

Laboratório de Princípios de Comunicação

Aulas Remotas - 2020.1

Experimento 08

PCM e DPCM (Parte I)

Formatação de Pulsos e Codificação de Linha (Parte II)

Datas: 09 e 16/11/2020

Pré-requisitos

1. Criação de blocos de funções no GRC
2. Conceitos de interpolação e decimação com filtros digitais

Instruções gerais

- Organize os arquivos em um diretório específico para o laboratório em seu computador. Em caso de necessidade, os arquivos serão usados como evidência da participação em sala. Use nomes de arquivos do GRC com o formato

Matricula_ExpXX_ARYY.grc → Exemplo: 12345678_Exp08_AR01.grc

- O relatório será enviado em formato “.pdf” usando a nomenclatura

Matricula_ExpXX_Rel.pdf → Exemplo: 12345678_Exp08_Rel.pdf

- Quando solicitado, envie um arquivo compactado com o formato “.zip” com todos os arquivos relacionados ao experimento, contendo os arquivos “.grc” e “.pdf”.

Matricula_ExpXX.zip → Exemplo: 12345678_Exp08.zip

- Preferencialmente (mas não obrigatoriamente) use GUI do tipo “QT”.
- Apresente no início do relatório todos os cálculos teóricos solicitados ao longo do roteiro, e coloque as tabelas no final.

Atividade 01 – Pulse Code Modulation – Parte I

Conceitos preliminares

Na Parte I deste experimento, realizaremos uma versão simplificada de um sistema PCM, com modulação e demodulação. Um sinal analógico $m(t)$ será representado por suas amostras $m(kT_s)$, cada uma quantizada para produzir 8 bits. Os bits são posteriormente serializados e codificados em pulsos analógicos a serem transmitidos. No receptor, os pulsos recebidos serão amostrados, convertidos para uma sequência de bits e posteriormente representados por um número $\hat{m}(kT_s)$ que representa uma estimativa das amostras da mensagem. Veremos, ainda, o efeito da presença de um ruído AWGN nos bits recebidos.

No exemplo da Figura 1 temos os blocos básicos do experimento, com um *exemplo* de amostras quantizadas em 3 bits. A cada intervalo T_s é necessário transmitir 3 pulsos, cada um associado a um dos bits que representam a amostra (pontos B, C, D, E). A duração dos pulsos será de T_b , resultando em uma taxa de transmissão de $R_b = 1/T_b$ bits/s.

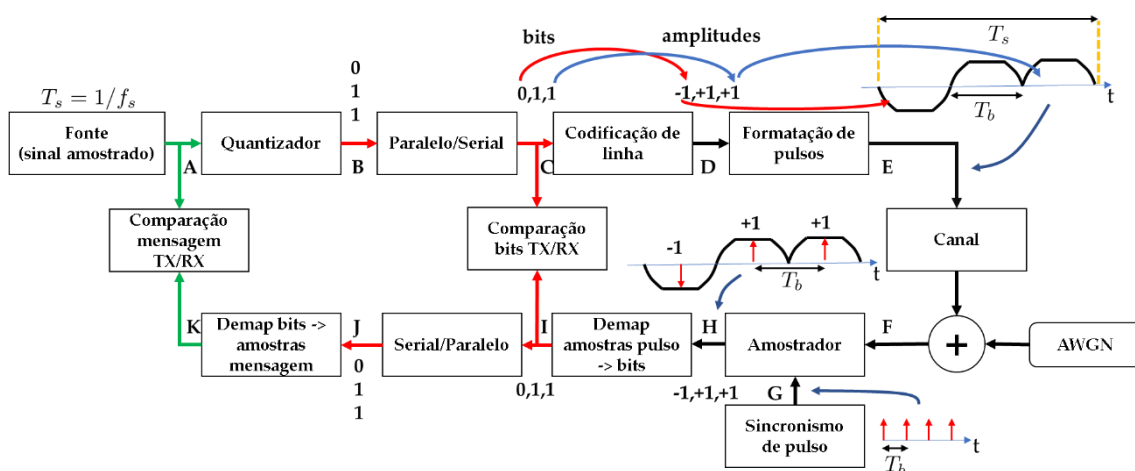


Figura 1 – Diagrama de blocos simplificado de sistema PCM.

A regra de mapeamento entre bits (0's e 1's) e amplitudes dos pulsos transmitidos é chamada de codificação (ou sinalização) de linha. No exemplo da figura, temos ($0 \rightarrow -1$ e $1 \rightarrow +1$), chamada de *codificação polar*. Um segundo exemplo (ON-OFF, usada em comunicações ópticas) está na Tabela 1.

Bits	POLAR		ON-OFF	
b_k	a_k	pulso	a_k	pulso
0	-1	$-p(t)$	0	0
1	+1	$+p(t)$	+1	$+p(t)$

Tabela 1 – Regras de mapeamento entre bits b_k e amplitudes de pulsos a_k . O índice k refere-se ao k -ésimo bit de informação.

A forma do pulso analógico $p(t)$ (quadrada, triangular, etc.) é definida pelo projetista de acordo com as características do meio de transmissão. Utilizaremos inicialmente dois formatos de pulso em **banda-base** de formato retangular, RZ e NRZ. O efeito detalhado da formatação de pulso será estudado na Parte II deste experimento.

Como estamos em um ambiente simulado, os pulsos analógicos serão aproximados por suas versões em tempo discreto. Em nosso caso, cada tempo de bit T_b será subdividido em 8 intervalos menores (em outras palavras, haverá uma sobreamostragem de fator $N = 8$). Os pulsos discretizados estão detalhados na Tabela 2, com a correspondente representação a ser usada em Python nos filtros interpoladores.

Pulso $p(t)$	Valores discretizados	Representação em Python
Retangular NRZ	[1, 1, 1, 1, 1, 1, 1, 1]	<code>8*[1]</code>
Retangular RZ	[1, 1, 1, 1, 0, 0, 0, 0]	<code>4*[1]+4*[0]</code>

Tabela 2 – Representação discretizada dos pulsos em banda-base.

No receptor, é realizada a amostragem dos pulsos a cada T_b e a conversão em bits (pontos **H**, **I**, **J** da Figura 1). O sinal recuperado no receptor pode então ser comparado ao sinal original. O erro entre os dois será denominado erro de reconstrução.

Atividade 02 – Modulação DPCM (Parte I)

Conceitos preliminares

Em alguns casos, é possível explorar as propriedades do sinal $m(t)$ para obter ganhos de qualidade ou de economia de largura de faixa. Na transmissão DPCM, transmitimos não as amostras $m(kT_s)$ de $m(t)$, mas a diferença entre $m(kT_s)$ e sua previsão $m_p(kT_s)$, esta última obtida por intermédio de um filtro preditor:

$$d(kT_s) = m(kT_s) - m_p(kT_s).$$

O sinal $d(kT_s)$ é então quantizado e codificado em pulsos como o PCM tradicional. Note que, se a previsão para $m(kT_s)$ for bem feita, o sinal $d(kT_s)$ terá amplitude pequena quando comparada à de $m(kT_s)$. Isto permite que seja possível usar menos bits para representá-lo (mantendo o mesmo intervalo de quantização) ou, mantendo a mesma quantidade de bits, o resíduo é expresso com maior precisão, o que equivale a representar o sinal mensagem com um maior número de bits. Nesta atividade, faremos um comparativo entre PCM e DPCM com base no erro de reconstrução.

A codificação em pulsos será eliminada para que o foco esteja no envio e recuperação do sinal diferencial. Trataremos os sinais como se fossem coletados nos pontos **B** e **J** do diagrama da Figura 1, abstraindo o processo de codificação de linha e formatação dos pulsos.

A Figura 2 [Couch 2013], apresenta um esquemático do transmissor e receptor DPCM.

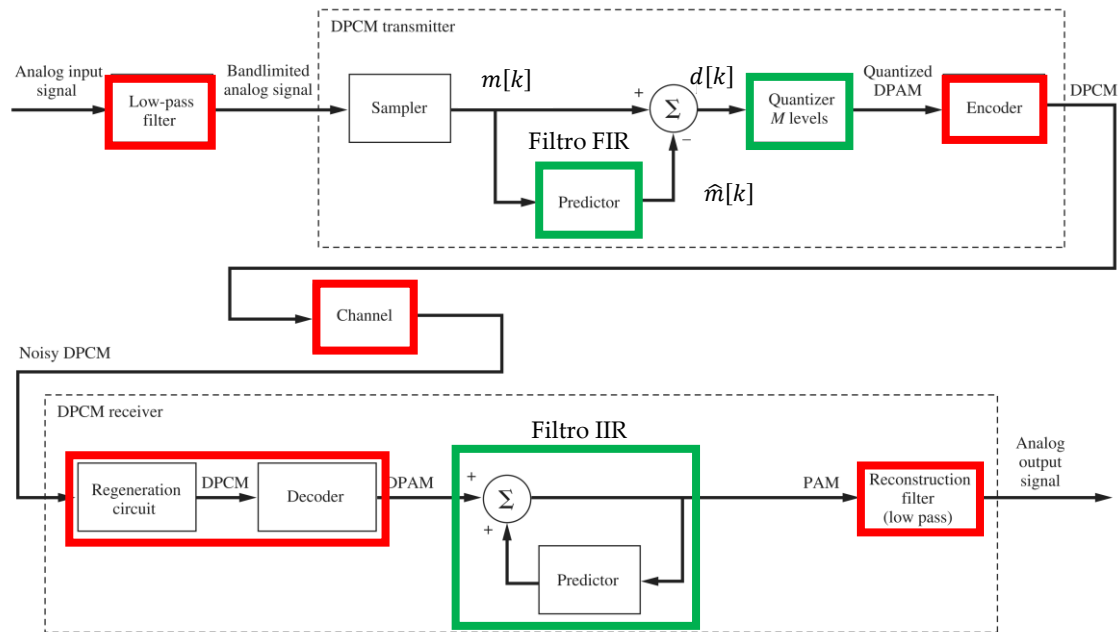


Figura 2 (Adaptado de Couch, 2013). Os blocos em vermelho não serão implementados.

O filtro a ser utilizado no transmissor DPCM é um preditor simples:

$$m_p(kT_s) = m_p[k] = 2m[k-1] - m[k-2],$$

correspondente a um filtro FIR com três coeficientes: 0, +2 e -1. Para o receptor, utilizaremos um filtro IIR que implementa naturalmente o preditor no laço de realimentação.

Atividade 03 – Codificação de Linha e Formatação de Pulsos (Parte II)

Conceitos preliminares

Nesta atividade, verificaremos o efeito da codificação de linha e formatação de pulsos na DEP.

É possível mostrar (Lathi, Cap. 7) que a DEP $S_p(f)$ de um trem de pulsos gerados por uma sequência de bits b_k depende do formato do pulso $p(t)$ e das amplitudes a_k associadas ao mapeamento entre bits e amplitudes (veja Tabela 1). De maneira geral,

$$S_p(f) = \frac{1}{T_b} |P(f)|^2 \sum_n R_n e^{-j2\pi f n T_b},$$

onde os coeficientes $R_n = E[(a_k \cdot a_{k+n})^2]$ dependem da função de autocorrelação da sequência de amplitudes $\{a_k\}$ e $P(f)$ é a Transformada de Fourier do pulso $p(t)$. Portanto, é possível alterar a DEP do trem de pulsos alterando não apenas o seu formato, mas também as propriedades de correlação entre os bits.

Levando em conta que os pulsos transmitidos $p(t)$ são amostrados no receptor a cada kT_b , é possível utilizar no instante k pulsos com duração superior a T_b , desde que não haja interferência nos instantes $k+1$, $k+2$, etc. Esta restrição é chamada de *primeiro critério*

de Nyquist para Interferência entre Símbolos (IES) nula, e os pulsos que satisfaçam este critério à vezes são chamados de pulsos de Nyquist.

Uma família de pulsos Nyquist são os pulsos chamados de **cosseno levantado**, amplamente usados em comunicações digitais. No domínio da frequência, o critério para IES nula se reflete no formato em frequência do pulso, que deve ter simetria vestigial no entorno de $R_b/2$. Um parâmetro relevante desta família de pulsos é o chamado **fator de roll-off** (r) que controla a largura de banda B_T do pulso. A relação é dada por

$$B_T = \frac{R_b}{2}(1 + r) \quad (1)$$

Para cada valor r , há um espectro distinto, com resposta ao impulso distinta. A Figura 3 ilustra o conceito.

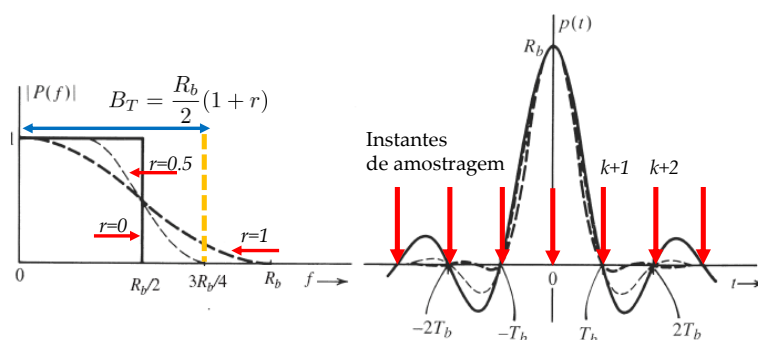


Figura 3 (Adaptado de Lathi, Cap. 7). Pulsos cosseno levantado: espectro (esquerda) e resposta no tempo (direita).

No ambiente GRC, usaremos um filtro FIR cuja resposta ao impulso seja uma versão discretizada no tempo de filtros cosseno levantado. A resposta será truncada no tempo em 5 símbolos (pulsos) à esquerda e à direita (antecessores e sucessores). Entre duas amostragens do pulso recebido, usaremos 8 valores de $p(t)$ originalmente contínuo. Finalmente, a resposta será atrasada no tempo, de modo a permitir causalidade. Os valores dos coeficientes estão no Anexo.

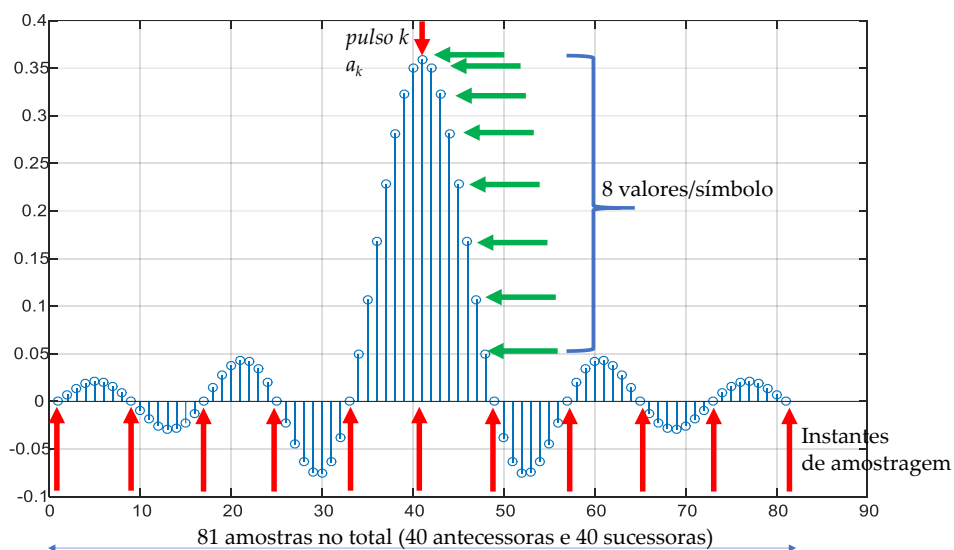


Figura 4 – Exemplo de pulso cosseno levantado a ser usado no GRC.

Relatório Parte I– Entregar até 15/10/2020

Faça um relatório respondendo cada item e imagem solicitada.

Apresente no início do relatório todos os cálculos e respostas a questões teóricas solicitadas e as tabelas no final.

AR 01 – PCM

Setup da simulação AR01

Taxa de amostragem	10 ksps (de referência)
Instrumentos virtuais	Osciloscópio Analisador de Espectro com N = 2048 pontos Histograma <QT GUI Histogram Sink> Gerador de ruído <Noise Source > <Noise Type = Gaussian> <Output Type = Float> Amplitude variável
Observações adicionais	<ul style="list-style-type: none"> - Use como sinal mensagem a forma de onda que desejar com amostras codificadas em 8 bits, respeitando a taxa de amostragem dada e normalização adequada para o quantizador - Utilizar bloco “Throttle” - Inserir bloco <import> → na Caixa de Configuração <Import>, digitar “import numpy as np” - Filtros interpoladores elevam a taxa de amostragem inserindo novas amostras; filtros decimadores reduzem a taxa de amostragem descartando amostras. Os instrumentos de medição devem ser configurados com a taxa de amostragem correta. - Ver configuração do bloco “Decisor: SGN(n)” no anexo B.

Como sugestão, adote um painel de controle similar ao mostrado na Figura 4.

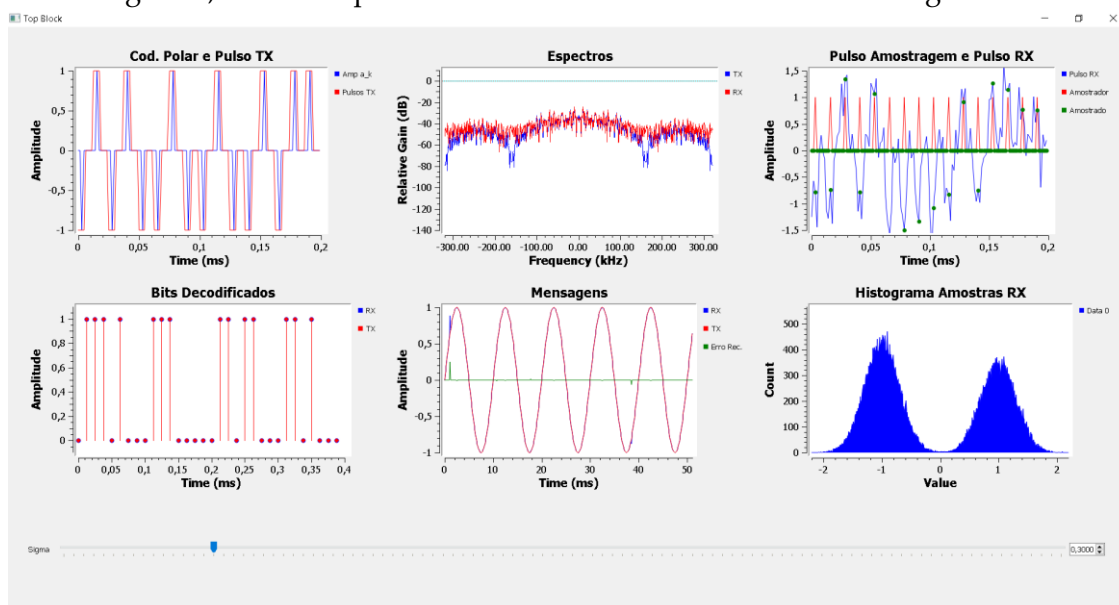


Figura 4 – Sugestão para realização das medições de PCM.

Implemente o diagrama de blocos da Figura 1, tendo como base a Figura 5. Faça um *print* (ou vários) da tela (**Fig. 1.1**). Certifique-se de entender o funcionamento dos blocos *<Char to Float>* e *<Float to Char>*, necessários para serialização dos bits. Os blocos em destaque “Sincronismo de Pulso” e “Amostrador” foram criados para simplificar o processo de amostragem no receptor.

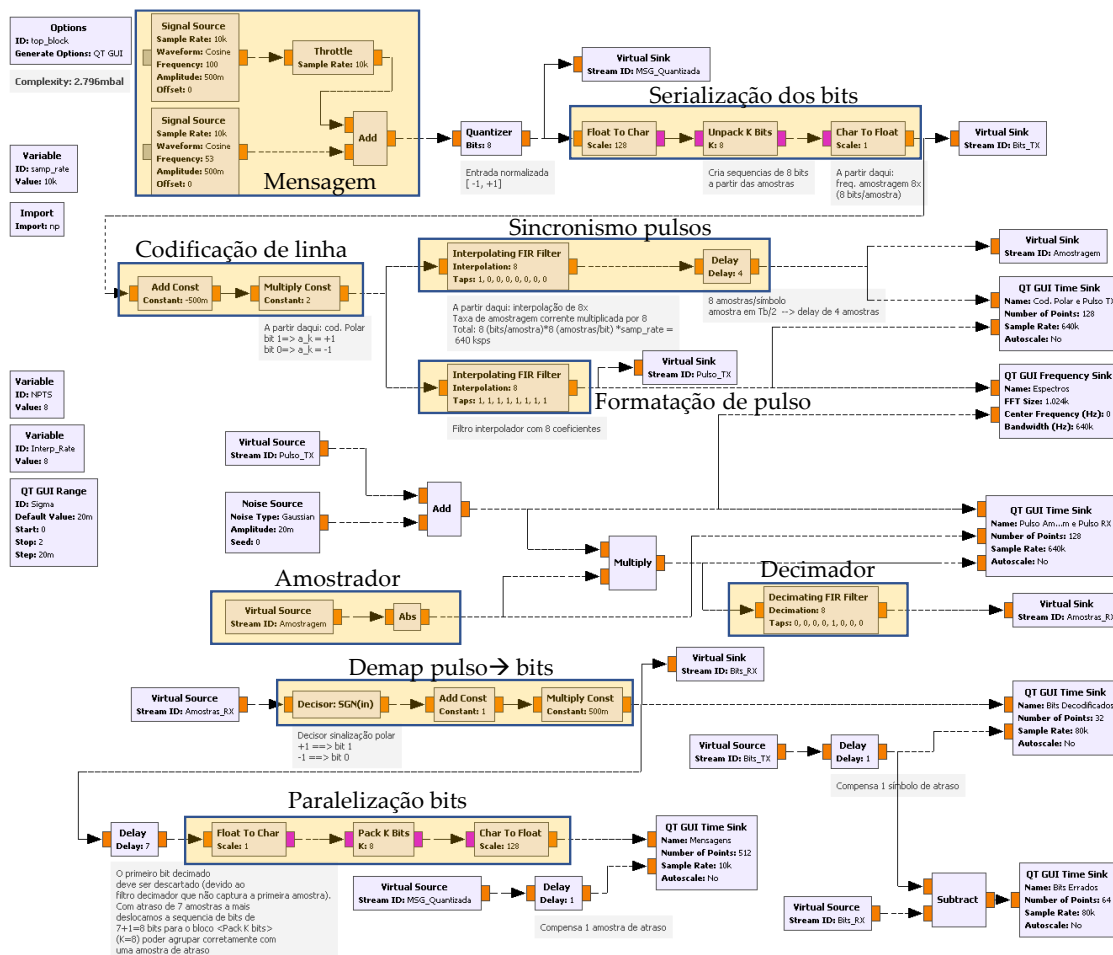


Figura 5 – Blocos para implementação da Figura 1. *São necessárias adaptações.*

a) Características do trem de pulsos transmitido

Configure o sistema PCM para codificação de linha POLAR (veja Tabela 1). Nesta situação, altere o formato do pulso de transmissão $p(t)$ mudando os coeficientes do filtro interpolador no bloco “Formatação de Pulso” de acordo com os casos da Tabela A.1. Para cada caso, mostre a forma de onda (Fig. 1.2a-d) e o espectro (Fig. 1.3a-d) e meça a largura de banda B_N até o primeiro nulo da DEP e a diferença (em dB) entre o pico máximo da DEP e o pico do primeiro lóbulo lateral (NLL) e preencha a **Tabela 1.1**. Preencha os valores teóricos considerando a transformada de Fourier de um pulso individual.

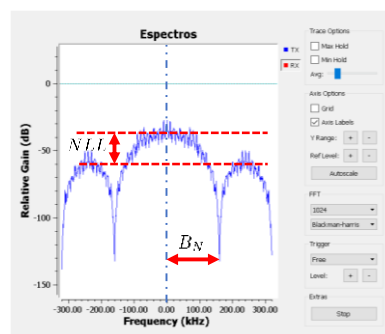


Tabela A.1. (*) Alterar Delay de 4 para 2 no bloco “Sincronismo pulsos”.

Pulso	Coefficientes
(1) Retangular NRZ	$8*[1.0]$
(2) Retangular RZ*	$4*[1.0]+4*[0.0]$
(3) Triangular	$\text{np.array}([0,1,2,3,4,3,2,1])/4.0$
(4) Hanning	$\text{np.hanning}(8)/0.95048443$

Com base nas medições, responda:

A1a) Qual a relação entre B_N e a taxa de transmissão de dados R_b ?

A1b) Qual dos pulsos apresenta menor B_N ?

A1c) Qual dos pulsos apresenta menor NLL?

A1d) Suponha a existência de um canal de transmissão com resposta em frequência tal que frequências acima de 120 kHz sejam eliminadas. Entre os pulsos (2), (3) e (4), qual você escolheria para formatar os pulsos de transmissão? Justifique.

b) Formas de onda e sequencias no tempo

Use pulsos retangulares NRZ polares, contaminados por ruído AWGN ($\langle \text{Amplitude} \rangle = 0.1$ V). Apresente os símbolos a_k e trem de pulsos de transmissão (Tx) na **Fig. 1.3a**, a forma de onda dos pulsos no receptor (Rx), os impulsos de amostragem e as amostra na **Fig. 1.3b**, os bits em Tx e em Rx (**Fig. 1.3c**), e as formas de onda da mensagem (**Fig. 1.3d**) original, recuperada e diferença entre elas (erro). Se quiser, você poderá apresentar todas numa imagem conjunta como na Figura 4, identificando cada uma adequadamente (Fig. x.yz).

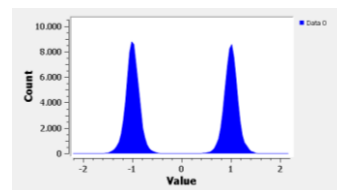
c) Processo de amostragem no receptor

No bloco “Sincronismo Pulsos” da Figura 5, altere o atraso “Delay” para o pulso “Retangular RZ” de 2 para 4. Deixe o valor do desvio padrão ($\langle \text{Amplitude} \rangle$) do AWGN em 0.1 V e observe o sinal decodificado.

A1e) Explique, ilustrando com as formas de onda dos pulsos contaminados (**Fig. 1.4a**), da sequência de bits (**Fig. 1.4b**) transmitida (Tx) e recebida (Rx), e o histograma (**Fig. 1.4c**) dos pulsos amostrados (opção <Accumulate >, ao menos 5000 contagens para os picos).

d) Efeito do ruído AWGN e critério de decisão

Utilize pulsos retangulares NRZ. Altere o desvio padrão ($\langle \text{Amplitude} \rangle$) do AWGN para os valores [0.2, 0.5 e 1.0] V. Apresente os correspondentes histogramas (**Fig. 1.4a-c**) e responda:



A1f) No receptor, relacione o critério de decisão (utilizado para converter valores amostras de pulsos para em bits) com o histograma obtido. Quando os bits são decodificados de maneira errada?

A1g) Quando utilizamos o receptor deste experimento, o formato do pulso é relevante para a recuperação dos bits? Justifique.

A1h) Por que, às vezes, o efeito do ruído é mais severo no sinal recuperado?

AR 02 – DPCM

Nesta atividade verificaremos as possíveis vantagens do uso do DPCM em qualidade ou redução da taxa de transmissão. Utilize as Figuras 6 e 7 como base para o setup experimental.

Setup da simulação AR02

Taxa de amostragem	10 ksp/s
Instrumentos virtuais	Osciloscópios QT GUI Constellation sink
Observações adicionais	Utilizar bloco “Throttle” O filtro IIR pode apresentar alguma instabilidade quando o ajuste dos parâmetros nos blocos <QT GUI Range> é feito de forma dinâmica. Estabilize os ajustes, anote os valores de K e do atraso e então meça as RSRs.

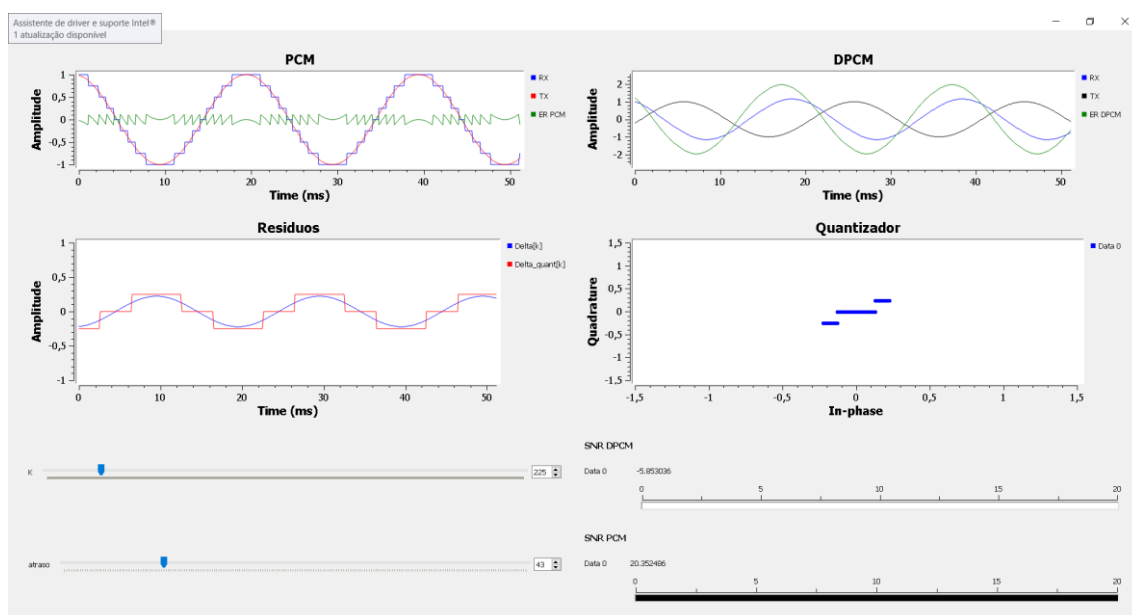


Figura 6 – Sugestão para medições para avaliação de DPCM.

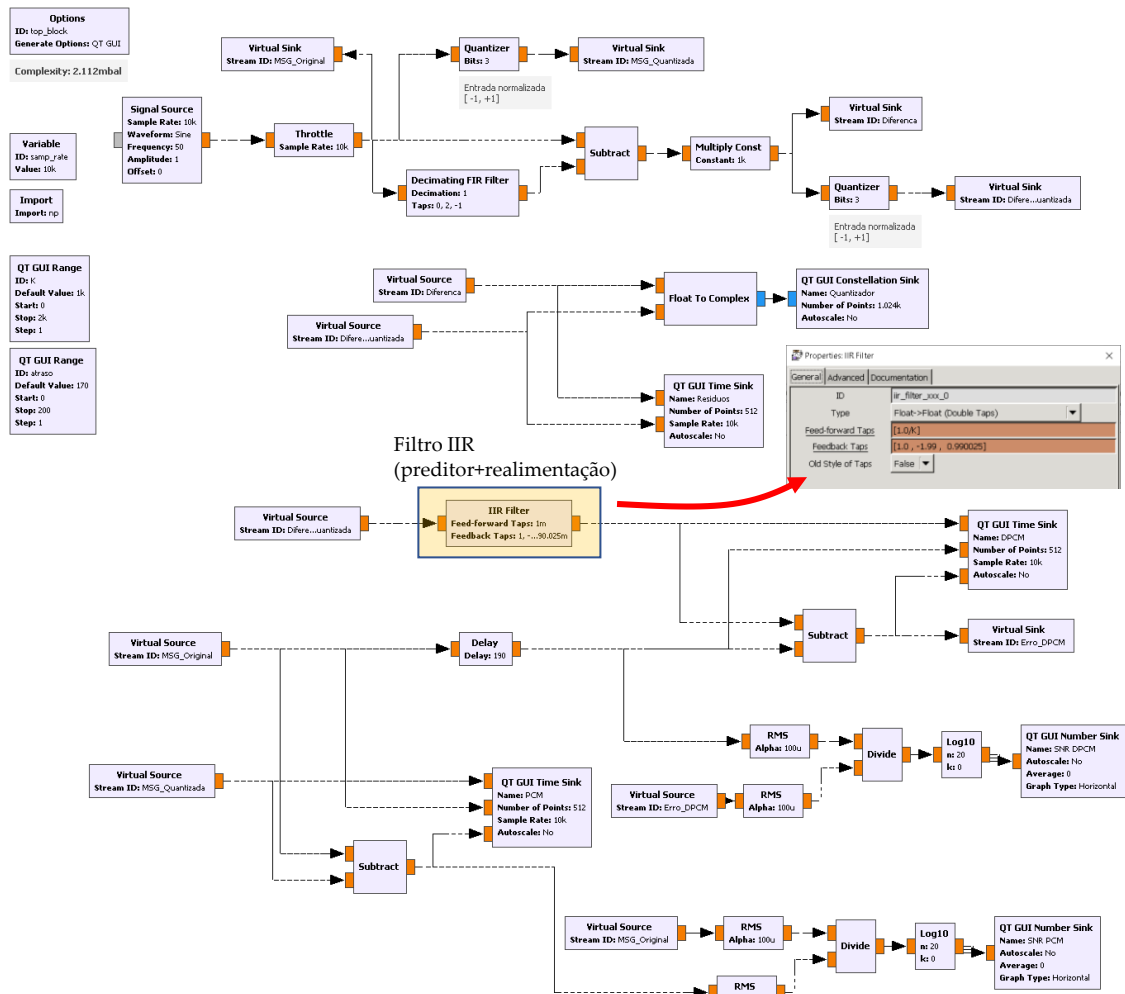


Figura 7 – Setup para DPCM.

a) Uso do DPCM para ganho de qualidade na recepção.

Faça um ou vários *prints* de seu diagrama (Fig. 2.1). Realize uma quantização linear com 3 bits (8 níveis) para a mensagem e meça a RSR de quantização, RSR_{PCM} , e preencha na Tabela 2.1.

Ajuste o ganho K do bloco <Multiply Const> no transmissor, de modo que os resíduos sejam quantizados em 3 bits. Comprove o ajuste apresentando a curva de quantização correspondente (Fig. 2.2) com o auxílio do bloco <QT GUI Constellation Sink> (ver Figura 5).

Apresente num mesmo gráfico as formas de onda original e recuperada no receptor PCM e DPCM (Fig. 2.3);

No receptor, o bloco <Delay> permite que seja possível medir o erro de reconstrução entre o sinal recuperado (saída do filtro IIR) e a mensagem original. Ajuste o atraso do bloco para que o valor do erro seja mínimo.

Preencha na **Tabela 2.1**:

- O valor de K que permite o uso de 8 níveis de quantização no transmissor DPCM;
- O valor do atraso no receptor;
- O valor da RSR_{DPCM} (dB) usando como medida de ruído o erro de reconstrução no receptor.

A2a) Compare as RSRs para as duas técnicas e comente.

b) Uso do DPCM para redução de taxa de transmissão.

Ajuste a quantidade de bits no PCM para que as duas RSR sejam similares.

A2b) Fixada a taxa de amostragem no transmissor em 10 ksps, quais as taxas de transmissão R_b nos dois casos?

Relatório Parte II– Entregar até 22/10/2020

AR 03 – Codificação de Linha e Formatação de Pulsos

Responda as questões teóricas no final do roteiro.

Utilize o diagrama de blocos da Figuras 5 (PCM) como base para o procedimento experimental. Todos os pulsos terão 8 amostras/símbolo.

Prepare o ambiente (arquivos .grc distintos) para:

- Permitir duas codificações de linha: POLAR (já realizado) e ON-OFF. Para sinalização ON-OFF, altere os blocos “Codificação de Linha” e “Decisor”.

A3a) Apresente e justifique no relatório as alterações necessárias.

- Inserir os coeficientes dos filtros interpoladores (bloco “Formatação de Pulso”) com coeficientes C1, C2 e C3 do Anexo.

Apresente os espectros (**Fig. 3.1a-g**) para os seguintes cenários da Tabela A.3. Preencha a **Tabela 3.1** medindo a largura de banda a partir da DEP. Critério: valor de frequência com atenuação de 40 dB do valor máximo da DEP.

Cenário	Pulso	Sinalização
1	Retangular NRZ	ON-OFF
2	Retangular NRZ	Polar
3	Retangular RZ	ON-OFF
4	Retangular RZ	Polar
5	C1	Polar
6	C2	Polar
7	C3	Polar

Tabela A.3 – Cenários utilizados na Parte II.

Sem adicionar ruído, responda e justifique no relatório:

A3b) Qual a principal mudança na DEP comparando os cenários 1 e 2?

A3c) Compare os cenários 1 e 3. Quais as alterações observadas?

A3d) Compare os cenários 3 e 4. Em qual dos casos é possível usar os pulsos recebidos como entrada bloco “Sincronismo de Pulso” da Figura 1? A mesma estratégia pode ser usada nos cenários 1 e 2?

A3e) Encontre o fator de *roll-off* para os cenários 5, 6 e 7. Observando o comportamento dos pulsos entre amostras, é possível

A3f) Verifique qual pulso tem menor fator de *roll-off* observando os pulsos transmitidos no tempo. Para ajudar, implemente um diagrama de olho para observação dos pulsos transmitidos no osciloscópio em uma janela de tempo de 2 símbolos. Como sugestão, use a estratégia da Figura 8.

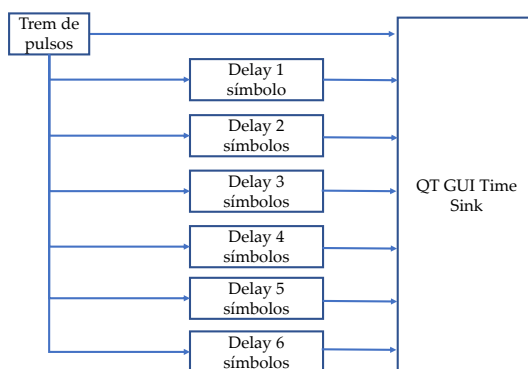


Figura 8 – Diagrama de olho no GRC.

Altere o desvio padrão (<Amplitude>) do AWGN para o valor 0.5 V.

A3g) Comparando os cenários 1 e 2, em qual deles houve maior efeito de ruído? Justifique, com base no histograma dos pulsos recebidos e amostrados.

A3h) Compare os cenários 4 e 5. O que acontece se o pulso for amostrado em um instante de tempo diferente? Em qual dos dois casos o efeito do erro de amostragem foi mais severo?

Apêndice

A - Coeficientes dos filtros cosseno levantado.

Copiar e colar no GRC.

#c1

```
np.array([0.0000, 0.0001, 0.0002, 0.0001, -0.0000, -0.0001, -0.0002, -
0.0002, 0.0000, 0.0002, 0.0003, 0.0002, -0.0000, -0.0003, -0.0005, -
0.0004, 0.0000, 0.0005, 0.0008, 0.0007, -0.0000, -0.0009, -0.0015, -
0.0013, 0.0000, 0.0019, 0.0033, 0.0030, -0.0000, -0.0051, -0.0099, -
0.0101, 0.0000, 0.0255, 0.0693, 0.1307, 0.2041, 0.2800, 0.3465, 0.3921,
0.4082, 0.3921, 0.3465, 0.2800, 0.2041, 0.1307, 0.0693, 0.0255, 0.0000,
-0.0101, -0.0099, -0.0051, -0.0000, 0.0030, 0.0033, 0.0019, 0.0000, -
0.0013, -0.0015, -0.0009, -0.0000, 0.0007, 0.0008, 0.0005, 0.0000, -
0.0004, -0.0005, -0.0003, -0.0000, 0.0002, 0.0003, 0.0002, 0.0000, -
0.0002, -0.0002, -0.0001, -0.0000, 0.0001, 0.0002, 0.0001,
0.0000])/0.4082
```

#c2

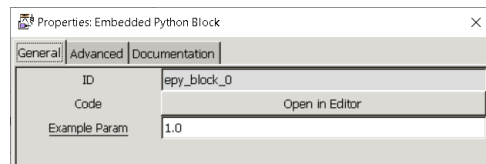
```
np.array([0.0000, -0.0001, -0.0003, -0.0007, -0.0010, -0.0012, -
0.0011, -0.0007, 0.0000, 0.0008, 0.0016, 0.0021, 0.0022,
0.0018, 0.0010, 0.0003, 0.0000, 0.0004, 0.0018, 0.0040,
0.0065, 0.0084, 0.0086, 0.0060, -0.0000, -0.0096, -0.0218, -
0.0347, -0.0454, -0.0504, -0.0463, -0.0301, 0.0000, 0.0438,
0.0992, 0.1621, 0.2269, 0.2868, 0.3353, 0.3670, 0.3780,
0.3670, 0.3353, 0.2868, 0.2269, 0.1621, 0.0992, 0.0438,
0.0000, -0.0301, -0.0463, -0.0504, -0.0454, -0.0347, -0.0218, -
0.0096, -0.0000, 0.0060, 0.0086, 0.0084, 0.0065, 0.0040,
0.0018, 0.0004, 0.0000, 0.0003, 0.0010, 0.0018, 0.0022,
0.0021, 0.0016, 0.0008, 0.0000, -0.0007, -0.0011, -0.0012, -
0.0010, -0.0007, -0.0003, -0.0001, 0.0000])/0.3780
```

#c3

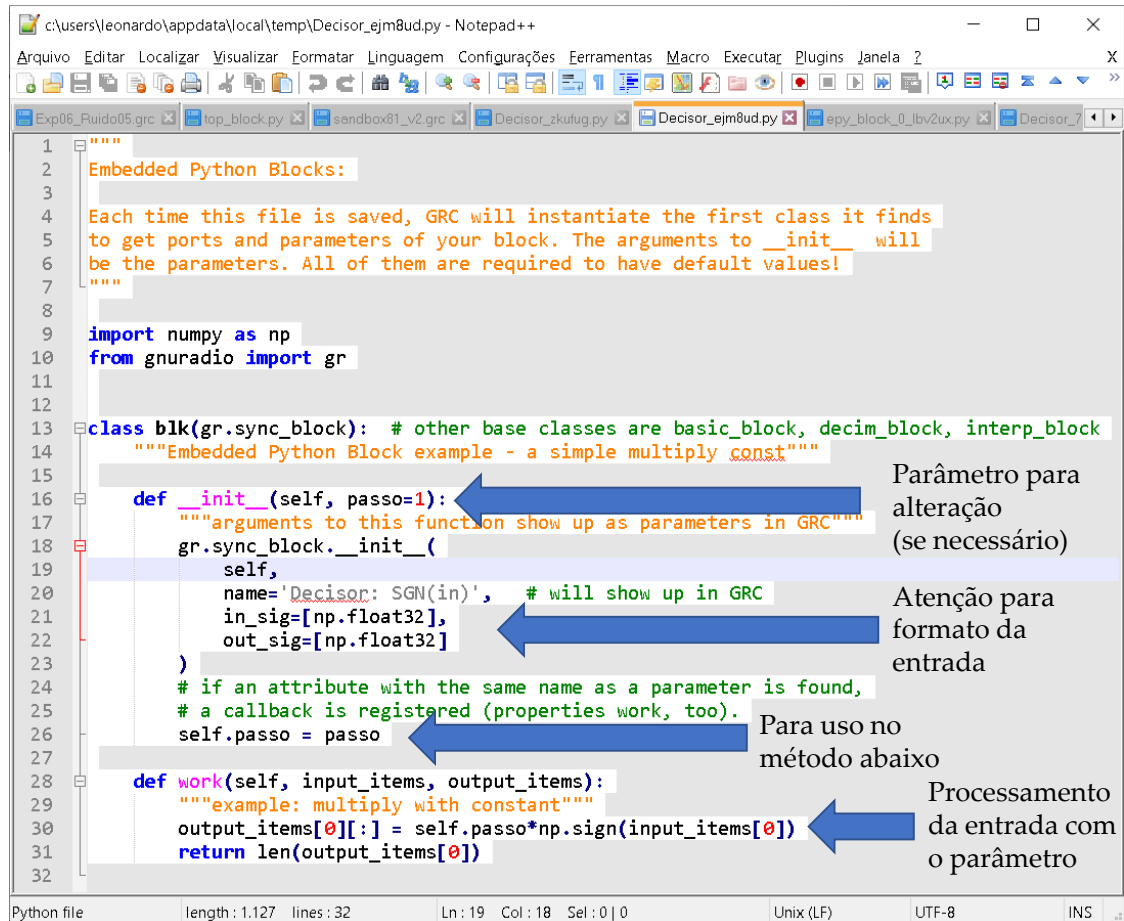
```
np.array([0.0000, 0.0071, 0.0137, 0.0186, 0.0209, 0.0201,
0.0160, 0.0090, -0.0000, -0.0098, -0.0188, -0.0257, -0.0290, -
0.0281, -0.0225, -0.0128, 0.0000, 0.0141, 0.0273, 0.0377,
0.0431, 0.0421, 0.0342, 0.0197, -0.0000, -0.0226, -0.0448, -
0.0633, -0.0746, -0.0754, -0.0637, -0.0384, 0.0000, 0.0496,
0.1071, 0.1682, 0.2279, 0.2810, 0.3229, 0.3496, 0.3588,
0.3496, 0.3229, 0.2810, 0.2279, 0.1682, 0.1071, 0.0496,
0.0000, -0.0384, -0.0637, -0.0754, -0.0746, -0.0633, -0.0448, -
0.0226, -0.0000, 0.0197, 0.0342, 0.0421, 0.0431, 0.0377,
0.0273, 0.0141, 0.0000, -0.0128, -0.0225, -0.0281, -0.0290, -
0.0257, -0.0188, -0.0098, -0.0000, 0.0090, 0.0160, 0.0201,
0.0209, 0.0186, 0.0137, 0.0071, 0.0000])/0.3588
```

B – Como alterar um bloco básico no GRC.

No Menu de blocos, busque por <Python Block>. Clique no bloco para editar suas propriedades com um editor de texto simples (use, por exemplo, Notepad no Windows).



Altere o conteúdo para realizar um processamento do sinal de entrada do bloco utilizando um parâmetro. Exemplo para a função sinal (SGN):



```

1  """
2  Embedded Python Blocks:
3
4  Each time this file is saved, GRC will instantiate the first class it finds
5  to get ports and parameters of your block. The arguments to __init__ will
6  be the parameters. All of them are required to have default values!
7  """
8
9  import numpy as np
10 from gnuradio import gr
11
12 class blk(gr.sync_block): # other base classes are basic_block, decim_block, interp_block
13     """Embedded Python Block example - a simple multiply const"""
14
15     def __init__(self, passo=1):
16         """arguments to this function show up as parameters in GRC"""
17         gr.sync_block.__init__(
18             self,
19             name='Decisor: SGN(in)', # will show up in GRC
20             in_sig=[np.float32],
21             out_sig=[np.float32]
22         )
23         # if an attribute with the same name as a parameter is found,
24         # a callback is registered (properties work, too).
25         self.passo = passo
26
27     def work(self, input_items, output_items):
28         """example: multiply with constant"""
29         output_items[0][:] = self.passo*np.sign(input_items[0])
30         return len(output_items[0])
31
32 """
  
```

Parâmetro para alteração (se necessário)

Atenção para formato da entrada

Para uso no método abaixo

Processamento da entrada com o parâmetro

Python file length: 1.127 lines: 32 Ln: 19 Col: 18 Sel: 0 | 0 Unix (LF) UTF-8 INS

Experimento 8: PCM e DPCM (Parte I)

Identificação

Turma	Matrícula	Nome

Tabela 1.1 – Largura de banda e nível¹ de lóbulo vizinho para pulsos polares

Pulso	B_N (kHz)		NLL (dB)	
	teórico	Medido	teórico	Medido
(1) Retangular NRZ				
(2) Retangular RZ*				
(3) Triangular				
(4) Hanning	---		---	

Tabela 2.1 – Parâmetros e medidas PCM/DPCM

RSR_{PCM} (dB)	RSR_{DPCM} (dB)	K	atraso (em amostras)

¹ Para os cálculos teóricos, considere a TF de um pulso individual.

Experimento 8: Formatação de Pulsos e Codificação de Linha (ParteII)

Identificação

Turma	Matrícula	Nome

Tabela 3.1 – Largura de banda para diversos sinais digitais.

Cenário	Pulso	Sinalização	B_T (kHz)
1	Retangular NRZ	ON-OFF	
2	Retangular NRZ	Polar	
3	Retangular RZ	ON-OFF	
4	Retangular RZ	Polar	
5	C1	Polar	
6	C2	Polar	
7	C3	Polar	

Questões teóricas

Considere o sinal digital banda básica $c(t) = \sum_{k=-\infty}^{\infty} a_k p(t - nT_b)$ com pulsos codificados de acordo com o valor do símbolo a_k , em que $p(t)$ dá forma aos pulsos. Calcule a largura de banda² para os seguintes sinais **em função da taxa de bits R_b** .

T3a) $c(t)$ composto por pulsos triangulares NRZ de sinalização polar;

T3b) $c(t)$ composto por pulsos retangulares RZ de sinalização polar;

T3c) $c(t)$, em que o pulso $p(t)$ possua espectro retangular entre $-B$ e B , e a_k tenha sinalização polar;

T3d) $c(t)\cos(\omega_c t)$, em que o pulso $p(t)$ possua espectro vestigial (cosseno levantado) com fator de roll-off $r = 0.5$, e a_k tenha sinalização polar;

T3e) $A_c \cos(2\pi f_c t + k_\omega \int c(\tau) d\tau)$, em que $k_\omega = 2\pi/T_b$, $c(t)$ varie entre -1 e 1, e o pulso $p(t)$ possua espectro vestigial com fator de roll-off de 1.

² Considere como critério de banda para os sinais não modulados (banda básica) o seguinte: o primeiro nulo do espectro para pulsos retangulares e triangulares, e a expressão $B_T = R_b(r + 1)/2$ para pulsos de espectro vestigial.