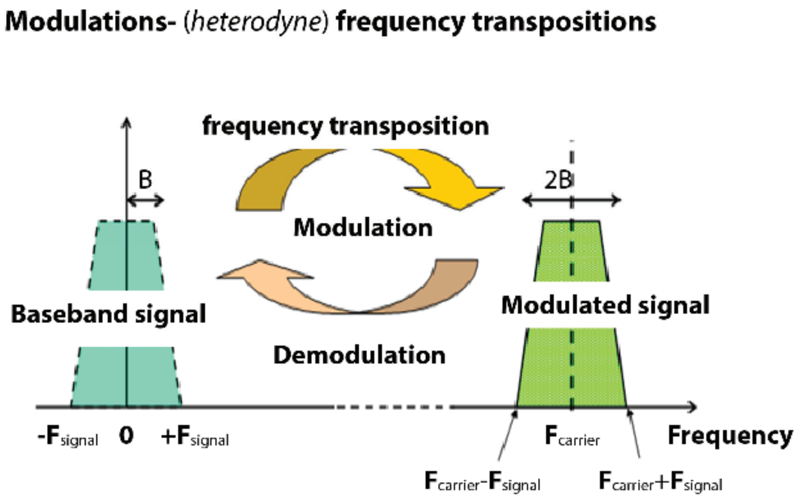


Figure 1.3. *Heterodyne system. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

Simplicity/bandwidth compromise

The need to have a broad passband limits the number of users. On the other hand, transmitters and receivers can be used with a lower passband. The increasing transition toward *spectral efficiency* calls for ever greater complexity in hardware; this requires sophisticated designs, it is not simple to manufacture or test. There are many compromises, depending on whether communication is aerial, wired, analog or digital.

Developments in industry

An important transition has occurred in recent years from analog AMs and FM/PM, towards digital modulation techniques.

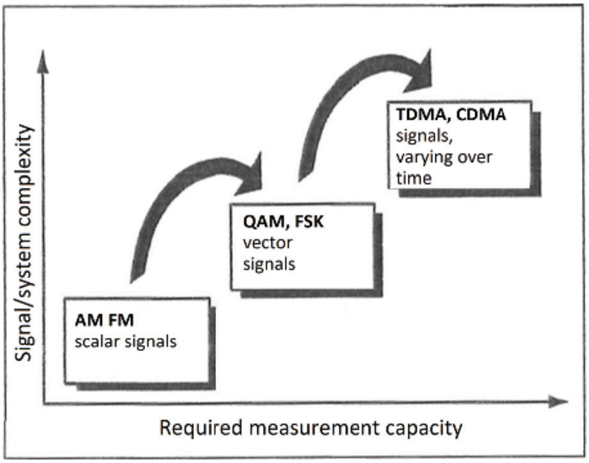


Figure 1.4. *Trends in industry*

NOTE.– The main types of *multiplexing* (or "multiple access") are time division multiple access (TDMA) and code division multiple access (CDMA). These are two different ways of adding diversity, enabling the signals to separate from one another (see below).

The strong trend toward *digital modulation*, compared to analog, provides more data capacity and more communication, it is more compatible with digital data services and it provides more data security, higher quality communication and faster systems. The constraints are as follows: *more bandwidth* available, the *noise* and *power* allowed the system. The RF spectrum should be shared; more and more users are involved. Communications services are growing. Digital modulation schemas are finder greater possibilities for conveying high quantities of information, if we compare them to analog modulation schemas.

Compromise between simplicity and bandwidth

There is a basic point in communication systems. A simple hardware can be used for transmitters and receivers, but this requires the spectrum to have a broad bandwidth, hence fewer users. On the other hand, complex transmitters and receivers can be used to transmit the same information over a lower transmission bandwidth; in contrast, this requires very sophisticated material. This compromise exists if communication occurs via air or via wires, if it is analog or digital.

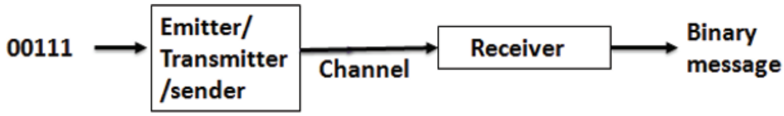


Figure 1.5. *Digital modulation chain*

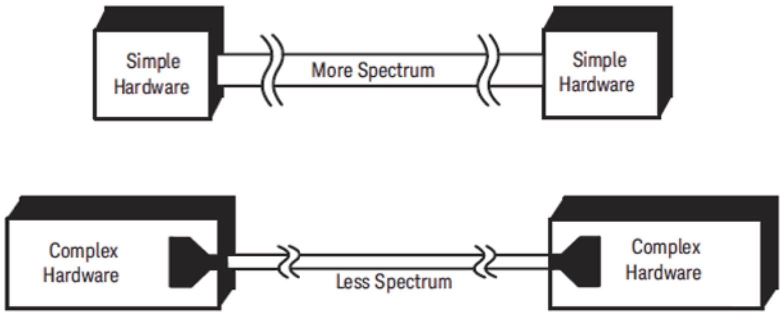


Figure 1.6. *Fundamental compromises*

1.3. Transmission of information (analog or digital)

To transmit a signal in the air, there are essentially three stages:

1) a pure *carrier* is created at the receiver;

2) this *carrier* is modulated with the information to be transmitted;

3) at the receiver, any modification or change of signal is detected and *demodulated*.

Lossless coding (see files, binary flow) is applied to the source's properties to reduce the volume of data to be transmitted; in fact, it means eliminating redundancies ("as" for compression). Lossy coding takes account of the receiver's properties; with lossy coding, it is information that is *a priori* useless or uninformative, that is deleted.

In fact, audio and video source coders use both lossless and lossy codes via digitization of analog data.

1.3.1. *Characteristics of the signal that can be modified*

There are therefore only three of a signal's characteristics that can be changed over time: *size, phase or frequency*. However, the phase and the frequency are simply different ways of visualizing or measuring the main changes in the signal.

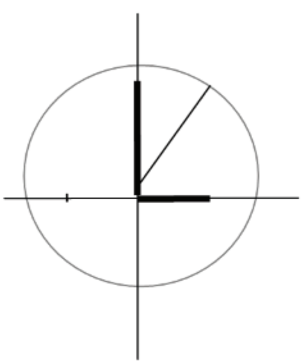
– For AM, the amplitude of a high-frequency conveyor signal (carrier) is changed in proportion to the instantaneous amplitude of the message's modulation.

– FM is the most popular analog modulation technique used in communications systems; the amplitude as well as the modulating carrier is kept constant, while its frequency is changed by the message's modulation signal (this frequency excursion may be even greater, the higher the central frequency).

– Amplitude and phase may be modulated simultaneously and separately, but it is difficult to generate and detect. In its place, in practical systems, the signal is separated into another set of independent components: 1 (*in-phase*) and Q (*quadrature*). These components are *orthogonal*, so they do not interfere with one another.

1.3.2. *Amplitude and phase representation in the complex plane*

The periodicity of the frequency response is clear; since after completing a circle (2 π: frequency variation from 0 to 1, or 0 to F_e), the vector will be found in the same position (Figure 1.7.).



$z = a + ib$

Amplitude (module):

$\rho = \sqrt{a^2 + b^2}$

2) Phase (argument):

φ = arctang (b/a) +/- k π

Remember:

$z = \rho * e^{i\varphi}$

$z = \rho(\cos\varphi + i.\sin\varphi)$

Figure 1.7. *Polar/rectangular (Cartesian) conversion*

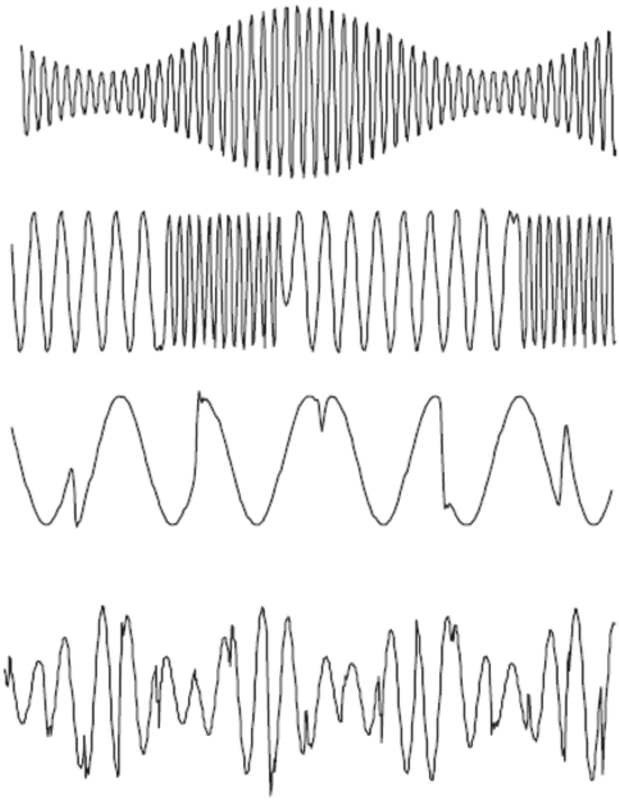


Figure 1.8. *Amplitude, frequency, phase and/or amplitude shift-keying*

Digital transmission systems broadcast information between a source and a receiver or an intended recipient using a physical support such as a cable, optical fiber or propagation along a radioelectric channel. The signals transported may be, *ab initio*, digital in origin, as in data networks, or analog in origin (speech, images, etc.). Where it is analog, it is then digitized. The transmission system should therefore convey signals containing the information from the source to the receiver, with as few errors as possible; this is *the domain of reliability*.

Below is a *synoptic schema of a modulator*, placed in a global digital transmission system: modulator/demodulator.

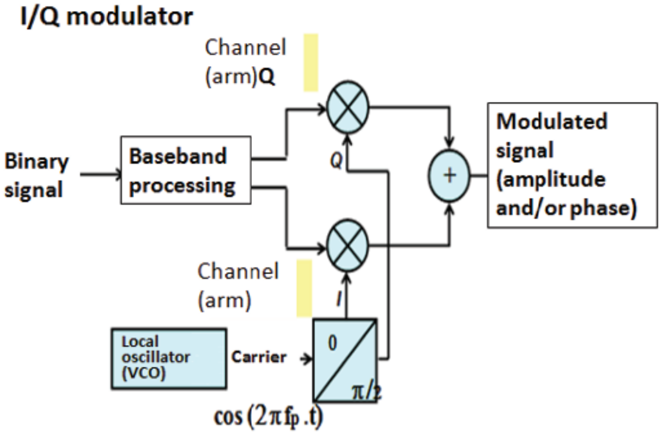


Figure 1.9. *General schema of a modulator*

– The source sends a digital message in the form of a series of bits (binary digits).

– The *coder* can sometimes delete binary elements, *a priori* insignificant (data *compression* or *source coding*; compression only being possible if there is redundancy), or, on the contrary, addition of *redundancy* to information with a view to protecting it against *noise* and parasites present in the transmission channel (*channel coding*). Channel coding it only possible if the flow from the source is less than the transmission channel's capacity (*the probability of error Pe* tends, in this case, toward 0, according to research by Hartley–Shannon).

– Modulation has the role of adapting the signal spectrum to the channel (physical medium) on which it is sent.

– On the side of the receiver, the demodulation and decoding functions are in fact the respective inverses of the modulation and coding functions on the side of the receiver.

There is an essential characteristic that makes it possible to compare the different transmission techniques with one another:

– *The probability of error Pe* per bit transmitted makes it possible to evaluate the quality of a transmission system. It depends on the transmission technique used as well as the system source/coder/modulator/channel.

1.4. Probabilities of error

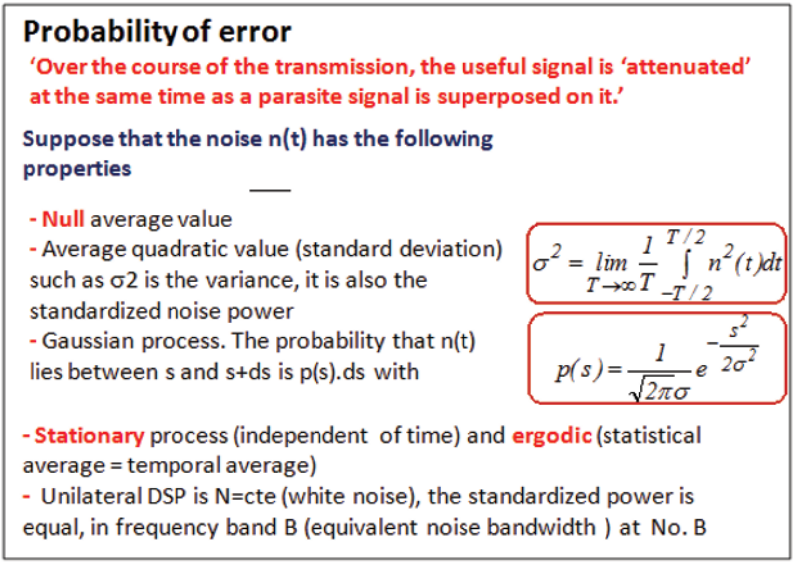


Figure 1.10. *Probability of error. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

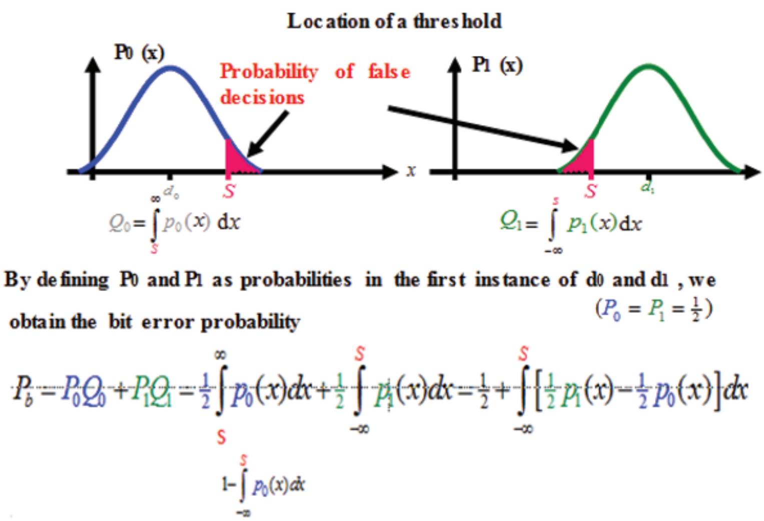


Figure 1.11. *Probability of wrong decisions. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

1.4.1. *Bit error ratio versus signal to noise ratio*

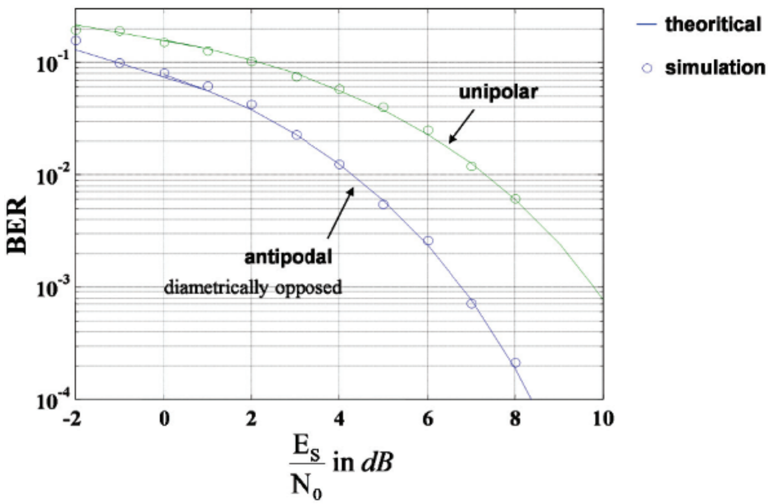


Figure 1.12. *Error rate by bit, for a unipolar and antipodal transmission, according to the signal to noise ratio. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

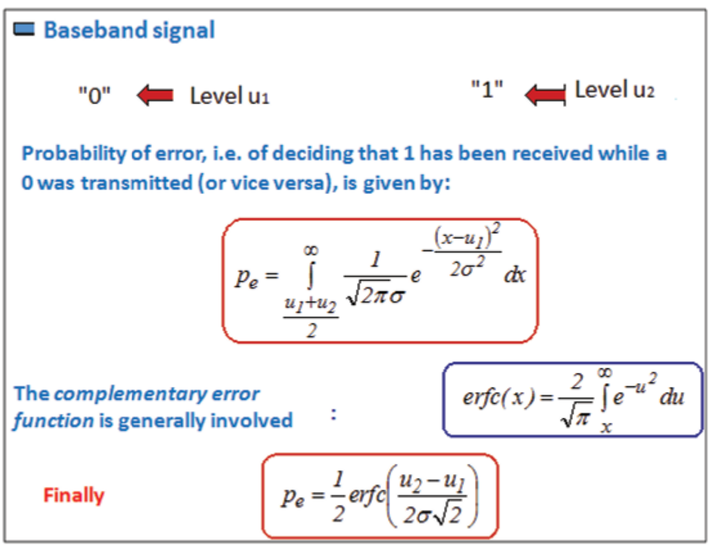


Figure 1.13. *Probability of error in erfc (erf complementary). For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

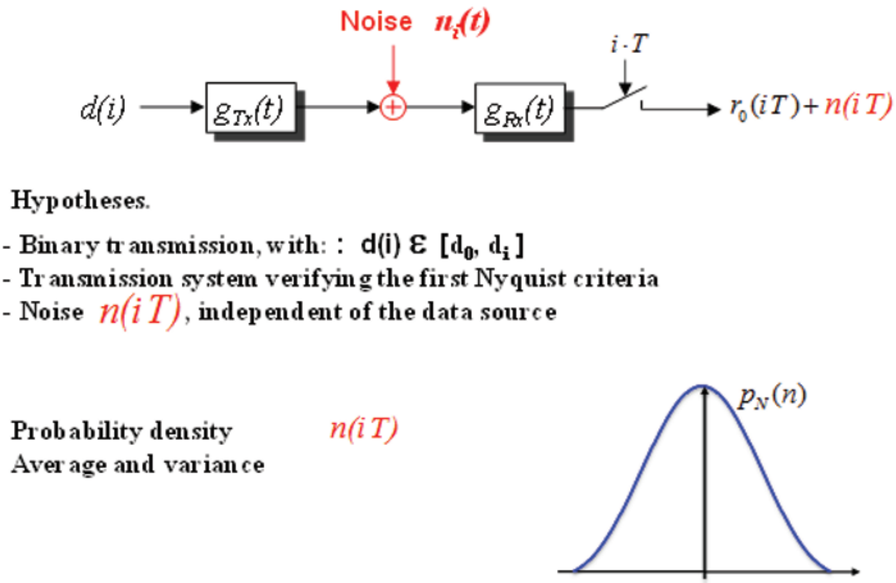


Figure 1.14. *Probability of error by bit. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

1.4.2. *Demodulator: intended recipient decoder*

The three essential characteristics that make it possible to compare the different techniques with one another are as follows:

– *The probability of error Pe* per bit transmitted is an *important criterion for the quality* of a transmission system. It depends on the transmission technique used, as well as *the channel on which the signal is transmitted*. Pe is a theoretical value, for which an unbiased estimation, in the statistical sense, is the *Bit Error Rate*: BER.

– *The spectral occupation* of the signal sent should be known so the passband of the transmission channel can be used effectively. The *current trend is increasingly to use* modulations with high spectral efficiency.

– *The complexity of the receiver*, which is there to reform the signal sent, is the third major aspect of a transmission system.

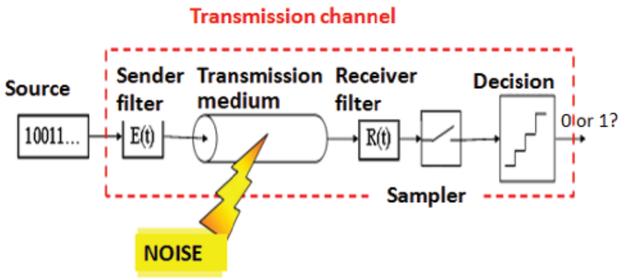


Figure 1.15. *The transmission chain*

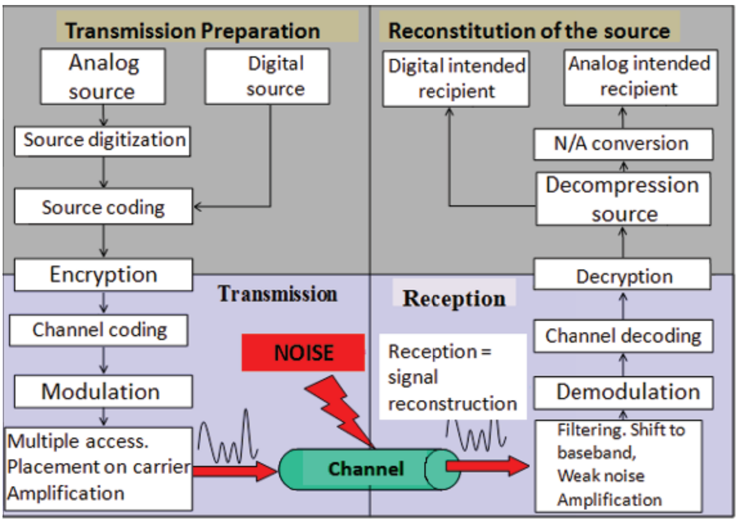


Figure 1.16. *Ordinogram of a transmission chain*

The purpose of modulation is therefore to adapt *the signal to be sent to the transmission channel.* This operation consists of "*playing*" on one or more parameters of a carrier wave $S(t) = A\cos(_c t + \varphi_c)$ centered on the channel's frequency band.

The modifiable parameters are therefore:

– the amplitude: A;

– the frequency: $f_c = \omega_c/2\pi$;

– the phase at origin: φ_c.

In *binary* modulation, information is transmitted with the aid of a parameter that takes only two possible values.

In the modulation procedures M-*ary* (M = 2: binary modulation; M-ary, e.g. bi-nary), information is transmitted with the aid of a parameter that takes *M values*. This makes it possible to link a *word* of n binary digits to a modulation state. The number of states is therefore $M = 2^n$. These n (= $\log_2$ M) digits result from breaking into packets the n digits from the *bit stream* that come from the coder.

The types of modulation most frequently used are the following:

– phase-shift keying: PSK;

– differential phase-shift keying: DPSK;

– quadrature amplitude modulation: QAM;

– frequency-shift keying: FSK;

– amplitude-shift keying: ASK.

1.5. Vocabulary of digital modulation

Symbol

A symbol is an element of an *alphabet*. If M *is the size of the alphabet*, the symbol is then called *M-ary*; *important example*: *M = 2*, the symbol is called *binary*. By grouping, in the form of a block, *n* independent binary symbols, we obtain an alphabet of $M = 2^n$ M-*ary* symbols. Thus, a symbol M-*ary* conveys the equivalent of $n = \log_2 M$ bits.

Modulation speed R

This is the number of state changes per second of one or more parameters modified simultaneously: a phase change in a carrier signal, a frequency excursion and a variation in amplitude are state changes.

The “modulation speed” R = 1/T is expressed in “bauds” (see the work of Émile Baudot, a French telegraphy engineer (1845–1903)).

Binary flow D = 1/T

This is defined as the number of bits transmitted per second; it is therefore expressed in “bits per second”. It is equal to or greater than the modulation speed at which a state change will represent a bit or group of bits.

For an M-ary alphabet, we have the fundamental relationship: $T = nT_b$ which is D = n R. There is an equality between source flow and modulation speed only in the case of a binary source (binary alphabet).

The quality of a link is connected to the *bit error rate*.

BER (bit error ratio)

This is the number of false bits/number of bits transmitted.

A priori, Pe and BER are different. In a statistical sense, we have Pe = E(BER) (expected value; BER tends toward Pe if the number of bits transmitted tends toward infinity).

Spectral efficiency

The spectral efficiency of a modulation is defined by the parameter: $\eta = D/B$.

It is expressed in “bit/second/Hz”. *D* is the “*binary flow*” and *B* is the width of the band occupied by the modulated signal. For a signal using M-ary symbols, we have: $\eta = 1/T\ B \log_2 M$ bit/sec/Hz. For given *B* and *T*, the *spectral efficiency* increases with the number of bits/symbol $n = log_2M$. Hence the M-ary modulation.

Spectral power density

As for the power spectral density (PSD) of the modulated signal m(t), some signal theory formulae remind us that if $\alpha_m(t) = x_c(t) + jx_s(t)$ represents the signal in baseband of $m(t) = Re[\alpha_m.\exp(j(\omega_c t+\varphi_c))]$ (see below), and if

$\gamma\alpha m(f)$ is the PSD of α m.(t) of the modulated signal m(t), then the PSD of the modulated signal will be:

$$\gamma m(f) = 1/4\ [\gamma_{\alpha m}\ (f - f_c) + \gamma_{\alpha m}\ (f + f_c)].$$

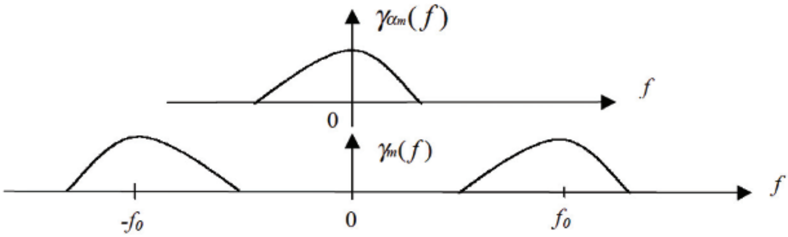


Figure 1.17. *Power spectral densities (low pass type)*

The PSDs for the (so-called) online codes are of the low frequency type. Indeed, the radio channel is RF. The problem is solved using an RF carrier.

As we will see later, different digital modulations PSDs are represented in dB (decibels) and the frequencies are positive. They display a central lobe at low frequencies, and weaker secondary lobes at higher frequencies (from 8 to 14 dB). The power is therefore essentially carried by the central lobe, hence the precaution of choosing the lowpass filter, so as not to alter, or to alter only slightly, this main lobe.

When the modulation is linear, carrying out modulation has the effect, in most cases, of translating the PSD of the modulating signal.

The PSD of the modulated signal m(t) is linked to the wave shape g(t) (which will often be rectangular) by its Fourier transform (FT): G(f).

In fact, the PSD is the FT of the *auto-correlation function* of g (see *temporal convolutions*).

Symbol clock

The symbol clock represents the exact frequency and synchronization of the transmission of different symbols. In symbol clock transmissions, the carrier transmitted is at the correct amplitude/phase values (I/Q) to represent a specific symbol (a specific point in the constellation).

Binary flow and symbol rates

The binary flow is the frequency of a bit stream of the system. For example, a series using system binary radio at 8 bits sampling at 10 kHz for the voice. The binary rate, the base radio flow rate, would be 8 bits multiplied by 10 Kbits per second or 80 Kbits per second (we will ignore, *a priori*, the extra bits needed for synchronization, error correction, etc.).

1.6. Principles of digital modulations

The message to be transmitted is issues from a binary source. The modulating signal, obtained after coding, is a (so-called) *baseband* signal, which may be complex and written as:

$$Z(t) = \sum_k Z_k.g(t - kT) \text{ with } Z_k = (I_k + jQ_k).$$

$$Z_K(t) = I_k(t) + jQ_k(t).$$

The function $g(t)$ is a *wave shape* considered in the interval [0, T[with: $kT < t < (k + 1)T$.

In ASK, PSK and QAM modulations, the modulation transforms this signal $c(t)$ into a modulated signal $m(t)$ so that:

$$m(t) = \mathrm{Re}[\sum_K Z_k(t)\, e^{j(\omega_c t + \phi_c)}]$$

The frequency $f_c = \omega_c/2\pi$ and the phase at origin (in time) φ_p characterize the sinusoidal carrier, with phase $\omega_c + f_c$ used for the modulation. If $Z_k(t) = I_k(t) + jQ_k(t)$ are *real* ($Q_k(t) = 0$), the modulation is called *one-dimensional* and if they are complex, the modulation is called *two dimensional*.

The modulated signal is also written more simply:

$$\boldsymbol{m(t) = \sum_k I_k(t).\cos(\omega_c t + \varphi_c) - \sum_k Q_k(t).\sin(\omega_c t + \varphi_c)}$$

or indeed: $m(t) = I(t).\cos(\omega_c t + \varphi_c) - Q(t).\sin(\omega_c t + \varphi_c)$,

positing: $I(t) = \sum_K I_k(t)$ and $Q(t) = \sum_k Q_k(t)$

The signal $I(t)$ modulates the amplitude of the in-*phase* carrier: $\cos(\omega_c t + \varphi_c)$ and the *s*ignal Q(t) modulates in amplitude the *quadrature carrier*: $\sin(\omega_c t + \varphi_c)$.

In most cases, the elementary signals $I_k(t)$ and $Q_k(t)$ are identical almost to a coefficient and they use the same impulse form $g(t)$ also called "*train*".

$$I_k(t) = I_k.g(t - kT) \text{ and } Q_k(t) = Q_k.g(t - kT)$$

The signals $I(t)$ and $Q(t)$ are also called "*modulating trains*" and are written as:

$$I(t) = \sum I_k (t)g(t - kT) \text{ and } Q(t) = \sum Q_k (t)g(t - kT)$$

Symbols I_k and Q_k, respectively, take their values in the *alphabets* (A_1, A_2, … A_M) and (B_1, B_2, … B_M).

The theoretical schema of the modulator is represented in Figure 1.18.

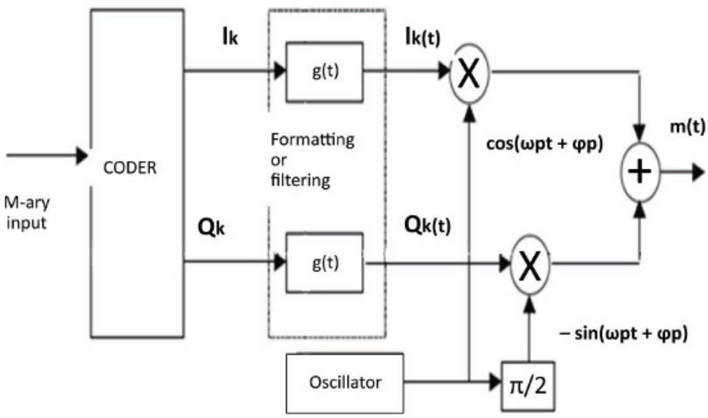


Figure 1.18. *General form of the modulator*

The various types of modulation are defined by the alphabets described above and by the g(t) function. To each symbol sent there corresponds an elementary signal of the form:

$$m_k(t) = I_k.g(t - kT).\cos(\omega_c t + \varphi_c) - Q_k.g(t - kT).\sin(\omega_c t + \varphi_c),$$

that can be represented (see Figure 1.20) in a two-dimensional space whose base vectors are: $g(t - kT).\cos(\omega_c t + \varphi_c)$ and $- g(t - kT).\sin(\omega_c t + \varphi_c)$.

1.6.1. *Polar display*

The amplitude and phase are represented together, simply, in a *polar* diagram, and the carrier is the *reference*. Just as for frequency and phase; these are measured in relation to a reference signal, the carrier in most communication systems. The amplitude is an absolute or relative value; both are used in digital communication systems.

Polar diagrams are the basis for many displays used in digital communications, although it is usual to describe the signal vector using its rectangular coordinates: I (*in-phase*) Q (*quadrature*).

1.6.2. *Variations of parameters: amplitude, phase, frequency*

Figure 1.19 shows different forms of polar *modulation*. The phase is represented as an angle. AM changes only the amplitude of (modulates) the signal. The PM changes only the phase (the argument) of the signal. *AM and PM* can be used together. FM seems similar to PM, although it is the frequency that is the command parameter, rather than the relative phase.

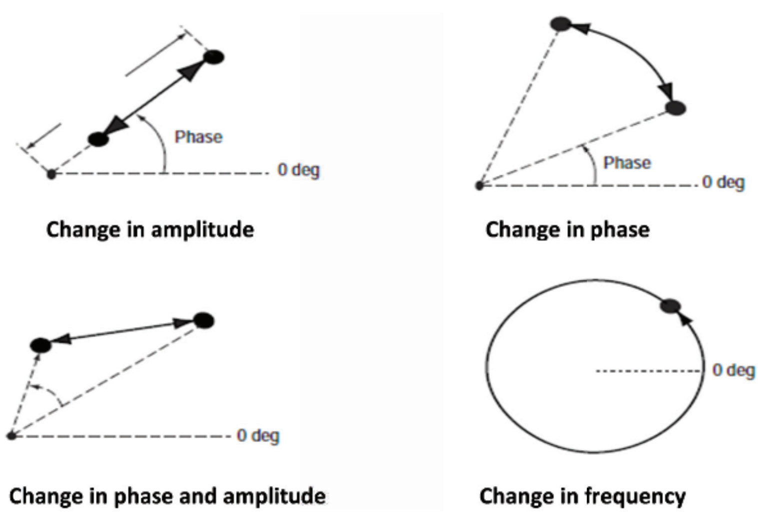


Figure 1.19. *Variations: amplitude, phase, frequency*

It is important to note the difference between the bit flow and the symbol flow. The signal bandwidth needed for a given communication channel depends on the flow of symbols, *not* on the binary flow. The flow of bits is the flow frequency of bits in a system.

1.6.3. *Representation in a complex plane*

The modulated signal $m(t)$ integrates *independent* information via $I_k(t)$ and $Q_k(t)$, which are two baseband signals, respectively, called in-phase and Q. The recovery of $Ik(t)$ and $Qk(t)$ could be achieved only if both these signals are in a band limited at the [-B,B] with $B < fp$ (Rayghley condition).

A representation in complex plane corresponds one point to the various types of modulation. The whole group linked to the symbols is called a *constellation*.

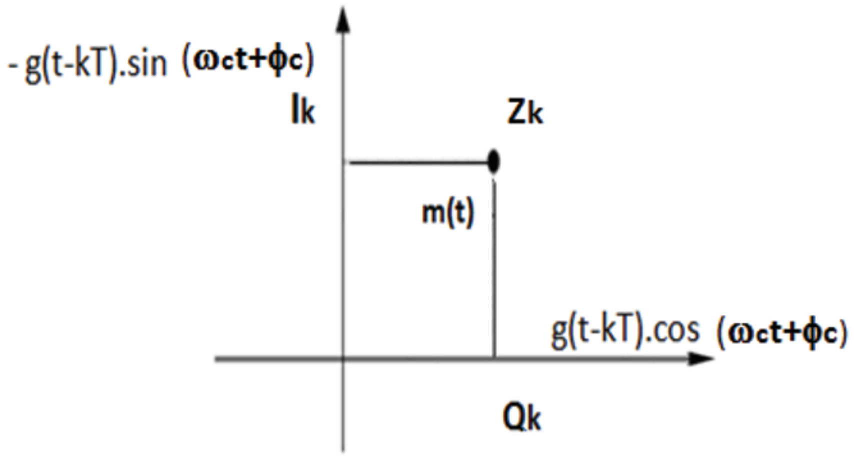


Figure 1.20. *Position of a symbol in complex plane (from Fresnel)*

The distribution of points depends on the following criteria:

– To be able to distinguish between two symbols, one should respect a minimum distance d_{min} between the points representing these symbols. The greater the distance, the lower the probability of error. The minimum distance between all these is:

$$d_{min} = \mathrm{Min}(d_{ij}) \text{ with } d_{ij} = |Ci - Cj|^2$$

– This should be compared to the definition of the Hamming distance.

A CODE'S HAMMING DISTANCE.– This is the criterion that makes it possible to evaluate a code's detecting power as well as its corrective power. The Hamming distance *between two words* (noted d_h) = the number of positions that have distinct values, for example: (110011, 101010) = 3: the number of 1 of the exclusive OR (XOR). A *code*'s Hamming distance C (written $D_h(C)$) = the minimum distance between two words in the code. For example: D_h = ([110, 101; 011]) = 2 and DH = ([001, 0101, 1001, 0110, 1010, 11001]) = 2.

To each signal sent, there corresponds an elementary signal $m_k(t)$ and by the same token, energy needed to transmit this symbol. In the constellation, the distance between a point and the origin is proportional to the *square root of the energy* that must be provided during the time interval [kT, (k+1)T[to send this symbol. The average power used to transmit symbols is proportional to $\sum |Ci|^2$: the peak power divided by 2.

The two criteria mentioned above are antagonistic; in fact, it is tempting, on the one hand, to extend the symbols to the maximum to decrease the *probability of error*, and, on the other hand, to make them close to the origin to *minimize the energy* needed for transmission.

The criteria for choosing a modulation are as follows:

– The constellation that, depending on the applications, highlights the low energy needed to transmit symbols or a low probability of error.

– The spectral occupation of the modulated signal.

– The simplicity of operation (with, possibly, a *symmetry* between the points in the constellation).

1.6.4. *Eye diagram*

The eye diagram represents the values at the receiver. These are sampled repetitively and are applied to the input of the vertical deviation, while the horizontal deviation is synchronized with the signal's flow. The name of this diagram comes from the fact that for many codings, the motif obtained resembles a series of eyes framed by two horizontal rails.

Several performance criteria can be deduced for this. Whether or not the signals are too long, too short, poorly synchronized with the system clock, at too high or low a level, have a great deal of noise, are very slow during state changes, or indeed if there are too many overtakings or too much inertia. An open eye indicates a signal carrying little distortion. This may be caused by intersymbol interferences (IES) or noise; the eye tends to close.

Separate eye diagrams can be produced, one for data from channel I and the other for data from channel Q. The eye diagram displays the amplitudes of I and Q depending on time, are in indefinite persistence mode, with tracings. Transitions I and Q are shown separately and an "eye" (or eyes) are formed when the symbol is decided. The binary phase-shift keying (BPSK)/*quadrature* phase-shift keying (QPSK) (Figure 1.21; see below) has four distinct states of I/Q, one in each quadrant of the circle: two levels for I, 2 levels for Q. This forms a simple eye for each I and Q. Other arrangements use more levels and create more nodes in time, where the traces pass. The example below is that of a 16-QAM signal (see below) that has four levels forming distinct eyes.

The eye is open at each symbol. A good signal has these crossing points wide open.

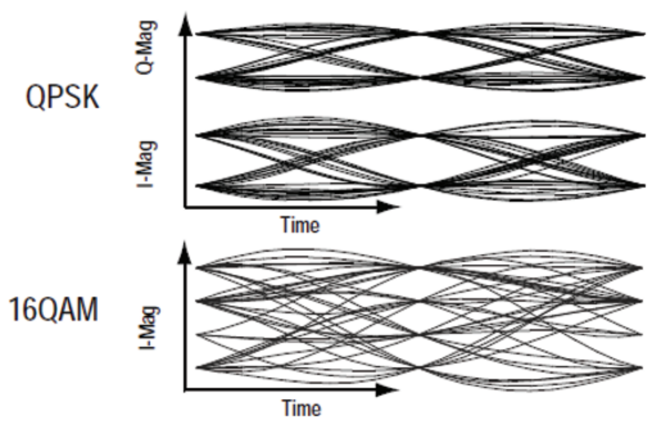


Figure 1.21. *Eye diagram: I and Q*

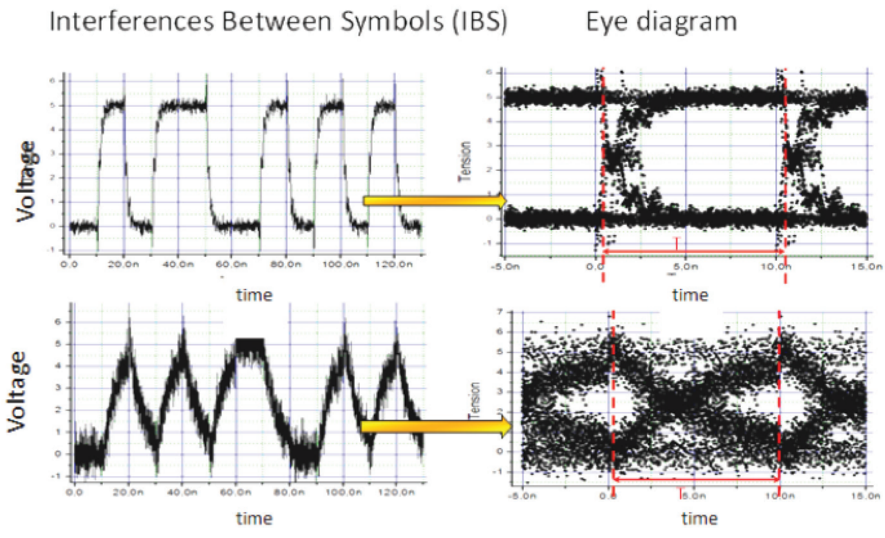


Figure 1.22. *Intersymbol interferences and eye diagrams. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

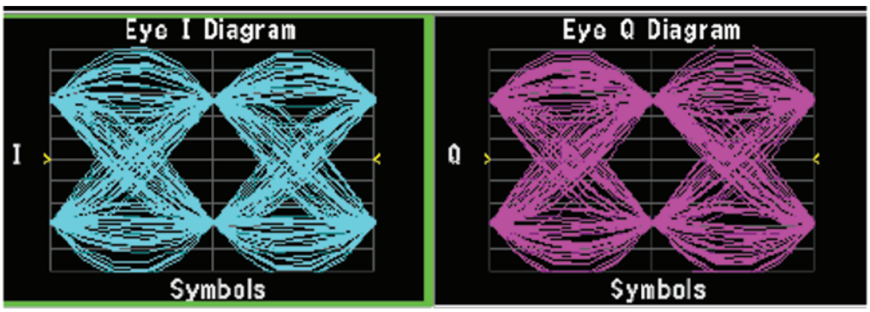


Figure 1.23. *Eye diagrams (e.g. QPSK; Agilent). For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

1.7. Multiplexing

In CDMA systems (see below), users use in *shared time*: a digital channel with very high flow, whilst also recovering at very high flow a digital sequence on their transmission. A different sequence is assigned at each output terminal so that the signals can be discerned from one another by correlating them with the superposed sequence. This is based on codes that are shared between base and mobile stations. Because of the choice of coding, there is a limit of 64 code channels on the direct line. The inverse line has no practical limit on the number of codes available.

Channel sharing

The RF spectrum is a finite resource. *Multiplexing* or channelization is used to separate different users of the spectrum.

We speak, below, of multiplexing using *frequency*, *time*, *code* and *geography*. Most communication systems are a combination of these multiplexing methods.

1.7.1. *Frequency multiplexing*

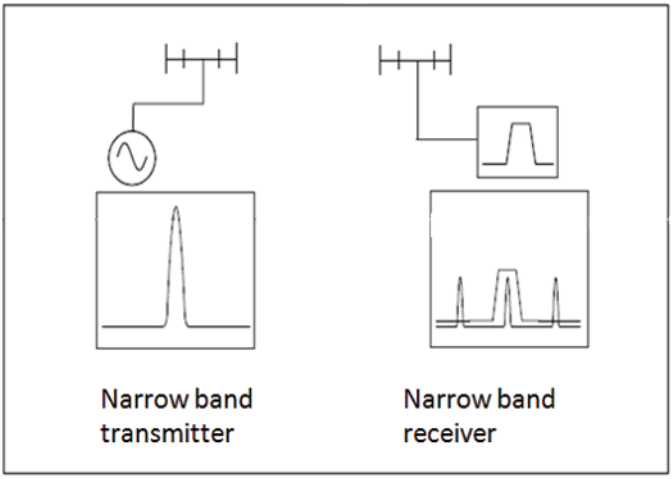


Figure 1.24. *Multiplexing frequency*

This breaks down the frequency band available into smaller channels at fixed frequency. Each transmitter or receiver uses a separate frequency. This technique has been used since the 1990s. Transmitters have a narrow or limited band. A narrow frequency band transmitter is used with a narrowband filter receiver so that it can demodulate the signal wanted and reject unwanted signals, such as signals interfering from adjacent radios.

1.7.2. *Multiplexing – time*

TDMA multiplexes several transmitters or receivers at the same frequency, i.e. separates the transmitters over time; users (A, B, C, etc., see Figures 1.23 and 1.24) have their own time interval (IT; *time slot* [TS]), thus sharing the same frequency. The simplest type is time-division duplex (TDD). This multiplexes the transmitters and the receiver on the same frequency. TDD is used, for example, in a simple two-directional radio where a button is pressed to speak and released to listen. This type of TDD, however, is very slow. Digital wireless radios such as CT2 (Cordless Telephone) and DECT (digital enhanced cordless telecommunications) use the TDD, but they multiplex hundreds of periods per second.

TDMA is also used in the digital cellular system GSM and also in the US NADC-TDMA system, the *public switched telephone network*.

If we are interested in the notion of space–time frequency, then we can define an equivalence between a group of users sharing a particular frequency band and a case where each user has access to a fraction of the band all the time.

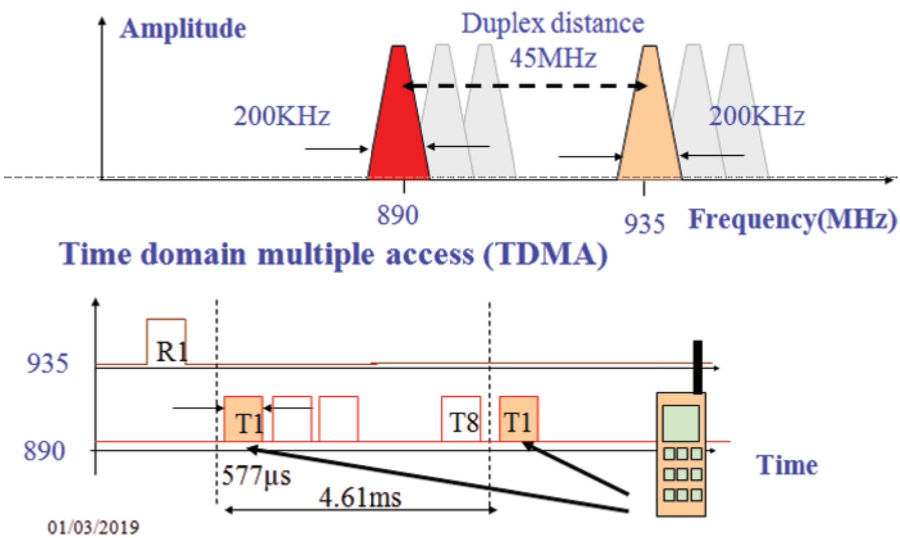


Figure 1.25. *CDMA: all users on each frequency and users are separated by code. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

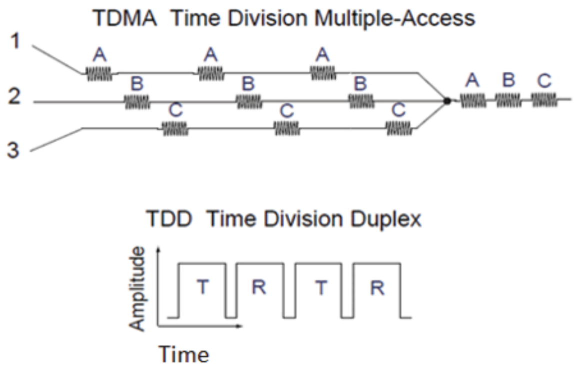


Figure 1.26. *TDMA principle*

1.7.3. *Multiplexing – code*

CDMA is a method of access where multiple users are authorized to transmit simultaneously on the same frequency; it is a code specific to each user, which discriminates between data. Frequency-division multiplexing is therefore used, but the channel is 1.23 megahertz wide. In the case of CDMA US telephones, an additional type of channelization is added in the coding format.

1.7.4. *Geographical (spatial) multiplexing*

Another sort of multiplexing: geographical or cellular. If two transmitter/receiver (transceiver) pairs are distant enough from one another, they can operate at the same frequency and not interfere with one another. There are few systems that do not use geographical multiplexing (see international broadcasting stations): clear-channel (in the US, in AM, very robust with regard to interference), amateur stations, and some low frequency military radios with no geographical limitations, broadcasting globally, etc.

1.8. Main formats for digital modulations

Here, we will discuss the main digital formats for digital modulation, their main applications, relative spectral efficiencies and some variations in modulation types used in practical systems. Fortunately, there is a limited number of modulations that form construction blocks for absolutely any system.

Table 1.1 covers different modulation formats in wireless and video communications; we will return to it.

Modulation format	Applications
MSK[1],GMSK	GSM, CDPD
BPSK	Deep-space telemetry, cable modems
QPSK, π/4 DQPSK	Satellite, CDMA, NADC, TETRA, PHS, OODC, LMDS, DVB-S, cable (return path), cable modems, TFTS
FSK, GFSK	DECT, Pagination, Rank mobile data, AMPS, CT2, HERMES, land mobile, public security
8, 16 VSB	North Americana IV digital (ATV), broadcasting, cable
8PSK	Satellite, aviation, telemetry points, broadband video surveillance systems
16 QAM	Digital microwave radio, modems, DVB-C, DVB-T
32 QAM	Terrestrial microwave, DVB-T
64 QAM	DVB-C, modems, set-top box (STB), MMDS
256 QAM	Modem, DVB-C (Europe) digital video (US)

Table 1.1. *Modulation formats and applications*

This is a digital FSK at continuous phase. Like QPSK, MSK is encoded into bits alternating the moments in quadrature, the component Q is delayed by half the duration of a symbol. But, unlike the squared signals used in QPSK, MSK modulation encodes each bit on a half-sine. We are dealing with a constant module signal, reducing problems with nonlinear distortions.

1 MSK: minimum-shift keying; the specific case of FSK ($\Delta f/D = 1/\pi$).

1.8.1. *Phase-shift keying*

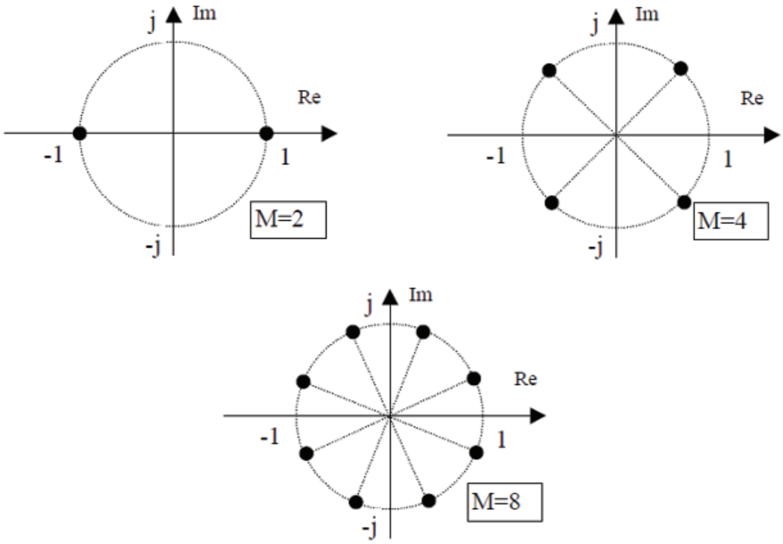


Figure 1.27. *Example of PSK modulations: constellation of symbols in phase modulation PSK-M*

It seems obvious to increase the *M* (i.e. the number of bits transmitted per symbol). This presents the following advantages and drawbacks:

– The *spectral efficiency 1/T*B* $\log_2 M$ bit/s/Hz increases for a given bandwidth *B*; but the probability of error per symbol *Ps*(*e*) then increases, and, so as not to degrade it, it will be necessary to increase the energy emitted per bit: *Eb*.

– *This type of modulation, which is simple to achieve, is scarcely used for M > 2*. Its performances are not as good as those of other modulations, as it happens, in noise resistance.

Phase shift modulations are also often known by the acronym *PSK*, for "*phase-shift keying*".

We return to the general expression for a digital modulation: $m(t) = \mathrm{Re}[\sum_K Z_k(t)\, e^{j(\omega c\, t + \phi c)}]$, with: $Z_k(t) = I_k(t) + j.\ Q_k(t)$, recalling below the allure of a "transceiver" (Figure 1.28).

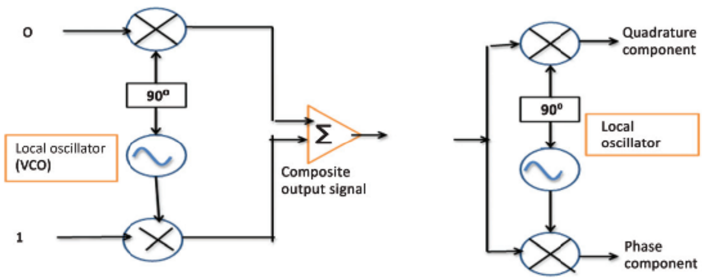


Figure 1.28. *I and Q: (a) radio transmitter; (b) radio receiver*

The elementary signals $I_k(t)$ and $Q_k(t)$ use the same waveform $g(t)$, which is here a rectangular phase, of duration T and an amplitude equal to A, if t belongs to the interval

[0, T[and moreover equal to 0 elsewhere.

We still have: $I_k(t) = I_k.g(t - kT)$ and $Q_k(t) = Q_k.g(t - kT)$

so

$$Z_k(t) = (Ik + jQ_k).g(t - kT) = Z_k.g(t - kT)$$

In this case, the Z_k symbols are distributed over a circle, hence it is written in Eulerian form as:

$$Z_k(t) = (I_k + jQ_k) = e^{j\varphi_k}.$$

Hence

$$I_k = \cos \varphi_k \text{ and } Q_k = \sin(\varphi_k)$$

and

$$I_k(t) = \cos(\varphi_k).g(t - kT)\ Q_k(t) = \sin(\varphi_k).g(t - kT).$$

One could build several PSK-Ms for a single given value of M where the symbols are arranged in any way around a circle. To improve performances over noise, the symbols must be distributed regularly around the circle (it will then be easier to distinguish between them, on average). All the possible phases are thus conveyed by the following expressions:

$\phi_k = \pi/M + 2k\pi/M$ when $M > 2$

$\phi_k = 0$ where $\pi/$ when $M = 2$

NOTE.– The Z_k symbols take their values in an alphabet of M > 2 elements $\{e^{j\phi k})$ where φk is defined above with k = 0,1,… M-1. We can also say that I_k and Q_k simultaneously take their values in the alphabet $\{\cos(\varphi k)\}$ and $\{\sin(\varphi k)\}$.

The modulated signal becomes:

$$m(t) = \mathrm{Re}[\Sigma_k \; e^{j\phi_k}.g(t\text{-}kT \; e^{j(\omega_c t+\phi_c)}] = \mathrm{Re}[\Sigma_k \; g(t\text{-}kT) \; e^{j(\omega_c t+\phi_c+\phi_k)}]$$

Considering only the interval [kT, kT+1]:

$$m(t) = \mathrm{Re}[Ae^{j(\omega_c t+\phi_c+\phi_k)}], \text{ i.e.:}$$

$$m(t) = A.\cos(\omega_c t + \varphi_c + \varphi_k)$$

$$m(t) = A.\cos(\omega_c t + \varphi_c)\cos(\varphi_k) - A.\sin(\omega_c t + \varphi_c)\sin(\varphi_k)$$

The expression above indicates that the carrier phase is modulated by the argument φ_k of each symbol, which explains the name given to PSK. It should also be noted that the carrier phase is modulated in amplitude by the signal $A.\cos(\varphi_k)$ and that the quadrature carrier $\sin(\omega_c t + \varphi_c)$ is modulated in amplitude by signal $A.\sin(\varphi k)$.

In other words, a PM can be considered an AM-C modulation (amplitude without carrier) by a binary anti-polar signal. Its bandwidth and spectrum are identical to OOK modulation (see below) unless there is no line frequency f_p (with OOK).

The PSK expression demonstrates well that it is a *constant envelope*; the envelope being the *module* of the complex envelope. This property is useful for transmissions on nonlinear channels, which makes PSK a tool of choice for satellite transmissions. The benefit of having a constant-envelope modulated signal is that it makes it possible to use amplifiers in their area of *best* performance (class E, class F), often linked to a nonlinear mode of operation.

Thus, the arrangement of the symbols on a circle does not only translate into a constant envelope, but also by an identical energy implemented to transmit each symbol. Both aspects are of course intimately linked.

1.8.2. *BPSK*

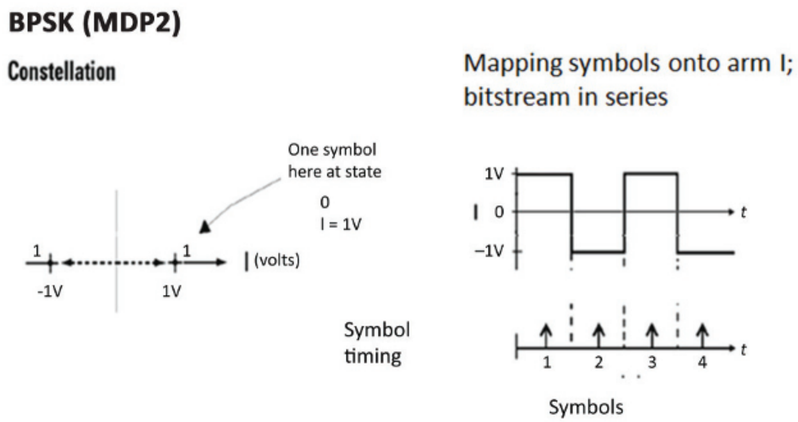


Figure 1.29. *BPSK*

Figure 1.29 is an example of a diagram where the states are mapped using *zeros* and *ones*. The symbol flow is the binary flow divided by the number of bits that can be transmitted for each symbol. If one bit is transmitted per symbol, as in BPSK, then the symbol rate is identical to the binary flow. If two bits are transmitted per symbol, as in QPSK, then the symbol rate is half the binary flow. The symbol is sometimes called the *baud* rate. Note that the baud rate is not identical to the binary flow. These limits are often confused. This is why more complex modulation formats use a higher number of states defining the information with a narrower RF spectrum band.

An important example of QPSK modulation is therefore BPSK[2] modulation. It is a binary modulation (a single bit is transmitted per period T):

$$n = 1, M = 2 \text{ and } \varphi k = 0 \text{ or } \pi$$

2 BPSK: Binary Phase-Shift Keying: phase change of the carrier wave. The BPSK digital modulation technique is the simplest form of modulation by phase shift. It uses two phases that are separated by 180°.

The symbol therefore takes its value in the alphabet $\{-1, 1\}$ $(\pm\left[e^{\,j\varphi}\right])$.

Here, the modulation only occurs on the carrier in phase I: $\cos(\omega_c t + \varphi_c)$. This is a one-dimensional modulation. The modulated signal is therefore written for t belonging to the interval

$[0, T[: m(t) = \pm A.\cos(\omega_c t + \varphi_c)$

The BPSK constellation is shown in Figure 1.30.

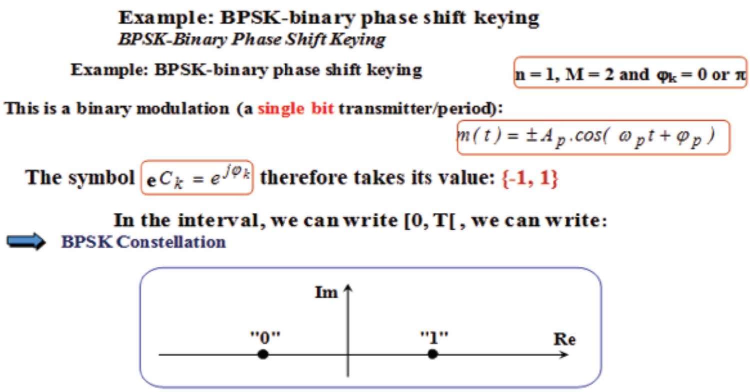


Figure 1.30. *BPSK. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

Binary input data in polar form with the symbol 1 and 0 are represented with a constant amplitude level. The process of signal transmission coding is followed by an immediate no-return-to-zero (NRZ) encoder. The amplitudes linked to 1 or 0 are $\sqrt{E_S}$ (1)) and $-\sqrt{E_S}$ (0), or the contrary.

The energy per symbol is therefore: Eb.

According to time:

$$-\sqrt{E_S} \times \sqrt{\frac{2}{T_S}}\mathrm{Cos}(2\pi f_c t) \qquad \Bigg| \qquad \sqrt{E_S} \times \sqrt{\frac{2}{T_S}}\mathrm{Cos}(2\pi f_c t)$$

.

To demodulate the original binary order of 1 and 0, the signal entering the BPSK is passed to a *correlator*, which is formed of the multiplier (between the signal received and all the possible signals) and the integrator via a rule. If the result is > 0, the device produces a "1", otherwise, 0.

BPSK is used more in satellite communications, because of its simplicity and robustness of use. Other advantages of BPSK include the improvement in bandwidth power, because it can only transmit one bit per symbol, and so cannot be used for high-flow applications.

1.8.2.1. *BPSK: modulation and demodulation*

The modulator represented in Figure 1.31 is formed of a multiplier that changes the frequency on a digital train coded NRZ.

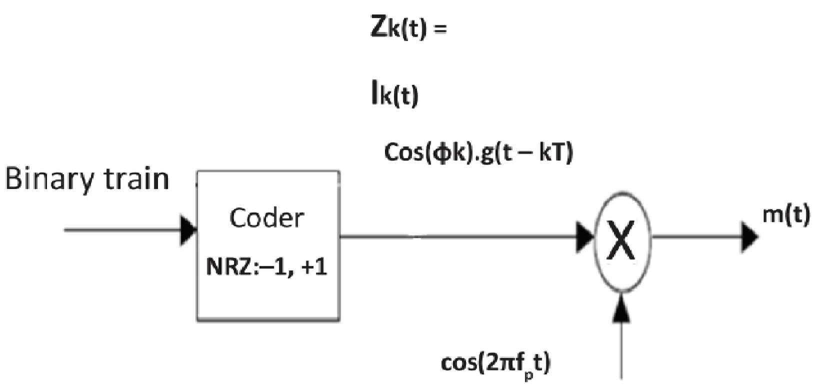


Figure 1.31. *BPSK modulator*

The receiver requires the use of a coherent demodulation (see Figure 1.33), a simplified synoptic of a BPSK demodulator.

1.8.2.2. *A small memorandum on (NRZ codes)*

Bipolar NRZ codes

NRZ: there is *no return* of the voltage to level 0 over a symbol's duration. There are two types of NRZ code: unipolar and bipolar (see Figure 1.32).

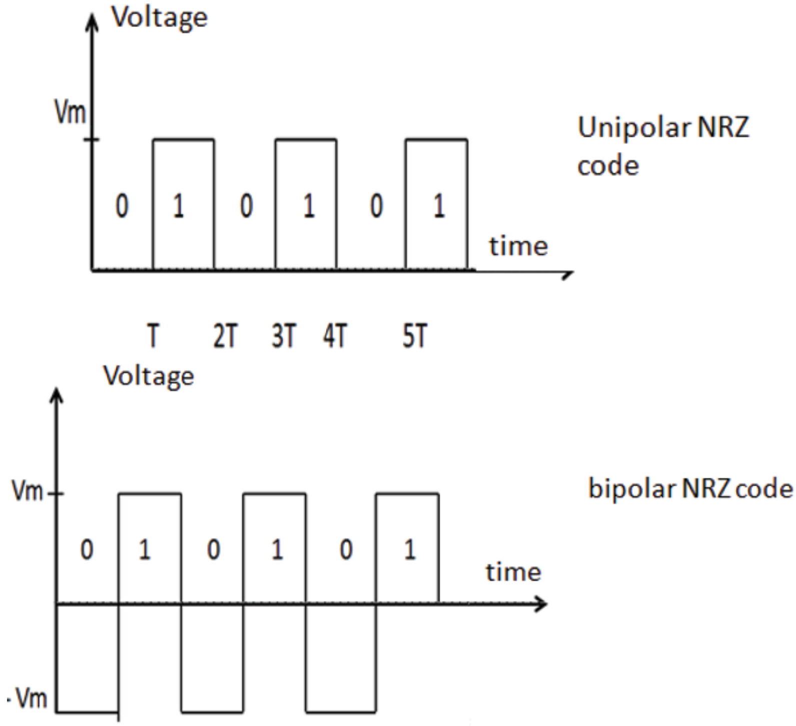
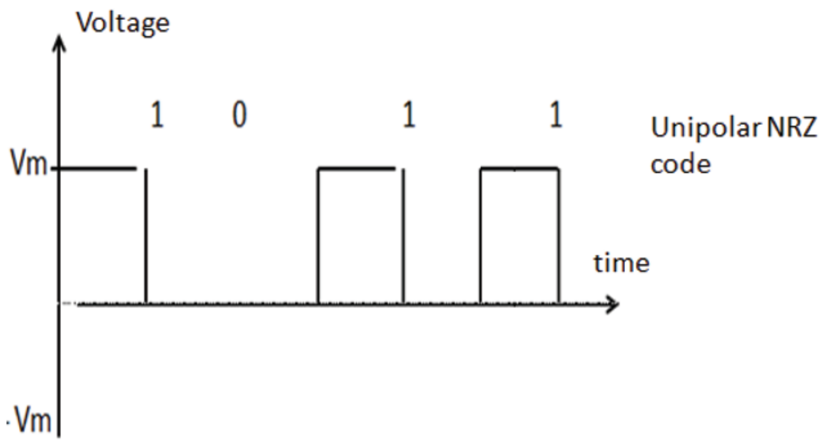


Figure 1.32. *Unipolar and biphase pulses*



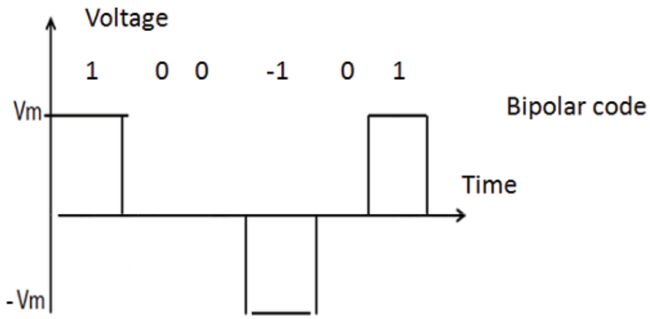


Figure 1.33. *Top: unipolar RZ code (remains at zero). Bottom: bipolar RZ code*

Manchester codes

Coding is achieved here via a *variati*on in the level of voltage:

– from –Vm to Vm: rising edge: codes at 1;

– from –Vm to Vm: falling edge: codes at 0;

– or the reverse.

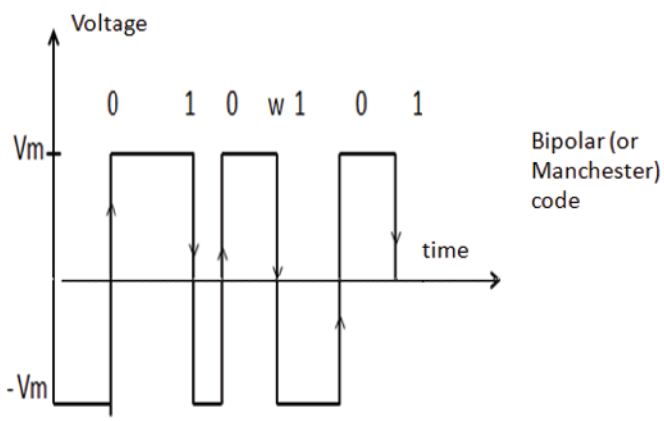


Figure 1.34. *Biphase pulses*

In the case of Manchester – or bi-phased – codes, the problems of desynchronizing the receiver are less prohibitive, as each symbol is marked by a variation in voltage.

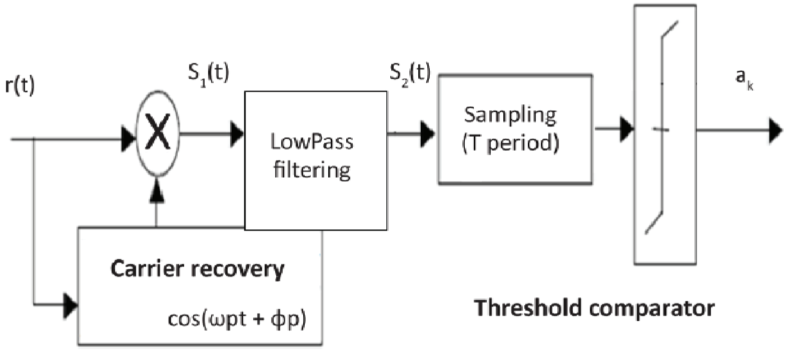


Figure 1.35. *BPSK demodulator*

If $r(t) = B.\cos(\omega_c t + \varphi_c + \varphi_k)$, the signal without noise is received by the receiver during the time interval [kT, (k + 1)T[. After multiplication with the recovered carrier, we obtain:

$$S1(t) = B.\cos(\omega_c t + \varphi_c + \varphi_k).\cos(\omega_c t + \varphi_c)$$

If, after *filtering*, to eliminate the frequency component $2f_c$: $S_2(t) = (B/2)\cos\varphi_k$.

The receiver should still recover the *rhythm* of the symbols transmitted, then *sample* the signal $S_2(t)$ in the middle of each period. Afterwards, the symbol that emitted –1 or 1, φ_k takes the value π or 0 and the sign $S_2(t)$ becomes negative or positive, showing the binary data received “0” or “1”.

1.8.2.3. *The “BPSK” spectrum*

The spectrum of the baseband signal is the power spectrum of $g(t)$ which, here, is a rectangular impulse (Figure 1.36):

$$\gamma_{am}(f) = A^2 T \left(\frac{\sin \pi f T}{\pi f T} \right)^2$$

The spectrum of the signal modulated is shifted by $\pm fp$.

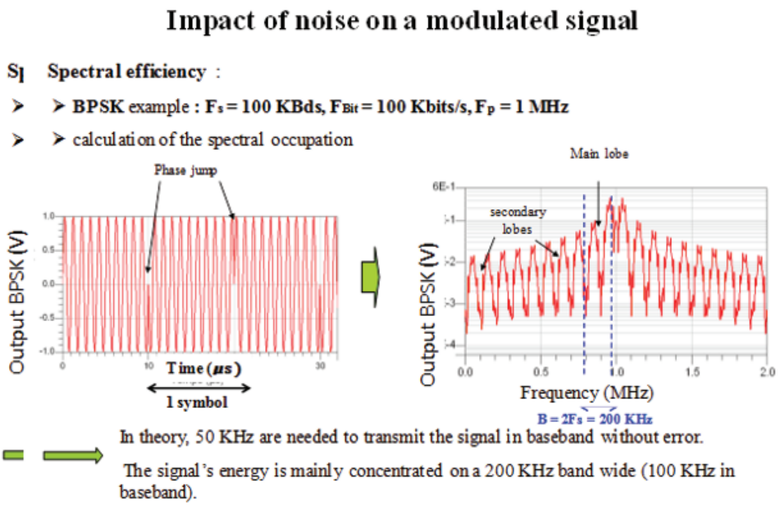


Figure 1.36. *Spectral efficiency of a BPSK. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

1.8.3. *The QPSK*

It is used intensively in applications that include CDMA, wireless local radio loop, iridium (a voice/data satellite system) and local DVB-S (Digital Video Broadcasting; videotrans). The signal is shifted in 90° increments: 45–135°, –45°, or –135°. These points are chosen as they are easy to represent, using an I/Q modulator. Only two values of Q and I are needed, giving two bits per symbol. There are four states (2^2). DPSK actually has much greater spectral efficiency (see bandwidth) than BPSK can have (B: binary), potentially a ratio of 2.

The TDMA version of the North American Digital Cell (NADC) achieves a data rate of 48 Kbits per second on a bandwidth of 30 kHz or 1.6 bits per second per hertz. The system is based on a π/4 DQPSK and transmits two bits per symbol. The theoretical efficiency should be 2 bits per second per hertz, while in practice, it is 1.6 bits per second and per hertz.

NOTE.– DQPSK: Differential QPSK: the information is not carried by an absolute state, but by the transitions between states.

Another example is a digital micro-wave radio using 16 QAM.

16 QAM (4 bits per symbol: 2^4 symbols) is more sensitive to noise and distortions than DPSK. This type of signal is often sent on a direct visibility micro-wave link (as well as an optimized high-power *transceiver*) or on a wire that generates little noise or interference. In the case of digital micro-wave radio, the bit rate is 140 Mbits per second on a broad frequency band: 52.5 MHz. The spectral efficiency is therefore 2.7 bits per second and per hertz.

It is also known as frequency splatter. Very slow changes in power waste precious transmission time as the transmitter cannot send data when it is not completely filled. This can also cause error rates on bits that are high at the start of the transmission. In addition, the peak and average power levels should be well framed, since excessive power from an amplifier can lead to *compression* or disconnection. These phenomena distort the modulated signal and usually lead to a spectrum *regrowth*.

The example in Figure 1.37 shows a differential quadrature phase shift (π/4 DQPSK) – *direct* transitions between states – as described in the standard TDMA of the NADC. In the figure, a *burst* of DQPSK is represented in 157 symbols.

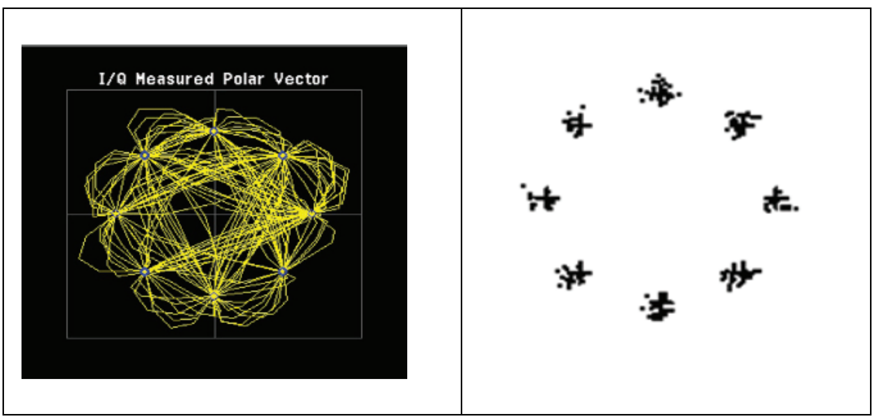


Figure 1.37. *Constellation diagram (Agilent). For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

The polar diagram shows several symbols at once. This means that it indicates the instantaneous value of the carrier at any time on the continuous line including the symbol times, represented as of I/Q or amplitude/phase values.

The constellation diagram shows an instant of this same burst, with values displayed only at points of decision.

The constellation diagram displays phase errors as well as amplitude errors at points of decision. Transitions between points of decision affect the bandwidth transmitted. This display shows the path that the carrier takes, but it does not indicate the errors at points of decision explicitly. The constellation diagrams provide specifications on variable power levels, the effects of filtering and spectrums (in video: moiré) or pixilation (aliasing).

How do we remove these interferences between symbols (IBS)? By considering the Nyquist criterion, which states in essence that it is not possible to transmit without interference between symbols a signal of symbol duration T on a bandwidth channel lower than 1/(2T).

Solution 1

If a signal is NRZ M-ary, the set of symbols is $S_i(t) = (2i - 1)V_m{*}h(t)$, with $-M/2 + 1 \leq i \leq M/2$ and h(t) being the door function (pulse); the signal will have a PSD in sinc, with limitless support.

If we replace h(t) with a sinc impulse, its PSD will have a (finite) gate width: 1/T of height $(M^2 - 1).V_mT/3$.

A signal with this PSD will not be altered by a low-pass channel of BP 1/2T; so, no IES,

This is not the case if we replace h with a sinc; the symbols are recovered as they are of infinite duration. The IES is null as PSD is limited and is the wise choice at the moment of sampling.

However, symbols of indefinite duration cannot be created in practice.

Solution 2

Instead of a sinc, we use *raised cosine* impulses:

$$f(t) = \sin\ c\left(\frac{t}{T_S}\right)\frac{\cos\left(\frac{\pi rt}{T_S}\right)}{1-\left(\frac{2\,rt}{T_S}\right)^2}$$

NOTE.– The closer r is to 1, the more f is attenuated.

Frequencies f including the module verify: (1-r)/2T ≤ $\lceil f \rceil$ ≤ 1 + r)/2Tn. Then, the PSD is the previous multiplied by 1/4.(1 + sin(πT/r(1/r.abs(f)))^2. Otherwise, the PSD is null. In this case, the PSD spreads further, the greater the size of r.

For frequencies whose module is ≤ 1-r/2T, the PSD has the same value as for the previous solution, i.e. one PSD of one sinc.

1.8.3.1. *More on quadrature phase-shift keying: towards some modulation and demodulation simulation programs*

Another extension of PSK is the digital modulation technique, of a higher order than PSK, which uses a four-level phase state to transmit 2 bits/symbol simultaneously, by selecting one of four carrier phase shifts spaced at 0, π/2, π and 3π/2, where each phase value corresponds to a distinct pair of message bits 00,01,10,11. This enables the signal to have twice the information using the same bandwidth. This means that QPSK is more effective by bandwidth than BPSK.

This is therefore an AM on two levels on each of the quadrature carriers. This has been chosen for GSM, as it produces a minimum product Q.B (Q: flow; B: spectral width).

In this case: $n = 2$, $M = 4$ and $\varphi_k = \pi/4 + k\pi/2$.

1.8.3.2. *QPSK: modulation and demodulation*

In the constellation diagram of a QPSK (Figure 1.43), there is a path from any one symbol to one of the three others. In fact, there is one chance in four that the signal's trajectory will traverse the origin (the 0 V). In this case, I and Q change. This critical case poses problems caused by great variations in amplitude exacerbated by possible nonlinearities in amplifier circuits, as these cause distortions (and *spectral regrowth*) widening modulation sub-bands.

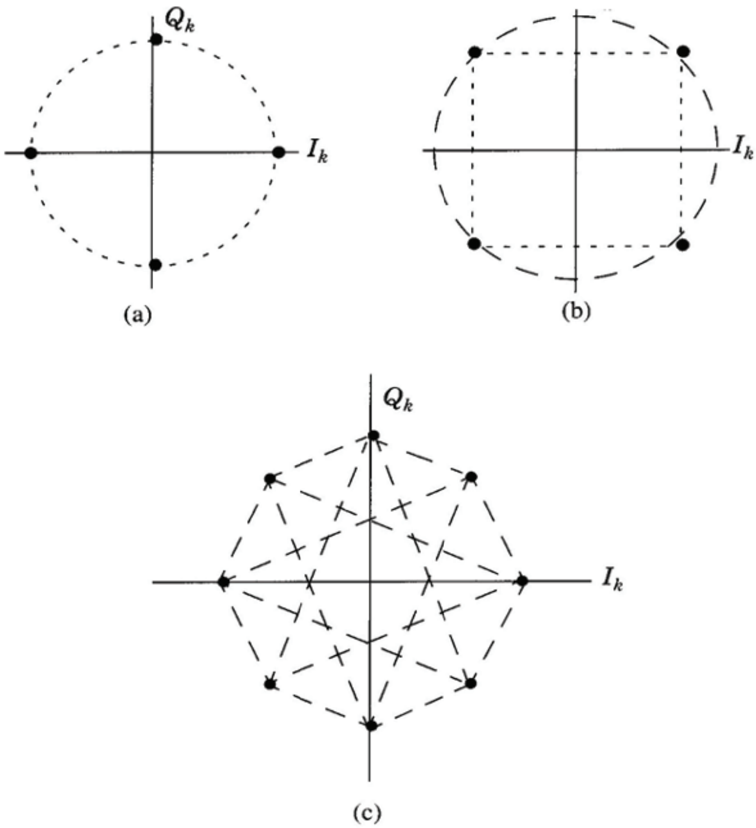


Figure 1.38. *Constellation diagram of a π/4 QPSK: (a) possible states for θ_k when $\theta_{k-1} = n\ \pi/2$; (b) possible states when $\theta_{k-1} = n\ \pi/4$; (c) all possible states*

To circumvent this problem, other modulation formats have been developed: one might consider the concept of differential modulation.

NOTE (Differential PSK or DPSK).– The information does not depend on the absolute value of the state, but on the transition between states. There may be unauthorized transmissions. This is the case for π/4 DPSK, where the trajectory does not cross the origin (0,0) (see Figure 1.38(c)). In addition, compared to GMSK, a classic cellular modulation, π/4DPSK, linked to filtering in raised cosine, has greater spectral efficiency (the filtering tends to round the corners of the signals on the right-hand side of Figure 1.39).

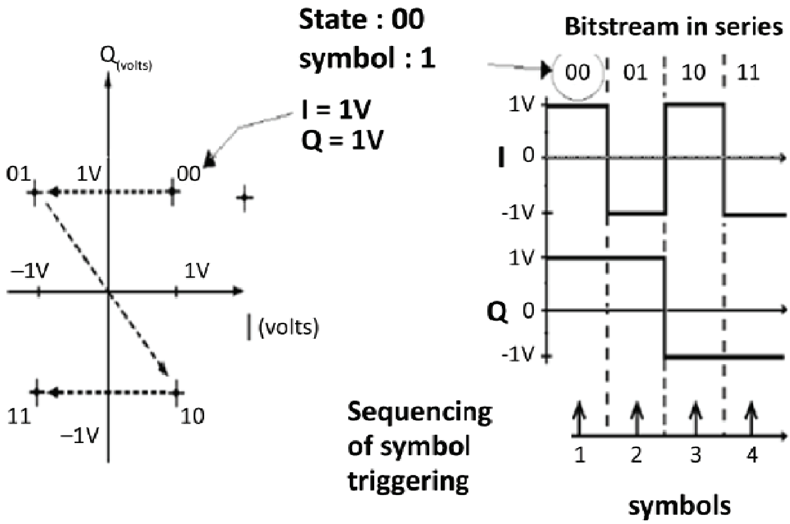


Figure 1.39. *Positions (states) in the BPSK constellation represent a motif of specific bits (symbol) and a symbol time*

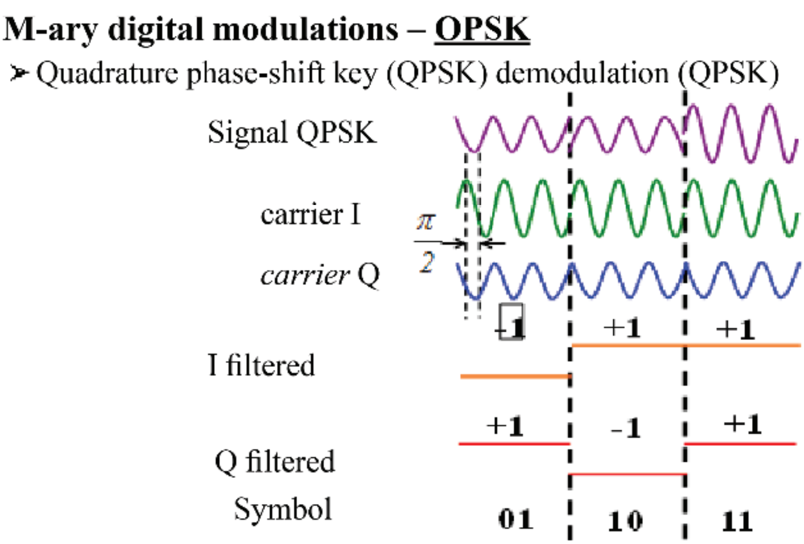


Figure 1.40. *QPSK: I/Q. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

A "QPSK" constellation

The QPSK constellation is represented in Figure 1.41. It shows that the assignment of bits at points of the constellation usually happens according to *Gray* code (modification of a single bit at a time, when a number is increased by one unit).

QPSK chronogram

Figure 1.41 represents a QPSK chronogram. It highlights the distribution of numbered bits in the incoming bit stream $\{ z_k \}$(o i_k) toward bitstreams $\{ I_k \}$ (or a_k) and$\{ Q_k \}$(or b_k) as well as the delay needed on the in-phase channel to achieve both bit flows. We also observe that the phase of the modulated signal $\underline{m}(t)$ can change from 0, $\pm\pi.2$, ου π radiants when passing from one symbol to another, which happens easily when we observe the QPSK constellation.

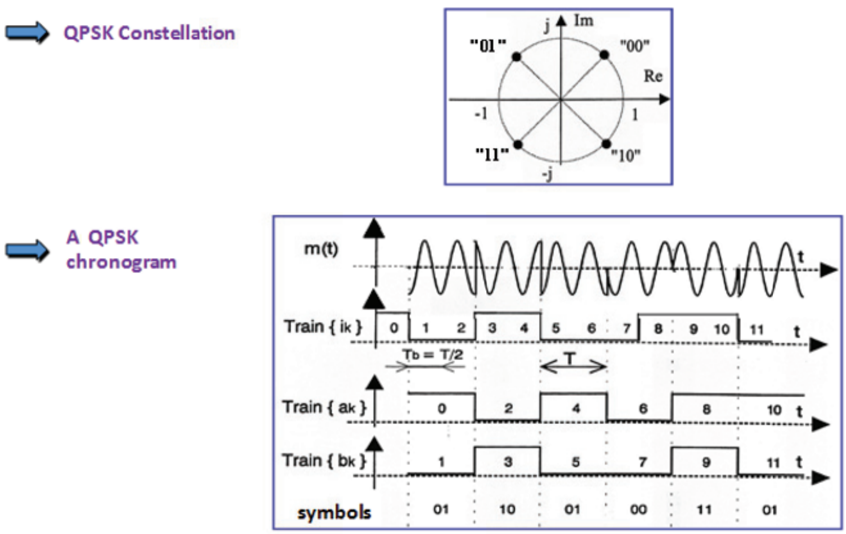


Figure 1.41. *QPSK phase modulation chronograph (Degauque/Kadionik)*

Modulation

The chronogram (Figure 1.41) highlights the simple relationship that exists between the even bit pairs and the I_k, and between the odd bits and the Q_k. For a homothety and by designating the set of values of the bit stream as $\{ Ik \}$ to the rhythm:

$$m(\mathrm{t}) = \mathrm{Re}[\Sigma_{\kappa}\ Z_k(t)\ e^{j(\omega c t + \phi c)}]$$

$$m(t) = I(t).\cos(\omega_c t + \varphi_c) - Q(t).\sin(\omega_c t + \varphi_c)\ ,$$

or only considering the time interval [kT, kT+1].

$$\mathrm{m(t)} = \mathrm{A}.\Sigma(1\text{-}2i_{2k})g(t\text{-}kT).\cos(\omega_c t+\varphi_c) - \Sigma(1\text{-}2i_{2k+1})g(t\text{-}kT).\sin(\omega_c t + \varphi_c),$$

$$m(t) = A.I_k.\cos(\omega_c t + \varphi_c) - A.Q_k.\sin(\omega_c t + \varphi_c)$$

The incoming bit stream { Zk } (or{ ik }) is divided into a { I_k }, or { a_k } bit stream, switched to the in-phase channel for the even bits, and a { Q_k }, or { b_k } bit stream, switched to the quadrature channel for the odd bits. The speed of bit streams { I_k } and { Q_k } is therefore half the speed of the incoming bit stream { ik }.

The synoptic schema of the modulator shown in Figure 1.42 shows the demultiplexing of the bit stream at the entry of the modulator in two bit streams on the in-phase and quadrature channels. Both bit streams are therefore coded in NRZ. The paths of the schema represent the relationship:

$$\underline{m}(t) = I(t).\cos(\omega_c t + \varphi_c) - Q(t).\sin(\omega_c t + \varphi_c)$$

and therefore calls for two multipliers.

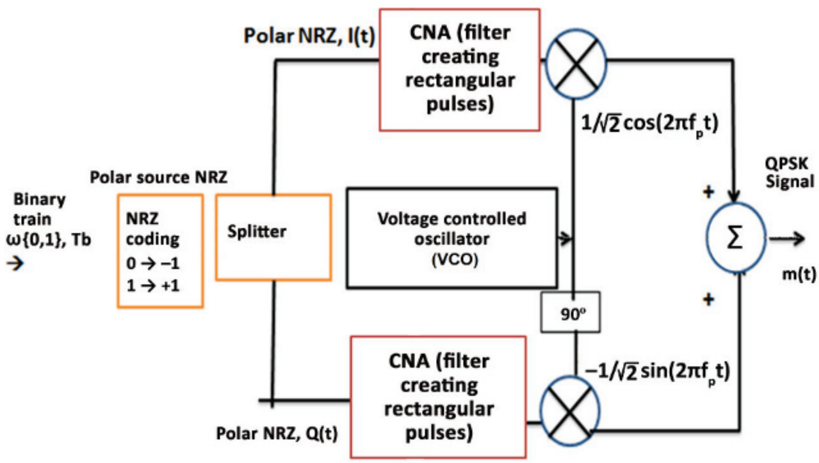


Figure 1.42. *QPSK modulator. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

$s_i\ (t) = \mathrm{A}\cos(2\pi f_p \mathrm{t} + \theta_i);\ \ 0 \leq \mathrm{t} \leq \mathrm{T},\ \mathrm{i} = 1,2,3,4$

with

$$\theta_i = \frac{(2i-1)\pi}{4}$$

$$s_{i1}\ (t) = \sqrt{E}\cos(\ \theta_i),$$

$$s_{i2} = \sqrt{E}\sin(\theta_i)$$

And so: $\theta_i = \tan^{-1}\frac{s_{i2}}{s_{i1}}$

PSD: $\gamma = A^2 * (2\mathrm{T}_B) * (\frac{\sin(2\pi \mathrm{f} T_B}{2\pi \mathrm{f} T_B})^2$

A QPSK, via MATLAB, SCILAB or OCTAVE

```
% function [s,t,I,Q] = qpsk_mod(a,fc,OF)

clear all

fc = 1000
fs = 20e3;
OF = 8
fs = OF*fc
t = 0:1/fs:0.1;
w = 7.5e-3;

z0 = [1,0,1,1,0,0,0,1,1,1]
z = 2*z0-1
z_even = [z(2),z(4),z(6),z(8),z(10)]
z_odd = [z(1),z(3),z(5),z(7),z(9)]

 %ak = [1,1,1,1,1,1,]
N = length(z0)
z = 2*z0-1
x = rectpuls(t,w);

%xfutur = zeros(N,1001)
%tpast = -45e-3;
```

```
xfutr0 = 0;
Bip0 = 0;

NRZ0 = 0
NRZ_even0 = 0
NRZ_odd0 = 0

%xpast = rectpuls(t-tpast,w)*(-1);
N2 = N/2
for k = 1: N
tfutr = 15e-3;

Bip = Bip0+(rectpuls(t-(tfutr*(k-1)),w))/1;

xfutr = z0(k)*(rectpuls(t-(tfutr*(k-1)),2*w))/1+xfutr0;
NRZ = NRZ0+z(k)*(rectpuls(t-(tfutr*(k-1)),2*w))/1;

xfutr0 = xfutr;
Bip0 = Bip;
NRZ0 = NRZ;

end

SIZE_NRZ = size(NRZ)

for k = 1: N2
tfutr = 15e-3;


% xfutr = z0(k)*(rectpuls(t-(tfutr*(k)),w))/1+xfutr0;

NRZ_even = NRZ_even0+z_even(k)*(rectpuls(t-(tfutr*(k-1)),2*w))/1;

NRZ_odd = NRZ_odd0+z_odd(k)*(rectpuls(t-(tfutr*(k-1)),2*w))/1;
xfutr0 = xfutr;
NRZ_odd0 = NRZ_odd;
NRZ_even0 = NRZ_even;

end
```

```
Figure1 = Figure
plot(t,Bip)
% plot(t,x,t,Bip)*
ylim([-0.2 1.2])

Figure2 = Figure
plot(t,xfutr)
ylim([-0.2 1.2])

Figure3 = Figure
plot(t,NRZ)
ylim([-1.2 1.2])
```

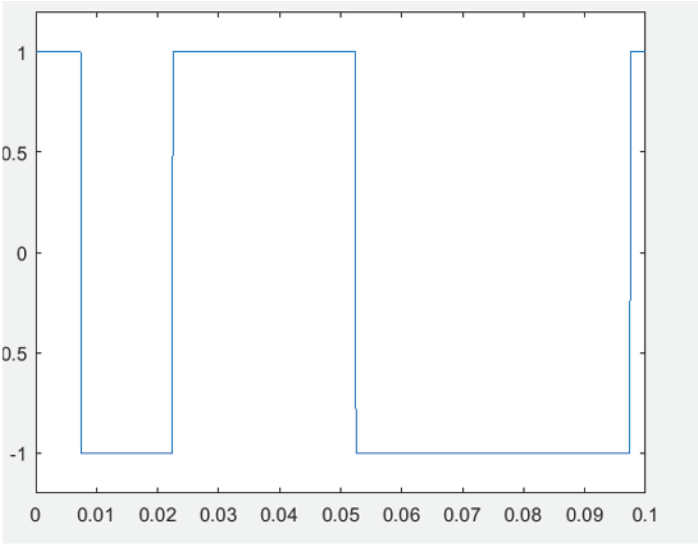


Figure 1.43. *NRZ*

```
 % SEQUENCE (to be concatenated with the
previous section)
 =====================

% Function to modulate an incoming binary flow, using a conventional
QPSK
% input: binary data flow (0 and 1) to modulate
```

```
%w = 5e-3
dim = 2001
% OF: -oversampling factor -, (multiples of fc) – takes at least 4

OF = 4
fe = OF*fc  % its sampling frequency

%t = -0.1:1/fs:0.1;
Te = 1/fe
% t = 0:w/10:10*w;
t = 0:Te:(length(dim)-1)/fe;
rp = rectpuls(t,w);
length = size(t)
lengthrp = size(rp)


%s – QPSK signal modulated with carrier
%t – time base for the signal modulated through the carrier
%I – non-modulated Ichannel (no carrier)
%Q – non- modulated channel (no carrier)
L = 2*OF % number of samples in each symbol (QPSK:  2 bits per symbol)
Lfe = L/fe
% ak = 2*a-1 %NRZ encoding: 0- > -1, 1- > +1

Q = NRZ_even(1:1:end);
I = NRZ_odd(1:1:end);
% Flow of odd and even bits

SIZE_I = size(I)
SIZE_Q = size(Q)

lengthI_initial = length(I)
I = repmat(I,1,1).';
Q = repmat(Q,1,1).';
I = I(:).';
Q = Q(:).';

lengthI = length(I)
%t = 0:1/fs:(length(I)-1)/fs; % time base
```

```
n = 10
nfe = 10*fe
t = 0:1/nfe:(length(I)-1)/nfe;

iChannel = I.*cos(2*pi*fc*t);qChannel = -Q.*sin(2*pi*fc*t)
        s = iChannel + qChannel;
      % Base band     signal modulated by QPSK


Pelot = 1; Pelot = 1; %Pelot = 0, if you do not intend to see shape plots
                                  if Pelot == 1, % Wave form at the transmitter
Figure;subplot(3,2,1);plot(t,I,'LineWidth',2);
% Wave form in base band on the I arms
% zoomed on the first symbols
 xlabel('t'); ylabel('I(t)- baseband');xlim([0,0.0125]);
%%
% _ el('t'); ylabel('I(t)-  baseband');xlim([0,10*Lfs]);
subplot(3,2,2);plot(t,-Q,'LineWidth',2); % Wave
form in base band on the Q arms
% zoomed on the first symbols
xlabel('t'); ylabel('Q(t)-
baseband');xlim([0,0.0125]);
subplot(3,2,3);plot(t,iChannel,'r');%I(t) with carrier'
xlabel('t'); ylabel('I(t)- with carrier');xlim([0,0.0125]);

subplot(3,2,4);plot(t,qChannel,'r');%Q(t) with carrier'
xlabel('t'); ylabel('Q(t) & carrier');xlim([0,0.0125]);
% waveform QPSK zoomed on the first symbols
%xlabel('t'); ylabel('s(t)');xlim([0,10*Lfe]);
hold on;
subplot(3,2,5);plot(t,s,'b');%s(t) with carrier'

% plot (t,s, 'g' )
xlabel('t'); ylabel('s(t)');xlim([0,0.0125]);
subplot(3,2,6);plot(t,s,'b');%s(t) with carrier'
xlabel('t'); ylabel('s(t)');xlim([0,0.0125]);

% end
```

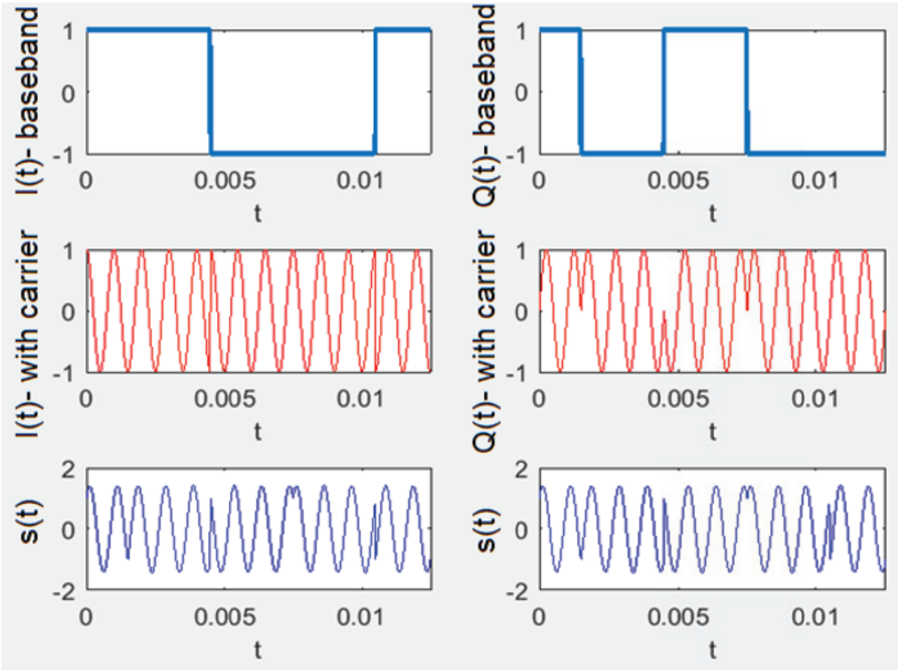


Figure 1.44. *QPSK; at the transmitter: timing diagrams. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

% continuation (to be concatenated with the previous MATLAB section)
==========================

In the demodulator, the signal received is multiplied by a reference signal (cf. coherence) and the two 2PSK are introduced into the in-phase and quadrature channels. The output multiplied from each channel is integrated. The output of the integrator is compared to a threshold value and a decision is taken. The receiver should also recover the rhythm of the symbols transmitted.

Once this has been done, the binary sequences in the ouptuts of both the in-phase and the quadrature channels are combined via the multiplexer to generate the sequence of modulated binary data. The QPSK is used for transmission using satellite applications such as videoconferencing, mobile telephone systems and other digital communications using an RF carrier. The mathematical representation of the QPSK signal is expressed as:

$$S_{qpsk}(t) = \left\{\sqrt{E_s}cos\left[(i-1)\frac{\pi}{2}\right]\emptyset_1(t) - \sqrt{E_s}sin\left[(i-1)\frac{\pi}{2}\right]\emptyset_2(t)\right\} \qquad i = 1,2,3,4$$

$$\emptyset_1(t) = \sqrt{\frac{2}{T_s}}\cos(2\pi f_c t) \ ,$$

$$\emptyset_2(t) = \sqrt{\frac{2}{T_s}}\sin(2\pi f_c t) \qquad 0 \le t \le T_s$$

where T_s is the period of the symbol, E_s is energy per symbol.

QPSK demodulation

In the same manner, we obtain for channel B:

The receiver

```
% function [a_cap] = qpsk_demod(r,fc,OF)
% Function to demodulate a conventional QPSK signal
% r – signal received upstream of the receiver
% fc – carrier frequency in Hertz
% OF – oversampling factor (multiples of fc) – at least 4 is better
% L – upsampling actor on the in-phase and
quadrature arms
% a_cap - binary flow detected


% OF = 8

fs = OF*fc; %  sampling frequency
L = 2*OF; % sampling over a duration 2Tb.
r = s;
t = 0:1/fs:(length(r)-1)/fs; % time base


x = r.*cos(2*pi*fc*t);              % Arms
% x = I.*cos(2*pi*fc*t);
y = -r.*sin(2*pi*fc*t);              % Arms Q
% y = -Q.*sin(2*pi*fc*t);
```

```
x = conv(x,ones(1,L)); % convolution; integration
over duration L (Tsym = 2*Tb)

y = conv(y,ones(1,L)); % convolution; integration
over a duration L (Tsym = 2*Tb)

%x = x(L:L:end);% Arm I – sample at each instant
of symbol Tsym
%y = y(L:L:end);% ArmsQ – sample at
each instant of symbol Tsym

a_cap = zeros(1, 2*length(x));
a_cap(1:2:end) = x.' > 0; % even bits
a_cap(2:2:end) = y.' > 0; % odd bits
doPlot = 1; % For the receiver constellation plot
% if doPlot = 1, Figure; plot(x(1:200),y(1:200),'o'); end
if doPlot = 1, Figure; plot(x(1:102),y(1:102),'o'); end
```

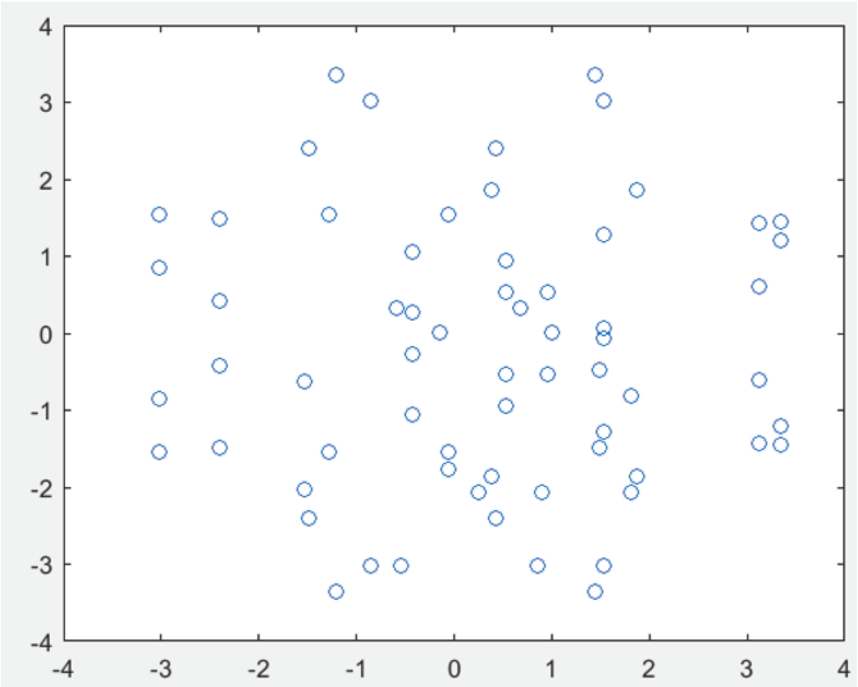


Figure 1.45. *A truncated constellation*

Coherent demodulation is applicable when the receiver has exact knowledge of the frequency and the phase of the carrier. The synoptic schema of a coherent demodulator for QPSK is shown in Figure 1.46.

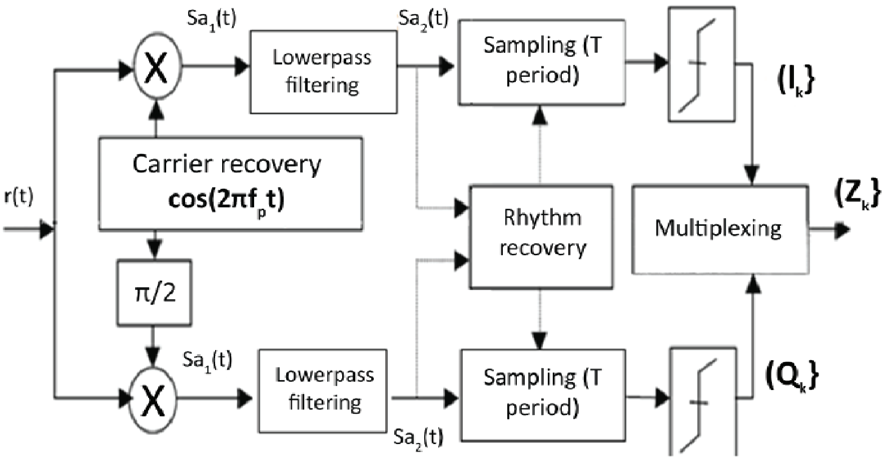


Figure 1.46. *Coherent QPSK demodulator*

The QPSK demodulator is essentially formed of two BPSK demodulators. In fact, the signal received (after potential bandpass filtering) is demodulated in two parallel channels by two quadrature carriers. Some techniques make it possible to synchronize the local oscillator with the carrier at transmission. The quadrature signal is generated from the local oscillator and a phase shifter of π/2.

If $r(t) = I_k.\cos(\omega_c + \varphi_c) - Q_k.\sin(\omega_c t + \varphi_c)$, the non-noisy signal is received by the receiver in the time interval [kT, (k + 1)T[. For channel *A* and after multiplication with the recovered carrier, we obtain:

$$Sa1(t) = [\, I_k.\cos(\omega_c t + \varphi_c) - Q_k(\omega_c t + \varphi_c)].\cos(\omega_c t + \varphi_c.$$

After filtering, we have eliminated the frequency component $2f_c$.

1.8.3.3. *A QPSK from MATLAB*

commQPSKTransmitterReceiver.m. This example shows a digital communications system using QPSK modulation. It uses communications system objects to simulate the QPSK transceiver. In particular, it illustrates methods using “wireless” in the world of real communications, such as carrier frequency and a phase offset, the re-establishment of synchronization and frame synchronization, and the time delay.

The data transmitted from the QPSK undergo fadings that simulate the effects of wireless transmission such as the addition of added white gaussian noise (AWGN). To confront this weakening, this example provides a reference design of a digital receiver including raw compensation based on a fast Fourier transform (FFT) frequency, the end compensation using a phase-locked loop (PLL); the re-establishment of symbol synchronization is based on this PLL, the frame synchronization techniques and the phase ambiguity resolution.

Three main objectives

– Modeling a wireless communication system that can recover a message corrupted by various simulated channel impairments.

– This example presents some blocks for QPSK system design from the MATLAB library, including the *raw* carrier frequency and end compensation, the *synchronization recovery* in closed circuit with bit consolidation and extraction.

– Illustrating the creation of higher level system objects, which contain other objects so as to model larger components of the system under test.

Models

The runQPSKSystemUnderTest function models this communication environment. The QPSK transceiver model is divided into three main components.

1) QPSK transmitter: produces the *bit stream*, *code*, *module* and *filter.*

2) QPSK channel: models the channel with the carrier offset, the synchronization offset, and AWGN.

3) QPSK receiver: models the receiver, including the components for re-establishing the phase, synchronization recovery, decoding, demodulation, etc.

Description of different components

Transmitter:

This component produces a message in ASCII characters, *converts these into bits* and *adds a Barker code* (oversampling by 2) for the frame synchronization sent to the receiver, plus the headers. The data are then modulated using *QPSK*, and filtered via a *square root-raised cosine* (Figure 1.52).

The payload, in bits, (of the frame) is scrambled, thus guaranteeing a balanced distribution of zeros (see an improvement in data transition density) and other zeros for achieving synchronization recovery at the receiver (see estimating the frequency offset).

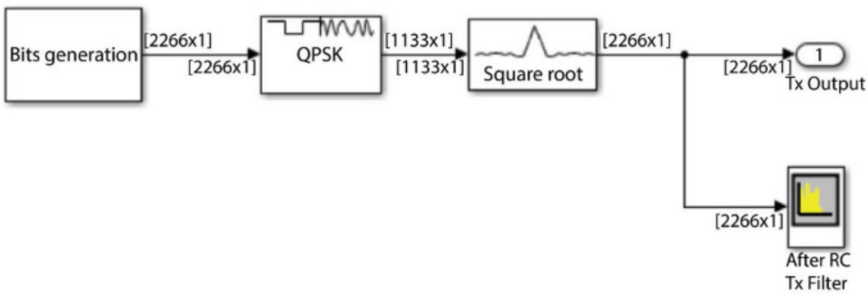


Figure 1.47. *QPSK transmitter*

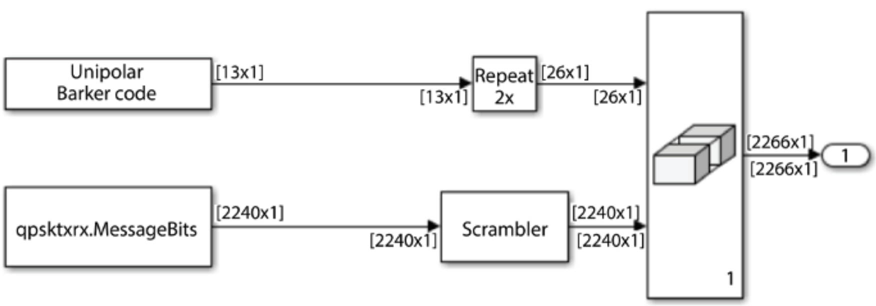


Figure 1.48. *Transmitter QPSK (MATLAB Inc)*

Below we indicate some effects, assessed using the autocorrelation function of Barker codes, by different pulse shaping (S. Rao).

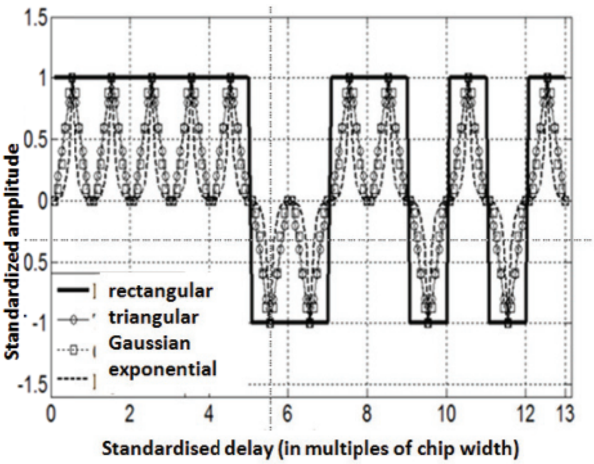


Figure 1.49. *Barker-13 phase coded pulses for different pulse shapes*

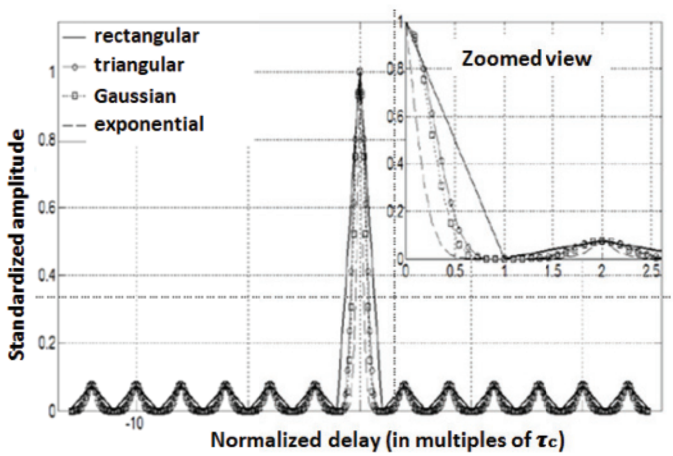


Figure 1.50. *Normalized autocorrelation function (lag) for a modulated Barker-13 impulse*

NOTE.– A Barker code is a finite sequence of N values of +1 and −1, with the ideal autocorrelation property, such that the off-peak autocorrelation coefficients (non-cyclical) verify:

$$c_i = \sum_{j=1}^{N-i} a_j a_{j+i}$$

are as small as possible: $|c_i| \leq 1, \ 1 \leq i < N.$

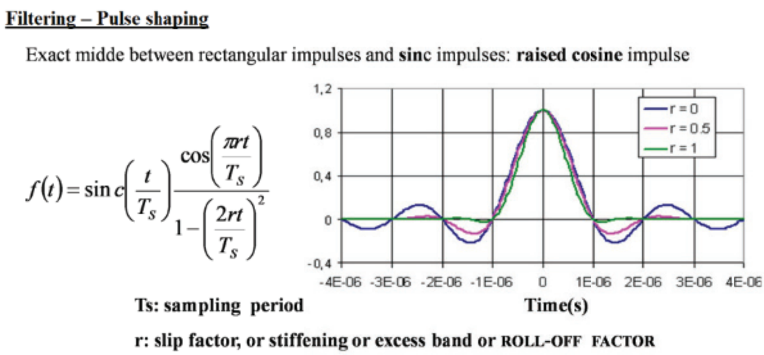


Figure 1.51. *Filtering through various raised cosines. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

Taking an FFT, we obtain the following low pass (Figure 1.52).

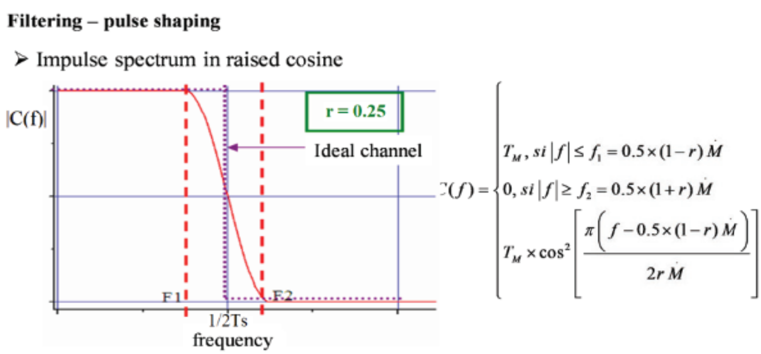


Figure 1.52. *Filtering an impulse in raised cosine. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

Channel:

Figures 1.53 (simulations) and 1.54 (measures) show the effect of filtering in raised cosine on a QPSK.

Spectrum; frequency (and below)

- **Example: signal modulated in QPSK**
- Using a filter in raised cosine **(r = 0.7**; r (roll off**))**

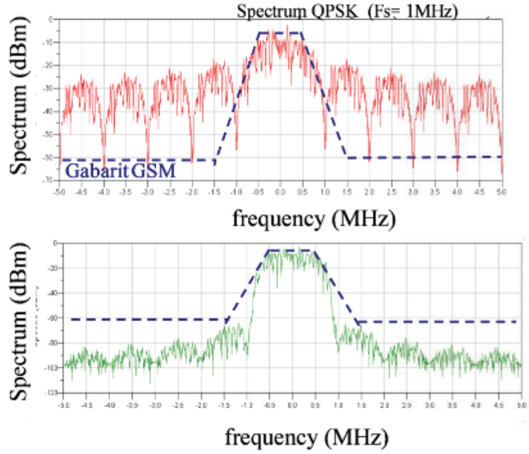


Figure 1.53. *Filtering a QPSK signal using a raised cosine. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

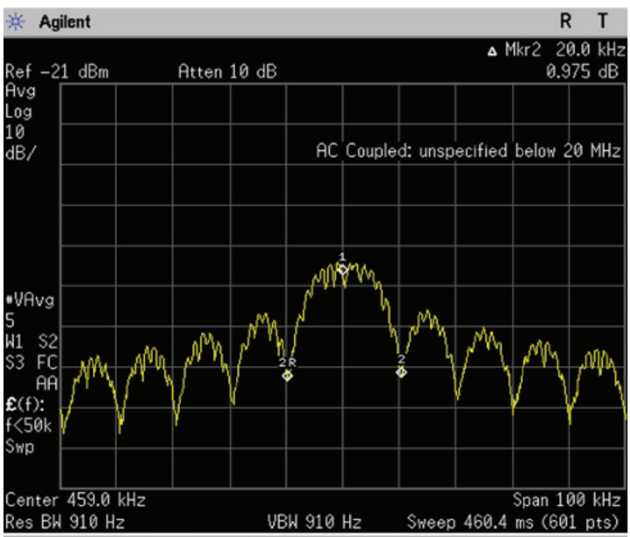


Figure 1.54. *Spectrum of the modulated QPSK signal for a binary flow of 20 Kbits/s without filtering streams I and Q (Agilent). For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

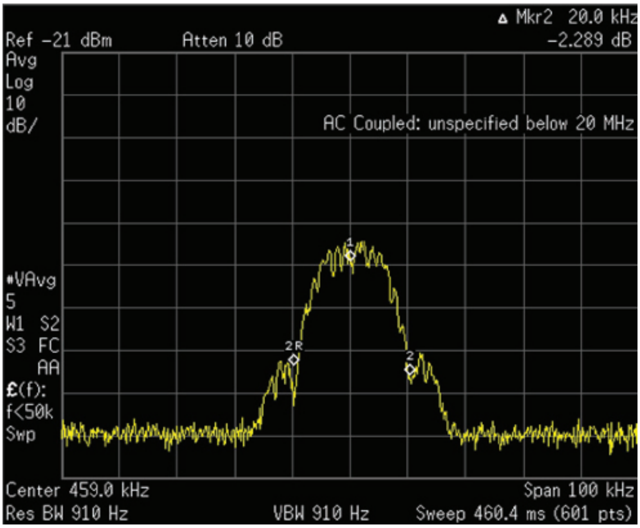


Figure 1.55. *Spectrum of the modulated QPSK signal for a binary flow of 20 Kbits/s with filtering (Agilent). For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

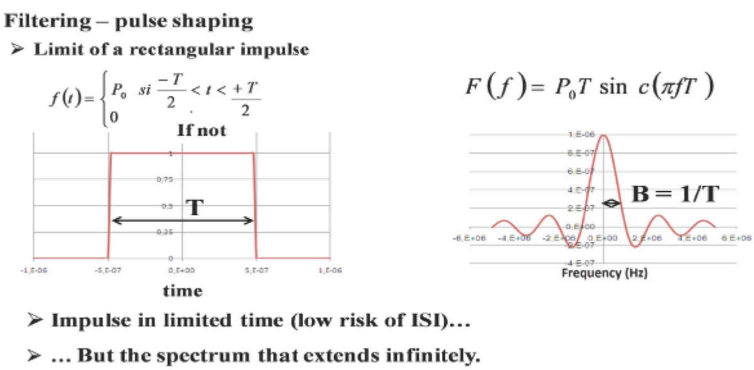


Figure 1.56. *Spectrum of a rectangular impulse. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

Spectral conditions (bandwidth) of the symbol rate can be observed in phase shift modulations in eight states (8PSK). This is a variation of PSK: eight possible states via signal transition, at any moment. The signal's phase can take any of the eight values at any time of the symbol: $2^3 = 8$: there are three bits per symbol. This means that the symbol rate is a third of the binary flow.

This component simulates the effects of free-to-air transmission. It breaks down the signal transmitted with the phase and the frequency offset, a temporal drift to imitate the clock's skew between the transmitter and the receiver, and the AWGN.

Receiver:

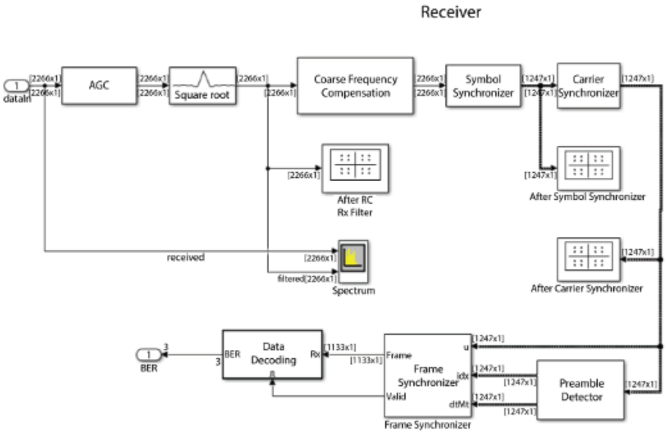


Figure 1.57. *This component recreates the original message sent. It is divided into seven subcomponents (Matlab Inc.)*

1) Automatic gain control: its output power is adjusted via the inverse of the square root of the oversampling factor, so that the input amplitude from the raw frequency compensation subcomponent is stable and approximately 1. This ensures that equivalent gains from the phase and synchronization error detectors remain constant in time. The CAG is placed before the raised cosine reception filter, so that the amplitude of the signal can be measured with an oversampling factor of four. This process improves the precision of the evaluation.

2) Gross frequency compensation: uses nonlinearities and an FFT to give an approximate estimate of the frequency offset to compensate.

3) The frequency offset is estimated using a system object, *comm.PSKCoarseFrequencyEstimator*, and compensation is achieved using *comm.PhaseFrequencyOffset*.

4) Synchronization recovery: activates synchronization recovery with closed-loop scalar processing to overcome the delay effects caused by the

channel, using the system object: *comm.SymbolSynchronizer*. The object implements a PLL to correct the symbol synchronization error in the signal received. A zero-crossing synchronization error detector is chosen here. The input is a frame of samples of constant length. The object's yield is a frame of symbols whose length can change via reinforcement and extraction, depending on delays in the channel involved.

5) Preambule detection: detects the place of the Barker code at input using the object *comm.PreambleDetector*. This runs an algorithm based on cross-correlations to detect the known order of the symbols at input.

6) Frame synchronization: runs frame synchronization and converts input of symbols of variable length into outputs of constant length using *FrameSynchronizer*.

7) Data decoder: achieves resolution and demodulation of phase ambiguity. In addition, the data decoder compares the regenerated message with the transmitted one and calculates the bit error ratios (BERs).

1.8.3.4. *Operation and results*

After using the *System Under Test script* and obtaining BERs values for simulated communication of a QPSK, the following MATLAB code is run.

When you launch simulations, error rate data for the bits is displayed, as well as some results in graph form, respectively:

– constellation: the output constellation diagram *Raised Cosine Receive Filter*;

– output power spectrum *Raised Cosine Receive Filter*;

– output constellation diagram *Fine Frequency Compensation* output;

– (partial) synchronization error estimated from *Timing Recovery*.

```
BER = runQPSKSystemUnderTest(prmQPSKTxRx, useScopes,
printReceivedData);
fprintf('Error rate = %f.\n',BER(1));
fprintf('Number of detected errors = %d.\n',BER(2));
fprintf('Total number of compared samples = %d.\n',BER(3));
```

Initialization

The *commqpsktxrx_init.m* script initializes simulation parameters and generates the structure prmQPSKTxRx.

```
prmQPSKTxRx = commqpsktxrx_init % QPSK
system parameters

useScopes = true; % true if scopes are to be used
printReceivedData = false; % true if the received data is to be printed
compileIt = false; % true if code is to be compiled
useCodegen = false; % true to run the generated mex
file
prmQPSKTxRx =

  struct with fields:

                            M: 4
                   Upsampling: 4
                 Downsampling: 2
                           Fs: 200000
                           Ts: 5.0000e-06
                    FrameSize: 100
                 BarkerLength: 13
                   DataLength: 174
                ScramblerBase: 2
          ScramblerPolynomial: [1 1 1 0 1]
    ScramblerInitialConditions: [0 0 0 0]
                         sBit: [17400×1 double]
             RxBufferedFrames: 10
         RaisedCosineFilterSpan: 10
                MessageLength: 105
                   FrameCount: 100
                  PhaseOffset: 47
                         EbNo: 13
              FrequencyOffset: 5000
                    DelayType: 'Triangle'
  CoarseCompFrequencyResolution: 25
```

```
    PhaseRecoveryLoopBandwidth: 0.0100
    PhaseRecoveryDampingFactor: 1
   TimingRecoveryLoopBandwidth: 0.0100
   TimingRecoveryDampingFactor: 1
      TimingErrorDetectorGain: 5.4000
              ModulatedHeader: [13×1 double]
                      Rolloff (fallout factor, band excess at: 0.5000)
  TransmitterFilterCoefficients: [1× 41 double]
    ReceiverFilterCoefficients: [1× 41 double]
```

1.9. Error vector module and phase noise

The script below, which is MATLAB/OCTAVE/SCILAB compatible, calculates the transmission spectrum from a symbol modulated using QPSK according to Es/N0 for different root mean square (RMS) phase noise values.

```
% Simulation script for a spectrum transmitted using
QPSK
% symbols altered by phase and thermal noise
% ————————————————————————————————

% Simulation script for a spectrum transmitted using QPSK
% symbols altered by phase and thermal noise
% ————————————————————————————————

clear; close all;
N = 10^5; % number of symbols
os = 4; % oversampling factor

%Es_N0_dB = 40;
Es_N0_dB = [0:1:6];

phi_rms_deg_vec = [0:1:6];

%  root-raised cosine filter
t_by_Ts = [-4:1/os:4];
beta = 0.5;
```

```
ht = (sin(pi*t_by_Ts*(1-beta)) +
4*beta*t_by_Ts.*cos(pi*t_by_Ts*(1+beta)))./(pi*t_by_Ts.*(1-
(4*beta*t_by_Ts).^2));
ht((length(t_by_Ts)-1)/2 + 1) = 1 -beta + 4*beta/pi;
ht([-os/(4*beta) os/(4*beta)] + (length(t_by_Ts)-1)/2
+ 1) = beta/sqrt(2)*((1 + 2/pi)*sin(pi/(4*beta)) + (1-
2/pi)*cos(pi/(4*beta)));
ht = ht/sqrt(os);

for ii = 1: length(Es_N0_dB)
   for jj = 1: 2: length(phi_rms_deg_vec)

      % Transmitter
      ip_re = rand(1,N) > 0.5; % % generation ds [
0,1]: uniform probability
      ip_im = rand(1,N) > 0.5; % generation ds [ 0,1]:
uniform probability
      s = 1/sqrt(2)*(2*ip_re-1 + j*(2*ip_im-1)); %
QPSK modulation

      % impulse formatting
      s_os = [s ; zeros(os-1,length(s))];
      s_os = s_os(:).';
      s_os = conv(ht,s_os);
      s_os = s_os(1:os*N);

      %  addition of thermal and phase noise
      n = 1/sqrt(2)*[randn(1,N*os) + j*randn(1,N*os)]; % thermal noise
      phi = phi_rms_deg_vec(jj)*(pi/180)*randn(1,N*os)*sqrt(os); % phase
noise
      y = s_os.*exp(j*phi) + 10^(-Es_N0_dB(ii)/20)*n;


      % spectrum transmitted
      [Pxx1(jj,:) W2 ] = pwelch(y,[],[],1024,'twosided');

      % adjusted filtering
      y_mf_out = conv(y,fliplr(ht));

      y_mf_out = y_mf_out(length(ht):os:end);
```

```
    % vector error
    error_vec = (y_mf_out-s);
    evm(ii,jj)  = error_vec*error_vec';
    theory_evm(ii,jj) =  10^(-Es_N0_dB(ii)/10) + 2
- 2*exp(-(phi_rms_deg_vec(jj)*pi/180).^2/2);
        %  plot(10*log10(theory_evm))
  end
end
figure (1)
plot(10*log10(theory_evm))
xlabel(' Es/N0'); ylabel(' vector error ');
%axis([1  6 - 40 30]); grid on
%hold on
figure (2)
plot([-512:511]/1024,10*log10(fftshift(Pxx1)))
% fftshift can be useful for visualizing the FT with
the  % zero-frequency component in the middle of
the spectrum.
pxx = pwelc(x):
```

NOTE.– *Welch* provides an estimator of the power spectral density. It reflects the evaluation of the power spectral density (PSD), pxx, of the input signal, x, found, breaking the signal down into several segments, then applying a window (see temporal data weighting, thus limiting the Gibbs phenomenon: this is a side effect, linked to drift discontinuities; window functions accord more importance to data from the center of the segment to that from the edges, which leads to a loss of information. The recovery of segments, which we will average, makes it possible to reduce this effect).

When x is a vector, it is treated as a simple channel. When x is a matrix, the PSD is calculated independently for each column and stored in the corresponding column of the pxx. If x has real values, pxx is a one-sided evaluation of the PSD. If x is complex, pxx is a double-sided evaluation of PSD. By default, x is divided into the longest segments possible so as not to exceed eight segments with an overlap of 50%. Each segment is provided with a Hamming window, whose coefficients can be given by:

$$w(n) = 0.54 - 0.46 \cos(2\pi n/N), \quad 0 \leq n \leq N$$

Fliplr: this MATLAB function reflects A with its columns reversed from left to right (i.e. turned around a vertical axis).

```
xlabel('frequency, kHz'); ylabel('amplitude, dB');
legend('0 deg rms', '2 deg rms', '4 deg rms');
title('Spectrum Es/N0 = 40dB, filtering in root-raised cosine;
phase noise rms');
%axis([-0.5 0.5 -50 5]); grid on
```

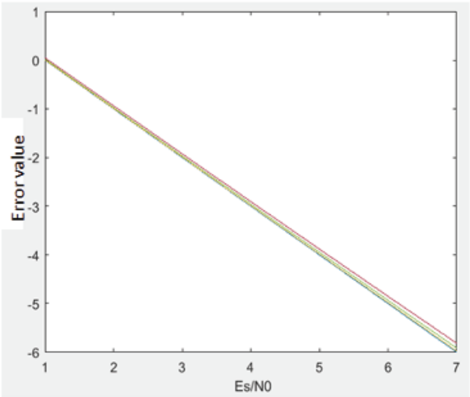


Figure 1.58. *Amplitude of the vector error depending on the signal/noise ratio. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

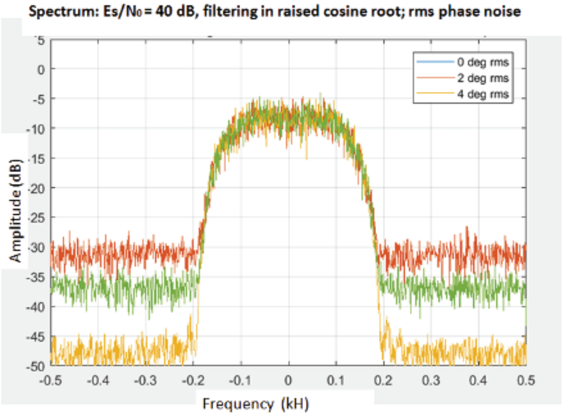


Figure 1.59. *Calculation of a typical spectrum of a vector error. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

The MATLAB documentation gives this type of oscillogram (Figure 1.60).

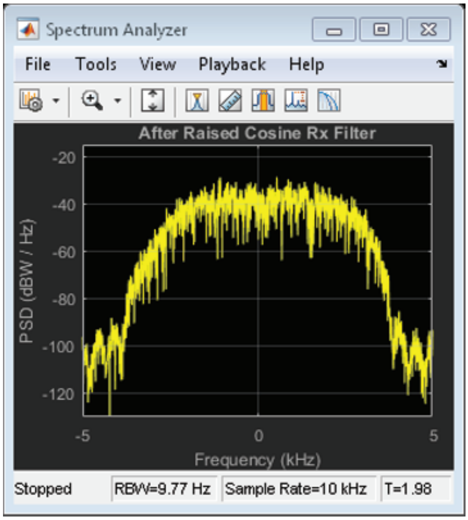


Figure 1.60. *Typical spectrum of a vector error, filtered by a raised cosine (MATLAB). For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

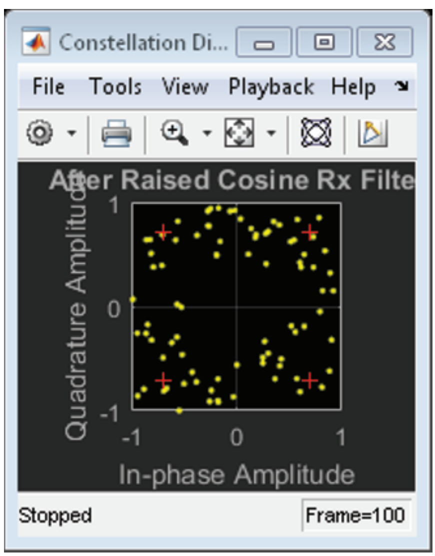


Figure 1.61. *Constellation after filtering in raised cosine. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

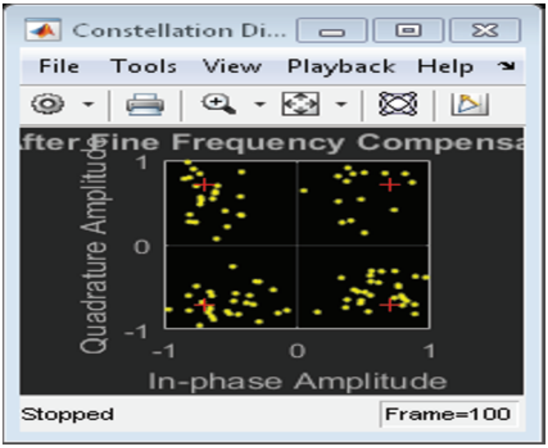


Figure 1.62. *Constellation after fine frequency offsets. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

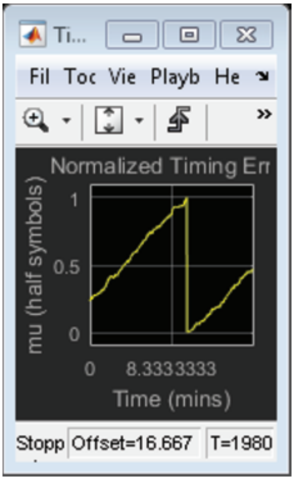


Figure 1.63. *Characteristic of a phase detector (the zig-zags are not ideal). For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

In Figures 1.64 and 1.65, we find simulations of QPSK constellations.

% Create a **QPSK** modulator.

```
mod = comm.QPSKModulator;
```

§ Determine the reference **constellation** points.

```
refC = constellation(mod)
refC =

   0.7071 + 0.7071i
  -0.7071 + 0.7071i
  -0.7071 - 0.7071i
   0.7071 - 0.7071i
```

% Plot of the **constellation**.

```
constellation(mod)
```

% Create a PSK demodulator having 0 phase offset.

```
demod = comm.QPSKDemodulator('PhaseOffset',0);
constellation(demod)
```

1.9.1. *Plot QPSK reference constellation*

Create a **QPSK** modulator.

```
mod = comm.QPSKModulator;
```

Determine **constellation** reference points.

```
refC = constellation(mod)
refC =

   0.7071 + 0.7071i
  -0.7071 + 0.7071i
  -0.7071 - 0.7071i
   0.7071 - 0.7071i
Of the constellation.
constellation(mod)
```

%% Plot QPSK Reference Constellation
%
%%

```
% Create a QPSK modulator.
mod = comm.QPSKModulator;
%%
% Determine the reference constellation points.
refC = constellation(mod)
%%
% Plot the constellation.
constellation(mod)
%%
% Create a PSK demodulator
%% having 0 phase offset.
demod = comm.QPSKDemodulator('PhaseOffset',0);

% Plot its reference constellation. The |constellation|
method works for
% both modulator and demodulator objects.
constellation(demod)
```

Result:

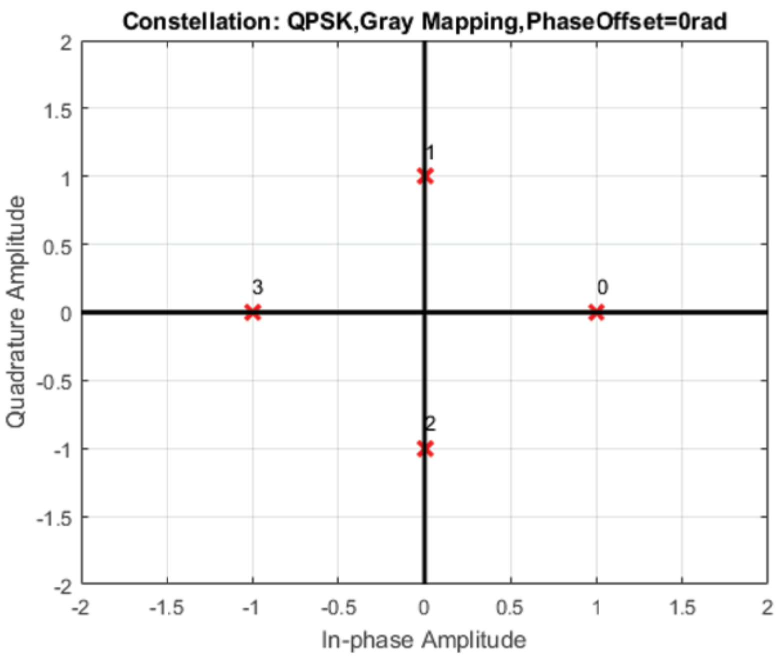


Figure 1.64. *Creating a QPSK constellation*

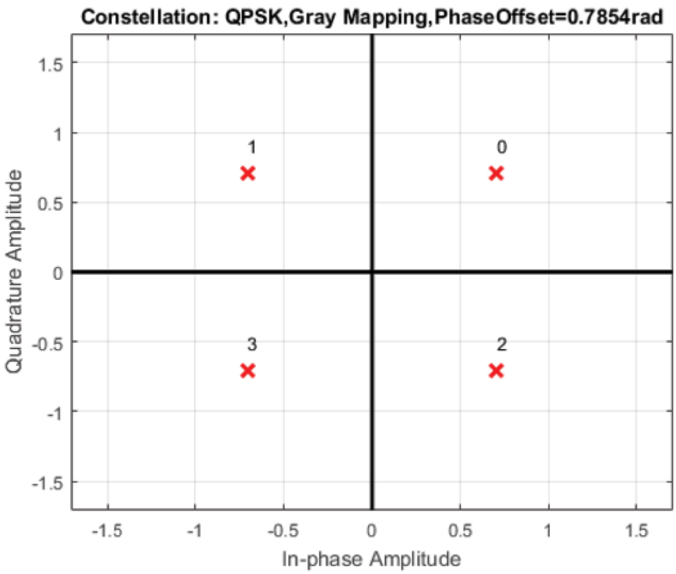


Figure 1.65. *Another QPSK constellation*

Creation of a PSK demodulator, with 0 offset

demod = comm.QPSKDemodulator('PhaseOffset',0);
constellation(demod)

Phase noise on a QPSK signal

% Creation f' of a QPSK modulator object and a phase noise object.

qpskModulator = comm;
phNoise = comm.*PhaseNoise*('Level',-55,'FrequencyOffset',20,'SampleRate',1000);
% Produces random QPSK data. Considers the signal via the phase noise object.
d = randi([0 3],1000,1); % random (random/stochastic): generation of pseudo-random numbers.
x = qpskModulator(d);
y = phNoise(x);

Displays ‘below’: Figure 1.66, the constellation diagram of the noisy QPSK signal. The *phase noise* shows a *rotation fluctuation* on the constellation diagram. = comm.ConstellationDiagram;

constDiagram(y): noisy in-phase constellation plot:

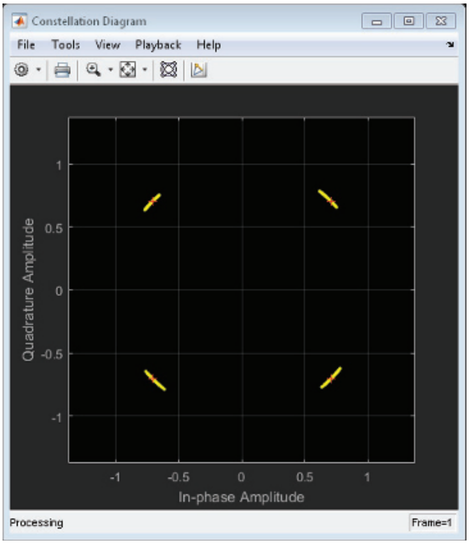


Figure 1.66. *Noisy in-phase constellation. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

Raised cosine function

%function [p,t,filtDelay] = raisedCosine Function(alpha,L, Nsym)
% Generate raised – cosine (RC) impulse
% alpha – roll-off factor (excess band factor; beyond
the Nyquist criteria),
alpha = 2

% L – oversampling factor
L = 4
% Nsym – filter span in symbols
Nsym = 4
% Returns the output impulse p(t) that spans the discrete-time

```
% base -Nsym:1/L:Nsym. Also returns the filter delay when the
% function is viewed as an FIR filter

Tsym = 1; t = -(Nsym/2):1/L:(Nsym/2); % ± discrete-time base

A = sin(pi*t/Tsym)./(pi*t/Tsym); B = cos(pi*alpha*t/Tsym);

% handle singularities at p(0) and p(t = ±1/2a)
p = A.*B./(1-(2*alpha*t/Tsym).^2);
p(ceil(length(p)/2)) = 1; %p(0) = 1 and p(0) occurs at center
temp = (alpha/2)*sin(pi/(2*alpha));

        p(t == Tsym/(2*alpha)) = temp;


filtDelay = (length(p)-1)/2;
%FIR filter delay = (N-1)/2 plot(p)
scatterplot(p)

end
```

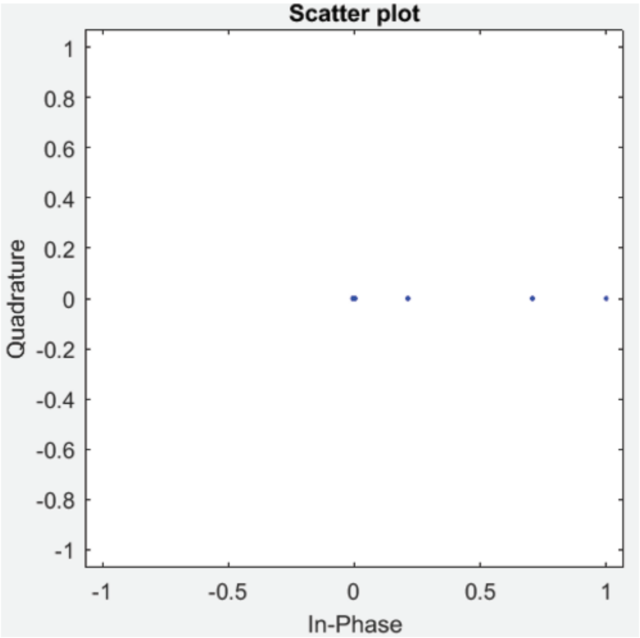


Figure 1.67. *Delay: FIR filter (digital filters with finite-duration impulse response)*

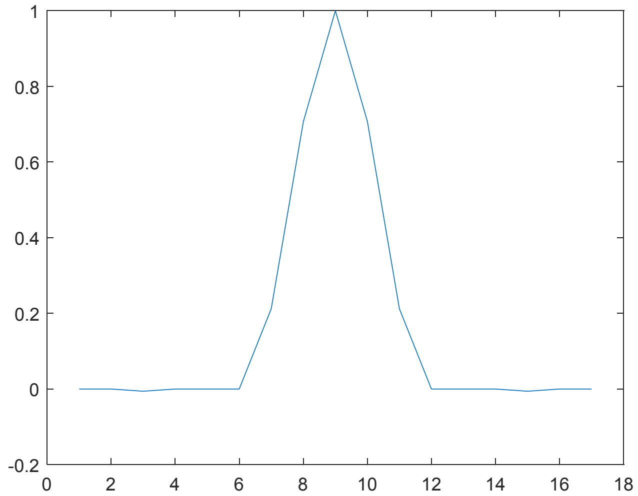


Figure 1.68. *Probability density*

```
%% Phase noise effects on 16-QAM (MATLAB Inc.)
% Add a phase noise vector and frequency offset vector to a 16-QAM signal.
% Then, plot the signal.
%%
% Create 16-QAM modulator having an average constellation
power of 10 W.
modulator = comm.RectangularQAMModulator(16,...
   'NormalizationMethod','Average power','AveragePower',10);
%%
% Create a phase noise object (MATLAB Inc.).
pnoise = comm.PhaseNoise('Level',-
50,'FrequencyOffset',20);
%%
% Generate modulated symbols.
data = randi([0 15],1000,1);
modData = modulator(data);
%%
% Apply phase noise and plot the result.
```

```
y = pnoise(modData);
scatterplot(y) % Scatter: dispersion, collision.
```

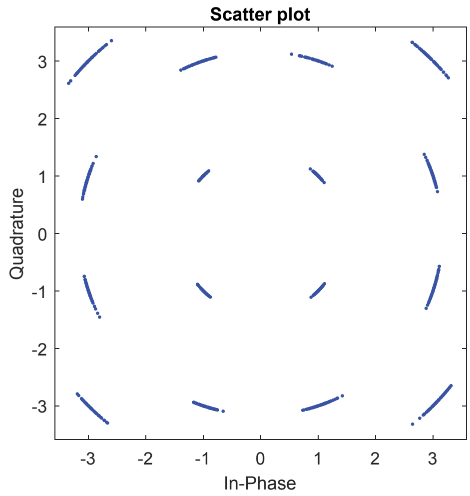


Figure 1.69. *QAM: phase error*

1.9.2. *Effects of phase noise on 16-QAM*

Adds a phase noise vector and offset to the frequency vector on a 16-QAM signal. Then, it plots the signal. It creates the 16-QAM modulator with an average constellation power of 10 W.

```
modulator = comm.RectangularQAMModulator(16,...
   'NormalizationMethod','Average
power', 'AveragePower',10);

% Create a phase noise object.

pnoise = comm.PhaseNoise ('Level',-
50,'FrequencyOffset',20);

% Generate modulated symbols.

data = randi([0 15],1000,1);
```

modData = modulator(data);

% Apply phase noise and plot the result.

y = pnoise(modData);

scatterplot(y)

Hence:

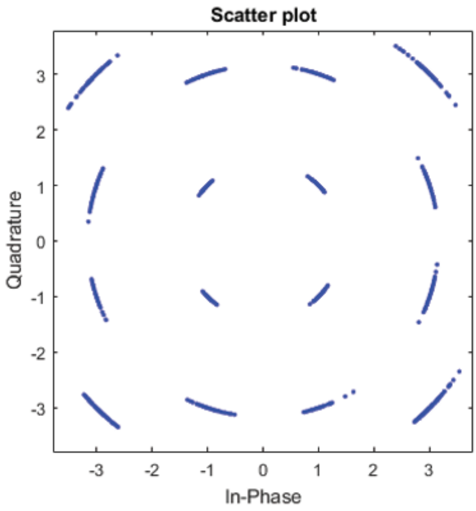


Figure 1.70. *Phase error (QAM)*

1.9.3. *Phase noise: effects of the signal spectrum*

Creation of a sinusiodal wave generator with a carrier frequency of 100 hertz, a sampling rate of 1000 hertz, and a frame size of 10.000 samples.

```
sinewave = dsp.SineWave('Frequency',100,
'SampleRate',1000,...

'SamplesPerFrame',1e4,'ComplexOutput',true);
```

% Creation of a phase noise object (pnoise). It indicates the level of phase noise for -40 dBc/Hz of offsets offset by 100 hertz and -70 dBc/Hz for offsets offset by 200 hertz

```
pnoise = comm.PhaseNoise('Level',[-40 -
70],'FrequencyOffset',[100 200],...
    'SampleRate',1000);
```

% Spectrum analyzer:

```
spectrum =
dsp.SpectrumAnalyzer('SampleRate',1000,'Spectral
Averages',10,'PowerUnits','dBW');
```

% Production of a sinusoidal wave. Adds the phase noise to the sine wave. Noisy signal spectrum plots (see Figure 1.72).

```
x = sinewave();

y = pnoise(x);


spectrum(y)
```

We obtain:

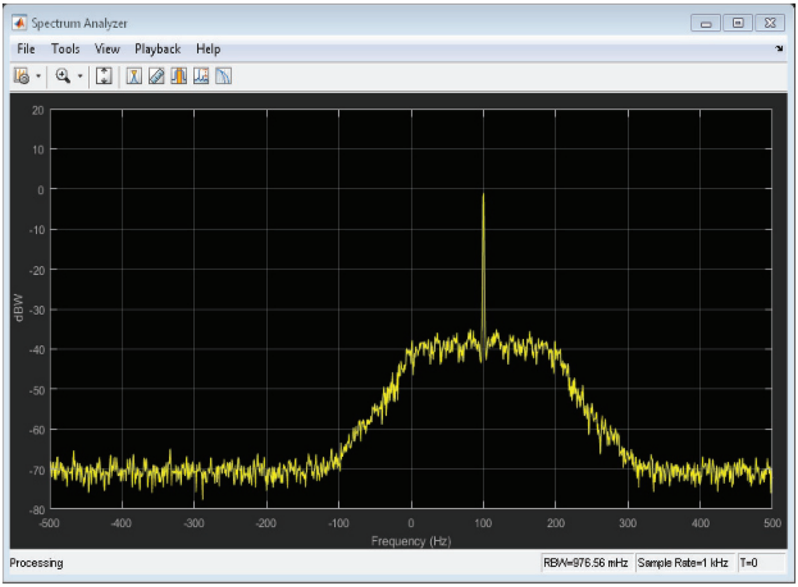


Figure 1.71. *Carrier (spur: see line) and phase noise. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

The phase noise is –40 dB to ±100 Hz of the carrier. The phase noise is –70 dB under the carrier for offsets greater than 200 Hz.

1.9.4. *Algorithms*

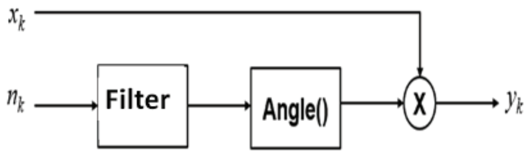


Figure 1.72. *Phase noise*

The output of the object of the phase noise, y_k, is linked to the input sequence x_k, input term, by $y_k = x_{ke}j\phi_k$, where ϕ_k is the phase noise. The phase noise is a Gaussian noise so that $\phi_k = f(h_k)$, where h_k is the term of the noise and f represents a filtering.

If FrequencyOffset is a scalar quantity, then we use the digital infinite impulse response (IIR) filter in which the numerator coefficient, λ, is $\lambda = G2\pi f_{offset}10^{\ L/10}$.

An IIR filter is characterized by a response to the values of the input signal as well as the previous values of the same response. It is also called a *recursive* filter. This filter is one of the two types of linear digital filter. The other possible type is the finite impulse response filter (RIF filter). Unlike the RII filter, the response of the RIF filter only depends on the values of the input signal. Consequently, the impulse response of an RIF filter is always of finite duration.

f_{offset} is the offset frequency in hertz and L is the level of phase noise in dBc/Hz.

FrequencyOffset is a vector; an FIR filter is used. The phase noise is set by the introduction of a scale log10 for frequency offset on the range $[df, f_s/2]$, where the DF is the resolution of the frequency and fs is the sample rate. The frequency resolution is equal to $f_{s/}2$ (= 1/NT), where N is the number of samples. The object increases the NT until the frequency resolution is less than the minimum value of the FrequencyOffset vector or a maximum value of 512 is reached. This value has been chosen to balance the contradictory conditions of a small frequency resolution and a rapid filtering. The object properties correspond to the block parameters.

This object uses a draw of random numbers often based on the remainder of the recursive divisions. In addition, the block uses an initial value to initialize the generator of the random number. Each time the system that contains the block re-runs, the block produces the same series of (*pseudo-*) random numbers.

1.9.5. *Spectrum analyzer*

In MATLAB, (and other simulators) it is easy to represent the spectrums (see Figure 1.73):

```
Fs = 100e6;  % Sampling frequency
   fSz = 5000;  % Size of the frame

   sin1 = dsp.SineWave(1e0,
5e6,0,'SamplesPerFrame',fSz,'SampleRate',Fs);
   sin2 = dsp.SineWave(1e-
1,15e6,0,'SamplesPerFrame',fSz,'SampleRate',Fs);
   sin3 = dsp.SineWave(1e-
2,25e6,0,'SamplesPerFrame',fSz,'SampleRate',Fs);
   sin4 = dsp.SineWave(1e-
3,35e6,0,'SamplesPerFrame',fSz,'SampleRate',Fs); sin5 =
dsp.SineWave(1e-
4,45e6,0,'SamplesPerFrame',fSz,'SampleRate',Fs);

   scope = dsp.SpectrumAnalyzer;
   scope.SampleRate = Fs;
   scope.SpectralAverages = 1;
   scope.PlotAsTwoSidedSpectrum = false;
   scope.RBWSource = 'Auto';
   scope.PowerUnits = 'dBW';
   for idx = 1:1e2
       y1 = sin1();
       y2 = sin2();
```

```
    y3 = sin3();
    y4 = sin4();
    y5 = sin5();
    scope(y1+y2+y3+y4+y5+0.0001*randn(fSz,1));
end
```

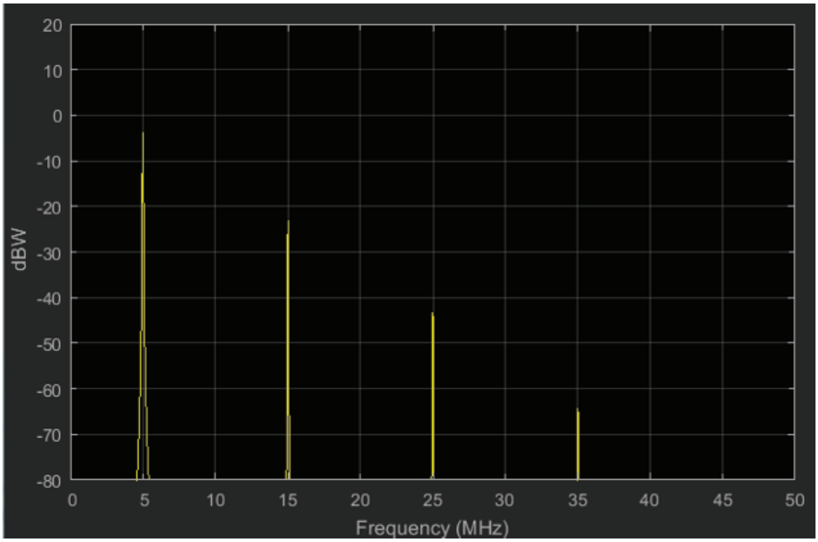


Figure 1.73. *Simulation of a spectrum analyzer. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

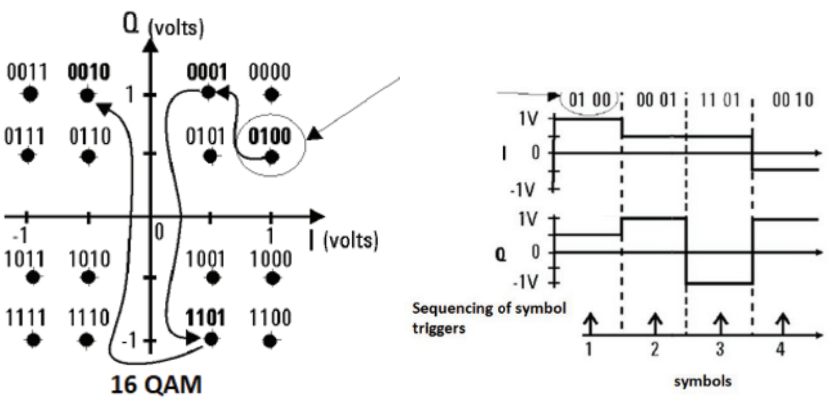


Figure 1.74. *A 16 QAM*

1.9.6. *Measures of the error vector module of a signal modulated by a noisy 16-QAM*

```
%comm.RectangularQAMModulator(16,...)

%
%Measure EVM of Noisy 16-QAM Modulated Signal(MATLAB InC.)
%%
% Create an EVM object. ConFigure it using name-value pairs to output
% maximum EVM, 90th percentile EVM, and the symbol count.
evm = comm.EVM('MaximumEVMOutputPort',true,...
    'XPercentileEVMOutputPort',true, 'XPercentileValue',90,...
    'SymbolCountOutputPort',true);
%%
% Generate random data symbols. Apply 16-QAM modulation. The
modulated
% signal serves as the reference for the subsequent EVM measurements.
data = randi([0 15],1000,1);
refSym = qammod(data,16,'UnitAveragePower',true);
%%
% Pass the modulated signal through an AWGN channel.
rxSym = awgn(refSym,20);
%%
% Measure the EVM of the noisy signal.
[rmsEVM,maxEVM,pctEVM,numSym] = evm(refSym,rxSym)

% Measure the EVM of a Noisy16 QAM Modulated Signal Example
```

Results:

rmsEVM = 10.0838
maxEVM = 30.9821
pctEVM = 14.9950
numSym = 1000

1.10. Gaussian noise (AWGN)

```
clear all % Create a sawtooth wave at.
t = (0:0.1:20)';
%x = sawtooth(t, 0.5);
x = square(2*pi*0.15*t)
```

```
% Apply white Gaussian noise (AWGN) and plot the results.
%y = awgn(x,20,'measured');
plot(t,[x x], 'LineWidth',1)
        legend ('non-noisy signal: pulses')
```

Consider a 101010 train.

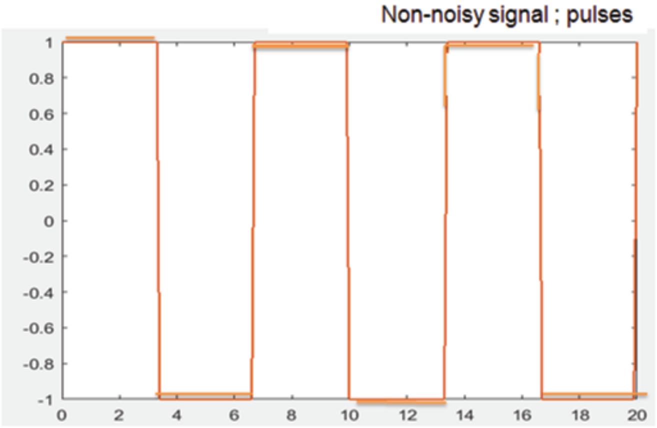


Figure 1.75. *At the transmitter: pulses: 101010. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

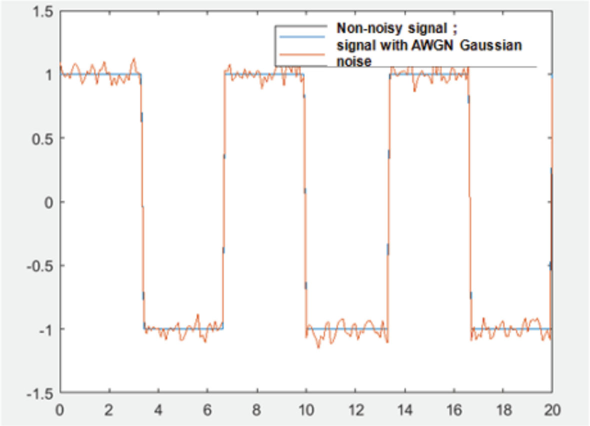


Figure 1.76. *In the channel: pulses: 101010. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

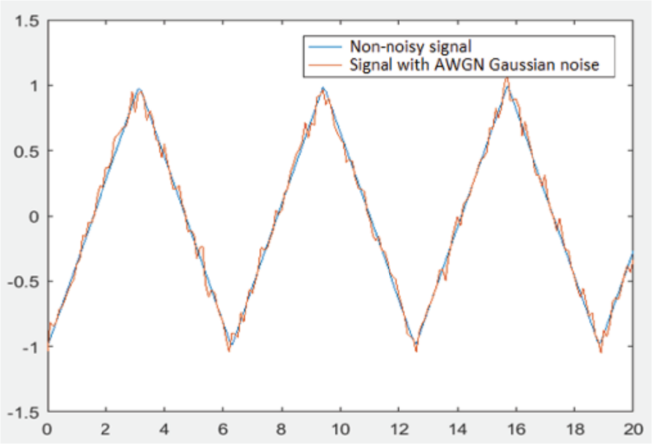


Figure 1.77. *Filtering (average, integration). For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

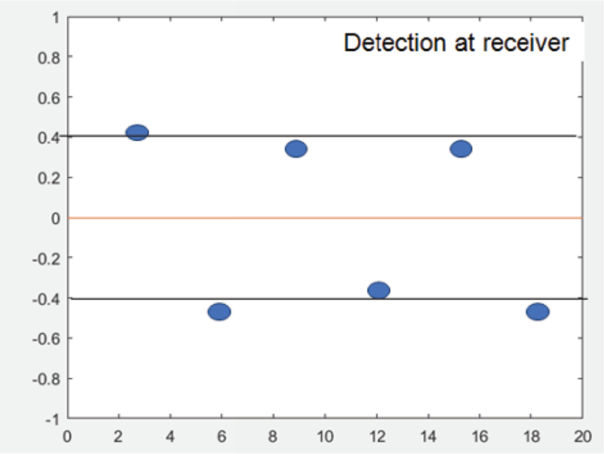


Figure 1.78. *Sampling/thresholding (without error) (101010). For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

1.10.1. *AWGN channel*

The following examples use an AWGN (added white Gaussian noise) channel: QPSK transceiver and general QAM modulation in the AWGN channel.

Level of AWGN channel noise:

The relative power of the noise in an AWGN channel is generally described by quantities such as:

– signal-to-noise ratio (SNR) per sample. This is the real input parameter of the AWGN function;

– energy to bit ratio over PSD of noise (EbNo). This quantity is used by the *BERTool* (from MATLAB) and the functions for evaluating this tool's performances;

– symbol energy/noise PSD (EsNo) ratio;

– ratio between EsNo and EbNo.

The ratio between EsNo and EbNo, both expressed in dB, is the following:

$$E_s/N_0 \text{ (dB)} = E_b/N_0 \text{ (dB)} + 10\log_{10}(k),$$

where k is the number of data bits per symbol: f_b /B; f_b is the net bit rate and B (= 1/ T_{samp}) is the channel BW.

In a communication system, k can be influenced by the size of the modulation alphabet or by the flow of an error-checking code. For example, if a system uses a ½ rate code and a 8-PSK modulation, the number of bits of data information per symbol (k) is the product of the code rate and the number of bits coded per symbol modulated: (1/2) $\log_2$ (8) = 3/2. In such a system, 3 bits of information correspond to six coded bits, which correspond in turn to two 8-PSK symbols.

1.10.2. *Ratio between EsNo and SNR*

The ratio between EsNo and SNR, both these quantities being expressed in dB, is, for complex input symbols, the following:

$$E_s/N_0\,_{(\text{dB})} = 10\log_{10}(T_{sym}/T_{sam}) + SNR_{(\text{dB})}$$

Tsym is the period and the *Tsamp* of the signal's symbol is its sampling period. For example, if a complex baseband signal is oversampled by a factor of 4, EsNo exceeds the corresponding SNR by 10 $\log_{10}$ (4).

1.10.3. *Behavior for real and complex input signals*

The following figures illustrate the difference between real and complex cases by showing the noise PSDs $S_n(f)$ of a real passband white noise process and its complex passband equivalent.

We need to be aware that sometimes the noise power is written as N_0 / 2 when negative frequencies and baseband signals equivalent to complex values are taken into account, rather than passband signals, and in this case, there will be a difference of 3 dB (see Figure 1.79).

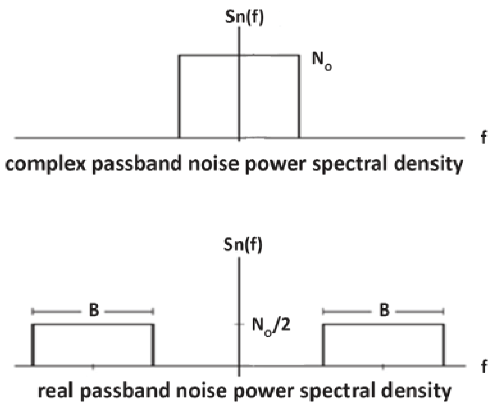


Figure 1.79. *PSDs*

1.11. QAM modulation in an AWGN channel

Our aim is to transmit and receive data using the 16 constellations in the presence of Gaussian noise. Show the “plot” of the noisy constellation and estimate the symbol error rate (SER) for two different signal/noise ratios.

Create a 16-QAM constellation based on the V.29 norm for telephone line modems.

```
c = [-5 -5i 5 5i -3 -3-3i -3i 3-3i 3 3+3i 3i -3+3i -1 -1i 1 1i];
M = length(c);
```

Generate randomized signals.

```
data = randi([0 M-1],2000,1);
```

Modulate data using the MATLAB *genqammod.* function. It is necessary to use general QAM modulation because the constellation is not rectangular.

```
modData = genqammod(data,c);
```

Pass the signal through an AWGN channel with a signal/noise ratio of 20 dB.

Show a scatter plot for the signal received with the reference constellation.

h = *scatterplot*(rxSig);
hold on
scatterplot(c,[],[],'r*',h)
grid

Result:

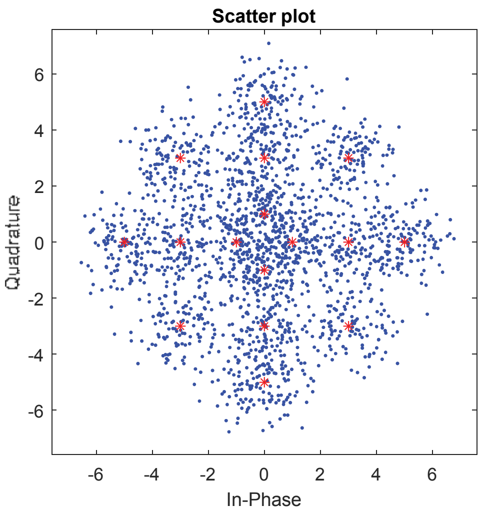


Figure 1.80. *Trace of scatter from a 16 QAM. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

Demodulate the signal received using the *genqamdemod function and determine* the number of symbol errors S and the symbol error ratio.

```
demodData = genqamdemod(rxSig,c);
[numErrors,ser] = symerr(data,demodData)
numErrors = 1
ser = 5.0000e-04
```

Repeat the transmission and demodulation process with an AWGN channel with an SNR of 10 dB. Determine the SER for the reduced signal/noise ratio. As predicted, running breaks down when the SNR diminishes.

```
rxSig = awgn (modData,10,'measured');
demodData = genqamdemod(rxSig,c);
[numErrors,ser] = symerr(data,demodData)
numErrors = 462
ser = 0.2310
```

```
%% General QAM Modulation in an AWGN Channel, MathWorks, Inc.
% Transmit and receive data using a nonrectangular 16-ary constellation in
% the presence of Gaussian noise. Show the scatter plot of the noisy
% constellation and estimate the symbol error rate (SER) for two different
% signal-to-noise ratios.
%%
% Create a 16-QAM constellation based on the V.29 standard for
% telephone-line modems.

c = [-5 -5i 5 5i -3 -3-3i -3i 3-3i 3 3+3i 3i -3+3i -1 -1i 1 1i];
M = length(c);
%%
% Generate random symbols.
data = randi([0 M-1],2000,1);
%%
% Modulate the data using the |genqammod| function. It is necessary to use
% general QAM modulation because the custom constellation is not
% rectangular.
modData = genqammod(data,c);
%%
% Pass the signal through an AWGN channel having a 20 dB signal-to-noise
% ratio (SNR).
rxSig = awgn(modData,20,'measured');
%%
% Display a scatter plot of the received signal along with the reference
```

```
% constellation, |c|.
h = scatterplot(rxSig);
hold on
scatterplot(c,[],[],'r*',h)
grid
%%
% Demodulate the received signal using the |genqamdemod| function and
% determine the number of symbol errors and the symbol error ratio.

demodData = genqamdemod(rxSig,c);
[numErrors,ser] = symerr(data,demodData)
%%
% Repeat the transmission and demodulation process with an AWGN
channel
% having a 10 dB SNR. Determine the symbol error rate for the reduced
% signal-to-noise ratio. As expected, the performance degrades when the
SNR
% is decreased.
rxSig = awgn(modData,10,'measured');
demodData = genqamdemod(rxSig,c);
[numErrors,ser] = symerr(data,demodData)
```

Hence:

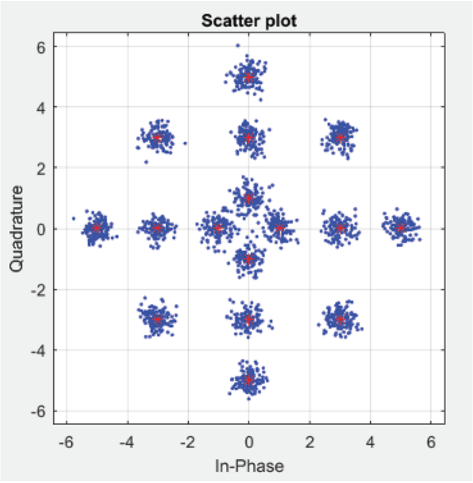


Figure 1.81. *16 constellations. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

1.11.1. *QAM demodulation*

genqamdemod (general quadrature amplitude demodulation)

```
%% General QAM Modulation and Demodulation
%%
% Create the points that describe a hexagonal constellation (MATLAB Inc.).
inphase = [1/2 1 1 1/2 1/2 2 2 5/2];
quadr = [0 1 -1 2 -2 1 -1 0];
inphase = [inphase;-inphase]; inphase = inphase(:);
quadr = [quadr;quadr]; quadr = quadr(:);
const = inphase + 1i*quadr;
%%
% Plot the constellation.
h = scatterplot(const);
%%
% Generate input data symbols. Modulate the symbols using this
% constellation.
x = [3 8 5 10 7];
y = genqammod(x,const);
%%
% Demodulate the modulated signal, |y|.
z = genqamdemod(y,const);
%%
% Plot the modulated signal in same figure.
hold on;
scatterplot(y,1,0,'ro',h);
legend('Constellation','Modulated signal');
%%
% Determine the number of symbol errors between the demodulated data to
the
% original sequence.
numErrs = symerr(x,z)
 y = genqammod(x,const);
%%
% Demodulate the modulated signal, |y|.
z = genqamdemod(y,const);
%%
% Plot the modulated signal in same figure.
hold on;
scatterplot(y,1,0,'ro',h);
```

legend('Constellation','Modulated signal');
%%
% Determine the number of symbol errors between the demodulated data to the
% original sequence.
numErrs = symerr(x,z)

Result:

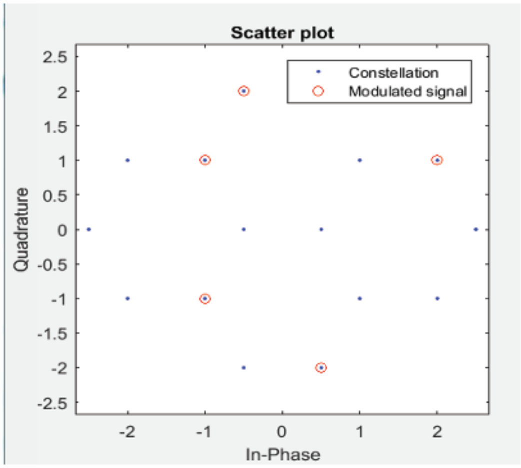


Figure 1.82. *Plot of a QAM constellation. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

Observe that there is a good agreement between the simulated and the theoretical data.

1.11.2. *Detecting phase error*

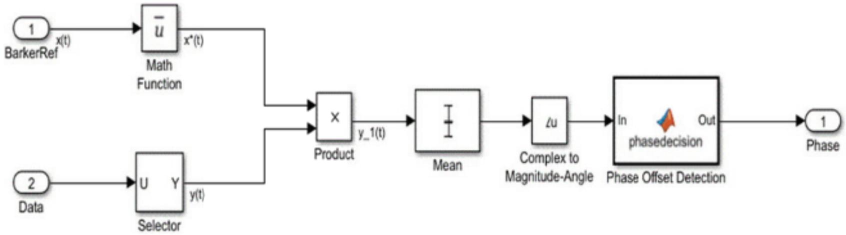


Figure 1.83. *Phase detection (Matlab Inc)*

The signal can be written as: $y(t) = x(t)e^{j\phi} + \eta(t)$, $t = 1,2,\ldots,N$.

We seek to express the value of de ϕ from the observation: $y_1(t)$ $t = 1,2,\ldots,N$ (i.e., to estimate the value $B = e^{j\phi}$).

If we suppose $\eta(t)$ assimilated to a Gaussian noise distribution, then an unbiased estimator may be:

$B_{est} = 1/N*sum(y_1(1:N))$ and $\phi_{is} = angle(B_{is})$

The variance of B_{est} is therefore: σ^2/N.

Phase offset:

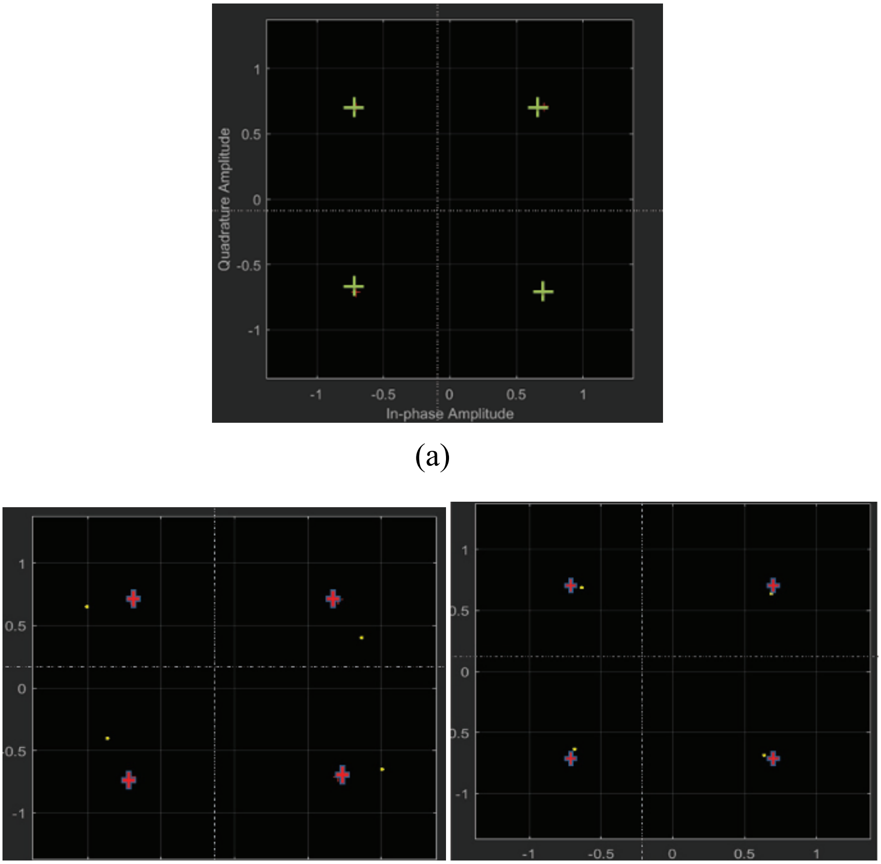
Receiver/Data Decoding/Phase OffsetEstimator/Phase Offset Detection

```
% Map the estimated phase to one of -pi, -pi/2, 0, pi/2,
%#codegen
Out = round(In.*2./pi)./2.*pi;
% The above processing is equivalent to the following processing using
% if/else statements
%
% if In<pi/4 && In>-pi/4
%     Out = 0;
% elseif In>=pi/4 && In<pi*3/4
%     Out = pi/2;
% elseif In>=pi*3/4 || In<=-pi*3/4
%     Out = -pi;
% else
%     Out = -pi/2;
% end
```

Suppression of I/Q imbalance

Systems (QPSK, …, OFDM) suffer from phase/quadrature-phase (IQ) imbalances in frontal analog processing, which can have a considerable impact on performances. Similarly, the local oscillator undergoes a carrier frequency shift. Since these modulations are very sensitive to carrier frequency shifts, this distortion should be considered when analyzing any schema for estimating/compensating IQ imbalance. Algorithms are developed for compensating these distortions in the digital domain.

```
%% Deleting the 'I/Q (Imbalance) of a QPSK signal
% Mitigating the impacts of amplitude and phase imbalance
% over a signal modulated by a QPSK using the system object:
comm.iqimbalancecompensator (TM).
%%
% Creating a constellation diagram object. To indicate the  non-value pairs
% to ensure the constellation diagram shows only the last 100
% data symbols.constDiagram = comm.ConstellationDiagram(...
   'SymbolsToDisplaySource','Property',...
   'SymbolsToDisplay',100);
%%
% To create an I/Q imbalance compensater.
iqImbComp = comm.IQImbalanceCompensator;
%%
% To generate random data symbols and apply %% modulation
%QPSK
data = randi([0 3],1e7,1);
txSig = pskmod(data,4,pi/4);
%%
% To apply an imbalance in amplitude and phase to the signal transmitted.
ampImb = 5; % dB
phImb = 15; % deg
gainI = 10.^(0.5*ampImb/20);
gainQ = 10.^(-0.5*ampImb/20);
imbI = real(txSig)*gainI*exp(-0.5i*phImb*pi/180);
imbQ = imag(txSig)*gainQ*exp(1i*(pi/2 + 0.5*phImb*pi/180));
rxSig = imbI + imbQ;
%%
% To plot the constellation diagram of the signal received. Observe
that this signal has undergone a shift in amplitude and phase.
constDiagram(rxSig)
%%
%   To apply the I/Q  compensation  algorithm  and observe the constellation
(b).
%  The constellation of the compensated signal (c) is almost identical to the
reference constellation (a)
compSig = iqImbComp(rxSig);
constDiagram(compSig)
```



(a)

(b) (c)

Figure 1.84. *Removing I/Q imbalance. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

1.12. Frequency-shift keying

Frequency- and phase-shift key shifts are very similar. A frequency shift of +1 Hz “shifts” the phase to the rhythm of 360° per second, compared to the phase of a unshifted signal. The amplitude remains unchanged, and the frequency varies with the rhythm of the modulating frequency. FSK is an FM modulation using a primary antipolar signal (–1; +1).

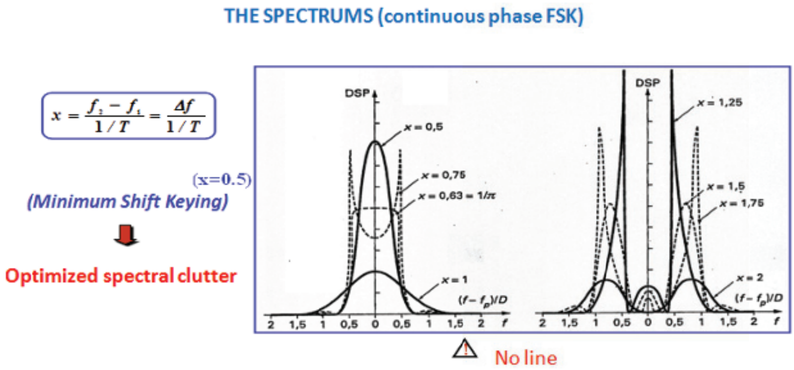


Figure 1.85. *Phase shifting. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

1.12.1. *Binary FSK*

The modulated signal is written as: $m(t) = A\cos\left(2\pi f_p t + \Phi(t)\right)$

In the interval [kT, (k+1)T[, we write: $\frac{1}{2\pi}\frac{d\Phi(t)}{dt} = \frac{\Delta f}{2} a_k \times g(t - kT)$ with g(t) the rectangular waveform of duration T and unitary amplitude.

As a result, in the interval [kT, (k+1)T[we can write after integration:

$$\Phi(t) = \pi.\Delta f.a_k.(t - kT) + \theta_k$$

The modulated signal can finally be written in the interval [kT, (k+1)T[as:

$$m(t) = A\cos\left(2\pi f_p t + \pi.\Delta f.a_k.(t - kT) + \theta_k\right)$$

Let us examine performance, in the interval [0, T[and take as an origin $\theta_0 = 0$:

$$m(t) = A\cos\left(2\pi f_p t + \pi.\Delta f.a_0.t\right)$$

A t = T (by negative value), we write: $m(T) = A\cos\left(2\pi f_p T + \pi.\Delta f.a_0.T\right)$

Let us examine performance, in the interval [T, 2T[:

$$m(t) = A\cos\left(2\pi f_p t + \pi.\Delta f.a_1.(t-T) + \theta_1\right)$$

A t = T, we write: $m(T) = A\cos\left(2\pi f_p T + \pi.\Delta f.a_1.(T-T) + \theta_1\right)$ $= A\cos(2\pi f_p T + \theta_1)$

So that there can be phase continuity, it is necessary that: $\theta_1 = \pi.\Delta f.a_0.T$

A t = 2T (per negative value), we write:

$$m(2T) = A\cos\left(2\pi f_p(2T) + \pi.\Delta f.a_1.(2T-T) + \theta_1\right) = A\cos(2\pi f_p(2T) + \pi.\Delta f.a_1.T + \theta_1)$$

Let us also examine performance in the interval [2T, 3T[:

$$m(t) = A\cos\left(2\pi f_p t + \pi.\Delta f.a_2.(t-2T) + \theta_2\right)$$

A t = 2T, we write:

$$m(2T) = A\cos\left(2\pi f_p(2T) + \pi.\Delta f.a_1.(2T-2T) + \theta_2\right) = A\cos(2\pi f_p(2T) + \theta_2)$$

So that there can be phase continuity; it is necessary that: $\theta_2 = \pi.\Delta f.a_1.T + \theta_1$

Finally, the condition of continuity in the interval [kT, (k+1)T[is written as:

$$\theta_k = \pi.\Delta f.a_{k-1}.T + \theta_{k-1}$$

1.13. Minimum-shift keying

Minimum frequency keying produces a phase advance or a phase delay; frequency keying can be detected through sampling, using phase sampling at each symbol period. Phase shifts of π(2N + 1)/2 radian are easily detected with a I/Q demodulator. At even symbols, the I arm of the channel gives the data transmitted, while at odd symbols, it is the Q arm. This orthogonality between I and Q simplifies detection algorithms and consequently reduces the power in a mobile receiver. The minimum frequency keying that plots the orthogonality of I and Q is the one that has a phase shift of π/2 radian per symbol as a consequence. FSK endowed with this deviation is called minimum-shift keying (MSK). The deviation should be precise so as to produce repeated phase shifts at 90°. FSK is used in a GSM cellular norm (global system for mobile communications). A phase shift of +90° represents one bit of information equal to "1", whereas –90° represents "0". The peak-to-peak frequency offset of an FSK is equal to half the binary flow. The FSK and MSK produce constant envelope conveyor signals, which therefore have no amplitude variation. This is a desirable characteristic for improving transmitter power efficiency. Amplitude variations can exert nonlinearities in the amplifiers' amplitude transfer function, producing spectral regrowth, a component of the power at the adjacent channel. Consequently, more effective but less linear amplifier scans can be used with constant envelope signals, reducing energy consumption.

NOTE.– GMSK uses a Gaussian filter. The central lobe is kept, unlike the others (see Figure 1.87);

– much used for 2G, or GSM, at around 270 kbits/s;

– 99% of the power is found at a band of 240 kHz;

– as for spectral efficiency, it is in the order of 1.3 bit/s.Hz;

– also used for DECT (Digital European Cordless Telephone), with a flow of 1.13 Mbits/s and a BW of around 1.73 MHz;

– spectral efficiency: around 2/3 bit/s.Hz.

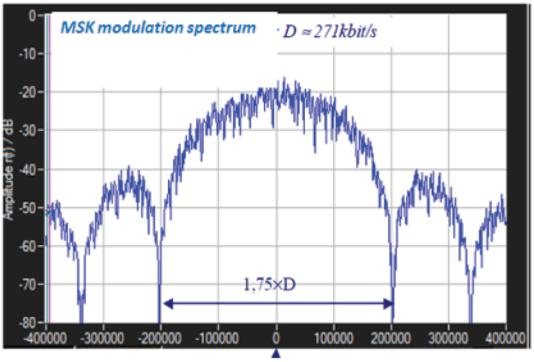


Figure 1.86. *Minimum-shift keying (MSK) modulation spectrum: continuous phase, modulation index: 0.5. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

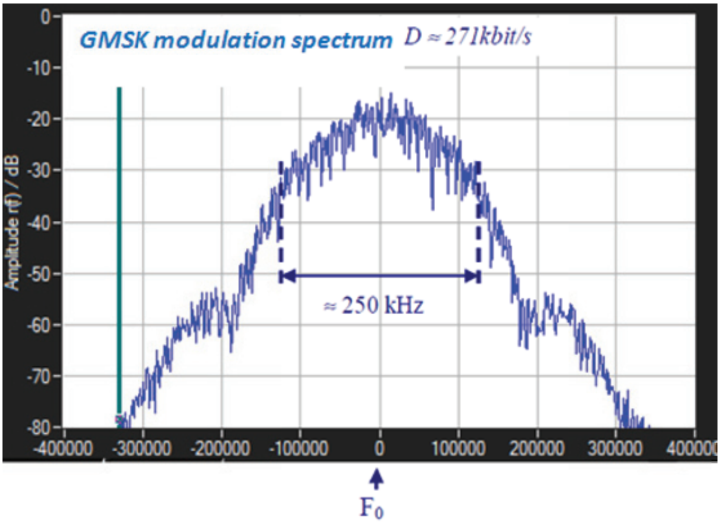


Figure 1.87. *Gaussian MSK (GMSK); the data are, from the outset, processed using a Gaussian filter. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

1.13.1. *Bit error ratio (BER)/Gaussian channel*

```
% Generate theoretical BER data for AWGN channels
% data for several modulation schemas, considering a AWGN channel.
% Create a vector of Eb/No values and define the modulation order M.

clear all
EbNo = (0:10)';
M = 4;
%%
% Generate theoretical BER data for QPSK modulation using berawgn

%=================================================
==========

% function.
berQ = berawgn(EbNo,'psk',M,'nondiff');
%%
% Generate equivalent data for DPSK and FSK.
berD = berawgn(EbNo,'dpsk',M);
berF = berawgn(EbNo,'fsk',M,'coherent');
```

```
%%
% Plotting results.
semilogy(EbNo,[berD berF berQ])
xlabel('Eb/No (dB)')
ylabel('BER')
legend('DPSK','FSK','QPSK')
grid
```

Result:

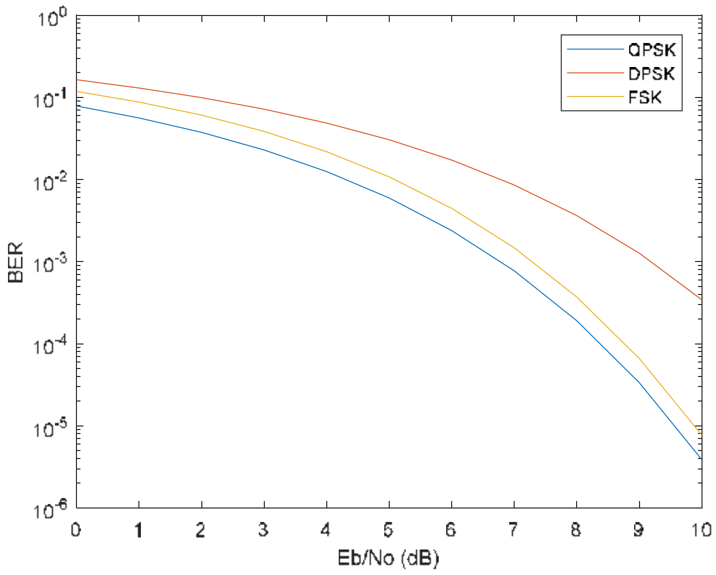


Figure 1.88. *BER for different modulations. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

1.13.2. *Typical analytical expressions used in "berawgn"*

The average bit error probability for each channel is:

$$\text{Pb} = \text{Pr(if e/1 sent)} = \text{Pr(if e/0 sent)}$$

$$\text{Pb} = \int_0^{\infty} \frac{1}{\sqrt{\pi N_0}} \exp\left[-\frac{\left(R_j+\sqrt{E/2)}\right)^2}{N_0}\right] \mathrm{d}R_j$$

$$= \int_{\sqrt{\frac{E}{N_0}}}^{\infty} \cdot \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{x}{2}\right] dx$$

$$= Q\left(\sqrt{\frac{E}{N_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

The demodulator output is none other than the multiplexed output of channels I, Q. The bit error rate for the output is the same as that for each channel. One symbol represents two bits from channels I, Q. A symbol error occurs if both are false. So the error probability per symbol will be:

Ps = 1 – Pr (both bits are correct)
$= 1 - (1 - Pb)^2$
$= 2Pb - Pb^2$

$$= 2Q\left(\sqrt{\frac{E}{N_0}}\right) - Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad \text{(coherent QPSK)}$$

1.14. Amplitude-shift keying

In this case, modulation only takes place on the in-phase carrier cos ($\omega_p t + \varphi_p$). There is no quadrature carrier. This modulation is sometimes called *one* dimensional. The modulated signal is then written as:

$$m(t) = \sum I_k \cdot g(t - kT).\cos(\omega_p t + \varphi_p)$$

The waveform $g(t)$ is rectangular of T duration and amplitude equal to 1 if t belongs to the interval [0, T[and equal to 0 elsewhere.

Remember that the I_k symbol takes its value in the (A_1, A_2, … A_M) alphabet. In other words, this alphabet highlights the $M = 2^n$ possible amplitudes of the signal, the value n designating the groups of n bits or symbols to be transmitted. The carriers' changes in amplitude will be produced at the rhythm R of the symbol transmission.

1.14.1. *On–off keying*

One example of AM is (binary) OOK. Here, a single bit is transmitted per period T, and so $n = 1$ and $M = 2$. The symbol I_k takes its value in the alphabet (0, I_0). We therefore observe carrier extinctions on a chronogram when $I_k = 0$ (see Figure 1.90).

We can note that this modulation is equivalent to an ASK with suppressed carrier (AM-P) by an often random binary *unipolar* signal (0; +1).

The continuous component of the unipolar signal causes a line at frequency f_p, for the PSD to which are added the lateral bands either side of the fp, corresponding to the binary, unipolar PSD signal.

If the B_2 filter is infinitely steep, then interference appears between moments (here: harmonic). We also have $B_2 = B_1$, B_1 being the baseband.

Then, we either reduce the primary band or we limit the smoothing, taking account of the central symmetry of the flanks in relation to $f_p \pm B_1/2$ (conditions directly related to Nyquist criteria).

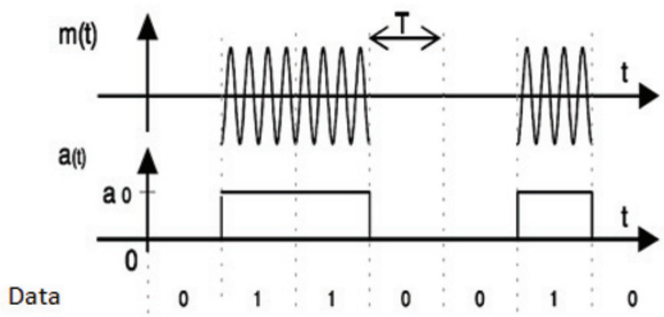


Figure 1.89. *On–off keying (OOK) amplitude modulation*

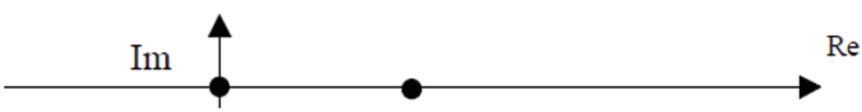


Figure 1.90. *OOK constellation*

On reception, demodulation is carried out by *envelope detection*. In the absence of noise, elevation to the square of the signal m(t) (i.e considering a signal proportional to power) induces a term at frequency *2fp*, eliminated by filtering (using a lowpass circuit), and an information-carrying term, in baseband, proportional to $\sum I_k^{\,2}.g(t-kT)^2$.

The spectrum of the baseband signal is given by the Fourier transform of an impulse:

$$\gamma_{\propto m}(f) = \frac{I_0^2}{4}\left(\frac{sin\ \pi f t}{\pi f t}\right)^2 + \frac{I_0^2}{4}\ \delta(f)$$

The spectrum of the modulated signal is the same spectrum, shifted by $\pm f_p$, and includes a line at $\pm f_p$ frequencies.

1.14.2. *Modulation at "M states"*

In this case, we instead use symmetrical modulation.

"ASK symmetrical modulations"

There are always $M = 2^n$ amplitudes possible for the signal, but here the alphabet values are such that:

$A_i = (2i - M + 1).a_0$ with i = 1,2, … M.

Following the values of *n*, we obtain:

n = 1, M = 2; $-1.I_0$, $1.\ I_0$

n = 1, M = 4; $-3\ I_0$, $-1\ I_0$, $1\ I_0$, $3\ I_0$

n = 1, M = 8; $-7\ I_0$, $-5\ I_0$, $-3\ I_0$, $-1\ I_0$, $1\ I_0$, $3\ I_0$, $5\ I_0$, $7\ I_0$

The constellation of the modulation at M symmetrical states is given in Figure 1.91 for M taking values 2, 4 and 8.

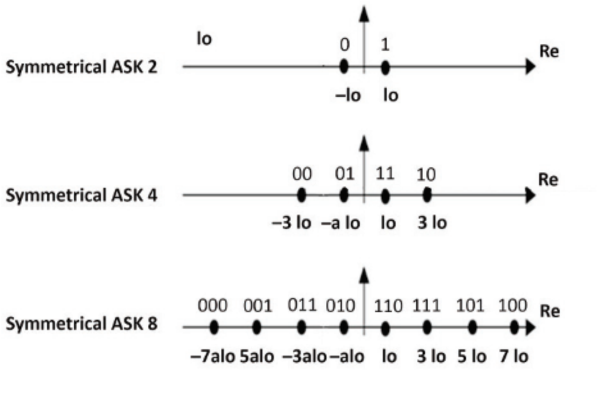


Figure 1.91. *Constellation of amplitude phase shift-keying at M states*

The arrangement of the symbols in fact uses a Gray code: a single bit changes when we pass from one point to another.

Symmetrical ASK 8 000 001 011 010 110 111 101 100
-7 I_0, -5 I_0, -3 I_0, -1 I_0, 1 I_0, 3 I_0, 5 I_0, 7 I_0

Symmetrical QPSK 00 01 11 10
-3 I_0, -1 I_0, 1 I_0, 3 I_0

Symmetrical BPSK 0 1
-1 I_0, 1 I_0

For example: chronogram of “symmetrical QPSK” (see Figure 1.92).

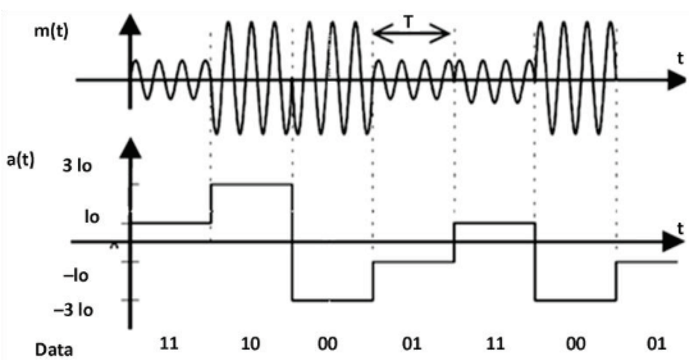


Figure 1.92. *Symmetrical ASK*

Figure 1.93 shows that two bits, at each period T, are transmitted simultaneously.

It is therefore not a question of achieving envelope detection at reception.

The spectrum of symmetrical QPSK

The spectrum of the signal in baseband does not show any line and is written as:

$$\gamma_{\propto m}(f) = \frac{M^2 - 1}{3} {a_0}^2 T (\frac{sin\ \pi f t}{\pi f t})^2$$

The spectrum of the modulated signal is the same spectrum shifted by $\pm f_p$.

Modulation and demodulation

Figures 1.92 and 1.93 show, respectively, a simplified synoptic of coherent (isochronous) modulation and demodulation on a single carrier.

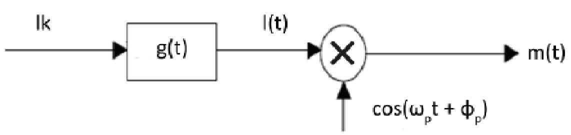


Figure 1.93. *Modulation on a single carrier*

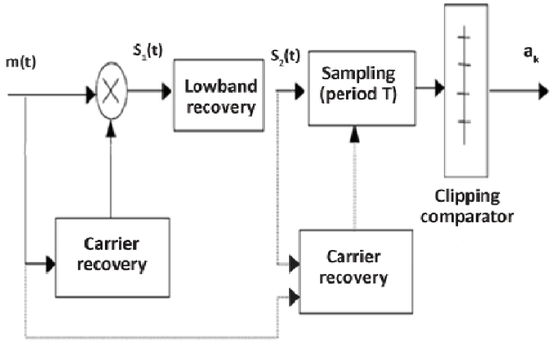


Figure 1.94. *Coherent demodulation on a single carrier*

On the side of the receiver, and assuming there is no noise, if we multiply the signal received $m(t) = \sum I_k .g(t - kT).\cos(\omega_p t + \varphi_p)$ by a sinusoidal wave coming from a local oscillator $A.\cos(\omega_p t + \varphi_p)$, we obtain:

$$S1(t) = \sum I_k.g(t - kT).\cos(\omega_p t + \varphi_p).\ A\cos((\omega_p t + \varphi_p)$$

By developing this expression and by eliminating the term in $\cos(2\omega_p t)$ by filtering, we obtain:

$$S1(t) = A/2 * \sum I_k.g(t - kT).\cos(\varphi_p - \varphi_l)$$

If the receiver has a local oscillator synchronized in frequency and in phase over that of the transmitter, φ_p will be close to φ_l and, so $\cos(\varphi_p - \varphi_l)$ will be close to 1, and consequently

$$S_2(t) = A/2 * \sum Ik.g(t - kT).$$

Thus, the signal $S_2(t)$ is, to a homothety, equal to the modulating train $\alpha(t) = S_1(t) = \sum I_k.g(t - kT)$ which is itself the signal carrying information. It remains for the receiver to *recover the rhythm*, of period T, of the symbols transmitted, to sample the signal $S_2(t)$ in the middle of each period, then to decide, via a comparator at (M-1) thresholds, of the value I_k received.

"ASK M" performances

We then express the error probability according to the N_0/Eb ratio, in which Eb represents the energy sent per bit.

Depending on this ratio, the error probability per symbol is given by the ratio:

$$Ps(e) = \frac{M-1}{M} erfc\left(\sqrt{\frac{3\log_2 M}{M^2 - 1}.\frac{Eb}{N_0}}\right)$$

1.15. Quadrature amplitude modulation

Another member of the digital modulation family is QAM. It is used in applications including digital radio, microwaves, and DVB-C (Digital Video Broadcasting) end modems.

Digital transmission: in cable and modems.

In 16-QAM at 16 states, there are four I values and four Q values; in total, 16 states for the signal. 16 = 2^4: 4 bits can be sent per symbol: 2 bits for I, 2 bits for Q. The rate per symbol is therefore four times the rate per bit: thus, this modulation format produces a more spectrally efficient transmission than BPSK, QPSK, 8PSK QPSK (itself identical to 4QAM).

Another form is 32QAM. In this, there are 16 values for I and l6 for Q. So 36 possible states (6 × 6 = 36). This is a lot (the closest power of 2 is 32). Thus, the symbol's states in four corners, which require a great deal of power at transmission, are omitted; this avoids power peaks at transmission: 2^6 = 32, if there are 5 bits per symbol. In this case, the rate per symbol is a fifth of the binary flow. The current limit is around 512 QAM or indeed 1024.

256QAM: 16 values for I and 16 ones for Q. 16*16 = 256 possible states of the signal, this signal lies across eight bits. There is therefore good spectral efficiency, but the symbols are very close. Hence, there are errors resulting from noise and distortion.

More power is therefore needed to transmit (by *spreading* the signal), and power efficiency is therefore lost compared to other modulations.

Solution? Two quadrature carriers (see QAM).

This is a modulation known as *two-dimensional*.

1.15.1. *Limits on theoretical spectral efficiency*

Passband efficiency measures data processing via this or that modulation in a limited passband. Table 1.2 indicates the limits for the main types of modulation, of the efficiency of the theoretical passband.

These are ideal values for modulators; since demodulators, filtering and transmission paths are never perfect.

Modulation format	Efficiency limits of the theoretical passband
MSK	1 bit/s/Hz
BPSK	1 bit/s/Hz
QPSL	2 bit/s/Hz
8PSK	3 bit/s/Hz
16 QAM	1 bit/s/Hz
32 QAM	5bit/s/Hz
64 QAM	6 bit/s/Hz
256 QAM	8 bit/s/Hz

Table 1.2. *Some modulation formats and their spectral eficiency*

1.15.2. *I/Q imbalance*

```
function [z] = iq_imbalance(r,g,phi)
% This function shows the I/Q imbalance in MATLAB. And the signal %
phase error between the in=phase and quadrature components of the signal
r in  % baseband. The g parameter of the model represents the disparity
between the Q/I % branches of the receiver, and "phi" represents the phase
error of the local oscillator (in % degrees).
Ri = real(r); Rq = imag(r);
Zi = Ri; %I branch
Zq = g*(-sin(phi/180*pi)*Ri + cos(phi/180*pi)*Rq);% Q branch crosstalk
(diaphony)
z = Zi + 1i*Zq;
end
```

```
function [Kest,Pest] = pilot_iq_imb_est(g,phi,dc_i,dc_q)
% Length 64 – Long preambule as defined in EEE 802.11a
preamble_freqDomain = [0,0,0,0,0,0,1,1,-1,-1,1,1,-1,1,...
   -1,1,1,1,1,1,1,-1,-1,1,1,-1,1,-1,1,1,1,1,...
   0,1,-1,-1,1,1,-1,1,-1,1,-1,-1,-1,-1,-1,1,1,...
   -1,-1,1,-1,1,-1,1,1,1,1,0,0,0,0,0];  % representation in the frequency
domain
Preamble = ifft(preamble_freqDomain,64);
% representation in the time domain
```

```
% Define the model of the preambule known by the imbalance of C.C and
Q.I. and estimate the r = receiver_impairments(preamble,g,phi,dc_i,dc_q);
z = dc_compensation(r);
% raises the C.C. imbalance before evaluating the Q.I imbalance.
I = real(z); Q = imag(z);
Kest = sqrt(sum((Q.*Q))./sum(I.*I)); % estimates the gain imbalance
Pest = sum(I.*Q)./sum(I.*I); % estimates the disparity of the phase
end


function [r,n] = add_awgn_noise(s,SNR_dB,samplesPerSymbol)
% Function that adds AWGN to the signal considered
 [r, n] = add_awgn_noise (s, SNR_dB)
 %   adds the noise vector to AWGN at the signal 's'
 % to produce the resulting signal vector 'r' of SNR.
% r this sends the global signal (s, SNR_dB, samplesPerSymbol)
% indicates the oversampling ratio used.
% This also reflects the noise vector that is added to the signal 's'


function y = blind_iq_compensation(z)
% Function to estimate and compensate the I/Q imbalance. During data
transmission %
% y = blind_iq_compensation (z) estimates and compensates this I/Q
imbalance present at the level of the complex signal received at the
processor, in baseband.
.
I = real(z); Q = imag(z);
theta1 = (-1)*mean(sign(I).*Q);
theta2 = mean(abs(I));
theta3 = mean(abs(Q));
c1 = theta1/theta2; c2 = sqrt((theta3^2-theta1^2)/theta2^2);
yI = I; yQ = (c1*I+Q)/c2; y = (yI + 1i*yQ);
end


function [v] = dc_compensation(z)
% Serves to estimate and remove impairments in QI branches.
```

```
% v = dc_compensation(z) cancels this DC alteration
iDC = mean(real(z)); % estimated DC on the I branch
1i*qDC = mean(imag(z)); % estimated DC on the Q branch
v = z-(iDC + 1i*qDC);%  raises the estimated  DC


function [grayCoded] = dec2gray(decInput)
% conversion decimal in Gray code
% example: x = [0 1 2 3 4 5 6 7] %decimal
% y = dec2gray(x)
% returns y = [ 0 1 3 2 6 7 5 4] % Gray code
[rows,cols] = size(decInput);
grayCoded = zeros(rows,cols);
for i = 1:rows
 for j = 1:cols
   grayCoded(i,j)=bitxor(bitshift(decInput(i,j),-1),decInput(i,j));
 end
end


function [y] = dc_impairment(x,dc_i,dc_q)
% Function for creating Compensator Coefficients in a complex model of
  % baseband   [le = iq_imbalance] of y (x, dc_i, dc_q) represents the
imbalance of
% these coefficients between the in-phase and quadrature components of the
% complex signal X. baseband.
%   The C.C polarizations linked to each I, each Q, are represented by
parameters dc_i and dc_q.

y = x + (dc_i +1i*dc_q);
```

ASK and PSK are not very good choices for using the energy sent effectively when the number of points *M* is large. In ASK, the points of the constellation form a straight line, while for PSK, the points form a circle. Indeed, the probability of error is dependent on the minimal distance between the points of the constellation, and the best modulation is one that maximizes this distance for a given average power. A more suitable choice is therefore a modulation that distributes the points *uniformly* in the plot.

It is known that the modulated signal *m(t)* can be written as:

$$m(t) = I(t).\cos(\omega_p t + \varphi_p) - Q(t).\sin(\omega_p t + \varphi_p)$$

and that the two signals *I*(*t*) and *Q*(*t*) have the expression:

$$I(t) = \sum I_k (t) g(t - kT) \textit{ and } Q(t) = \sum Q_k (t) g(t - kT)$$

The *m*(*t*) *modulated* signal is thus the sum of the two quadrature carriers, modulated in amplitude by signals *I*(*t*) and *Q*(*t*).

1.15.3. *QAM-M constellations*

Symbols I_k and Q_k have their respective values in the two M alphabets elements (A_1, A_2, … A_M) and (B_1, B_2, … B_M), which give rise to a modulation that possesses a number [of] $E = M^2$ states. Each state is represented by a pairing (I_k, Q_k), otherwise known by a complex symbol: $Z_k = I_k + jQ_k$.

In the particularly common case where M can be written as $M = 2^n$, the I_k s and the Q_k s represent words of n bits. The complex symbol $Zk = Ik + jQk$ can consequently represent a word of $2n$ bits. The benefit is that the signal *m*(*t*) is a combination of two quadrature carriers modulated in amplitude by independent symbols I_k and Q_k.

In addition, symbols I_k and Q_k very often take their values in the same alphabet of M elements.

For example, QAM-16 is created from symbols I_k and Q_k taking their values in the alphabet {±d, ±3d} hence d is a given constant. A representation of the constellation of this modulation is given in Figure 1.95. QAM-16 has often been used for transmission on a public switched network telephone line (at 9600 bit/s) and for high capacity microwave transmissions (140 Mbits/s) developed in the 1980s.

More generally, symbols I_k and Q_k take their values in the alphabet {±d, ±3d, ±5d, …, ±(M-1)d} with *M*, e.g. QAM-4, QAM-16, QAM-64 and QAM-256.

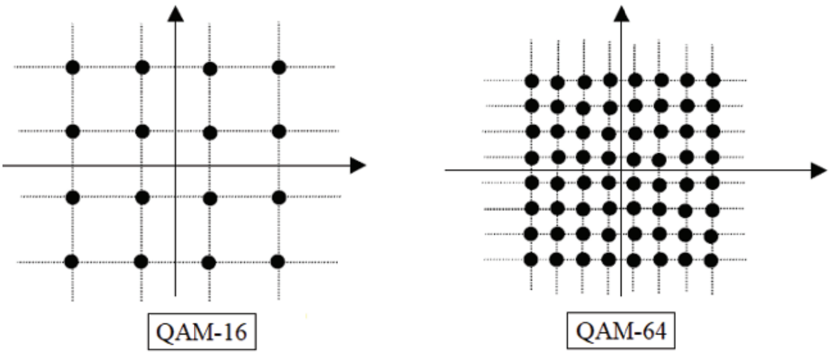


Figure 1.95. *QAM-16 and QAM-64 constellations*

Modulation and demodulation

When the signal $m(t)$ is obtained through a combination of two quadrature carriers modulated in amplitude by independent symbols I_k and Q_k, this simplifies the modulator and demodulator. In fact, for the modulator the incoming binary train $\{i_k\}$ is easily divided into two trains, $\{Ik\}$ and $\{Qk\}$ (see Figure 1.96).

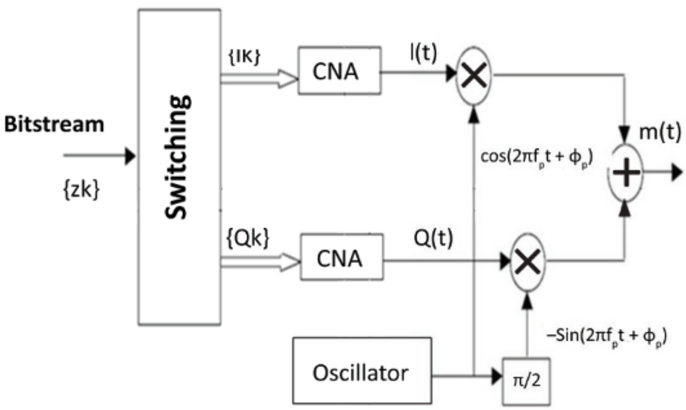


Figure 1.96. *QAM-M modulator*

The reception of a QAM signal uses a *coherent* (isochrone) demodulation and therefore requires extraction of a carrier *synchronized* in phase and frequency with the carrier sent. The signal received is therefore demodulated in two branches: with the in-phase carrier and with the quadrature carrier. The demodulated signals are converted by two CANs; then, a logical decoding determines the symbols and regenerates the bitstream received. The synoptic schema-block of the QAM-M demodulator is very similar to the PSK demodulation.

Spectral efficiency

For a single modulation speed

$R = 1/T$, the binary flow $D = 1/T_b$ of the QAM-M is multiplied by $n = \log_2 M$ compared to that of QAM-2. In other words, for a given BW B, the spectral efficiency:

$\eta = \Delta/B$ is multiplied by $n = \log_2 M$.

$M = 2^n$ modulation; binary flow: D; spectral efficiency: η

n	$M = 2^n$	Modulation	Binary flow	Spectral efficiency η
1	2	QAM-4	D	η
2	4	QAM-2	2.D	$2.\eta$
4	16	QAM-16	4.D	$4.\eta$
6	64	QAM-64	6.D	$6.\eta$
8	256	QAM-256	8.D	$8.\eta$

Table 1.3. *Gain obtained on the spectral efficiency and on the binary flow for different QAM-M modulations for a single modulation speed. We increase M, but with an increased complexity*

"QAM": a generalization of ASK and PSK

By considering the signal m(t) during a period T, we have:

$$m(t) = I_k.\cos(\omega_p t + \varphi_p) - Q_k.\sin(\omega_p t + \varphi_p),$$

$$m(t) = \text{Re}[\ (I_k + j\ Q_k).e^{j(\omega p\ t + \phi p)}]$$

with: $Z_k = I_k + j\ Q_k$

and $\rho_k = \sqrt{I_k{}^2 + Q_k{}^2}$

and $\varphi = \text{arctang}\,(I_k/Q_k) \pm \mathrm{k}\,\pi$

The signal m(t) is then written as: $m(t) = \rho_k.\cos\,(\omega_p t + \varphi_p + \varphi_k)$.

We can see very well that QAM modulation can be considered a *simultaneous* modulation of the phase and the amplitude. Thus:

– PSK phase modulation can be seen as an ASK modulation but one where I_k is constant;

– similarly, ASK AM can be considered as a QAM modulation, where the Q_k are null. Hence, the name “amplitude and phase shift-keying” (APSK) sometimes called QAM;

– CIR(4,4,4,4) modulation at 4 amplitudes and 4 phases, whose constellation is given in Figure 1.97, is a good example of this (see IUT – International Union of Telecommunications).

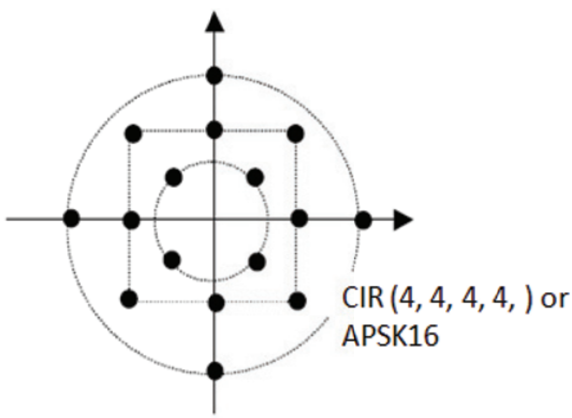


Figure 1.97. *APSK-16 constellation*

Comparison of ASK and PSK

ASK and PSK can be compared as functions of M using curves of the error probabilities per symbol *Ps(e)*. For example, for a probability of error per symbol *Ps(e)* of 10^{-5} and for a signal to noise ratio of Eb/N_0 of 4 dB, ASK can only send 2 bits per symbol (M = 4), whereas PSK can send 3 (M = 8).

This gives ASK a clear advantage for M, ranging from 2 to 16. For values of M higher than 16, degradation of PSK performances leads us to look for other modulations at the cost of increased modulator and demodulator complexity. From the point of view of simplicity of performance, it is ASK that has the advantage, as it is one-dimensional.

The noise level and any other “distortion” gives the lowest error rate on bits since the BPSK system gives the greatest distance between the points of the signal. The probability of a BPSK’s bit error in an AWGN can be obtained by

$$P_{e,BPSK} = Q\sqrt{\frac{2E_b}{N_0}}$$

where

– E_b: energy per bit;

– N_o: spectral of noise power;

– P_e: PSD of noise.

Q is calculated via the area under the Gaussian tail:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\left(\frac{t^2}{2}\right)} dt$$

$$\int a^x dx = \frac{a^x}{lna} + C$$

% Demonstration of Eb/N0 Vs BER for baseband differentially encoded BPSK (MATLAB Inc.) clear all;clc;

```
%——Input Fields—————
N = 1000000; %Number of symbols to transmit
EbN0dB = -4:2:20; % Eb/N0 range in dB for simulation

%————Transmitter—————
```

```
m = rand(1,N) > 0.5; % random symbols from 0's and 1's
b = filter(1,[1 -1],m,0); %IIR filter implementing the differential encoding
b = mod(b,2); % XOR operation is equivalent to modulo-2 and binary
negation
s = 2*b-1; % BPSK antipodal Mapping

SER = zeros(length(EbN0dB),1); % Place holder for SER values for each
Eb/N0
EbN0lin = 10.^(EbN0dB/10); % Converting Eb/N0 from dB to linear scale
for i = 1:1:length(EbN0dB),

    Esym = sum(abs(s).^2)/length(s); % calculate actual symbol energy from
    generated samples
    N0 = Esym/EbN0lin(i); % find the noise spectral density
    noiseSigma = sqrt(N /2); % standard deviation for AWGN Noise
    n = noiseSigma*(randn(1,N) + 1i*randn(1,N));
    y = s + n;%received signal = signal + awgn noise
    %————Receiver——————
    bCap = (y > = 0);% clipping detection (BPSK demod)
    mCap = filter([1 1],1,bCap,0); % FIR filter implementing the differential
   decoding
    mCap = mod(mCap,2); % binary messages, therefore modulo-2
    SER(i) = sum((m~ = mCap))/N;%—— Symbol Error Rate Computation—
–
end
%——Theoretical Symbol Error Rate————
theoreticalSER_DPSK = erfc(sqrt(EbN0lin)).*(1-0.5*erfc(sqrt(EbN0lin)));
%Theoretical symbol error rate for DPSK
theoreticalSER_BPSK = 0.5*erfc(sqrt(EbN0lin)); %Theoretical symbol error
rate for BPSK
%————Plotting————————
semilogy(EbN0dB,SER,'k*'); hold on;
semilogy(EbN0dB,theoreticalSER_DPSK,'r-','LineWidth',1.0);
semilogy(EbN0dB,theoreticalSER_BPSK,'b-','LineWidth',1.0);
set(gca,'XLim',[-4 12]);set(gca,'YLim',[1E-6 1E0]);set(gca,'XTick',-4:2:12);
title('Probability of Symbol Error for BPSK signals');
xlabel('E_b/N_0 (dB)');ylabel('Probability of Symbol Error - P_s');
legend('Simulated', 'Theoretical');grid on;
```

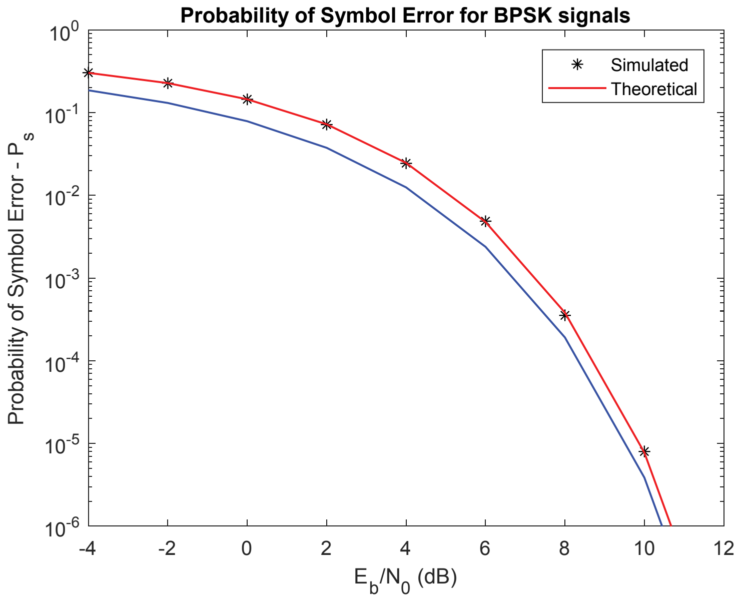


Figure 1.98. *BPSK: probability of symbol error. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

The bit error probability for QPSK is similar to that for BPSK, but QPSK makes it possible to double the information transmitted, without increasing the BW transmitter. In addition, QPSK offers double the spectrum efficiency with the same energy efficiency. The bit error probability can be obtained using the following relationship;

$$P_{e,QPSK} = \frac{1}{2}\, erfc\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

where N_0 is the noise spectral density.

QPSK has the same probability of error as BPSK because of the 3 dB reduction in the error distance of QPSK, compensated by the 3 dB reduction in its BW.

QAM involves two different signals at the same time (two-dimensional modulation). It can be seen as an AM or a PM. It is used mainly in digital telecommunications systems at very high data flows. Errors in information are reduced; the adjacent constellation points are well distributed.

There are different forms of QAM but the most common are 16QAM, 64QAM, 128QAM and 256QAM. QAM M makes it possible to transmit more bits per symbol; this *a priori* makes it possible to transmit data in a much smaller passband. However, if the average energy of the constellation remains constant, the symbols should be very close to one another, which makes them more vulnerable to noise and other distortions, which leads to an error rate on the higher bits. This signal should be transmitted using more power so that the symbol spreads further, thus reducing this technology's energy efficiency compared to other modulation techniques. Nevertheless, higher order QAM can transmit more data, which makes them more effective in terms of spectral transmission, but they are not as weak as lower order QAM. The general form of QAM M-ary signals can be expressed by:

$$S_i(t) = \sqrt{\frac{2E\min}{T_S}} a_1 \cos(2\pi f_C t) + \sqrt{\frac{2E\min}{T_S}} b_1 \sin(2\pi f_C t),$$
$$0 \le t \le T, i = 1, 2, 3, \ldots M$$

where E_{min} is the energy of the signal with the lowest amplitude, and a_1 and b_1 are a pair of independent integers, chosen according to the location of the particular signal point. The average probability of error in an AWGN for a QAM M-ary can be represented by:

$$P_{e,QPSK} \cong= 4\,(1 - \frac{1}{\sqrt{M}}) erfc\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

Higher order QAM modulation schemas are therefore vulnerable to errors. Consequently, error correction encoding ensures that there is a greater chance of the signal surviving in AWGN and the Rayleigh channel along multiple paths and thus improves system performances.

However, there is always a compromise in applications because a particular application may necessitate greater precision in data reception, whereas for another application, the desirable requirement may be the passband or the power available.

The choice of technology used is crucial, as it greatly influences the characteristics, performances and overall physical realization of a communication system.

Digital modulation will continue to be pertinent in the world of communication, voice and high flow data, since the communication systems designer has the main aim of transmitting information in the shortest possible time, in the available passband, at an affordable cost and with the lowest error rates possible.

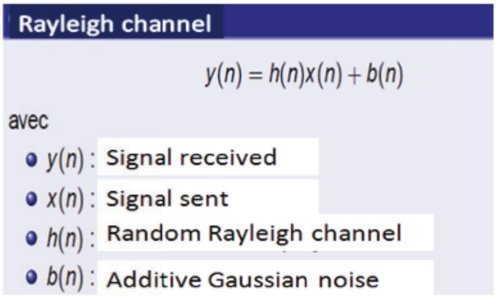


Figure 1.99. *Rayleigh channel*

1.16. Digital communications transmitters

Below is represented a block schema (Figure 1.100) from a digital communications transmitter. Its ends are used for analog signals. The first state, then, is to convert a continuous analog signal into discreet digital binary train; this is digitization.

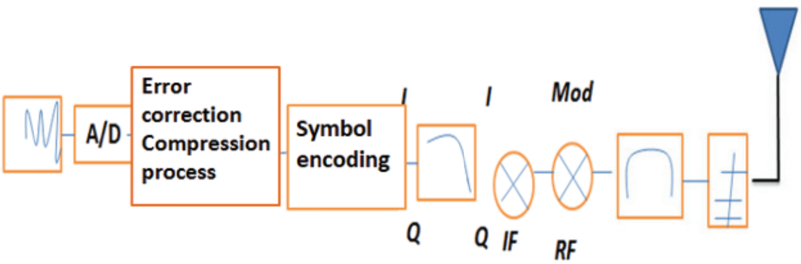


Figure 1.100. *Block schema from a digital communications transmitter*

The following stage consists of adding coding, for example voice coding, for data compression. Some channel codings are then added. Channel coding encodes data so as to reduce as much as possible the effects of noise and interferences in the transmission path. It adds additional bits to the incoming data flow and removes redundancies. These additional bits are intended for error correction or sometimes for sending sequences of streams for identification or equalization. This can facilitate synchronization for different symbols. In symbol clock transitions, the carrier transmitted is the correct I/Q (or amplitude/phase) value to represent a given symbol (a specific point in the constellation). Then, the I/Q or amplitude/phase values of the carrier transmitted are changed to represent another symbol. The interval between these two moments is the period of the symbol clock. The symbol clock phase is correct when the symbol clock coincides with optimal instants to detect symbols.

The next step in the transmitter: *filtering*. This is essential for good BW efficiency. Without filtering, signals would have transitions that were too quick between states and so a broader frequency spectrum needed to send information. A single filter creates a compact and spectrally efficient signal to do this, which can be placed on a carrier. The output of the channel coder is then introduced into the modulator. Since there are I and Q components in radio, half of the information can be sent in I, the other half on Q. This is why digital radios perform well with this type of digital signal. I and Q components are separated.

The rest of the transmitter is similar to a typical RF transmitter or a microwave transmitter/receiver pairing. The signal's frequency is converted into a higher RF – *upconverted* – intermediate frequency (IF). Undesirable signals, which would be produced by high frequency conversion, are then rejected though filtering.

1.16.1. *A digital communications receiver*

The receiver is similar to the transmitter, but in reverse. The arriving signal is "*down* converted" to an "IF" and demodulated. The ability to demodulate the signal is hampered by factors including atmospheric noise; the signals and multiple paths compete with one another, indeed destroy one another and disappear.

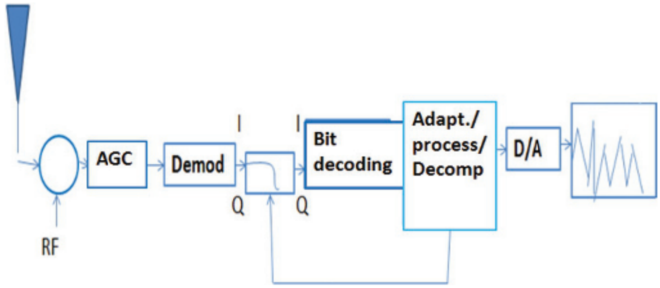


Figure 1.101. *At the receiver: demodulation*

In general, demodulation involves the following stages:

1) re-establishing the carrier frequency (*carrier lock*);

2) re-establishing the symbol clock;

3) decomposition of the signal into I and Q components;

4) determining the I and Q values for each symbol slicing;

5) decoding and interleaving;

6) expansion of bit flow;

7) digital-analog conversion, if it takes place.

In more and more systems, nevertheless, the signal starts off digital and remains digital. It is never analog in the sense of a continuous analog signal, as for audio. The main difference between the transmitter and receiver is the question of re-establishing the carrier and the (symbol) clock. The frequency and the phase (or the synchronization) of the symbol clock should be correct in the receiver to demodulate the bits correctly and recover the information transmitted. A symbol clock could have the right frequency but the wrong phase; if the symbol clock is aligned with the transitions between symbols rather than the symbols themselves, the demodulation would be of poor quality.

Symbol clocks are usually fixed in frequency and this frequency is known exactly by the transmitter and receiver. The difficulty lies in obtaining them aligned in phase or by synchronization. There is a variety of techniques and most systems use two or three. If the signal's amplitude changes during the modulation, a receiver can measure these variations. The transmitter can send a specific synchronization signal or a predetermined order of bits such as 10101010101010 to “train” the receiver’s clock. In systems with a pulsed carrier, the symbol clock can be aligned with the carrier power. In the transmitter, RF carrier and digital data are well distributed, since they are produced inside the transmitter itself. This is not the case for the receiver. The receiver can approximate where the carrier is, but has no information on its phase or the timing/synchronization of symbol clocks. One difficulty in receiver design is that of the algorithms for re-establishing the coding channel. This can be facilitated by channel coding carried out on the transmitter.

Compromises in frequency, phase, synchronization and modulation are made to cancel interferences for multi-user communications systems. It is necessary to measure parameters in RF digital communication systems to make the right compromises. Measures include analyzing the modulator and demodulator, characterizing the quality of the transmitted signal, locating causes of high BER and studying new types of modulation. Measures on RF digital communications systems fall into more or less four categories: *power*, *frequency*, *synchronization* and *precision in modulation*.

1.16.2. *Measures of power*

Measures of power include the power of the carrier and measures associated with amplifier gain as well as loss of insertion in filters and attenuators. The signals used in digital modulation are noise signals. Measures of band power (power integrated over a particular range of frequencies) or PSD are often made. PSD measures normalize power in a certain BW, usually 1 Hz.

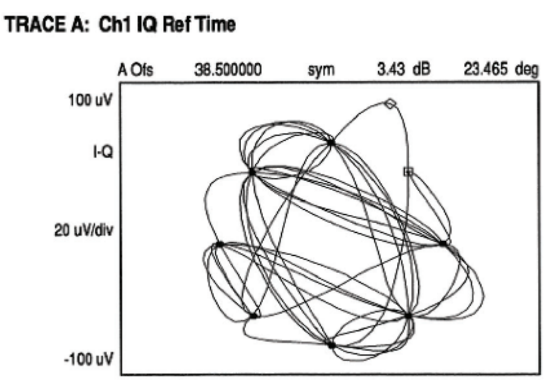


Figure 1.102. *Power of the adjacent channel (Agilent)*

1.16.3. *Power of the adjacent channel*

The power of the adjacent channel is a measure of the interferences created by a user that affects other users in neighboring channels. This test quantifies the energy of a digitally modulated RF signal, which tackles the predicted transmission path in an adjacent channel. The result of the measurements is the ratio (often in dB) of the power measured in the adjacent channel at any power transmitted. A similar measure is the power of an alternative channel that keeps the same ratio to two channels, far from the predicted communication channel.

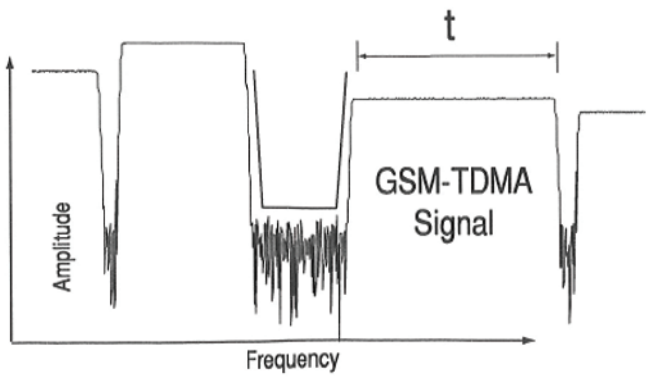


Figure 1.103. *Measures of power and synchronization*

For pulsed symbols (such as TDMA), power measures have a time component and can also have a frequency component. Burst profiles (power depending on time) have opening and closing times, which can be measured. Another measure is that of average power, when the carrier is on, averaged over many “closed/opened” cycles.

1.16.4. *Frequency measures*

Frequency measures are often complex in digital systems since factors other than pure tonalities can be considered. The occupied BW is an important measure. It ensures that the operators remain in the passband allocated to them.

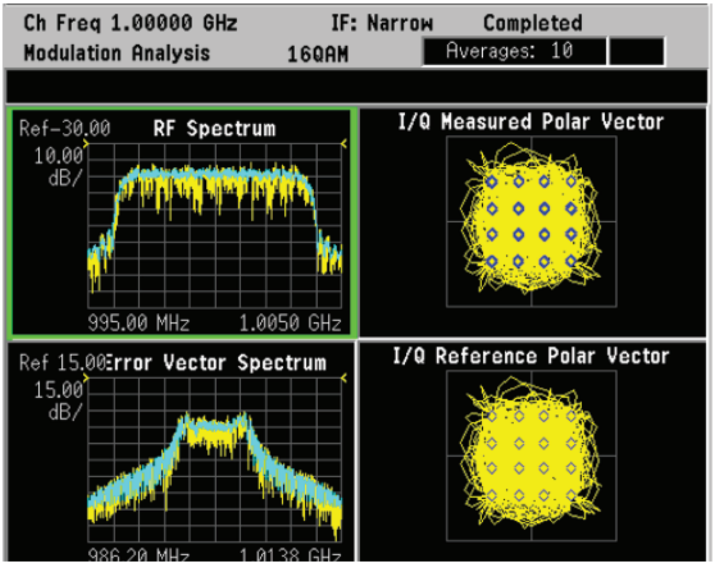


Figure 1.104. *Occupied bandwidth (Agilent). For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

The occupied BW is a measure of how the frequency spectrum is covered by the considered signal. The units are in hertz, and the measure of the occupied BW generally involves a percentage or ratio of power. Typically, some of the power in a signal to be measured should be specified. A commonly used percentage is 99. A measure of power according to frequency (such as power in the band) is made to add power in order to achieve the percentage indicated. For example, one might indicate that "99%" of the power in this signal is contained in a BW of "30 kHz". One could also indicate that the BW occupied by this signal is 30 kHz if the desired power ratio of 99% is known.

The numbers of BWs typically occupied change drastically, depending on the symbol and filtering rate. The value is around 30 kHz for the NADC π/4 DQPSK signal and around 350 kHz for a GSM 0.3 GMSK signal. For digital video signals, the occupied BW is generally 6–8 MHz.

Simple frequency measurement techniques are not often precise or sufficient enough to measure the central frequency. We calculate a "centroid" carrier, which is the frequency distribution center for the PSD for a modulated signal.

1.16.5. *Synchronization measures*

Synchronization measures are most often made in pulsed or in burst systems. The measures include intervals of repetition of pulses, in time, duty cycle, and the times between bit errors. Lighting or extinguishing times also involve power measures.

Modulation precision

Measures of modulation precision lead us to measure how close constellation states or the signal trajectory relates to an (ideal) trajectory of the reference signal.

Measures of modulation exactitude usually include precision demodulation of a signal and a comparison of this demodulated signal with the mathematically generated “reference” signal. The difference, or residue, between the two is the modulation error, and it can be expressed by a range of means including the amplitude of the error vector module (EVM), the phase error, the error on I and the error on Q. The reference signal is subtracted from the demodulated signal, starting from a residual error signal. Once the reference signal has been subtracted, it is easier to see small errors that have been drowned or obscured by the modulation itself. The error signal can be examined in many ways: in the time domain or (since it is a vector quantity) depending on its I/Q components. A frequency transformation can also be achieved and the spectral composition of this signal error can be observed alone.

Let us return to the bases of the vector modulation: the bits are transferred on an RF carrier by changing the amplitude and phase of the carrier. At each transition of the symbol clock, the carrier occupies a unique position among several possible ones in the I/Q plane. Each position encodes a specific data symbol which is formed of one or more data bits. A constellation diagram indicates the effective positions (in other words, the size and phase relative to the carrier) for all the authorized 2^n symbols. To demodulate the incoming data, the amplitude and phase of the signal received for each clock transition should be accurately determined.

The drawing of the constellation diagram, including the ideal locations for the symbols, is decided generically by the modulation format chosen (BPSK, 16QAM, $\pi/4$ DQPSK, etc.). The trajectory taken by the signal from one digital location to another depends on implementation of the dedicated system.

At any moment, the phase or amplitude of the signal can be measured. These values define the real or "measured" *phaser*. At the same time, an ideal "reference" phaser can be calculated, given the knowledge of the data flow transmitted, symbol-clock synchronization, baseband filtering parameters, etc. The differences between these two phasers form the basis for EVM measurements.

The schema in Figure 1.105 defines the EVM and several related limits. As shown, the EVM is the scalar distance between the phaser's two limits, i.e. the importance of the difference vector. In other words, it is the residual noise and the residual distortion that remain after an ideal version of the signal has been extracted.

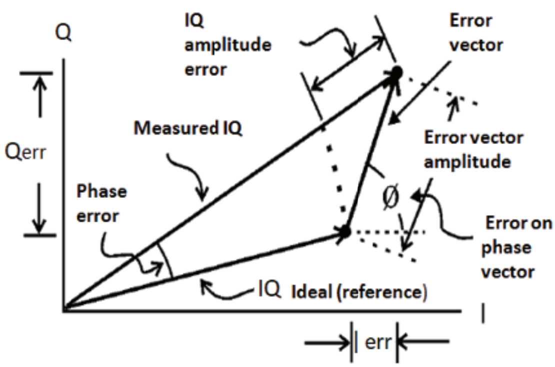


Figure 1.105. *Different types of errors*

In standard US NADC-TDMA (IS-54), the EVM is defined as a percentage of the signal voltage at the symbols. In the $\pi/4$ DQPSK modulation format, these symbols all have the same level of voltage, which is not true of all formats. IS-54 is currently the only standard that defines EVM explicitly; so EVM can be defined differently than other modulation formats. In a format such as 64QAM, the symbols, representing all levels of the EVM's voltage, can be defined by an average level of voltage for all the symbols (a value close to the average level of the signal) or by the highest voltage of the four exterior symbols, although the error vector has a phase value, generally random, linked to the symbol values.

This is a function of the error itself (which may or may not be random) and the position of the data symbol on the constellation which, in practice, is random. A more interesting angle is measured between real and ideal phasers (1/Q phase error) that contains the information useful in signal-reconstruction problems. Similarly, the I/Q amplitude error shows the difference in amplitude between the real and ideal signals. The EVM, as indicated in the standard, is the RMS value of the error values at the instant of the clock symbol's transition. The trajectory errors between the symbols are avoided.

Trouble shooting error vector measurements

Error vector amplitude measurements of the quantity concerned can, once correctly applied, provide a great deal of information on the quality of a digitally modulated signal. They can also indicate the exact causes of problems encountered by identifying precisely the type of degradation effective in a signal; indeed, they can even help identify their sources. EVM measurements are developed rapidly, having already been written in system standards such as NADC and PHS (phone standard), and they appear in several standards including those for digital visual transmission.

Amplitude versus phase error

Different error mechanisms affect signals in different ways: the amplitude alone, the phase alone, or both simultaneously. The relative quantities of each type of error can quickly confirm or eliminate certain types of problems. The last diagnostic stage is to solve the EVM in its amplitude and phase error components, and to compare their relative sizes. When the average phase error (in degrees) is substantially larger than the average amplitude error (in percentage), an unwanted PM becomes the dominant error mode. This can be triggered by noise, inter-pairing problems in the frequency reference, phase-locked loops, or other frequency synthesizing stages. The residual AM is demonstrated by amplitude errors, which are substantially greater than angle errors.

IQ phase error over time

The phase error is the instantaneous angle difference between the signal measured and the ideal reference signal. Once seen as a function of time (or of the symbol), this shows the modulation wave form of any PM signal (phase magnitude), residual or interfering. Sinusoidal waves or other regular wave forms indicate an interfering signal, a residual PM/FM noise, etc.).

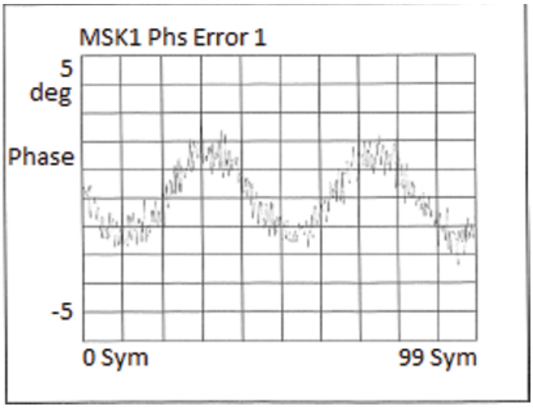


Figure 1.106. *Phase error*

An ideal signal will have a uniform constellation, perfectly symmetrical with the original. The 1/Q imbalance appears when the constellation is not "squared", i.e. when the height of the Q axis does not equal the breadth of the I axis. The quadrature error is seen at any tilt of the constellation. Quadrature error is caused when the phase ratio between vectors I and Q varies from 90°.

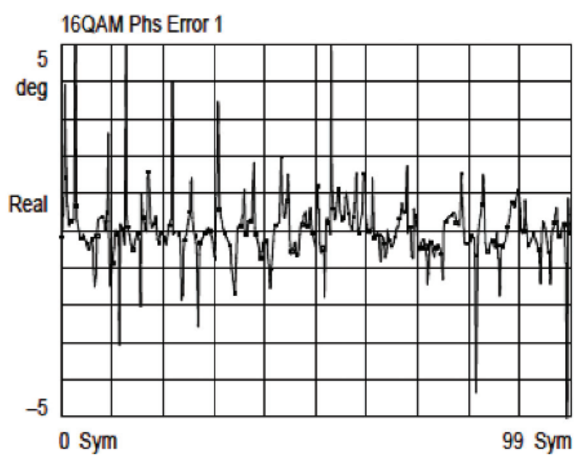


Figure 1.107. *Phase noise versus time: appears random (Agilent)*

The error vector as a function of time

The EVM is therefore the difference between the incoming and the ideal reference generated internally. Seen as a function of the symbol or of time, the errors can be correlated to specific points on the incoming wave forms, such as peaks or passages to zero. The EVM is a scalar value (amplitude alone). Peak errors are produced with the signal's peaks; they indicate compression or clipping phenomena. Error peaks correlated with minima suggest nonlinearities in the passage to zero.

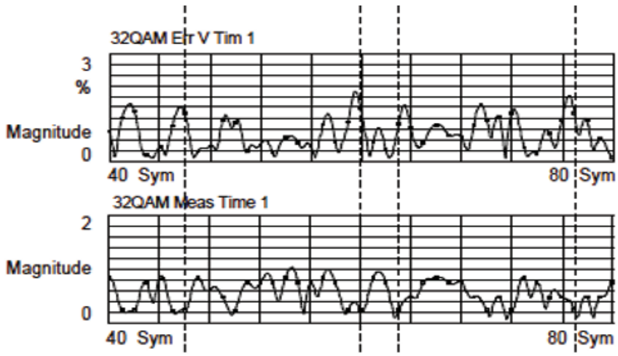


Figure 1.108. *EVM peaks (above) appear during the amplitude's passage to zero (lower) (see phase noise)*

One example of *zero-crossing* nonlinearities involves *push-pull* amplifiers (*one of the components is a source (of current), whereas the other is a reservoir, or sink, for the current*); the positive and negative halves of the signal are provided by additional transistors (e.g. Bipolar, NPN and PNP; MOS, N and P, mounted on a totem pole, in phase. This can be a real challenge, particularly in high power amplifiers) with precision to polarize and stabilize the amplifiers, one being dimmed while the other is on, without creating distortions. The critical moment is the passage to zero, an effect well-known in analog design (diode-mounted transistors are then added so that the currents are not cancelled on a voltage range around the commutation of additional transistors).

The error spectrum (EVM in terms of frequency)

The error spectrum is calculated using the waveform's FFT and the EVM results in the frequency domain. In most digital systems, the non-uniform distribution noise, or the peaks of discreet signals, indicate the presence of coupled external interferences.

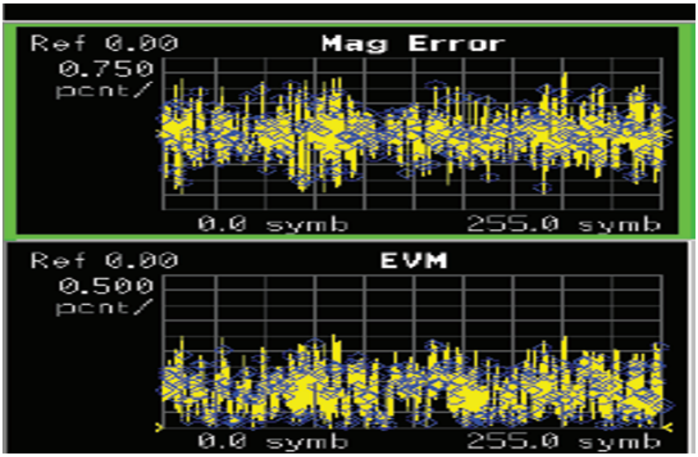


Figure 1.109. *EVM peaks (above) appear during the amplitude's passage to zero (below) (see phase noise – measures). For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

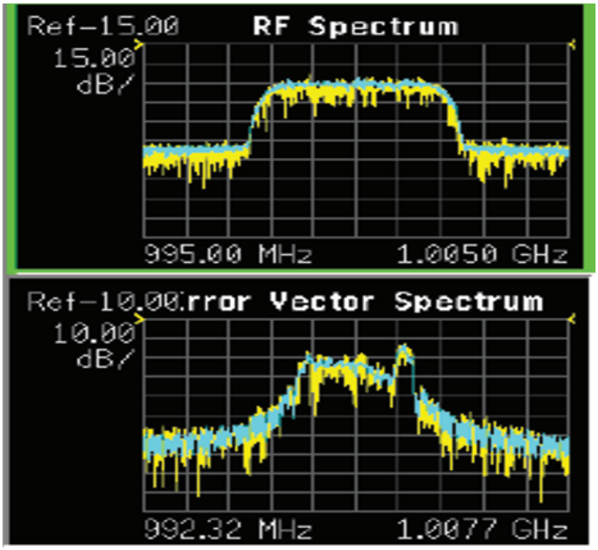


Figure 1.110. *RF spectrum (above) and error vector spectrum (below) (QPSK). For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

1.17. Applications

1.17.1. *Domains*

There is a "plethora" of domains for applying the different digital transmission techniques that we have just demonstrated. Some are described below.

Telephone modems

During the 1960s and 1970s, the transmission of data along telephone channels was at the origin of the development of many signal processing techniques in telecommunications. The transmission of a high flow on a telephone channel (on a frequency band of around 3,500 Hz) necessitated the manufacture of modulators with a large number of states: QAM-16, QAM-32 and QAM-128.

It was thought that flow would typically be limited to 9.6 kbit/s, since the S/B ratio of the links was prohibitive. In fact, coding and filtering techniques and the use of lattice constellations made it possible to cross a significant limitation on the quality and flow that could be attained.

Microwave radios

When digitization began, microwave radios used simple modulations such as 4-PSK, but efficient use of the accessible radio spectrum pushed the development of microwave radios using modulations with a high number of states such as QAM-16 and QAM-64. It was QAM-16 modulation that made possible the transmission of a flow of 140 Mbit/s in the 6.4–7.1 GHz band for channels spaced at 40 MHz.

Note that the transmitter should have a good linearity to transmit this type of modulation.

The crucial problem in digital microwave radios is propagation by multiple paths, which greatly degrades quality and limits the possibility of high-capacity links. This phenomenon is accentuated when the number of modulation states increases.

Unlike high-capacity microwave radios, there is low cost low BW radio (2 Mbit/s), in which spectral efficiency is not the first concern. The modulations used are generally MDF-PC at two or four states, which makes it possible to use a nonlinear amplification in the transmitter.

Satellite transmissions

Satellite transmissions display significant attenuation in space and a more restricted power from their transmitter. This calls for power efficiency (immunity from noise) versus spectral efficiency of links. The modulations most often used are PSK-2, PSK-4 and PSK-8.

With the latter, the amplifier allows the available power to be used efficiently. However, today, PSK-16 and QAM-16 modulations, linked to powerful coding, are in vogue. The standard in Europe for radio broadcasting digital television by satellite is based on a PSK-4.

Radiocommunications with cell phones

We know that digital radiocommunications systems cover the entire world. Japanese and American cellular systems use a different modulation to the one used in European systems. The modulation used in the United States and in Japan is $\pi/4$-DQPSK, a PSK-4 whose axes turn by $\pi/4$ from one symbol to the next. Phase rotations of π are therefore forbidden in this modulation. Signal envelope passages through zero are thus avoided and this considerably reduces its temporal fluctuations.

The modulation used in the European cellular system, called GSM (*Groupe Spécial Mobile*), is a constant-envelope modulation known by the name of GMSK (Gaussian minimum-shift keying). This is in fact a variant of MSK modulation, in which the impulses coming into the modulator are Gaussian. This temporal and spectral shaping smooths the signal's phase trajectory and reduces its spectral occupation compared to initial MSK modulation; the Nyquist criteria is therefore respected.

The data stream sent in a 200 kHz band is a multiplex of eight telephone channels. Taking into account the error correction code, the synchronization bits and identification of the channel, as well as other auxiliary data, the overall flow is around 270 kbit/s.

Cellular systems	**American**	**Japanese**	**European**
Standard	IS-54/-56 P	DC	
Frequency range	Rx:869-894 Tx:824-849	Rx:810-826 Tx:940-956	Rx:925-960 Tx:880-915
Cellular systems	American	Japanese	European
Standard	IS-54/-56 P	PDC	GSM
Frequency range	Rx:869-894 Tx:824-849	Rx:810-826 Tx:940-956	Rx:925-960 Tx:880-915
Number of channels	832	1,600	124
Number of users per channel	3	3	8
Channel spacing	30 kHz	25 kHz	200 kHz
Modulation	π/4-DQPSK	π/4 DQPSK	GMSK
Binary flow	48.6 kbit/s	42 kbit/s	270 kbit/s

Table 1.4. *Cellular systems*

1.17.2. *Digressions or precisions, around modulations*

QAM is a form of carrier modulation using modification of the *amplitude* of the carrier itself and a quadrature wave (a wave *dephased* by 90° with the carrier) according to the information transported by input signals.

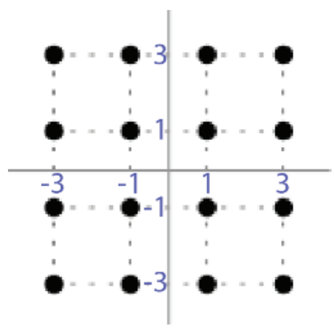


Figure 1.111. *Diagram of constellations for QAM at 16 states. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

In other words, this can be considered (using a complex notation) as a simple wave AM, expressed in complex, by a signal.

This means that the amplitude and phase of the carrier are simultaneously modified according to the information to be transmitted.

PM can also be considered as a particular instance of QAM, where only the phase varies. This remark can also be extended to FM, as this can be seen as a particular instance of PM.

```
M = 64;
refC = qammod(0:M-1,M);% - MATLAB Inc.-
constDiagram = comm.ConstellationDiagram(...
    'SymbolsToDisplaySource','Property', ...
    'SymbolsToDisplay',256, ...
       'XLimits',[-10 10],'YLimits',[-10 10], ...
   'ReferenceConstellation',refC);
   iqImbComp = comm.IQImbalanceCompensator('StepSizeSource','Input
port', ...
 'CoefficientOutputPort',true);
nSym = 25000;
data = randi([0 3],600,1);
txSig = pskmod(data,4,pi/4,'gray');
iqImbComp = comm.IQImbalanceCompensator('AdaptInputPort',true, ...
    'StepSize',0.001,'CoefficientOutputPort',true);

ampImb = 5; % dB
phImb = 15; % deg
gainI = 10.^(0.5*ampImb/20);
gainQ = 10.^(-0.5*ampImb/20);
imbI = real(txSig)*gainI*exp(-0.5i*phImb*pi/180);
imbQ = imag(txSig)*gainQ*exp(1i*(pi/2 + 0.5*phImb*pi/180));
rxSig = imbI + imbQ;
[~,isCoef1] = iqImbComp(rxSig(1:200),true);
[~,isCoef2] = iqImbComp(rxSig(201:400),false);
[~,isCoef3] = iqImbComp(rxSig(401:600),true);

isCoef = [isCoef1; isCoef2; isCoef3];
plot((1:600)',[real(isCoef) imag(isCoef)])
grid
xlabel('Symbols')
ylabel('Coefficient Value')
legend('Real','Imaginary','location','best')
```

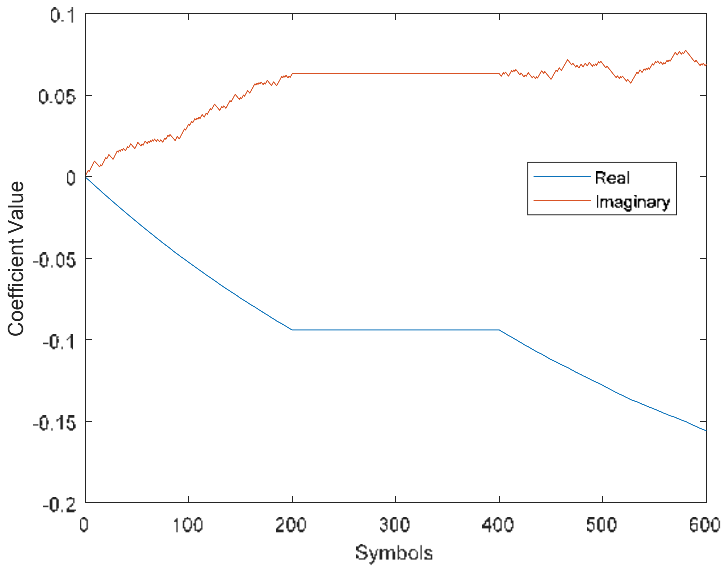


Figure 1.112. *I/Q imbalance measure – compensation coefficients. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

The I/Q imbalance compensator system object compensates the imbalance between the in-phase and the quadrature components of a modulated signal.

To compensate for the I/Q imbalance:

– Define and configure the object IQImbalanceCompensator (Matlab Inc.).

– Select "step" to compensate the I/Q imbalance according to the properties of comp.IQImbalanceCompensator. The "step's" behavior is specific to each object in the toolbox.

– The adaptive algorithm within the I/Q imbalance compensator is compatible with M-PSK, M-QAM and OFDM modulation schemas, where M > 2.

Attenuate the impacts of the amplitude and phase imbalance on a signal modulated by QPSK using the system object: comm.IQImbalance Compensator.

– Create an object “constellation” diagram.

```
constDiagram = comm.ConstellationDiagram(...
    'SymbolsToDisplaySource','Property', ...
    'SymbolsToDisplay',100).
```

– Create an I/Q imbalance compensator: iqImbComp = comm. IQImbalanceCompensator.

– Produce random data symbols and apply QPSKdata = randi([0 3],1e7,1) modulation.

```
txSig = pskmod(data,4,pi/4);
% Apply amplitude and phase imbalance to the signal transmitted.
ampImb = 5; % dB
phImb = 15; % deg
gainI = 10.^(0.5*ampImb/20);
gainQ = 10.^(-0.5*ampImb/20);
imbI = real(txSig)*gainI*exp(-0.5i*phImb*pi/180);
imbQ = imag(txSig)*gainQ*exp(1i*(pi/2 + 0.5*phImb*pi/180));
rxSig = imbI + imbQ;
```

% Plot of the constellation diagram of the signal received. Observe that the signal received really has undergone % shifts in amplitude and phase.

```
constDiagram(rxSig)

hMod = comm.PSKModulator(8);
refC = constellation(hMod);
hScope = comm.ConstellationDiagram(...
    'SymbolsToDisplaySource','Property', ...
    'SymbolsToDisplay',100, ...
  'ReferenceConstellation',refC);
hIQComp = comm.IQImbalanceCompensator('CoefficientSource','Input
port');
data = randi([0 7],1000,1);
txSig = step(hMod,data);

ampImb = 5; % dB
phImb = 15; % deg
gainI = 10.^(0.5*ampImb/20);
gainQ = 10.^(-0.5*ampImb/20);
imbI = real(txSig)*gainI*exp(-0.5i*phImb*pi/180);
imbQ = imag(txSig)*gainQ*exp(1i*(pi/2 + 0.5*phImb*pi/180));
rxSig = imbI + imbQ;
```

```
step(hScope,rxSig);
% Sequence: deletion of I/Q imbalance
%Apply the I/Q compensation algorithm and display the constellation. %The
constellation of compensated signals is almost aligned on the
%reference constellation

compCoef = iqimbal2coef(ampImb,phImb);
compSig = step(hIQComp,rxSig,compCoef);
step(hScope,compSig)

% Object:  comm.IQImbalanceCompensatorSystem ™ with external
coefficients.
% Create 8-PSK modulator system objects and a constellation diagram.
% Use non-value pairs to ensure that the %constellation diagram displays
only the last 100 data symbols and
 %to provide the reference constellation.

hMod = comm.PSKModulator(8);
refC = constellation(hMod);
hScope = comm.ConstellationDiagram(...
   'SymbolsToDisplaySource','Property', ...
   'SymbolsToDisplay',100, ...
   'ReferenceConstellation',refC);
% Create an I/Q imbalance compensator object with an input port %for the
algorithm's coefficients.
hIQComp = comm.IQImbalanceCompensator('CoefficientSource','Input
port');
% Generate random data symbols (see random) and apply the modulation
% 8-PSK.
data = randi([0 7],1000,1);
txSig = step(hMod,data);
% Apply an amplitude and phase imbalance to the signal transmitted.

ampImb = 5; % dB
phImb = 15; % deg
gainI = 10.^(0.5*ampImb/20);
gainQ = 10.^(-0.5*ampImb/20);
imbI = real(txSig)*gainI*exp(-0.5i*phImb*pi/180);
imbQ = imag(txSig)*gainQ*exp(1i*(pi/2 + 0.5*phImb*pi/180));
rxSig = imbI + imbQ;
```

%Plot of the constellation diagram of the signal received. Observe that the %signal received has shifted in amplitude and phase.

```
step(hscope,rxsig);
```

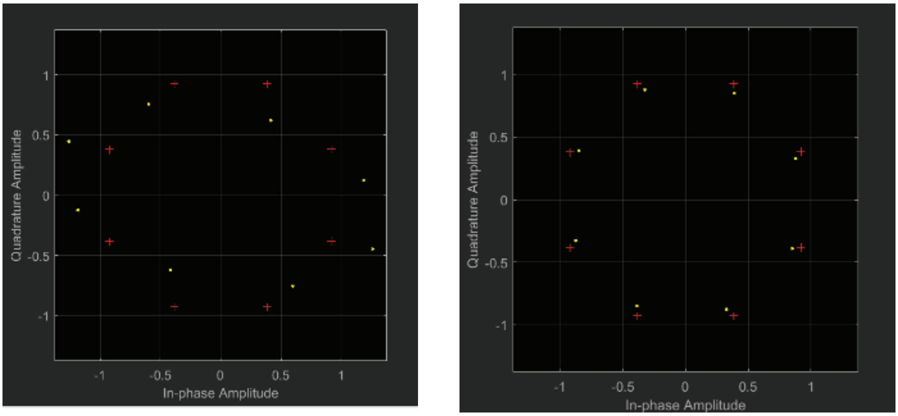


Figure 1.113. *Removal of I/Q imbalance*