



Modelo genérico de um sistema de comunicação digital. Digitalização e compressão da informação para mínima ocupação espectral. *Forward error correction* para mínima taxa de erro de bit. Modulação e demodulação. *Design flow*.



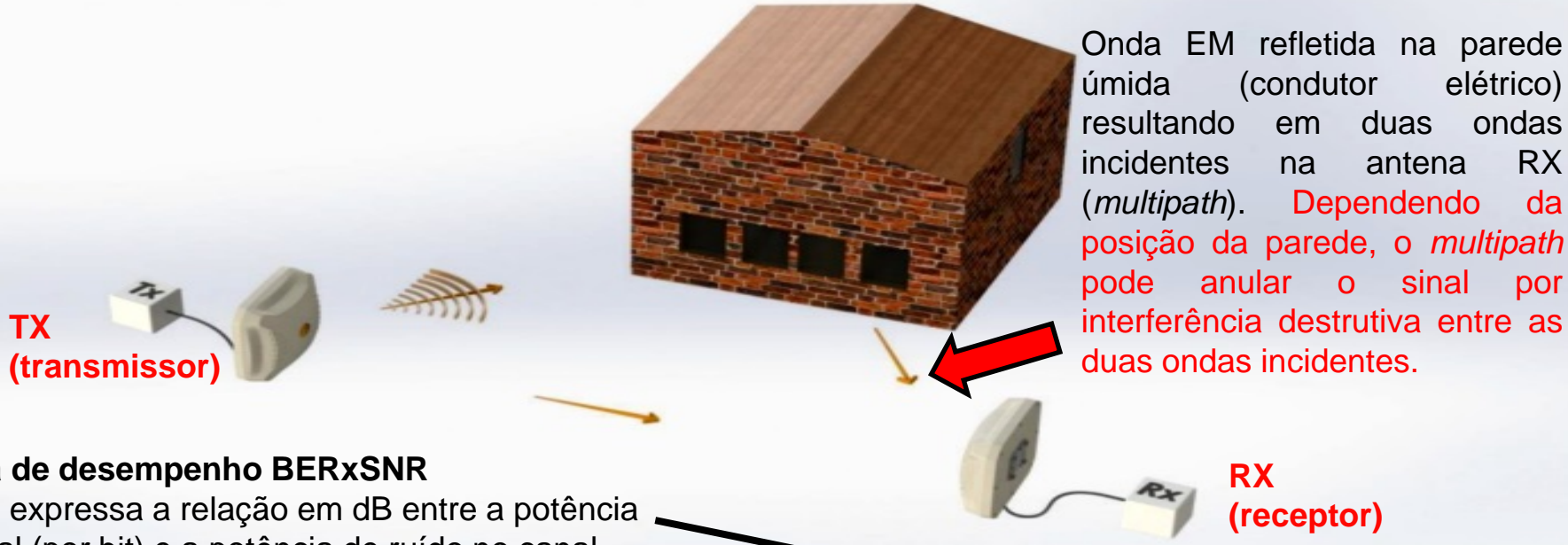
Departamento de Eletrônica e Computação

Centro de Tecnologia

ELC1120 – Telecomunicações II

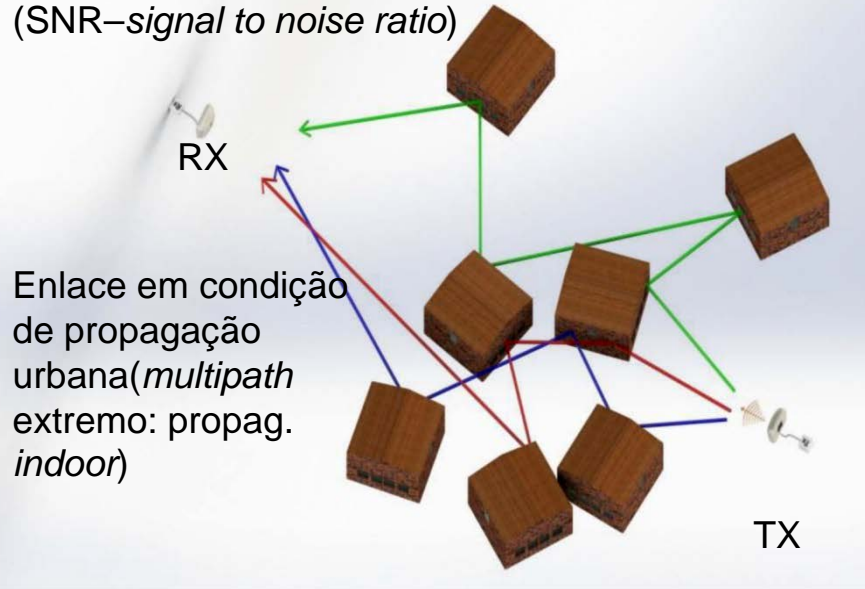
Profa. Candice Müller Prof. Fernando DeCastro

# Parâmetros de desempenho de um sistema de comunicação digital em função das condições operacionais no canal de comunicação (no caso-exemplo, um canal *wireless*):

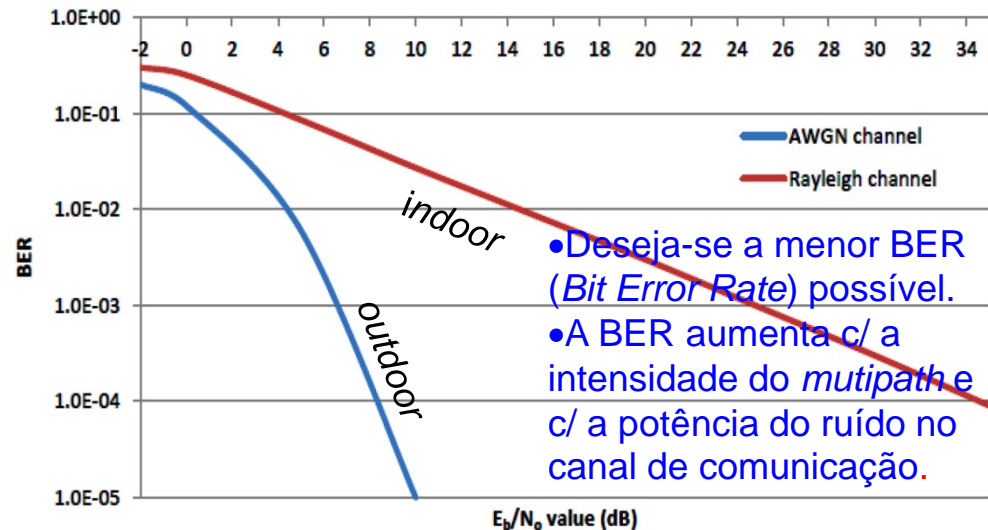


## Curva de desempenho BERxSNR

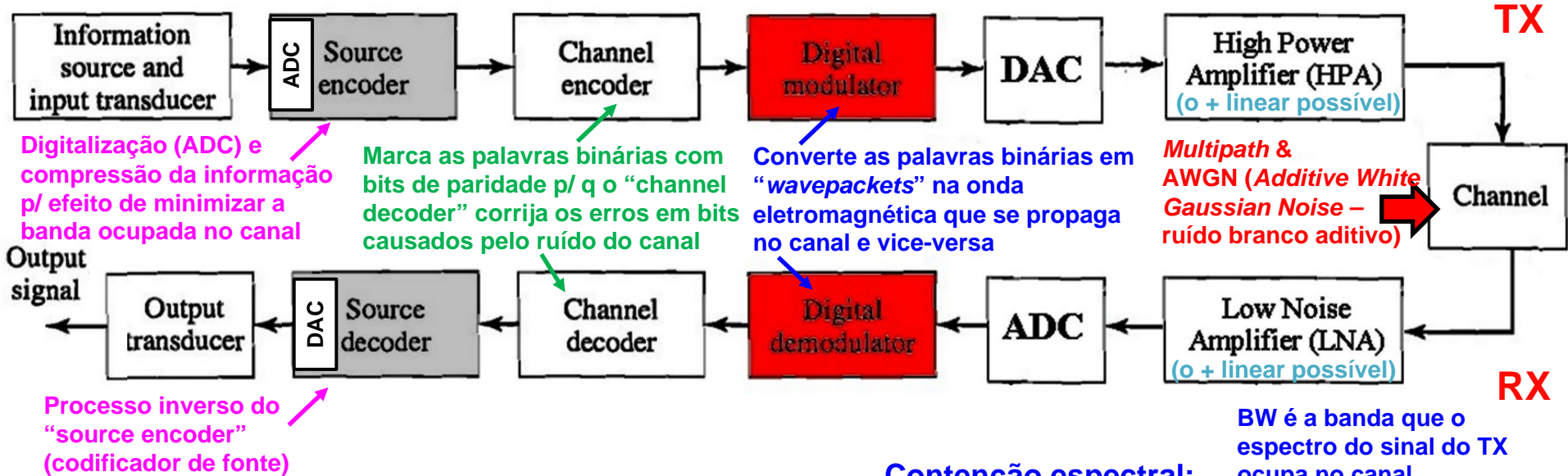
$E_b/N_0$  expressa a relação em dB entre a potência do sinal (por bit) e a potência de ruído no canal (SNR—*signal to noise ratio*)



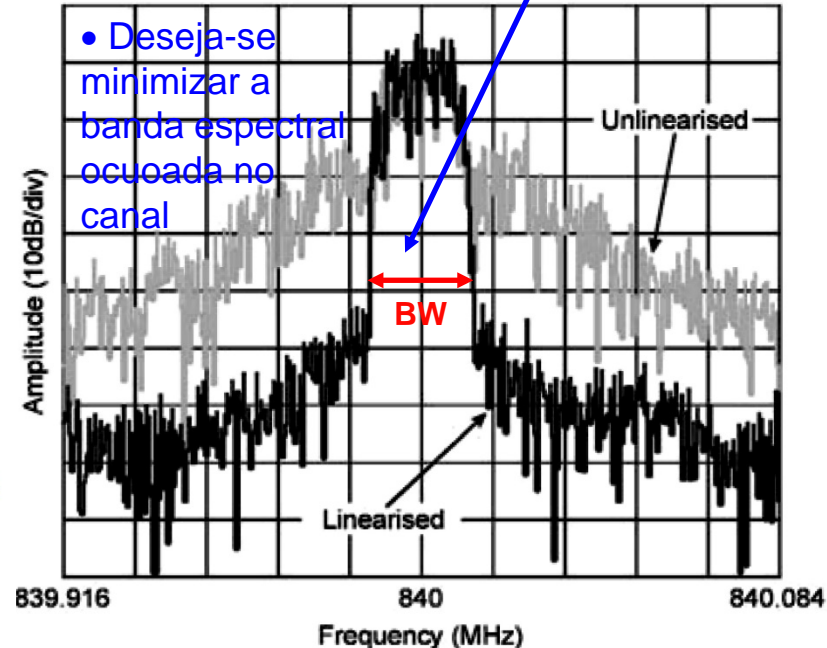
BER x  $E_b/N_0$  for BPSK



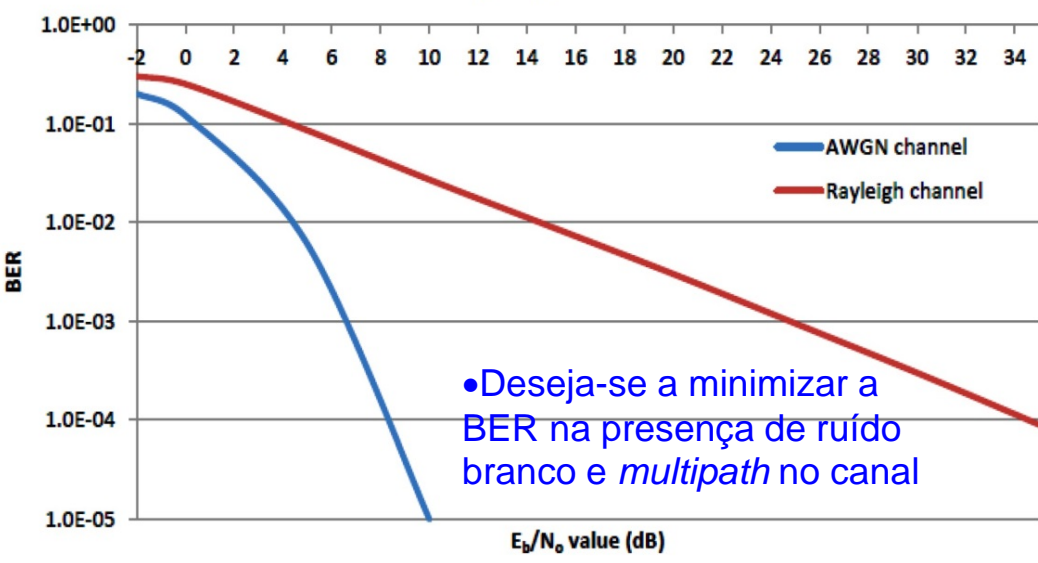
# Arquitetura genérica dos blocos funcionais de um sistema de comunicações p/ que os parâmetros de desempenho desejados para o sistema sejam alcançados:



## Contenção espectral:



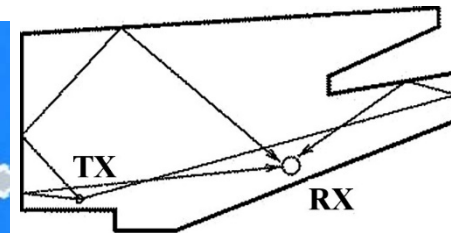
BER x  $E_b/N_0$  for BPSK





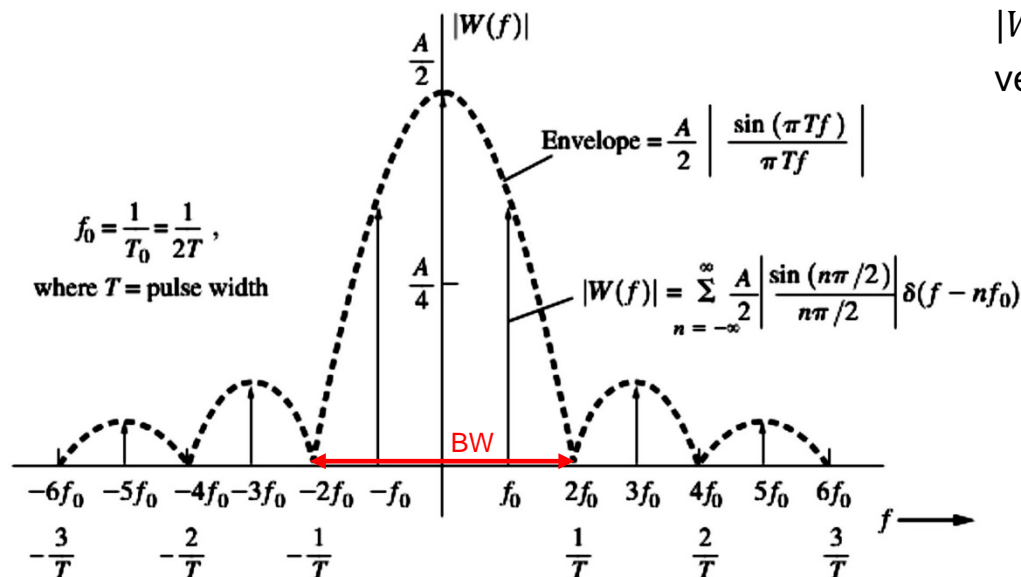
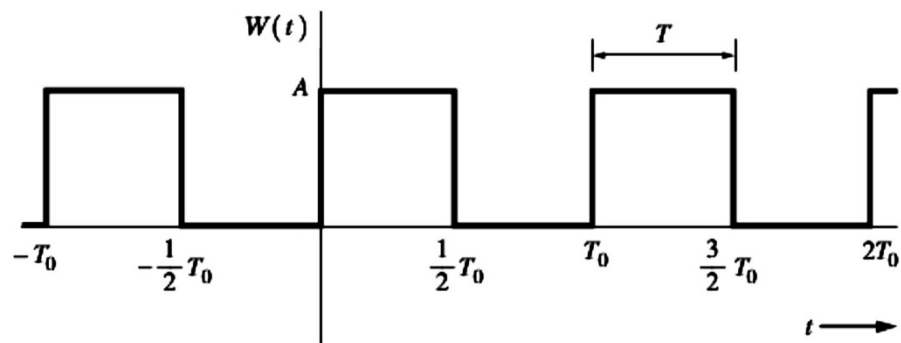
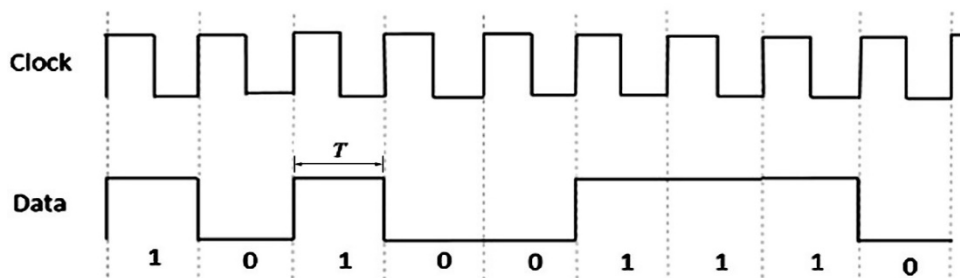
## Degradação da inteligibilidade do sinal recebido causada pelo *multipath* no canal de comunicação – analogia acústica:

Duas pessoas, uma em cada extremidade de um longo corredor tentando estabelecer conversação terão dificuldade na inteligibilidade das palavras (mesmo não havendo ruído) **caso a duração das palavras proferidas seja aproximadamente de mesma duração dos ecos nas paredes do corredor**. Isto ocorre porque cada palavra recebida será interferida pelo seu eco que se superpõe à palavra original tornando-a ininteligível. **Note que é inútil falar mais alto porque é o próprio sinal interferindo nele mesmo**. Este fenômeno é denominado de **Interferência Intersimbólica** (ISI – *Inter Symbol Interference*), onde, neste exemplo acústico cada símbolo na conversação é uma palavra proferida. Em um sistema de comunicação digital, um símbolo é um “*wavepacket*” na onda eletromagnética que se propaga no canal, e que é atribuído à respectiva palavra binária no modulador, conforme veremos adiante.



Uma possível solução é falar bem mais devagar, quase soletrando, de modo que a duração do eco interferente seja desprezível em relação a das palavras (OFDM).

## Compressão da informação no codificador de fonte (“source encoder”) p/ minimizar a banda BW ocupada no canal pelo sinal do TX:



Vamos supor que o *stream* de bits “Data” transporte informação que é transmitida a uma taxa (= *throughput*) de 1.0 Kbps. Então a duração  $T$  de cada pulso  $W(t)$  que representa um bit “0” ou um bit “1” no *stream* “Data” é  $T = 1/1.0 \text{ Kbps} = 1\text{ms}$ . Portanto, a largura da banda BW (BW = *bandwidth*) ocupada entre os dois primeiros nulos da magnitude  $|W(f)|$  do espectro de  $W(t)$  é  $\text{BW} = \frac{2}{T} = 2\text{KHz}$ .

Vamos agora supor que o *stream* de bits “Data” transporte informação que é transmitida a uma taxa de 1.0 Mbps (1000 vezes maior que o caso anterior). Então a duração  $T$  de cada pulso  $W(t)$  que representa um bit “0” ou um bit “1” no *stream* “Data” é  $T = 1/1.0 \text{ Mbps} = 1\mu\text{s}$ . Portanto, a banda BW ocupada entre os dois primeiros nulos da magnitude  $|W(f)|$  do espectro de  $W(t)$  é  $\text{BW} = \frac{2}{T} = 2\text{MHz}$  (1000 vezes maior que o caso anterior).

**“Moral da estória”:** Quanto mais bits forem transmitidos por segundo maior será a largura de banda BW necessária p/ transportar o sinal. Note que esta conclusão se refere ao sinal na saída do codificador de fonte, mas como os blocos estão em sequencia (vide figura no slide 3), ela se estende ao espectro do sinal no canal. Portanto, a compressão da informação efetuada no codificador de fonte minimiza a BW ocupada no canal pelo espectro do sinal transmitido.

## Exemplo de compressão da informação – código morse binário (“•” = “0” e “—” = “1”):

### Morse Code Codebook

A ●—	J ●— — —	S ●●●●
B —●●●	K —●—	T —
C —●—●	L ●—●●	U ●●—
D —●●	M — —	V ●●●—
E ●	N —●	W ●— —
F ●●—●	O — — —	X —●●—
G — — ●	P ●— — ●	Y —●— —
H ●●●●	Q — — ● —	Z — — ●●
I ●●	R ●—●	

A cada instante que o manipulador é manualmente pressionado no TX um sinal de RF é transmitido através do canal. O RX faz a heterodinação através de um mixer entre o sinal de RF recebido e o sinal de um oscilador local (BFO – *Beat Frequency Oscillator*) c/ um desvio de frequência de aproximadamente 1KHz da frequência do sinal recebido de modo que um tom audível de 1KHz é produzido no alto-falante do RX.



**LOW**  
... ..  
**ENEMY**  
. . . . .



## Exemplo de compressão da informação – código morse binário (“•” = “0” e “—” = “1”):

### Morse Code Codebook

A ●—	J ●— — —	S ●●●●
B —●●●	K —●—	T —
C —●—●	L ●—●●	U ●●—
D —●●	M — —	V ●●●—
E ●	N —●	W ●— —
F ●●—●	O — — —	X —●●—
G — — ●	P ●— — ●	Y —●— —
H ●●●●	Q — — ● —	Z — — ●●
I ●●	R ●—●	

“E” e “T” são os caracteres utilizados c/ + frequência na língua inglesa, então o Codebook (*look up table*) atribui a eles palavras binárias de 1 bit. “A”, “I”, “M”, “N” são os segundos caracteres utilizados c/ + frequência, então é atribuído a eles palavras binárias de 2 bits. E assim sucessivamente de modo que na média ao longo do tempo ocorre uma significativa redução do número de bits transmitidos porque os caracteres que ocorrem com mais frequência são representados por palavras binárias de menor número de bits enquanto os caracteres que ocorrem raramente são representados por palavras binárias de maior número de bits..



Correção de bits recebidos em erro – exemplo em que um código de bloco (6,3) no “channel encoder” do TX mapeia mensagens (3 bits info útil) em palavras-código (3 bits info útil + 3 bits paridade) e a info de paridade é usada pelo “channel decoder” do RX para corrigir o bit recebido em erro

Exemplo do *Codebook (look-up table)* de um código de bloco (6,3)

Mensagem - palavra binária de 3 bits na entrada do “channel-encoder” (e na saída do “channel decoder” se o erro de bits ocorrido no canal for corrigível)			Palavra-código ( <i>codeword</i> ) - palavra binária de 6 bits na saída do “channel-encoder” (e na entrada do “channel decoder” se não ocorreu erro de bits no canal)					
0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	0	1
0	1	0	0	1	0	1	1	1
0	1	1	0	0	1	0	1	1
1	0	0	1	1	0	0	1	0
1	0	1	1	0	1	1	1	0
1	1	0	0	1	1	1	0	0
1	1	1	1	1	1	0	0	1

$T_{msg}$

$T_{cwd}$

$\Rightarrow T_{msg} = T_{cwd}$  caso contrário o espectro da informação codificada será alterado.

$d_H \Rightarrow$  distância de hamming: número de bits desiguais entre duas palavras binárias. Quanto maior for a  $d_H$  mínima no conjunto de  $d_H$ 's entre todos os pares de *codewords* do *codebook*, mais dissimilares serão entre si as *codewords* e mais resiliente será o código a erros de bit. No *codebook* c/ 8 *codewords* deste exemplo, se calcularmos as  $7+6+5+4+3+2+1=28$   $d_H$ 's entre todos os pares de *codewords* e identificarmos o valor mínimo de  $d_H$  obteremos  $d_{Hmin}=3$ .

Vamos supor que, em um determinado instante, a palavra binária na entrada do “channel-encoder” no TX (mensagem) seja “010”, a qual o *codebook* transforma na *codeword* “010111”. Ruído no canal degrada a informação fazendo com que o *codeword* recebido na entrada do “channel-decoder” no RX seja “010011” (1 bit errado).

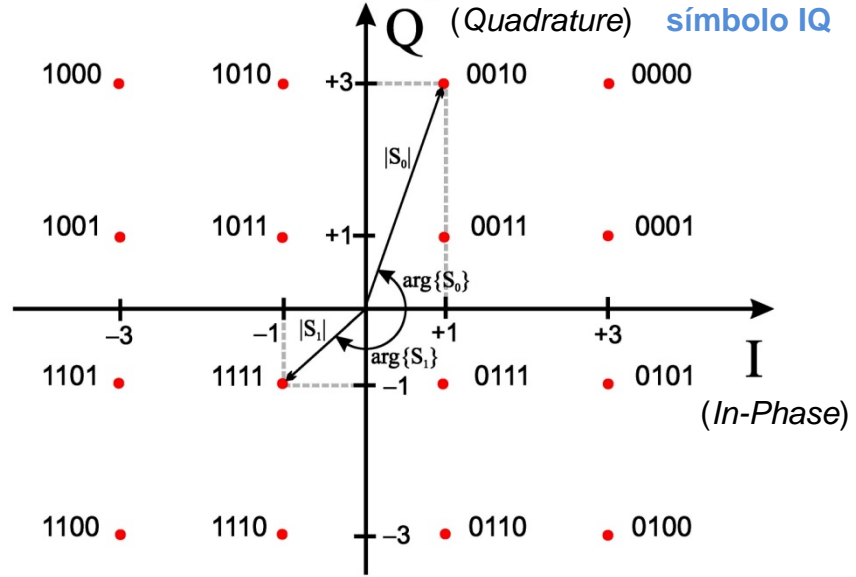
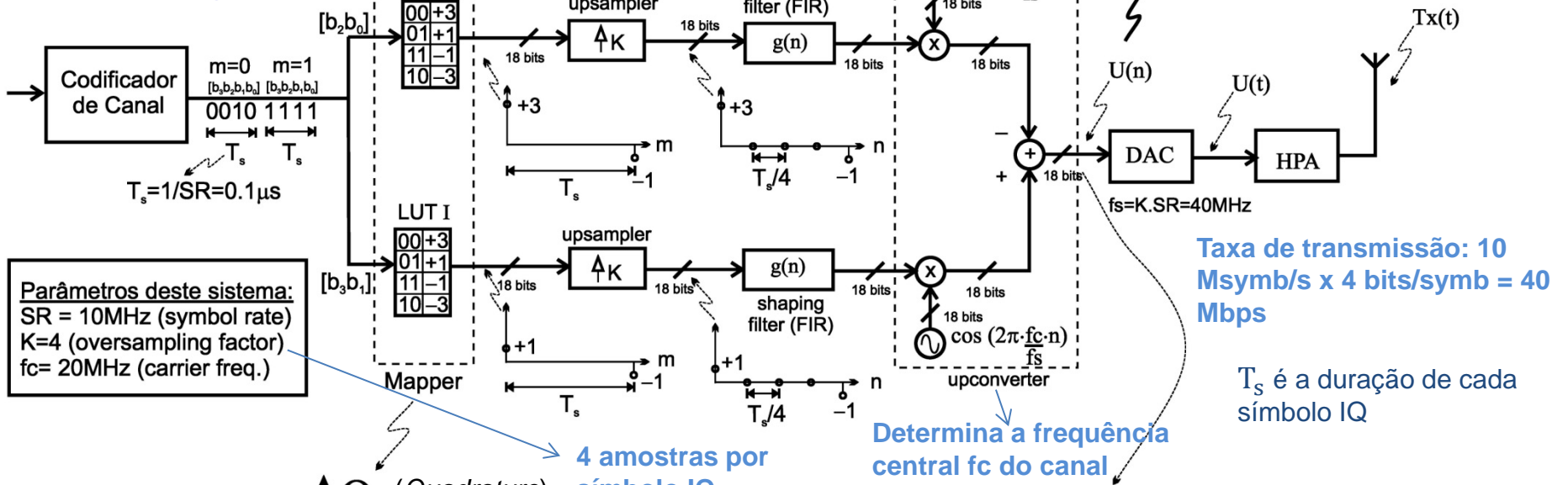
Para corrigir o bit errado em “010011”, simplificadaamente, o “channel-decoder” determina as 8 respectivas  $d_H$ 's entre “010011” recebido e as 8 *codewords* do *codebook*, resultando no conjunto  $d_{H}RX = \{ 3, 4, 1, 2, 2, 5, 5, 3 \}$ . O decoder identifica então qual a menor  $d_H$  no conjunto  $d_{H}RX$  (no caso, a menor  $d_H$  é 1) e infere que a mensagem transmitida foi “010”, porque esta é a mensagem associada à *codeword* “010111” que resultou na menor  $d_H$  no conjunto  $d_{H}RX$ . Note que o bit errado foi corrigido porque a mensagem originalmente transmitida “010” foi recuperada no RX, apesar do bit errado.



Correção dos bits recebidos em erro pelo “channel decoder” – a importância dos bits adicionais de paridade para maximizar a  $d_H$  mínima no conjunto de  $d_H$ 's entre todos os pares de *codewords* do *codebook*, de modo a tornar as *codewords* do *codebook* o mais dissimilar entre si. A dissimilaridade entre *codewords* viabilizada pelos bits de paridade facilita a sua identificação pelo “channel decoder”, que age como uma pessoa tentando identificar o indivíduo culpado em uma cena de identificação de criminoso através de características particulares (bits de paridade) nos indivíduos:



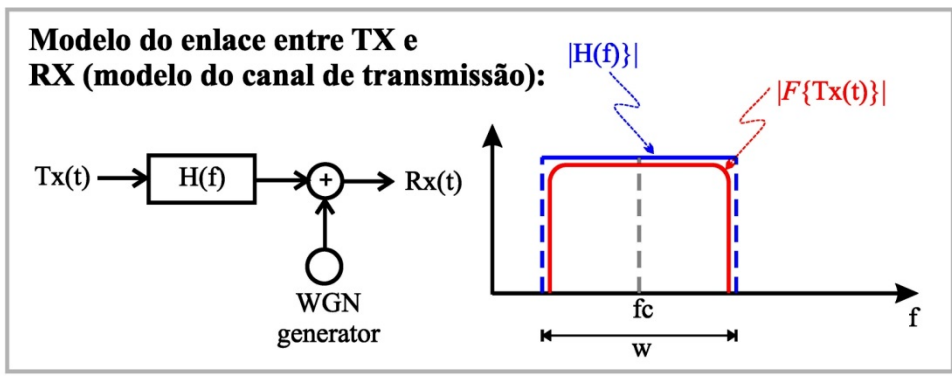
# Exemplo e detalhe do Modulador Digital (TX)



**$U_m(n)$  p/  $m=0$  e  $m=1$  neste exemplo:**

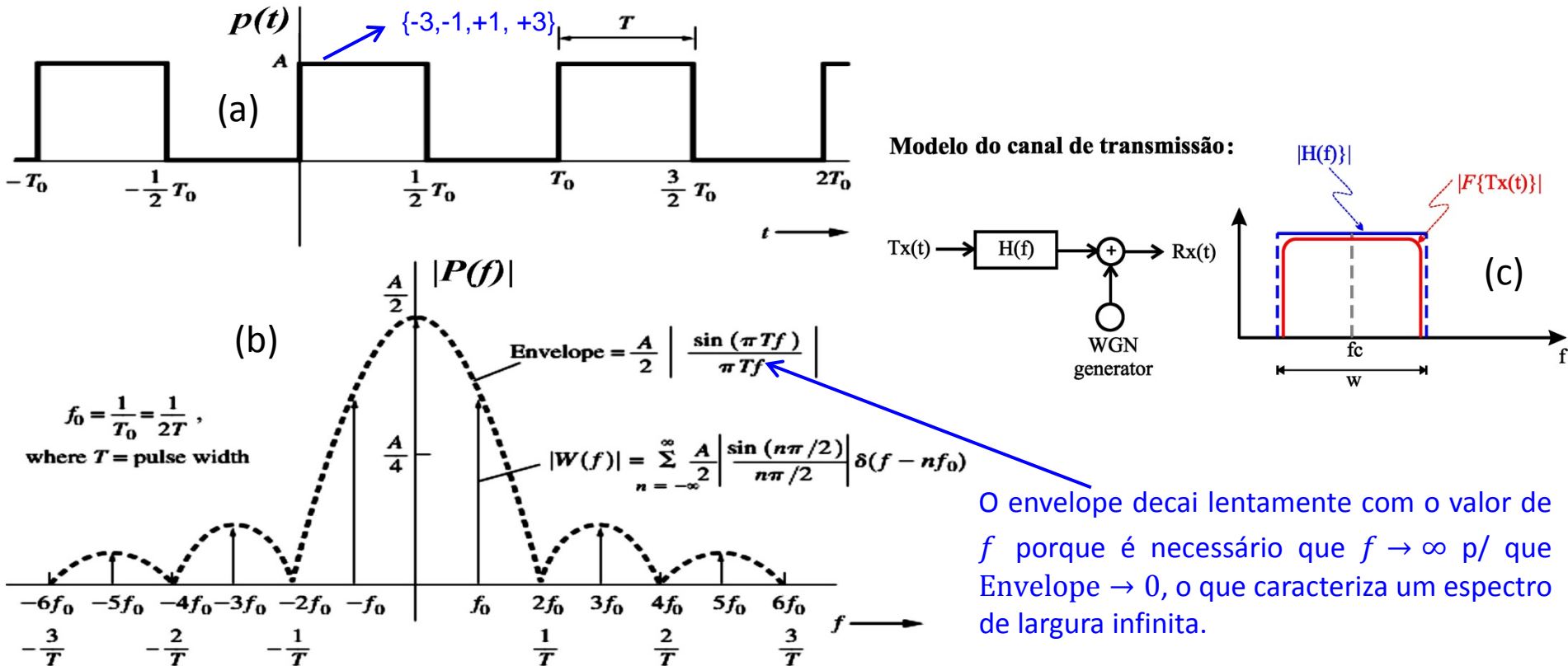
$$U_0(n) = \{\sqrt{3^2 + 1^2} * g(n)\} \cdot \cos(2\pi \cdot \frac{20}{40} \cdot n + \text{atan}(3/1))$$

$$U_1(n) = \{\sqrt{(-1)^2 + (-1)^2} * g(n)\} \cdot \cos(2\pi \cdot \frac{20}{40} \cdot n - \text{atan}(-1/-1))$$



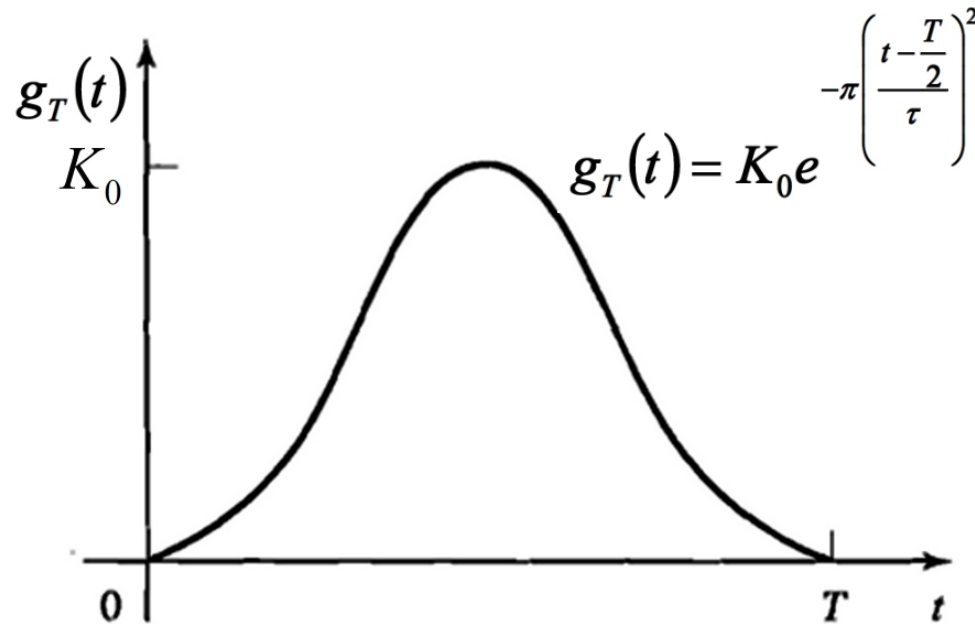
## Shaping filter (filtro de formatação de pulso) e Up-sampler (superamostrador) – contenção espectral da sequencia de pulsos quadrados na saída das LUTs I e Q.

Vamos supor que retirássemos o DAC na saída do *upconverter* no slide anterior e em substituição colocássemos dois DACs respectivamente nas saídas das LUTs I e Q, de modo que o sinal seria discreto antes dos DACs e contínuo após os DACs. Fizemos isto para que o sistema opere com um *shaping filter* analógico, cuja análise no domínio frequência é mais simples (faremos a análise do caso digital ao final desta discussão). Note nesta situação, com referência ao slide anterior, que a amplitude de cada pulso quadrado de duração  $T_s$  no trem de pulso na saída dos DACs I e Q é um dos valores na 2ª coluna das respectivas LUTs  $\{-3, -1, +1, +3\}$ , valores que dependem da palavra binária de dois bits na 1ª coluna, na entrada da LUT. Note também que a largura do espectro  $P(f)$  de um trem de pulsos  $p(t)$  de duração  $T = T_s$  (conforme (a) abaixo) é **ilimitado no domínio frequência** (conforme (b) abaixo), e, portanto, é incompatível com a largura de banda do canal limitada em  $W$  (ver slide anterior e (c) abaixo), alocado pelo órgão regulador (Anatel, FCC, etc ...).



## Shaping filter (filtro de formatação de pulso) e Up-sampler (superamostrador) – contenção espectral da sequencia de pulsos quadrados na saída das LUTs I e Q.

Uma solução usual para efetuar a contenção espectral de um trem de pulsos  $p(t)$  de duração  $T = T_s$  (ver (a) slide anterior) é aplicar um *shaping filter* Gaussiano, conforme abaixo, p/ suavizar as bordas de subida e descida dos pulsos, dado que a rápida (idealmente instantânea) variação no tempo das bordas de subida e descida dos pulsos quadrados são os responsáveis pela largura infinita do espectro  $P(f)$  conforme mostrado em (b) do slide anterior.

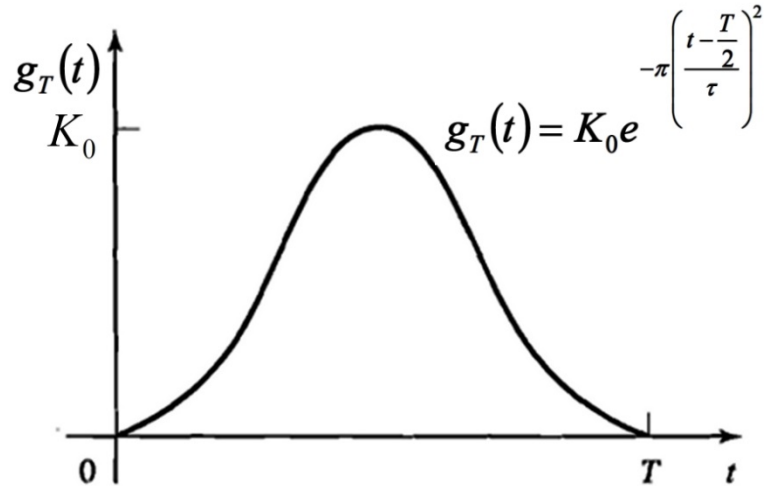


Para cada pulso quadrado de duração  $T_s$  no trem de pulso na saída das LUTs+DACs, o *shaping filter* Gaussiano responde com um pulso  $g_T(t)$  conforme acima, de amplitude proporcional ao pulso quadrado na entrada do filtro e de duração  $T \leq T_s$  (usualmente  $T = T_s$ ). **Deve-se evitar a todo custo  $T > T_s$  porque os pulsos se superporiam no tempo e gerariam a mesma ISI gerada pelo multipath no canal, só que agora estaríamos gerando ISI no próprio hardware ...**

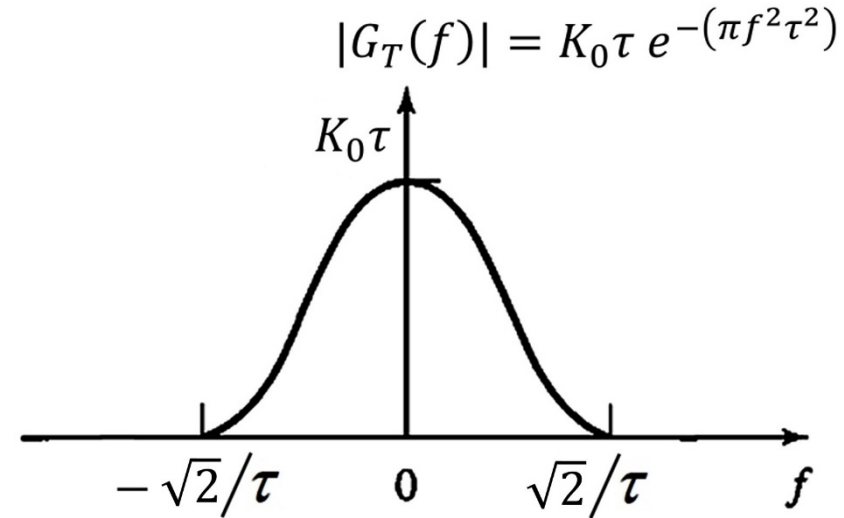


## Shaping filter (filtro de formatação de pulso) e Up-sampler (superamostrador) – contenção espectral da sequencia de pulsos quadrados na saída das LUTs I e Q.

O espectro do pulso  $g_T(t)$  é obtido aplicando a Transformada de Fourier à  $g_T(t) = K_0 e^{-\pi \left(\frac{t-T}{\tau}\right)^2}$ , isto é,  $G_T(f) = \mathcal{F}\{g_T(t)\} = K_0 \tau e^{-(\pi f^2 \tau^2 + j\pi f T)} = K_0 \tau e^{-\pi f^2 \tau^2} e^{-j\pi f T}$  (ver apêndice A), de modo que  $|G_T(f)| = K_0 \tau e^{-\pi f^2 \tau^2}$ ,  $\angle G_T(f) = -j\pi f T$  e  $T = T_s$ .



$\mathcal{F}\{\cdot\}$   
 $\Rightarrow$



Note que a magnitude  $|G_T(f)|$  do espectro de  $g_T(t)$  decai exponencialmente com o quadrado de  $f$ , que é um decaimento muito rápido se comparado com o decaimento do envelope do espectro  $P(f)$  de um trem de pulsos retangulares  $p(t)$ , conforme (b) no slide 11, envelope que decai apenas lentamente com o valor de  $f$ . A rapidez exponencial do decaimento da magnitude  $|G_T(f)|$  é tal que  $20\log|G_T(f = \sqrt{2}/\tau)| - 20\log|G_T(f = 0)| = -54.6$  dB (535.5 vezes menor), o que permite considerar  $|G_T(f)|$  nulo para frequências fora do intervalo  $-\sqrt{2}/\tau \leq f \leq \sqrt{2}/\tau$ . Já a magnitude do espectro  $P(f)$  do trem de pulsos retangulares  $p(t)$  visto no slide 11 decai tão lentamente com o valor de  $f$  que é necessário que  $f \rightarrow \infty$  p/ que Envelope  $\rightarrow 0$ , o que caracteriza um espectro de largura infinita dado que a magnitude do espectro somente pode ser considerada nula para frequências fora do intervalo  $-\infty \leq f \leq \infty$ .

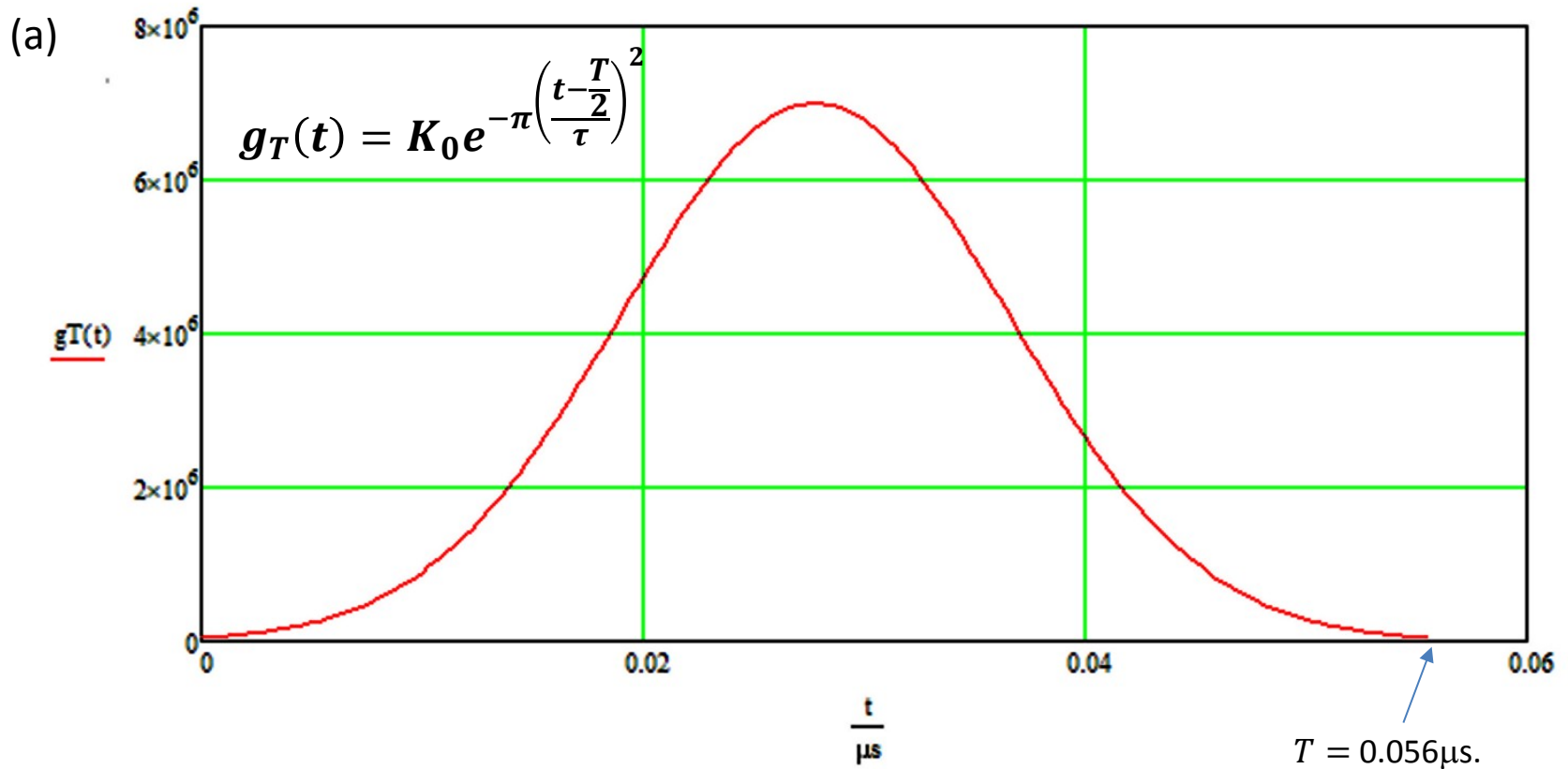
Portanto, transmitir pulsos na forma  $s_m(t) = A g_T(t) = A K_0 e^{-\pi \left(\frac{t-T}{\tau}\right)^2}$ , com  $A$  correspondendo à amplitude dos pulsos retangulares na saída da LUT + DAC resulta um sinal espectralmente contido e factível de ser transmitido através do canal de largura  $W$ .

## Shaping filter (filtro de formatação de pulso) e Up-sampler (superamostrador) – exemplo da contenção espectral obtida com um filtro Gaussiano contínuo (analógico)

Consideremos um *shaping filter* Gaussiano com  $K_0 = 7 \times 10^6$  e  $\tau = 0.022\mu\text{s}$  que faz a contenção espectral em um modulador digital cujo SymbolRate=18MHz. **Pede-se:** (a) Plote o gráfico de  $g_T(t) = K_0 e^{-\pi\left(\frac{t-T}{\tau}\right)^2}$  (b) Plote o gráfico da magnitude do espectro  $|G_T(f)| = K_0 \tau e^{-\pi f^2 \tau^2}$ .

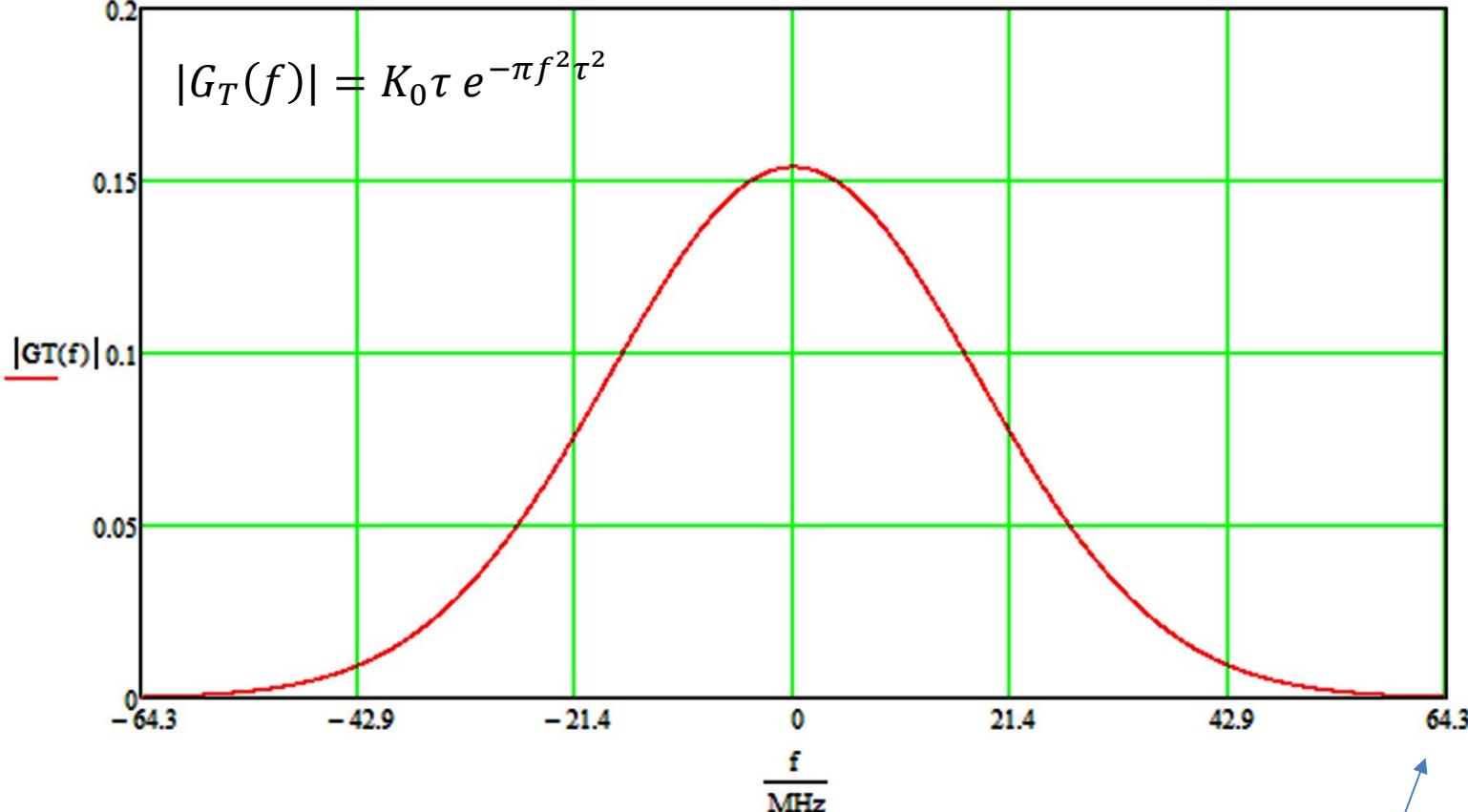
### Solução:

Duração do símbolo IQ:  $T = 1/\text{SymbolRate} = 1/18\text{MHz} = 0.056\mu\text{s}$ .



**Shaping filter (filtro de formatação de pulso) e Up-sampler (superamostrador) – exemplo da contenção espectral obtida com um filtro Gaussiano contínuo (analógico)**

(b)



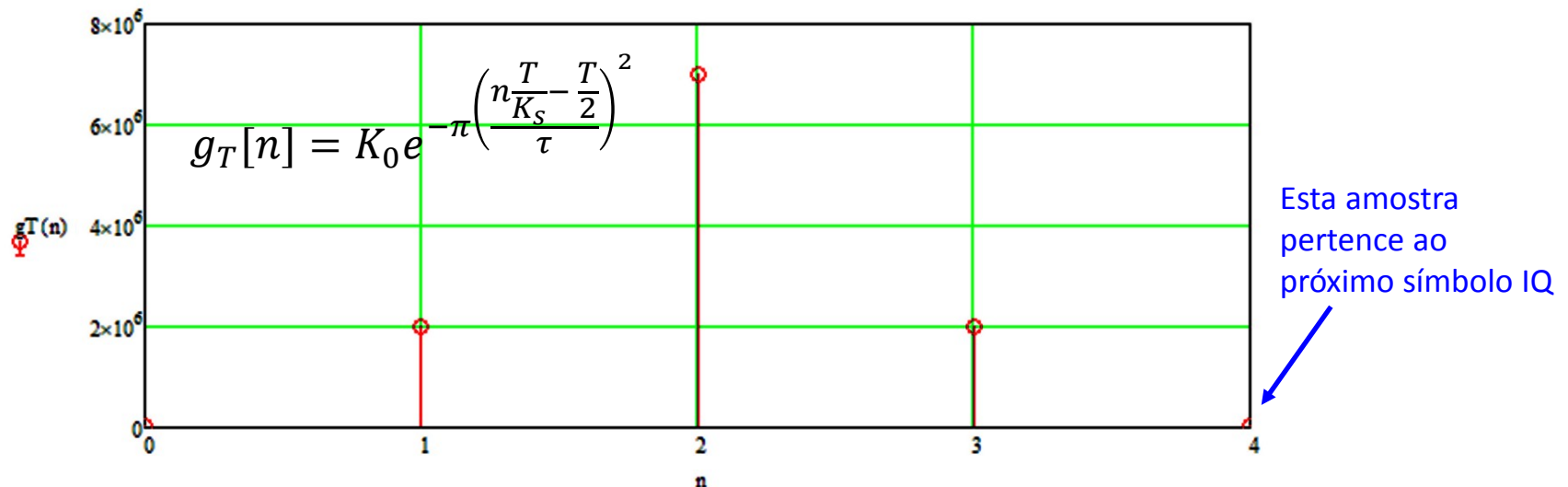
$\frac{\sqrt{2}}{\tau} = 64.3\text{MHz}$

## Shaping filter (filtro de formatação de pulso) e Up-sampler (superamostrador) – discretização no tempo do filtro Gaussiano contínuo

Para um modulador digital em que o DAC está na saída do *upconverter* (que é a arquitetura usual), como mostrado no slide 10, o *shaping filter* é digital e a sua resposta no tempo é discreta, diferente do *shaping filter* analógico analisado nos slides anteriores cuja resposta é contínua. A resposta discreta do *shaping filter* é facilmente obtida da resposta contínua. Para um modulador com um *oversampling fator*  $K_s$ , a passagem da resposta contínua para a resposta discreta é conforme:

$$g_T(t) = K_0 e^{-\pi \left( \frac{t - \frac{T}{2}}{\tau} \right)^2} \quad \xrightarrow{t = n \frac{T}{K_s}} \quad g_T[n] = K_0 e^{-\pi \left( \frac{n \frac{T}{K_s} - \frac{T}{2}}{\tau} \right)^2}$$

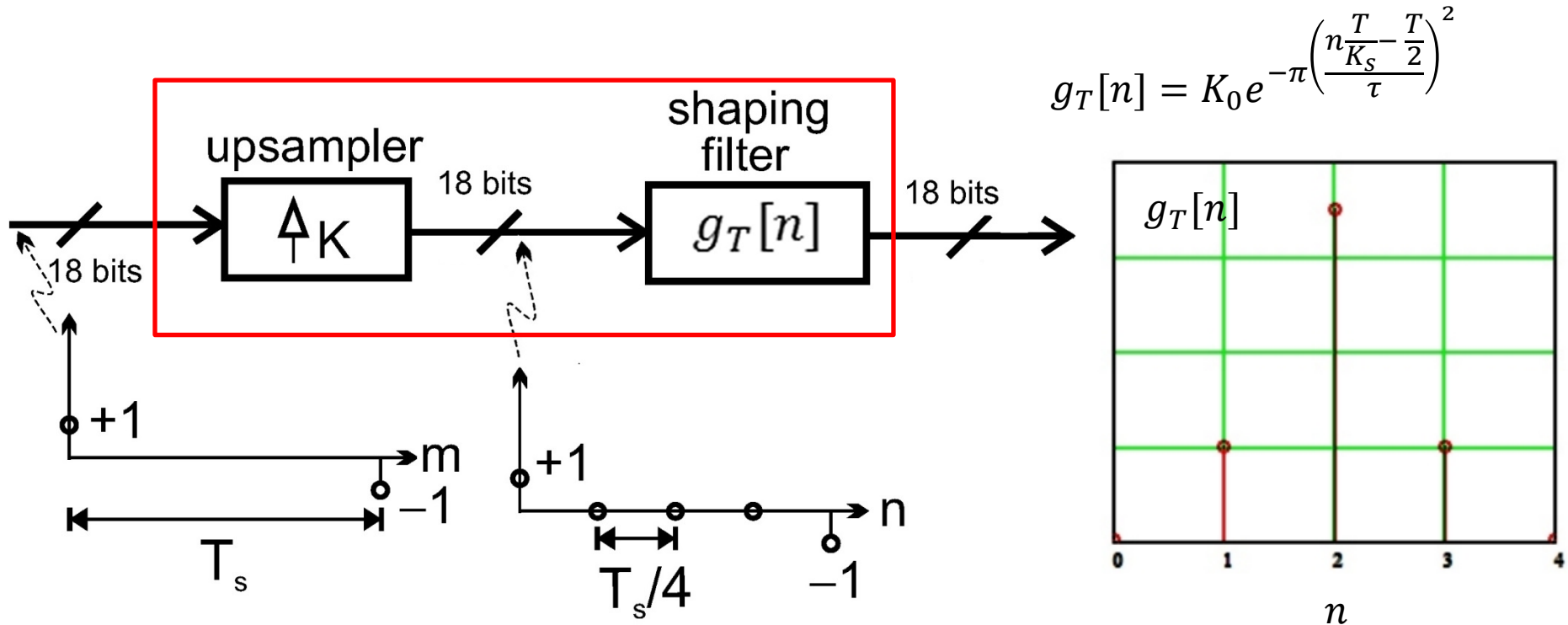
Por exemplo, o *shaping filter* Gaussiano com  $K_0 = 7 \times 10^6$  e  $\tau = 0.022 \mu\text{s}$  contínuo no tempo, cuja resposta  $g_T(t) = K_0 e^{-\pi \left( \frac{t - \frac{T}{2}}{\tau} \right)^2}$  é mostrada no gráfico do slide 14, ao ser discretizado com  $K_s = 4$  (4 amostras por símbolo IQ) como mostrado no slide 10, apresentará a resposta  $g_T[n] = K_0 e^{-\pi \left( \frac{n \frac{T}{K_s} - \frac{T}{2}}{\tau} \right)^2}$  conforme gráfico abaixo:





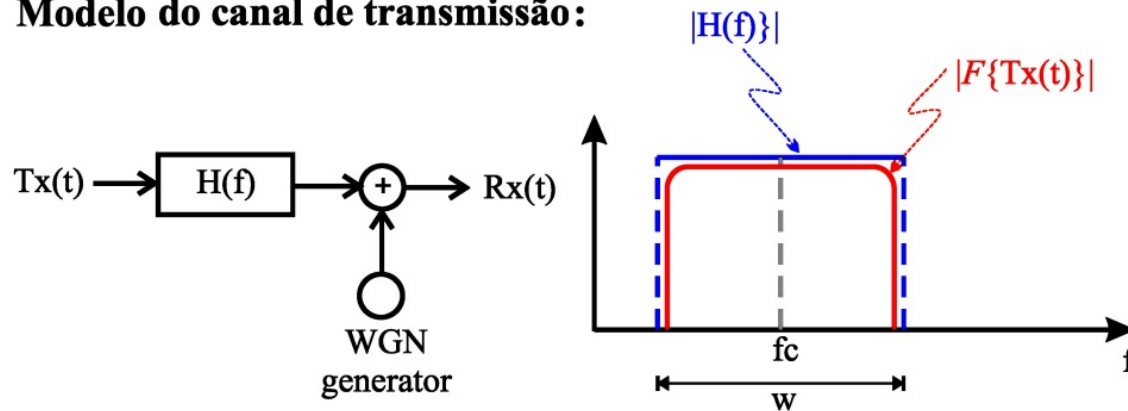
## Shaping filter (filtro de formatação de pulso) e Up-sampler (superamostrador) – discretização no tempo do filtro Gaussiano contínuo

Portanto, note que, para um modulador com um *oversampling fator*  $K_s$ , como o modulador mostrado no slide 10, a resposta  $g_T[n]$  para  $K_s = 4$  mostrada no exemplo do slide anterior pode ser considerada a resposta ao impulso do bloco formado pelo *upsampler* seguido do *shaping filter* (bloco em vermelho abaixo):



## Upconverter – a definição da frequência central $f_c$ do canal de largura $W$

### Modelo do canal de transmissão:

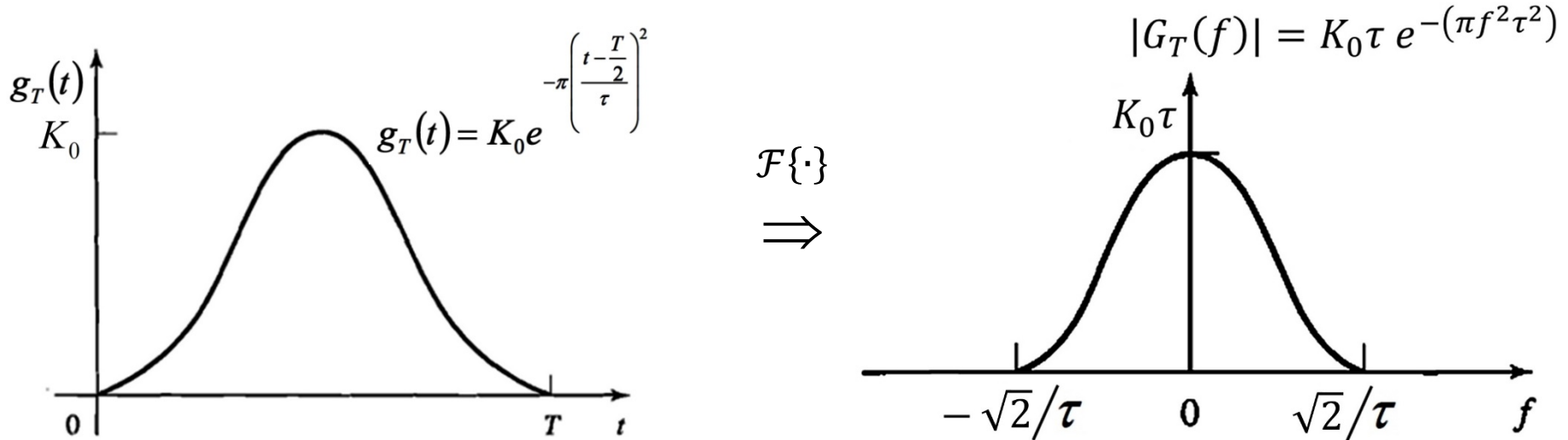


Passamos a discutir agora o processo que define a frequência central  $f_c$  do canal de comunicações, conforme determina a alocação de frequências determinada pelo órgão regulador (Anatel, FCC, etc ...).

Para efeito de análise do *upconverter* (vide slide 10), vamos novamente utilizar a abordagem simplificada em que retira-se o DAC na saída do *upconverter* no slide 10 e em substituição coloca-se dois DACs respectivamente nas saídas das LUTs I e Q, de modo que o sinal é discreto antes dos DACs e contínuo após os DACs. Novamente estamos fazendo isto para que o sistema opere com um *shaping filter* analógico e com um *upconverter* analógico, cuja análise no domínio frequência é mais simples do que o caso discreto. As inferências e conclusões são as mesmas nos dois casos – discreto ou contínuo, com a única diferença de que no caso discreto as frequências das expressões analíticas envolvidas são normalizadas pela frequência  $f_s$  de amostragem do DAC.

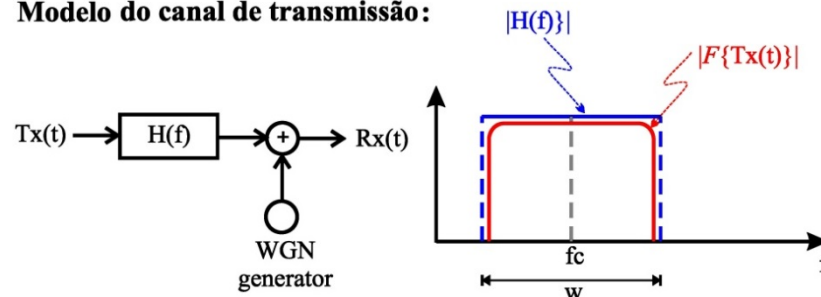
## Upconverter – a definição da frequência central $f_c$ do canal de largura $W$

Conforme já visto no slide 13, a informação é transmitida através de um trem de pulsos da forma  $s_m(t) = Ag_T(t) = AK_0 e^{-\pi \left(\frac{t-T/2}{\tau}\right)^2}$ , com  $A$  correspondendo à amplitude dos pulsos retangulares na saída da LUT + DAC, e resulta um sinal espectralmente contido e factível de ser transmitido através do canal de largura  $W$ , mas cujo espectro é *baseband*, isto é, a frequência central é  $f_c = 0$ , conforme mostra o gráfico da magnitude  $|G_T(f)|$  do espectro, abaixo:



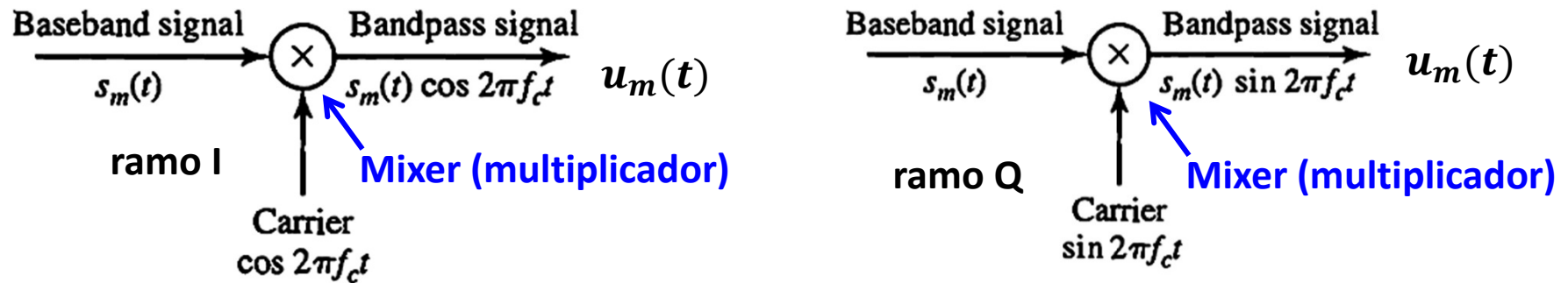
No entanto, o órgão regulador (Anatel, FCC ...) determina uma frequência central  $f_c$  que depende do tipo de serviço. É necessário, portanto, transformar o espectro  $G_T(f)$  *baseband* em um espectro *passband* com frequência central  $f_c$  definida pelo órgão regulador:

**Modelo do canal de transmissão:**



## Upconverter – a definição da frequência central $f_c$ do canal de largura $W$

Vimos no slide 10 que o *upconverter* multiplica o trem de pulsos  $s_m(t) = Ag_T(t) = AK_0 e^{-\pi\left(\frac{t-T}{\tau}\right)^2}$  na saída de cada *shaping filter* respectivamente por um seno e por um cosseno, ambos de frequência  $f_c$ , respectivamente no ramo I e no ramo Q do modulador, processo que é denominado de **heterodinação** (vide Apêndice B):



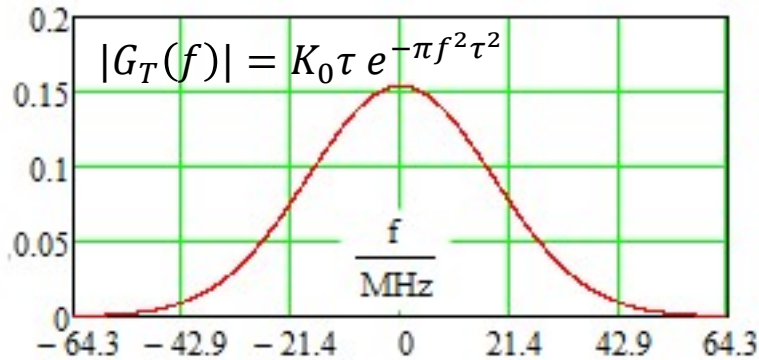
A *heterodinação* entre o sinal  $s_m(t)$  e  $\cos 2\pi f_c t$  no ramo I do modulador e a *heterodinação* entre o sinal  $s_m(t)$  e  $\sin 2\pi f_c t$  no ramo Q do modulador resultará, conforme Apêndice B, na translação do espectro *baseband*  $G_T(f)$  dos pulsos  $s_m(t)$  para um espectro *passband*  $U_m(f)$  na forma:

$$U_m(f) = \frac{A}{2} [G_T(f - f_c) + G_T(f + f_c)]$$



## Upconverter – a definição da frequência central $f_c$ do canal de largura $W$ – exemplo

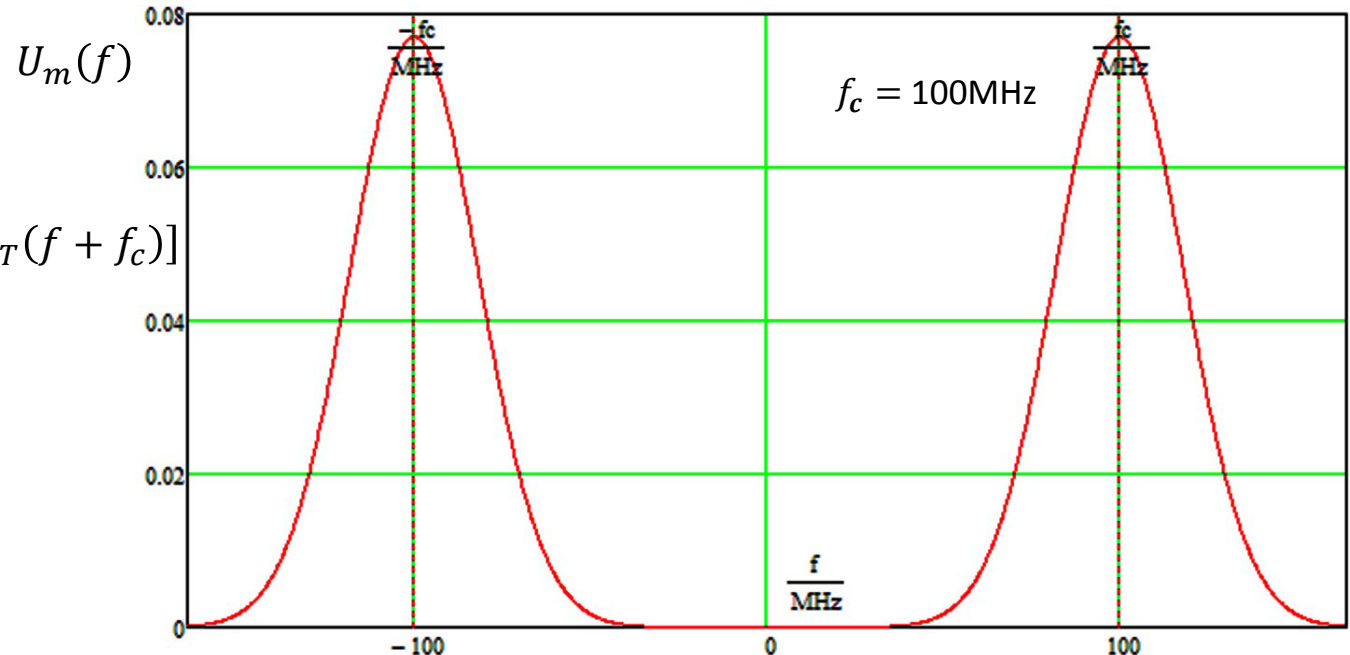
**Exemplo:** Consideremos o trem de pulsos gerados por um *shaping filter* Gaussiano com resposta  $g_T(t) = K_0 e^{-\pi \left(\frac{t-T}{\tau}\right)^2}$ ,  $K_0 = 7 \times 10^6$  e  $\tau = 0.022 \mu\text{s}$ , que faz a contenção espectral em um modulador digital cujo SymbolRate=18MHz e  $T = 1/\text{SymbolRate} = 1/18\text{MHz} = 0.056 \mu\text{s}$ , conforme slide 14. O gráfico da magnitude do espectro  $|G_T(f)| = K_0 \tau e^{-\pi f^2 \tau^2}$  do trem de pulsos antes do *upconverter* é conforme abaixo (vide slide 15):



**Pede-se:** Sabendo que o modulador efetua a *heterodinação* entre o sinal  $s_m(t)$  e  $\cos 2\pi f_c t$  no ramo **I** e efetua a *heterodinação* entre o sinal  $s_m(t)$  e  $\sin 2\pi f_c t$  no ramo **Q**, sendo  $f_c = 100\text{MHz}$  a frequência central do canal, plote o gráfico da magnitude do espectro do espectro do sinal  $u_m(t)$  na saída do upconverter.

**Solução:**

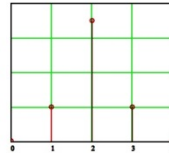
$$U_m(f) = \frac{A}{2} [G_T(f - f_c) + G_T(f + f_c)]$$



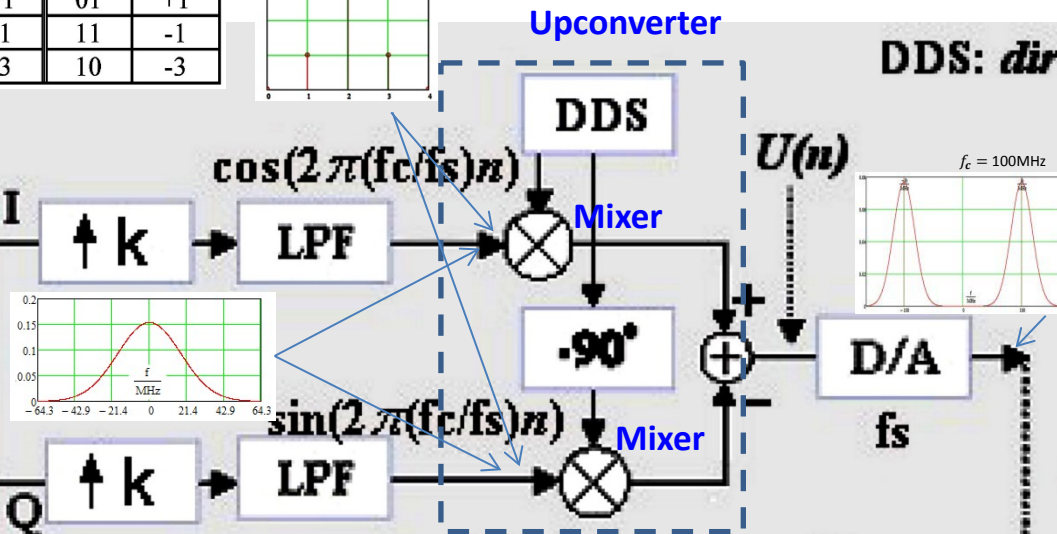
LUT – Look Up Table (TX)

I: [b <sub>3</sub> b <sub>1</sub> ]		Q: [b <sub>2</sub> b <sub>0</sub> ]	
00	+3	00	+3
01	+1	01	+1
11	-1	11	-1
10	-3	10	-3

O TX (ver slide 10) e o RX (demodulador) operando em conjunto:



bit stream de entrada



DDS: *direct digital synthesizer*

$$k = \frac{f_s}{\text{SymbolRate}}$$

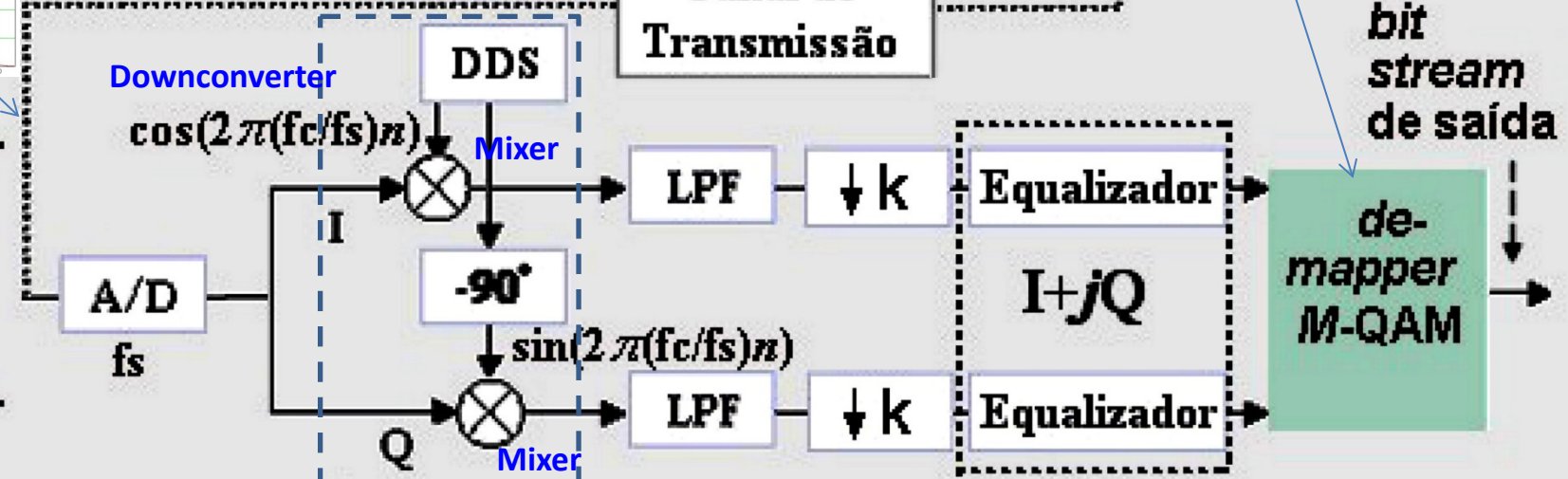
TX  
LUT - Look Up Table (RX)

I: [b <sub>3</sub> b <sub>1</sub> ]		Q: [b <sub>2</sub> b <sub>0</sub> ]	
NN{+3}	00	NN{+3}	00
NN{+1}	01	NN{+1}	01
NN{-1}	11	NN{-1}	11
NN{-3}	10	NN{-3}	10

Canal de Transmissão

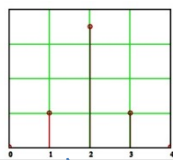
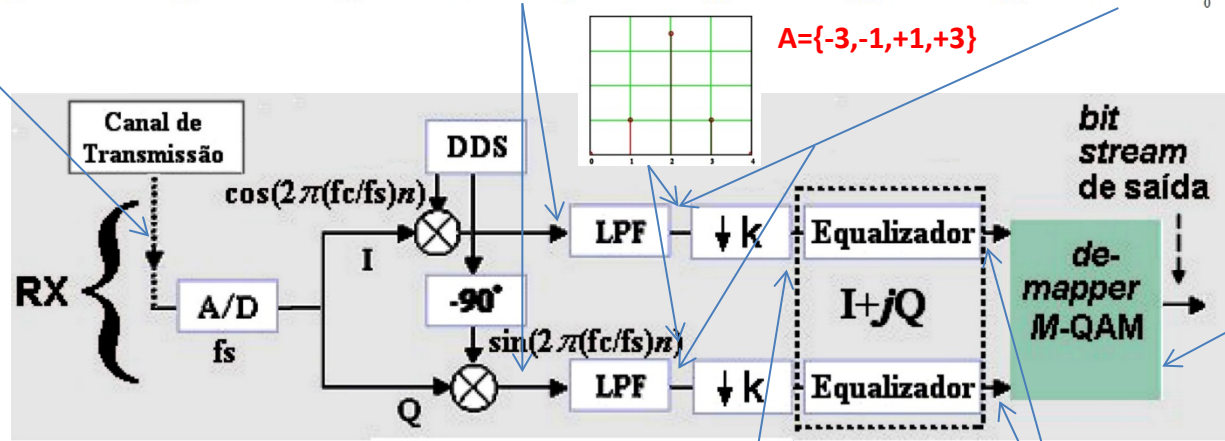
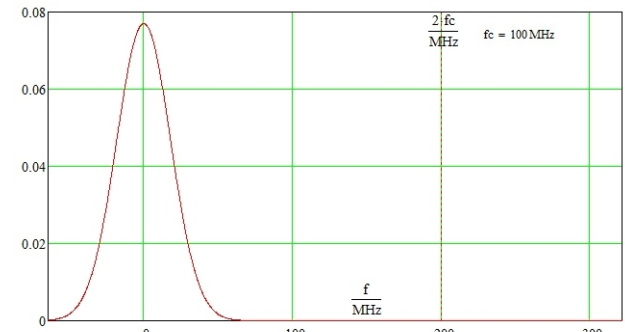
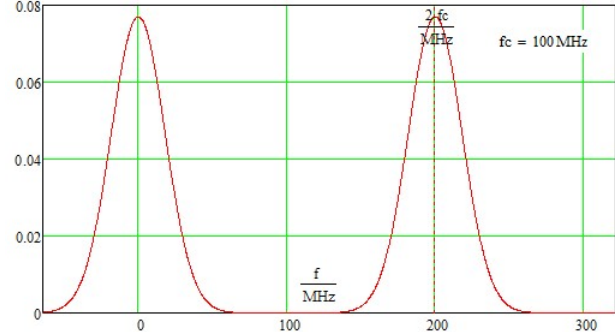
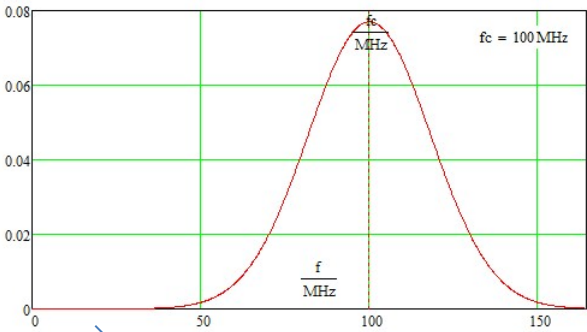
bit stream de saída

RX



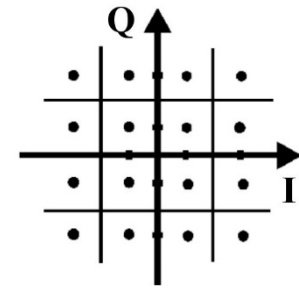
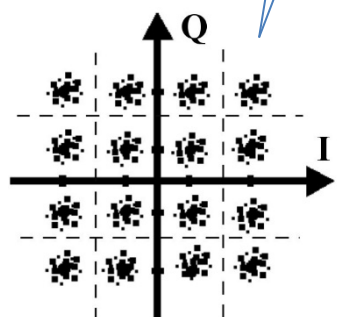
NN{·} é o operador *Nearest Neighbour* (vizinho mais próximo), o qual retorna "Verdadeiro" quando o valor de entrada na coluna esquerda da LUT RX é o vizinho mais próximo (menor distância Euclidiana) de seu argumento {·}, e retorna "Falso" em caso contrário.

# O demodulador digital (RX)



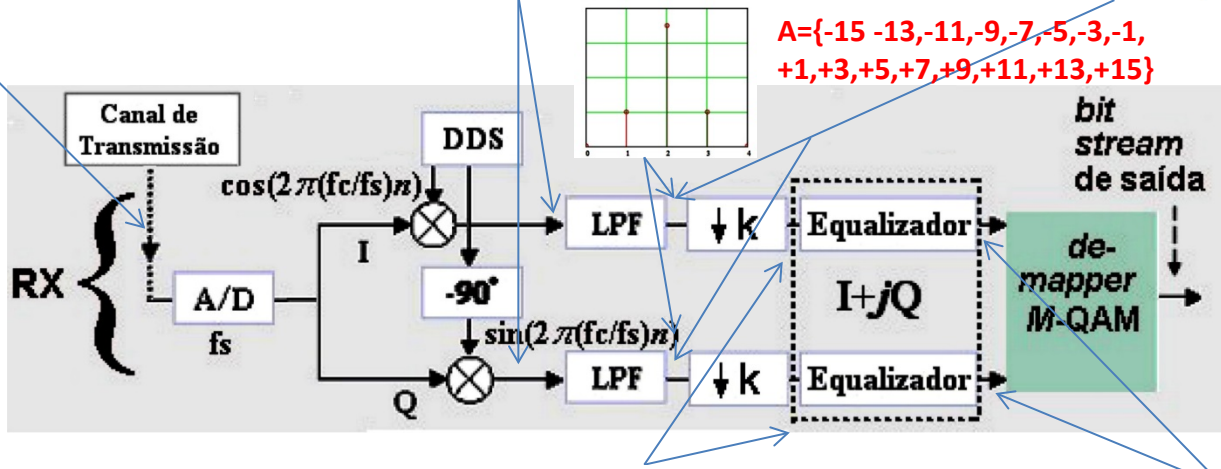
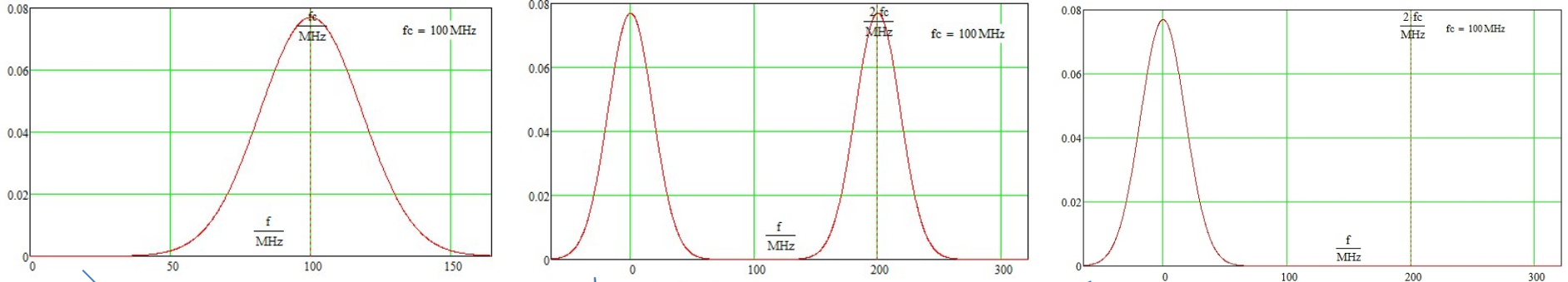
$A = \{-3, -1, +1, +3\}$

I: $[b_3 b_1]$		Q: $[b_2 b_0]$	
$NN\{+3\}$	00	$NN\{+3\}$	00
$NN\{+1\}$	01	$NN\{+1\}$	01
$NN\{-1\}$	11	$NN\{-1\}$	11
$NN\{-3\}$	10	$NN\{-3\}$	10



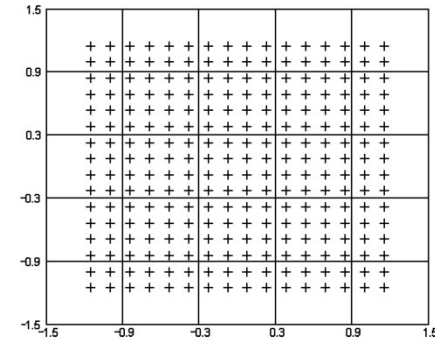


O que acontece com a taxa e com a robustez do enlace se configurarmos o modulador do slide 10, para utilizar uma modulação 256-QAM ao invés da modulação 16-QAM, mantendo symbol rate = 10Msymb/s?

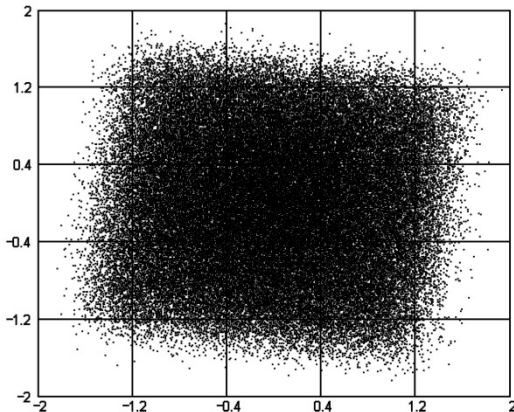


$A = \{-15, -13, -11, -9, -7, -5, -3, -1, +1, +3, +5, +7, +9, +11, +13, +15\}$

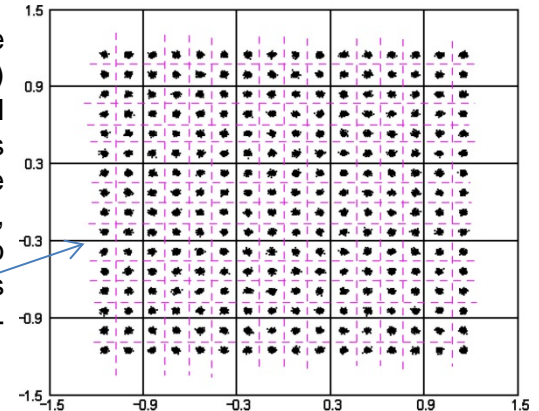
Constelação 256-QAM de referência (normalizada p/ potência unitária):



Taxa de transmissão: 10 Msymb/s x 8 bits/symb = 80 Mbps ⇒ dobro da taxa obtida com 16-QAM (ver slide 10)

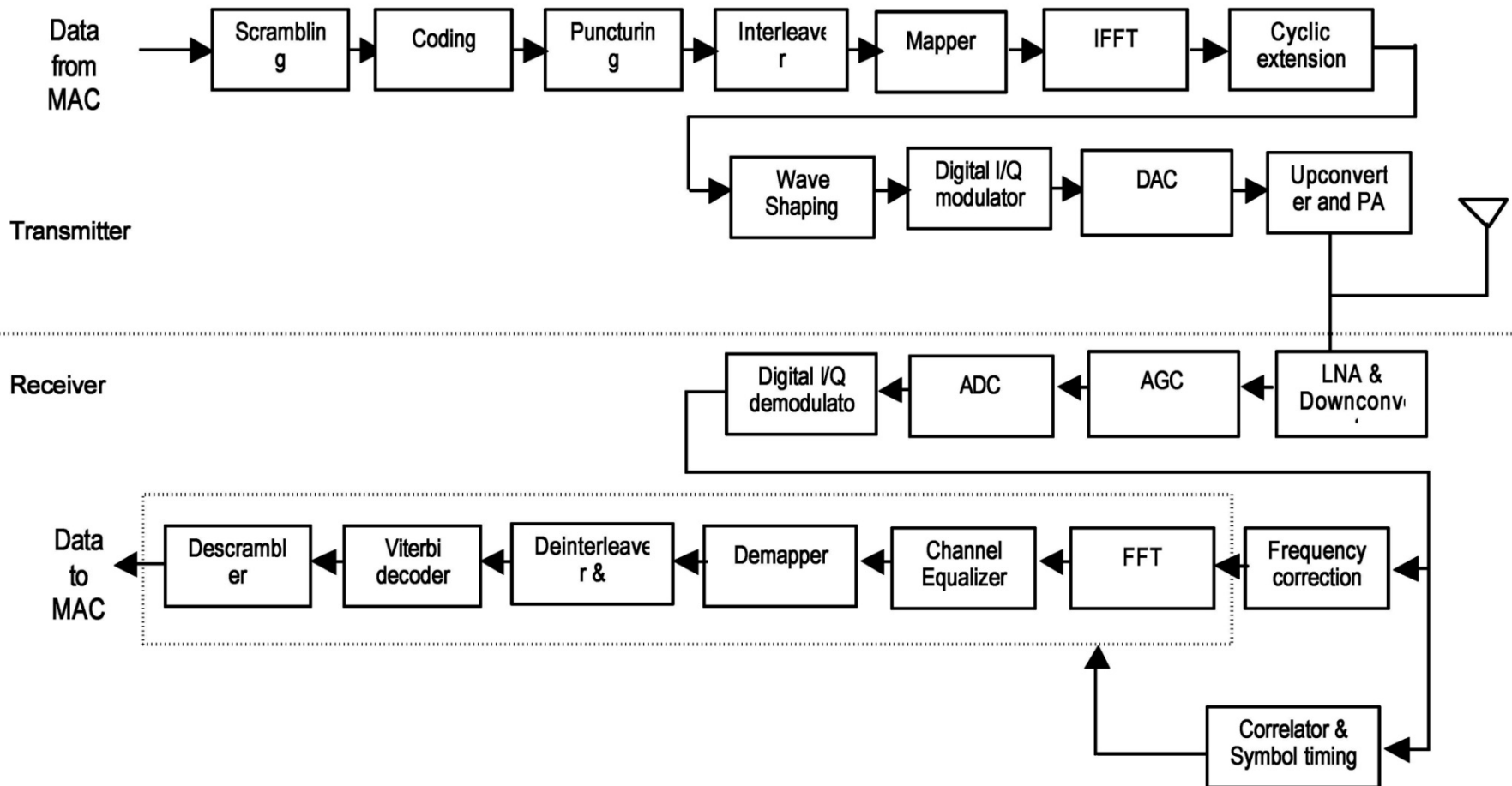


Note que as regiões de decisão (em magenta) da constelação 256-QAM são 4 vezes menores que as regiões de decisão de 16-QAM, portanto a modulação 256-QAM é 4 vezes menos robusta que 16-QAM:



## Design Flow – metodologia de projeto de sistemas de comunicações – implementação em ASIC (Application-Specific Integrated Circuit) e/ou em FPGA (Field-Programmable Gate Array)

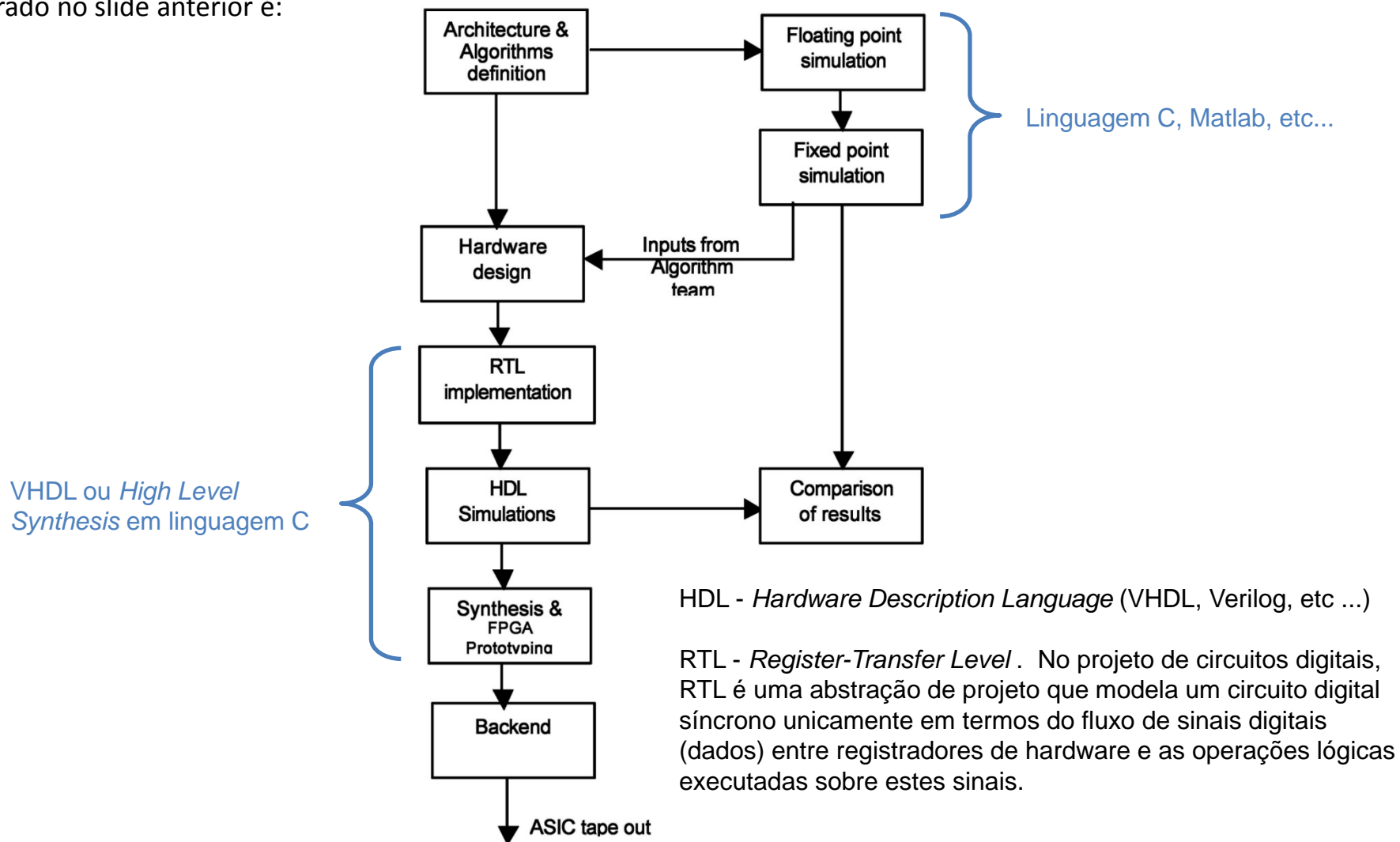
Consideremos, por exemplo, o projeto em VLSI (Very Large Scale Integration) de um modem OFDM (Orthogonal Frequency Division Multiplexing) IEEE 802.11a, cujo diagrama de blocos funcionais é similar ao abaixo (ver [http://www.fccdecastro.com.br/pdf/VLSI\\_OFDM.pdf](http://www.fccdecastro.com.br/pdf/VLSI_OFDM.pdf)):





## Design Flow – metodologia de projeto de sistemas de comunicações – implementação em ASIC (Application-Specific Integrated Circuit) e/ou FPGA (Field-Programmable Gate Array)

Uma possível metodologia de projeto para um modem OFDM IEEE 802.11a, conforme arquitetura de blocos funcionais mostrado no slide anterior é:



**High Level Synthesis em linguagem C:**

[https://www.xilinx.com/support/documentation/sw\\_manuals/xilinx2019\\_1/ug871-vivado-high-level-synthesis-tutorial.pdf](https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_1/ug871-vivado-high-level-synthesis-tutorial.pdf)

## Apêndice A:

<b>Operation</b>	<b>Formula</b>
Rectangular to Polar Conversion	$z = x + jy = re^{j\theta}$ where $r = \sqrt{x^2 + y^2}$ and $\theta = \arctan(y/x)$
Polar to Rectangular Conversion	$z = re^{j\theta} = r [\cos(\theta) + j\sin(\theta)] = x + jy$ where $x = r \cos(\theta)$ and $y = r \sin(\theta)$
Add: $z_3 = z_1 + z_2$	$(x_1 + x_2) + j(y_1 + y_2)$
Subtract: $z_3 = z_1 - z_2$	$(x_1 - x_2) + j(y_1 - y_2)$
Multiply: $z_3 = z_1 z_2$ (polar form)	$(x_1 x_2 - y_1 y_2) + j(x_1 y_2 + y_1 x_2)$ $r_1 r_2 e^{j(\theta_1 + \theta_2)}$
Divide: $z_3 = z_1 / z_2$  (polar form)	$\frac{(x_1 x_2 - y_1 y_2) - j(x_1 y_2 - y_1 x_2)}{x_2^2 + y_2^2}$ $\frac{r_1}{r_2} e^{j(\theta_1 - \theta_2)}$

## Apêndice B:

Sejam  $u = 2\pi f_1 t$  e  $v = 2\pi f_2 t$ . Valem as seguintes relações (*relationships*) trigonométricas:

<i>Relationship</i>	<i>Relationship</i>
$\sin u = \cos(u - \pi/2)$	$\cos u = \sin(u + \pi/2)$
$\cos(-u) = \cos u$	$\sin(-u) = -\sin(u)$
$\sin^2 u + \cos^2 u = 1$	$\cos^2 u = \frac{1}{2}(1 + \cos 2u)$
$\sin^2 u = \frac{1}{2}(1 - \cos 2u)$	$\cos(u \pm v) = \cos u \cos v \mp \sin u \sin v$
$\sin(u \pm v) = \sin u \cos v \pm \cos u \sin v$	$\cos u \cos v = \frac{1}{2}[\cos(u - v) + \cos(u + v)]$
$\sin u \sin v = \frac{1}{2}[\cos(u - v) - \cos(u + v)]$	$\sin u \cos v = \frac{1}{2}[\sin(u - v) + \sin(u + v)]$
$\cos u = \frac{1}{2}[e^{ju} + e^{-ju}]$	$\sin u = \frac{1}{2j}[e^{ju} - e^{-ju}]$
$e^{ju} = \cos u + j \sin u$	

→ **heterodinação**  
das frequências  $f_1$  e  $f_2$

## Apêndice C:

<b>Name</b>	<b>Continuous</b>	<b>Discrete</b>
Impulse	$\delta(t) = 0, t \neq 0$ $\int_{-\infty}^{\infty} \delta(t) dt = 1$	$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & \text{otherwise} \end{cases}$
Step	$u(t) = \begin{cases} 1, & t > 0 \\ 0, & t < 0 \end{cases}$	$u[n] = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$
Rectangle Pulse	$\Pi\left(\frac{t}{\tau}\right) = \begin{cases} 1, & t \leq  \tau/2  \\ 0, & t >  \tau  \end{cases}$	$u[n] - u[n - M]$
Triangle Pulse	$\Lambda\left(\frac{t}{\tau}\right) = \begin{cases} 1 -  t/\tau , & t \leq  \tau  \\ 0, & t >  \tau  \end{cases}$	$\begin{cases} n + 1, & 0 \leq n \leq M - 1 \\ 2M - 1 - n, & M - 1 < n \leq 2M - 2 \end{cases}$
sinc( ) and aliased sinc( )	$\text{sinc}(t) = \frac{\sin \pi t}{\pi t}$	$\frac{\sin(N\pi x)}{\sin(\pi x)}, \text{ aliased sinc}()$
Sinusoid	$A \cos(2\pi f_0 t + \phi)$	$A \cos(\hat{\omega}_0 n + \phi), \hat{\omega} \text{ is mod } 2\pi$

## Apêndice D:

<b>Property</b>	<b>Continuous</b>	<b>Discrete</b>
<b>Energy</b>	$E_x = \int_{-\infty}^{\infty}  x(t) ^2 dt < \infty$	$E_x = \sum_{n=-\infty}^{\infty}  x[n] ^2$
<b>Power</b>	$P_x = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T  x(t) ^2 dt < \infty$	$P_x = \lim_{N \rightarrow \infty} \frac{1}{2N} \sum_{n=-\infty}^{\infty}  x[n] ^2 < \infty$
<b>Periodic</b>	$x(t - T_0) = x(t), T_0 = \text{period}$	$x[n - N_0] = x[n], N_0 = \text{period}$
<b>Even</b>	$x(-t) = x(t)$	$x[-n] = x[n]$
<b>Odd</b>	$x(-t) = -x(t)$	$x[-n] = -x[n]$