Series Editor
Guy Pujolle

Digital Communication Techniques

Christian Gontrand

Color section

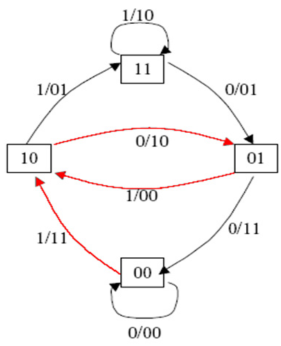


Figure I.7. *Transition diagram*

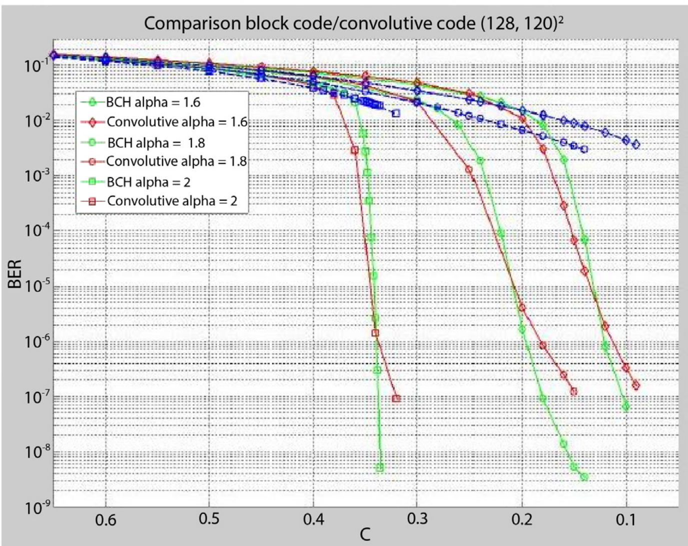


Figure I.14. *Turbocodes*

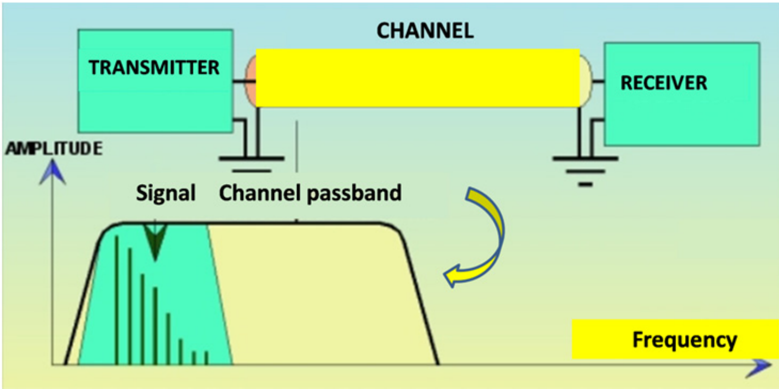


Figure 1.1. *Transmission of a signal in baseband*

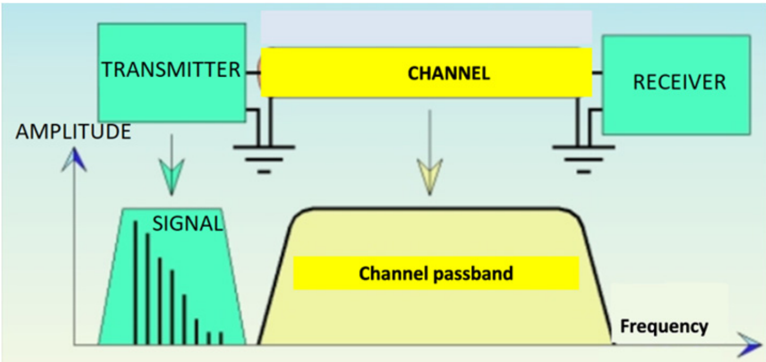


Figure 1.2. *Transmission of a signal via modulation*

Modulations- (*heterodyne*) **frequency transpositions**

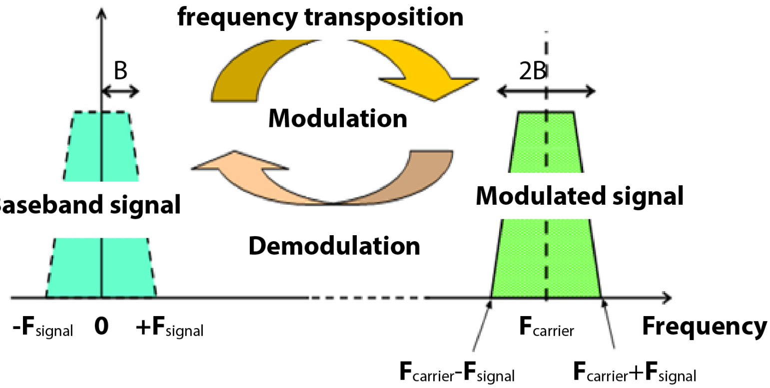


Figure 1.3. *Heterodyne system*

Probability of error

'Over the course of the transmission, the useful signal is 'attenuated' at the same time as a parasite signal is superposed on it.'

Suppose that the noise n(t) has the following properties

- **Null** average value
- Average quadratic value (standard deviation) such as σ2 is the variance, it is also the standardized noise power

$$\sigma^2 = \lim_{T\to\infty} \frac{1}{T}\int_{-T/2}^{T/2} n^2(t)dt$$

- Gaussian process. The probability that n(t) lies between s and s+ds is p(s).ds with

$$p(s) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{s^2}{2\sigma^2}}$$

- **Stationary** process (independent of time) and **ergodic** (statistical average = temporal average)
- Unilateral DSP is N=cte (white noise), the standardized power is equal, in frequency band B (equivalent noise bandwidth) at No. B

Figure 1.10. *Probability of error*

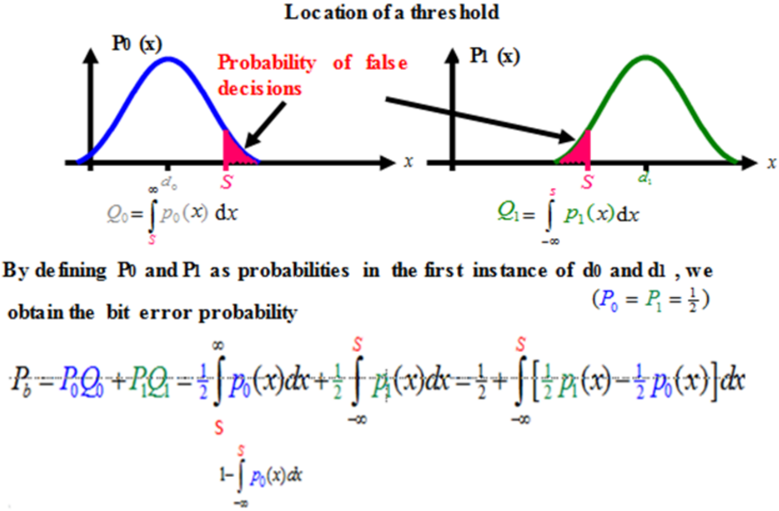


Figure 1.11. *Probability of wrong decisions*

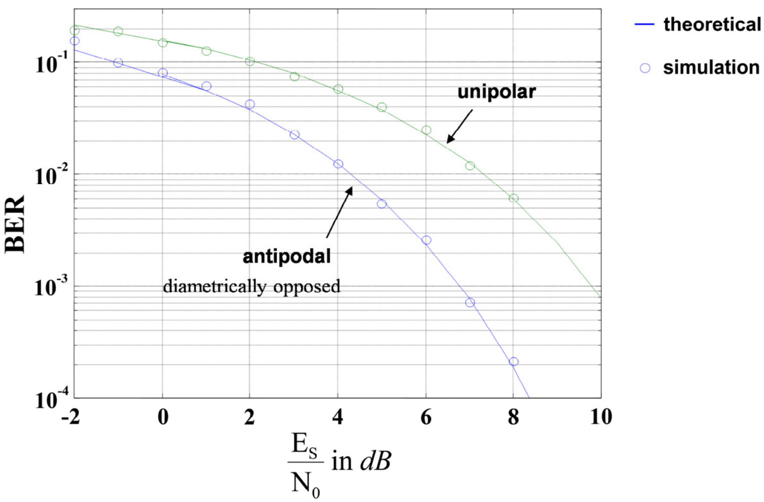


Figure 1.12. *Error rate by bit, for a unipolar and antipodal transmission, according to the signal to noise ratio*

Baseband signal

"0" ⟵ Level u_1 "1" ⟵ Level u_2

Probability of error, i.e. of deciding that 1 has been received while a 0 was transmitted (or vice versa), is given by:

$$p_e = \int_{\frac{u_1+u_2}{2}}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-u_1)^2}{2\sigma^2}} dx$$

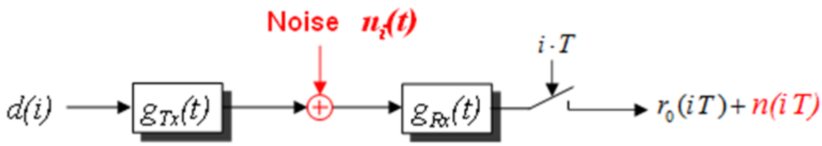
The *complementary error function* is generally involved :

$$erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-u^2} du$$

Finally

$$p_e = \frac{1}{2} erfc\left(\frac{u_2-u_1}{2\sigma\sqrt{2}}\right)$$

Figure 1.13. *Probability of error in erfc (erf complementary)*



Hypotheses.

- Binary transmission, with: : $d(i) \in [d_0, d_1]$
- Transmission system verifying the first Nyquist criteria
- Noise $n(iT)$, independent of the data source

Probability density $n(iT)$
Average and variance

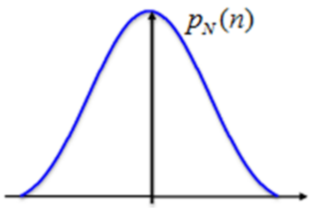


Figure 1.14. *Probability of error by bit*

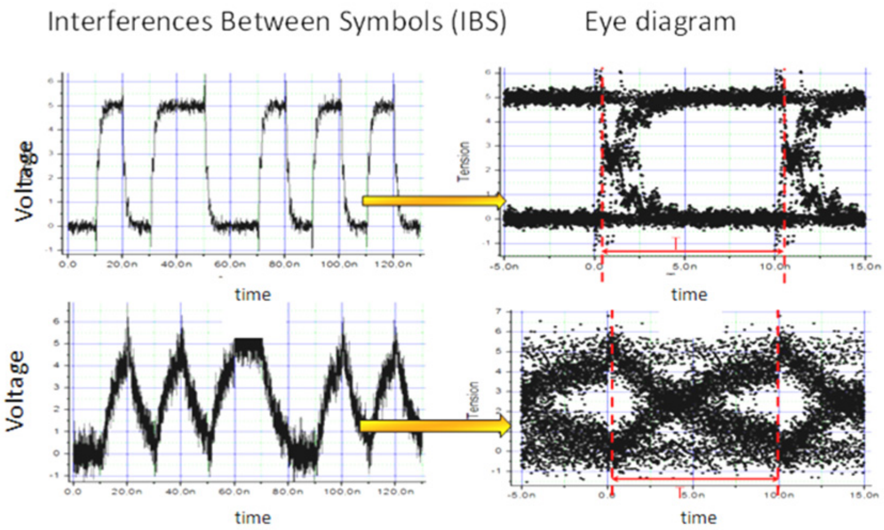


Figure 1.22. *Intersymbol interferences and eye diagrams*

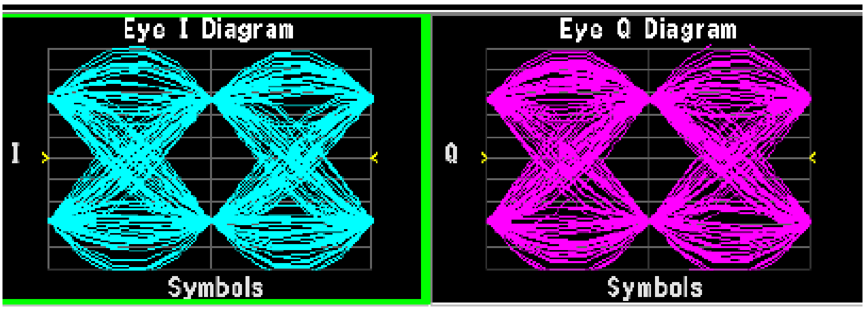


Figure 1.23. *Eye diagrams (e.g. QPSK; Agilent)*

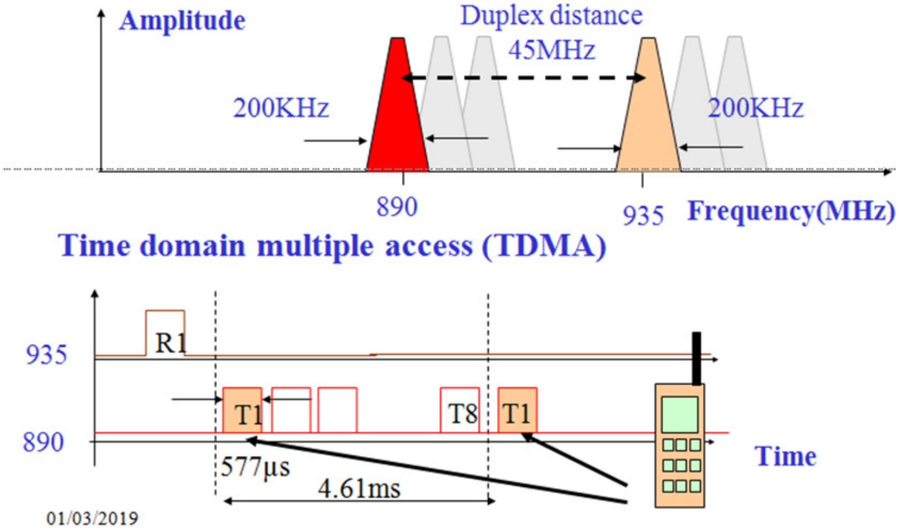


Figure 1.25. *CDMA: all users on each frequency and users are separated by code*

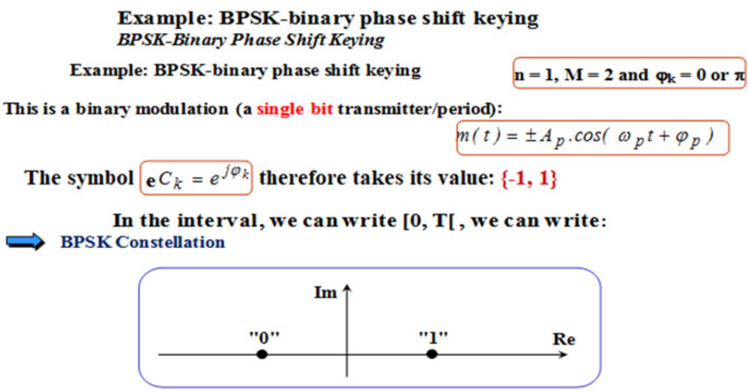


Figure 1.30. *BPSK*

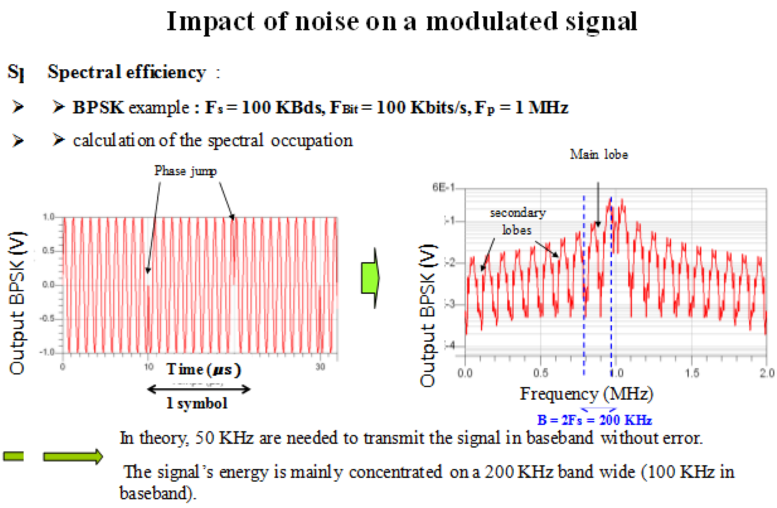


Figure 1.36. *Spectral efficiency of a BPSK*

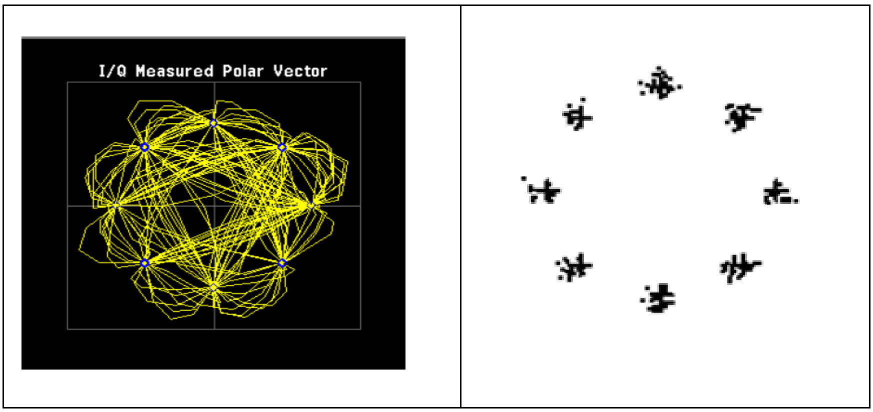


Figure 1.37. *Constellation diagram (Agilent)*

M-ary digital modulations – <u>OPSK</u>

➢ Quadrature phase-shift key (QPSK) demodulation (QPSK)

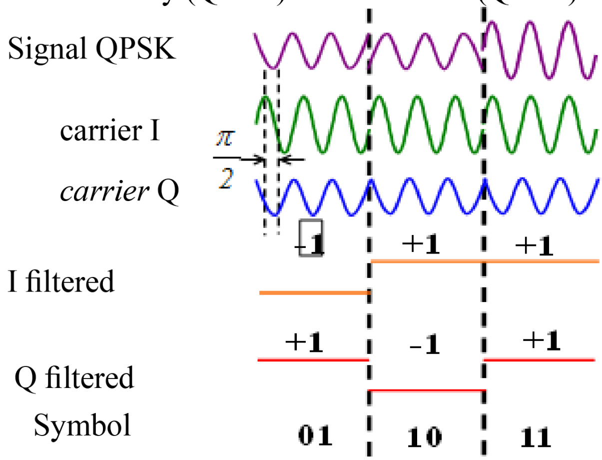


Figure 1.40. *QPSK: I/Q*

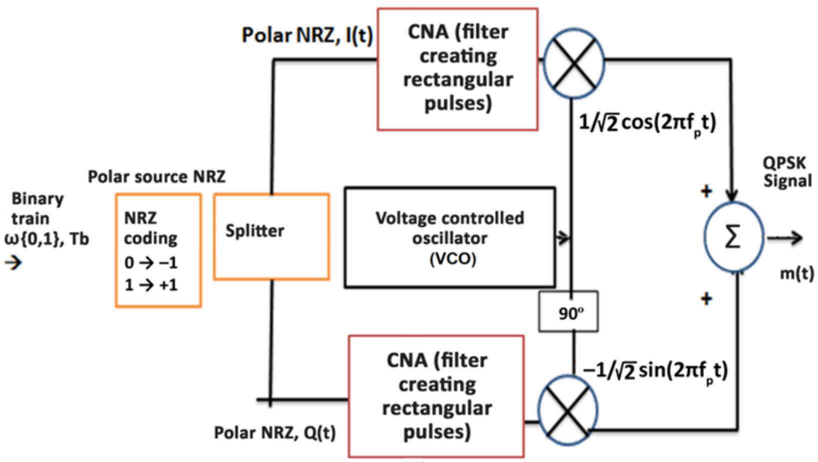


Figure 1.42. *QPSK modulator*

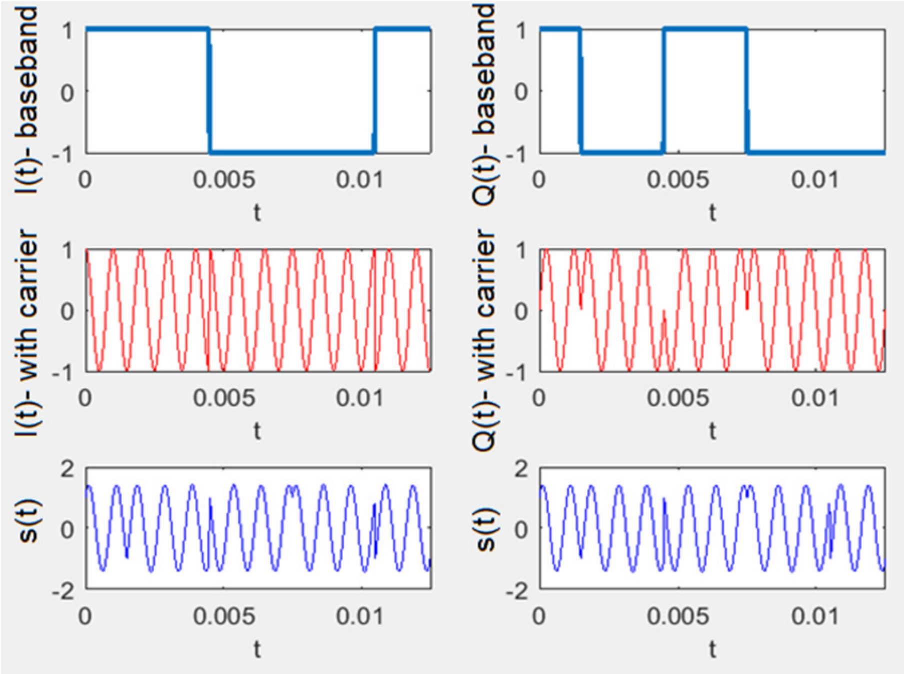
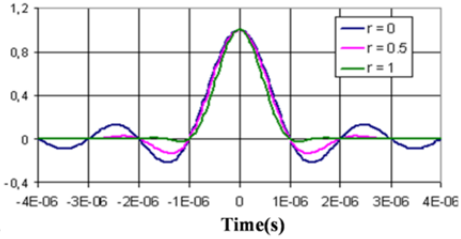


Figure 1.44. *QPSK; at the transmitter: timing diagrams*

<u>**Filtering – Pulse shaping**</u>

Exact midde between rectangular impulses and **sinc** impulses: **raised cosine** impulse

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



Ts: sampling period

r: slip factor, or stiffening or excess band or ROLL-OFF FACTOR

Figure 1.51. *Filtering through various raised cosines*

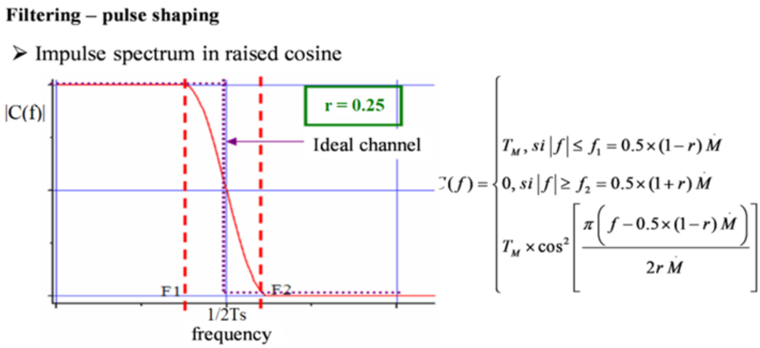


Figure 1.52. *Filtering an impulse in raised cosine*

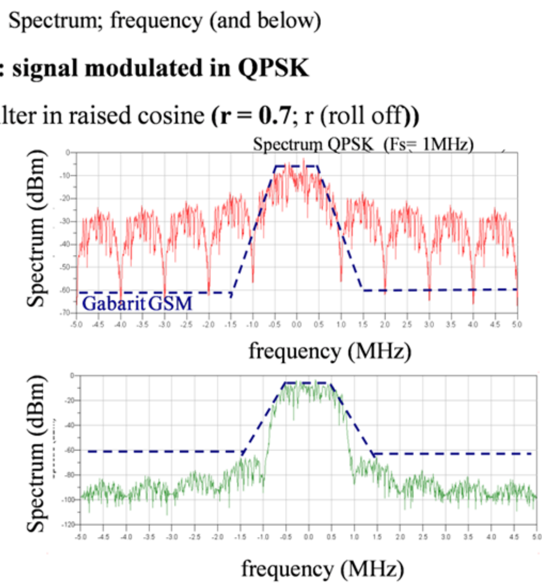


Figure 1.53. *Filtering a QPSK signal using a raised cosine*

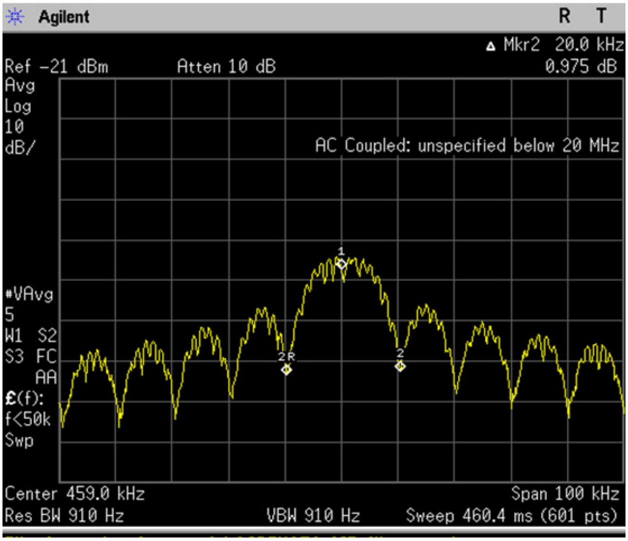


Figure 1.54. *Spectrum of the modulated QPSK signal for a binary flow of 20 Kbits/s without filtering streams I and Q (Agilent)*

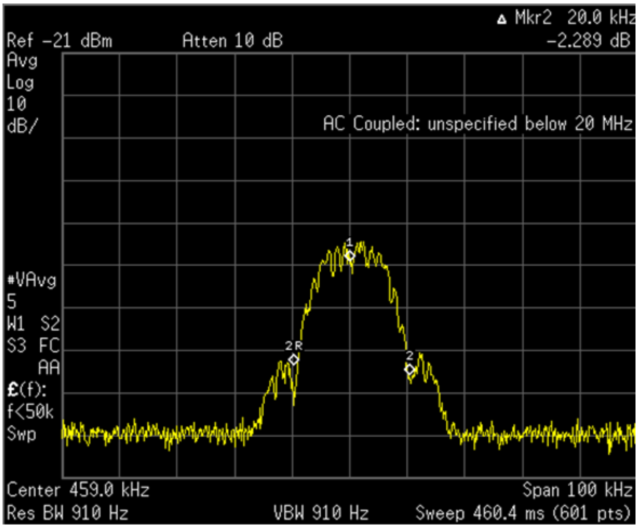


Figure 1.55. *Spectrum of the modulated QPSK signal for a binary flow of 20 Kbits/s with filtering (Agilent)*

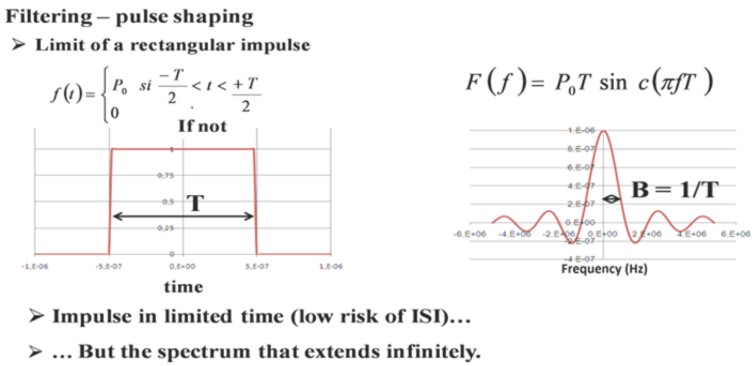


Figure 1.56. *Spectrum of a rectangular impulse*

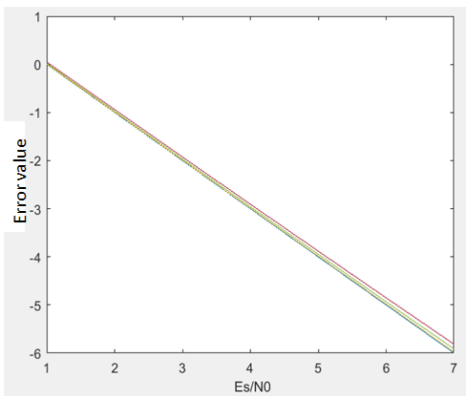


Figure 1.58. *Amplitude of the vector error depending on the signal/noise ratio*

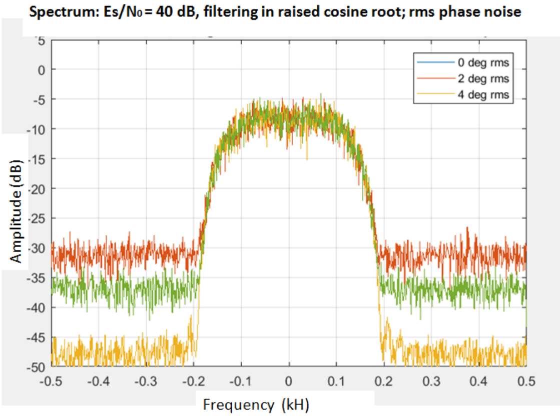


Figure 1.59. *Calculation of a typical spectrum of a vector error*

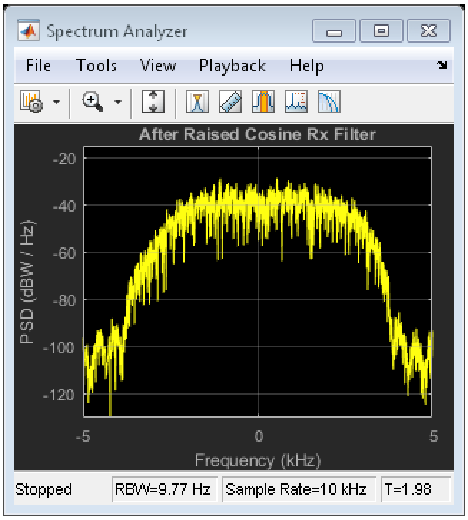


Figure 1.60. *Typical spectrum of a vector error, filtered by a raised cosine (MATLAB)*

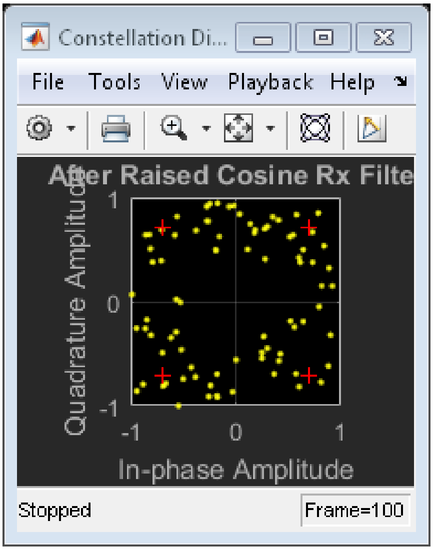


Figure 1.61. *Constellation after filtering in raised cosine*

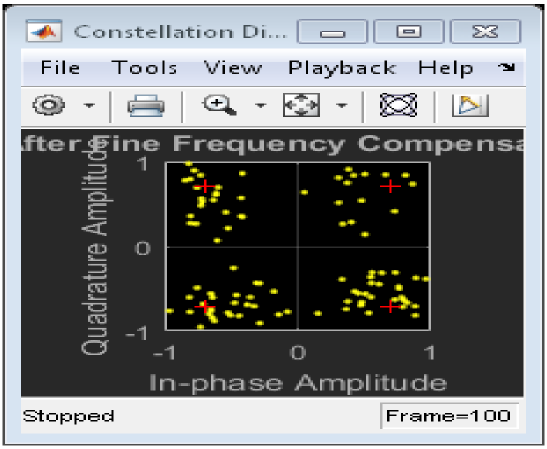


Figure 1.62. *Constellation after fine frequency offsets*

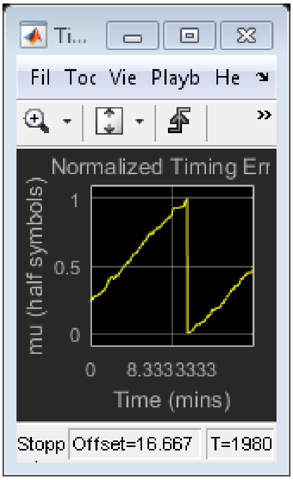


Figure 1.63. *Characteristic of a phase detector (the zig-zags are not ideal)*

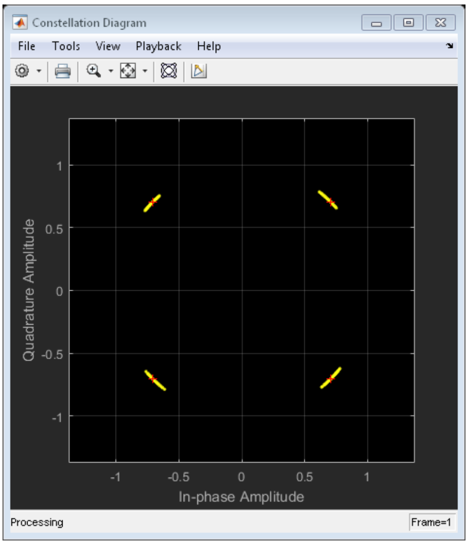


Figure 1.66. *Noisy in-phase constellation*

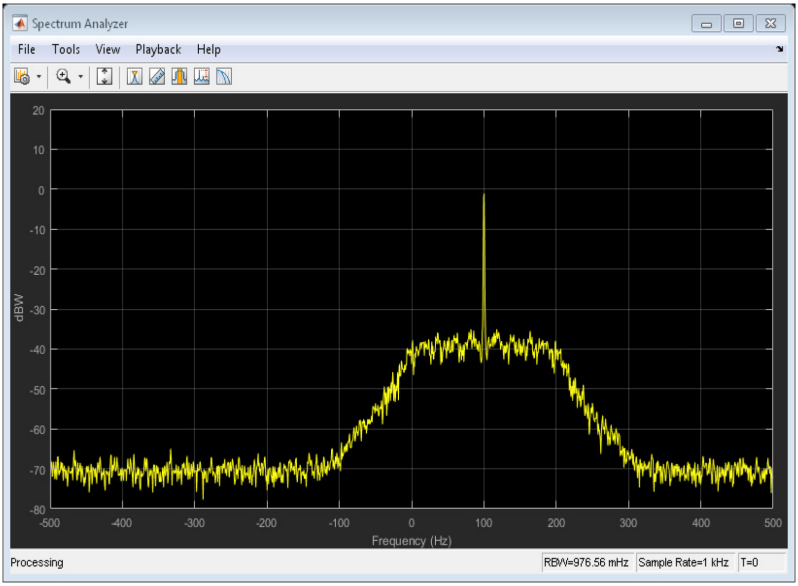


Figure 1.71. *Carrier (spur: see line) and phase noise*

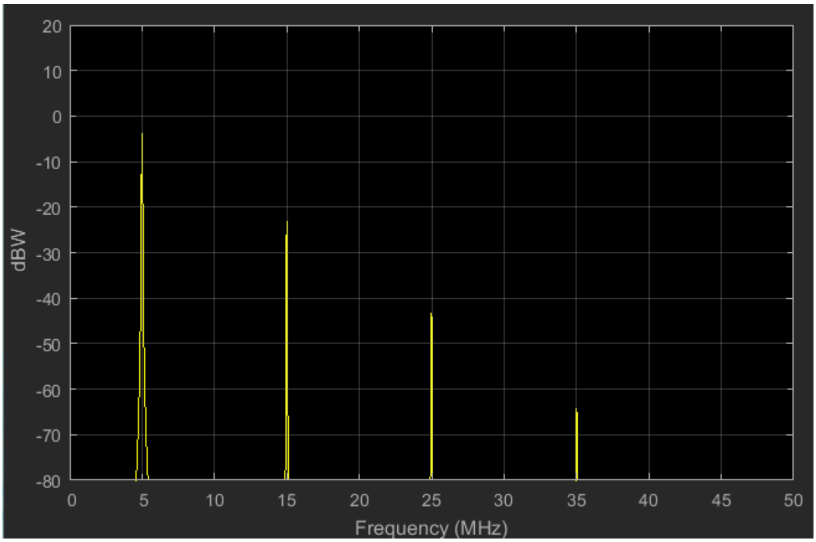


Figure 1.73. *Simulation of a spectrum analyzer*

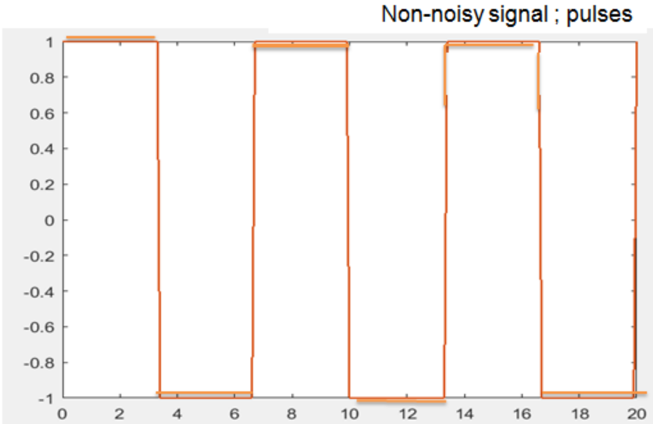


Figure 1.75. *At the transmitter: pulses: 101010*

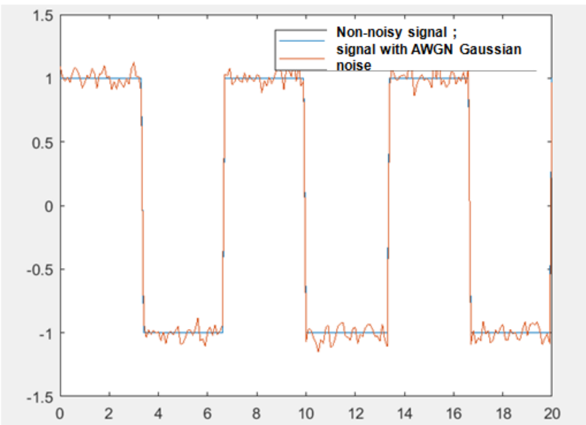


Figure 1.76. *In the channel: pulses: 101010*

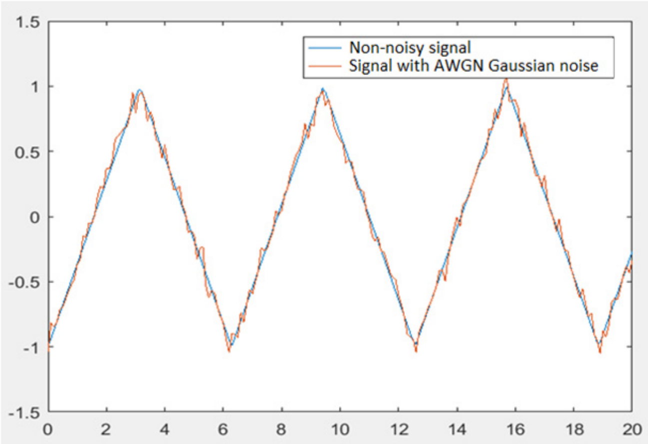


Figure 1.77. *Filtering (average, integration)*

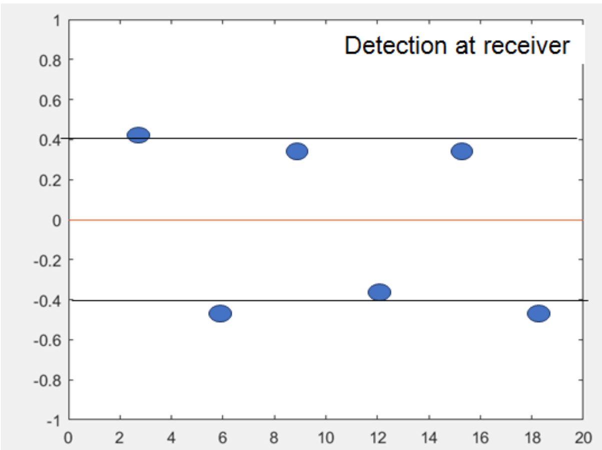


Figure 1.78. *Sampling/thresholding (without error) (101010)*

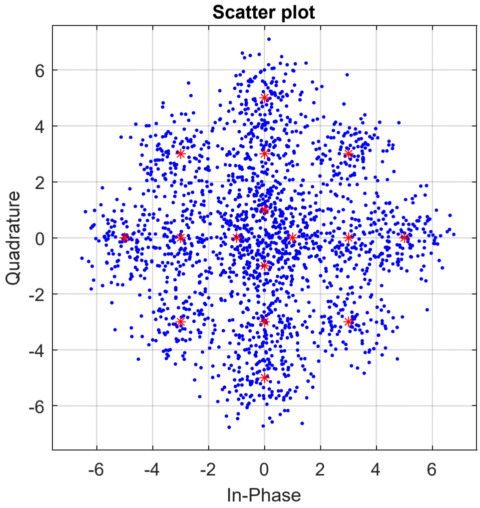


Figure 1.80. *Trace of scatter from a 16 QAM*

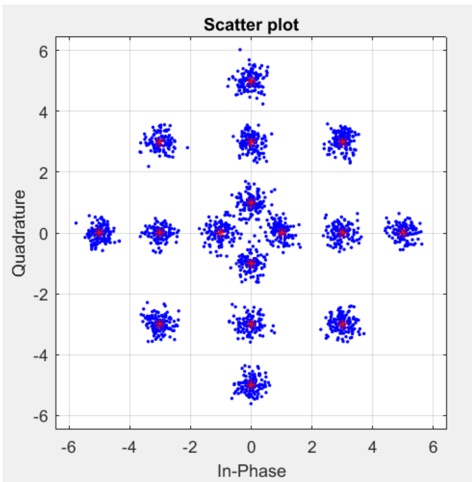


Figure 1.81. *16 constellations*

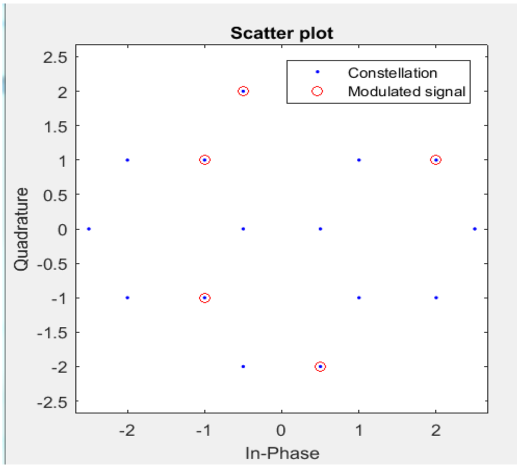
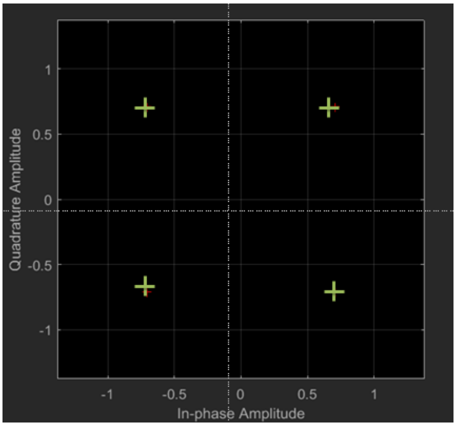
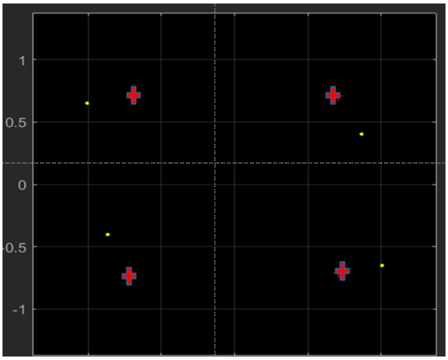
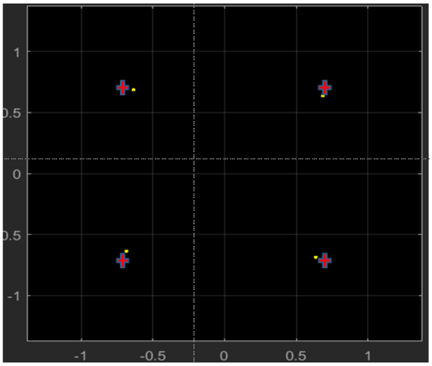


Figure 1.82. *Plot of a QAM constellation*



(a)





(b) (c)

Figure 1.84. *Removing I/Q imbalance*

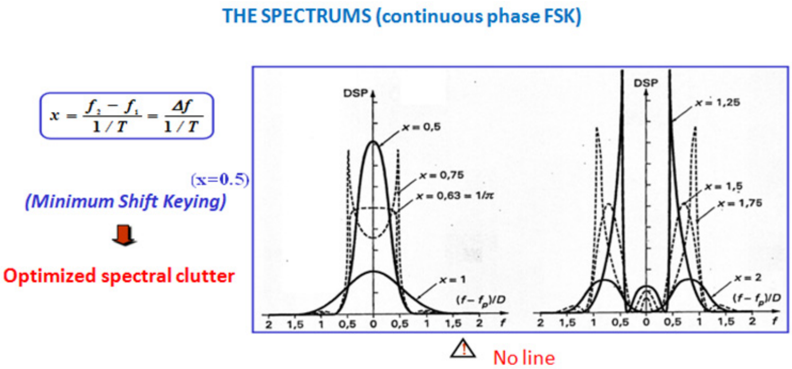


Figure 1.85. *Phase shifting*

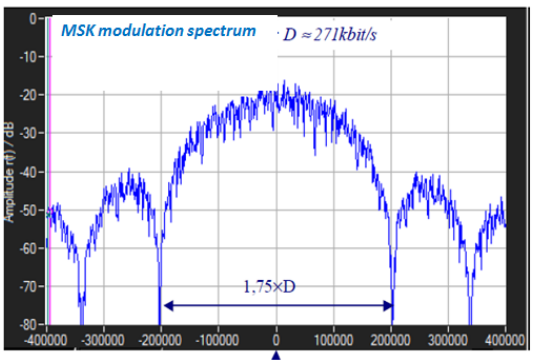


Figure 1.86. *Minimum-shift keying (MSK) modulation spectrum: continuous phase, modulation index: 0.5*

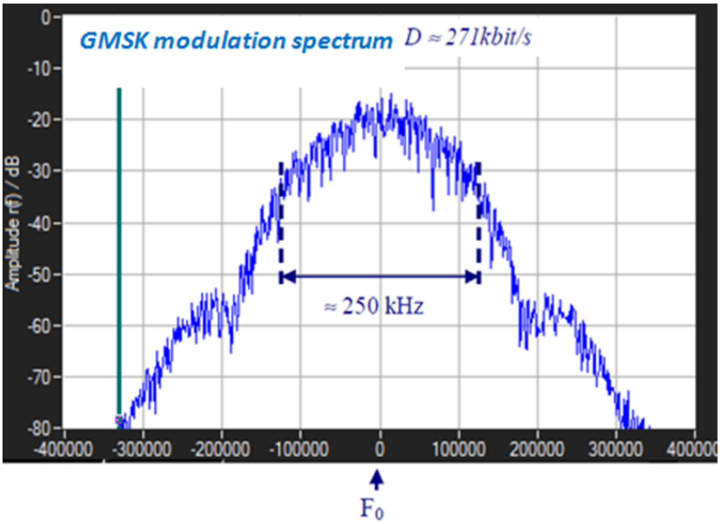


Figure 1.87. *Gaussian MSK (GMSK); the data are, from the outset, processed using a Gaussian filter*

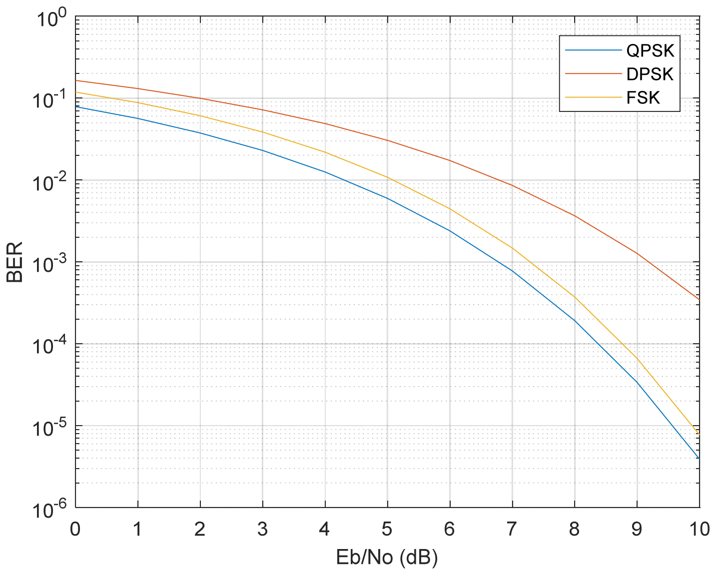


Figure 1.88. *BER for different modulations*

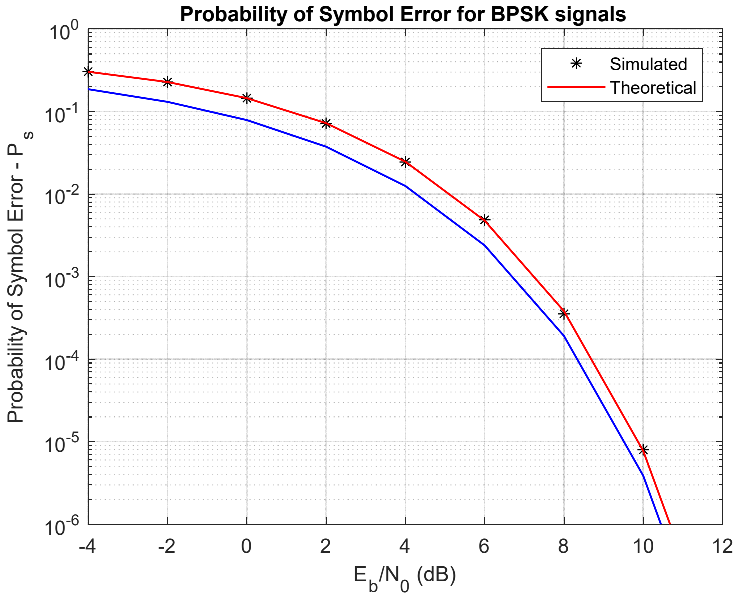


Figure 1.98. *BPSK: probability of symbol error*

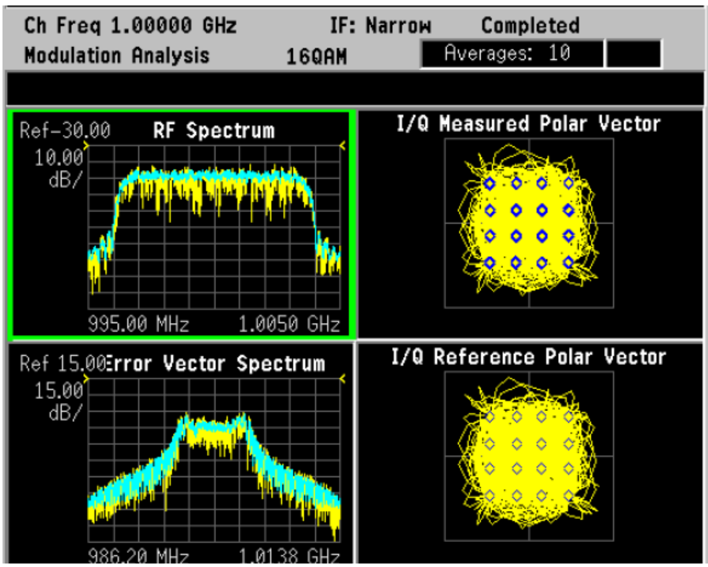


Figure 1.104. *Occupied bandwidth (Agilent)*

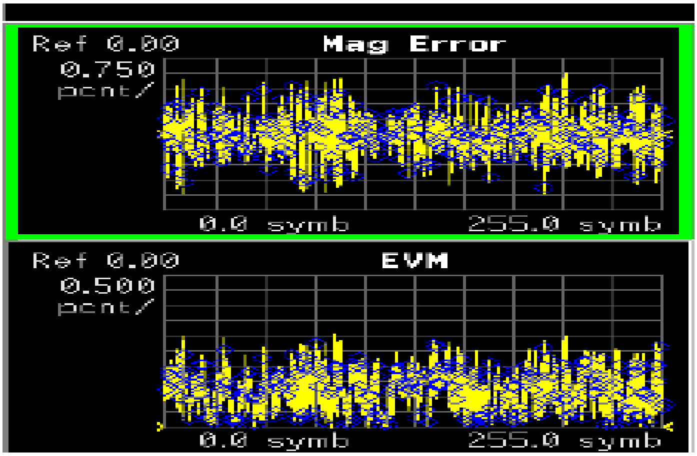


Figure 1.109. *EVM peaks (above) appear during the amplitude's passage to zero (below) (see phase noise – measures)*

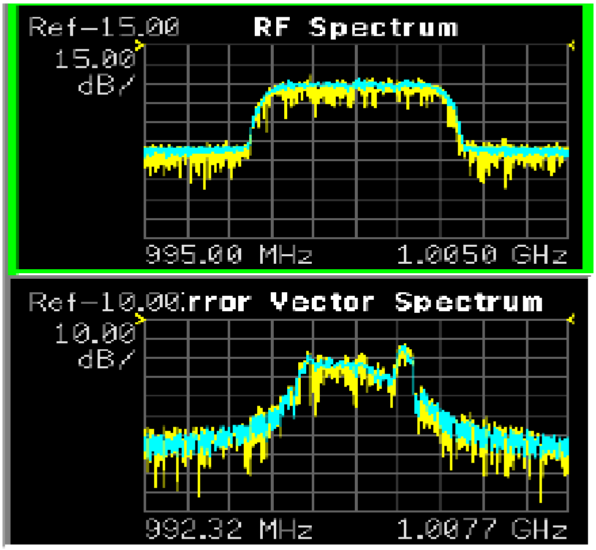


Figure 1.110. *RF spectrum (above) and error vector spectrum (below) (QPSK)*

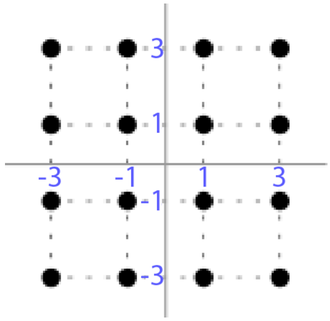


Figure 1.111. *Diagram of constellations for QAM at 16 states*

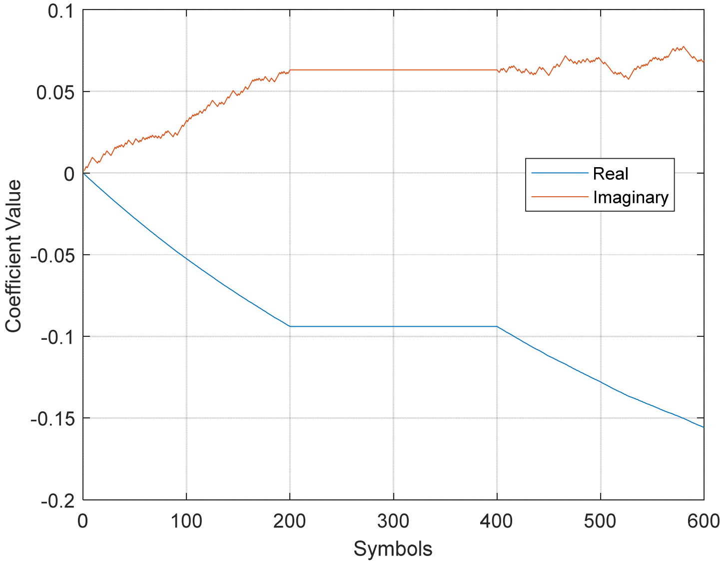


Figure 1.112. *I/Q imbalance measure – compensation coefficients*

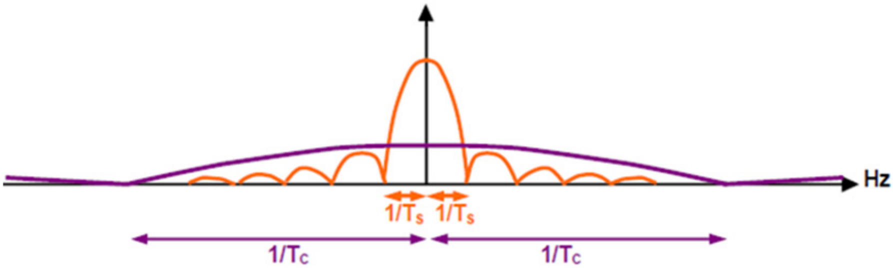


Figure 2.1. *Spectrum spread*

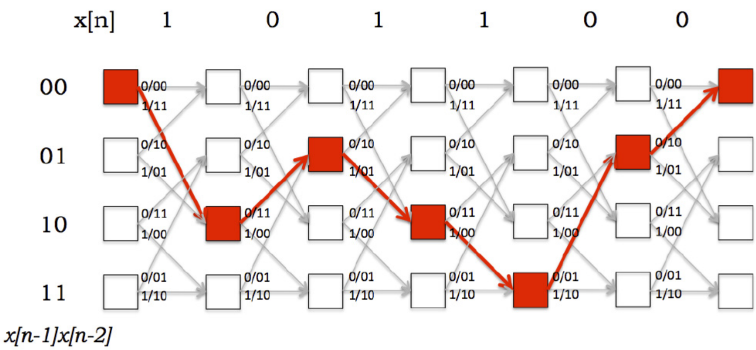
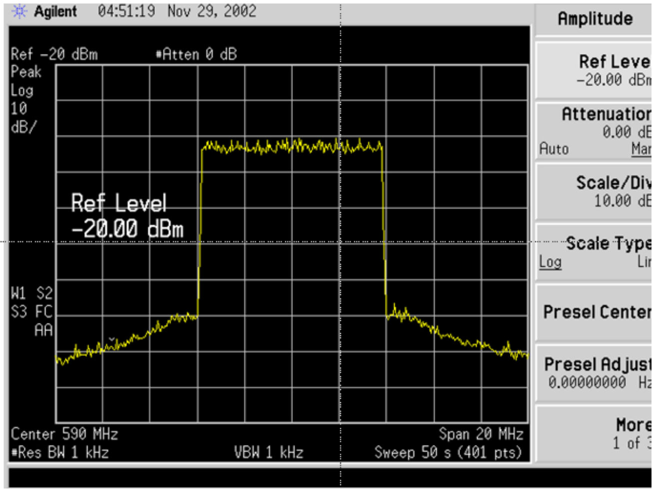
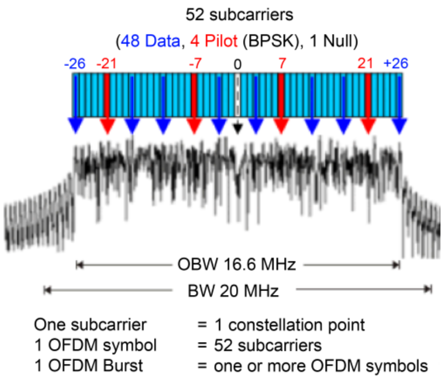


Figure 2.4. *Viterbi algorithm. The lattice makes it possible to visualize the decoding and grasp the temporal evolution (from left to right) of a state machine*



(a)

802.11a OFDM PHY Parameters	
BW	20 MHz
OBW	16.6 MHz
Subcarrier Spacing	312.5 KHz (20MHz/64 Pt FFT)
Information Rate	**6**/9/**12**/18/**24**/36/48/54 Mbits/s
Modulation	BPSK, QPSK, 16QAM, 64QAM
Coding Rate	1/2, 2/3, 3/4
Total Subcarriers	52 (Freq Index -26 to +26)
Data Subcarriers	48
Pilot Subcarriers*	4 (-21, -7, +7, +21) *Always BPSK
DC Subcarrier	Null (0 subcarrier)



(b)

Figure 2.6. *(a) A typical OFDM spectrum (measure). (b) Example: signal and parameters of the physical layer OFDM 802.11a*

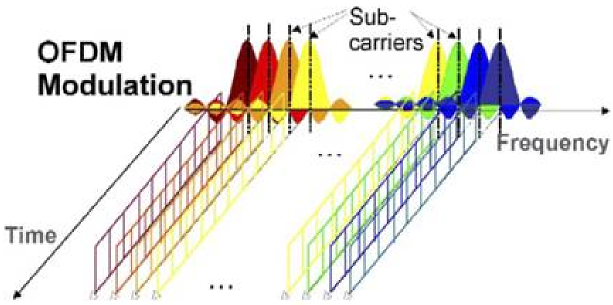


Figure 2.8. *OFDM spectrum*

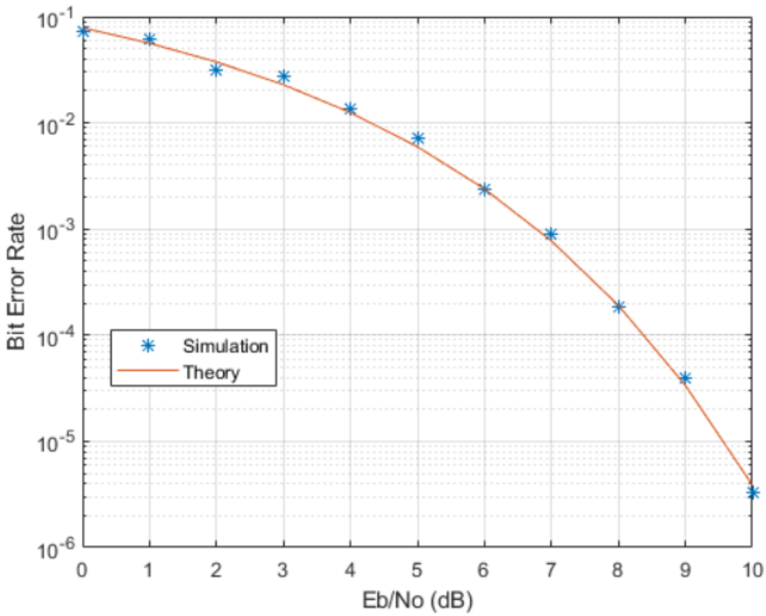


Figure 2.10. *BER versus Eb/No*

```
M = 4;                 % Modulation alphabet
k = log2(M);           % Bits/symbol
numSC = 128;           % Number of OFDM subcarriers
cpLen = 32;            % OFDM cyclic prefix length
maxBitErrors = 100;    % Maximum number of bit errors
maxNumBits = 1e7;      % Maximum number of bits transmitted
```

Construct System objects needed for the simulation: QPSK modulator, QPSK demodulator, OFDM modulator, OFDM demodulator, AWGN channel, and an error rate calculator. Use name-value pairs to set the object properties.

Set the QPSK modulator and demodulator so that they accept binary inputs.

```
qpskMod = comm.QPSKModulator('BitInput',true);
qpskDemod = comm.QPSKDemodulator('BitOutput',true);
```

Set the OFDM modulator and demodulator pair according to the simulation parameters.

```
ofdmMod = comm.OFDMModulator('FFTLength',numSC,'CyclicPrefixLength',cpLen);
ofdmDemod = comm.OFDMDemodulator('FFTLength',numSC,'CyclicPrefixLength',cpLen);
```

Set the `NoiseMethod` property of the AWGN channel object to `Variance` and define the `VarianceSource` property so that the noise power can be set from an input port.

```
channel = comm.AWGNChannel('NoiseMethod','Variance',...
'VarianceSource','Input port');
```

Set the `ResetInputPort` property to `true` to enable the error rate calculator to be reset during the simulation.

```
errorRate = comm.ErrorRate('ResetInputPort',true);
```

Use the `info` function of the `ofdmMod` object to determine the input and output dimensions of the OFDM modulator (Matlab Inc).

```
ofdmDims = info(ofdmMod)
```

ofdmDims = struct with fields:
DataInputSize: [117 1]
OutputSize: [160 1]

Determine the number of data subcarriers from the `ofdmDims` structure variable.

```
numDC = ofdmDims.DataInputSize(1)
```

numDC = 117

Determine the OFDM frame size (in bits) from the number of data subcarriers and the number of bits per symbol.

```
frameSize = [k*numDC 1];
```

Set the SNR vector based on the desired Eb/No range, the number of bits per symbol, and the ratio of the number of data subcarriers to the total number of subcarriers.

```
EbNoVec = (0:10)';
snrVec = EbNoVec + 10*log10(k) + 10*log10(numDC/numSC);
```

Initialize the BER and error statistics arrays.

```
berVec = zeros(length(EbNoVec),3);
errorStats = zeros(1,3);
```

Simulate the communication link over the range of Eb/No values. For each Eb/No value, the simulation runs until either `maxBitErrors` are recorded or the total number of transmitted bits exceeds `maxNumBits`.

```
for m = 1:length(EbNoVec)
snr = snrVec(m);

while errorStats(2) < = maxBitErrors && errorStats(3) < = maxNumBits
dataIn = randi([0,1],frameSize);              % Generate random binary data
qpskTx = qpskMod(dataIn);                     % Apply QPSK modulation
txSig = ofdmMod(qpskTx);                      % Apply OFDM modulation
powerDB = 10*log10(var(txSig));               % Calculate Tx signal power
noiseVar = 10.^(0.1*(powerDB-snr));           % Calculate the noise variance
rxSig = channel(txSig,noiseVar);              % Pass the signal through a noisy channel
qpskRx = ofdmDemod(rxSig);                    % Apply OFDM demodulation
dataOut = qpskDemod(qpskRx);                  % Apply QPSK demodulation
errorStats = errorRate(dataIn,dataOut,0);     % Collect error statistics
end

berVec(m,:) = errorStats;                     % Save BER data
errorStats = errorRate(dataIn,dataOut,1);     % Reset the error rate calculator
```

```
end
```

Use the `berawgn` function to determine the theoretical BER for a QPSK system.

```
berTheory = berawgn(EbNoVec,'psk',M,'nondiff');
```

Plot the theoretical and simulated data on the same graph to compare results

```
Figure
semilogy(EbNoVec,berVec(:,1),'*')
hold on
semilogy(EbNoVec,berTheory)
legend('Simulation','Theory','Location','Best')
xlabel('Eb/No (dB)')
ylabel('Bit Error Rate')
grid on
hold off
```

Box 2.1. *Bit error rate versus energy per bit*

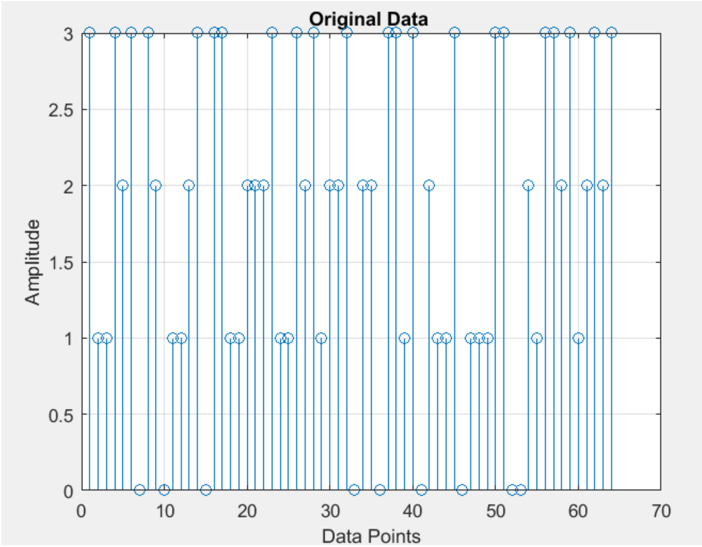


Figure 2.11. *Starting data*

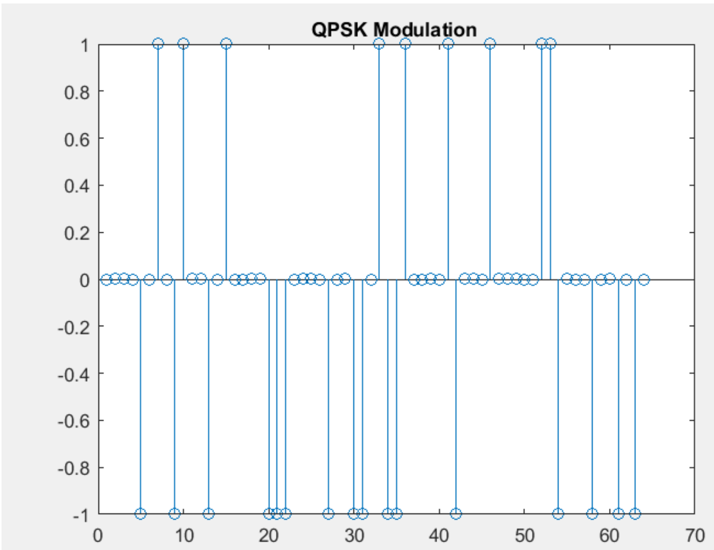


Figure 2.12. *QPSK modulations*

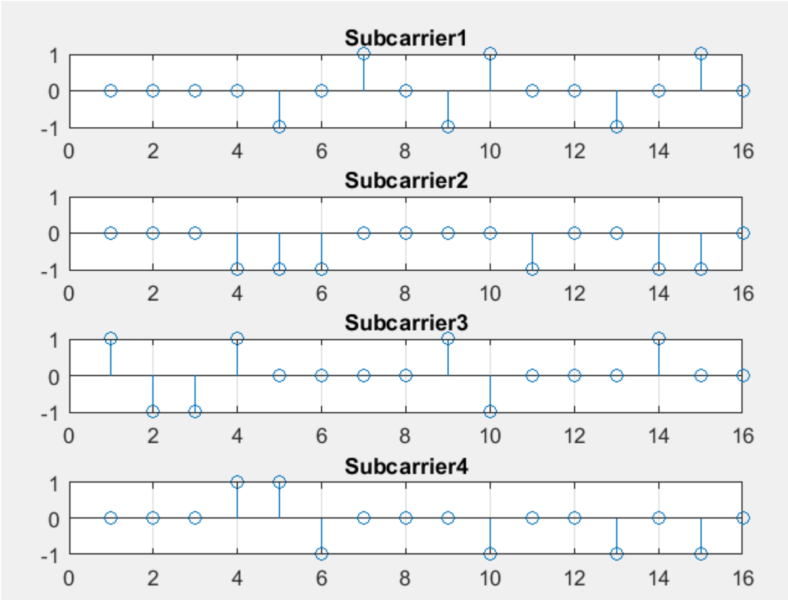


Figure 2.13. *Subcarriers*

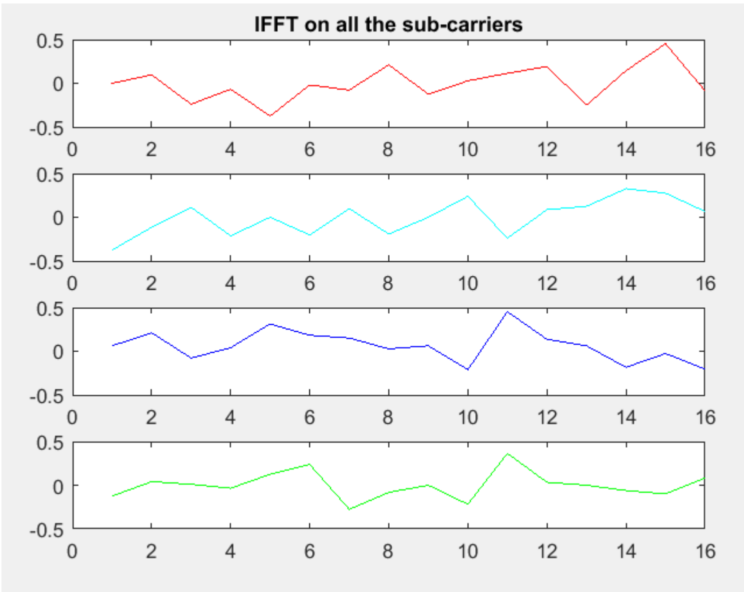


Figure 2.14. *Inverse Fourier transforms of the subcarriers*

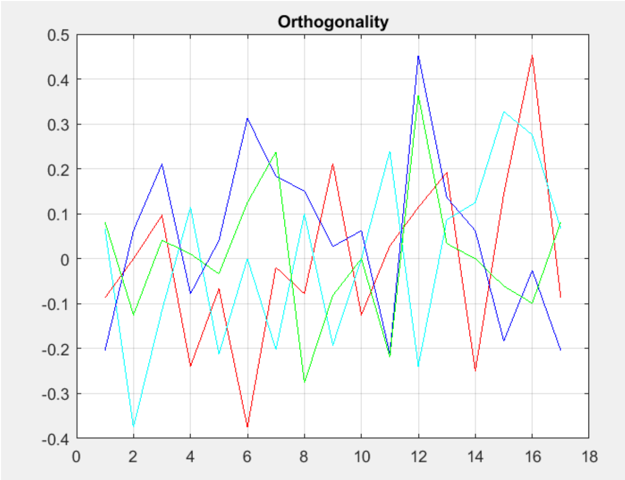


Figure 2.16. *Orthogonality*

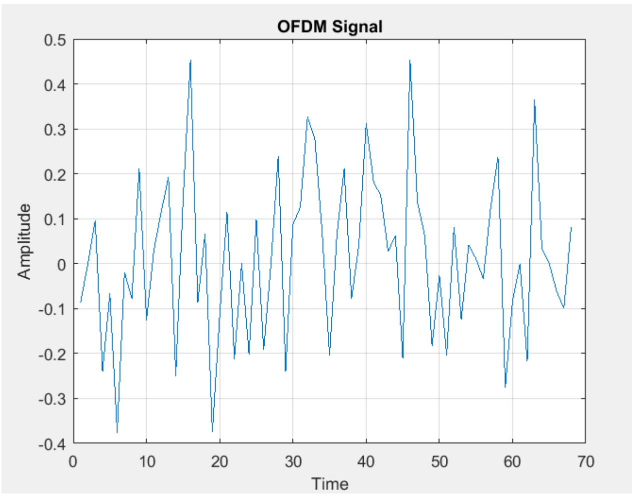


Figure 2.17. *The OFDM signal*

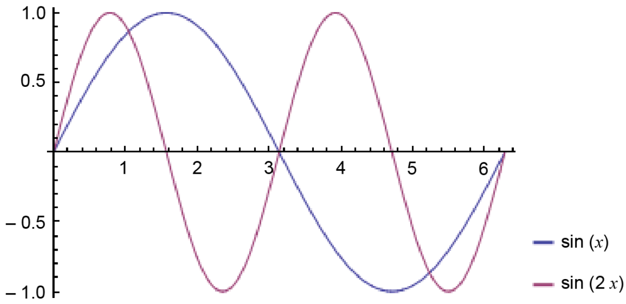


Figure 2.18. *Two sinusoidal curves of different periods; with a null algebric sum*

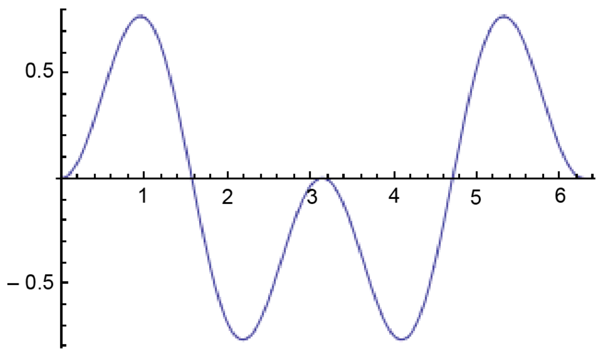


Figure 2.19. *Null mean value: the positive surface equals the negative one*

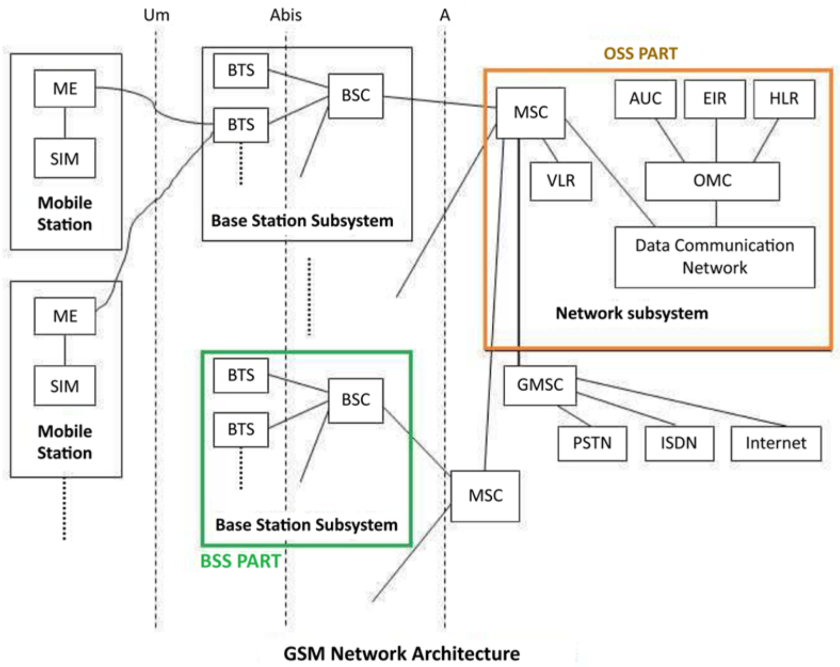


Figure 2.20. *GSM network architecture*

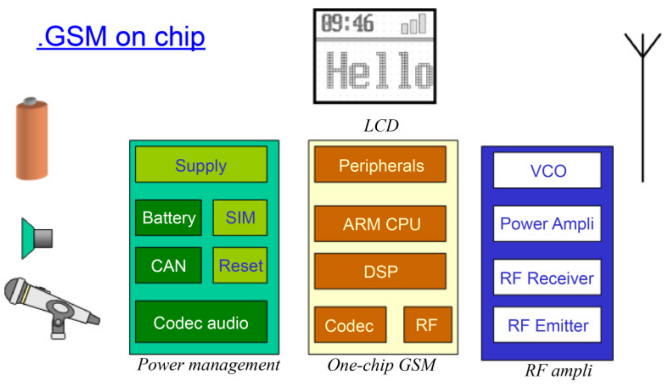


Figure 2.21. *Composition of a GSM*

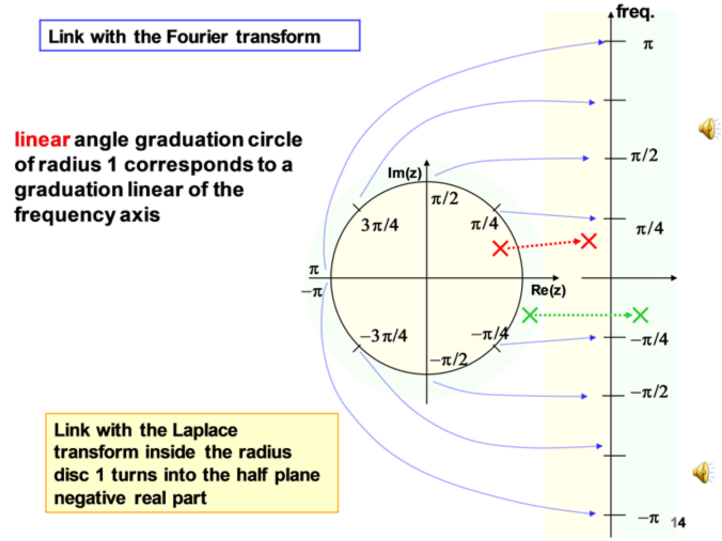


Figure 3.4. *Links between z-transforms, Fourier z-transforms and Laplace z-transforms*

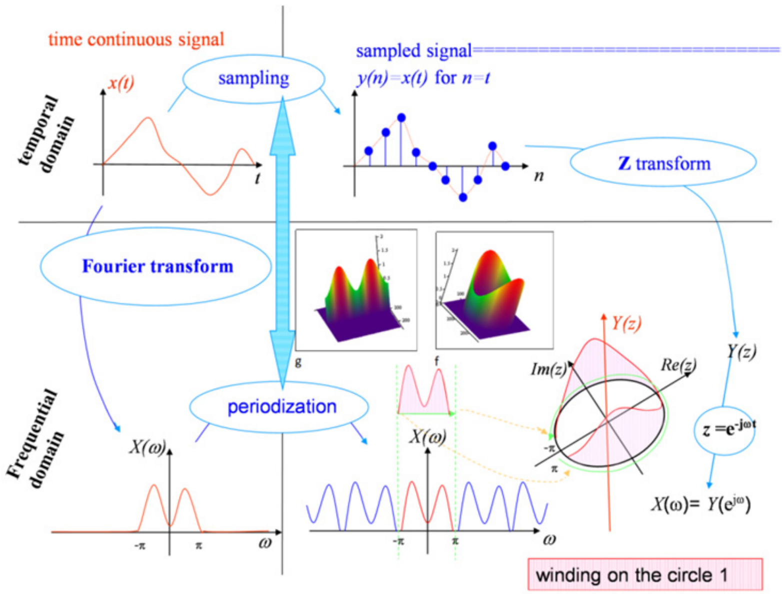


Figure 3.5. *Continuous/discrete; temporal/frequency*

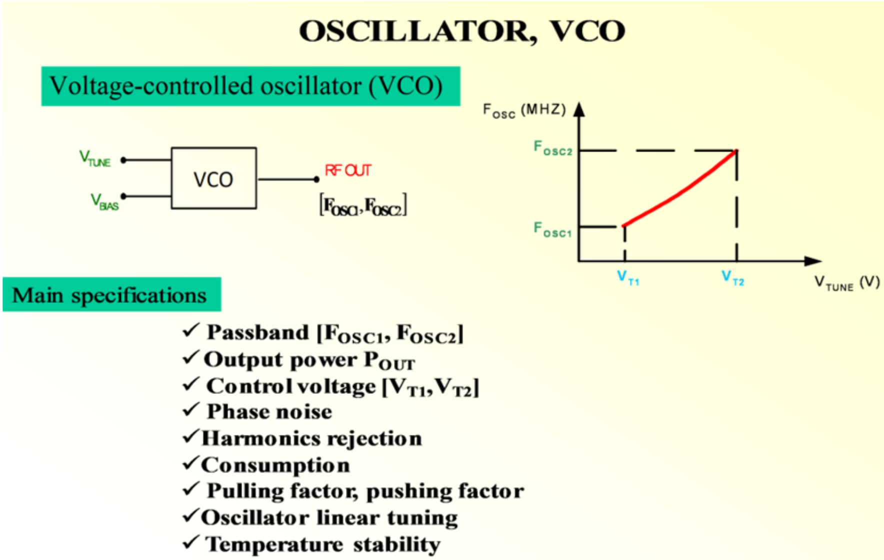


Figure 4.1. *Voltage-controlled oscillator*

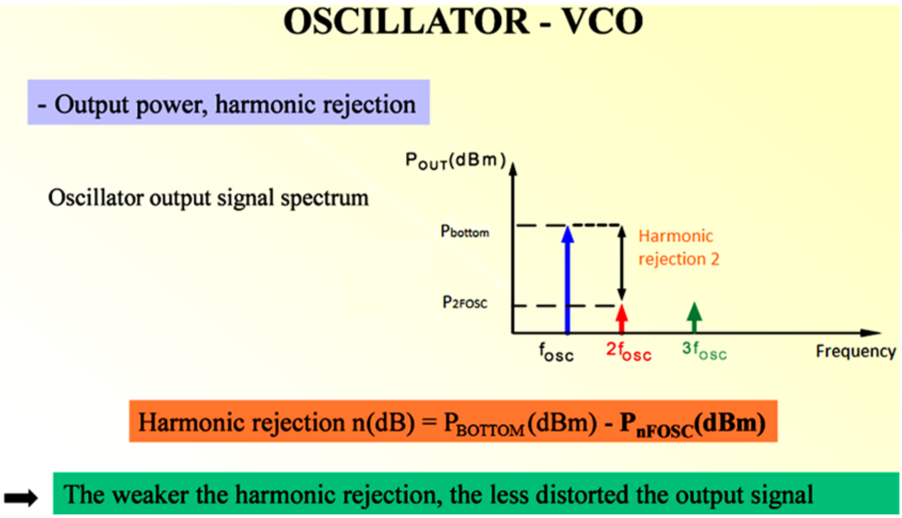


Figure 4.2. *Power: harmonic rejection*

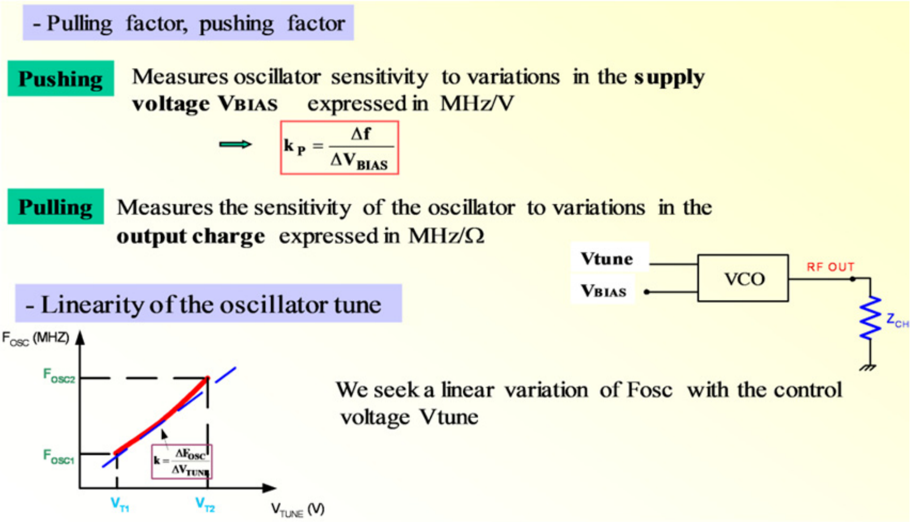


Figure 4.3. *Pushing/pulling of a VCO*

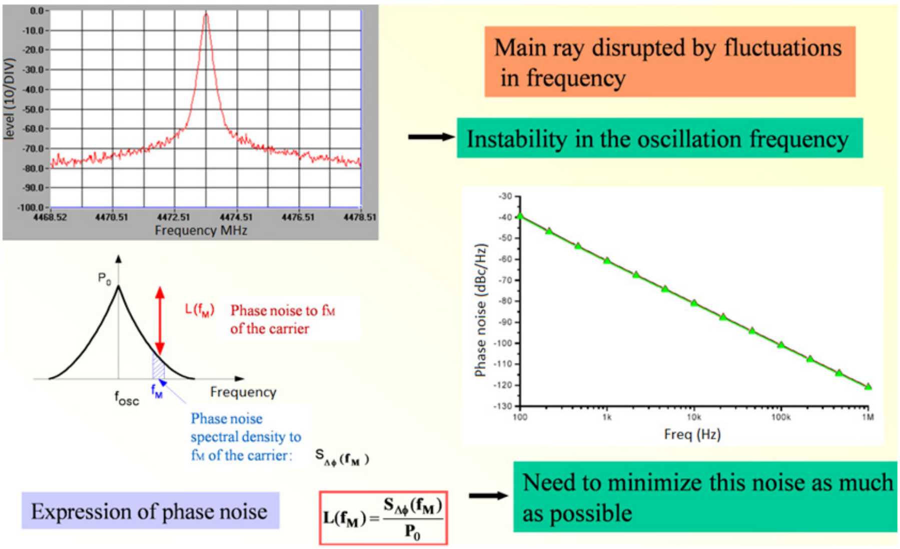


Figure 4.4. *Phase noise*

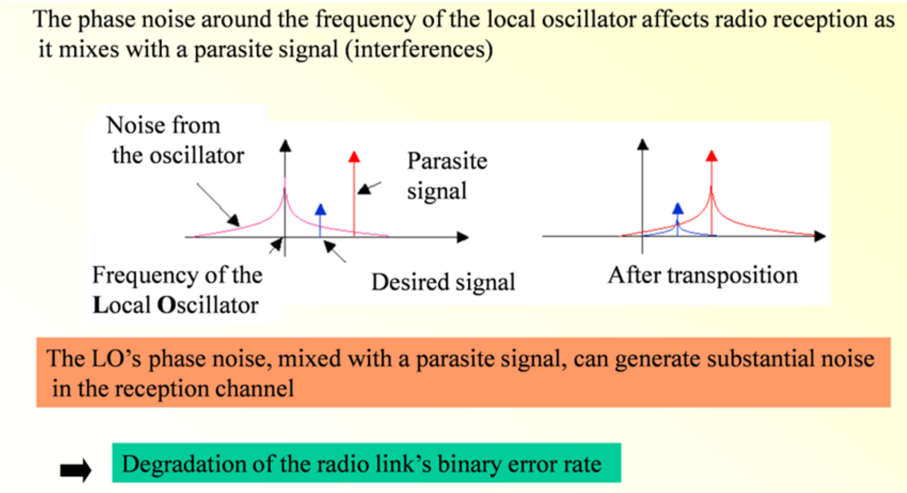


Figure 4.5. *Phase noise caused by parasite signals. We note that LO stands for local oscillator*

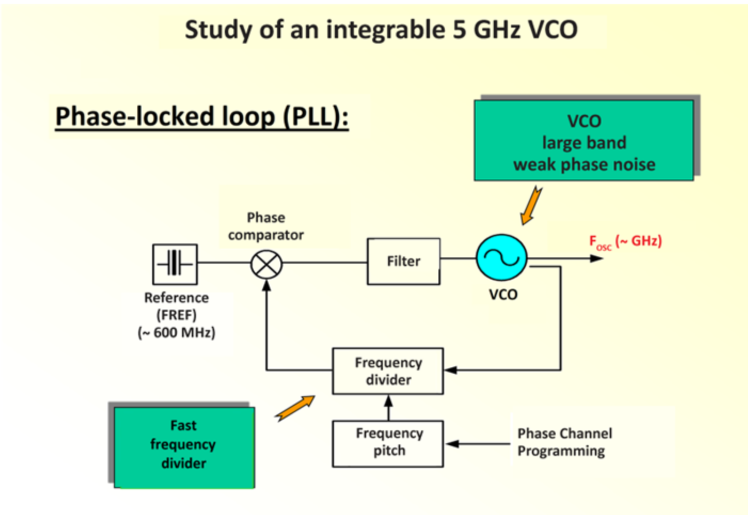


Figure 4.6. *Block schema of a phase-locked loop*

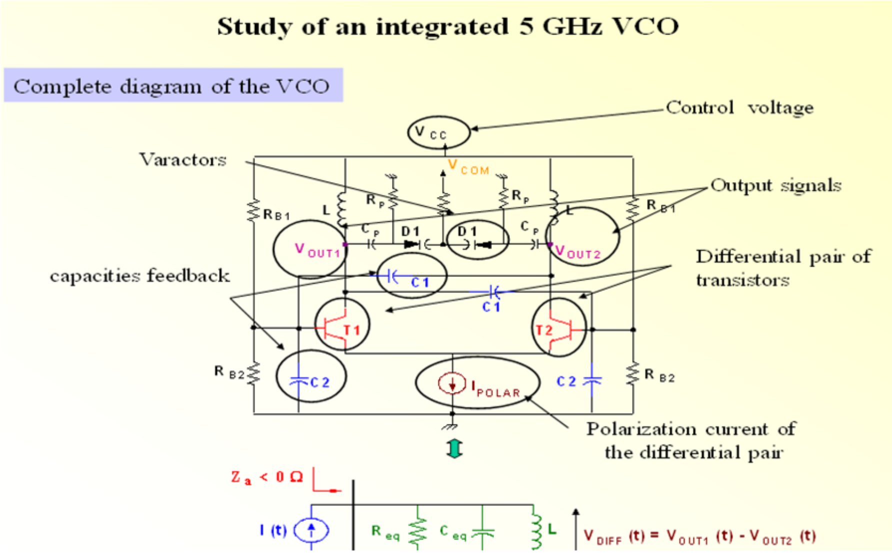


Figure 4.7. *Diagram circuit of a VCO*

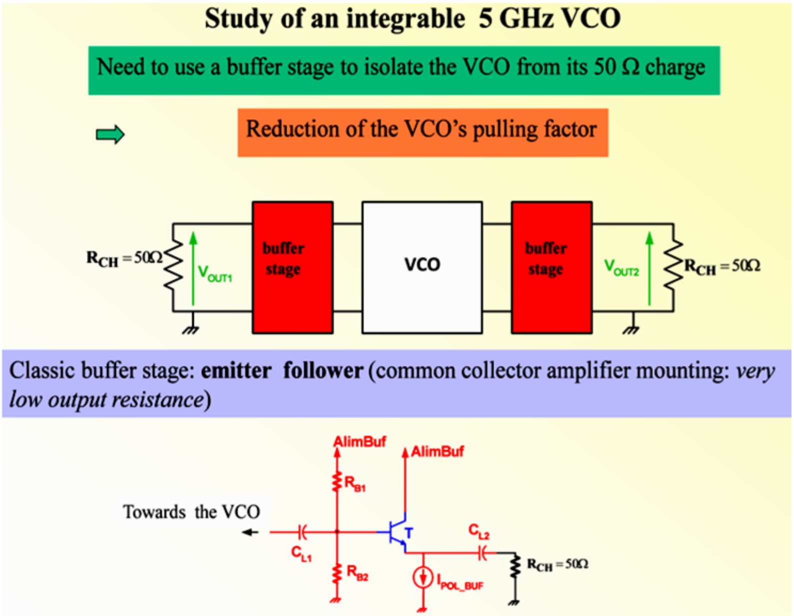


Figure 4.8. *Adaptation at output*

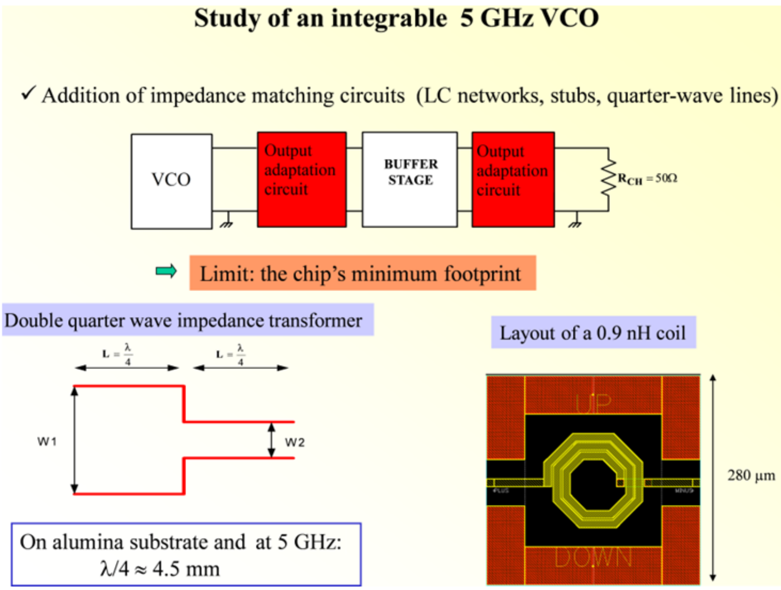


Figure 4.9. *Adaptation and layout of an antenna*

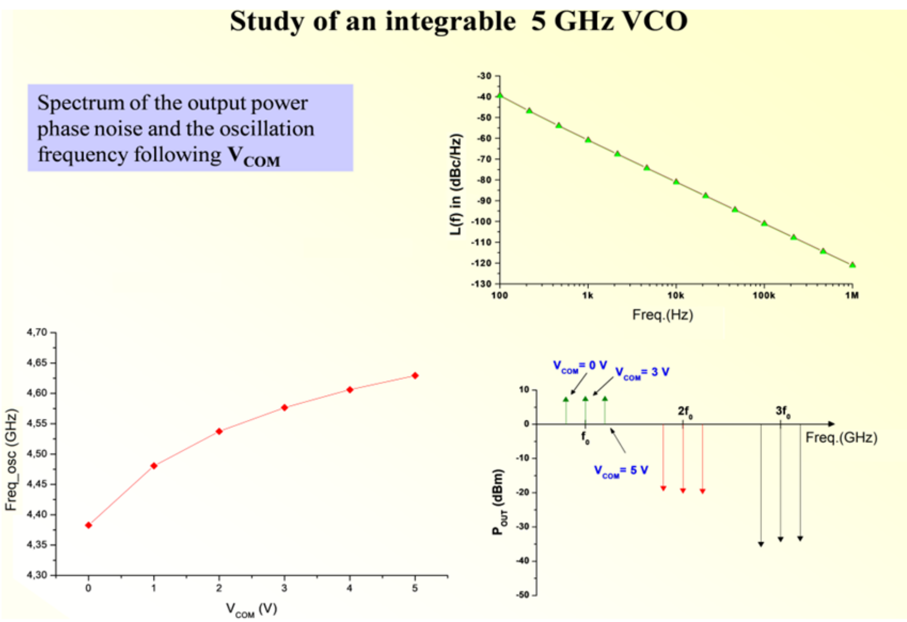


Figure 4.10. *Important parameters of a VCO*

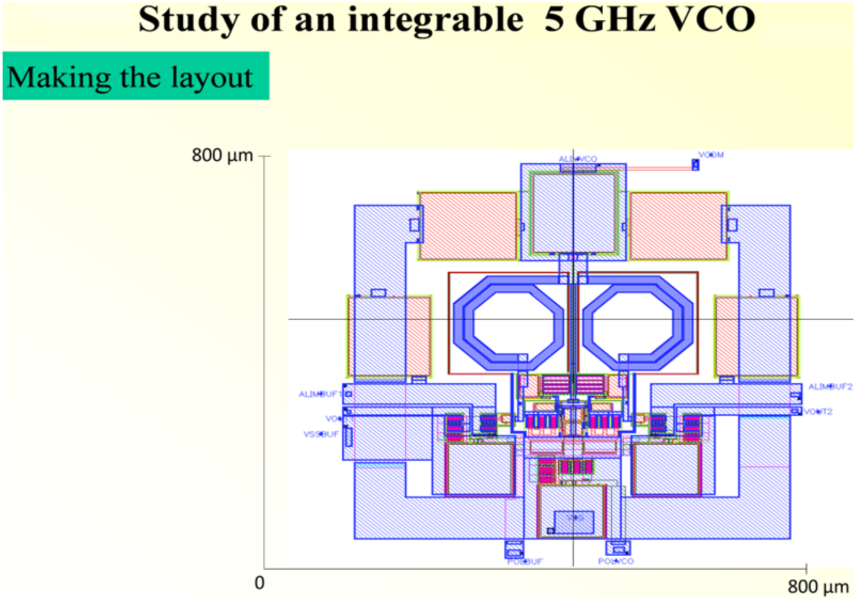


Figure 4.11. *Sketch of layout*

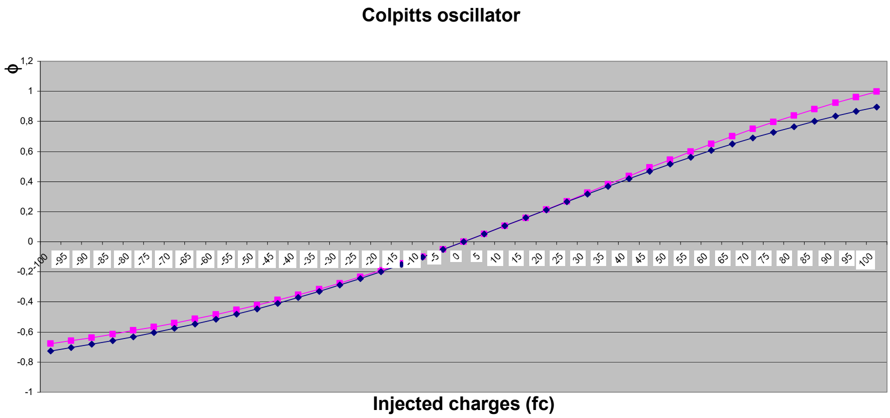


Figure 4.14. *Phase shift versus injected charges – between collector (L) and ground (see Figure 4.12) – for oscillator in Figure 4.12. Mixed mode: squares; "arctan" fit: diamonds (lower curve)*

Example of a phase detector(PD)

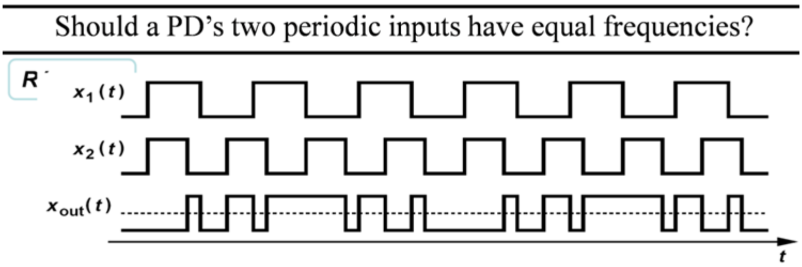
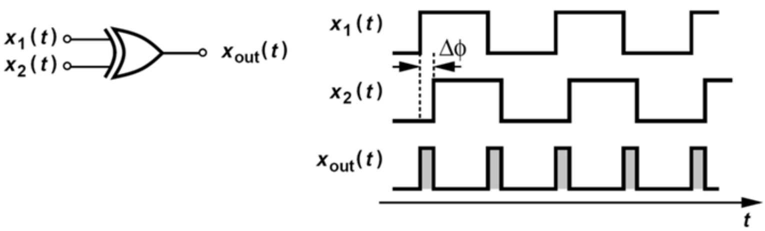


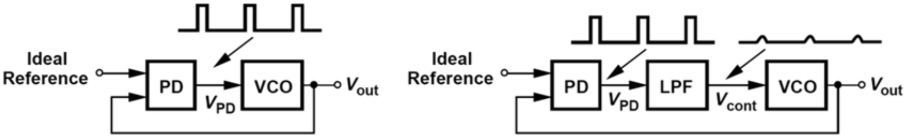
Figure 4.19. *Phase difference versus frequency*

How is the PD implemented?



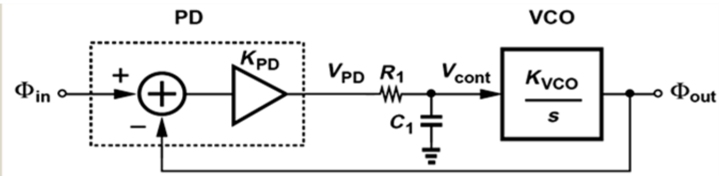
- We seek a circuit whose average output is proportional to the input phase difference.
- An exclusive-or 'XOR' window can serve this purpose, generating impulses whose breadth is equal to Δφ

Figure 4.20. *Phase difference via an exclusive-or*



- Negative feedback loop: if the 'loop gain' is high enough, the circuit minimizes the sampling error.
- Interpose a low-pass filter between the PD and the VCO to remove these impulses.

Figure 4.21. *Simple PLL and filter loop. Note that the negative feedback loop should force the phase error to zero, in which case the PD generates no impulse and the VCO is not disrupted. Thus, the low-pass filter would not be needed. In fact, this feedback system suffers a finite loop gain presenting a finite phase error in stationary state. Even the PLL has an infinite loop gain containing nonlinearities that disturb V_{cont}*

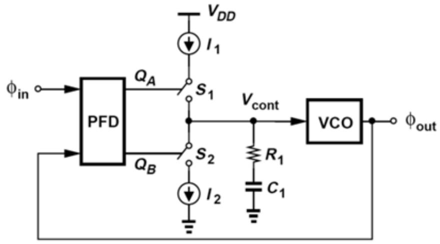


The transfer function in open loop: $[K_{PD}/(R_1C_1s+1)](K_{VCO}/s)$

transfer function in closed loop:

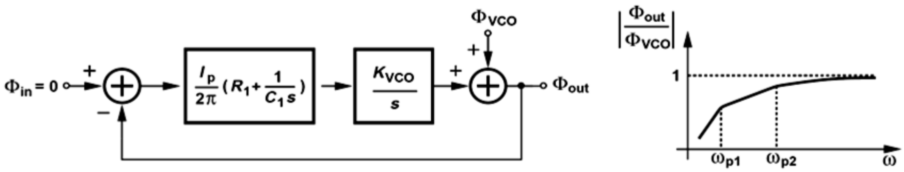
$$H(s) = \frac{\phi_{out}}{\phi_{in}}(s) = \frac{K_{PD}K_{VCO}}{R_1C_1s^2 + s + K_{PD}K_{VCO}}.$$

Figure 4.24. *Loop dynamic: model of the phase domain*



➢ If one of the integrators has losses, the system can be stabilized. This can be accomplished by inserting a resistance in series.

Figure 4.25. *PLL: charge pump*



$$-\phi_{out}\left[\frac{I_p}{2\pi}\left(R_1 + \frac{1}{C_1 s}\right)\right]\cdot\frac{K_{VCO}}{s} + \phi_{VCO} = \phi_{out}$$

$$\frac{\phi_{out}}{\phi_{VCO}} = \frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

➢ The PLL cancels only the slow phase variations of the VCO.

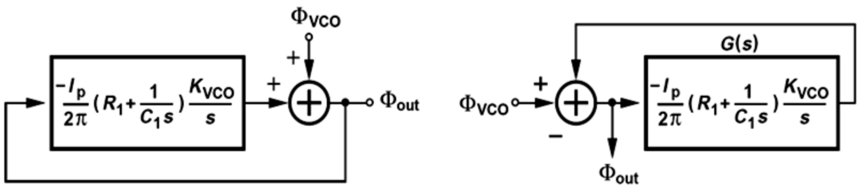


Figure 4.26. *Phase noise in PLLs: phase noise*

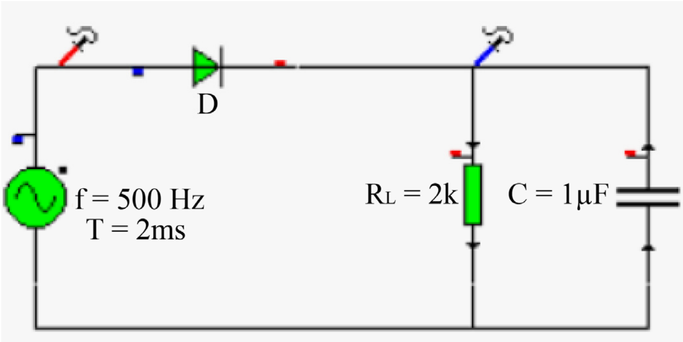


Figure A1.1. *Schematic of a filtered mono-alternance rectifier*

$$(V_{DC} + m(t))\cos(\omega_c t) * \cos(\omega_c t)$$

$$(V_{DC} + m(t))\cos^2(\omega_c t)$$

$$(V_{DC} + m(t))\left(\frac{1}{2} + \frac{1}{2}\cos(2\omega_c t)\right)$$

$$V_x = \frac{V_{DC}}{2} + \frac{m(t)}{2} + \frac{V_{DC}}{2}\cos(2\omega_c t) + \frac{m(t)}{2}\cos(2\omega_c t)$$

Figure A2.6. *Coherent detection*

$$s(t) = \begin{cases} A\cos(2\pi f_1 t) & \text{binary 1} \\ A\cos(2\pi f_2 t) & \text{binary 0} \end{cases}$$

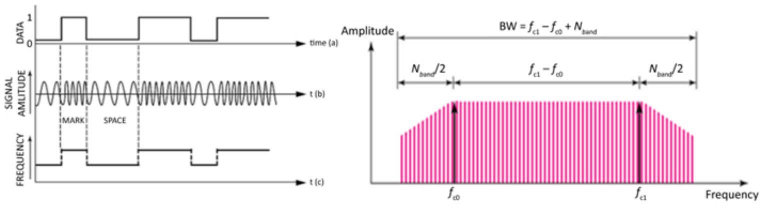


Figure A2.12. *FSK spectrum*

Carrier : $p(t) = A_p \cos 2\pi f_p t$ modulating: $m(t)$

$$\boxed{s(t) = \left[A_p + m(t)\right]\cos 2\pi f_p t}$$

Or:

$$s(t) = A_p\left(1 + \frac{a}{A_p} m_0(t)\right)\cos 2\pi f_p t \qquad \text{with } a = |\min(m(t))| \text{ and } m_0(t) = \frac{m(t)}{a}$$

EXPRESSION THAT BECOMES :

$$\boxed{s(t) = A_p\left(1 + m.m_0(t)\right)\cos 2\pi f_p t \qquad \text{with } m = \frac{a}{A_p} \qquad \text{index or modulation rate}}$$

Figure A2.15. *Linear amplitude modulations*

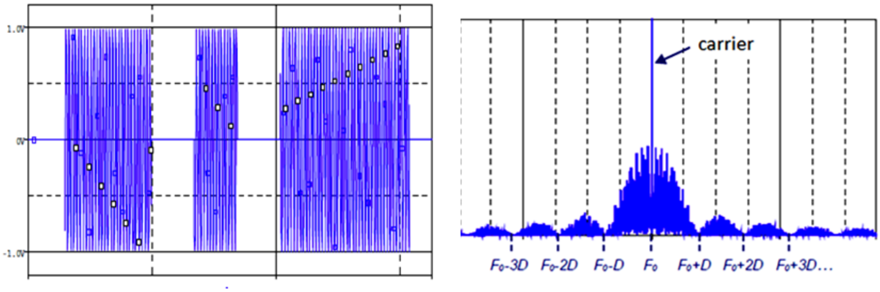


Figure A2.17. *ASK modulation, with its Fourier transform*

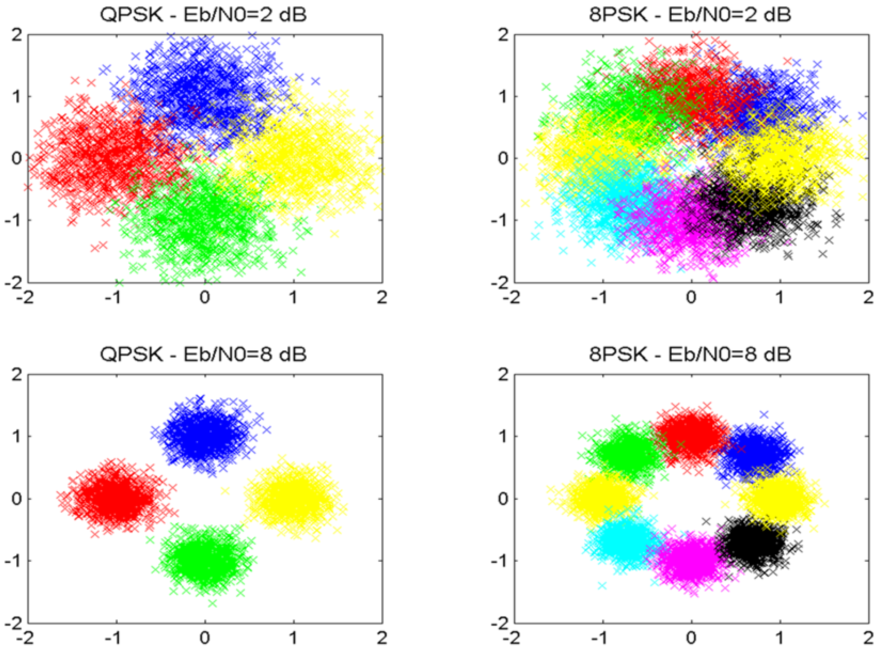


Figure A2.18. *Example of samples of output from a filter adapted for some examples of passband QPSK/8PSK modulation*

Signal in baseband: $$x_{Tp}(t) = T\sum_{l=-\infty}^{\infty}\left(d'(l) + jd''(l)\right) g_{Tx}(t-lT)$$

Signal in the carrier's band: $$\overline{E}_S = T^2\overline{|D|^2}\int_{-\infty}^{\infty} g_{Tx}^2(t)dt$$

$$x_{Bp}(t) = \sqrt{2}\,\mathrm{Re}\left\{x_{Tp}(t)e^{j2\pi f_0 t}\right\}$$

$$= \sqrt{2}T\left[\cos(2\pi f_0)\sum_{l=-\infty}^{\infty} d'(l)g_{Tx}(t-lT) - \sin(2\pi f_0)\sum_{l=-\infty}^{\infty} d''(l)g_{Tx}(t-lT)\right]$$

Average symbol energy:

$$\overline{E}_X = T^2\cdot 2\cdot\left[\frac{\overline{|D'|^2}}{2} + \frac{\overline{|D''|^2}}{2}\right]\cdot\int_{-\infty}^{\infty} g_{Tx}^2(t)dt = \overline{E}_S$$

Figure A2.19. *Analysis of a baseband system*

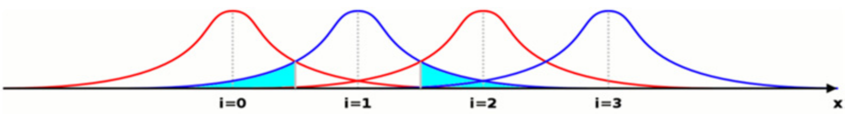


Figure A2.25. *Error rate per bit*

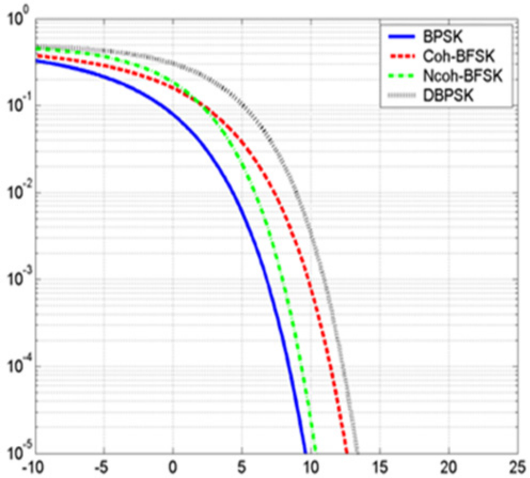


Figure A2.26. *Example of BER – binary modulations*

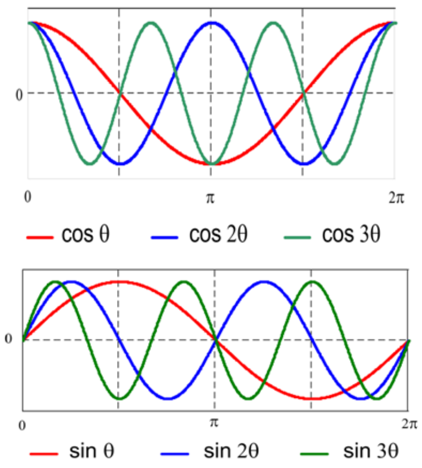


Figure A3.1. *Some sines and cosines representations*

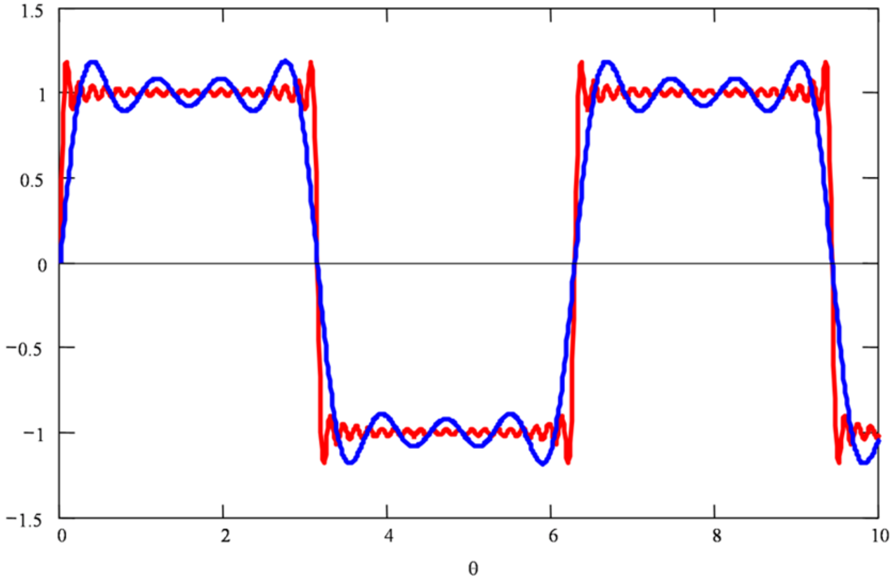


Figure A3.2. *Fourier analysis. The red curve is calculated with 20 terms and the blue with 4 terms*

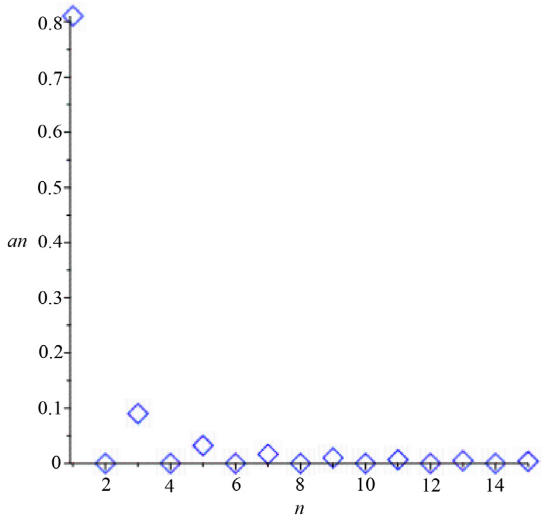


Figure A3.3. *an: function of n*

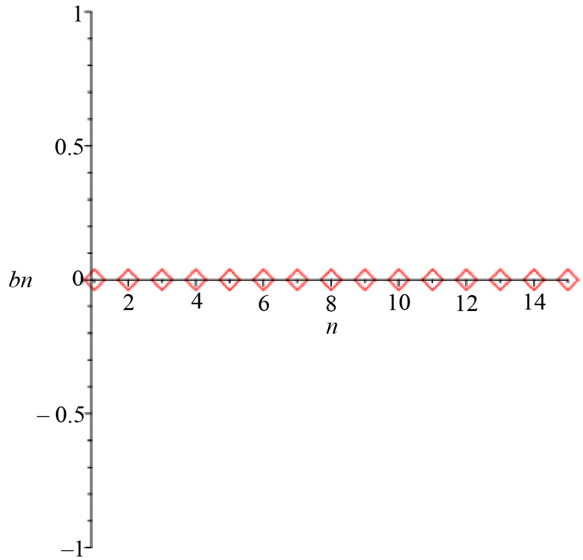


Figure A3.4. *bn: function of n*

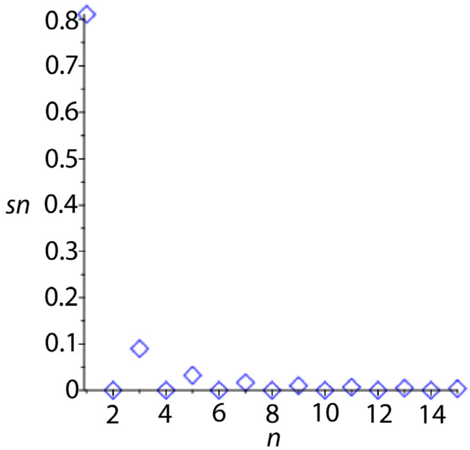


Figure A3.5. *sn: function of n*

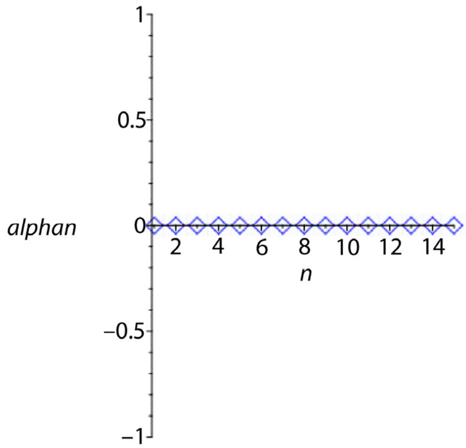


Figure A3.6. *alphan: function of n*

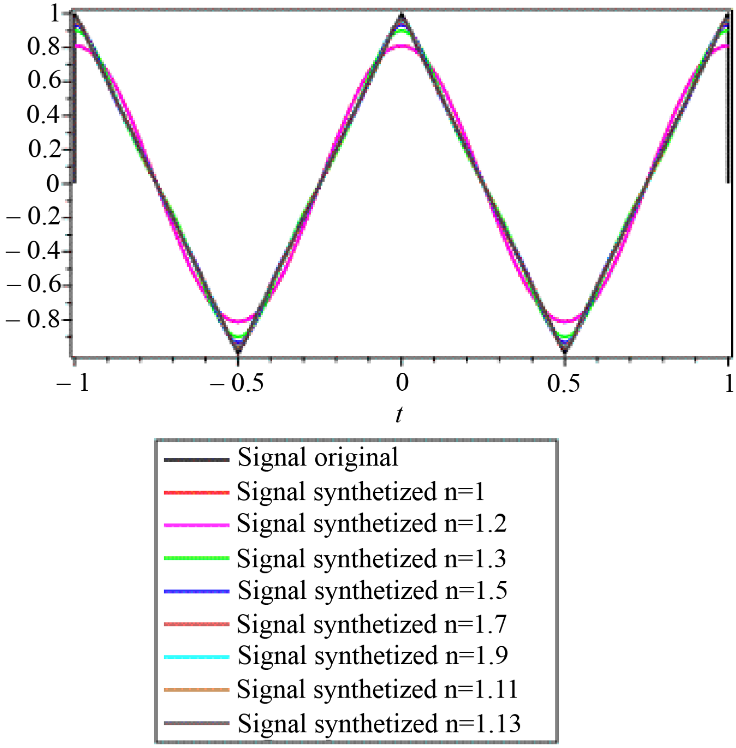


Figure A3.8. *Signal synthetized with different numbers of harmonics*