Institut National des Télécommunications et des Technologies de l'Information et de la Communication d'Oran

Module : Communications Numériques

TP : N°04

Digital Communication Through Band-Limited Channels

Préparé par : Mr Roumane Ahmed

Année universitaire : 2020/2021.

Estimation du temps nécessaire : 16h

Objective :

Réalisation d'une transmission numérique mono-porteuse en mettant l'accent sur : filtre de mise en forme, filtre adapté, détection cohérente, BER, ISI, Diagramme de l'œil, Synchronisation : Estimation et correction de l'offset en fréquence, récupération du rythme symbole.

Se préparer au TP :

Il est vraiment primordial de réviser les points suivants : ISI, Nyquist pulse, PRS, Equalization, Synchronization.

Voir l'annexes.

Travail demandé :

Remarque : mettez votre script principal dans un fichier « .m » et Commentez tous les ligne de code que vous introduisez. En cas de plusieurs fichiers, mettez-les dans un seul répertoire qui porte votre nom et l'intitulé du TP.

Organiser votre script en fonctions, dont chacune représente un bloc dans le schéma de bloc de la chaine de transmission. (en Matlab, il faut définir chaque fonction dans un fichier «Nom_de_fonction.m » séparé, puis vous mettez tous les fichier dans un seul dossier)

Assurez-vous que l'exécution des fichiers « .m » fonctionne comme prévue (surtout la clarté de l'affichage).

Question 01:

Lire attentivement/exécuter le script **'Pulse_shaping_SRRC.m'** et expliquer l'effet du paramètre **'D'** sur le spectre et la durée du signal bande base. Exprimer l'instant d'apparition du premier symbole en fonction du D et L.

Changer le paramètre « alpha » et observer son effet sur le spectre. Conclure.

Comparer le spectre du signal pour D=5 avec celui généré par une impulsion de mise en forme rectangulaire.

Question 02: Lire attentivement/exécuter le script « **LPF_design. m».**

Choisir les bons paramètres pour avoir un filtre « h_demo» ayant un spectre cohérent avec le signal bande base.

Question 03: Lire attentivement/exécuter le script « QAM_mixer_LPF.m » "voir l'annexe"

Comparer les bandes passantes des signaux bande-base et passe-bande.

Exprimer l'instant d'apparition du premier symbole sur le signal v en fonction du D, L et lh.

Question 04: Mise en évidence du critère de Nyquist :

Lire attentivement/exécuter les scripts «**ISI_channel_Rect.m**" puis "**ISI_channel_Raise_cos.m**"

Faite décrémenter (par pas de 2) la valeur de L jusqu'à 2, calculer le débit symbole (R=Fs/L) et observer son influence sur le signal émis ainsi que le signal reçu. (Bande passante et distorsion du signal)

Observer le diagramme de l'œil et le graph «`transmitted symbols VS RX MF output`»

Conclure.

Question 05: Lire attentivement/exécuter le script « **Time_syn_CMA.m** »

Que fait ce script ?

Question 06: Lire attentivement le script « **bandpass_ISI_Syn_telmplate.m** »

Lire attentivement les commentaires et compléter le script.

Question 07:

Compléter les script « char2qam.m » et « qam2char.m » en utilisant ce tableau.

Input bits MSB, LSB	Symbol
11	$(1+j)/\sqrt{2}$
01	$(-1+j)/\sqrt{2}$
00	$(-1-j)/\sqrt{2}$
10	$(1-j)/\sqrt{2}$

.

Challenge 08: (Optionnel)

Modifier le script « **bandpass_ISI_Syn_telmplate.m** » en remplaçant les lignes adéquates par des fonctions, qu'on doit les créer si elles n'existent pas :

char2qam.m, qam2char.m, isi_flat_channel.m, min_dist_dect.m

S'inspirer de "do_PLL_V2.m" et "do_sym_sync_earlylate_V2.m" pour implementer la synchronization PLL.

Introduire une variation temporelle dans l'oscillateur du récepteur et analyser la capacité du système à se synchroniser.

Explorer le fonctionnement du scripts ffe_zf.m (le corriger si nécessaire), utiliser cette fonction pour égaliser le signal reçu dans le script principal.

Question 09:

Rédiger une conclusion générale qui résume ce que vous avez appris dans ce travail

QAM Demodulation

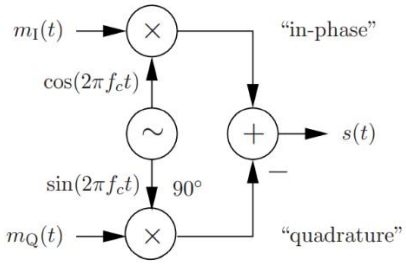
Let : $m_I(t) = g(t) * \text{barker\_up\_sampled}$

$m_Q(t) = g(t) * fliplr(\text{barker\_up\_sampled})$

With barker=[1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1]

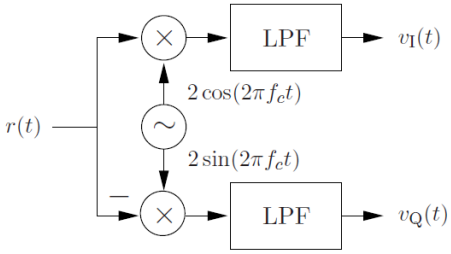
For visualization, 8 samples per symbol is the minimum accepted value. (We chose L=10)

The two real-valued messages are modulated simultaneously :



$$s(t) = m_I(t)\cos(2\pi f_c t) - m_Q(t)\sin(2\pi f_c t)$$

Ideal **QAM demodulation** :



Low-pass filtering removes the double frequency terms, cos(4πfct) and sin(4πfct).

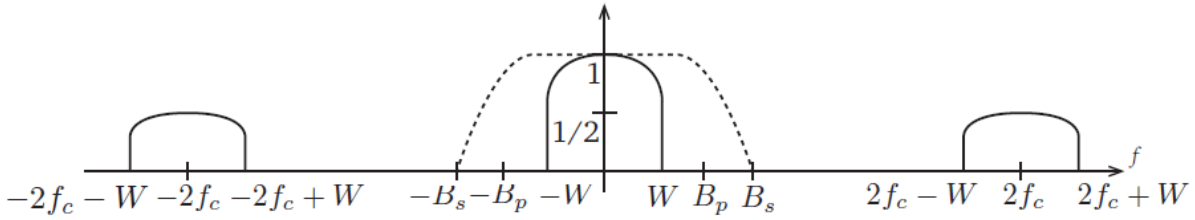
For the **coherent** receiver: we require that the receiver oscillator is perfectly synchronized (in frequency and phase) to the transmitter oscillator.

complex-baseband notation:

$$\tilde{m}(t) = m_I(t) + jm_Q(t)$$
$$\tilde{v}(t) = v_I(t) + jv_Q(t)$$
$$S(t) = Re\{\tilde{m}(t)e^{j2\pi f_c t}\}$$

The demodulation structure is: $\tilde{v}(t) = LPF\{s(t) \cdot 2e^{-j2\pi f_c t}\}$

Filtering the baseband signals {mI(t), mQ(t)} with one-sided bandwidth W Hz,



the lowpass filter (LPF) has passband edge frequency Bp ≥ W Hz and stopband edge frequency Bs ≤ (2fc − W) Hz.

Recal that the order of a filter is = (the length the impulse response – 1).

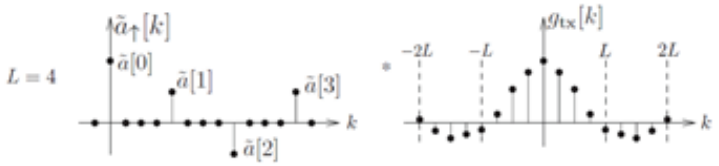
Symbol detection

The nearest neighbor rule, or minimum distance detector.

Pulse Shaping and Matched Filtering

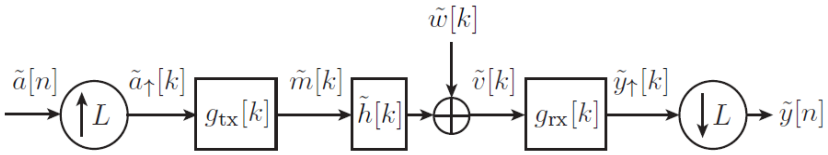
Symbols are converted to a sampled-data baseband message;

For visualization, L ≥ 8 samples per symbol is preferred.



$$\tilde{m}(k) = \sum_n \tilde{a}[n] g_{tx}(kT_s - nT)$$

T : symbol period, and $T_s = 1/\text{fs}$ is the DAC sampling interval.
The summation is a convolution of the upsampled symbol sequence, $\tilde{a} \uparrow [n]$, and the sampled pulse $g_{tx}(k)$. The upsampling operation inserts L−1 zeros between consecutive samples of $\tilde{a}[n]$, with L = T /Ts, in order to match the baseband sampling interval, Ts < T.
At the receiver, ideal performance would result in the n$^{\text{th}}$ output $\tilde{y}[n]$ equal to the n$^{\text{th}}$ input symbol $\tilde{a}[n]$.
Filter with impulse response $q[k]$ is needed to: prevent ISI and suppress noise.

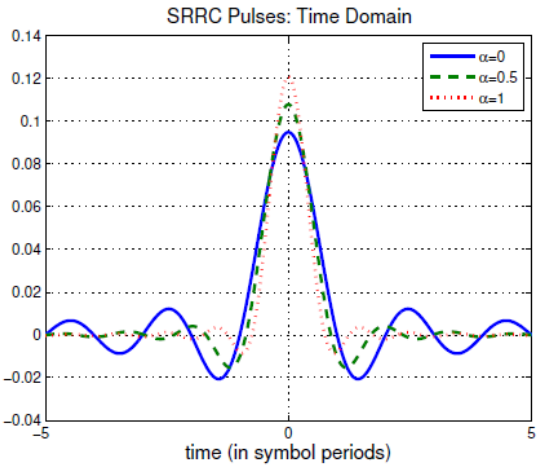


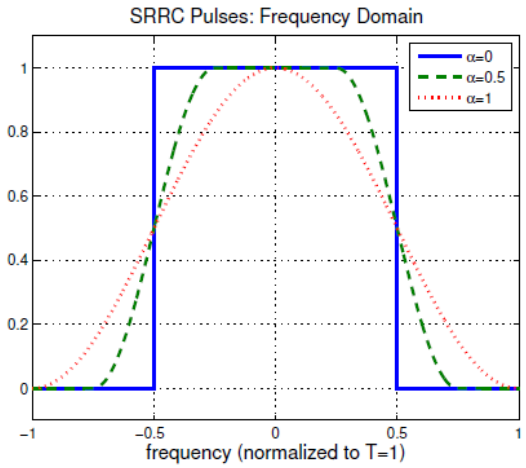
To maximize the signal to noise ratio at $\tilde{y}(n)$, $g_{rx}(\text{t})$ must be **matched** to $g_{tx}(\text{t})$:
$g_{rx}(t) = g_{tx}(-t)^*$ and $G_{rx}(f) = G_{tx}(f)^*$

for SNR-maximizing ISI-free performance, the filter design requires: $\text{p(t)} = g_{tx}(\text{t}) * g_{rx}(\text{t})$ is a Nyquist pulse and
$g_{rx}(t) = g_{tx}(-t)^*$ = common design choice is the square − root raised cosine (SRRC) pulse.

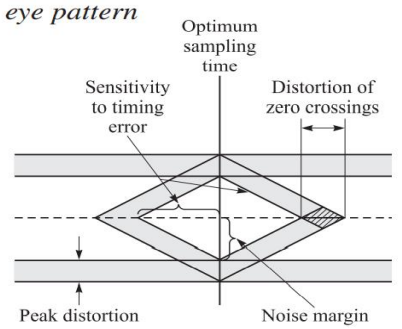
$$g_{\text{SRRC}}(t) = \begin{cases} \frac{1}{\sqrt{T}} \frac{\sin(\pi(1-\alpha)t/T) + (4\alpha t/T)\cos(\pi(1+\alpha)t/T)}{(\pi t/T)(1-(4\alpha t/T)^2)} & t \neq 0, \pm\frac{T}{4\alpha} \\ \frac{1}{\sqrt{T}}(1 - \alpha + 4\alpha/\pi) & t = 0 \\ \frac{\alpha}{\sqrt{2T}}\left\{\left(1+\frac{2}{\pi}\right)\sin\left(\frac{\pi}{4\alpha}\right) + \left(1-\frac{2}{\pi}\right)\cos\left(\frac{\pi}{4\alpha}\right)\right\} & t = \pm\frac{T}{4\alpha}. \end{cases} \quad (4.6)$$

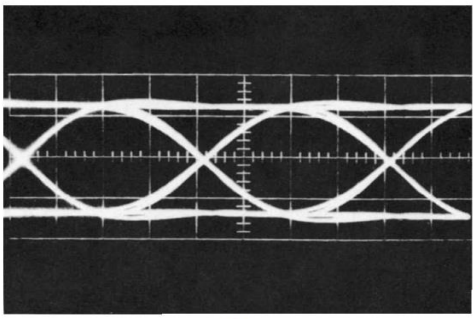
Note that, for $\alpha > 0$, the square-root raised cosine pulse is not a Nyquist pulse; however, the convolution of $g_{\text{SRRC}}(\text{t}) * g_{\text{SRRC}}(\text{t})$ produce a Nyquist pulse.



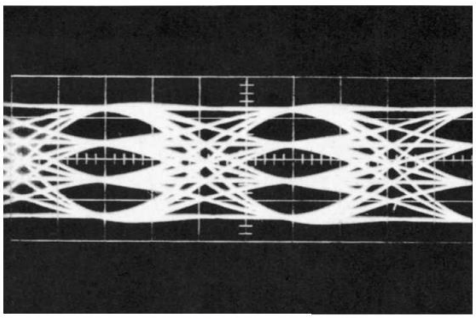


Eye diagram (PAM)





(a) Binary



(b) Quaternary

If the symbols are complex-valued, then eye diagrams may be plotted for both the I and Q channels separately.

Synchronization

Symbol timing will be estimated for the downsampling of the matched filter outputs.

Let "L" the number of samples per symbol.

Here we describe a block-processing version of the constant modulus algorithm (CMA). That is to select the sampling instant such that the downsampled subsequence has the smallest variance.

Let γ_p the mean of the absolute values of matched filter outputs with offset p samples and subsampled to the symbol rate:

$$\gamma_p = mean\{\left|\widetilde{y\uparrow}\,[\mathrm{p} + \mathrm{nL}]\right|, \mathrm{n} = 1, 2, \ldots, [\mathrm{N/L}] - 1\} \quad (5.1)$$

Then, minimization of the variance by selection of p can be written:

$$\hat{p} = \min_{\mathrm{p}\in\{0,\ldots,\mathrm{L}-1\}} \sum_{n=0}^{\lceil \frac{\mathrm{N}}{\mathrm{L}} \rceil - 1} \left(\left|\widetilde{y\uparrow}\,[\mathrm{p} + \mathrm{nL}]\right| - \gamma_p\right)^2$$

Frame timing for flat channels

Frame synchronization is performed using known training sequence of symbols ("marker sequence" or "pilots").

In a flat fading channel model, $\tilde{h}(t) = A\delta(t - \tau)$, the transmitted signal, s(t), experiences a gain, a phase, and a delay.

$$\tilde{v}(t) = \tilde{H} \cdot \tilde{m}(t - \tau) + \tilde{w}(t)$$

$\tilde{H}$ is the gain and phase as a complex scalar.

For the AWGN model, the maximum likelihood (ML) estimate, $\hat{\tau}$, of the delay is computed using the absolute value of the matched filter output:

$$\begin{aligned} \tilde{y}(t) &= \int \tilde{v}(\lambda)\tilde{m}^*(\lambda - t)d\lambda \\ \hat{\tau} &= \arg\max_t \ |\tilde{y}(t)|. \end{aligned} \tag{5.4}$$

The matched filter output also gives the ML estimate of the unknown amplitude,

$$\tilde{H}_{est} = \frac{\tilde{y}(\hat{\tau})}{\int |\tilde{m}(\lambda)|^2 d\lambda}. \tag{5.5}$$

That is, $\tilde{H}_{est}$ is the ratio of the observed peak value to the known noiseless autocorrelation peak from the training signal $\tilde{m}(t)$.

For sampled data and ±1 marker symbols, Equation 5.5 tells us :

H_est = peak/length(pilots) (5.6)

where peak is the complex-valued sample of maximum absolute value, $peak = |\tilde{y}(\hat{\tau})|$.

A good marker sequence must have a very sharp autocorrelation peak to provide noise robustness.

Barker code:

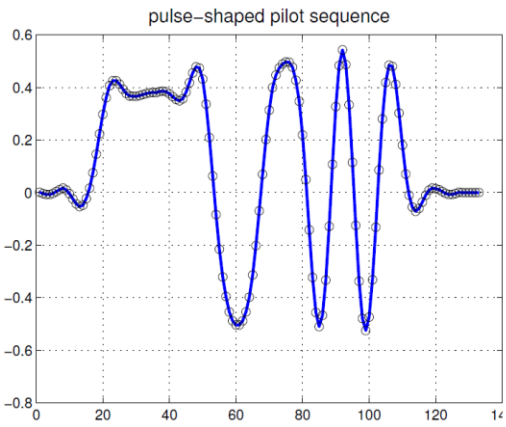
consider a Barker code. A length N_{tr} Barker code $\{a_k\}_{k=0}^{N_{tr}-1}$ is a sequence of binary symbol values, ±1, such that

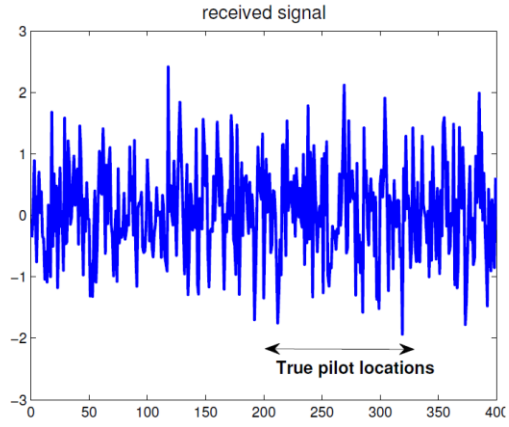
$$\left| \sum_{k=0}^{N_{tr}-i-1} a_k a_{i+k} \right| \leq 1 \quad \text{for } i \neq 0. \tag{5.7}$$

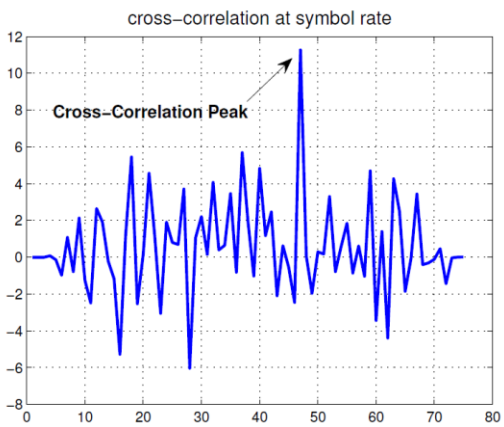
For example, a length-13 Barker code is

$$\{a_k\}_{k=0}^{12} = \{1,\ 1,\ 1,\ 1,\ 1,\ -1,\ -1,\ 1,\ 1,\ -1,\ 1,\ -1,\ 1\}. \tag{5.8}$$

Frame timing recovery send the Barker code in noise and the Rx computes the cross-correlation between the noisy signal and the known Barker code. From the result, the visible peak serves to identify the location of the marker sequence within the noisy signal.







var *computes the variance of a list of numbers*
fliplr *reverses the column ordering of an array*

Frequency Recovery

Consider a receiver oscillator given by $e^{j(2\pi(f_c+f_\Delta)t+\varphi)}$ with a frequency mismatch f_Δ.
The received message is: $\tilde{v}(\mathrm{t}) = \tilde{m}(\mathrm{t})\, e^{-j(2\pi f_\Delta t+\varphi)}$
Suppose $\tilde{a}[\mathrm{n}]$ is known; thus, after proper receiver filtering, symbol-timing recovery and frame-timing recovery, the received symbols suffer a time-dependent phase error:

$$\tilde{y}[\mathrm{n}] = \tilde{a}[\mathrm{n}]\, e^{-j(2\pi f_\Delta nT+\varphi)}$$

where T is the symbol period and the index n counts the sampled marker sequence from $n = 0$. the phase difference between known pilots and received symbols is a linear function of sample index,

$$\text{angle}\left\{\frac{\tilde{a}[n]}{\tilde{y}[n]}\right\} = (2\pi f_\Delta T)\, n,$$

Notice that the quantity (2πfΔT) gives the incremental phase error in radians per symbol and spec□ifies the slope of the linear function. Thus, a line fit to the un□wrapped phase angle of the known markers divided by the received symbols gives an estimate for the frequency recovery.
p = polyfit(x,y,d) finds the coeffi□cients of a polynomial p(x) of degree d that best fits the data, y(i), at sampling instants, x(i), in a least squares sense.