

# Relatório de Projeto

## Simulink

# Contents

## Articles

Introdução	<b>1</b>
INCENTIVANDO O USO DO SIMULINK NOS CURSOS DE TELECOMUNICAÇÕES	1
Modulação de Pulso	<b>3</b>
Modulações Analógicas	3
Amostragem de sinais	5
Quantização	8
Modulação por Codificação de Pulso (PCM)	10
Modulação Delta (DM)	13
Modulação Delta-Sigma	15
Modulação por Codificação Diferencial de Pulso(DPCM)	17
Transmissão de Pulsos em Banda Base	<b>20</b>
Códigos de linha	20
Modulação por Amplitude de Pulso M-ária (PAM-M)	22
Transmissão digital em banda base	24
Transmissão de Dados em Banda Passante	<b>27</b>
Modulações digitais binárias	27
Modulações digitais M-árias	29
Passband Modulation	31
Multipath Fading Channel	35
Digital Communications Ber Performance in AWGN or Fading	37
Orthogonal Frequency Multiplex Division (OFDM)	40
References	
Article Sources and Contributors	42
Image Sources, Licenses and Contributors	43
Article Licenses	
Licença	44

---

# Introdução

---

## INCENTIVANDO O USO DO SIMULINK NOS CURSOS DE TELECOMUNICAÇÕES

---

### Introdução e Justificativa da Proposição

Simulink® é um ambiente de diagramas de blocos para a simulação de projetos baseados em modelos. Além da simulação este software suporta a geração automática de código para aplicações em software e hardware (HDL), teste e verificação de sistemas (MATHWORKS, 2015a, 2015c). O editor gráfico possui uma biblioteca de componentes os quais são conectados para a modelagem de sistemas dinâmicos. Ele é integrado com o MATLAB®, permitindo incorporar algoritmos desenvolvidos nesta linguagem de programação para a criação de modelos e simulação. Em virtude destas características esse software tem um alto potencial tanto para modelagem, simulação e implementação e avaliação de sistemas na área de telecomunicações principalmente sistemas de comunicações, processamento de sinais digitais, processamento de imagem e vídeo, sistemas embarcados entre outros (AMIR, 2000; GAN; KUO, 2006). O campus São José do IFSC adquiriu 10 licenças deste software as quais estão disponibilizadas nos laboratórios físicos e também de forma virtual através da IFSC-CLOUD (TORRESINI, 2015), na qual os professores e alunos podem acessar as licenças do IFSC mesmo usando máquinas que estão fora do domínio de endereços do IFSC. Neste projeto pretende-se dar uma maior ênfase no uso dessa ferramenta para a melhoria do processo ensino-aprendizagem, principalmente em disciplinas que lidam com sistemas de maior complexidade. Para este projeto inicial foi escolhida a disciplina de PSD29007 em função da grande quantidade de aplicações que já existem para o ambiente Simulink® (KRUKOWSKI; KALE, 2011) e também pela necessidade de implantar a disciplina no curso de Engenharia de Telecomunicações (AMIR, 2000).

### Objetivos do Projeto

- Estudar o Simulink®.
- Analisar de aplicações disponibilizadas por outras universidades e institutos de pesquisa.
- Desenvolver aplicações para uso nas aulas de laboratório do curso de Engenharia de Telecomunicações utilizando o software Simulink®
- Ampliar o uso do software de desenvolvimento Simulink® nas disciplinas dos cursos da área de telecomunicações.

### Metodologia do Projeto

As metas previstas para a execução do projeto são as seguintes:

1. ) Estudo do Simulink® incluindo a sua configuração, interfaces, sinais, blocos, modelagem e simulação de sistemas, geração de código de software e hardware. Esta etapa será baseada na leitura de manuais e também sites de ajuda do próprio fabricante (MATHWORKS, 2015a, 2015c, 2015b).
  2. ) Análise de aplicações disponibilizadas por outras universidades e institutos de pesquisa (NAVAS, 2014; TYAGI, 2012; GIORDANO; LEVESQUE, 2015; KRUKOWSKI; KALE, 2011). Nesta etapa serão catalogadas as aplicações já disponíveis e testadas, avaliando a pertinência aos cursos da área de telecomunicações e principalmente a disciplina de DSP29007 – Processamento de Sinais Digitais do curso de Engenharia de Telecomunicações.
  3. ) Desenvolvimento de aplicações para uso nas aulas de laboratório da disciplina de PSD29007.
  4. ) Disponibilização das aplicações avaliadas e também as desenvolvidas em site (Wiki) para uso nas aulas práticas de laboratório (NEJKOVIC; TOSIC, 2014).
-

5. ) Escrita de relatórios parciais e do relatório final do projeto. Preparação de um pôster para apresentação em eventos do IFSC.

Alunos envolvidos

- Rogério Pereira Junior

Professores Orientadores do projeto

- prof. Marcos Moecke

Edital de origem do projeto

- EDITAL Nº 13/2015/PROPPI <sup>[1]</sup>

## Tutoriais para uso do SIMULINK

- Introduction to Simulink <sup>[2]</sup>
- Build and simulate a model with Simulink® <sup>[3]</sup>
- Use MATLAB Function block to incorporate MATLAB® code in a Simulink® model <sup>[4]</sup>

## Tutoriais para experimentos com SIMULINK

- Digital Video Broadcasting - Terrestrial - da Mathworks
- Modulações Analógicas (AM,FM,PM)
- Multipath Fading Channel [5]
- Passband Modulation [6]
- Digital Communications Ber Performance in AWGN or Fading

## References

- [1] [http://www.ifsc.edu.br/arquivos/pesquisa/coord\\_pesq/editais2015/edital%2013/edital13\\_proppi\\_dp\\_sj\\_2015.pdf](http://www.ifsc.edu.br/arquivos/pesquisa/coord_pesq/editais2015/edital%2013/edital13_proppi_dp_sj_2015.pdf)
- [2] [http://www.mathworks.com/videos/introduction-to-simulink-81623.html?form\\_seq=conf840&elqsid=1430251491350&potential\\_use=Education&country\\_code=BR](http://www.mathworks.com/videos/introduction-to-simulink-81623.html?form_seq=conf840&elqsid=1430251491350&potential_use=Education&country_code=BR)
- [3] [http://www.mathworks.com/support/2015a/simulink/8.5/demos/sl\\_env\\_intro\\_web.html](http://www.mathworks.com/support/2015a/simulink/8.5/demos/sl_env_intro_web.html)
- [4] <http://www.mathworks.com/videos/incorporating-matlab-algorithms-into-a-simulink-model-69028.html?type=shadow>
- [5] <https://www.mathworks.com/examples/simulink-communications/2800-multipath-fading-channel>
- [6] <https://www.mathworks.com/examples/simulink-communications/2826-passband-modulation>
-

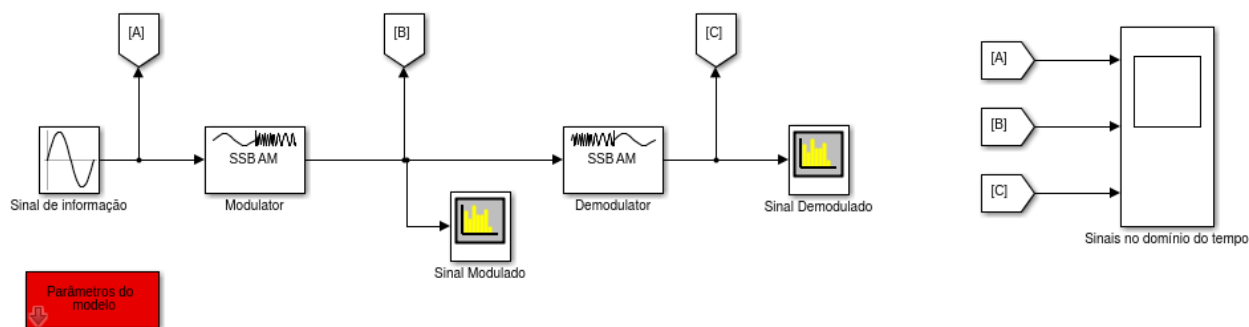
# Modulação de Pulso

## Modulações Analógicas

### Modulações Analógicas - Simulink

Este modelo simula a transmissão de sinais utilizando modulações analógicas do Communications System Toolbox™ fornecido pelo próprio Simulink™. Os esquemas de modulação de portadora analógica foram os primeiros a serem implementados. As portadoras sendo sinusoidais, podem fazer-se variar em amplitude, em fase, ou em frequência. Neste modelo apresentamos as modulações AM (Amplitude modulation), FM (Frequency modulation) e PM (Phase modulation).

### Modulações Analógicas



Primeiramente baixe o arquivo a seguir Sistema.zip. Descompacte e certifique-se que no Matlab você esteja no diretório onde descompactou o arquivo. Digite no terminal do Matlab:

```
open_system('analog_modulation')
```

ou

```
analog_modulation
```

### Toolbox e blocos necessários

Para realização da simulação, são necessários os **Communications System Toolbox** html <sup>[1]</sup>, pdf <sup>[2]</sup> e **DSP System Toolbox**™ html <sup>[3]</sup>, pdf <sup>[4]</sup> fornecido pelo próprio Simulink. A seguir temos os blocos utilizados no modelo se simulação:

Simulink

- Sine Wave <sup>[5]</sup>;
- Scope <sup>[6]</sup>;
- Goto <sup>[7]</sup>;
- From <sup>[8]</sup>.

Communications System Toolbox

- Analog Passband Modulation <sup>[9]</sup>;

DSP System Toolbox

- Spectrum Analyser <sup>[10]</sup>;

Para uma melhor visualização dos blocos e subsistemas usar [Tools> Model Explorer <sup>[11]</sup>]

## Modelo e parâmetros

Modulação é o processo na qual a informação é adicionada a ondas eletromagnéticas. O modelo apresenta diversas modulações analógicas, onde podemos ver suas características principais, bem como suas diferenças, vantagens e desvantagens. Neste modelo é possível controlar através das variáveis do bloco "Parâmetros do modelo", os parâmetros para realização do mesmo bem como eventual testes. Os parâmetros fornecidos por este bloco são:

Parâmetros do modelo

- amp -> Amplitude do sinal de informação;
- f\_sinal -> Frequência do sinal de informação (Hz);
- fs -> Frequência de amostragem (Hz);
- Tipo de modulação -> Escolha da modulação para simulação (AM DSB-FC, AM DSB-SC, AM SSB, FM, PM);
- fp -> Frequência da portadora (Hz);
- Banda lateral modulada -> Escolha da banda lateral a ser modulada na modulação AM SSB;
- fh -> Ordem do filtro de hilbert de recepção para as modulações AM SSB,FM,PM;
- df -> Desvio de frequência;
- d\_fase -> Desvio de fase.

Outras possibilidades de configuração

- Configuração do filtro necessário na recepção no bloco *demodulador*;

Ver também

Dicas para o uso e configuração do Analisador de Espectros do Simulink

## Testes que podem ser feitos

Os testes foram realizados nas versões 2014a e 2015a do software Matlab, funcionando perfeitamente nas mesmas. Você pode alterar alguns parâmetros mesmo que a simulação esteja rodando, basta acessar novamente o bloco parâmetros do modelo. Para isso é necessário que o tempo de simulação seja relativamente longo ou até mesmo infinito (inf). Assim podemos ver possíveis diferenças quando alteramos esses mesmos parâmetros como os relacionados ao filtro de recepção, entre outros. A seguir alguns testes que podem ser feitos com este modelo.

- Visualização dos efeitos de modulação e demodulação no domínio do tempo e frequência;
- Alteração da modulação analógica utilizada para fins de comparação;
- Alteração do filtro de recepção;

Caso queira analisar o procedimento das modulações AM DSB-FC, AM DSB-SC e AM SSB em uma outra forma digite no matlab:

```
AM_DSB_FC
```

```
AM_DSB_SC
```

```
AM_SSB
```

## References

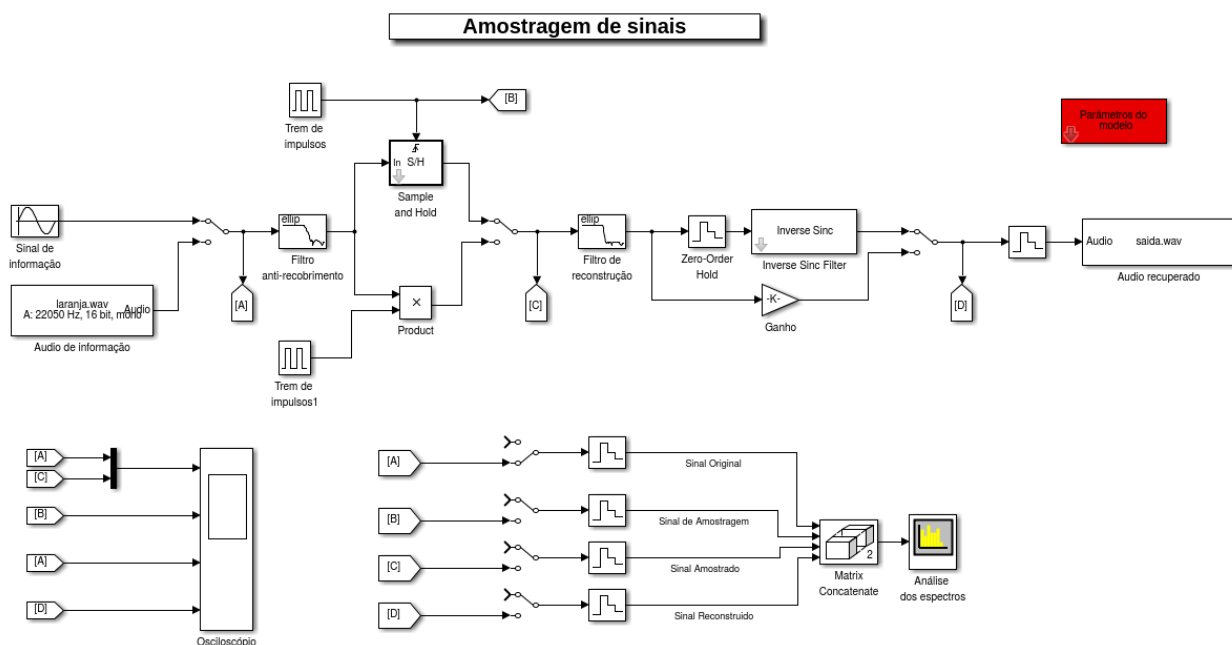
- [1] <http://www.mathworks.com/help/comm/index.html>
- [2] [http://www.mathworks.com/help/pdf\\_doc/comm/comm.pdf](http://www.mathworks.com/help/pdf_doc/comm/comm.pdf)
- [3] <http://www.mathworks.com/help/dsp/index.html>
- [4] [http://www.mathworks.com/help/pdf\\_doc/dsp/dsp Ug.pdf](http://www.mathworks.com/help/pdf_doc/dsp/dsp Ug.pdf)
- [5] <http://www.mathworks.com/help/simulink/slref/sinewave.html>
- [6] <http://www.mathworks.com/help/simulink/slref/scope.html>
- [7] <http://www.mathworks.com/help/simulink/slref/goto.html>
- [8] <http://www.mathworks.com/help/simulink/slref/from.html>
- [9] <http://www.mathworks.com/help/comm/analog-passband-modulation.html>
- [10] <http://www.mathworks.com/help/dsp/ref/spectrumanalyzer.html>
- [11] <http://www.mathworks.com/help/simulink/ug/the-model-explorer-overview.html>

# Amostragem de sinais

## Introdução

O processo de amostragem é uma operação básica ao processamento digital de sinais e comunicações digitais. Através da utilização do processo de amostragem, um sinal analógico passa a ser representado por um conjunto discreto de números, ou amostras que geralmente são uniformemente espaçadas no tempo. Claramente, para este procedimento ter utilidade prática, é necessário que escolhamos a taxa de amostragem ( $F_s$  dado por  $1/T_s$ ) adequadamente, de tal forma que a sequência de amostrar definam unicamente o sinal analógico original. Esta é a essência do teorema da amostragem onde diz que para que seja possível reconstituir o sinal original é necessário que a frequência de amostragem ( $F_s$ ) seja, no mínimo, igual ao dobro da frequência máxima contida no sinal analógico. Caso contrário produz-se um fenómeno indesejável, denominado de aliasing, que se traduz numa sobreposição de espectro que inviabiliza a correcta recuperação do sinal. Ocorre essa sobreposição do espectro devido que o processo de amostragem uniforme de um sinal contínuo de energia finita resulta em um espectro periódico cuja frequência de repetição é igual à taxa de amostragem. Assim é necessário que se respeite o teorema de amostragem para não ocorra esta sobreposição.

## Modelo em Simulink



Primeiramente baixe o arquivo a seguir Sistema.zip. Descompacte e certifique-se que no Matlab você esteja no diretório onde descompactou o arquivo. Digite no terminal do Matlab:

```
amostragem
```

## Toolbox e blocos necessários

Para realização da simulação, é necessário o **DSP System Toolbox™** [html](#) <sup>[3]</sup>, [pdf](#) <sup>[4]</sup> fornecido pelo próprio Simulink. A seguir temos os blocos utilizados no modelo de simulação:

Simulink

- Sine Wave <sup>[5]</sup>;
- Product <sup>[1]</sup>;
- Pulse Generator <sup>[2]</sup>
- Scope <sup>[6]</sup>;
- Mux <sup>[3]</sup>;
- Gain <sup>[4]</sup>;
- Goto <sup>[7]</sup>;
- From <sup>[8]</sup>;
- Matrix Concatenate <sup>[5]</sup>;
- Manual Switch <sup>[6]</sup>;
- Zero-Order Hold <sup>[7]</sup>.

DSP System Toolbox

- Spectrum Analyser <sup>[10]</sup>;
- Analog Filter Design <sup>[8]</sup>;
- From Multimedia File <sup>[9]</sup>;
- To Multimedia File <sup>[10]</sup>.

Para uma melhor visualização dos blocos e subsistemas usar [**Tools> Model Explorer** <sup>[11]</sup>]

## Descrição do Modelo

O modelo simula o processo de diferentes tipos de amostragem de um sinal senoidal ou de áudio. Na Amostragem Instantânea, o sinal de mensagem é multiplicado por um trem periódico de impulsos de *Dirac* onde suas amostras são instantâneas (sem duração). Já na amostragem natural, o sinal de mensagem é multiplicado por um trem periódico de pulsos retangulares, onde o topo de cada pulso retangular pode variar com o sinal de mensagem diferente da amostragem a topo plano onde o topo de cada pulso se mantém plano. A versão amostrada do sinal de mensagem é obtida e a recuperação do sinal original é realizada passando sua versão amostrada por um filtro passa-baixa. Para o caso da amostragem a topo plano é necessário um filtro equalização de fator  $H(f) = \frac{\sin(x)}{x}$  para compensar o *efeito de abertura*.



## Parâmetros e Seleção das Configurações do Modelo

Neste modelo é possível controlar através das variáveis do bloco "Parâmetros do modelo", os parâmetros para realização do mesmo bem como eventual testes. Os parâmetros fornecidos por este bloco são:

Parâmetros do modelo

- amp -> Amplitude do sinal de informação;
- fm -> Frequência do sinal de informação (Hz);
- fs -> Frequência de amostragem (Hz);
- largura -> Largura dos pulsos utilizados para amostragem dos sinais;

Outras possibilidades de configuração

- Configuração dos parâmetros dos filtros pelo próprio bloco do mesmo;
- Chaveamento escolhendo o tipo de amostragem que deseja simular;
- Chaveamento escolhendo o tipo de sinal de mensagem que deseja simular.
- Chaveamento escolhendo um ou vários sinais para análise espectral;

Ver também

Dicas para o uso e configuração do Analisador de Espectros do Simulink

## Possibilidades de Testes

Os testes foram realizados nas versões 2014a e 2015a do software MATLAB, funcionando perfeitamente nas mesmas. A seguir alguns testes que podem ser feitos com este modelo.

- Visualização o processo de amostragem instantanea,natural e a topo plano no domínio do tempo e frequência;
- Escolher o tipo de sinal de mensagem a simular;
- Analisar a reconstrução do sinal original pelo filtro passa-baixa;
- Configuração dos filtros de anti-recobrimento e reconstrução da maneira desejada;
- Altera o valor de taxa de amostragem (Ts), observando o efeito da amostragem abaixo da frequência de Nyquist;
- Caso queira analisar o procedimento da amostragem a topo plano de outra maneira digite no matlab:

```
Amostragem_topo_plano
```

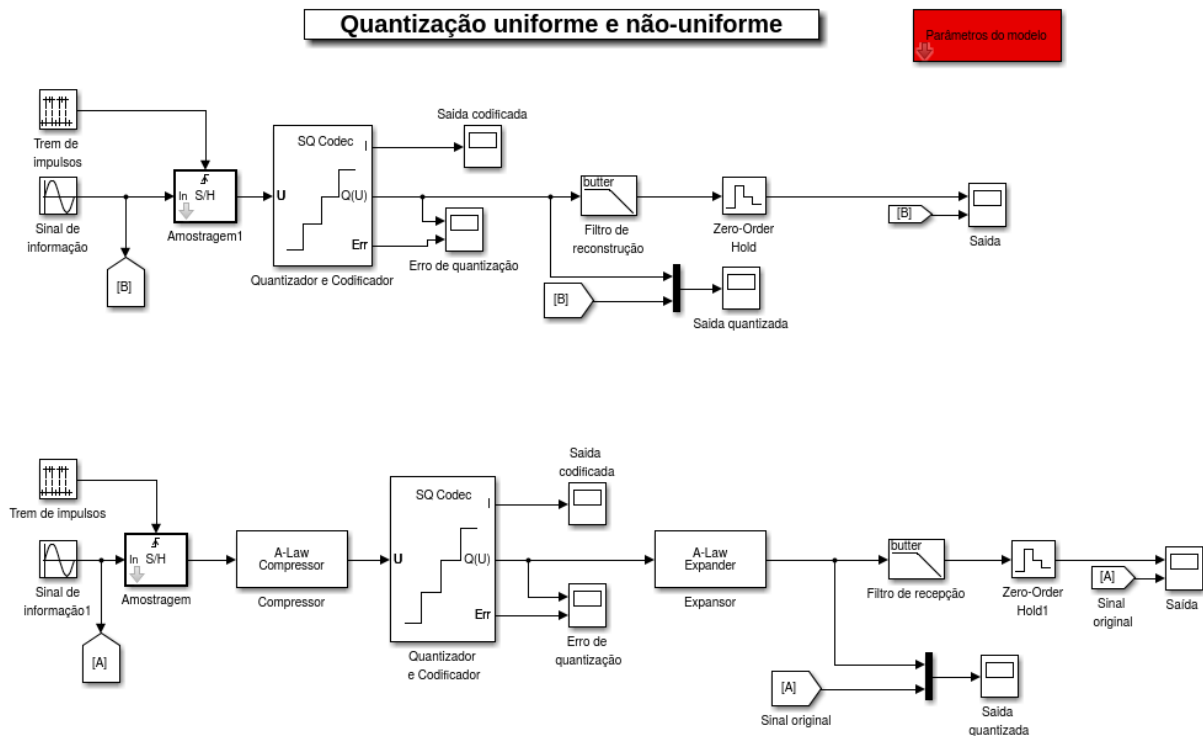
## References

- [1] <http://www.mathworks.com/help/simulink/slref/product.html>
- [2] <http://www.mathworks.com/help/simulink/slref/pulsegenerator.html>
- [3] <http://www.mathworks.com/help/simulink/slref/mux.html>
- [4] <http://www.mathworks.com/help/simulink/slref/gain.html>
- [5] <http://www.mathworks.com/help/simulink/slref/vectorconcatenate.html>
- [6] <http://www.mathworks.com/help/simulink/slref/manualswitch.html;jsessionid=60c72290b7b6368a96ced6247c66>
- [7] <http://www.mathworks.com/help/simulink/slref/zeroorderhold.html>
- [8] <http://www.mathworks.com/help/dsp/ref/analogfilterdesign.html>
- [9] <http://www.mathworks.com/help/dsp/ref/frommultimediafile.html>
- [10] <http://www.mathworks.com/help/dsp/ref/tomultimediafile.html>

# Quantização

## Processo de Quantização - Simulink

A quantização em amplitude é definida como o processo de transformar a amplitude da amostra de um sinal banda base no tempo em uma amplitude discreta tomada de um conjunto finito de possíveis níveis. O processo de quantização é a ligação entre a forma de onda analógica e sua representação em amplitude discreta. Este processo pode seguir uma lei uniforme. Em algumas aplicações, entretanto, é preferível utilizar uma separação variável entre os níveis de representação. Por exemplo, a faixa de tensões cobertas por sinais de voz, do pico de uma fala alta a uma passagem baixa de uma fala baixa, é da ordem de 1000 para 1. Utilizando-se um quantizador não uniforme com a característica do aumento do tamanho do degrau quando a separação da origem da característica entrada/saída é aumentada, os últimos degraus do quantizador podem assumir tais excursões que possibilitam o sinal de voz excursionar em grandes faixas de amplitude que ocorrem com baixa frequência. Em outras palavras, passagens fracas que precisam de mais proteção são favorecidas ao custo de altas passagens. Desta forma, uma precisão percentual praticamente uniforme é atingida em grande parte da faixa de amplitude do sinal de entrada, utilizando menos degraus do que seriam necessários se utilizássemos um quantizador uniforme.



Primeiramente baixe o arquivo a seguir Sistema.zip. Descompacte e certifique-se que no Matlab você esteja no diretório onde descompactou o arquivo. Digite no terminal do Matlab:

```
open_system('quantizacao')
```

ou

```
quantizacao
```

## Toolbox e blocos necessários

Para realização da simulação, são necessários os **Communications System Toolbox** [html](#) <sup>[1]</sup>, [pdf](#) <sup>[2]</sup> e **DSP System Toolbox™** [html](#) <sup>[3]</sup>, [pdf](#) <sup>[4]</sup> fornecido pelo próprio Simulink. A seguir temos os blocos utilizados no modelo de simulação:

### Simulink

- Sine Wave <sup>[5]</sup>;
- Scope <sup>[6]</sup>;
- Goto <sup>[7]</sup>;
- From <sup>[8]</sup>;
- Mux <sup>[3]</sup>;
- Zero-Order Hold <sup>[7]</sup>;
- Pulse Generator <sup>[2]</sup>.

### Communications System Toolbox

- A-Law Expander <sup>[1]</sup>;
- A-Law Compressor <sup>[2]</sup>;
- Mu-Law Expander <sup>[3]</sup>;
- Mu-Law Compressor <sup>[4]</sup>.

### DSP System Toolbox

- Analog Filter Design <sup>[8]</sup>;
- Scalar Quantizer Encoder <sup>[5]</sup>;
- Sample and Hold <sup>[6]</sup>.

Para uma melhor visualização dos blocos e subsistemas usar [**Tools> Model Explorer** <sup>[11]</sup>]

## Modelo e parâmetros

O modelo simula o processo de quantização pós amostragem de um sinal senoidal e um sinal de áudio. O sinal de entrada (banda base) é amostrado por um trem de pulsos retangulares, estreitos o suficiente para se aproximarem do processo de amostragem instantânea. A versão amostrada do sinal de mensagem é, então, quantizada, fornecendo uma nova representação do sinal que, agora, é discreto tanto no tempo quanto em amplitude. O processo de quantização simulado são do tipo uniforme e não-uniforme. Neste modelo é possível controlar através das variáveis do bloco "Parâmetros do modelo", os parâmetros para realização do mesmo bem como eventual testes. Os parâmetros fornecidos por este bloco são:

- amp -> Amplitude do sinal de informação;
  - fm -> Frequência do sinal de informação (Hz);
  - fs -> Frequência de amostragem (Hz);
  - Tipo de quantização -> Escolha entre quantizador mid-tread ou mid-rise;
  - bits -> Número de bits utilizados para obter o número de níveis de amplitude utilizados no quantizador;
  - Lei de compressão -> Escolha entre lei A ou Mu para o quantizador não-uniforme;
  - Valor\_A -> Constante positiva que controla o nível de compressão na lei A;
  - Mu -> Constante positiva que controla o nível de compressão na lei Mu.
-

## Testes que podem ser feitos

Os testes foram realizados nas versões 2014a e 2015a do software Matlab, funcionando perfeitamente nas mesmas. Você pode alterar alguns parâmetros de simulação, basta acessar novamente o bloco parâmetros do modelo. Assim podemos ver possíveis diferenças quando alteramos esses mesmos parâmetros como os relacionados ao quantizador, entre outros. A seguir alguns testes que podem ser feitos com este modelo.

- Analisar as operações de quantização uniforme e não uniforme de um sinal analógico amostrado.
- Analisar o processo de amostragem para diferentes valores de frequência de amostragem (fs);
- Analisar o processo de quantização para diferentes números de níveis de quantização;
- Analisar o quantizador do tipo mid-tread ou mid-rise;
- Analisar a diferença de quantização uniforme e não-uniforme;
- Visualizar a diferença entre as leis de compressão;
- Visualização do erro de quantização;
- Visualizar a saída codificada;

## References

- [1] <http://www.mathworks.com/help/comm/ref/alawexpander.html>
- [2] <http://www.mathworks.com/help/comm/ref/alawcompressor.html>
- [3] <http://www.mathworks.com/help/comm/ref/mulawexpander.html>
- [4] <http://www.mathworks.com/help/comm/ref/mulawcompressor.html>
- [5] <http://www.mathworks.com/help/dsp/ref/scalarquantizerencoder.html>
- [6] <http://www.mathworks.com/help/dsp/ref/sampleandhold.html>

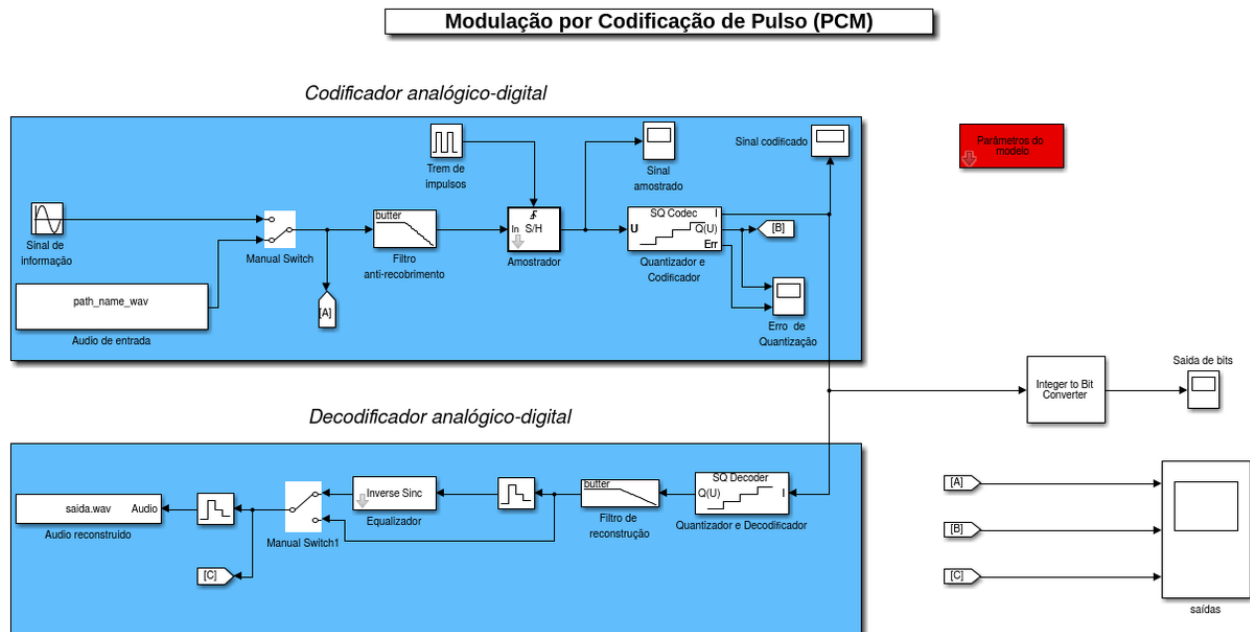
# Modulação por Codificação de Pulso (PCM)

---

## Modulação por Codificação de Pulso (PCM) - Simulink

A modulação por codificação de pulso, que é a forma mais básica de modulação digital de pulso. Na modulação por codificação de pulso (PCM – do inglês, pulse code modulation), um sinal de mensagem é representado por uma sequência de pulsos codificados, obtidos pela discretização do sinal no tempo (amostragem) e na amplitude (quantização). As operações básicas realizadas no codificador PCM são amostragem, quantização e codificação. As operações de quantização e codificação são, geralmente, executadas pelo mesmo circuito, o qual é chamado de conversor analógico digital. As operações básicas no decodificador PCM são decodificação e reconstrução do trem de amostras quantizadas, filtro passa-baixa para recuperar o sinal de mensagem, e filtro equalização do fator

$$H(f) = \frac{\sin(x)}{x}.$$



Primeiramente baixe o arquivo a seguir Sistema.zip. Descompacte e certifique-se que no Matlab você esteja no diretório onde descompactou o arquivo. Digite no terminal do Matlab:

```
open_system('pcm')
```

ou

```
pcm
```

## Toolbox e blocos necessários

Para realização da simulação, são necessários os **Communications System Toolbox** html <sup>[1]</sup>, pdf <sup>[2]</sup> e **DSP System Toolbox™** html <sup>[3]</sup>, pdf <sup>[4]</sup> fornecido pelo próprio Simulink. A seguir temos os blocos utilizados no modelo se simulação:

Simulink

- Sine Wave <sup>[5]</sup>;
- Scope <sup>[6]</sup>;
- Goto <sup>[7]</sup>;
- From <sup>[8]</sup>;
- Zero-Order Hold <sup>[7]</sup>;
- Pulse Generator <sup>[2]</sup>;

Communications System Toolbox

- Integer to Bit Converter <sup>[1]</sup>;

DSP System Toolbox

- From Multimedia File <sup>[9]</sup>;
- To Multimedia File <sup>[10]</sup>;
- Sample and Hold <sup>[6]</sup>;
- Analog Filter Design <sup>[8]</sup>;
- Scalar Quantizer Encoder <sup>[5]</sup>;
- Scalar Quantizer Decoder <sup>[2]</sup>;

Para uma melhor visualização dos blocos e subsistemas usar **[Tools> Model Explorer <sup>[11]</sup>]**

## Modelo e parâmetros

O modelo simula um sistema PCM, utilizando na transmissão um sinal senoidal e em outro um sinal de áudio. O sinal de entrada (banda base) é amostrado por um trem de pulsos retangulares, estreitos o suficiente para se aproximarem do processo de amostragem instantânea. Para garantir a reconstrução perfeita do sinal de mensagem no receptor, a taxa de amostragem deve de ser maior do que duas vezes a largura de banda do sinal de mensagem, de acordo com o teorema da amostragem. A versão amostrada do sinal de mensagem é, então, quantizada, fornecendo uma nova representação do sinal que, agora, é discreto tanto no tempo quanto em amplitude. O processo de quantização pode seguir uma lei uniforme ou não-uniforme. Combinando-se o processo de amostragem e a quantização, o sinal analógico contínuo de mensagem (banda base) é transformado em um conjunto de valores discretos e representados por uma sequência de bits. No receptor o sinal digital é decodificado e reconstruído por um filtro cuja frequência de corte é igual à  $f_s/2$ . Neste modelo é possível controlar através das variáveis do bloco "Parâmetros do modelo", os parâmetros para realização do mesmo bem como eventuais testes. Através de chaves é possível inserir ou não alguns circuitos para observar a sua necessidade no processo PCM.

Parâmetros do modelo

- *amp* -> Amplitude em Volts do sinal de informação  $x(t)$ ;
- *fm* -> Frequência em Hertz do sinal de informação  $x(t)$ ;
- *fs* -> Frequência em Hertz do trem de pulsos de amostragem  $\delta(t)$ ;
- Tipo de quantização -> Escolha entre quantizador *mid-tread* ou *mid-rise*;
- bits -> Número de bits utilizados para obter o número de níveis de amplitude utilizados no quantizador, e os bits utilizados na codificação do sinal digital.

Possibilidades de configuração

- Configuração dos parâmetros dos filtros pelo próprio bloco do mesmo;
- Chave que seleciona o tipo de sinal de entrada (senoide, arquivo)
- Chave que seleciona se o filtro passa baixa de anti-recobrimento é utilizado ou não.
- Chave que seleciona se o filtro equalizador será utilizado ou não.

## Testes que podem ser feitos

Os testes foram realizados nas versões 2014a e 2015a do software Matlab, funcionando perfeitamente nas mesmas. Você pode alterar alguns parâmetros de simulação, basta acessar novamente o bloco parâmetros do modelo. Assim podemos ver possíveis diferenças quando alteramos esses mesmos parâmetros como os relacionados ao amostrador, quantizador, decodificador, entre outros. A seguir alguns testes que podem ser feitos com este modelo.

- Analisar as operações de amostragem, quantização e codificação de um sinal analógico que constituem um sistema PCM;
- Alterar o número de níveis de quantização;
- Visualizar o processo de conversão de um sinal analógico em digital;
- Visualização do erro de quantização (necessário para medir analiticamente o desempenho do sistema);
- Alteração de outros parâmetros para fins de comparação de resultados.

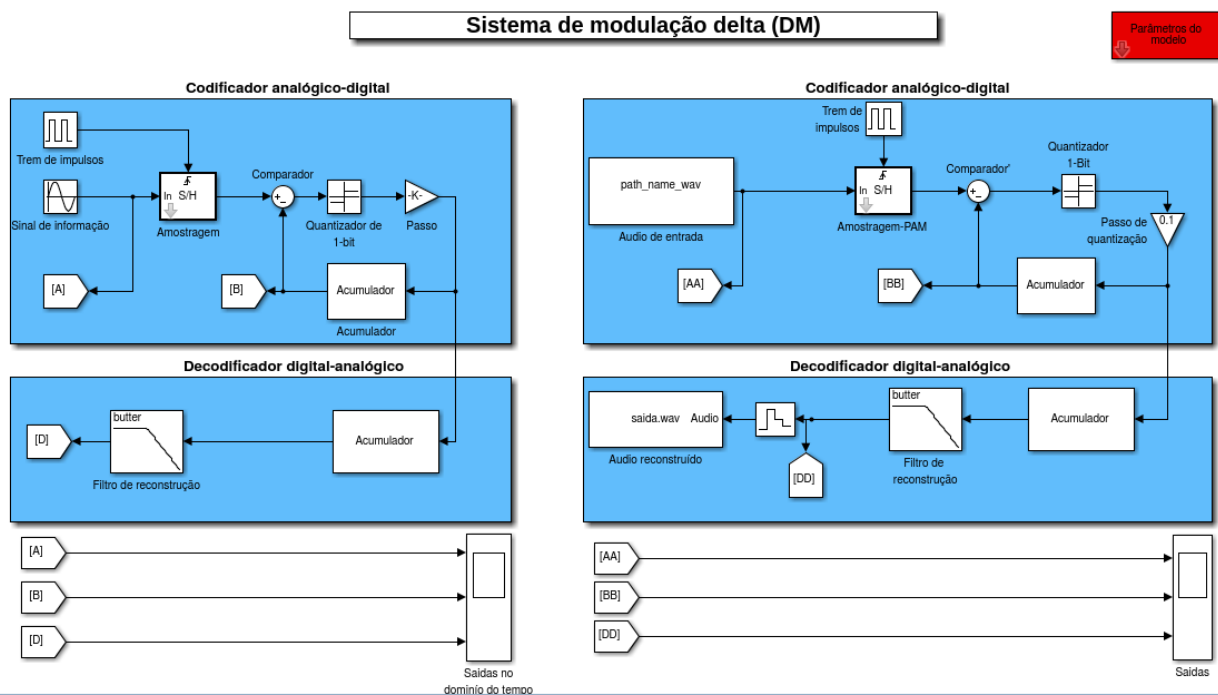
## References

- [1] <http://www.mathworks.com/help/comm/ref/integertobitconverter.html>
- [2] <http://www.mathworks.com/help/dsp/ref/scalarquantizerdecoder.html>

# Modulação Delta (DM)

## Modulação Delta (DM) - Simulink

Na modulação delta (DM – do inglês, delta modulation), um sinal de mensagem de entrada é superamostrado (isto é, em uma taxa muito superior à taxa de Nyquist) para aumentar, propositalmente, a correlação entre amostras adjacentes do sinal. O aumento da correlação é feito de forma a permitir a utilização de uma estratégia de quantização simples para a construção do sinal codificado. Nesta forma básica, a DM fornece uma aproximação em degrau para a versão superamostrada do sinal de mensagem. Ao contrário do PCM, a diferença entre o sinal de entrada e sua aproximação é quantizada em apenas dois níveis. A principal vantagem da modulação delta é sua simplicidade, entretanto, esta vantagem é paga com o aumento da taxa de transmissão de dados.. Ela pode ser implementada aplicando a versão amostrada do sinal de mensagem de entrada a um transmissor constituído por um comparador, um quantizador e um acumulador.



Primeiramente baixe o arquivo a seguir Sistema.zip. Descompacte e certifique-se que no Matlab você esteja no diretório onde descompactou o arquivo. Digite no terminal do Matlab:

```
open_system('dm')
```

ou

```
dm
```

## Toolbox e blocos necessários

Para realização da simulação, é necessário o **DSP System Toolbox™** [html](#) <sup>[3]</sup>, [pdf](#) <sup>[4]</sup> fornecido pelo próprio Simulink. A seguir temos os blocos utilizados no modelo de simulação:

### Simulink

- Sine Wave <sup>[5]</sup>;
- Scope <sup>[6]</sup>;
- Goto <sup>[7]</sup>;
- From <sup>[8]</sup>;
- Zero-Order Hold <sup>[7]</sup>;
- Pulse Generator <sup>[2]</sup>;
- Sum, Add Subtract <sup>[1]</sup>;
- Quantizer <sup>[2]</sup>;
- Gain <sup>[4]</sup>;
- Unit Delay <sup>[3]</sup>.

### DSP System Toolbox

- Analog Filter Design <sup>[8]</sup>;
- Sample and Hold <sup>[6]</sup>;
- From Multimedia File <sup>[9]</sup>;
- To Multimedia File <sup>[10]</sup>.

Para uma melhor visualização dos blocos e subsistemas usar [**Tools> Model Explorer** <sup>[11]</sup>]

## Modelo e parâmetros

O modelo simula o processo de modulação delta (DM) de um sinal senoidal e um sinal de áudio. No transmissor, após o processo de sobreamostragem, o comparador calcula a diferença entre duas entradas. Em seguida o quantizador constituído por um limitador. Portanto, se a aproximação está abaixo do sinal de entrada em qualquer instante amostrado, ela é aumentada no valor do passo de quantização. Por outro lado, se a aproximação está abaixo do sinal, ela é diminuída pelo mesmo valor do passo. Por fim o acumulador opera com a saída do quantizador para produzir uma aproximação do sinal de mensagem. Na recepção, a aproximação em degrau é reconstruída passando a sequência de pulsos positivos ou negativos, produzidas através de um acumulador, de forma similar à utilizada no transmissor. O sinal é reconstruído passando por um filtro passa-baixa com largura de faixa igual à largura de faixa da mensagem original. Neste modelo é possível controlar através das variáveis do bloco "Parâmetros do modelo", os parâmetros para realização do mesmo bem como eventuais testes. Os parâmetros fornecidos por este bloco são:

### Simulink

- amp -> Amplitude do sinal de informação;
- fm -> Frequência do sinal de informação (Hz);
- fs -> Frequência de amostragem (Hz);

### Outras configurações

- Configuração dos parâmetros dos filtros pelo próprio bloco do mesmo;



## Testes que podem ser feitos

Os testes foram realizados nas versões 2014a e 2015a do software Matlab, funcionando perfeitamente nas mesmas. A seguir alguns testes que podem ser feitos com este modelo.

- Visualizar o processo de conversão de um sinal analógico em digital;
- Alteração da taxa de amostragem para eventuais testes;
- Analisar a aproximação em escada realizada pela modulação delta;
- Analisar a distorção por sobrecarga de inclinação e ruído granular;
- Alteração de outros parâmetros para fins de comparação de resultados.

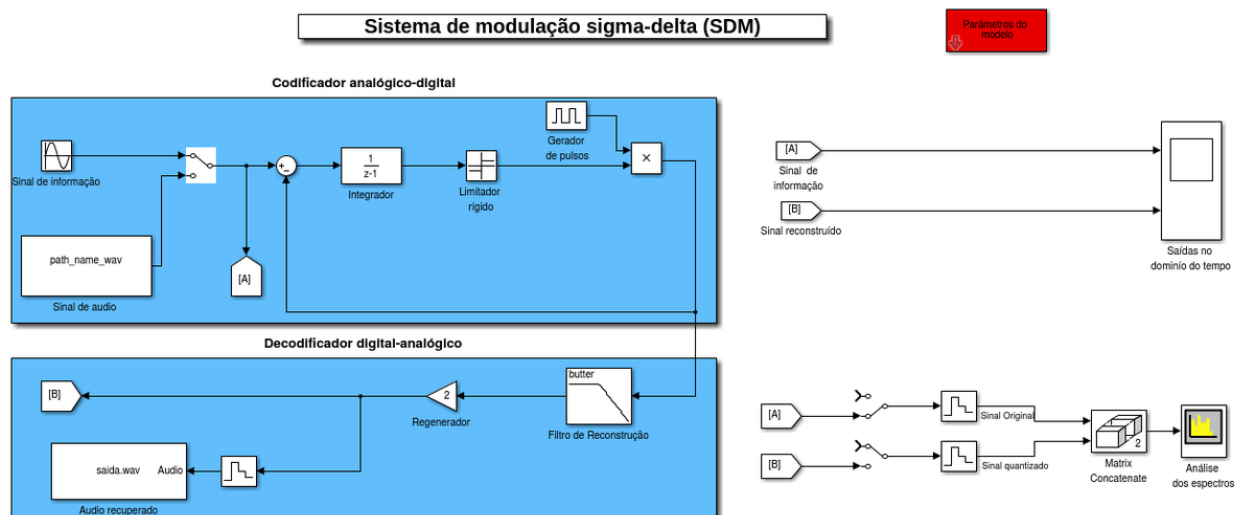
## References

- [1] <http://www.mathworks.com/help/simulink/slref/sum.html>  
 [2] <http://www.mathworks.com/help/simulink/slref/quantizer.html>  
 [3] <http://www.mathworks.com/help/simulink/slref/unitdelay.html>

# Modulação Delta-Sigma

## Modulação Delta-Sigma (SDM) - Simulink

A entrada de um quantizador em uma forma convencional de modulação delta pode ser vista como uma aproximação de uma derivada do sinal de mensagem de entrada. Este comportamento resulta em uma desvantagem, pois distúrbios de transmissão, tais como ruído, resultam em um erro acumulativo no sinal demodulado. Esta desvantagem pode ser superada pela integração do sinal de mensagem antes da modulação delta. O sistema de modulação delta que incorpora a integração em sua entrada é chamado de modulação delta-sigma (D- $\Sigma$ M). Para ser mais preciso, entretanto, ela deve ser chamada de modulação sigma-delta, pois a integração é, de fato, realizada antes da modulação delta.



Primeiramente baixe o arquivo a seguir Sistema.zip. Descompacte e certifique-se que no Matlab você esteja no diretório onde descompactou o arquivo. Digite no terminal do Matlab:

```
open_system('delta_sigma')
```

ou

```
delta_sigma
```

## Toolbox e blocos necessários

Para realização da simulação, é necessário o **DSP System Toolbox™** [html](#) <sup>[3]</sup>, [pdf](#) <sup>[4]</sup> fornecido pelo próprio Simulink. A seguir temos os blocos utilizados no modelo de simulação:

### Simulink

- Sine Wave <sup>[5]</sup>;
- Discrete-Time Integrator <sup>[1]</sup>;
- Scope <sup>[6]</sup>;
- Goto <sup>[7]</sup>;
- From <sup>[8]</sup>;
- Zero-Order Hold <sup>[7]</sup>;
- Pulse Generator <sup>[2]</sup>;
- Quantizer <sup>[2]</sup>;
- Gain <sup>[4]</sup>;
- Unit Delay <sup>[3]</sup>.

### DSP System Toolbox

- Analog Filter Design <sup>[8]</sup>;
- From Multimedia File <sup>[9]</sup>;
- To Multimedia File <sup>[10]</sup>.

Para uma melhor visualização dos blocos e subsistemas usar [**Tools> Model Explorer** <sup>[11]</sup>]

## Modelo e parâmetros

O modelo simula o processo de modulação delta-sigma de um sinal senoidal e um sinal de áudio. O sinal de mensagem é definido em sua forma contínua no tempo, o que significa que o modulador de pulso é constituído, agora, por um limitador seguido por um multiplicador. Este último componente é alimentado por um gerador de pulso externo (clock) para produzir um sinal codificado em um bit. A utilização da integração na entrada do transmissor requer, claramente, uma ênfase inversa do sinal – chamada de diferenciação – no receptor. A necessidade desta diferenciação é, entretanto, eliminada, devido ao seu cancelamento pela integração no receptor DM convencional. Portanto, o receptor de um sistema de modulação delta-sigma é constituído apenas de um filtro passa-baixa. Neste modelo é possível controlar através das variáveis do bloco "Parâmetros do modelo", os parâmetros para realização do mesmo bem como eventuais testes. Os parâmetros fornecidos por este bloco são:

### Parâmetros do modelo

- amp -> Amplitude do sinal de informação;
- fm -> Frequência do sinal de informação (Hz);
- fs -> Frequência de amostragem (Hz);

### Outras possibilidades de configuração

- Configuração dos parâmetros dos filtros pelo próprio bloco do mesmo;
- Chaveamento escolhendo o tipo de sinal de mensagem que deseja simular.
- Chaveamento escolhendo um ou vários sinais para análise espectral;

### Ver também

Dicas para o uso e configuração do Analisador de Espectros do Simulink

## Testes que podem ser feitos

Os testes foram realizados nas versões 2014a e 2015a do software Matlab, funcionando perfeitamente nas mesmas. A seguir alguns testes que podem ser feitos com este modelo.

- Visualizar o processo de conversão de um sinal analógico em digital;
- Analisar o benefício da integração (correlação, projeto do receptor, conteúdo de baixa frequência do sinal é pré-enfatizado);
- Alteração de outros parâmetros para fins de comparação de resultados.

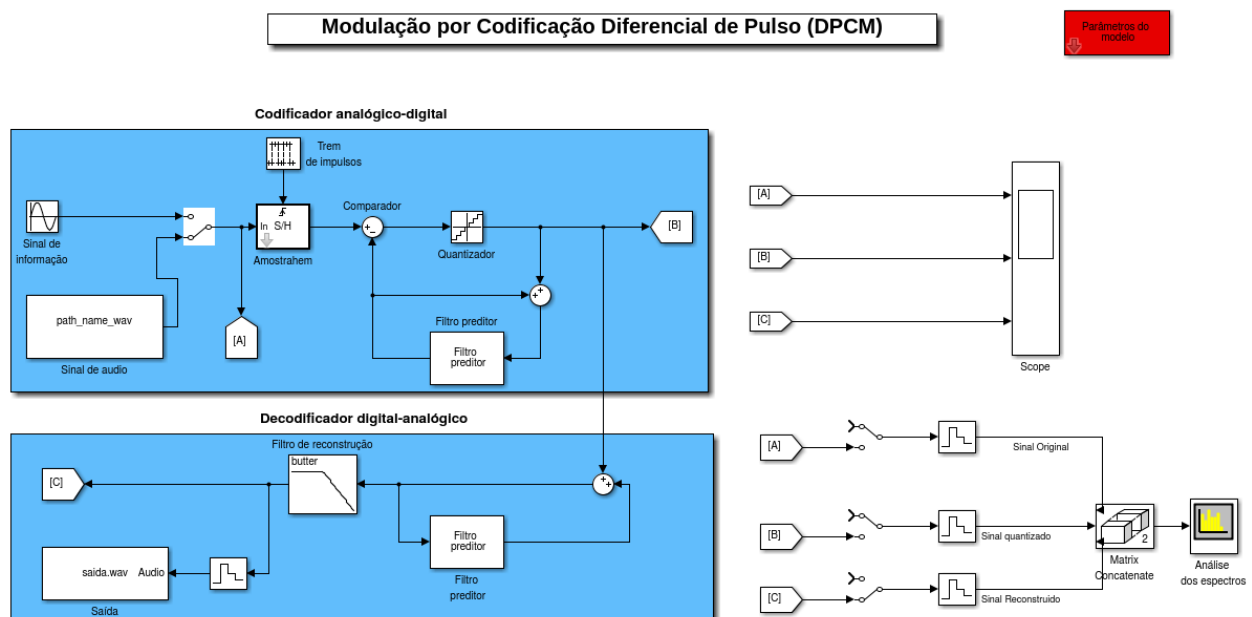
## References

[1] <http://www.mathworks.com/help/simulink/slref/discretetimeintegrator.html>

# Modulação por Codificação Diferencial de Pulso(DPCM)

## Modulação por Codificação Diferencial de Pulso (DPCM) - Simulink

Para outra forma de modulação digital de pulso, observamos que quando um sinal de voz ou vídeo é amostrado a uma taxa um pouco mais alta do que a taxa de Nyquist, o sinal amostrado resultante exibe um alto grau de correlação entre amostras adjacentes. O significado desta alta correlação é que, na média, o sinal não varia rapidamente de uma amostra para a próxima. Quando estas amostras altamente correlacionadas são codificadas como um sistema PCM padrão, o sinal codificado resultante possui informação redundante. A redundância significa que símbolos que não são absolutamente essenciais à transmissão de informação são gerados como resultado do processo de codificação. Removendo esta redundância antes da codificação obtemos um sinal codificado mais eficiente, comparado ao PCM. Agora, se conhecermos uma parte suficiente de um sinal redundante, podemos inferir o resto ou, ao menos, fazer uma estimativa altamente provável. Em particular, se soubermos o comportamento passado de um sinal até um certo ponto no tempo, é possível fazer alguma inferência sobre seus valores futuros, este processo é chamado de predição. O fato de ser possível prever os valores futuros do sinal de entrada é a motivação par ao esquema de quantização diferencial.



Primeiramente baixe o arquivo a seguir Sistema.zip. Descompacte e certifique-se que no Matlab você esteja no diretório onde descompactou o arquivo. Digite no terminal do Matlab:

```
open_system('dpcm')
```

ou

```
dpcm
```

## Toolbox e blocos necessários

Para realização da simulação, é necessário o **DSP System Toolbox™** [html](#) <sup>[3]</sup>, [pdf](#) <sup>[4]</sup> fornecido pelo próprio Simulink. A seguir temos os blocos utilizados no modelo de simulação:

Simulink

- Sine Wave <sup>[5]</sup>;
- Sum <sup>[1]</sup>;
- Product <sup>[1]</sup>;
- Scope <sup>[6]</sup>;
- Goto <sup>[7]</sup>;
- From <sup>[8]</sup>;
- Zero-Order Hold <sup>[7]</sup>;
- Pulse Generator <sup>[2]</sup>;
- Quantizer <sup>[2]</sup>;
- Gain <sup>[4]</sup>;
- Unit Delay <sup>[3]</sup>;

DSP System Toolbox

- Analog Filter Design <sup>[8]</sup>;
- From Multimedia File <sup>[9]</sup>;
- To Multimedia File <sup>[10]</sup>.

Para uma melhor visualização dos blocos e subsistemas usar [**Tools> Model Explorer** <sup>[11]</sup>]

## Modelo e parâmetros

O modelo simula o processo de modulação por Codificação Diferencial de Pulso(DPCM) de um sinal senoidal e um sinal de áudio. Neste esquema, o sinal de entrada do quantizador é definido pela diferença entre a amostra de entrada e a sua predição. Este valor predito é produzido por um filtro de predição, cuja entrada, consiste da versão quantizada das amostras de entrada. O sinal de saída do comparador é chamado de erro de predição, pois ele é o total pelo qual o filtro de predição falha em prever exatamente o sinal de mensagem de entrada. Uma abordagem simples, mas eficaz, para a implementação do filtro de predição é utilizar um filtro de atraso de linha ou um filtro em tempo discreto, com o atraso básico igual ao período de amostragem. Codificando a saída do quantizador, obtemos a variação do PCM chamada de modulação por codificação diferencial de pulso (DPCM, do inglês differential pulse-code modulation). Este é o sinal codificado utilizado na transmissão. No receptor a versão quantizada da entrada original é reconstruída utilizando o mesmo filtro de predição do transmissor. Finalmente, uma estimativa do sinal de mensagem original é obtida passando a versão quantizada através de um filtro de reconstrução passa-baixa. Neste modelo é possível controlar através das variáveis do bloco "Parâmetros do modelo", os parâmetros para realização do mesmo bem como eventual testes. Os parâmetros fornecidos por este bloco são:

Parâmetros do modelo

- amp -> Amplitude do sinal de informação;

- fm -> Frequência do sinal de informação (Hz);
- fs -> Frequência de amostragem (Hz);
- niveis -> Número de níveis de quantização;

Outras possibilidades de configuração

- Configuração dos parâmetros dos filtros pelo próprio bloco do mesmo;
- Chaveamento escolhendo o tipo de sinal de mensagem que deseja simular.
- Chaveamento escolhendo um ou vários sinais para análise espectral;

Ver também

Dicas para o uso e configuração do Analisador de Espectros do Simulink

## Testes que podem ser feitos

Os testes foram realizados nas versões 2014a e 2015a do software Matlab, funcionando perfeitamente nas mesmas. A seguir alguns testes que podem ser feitos com este modelo.

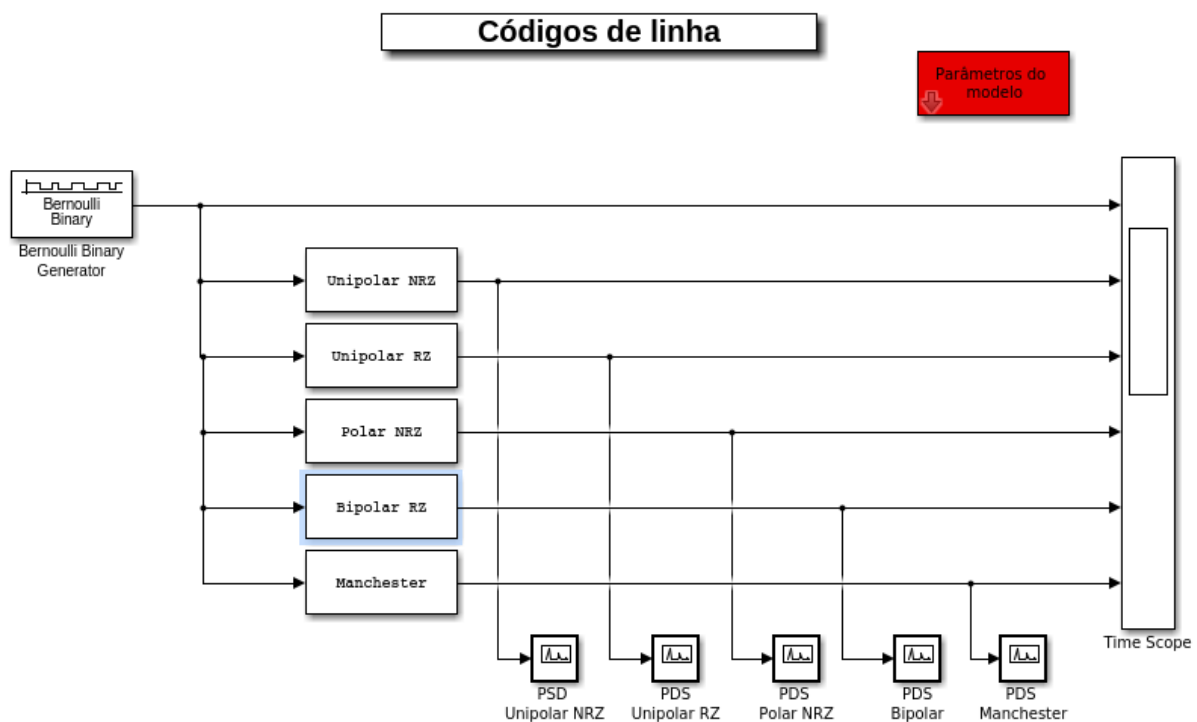
- Visualizar o processo de conversão de um sinal analógico em digital;
  - Analisar as diferenças entre o DPCM e o DM;
  - Alteração de outros parâmetros para fins de comparação de resultados.
  - Analisar o ruído de quantização;
-

# Transmissão de Pulsos em Banda Base

## Códigos de linha

### Códigos de linha - Simulink

Na realidade, PCM, DM e DPCM representam estratégias diferentes para a codificação de fonte, pela qual um sinal analógico é convertido em uma forma digital. Entretanto, todos os três compartilham uma característica comum: uma vez que a sequência binária de 1s e 0s for produzida, um código de linha é necessário para a representação elétrica da sequência binária. Existem vários códigos de linha que podem ser utilizados para esta representação. Cada código de linha tem as suas vantagens e desvantagens, mas eles podem ser caracterizados em geral, como diferentes formas de pulso em banda base.



Primeiramente baixe o arquivo a seguir Sistema.zip. Descompacte e certifique-se que no MATLAB você esteja no diretório onde descompactou o arquivo. Digite no terminal do Matlab:

```
open_system('codigos_linha')
```

ou

```
codigos_linha
```

## Toolbox e blocos necessários

Para realização da simulação, são necessários os **Communications System Toolbox** [html](#) <sup>[1]</sup>, [pdf](#) <sup>[2]</sup> e **DSP System Toolbox™** [html](#) <sup>[3]</sup>, [pdf](#) <sup>[4]</sup> fornecido pelo próprio Simulink. A seguir temos os blocos utilizados no modelo de simulação:

Simulink

- Product <sup>[1]</sup>;
- Pulse Generator <sup>[2]</sup>
- Scope <sup>[6]</sup>;
- Fcn <sup>[1]</sup>.

Communications System Toolbox

- Bernoulli Binary Generator <sup>[2]</sup>;
- Unipolar to Bipolar Converter <sup>[3]</sup>.

DSP System Toolbox

- Spectrum Analyser <sup>[10]</sup>;
- Repeat <sup>[4]</sup>;
- Cumulative Sum <sup>[5]</sup>.

Para uma melhor visualização dos blocos e subsistemas usar [**Tools> Model Explorer** <sup>[11]</sup>]

## Modelo e parâmetros

O modelo simula a utilização de códigos de linha para realizar a representação elétrica de uma sequência binária produzida por um conversor analógico-digital. Os testes foram realizados nas versões 2014a e 2015a do software MATLAB, funcionando perfeitamente nas mesmas. Podemos visualizar os códigos de linha implementados e sua densidade espectral de potência. Neste modelo é possível controlar através das variáveis do bloco "Parâmetros do modelo", os parâmetros para realização do mesmo bem como eventuais testes. Os parâmetros fornecidos por este bloco são:

- Rb -> Taxa de transmissão em bits/s;
- amostras\_por\_bit -> Amostras por Bit.

Para analisar modelo alternativo digite no terminal do MATLAB:

line\_code

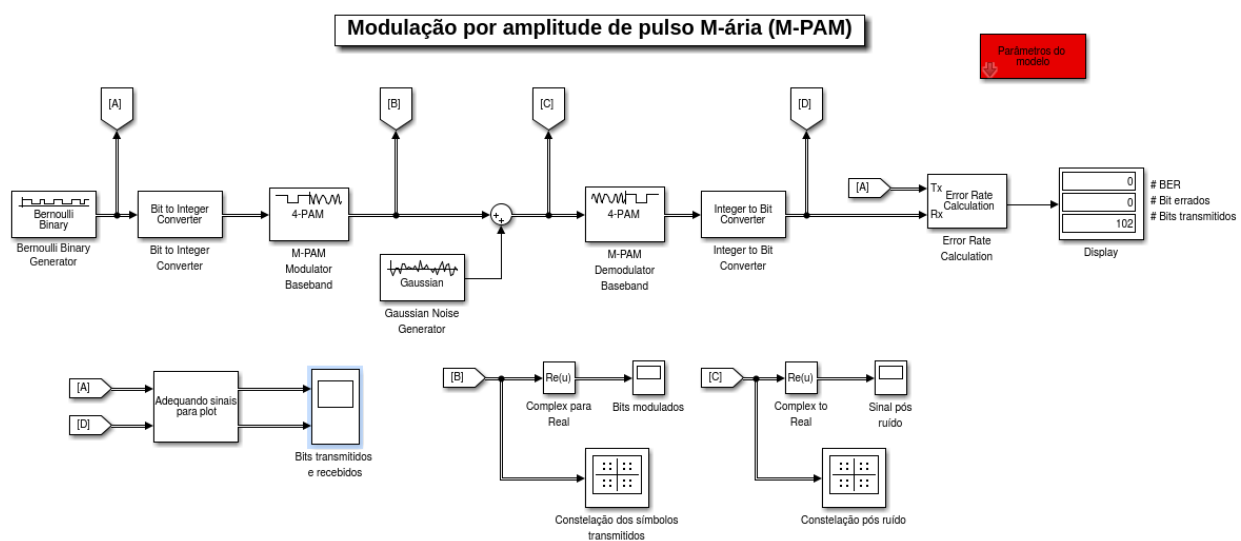
## References

- [1] <http://www.mathworks.com/help/simulink/slref/fcn.html>
- [2] <http://www.mathworks.com/help/comm/ref/bernoullibinarygenerator.html>
- [3] <http://www.mathworks.com/help/comm/ref/unipolartobipolarconverter.html>
- [4] <http://www.mathworks.com/help/dsp/ref/repeat.html>
- [5] <http://www.mathworks.com/help/dsp/ref/cumulativesum.html>

# Modulação por Amplitude de Pulso M-ária (PAM-M)

## Modulação por Amplitude de Pulso M-ária (PAM-M) - Simulink

No sistema PAM binário a sequência emitida pela fonte de informação consiste de símbolos binários que representam um dentre dois possíveis níveis de amplitude,  $-1$  para o símbolo 0 e  $+1$  para o símbolo 1. Por outro lado, em uma versão M-ária banda base do sistema, a saída do codificador de linha assume um dentre  $M$  possíveis níveis de amplitude, com  $M > 2$ . Em um sistema M-ário, a fonte de informação emite a sequência de símbolos de um alfabeto que consiste de  $M$  símbolos onde consegue-se transmitir  $\log_2(M)$  bits em cada um desses símbolos. Cada nível de amplitude da saída do codificador de linha corresponde a um símbolo distinto, de tal forma que existem  $M$  níveis distintos de amplitude para serem transmitidos.



Primeiramente baixe o arquivo a seguir Sistema.zip. Descompacte e certifique-se que no Matlab você esteja no diretório onde descompactou o arquivo. Digite no terminal do Matlab:

```
open_system('M_PAM')
```

ou

```
M_PAM
```

## Toolbox e blocos necessários

Para realização da simulação, é necessários o **Communications System Toolbox** html <sup>[1]</sup>, pdf <sup>[2]</sup> fornecido pelo próprio Simulink. A seguir temos os blocos utilizados no modelo se simulação:

Simulink

- Display <sup>[1]</sup>;
- Complex to Real-Imag <sup>[2]</sup>;
- Sum <sup>[1]</sup>;
- Scope <sup>[6]</sup>;
- Mux <sup>[3]</sup>;
- Goto <sup>[7]</sup>;
- From <sup>[8]</sup>.



Communications System Toolbox

- M-PAM Modulator Baseband <sup>[3]</sup>;
- M-PAM Demodulator Baseband <sup>[4]</sup>;
- Error Rate Calculation <sup>[5]</sup>;
- Integer to Bit Converter <sup>[1]</sup>;
- Bit to Integer Converter <sup>[6]</sup>;
- Bernoulli Binary Generator <sup>[2]</sup>;
- Constellation Diagram <sup>[7]</sup>;
- Gaussian Noise Generator <sup>[8]</sup>.

Para uma melhor visualização dos blocos e subsistemas usar [**Tools> Model Explorer** <sup>[11]</sup>]

## Modelo e parâmetros

O modelo simula o processo de Modulação por Amplitude de Pulso M-ária (PAM-M) de uma sequência de bits codificadas. Neste tipo de modulação a constelação no espaço de sinais é constituída por mais de dois pontos. Em um sistema PAM M-ário, somos capazes de transmitir dados em uma taxa que é  $\log_2(M)$  mais rápida do que o correspondente em um sistema PAM binário. Neste modelo é possível controlar através das variáveis do bloco "Parâmetros do modelo", os parâmetros para realização do mesmo bem como eventual testes. Os parâmetros fornecidos por este bloco são:

- M\_ary -> Ordem da modulação;
- Rb -> Taxa de transmissão em bits/s;
- variancia -> Variância do ruído gaussiano branco;

Os testes foram realizados nas versões 2014a e 2015a do software MATLAB, funcionando perfeitamente nas mesmas. Nela podemos visualizar a constelação dos símbolos transmitidos, bem como o desempenho de taxa de erro de bit (BER) do sistema.

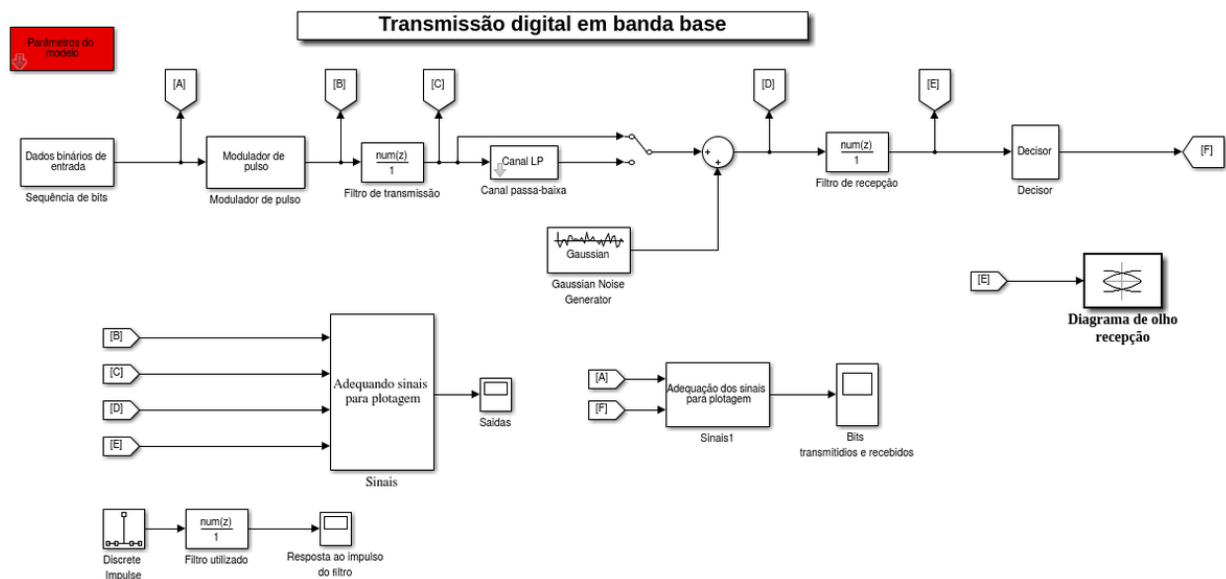
## References

- [1] <http://www.mathworks.com/help/simulink/slref/display.html>
- [2] <http://www.mathworks.com/help/simulink/slref/complextoimag.html>
- [3] <http://www.mathworks.com/help/comm/ref/mpammodulatorbaseband.html>
- [4] <http://www.mathworks.com/help/comm/ref/mpamdemodulatorbaseband.html>
- [5] <http://www.mathworks.com/help/comm/ref/errortatecalculation.html>
- [6] <http://www.mathworks.com/help/comm/ref/bittointegerconverter.html>
- [7] <http://www.mathworks.com/help/comm/ref/constellationdiagram.html>
- [8] <http://www.mathworks.com/help/comm/ref/gaussiannoisegenerator.html>

# Transmissão digital em banda base

## Transmissão digital em banda base - Simulink

A transmissão em banda base de dados digitais requer o uso de um canal limitado em banda com uma largura de banda suficientemente grande para acomodar o conteúdo de frequência essencial do fluxo de dados. Isso é necessário devido ao dados digitais possuírem um espectro amplo com conteúdo significativo em baixa frequência. Entretanto, o canal geralmente é dispersivo (resposta em frequência se afasta daquela de um filtro passa-baixas ideal), assim o pulso recebido é bastante afetado por pulsos adjacentes, provocando o surgimento de *interferência intersimbólica*. Para corrigi-la, exercemos controle sobre o formato do pulso em todo o sistema. Outro grande problema em um sistema de transmissão de dados em banda base é o *ruído do canal*, a utilização de um filtro linear invariante no tempo é essencial para detecção ótima do pulso transmitido corrompido por ruído.



Primeiramente baixe o arquivo a seguir Sistema.zip. Descompacte e certifique-se que no Matlab você esteja no diretório onde descompactou o arquivo. Digite no terminal do Matlab:

```
open_system('transmissao_banda_base')
```

ou

```
transmissao_banda_base
```

## Toolbox e blocos necessários

Para realização da simulação, são necessários os **Communications System Toolbox** html <sup>[1]</sup>, pdf <sup>[2]</sup> e **DSP System Toolbox™** html <sup>[3]</sup>, [4] fornecido pelo próprio Simulink. A seguir temos os blocos utilizados no modelo de simulação:

Simulink

- Goto <sup>[7]</sup>;
- From <sup>[8]</sup>;
- Discrete FIR Filter <sup>[1]</sup>;
- Sum, Add, Subtract, Sum of Elements <sup>[2]</sup>;
- Scope <sup>[6]</sup>;
- Sine Wave <sup>[5]</sup>;

- Product <sup>[1]</sup>.

#### Communications System Toolbox

- Gaussian Noise Generator <sup>[8]</sup>;
- Eye Diagram <sup>[3]</sup>;
- Bernoulli Binary Generator <sup>[2]</sup>;
- M-PAM Demodulator Baseband <sup>[4]</sup>;
- M-PAM Modulator Baseband <sup>[3]</sup>.

#### DSP System Toolbox

- Lowpass Filter <sup>[4]</sup>;
- Discrete Impulse <sup>[5]</sup>.

Para uma melhor visualização dos blocos e subsistemas usar [**Tools> Model Explorer** <sup>[11]</sup>]

## Modelo e parâmetros

O modelo considera então um sistema M-PAM de banda base. A sequência binária transmitida de entrada consiste em símbolos 1 e 0. No modulador de pulso modificamos a sequência binária para uma nova sequência de pulsos breves (que se aproxima de um impulso unitário). A sequência de pulsos breves é assim aplicada a um filtro de transmissão produzindo o sinal transmitido, o sinal é modificado ao passar pelo canal, além de acrescentar o ruído aleatório ao sinal na entrada do receptor. O sinal passa pelo filtro de recepção (filtro casado) e por fim por meio de um dispositivo de decisão para recepção de forma correta dos símbolos 1 e 0 transmitidos. Neste modelo é possível controlar através das variáveis do bloco "Parâmetros do modelo", os parâmetros para realização do mesmo bem como eventual testes. Os parâmetros fornecidos por este bloco são:

- rs -> Taxa de transmissão em símbolos/s;
- tipo -> Tipo de pulso de formatação (Retangular,Sinc,Cosseno elevado, Gerador de ISI);
- span -> Extensão do filtro de transmissão e recepção;
- up\_sampling -> Fator de sobreamostragem (Amostras por símbolo);
- roll\_off -> Fator de decaimento do filtro cosseno elevado;
- variancia -> Variância do ruído gaussiano branco;
- M\_ary -> Ordem da modulação;

## Testes que podem ser feitos

Os testes foram realizados nas versões 2014a e 2015a do software Matlab, funcionando perfeitamente nas mesmas. A seguir alguns testes que podem ser feitos com este modelo.

- Visualizar o processo de transmissão de dados digitais em banda base;
- Visualizar o processo de formatação do pulso, bem como a utilização do filtro casado para maximizar a relação sinal-ruído e consequentemente detectar de maneira ótima os dados transmitidos;
- Você pode alterar o filtro utilizado para formatação do pulso;
- As opções disponíveis de pulsos são: Retangular ideal, Pulso Sinc, Pulso Cosseno Elevado e Pulso gerador de *interferência intersímbolica*;
- Alteração de alguns parâmetros relacionado a estes pulsos, como: Extensão do pulso no tempo, amostras por símbolo e no caso do pulso Cosseno Elevado alterar o valor de *roll-off*;
- Visualizar o atraso imposto pelo canal passa-baixas bem como dos filtros de recepção de transmissão (Caso eles forem bastante extensos);
- Alteração dos valores em frequência do canal passa-baixas;
- Visualizar o diagrama de olho do sinal recebido para a visualização da distorção na forma do mesmo;

## Modelo alternativo

Digite no terminal do Matlab:

```
open_system('transmissao_bandbase')
```

ou transmissao\_bandbase

## References

- [1] <http://www.mathworks.com/help/simulink/slref/discretefirfilter.html?refresh=true>
- [2] <http://www.mathworks.com/help/simulink/slref/add.html>
- [3] <http://www.mathworks.com/help/comm/ref/eyediagramscope.html>
- [4] <http://www.mathworks.com/help/dsp/ref/lowpassfilter.html>
- [5] <http://www.mathworks.com/help/dsp/ref/discreteimpulse.html>

---

# Transmissão de Dados em Banda Passante

---

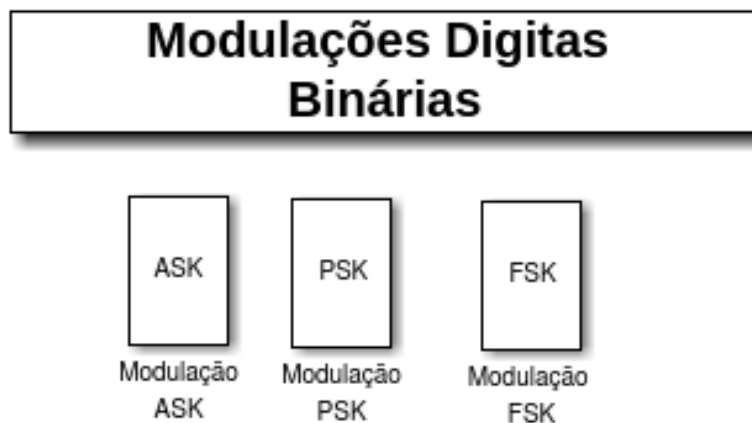
## Modulações digitais binárias

---

### Modulações digitais binárias - Simulink

Como faríamos se quisermos transmitir uma sequência de dados em um canal passa-faixa, exemplificado por um canal de comunicação sem fio ou por satélite? Em aplicações deste tipo, geralmente optamos por trabalhar com uma estratégia de modulação configurada ao redor de uma portadora senoidal cuja amplitude, fase ou frequência é variada de acordo com a sequência de dados contendo a informação. Dada uma fonte binária que emite os símbolos 0 e 1, o processo de modulação envolve o chaveamento da amplitude, fase ou frequência de uma onda senoidal dentre um par de possíveis valores de acordo com os símbolos 0 e 1. Alterando o valor de uma destas variáveis em função do sinal a modular, é possível apresentar as seguintes modulações binárias?

- ASK – Amplitude Shift Keying,
- FSK – Frequency Shift Keying;
- PSK – Phase Shift Keying.



Primeiramente baixe o arquivo a seguir Sistema.zip. Descompacte e certifique-se que no Matlab você esteja no diretório onde descompactou o arquivo. Digite no terminal do Matlab:

```
open_system('modulacoes_binarias')
```

ou

```
modulacoes_binarias
```

---

## Toolbox e blocos necessários

Para realização da simulação, é necessário o **Communications System Toolbox™** [html](#) <sup>[1]</sup>, [pdf](#) <sup>[2]</sup> fornecido pelo próprio Simulink e os seguintes blocos:

Simulink

- Sine Wave <sup>[5]</sup>;
- Switch <sup>[1]</sup>;
- Product <sup>[1]</sup>;
- Constant <sup>[2]</sup>;
- Scope <sup>[6]</sup>;
- Goto <sup>[7]</sup>;
- Sum <sup>[1]</sup>;
- From <sup>[8]</sup>;
- Delay <sup>[3]</sup>.

Communications System Toolbox

- Bernoulli Binary Generator <sup>[2]</sup>;
- Integrate and Dump <sup>[4]</sup>;
- Windowed Integrator <sup>[5]</sup>.

Para uma melhor visualização dos blocos e subsistemas usar [**Tools> Model Explorer** <sup>[11]</sup>]

## Modelo

O modelo simula o processo de modulação digital, dada uma fonte binária que emite os símbolos 0 e 1. O processo de modulação envolve o chaveamento da amplitude (ASK), fase (PSK) ou frequência (FSK) de uma onda senoidal dentre um par de possíveis valores de acordo com os símbolos 0 e 1. O modelo apresenta de forma simples o modulador e demodulador das modulações ASK, PSK e FSK. A partir do modelo, podemos visualizar os efeitos de modulação e demodulação no domínio do tempo. Os testes foram realizados nas versões 2014a e 2015a do software Matlab.

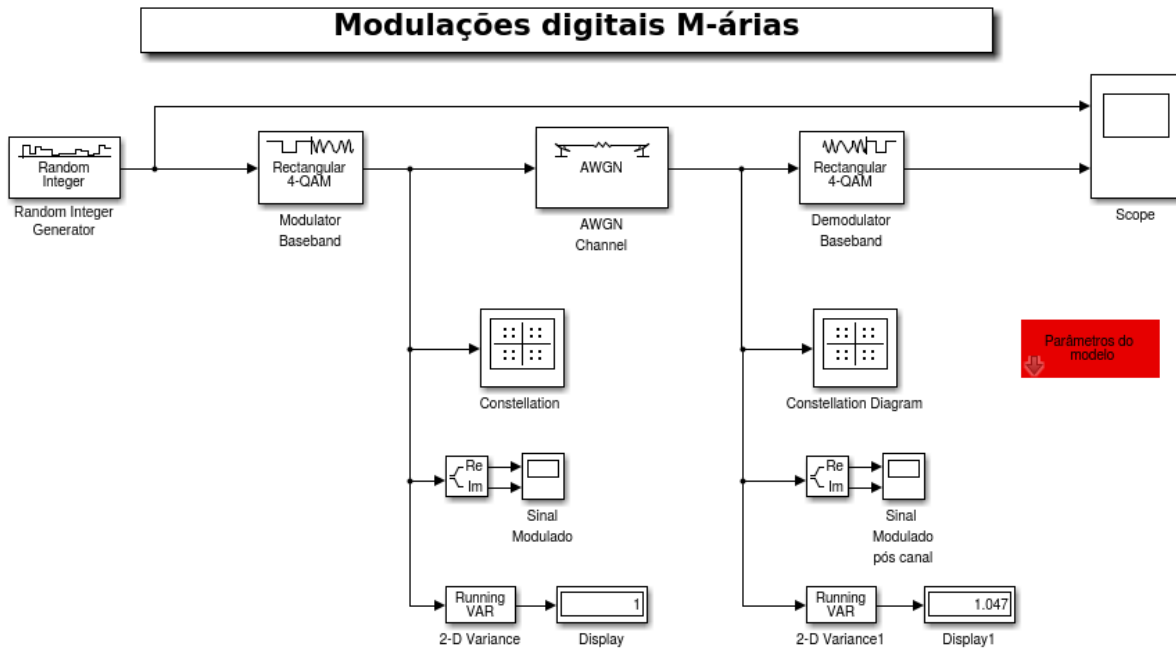
## References

- [1] <http://www.mathworks.com/help/simulink/slref/switch.html>
- [2] <http://www.mathworks.com/help/simulink/slref/constant.html>
- [3] <http://www.mathworks.com/help/simulink/slref/delay.html>
- [4] <http://www.mathworks.com/help/comm/ref/integrateanddump.html>
- [5] <http://www.mathworks.com/help/comm/ref/windowedintegrator.html>

# Modulações digitais M-árias

## Modulações digitais M-árias - Simulink

Por definição, em um esquema de modulação digital M-ário, enviamos qualquer um de M possíveis sinais  $s_1(t)$ ,  $s_2(t)$ , ...,  $s_M(t)$  durante cada intervalo de sinalização (símbolo) de duração T. Em quase todas as aplicações,  $M = 2^m$ , onde m é um inteiro. Sob esta condição a duração do símbolo é  $T = mT_b$  onde  $T_b$  é a duração do bit. Aumentar M significa aumentar o número de símbolos transmitidos. Supondo que a taxa de transmissão de símbolos seja constante, o aumento de M implica em um aumento da taxa de bits, sem que a largura de faixa seja aumentada. Em outras palavras, tem-se maior utilização da banda. Assim, esquemas de modulação M-ário são preferidos em detrimento de esquemas de modulação binário para a transmissão de dados digitais em canais passa-faixa quando a condição é conservar largura de faixa ao custo de aumento de potência e complexidade do sistema. Na prática, raramente encontramos um canal de comunicação que possui a largura de faixa exata necessária para a transmissão da saída de uma fonte de informação através de esquemas de modulação binária. Portanto, quando a largura de faixa do canal é menor do que o valor necessário, utilizamos um esquema de modulação M-ário para a conservação máxima de largura de faixa.



Primeiramente baixe o arquivo a seguir Sistema.zip. Descompacte e certifique-se que no Matlab você esteja no diretório onde descompactou o arquivo. Digite no terminal do Matlab:

```
open_system('Modulacoes_digitais')
```

ou

```
Modulacoes_digitais
```

## Toolbox e blocos necessários

Para realização da simulação, são necessários os **Communications System Toolbox** [html](#) <sup>[1]</sup>, [pdf](#) <sup>[2]</sup> e **DSP System Toolbox™** [html](#) <sup>[3]</sup>, [pdf](#) <sup>[4]</sup> fornecido pelo próprio Simulink. A seguir temos os blocos utilizados no modelo de simulação:

Simulink

- Scope <sup>[6]</sup>;
- Complex to Real-Imag <sup>[2]</sup>;
- Display <sup>[1]</sup>.

Communications System Toolbox

- Random Integer Generator <sup>[1]</sup>;
- AWGN Channel <sup>[2]</sup>;
- Digital Baseband Modulation <sup>[3]</sup>;
- Constellation Diagram <sup>[7]</sup>;

DSP System Toolbox

- Variance <sup>[4]</sup>;

Para uma melhor visualização dos blocos e subsistemas usar [**Tools> Model Explorer** <sup>[11]</sup>]

## Modelo e parâmetros

O modelo simula o processo de modulação e demodulação digital M-ária, dada uma fonte que emite os símbolos de 0 a M-1. As modulações M-árias apresentam uma constelação no espaço de sinais constituída por mais de dois pontos. Ela fornece a base matemática para a representação geométrica de sinais de energia, exemplificado pelas formas de onda digitalmente moduladas. Para um método específico de modulação, a representação geométrica é mostrada na forma de uma constelação de pontos no diagrama de espaço de sinal, a qual é única para aquele método. As modulações simuladas aqui são QAM e M-PSK. É possível controlar através das variáveis do bloco "Parâmetros do modelo", os parâmetros para realização do mesmo bem como eventuais testes. Os parâmetros fornecidos por este bloco são:

- bitRate -> Taxa de transmissão (b/s);
- Modulação digital utilizada -> Escolha da modulação usada (M-QAM, M-PSK);
- M\_ary -> Ordem da modulação.
- inputSignal -> Potência do sinal de entrada (watts);
- snr -> Relação sinal-ruído (SNR);
- EbNo -> Relação sinal-ruído (Eb/No).

## Testes que podem ser feitos

Os testes foram realizados nas versões 2014a e 2015a do software Matlab. A seguir alguns testes que podem ser feitos com este modelo.

- Visualização dos efeitos de modulação e demodulação no domínio do tempo (parte real e parte imaginária);
- Outro teste válido é alterar o tipo de modulação utilizada para fins de comparação;
- Analisar a constelação dos símbolos transmitidos alterando a relação sinal-ruído com a simulação rodando para fins de comparação.



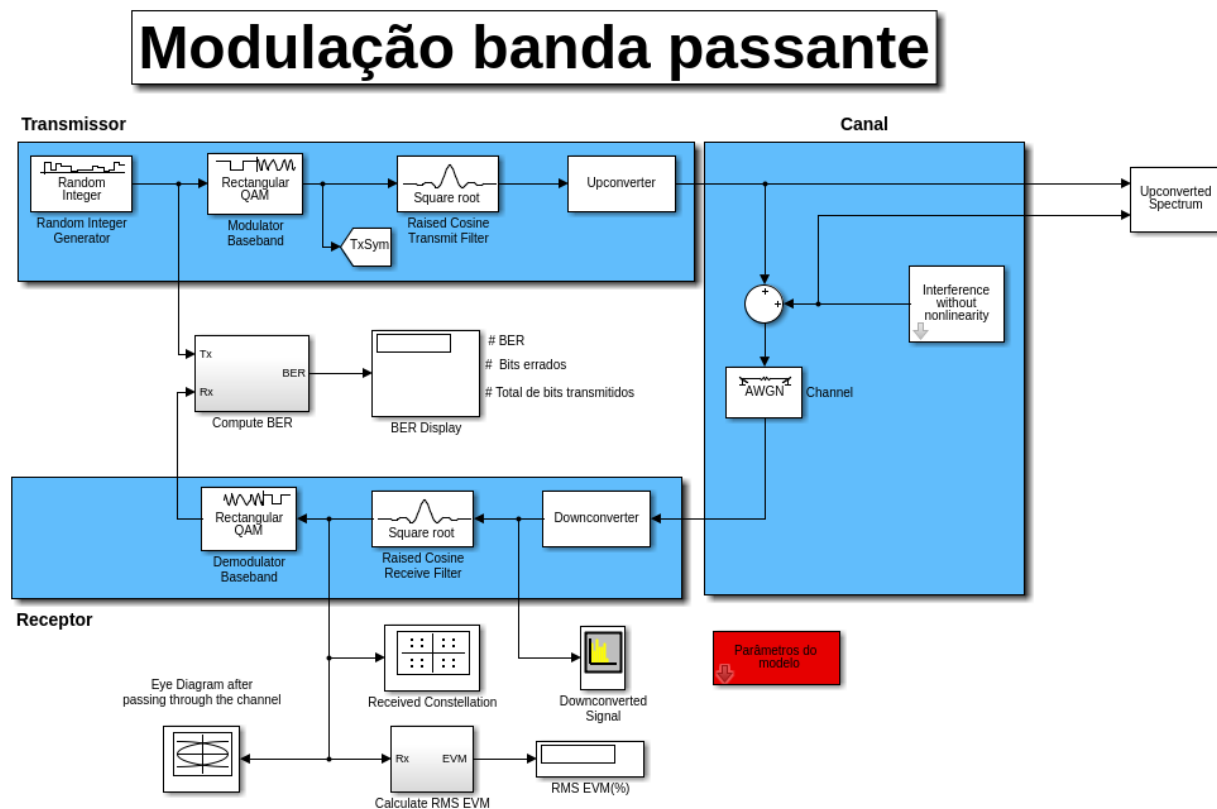
## References

- [1] <http://www.mathworks.com/help/comm/ref/randomintegergenerator.html>
- [2] <http://www.mathworks.com/help/comm/ref/awgnchannel.html>
- [3] <http://www.mathworks.com/help/comm/digital-baseband-modulation.html>
- [4] <http://www.mathworks.com/help/dsp/ref/variance.html>

# Passband Modulation

## Passband Modulation

Este modelo mostra uma maneira simples para executar a modulação de banda passante, multiplicando um sinal complexo modulado com uma onda senoidal para transladar o sinal na frequência. Mais informações [6].



Primeiramente baixe o arquivo a seguir Sistema.zip (que é uma versão parametrizável daquele encontrado na pasta em /opt/MATLAB/R20xxx/toolbox/comm/commdemos). Descompacte e certifique-se que no Matlab você esteja no diretório onde descompactou o arquivo. Digite no terminal do Matlab:

```
open_system('bandpass_modulation')
```

ou

```
'bandpass_modulation'
```

## Toolbox e blocos necessários

Para realização da simulação, são necessários os **Communications System Toolbox** html <sup>[1]</sup>, pdf <sup>[2]</sup> e **DSP System Toolbox™** html <sup>[3]</sup>, pdf <sup>[4]</sup> fornecido pelo próprio Simulink. O exemplo é realizado de acordo com o seguinte diagrama de blocos:

- Random Integer Generator <sup>[1]</sup>;
- QPSK Modulator Baseband <sup>[1]</sup>;
- Goto <sup>[7]</sup>;
- Raised Cosine Transmit Filter <sup>[2]</sup>;
- Upconverter (Subsystem) <sup>[3]</sup>;
- Sum <sup>[1]</sup>;
- Interference (Subsystem) <sup>[3]</sup>;
- Upconverted Spectra (Subsystem) <sup>[3]</sup>;
- AWGN Channel <sup>[2]</sup>;
- Downconverter (Subsystem) <sup>[3]</sup>;
- QPSK Demodulator Baseband <sup>[4]</sup>;
- Compute BER (Subsystem) <sup>[3]</sup>;
- Display <sup>[1]</sup>;
- Constellation Diagram <sup>[7]</sup>;
- Calculate RMS EVM (Subsystem) <sup>[3]</sup>;
- Spectrum Analyzer <sup>[10]</sup>;
- Eye Diagram <sup>[3]</sup>.

Para uma melhor visualização dos blocos e subsistemas usar [**Tools> Model Explorer** <sup>[11]</sup>]

## Subsistemas

Alguns dos blocos do diagrama citado acima, correspondem a um conjunto de partes inter-relacionadas integrante de um sistema mais amplo, em outras palavras um subsistema. Os blocos utilizados em cada subsistema são especificados abaixo.

### Upconverter

- Inport <sup>[5]</sup>;
- Sine Wave <sup>[5]</sup>;
- Product <sup>[1]</sup>;
- Outport <sup>[6]</sup>;

### Interference

- Sine Wave <sup>[5]</sup>;
- Constant <sup>[2]</sup>;
- Math <sup>[7]</sup>;
- Gain <sup>[4]</sup>;
- Outport <sup>[6]</sup>;

### Upconverted Spectra

- Inport <sup>[5]</sup>;
- Concatenate <sup>[5]</sup>;
- Spectrum Analyzer <sup>[10]</sup>;

### Downconverter

- Inport <sup>[5]</sup>;

- Sine Wave <sup>[5]</sup>;
- Product <sup>[1]</sup>;
- Math (conj) <sup>[7]</sup>;
- Outport <sup>[6]</sup>;

Compute BER

- Inport <sup>[5]</sup>;
- Error Rate Calculation <sup>[5]</sup>;
- Integer to Bit Converter <sup>[1]</sup>;
- Constant <sup>[2]</sup>;
- Outport <sup>[6]</sup>;

Calculate RMS EVM

- From <sup>[8]</sup>;
- Delay <sup>[8]</sup>;
- Inport <sup>[5]</sup>;
- Outport <sup>[6]</sup>;
- EVM Measurement <sup>[9]</sup>;

## Modelo e parâmetros

O modelo realiza a transmissão em banda passante de sinais modulados digitalmente por um canal ruidoso ou com desvanecimento por multipercurso (Rayleigh ou Rician). Uma fonte de interferência pode ser combinada com o canal. A interferência tem uma não-linearidade cúbica que pode ser ativada ou desativada. Quando a não-linearidade está desligado, a interferência cai completamente fora de banda, mas quando ativada, o terceiro harmônico desta interferência é introduzido na banda desejada, causando interferência co-canal. A transmissão dos símbolos é realizado em *frames*. Neste modelo é possível controlar através das variáveis do bloco "Parâmetros do modelo", os parâmetros para realização do mesmo bem como eventual testes. Os parâmetros fornecidos por este bloco são:

Parâmetros do modelo

- symbolRate -> Taxa de símbolos (bauds);
- symbolPerFrame -> Número de símbolos por frame;
- Modulação digital utilizada -> Escolha da modulação usada (M-QAM,BPSK,QPSK,M-DPSK);
- M\_ary -> Ordem da modulação;
- maxDopplerShift -> Maximum Doppler shift of diffuse components (Hz);
- delayVector -> Discrete delays of channel (s);
- gainVector -> Average path gains (dB);
- LOSDopplerShift -> Doppler shift of line-of-sight component (Hz);
- KFactor -> Ratio of specular power to diffuse power;
- inputSignal -> Potência do sinal de entrada (watts);
- snr -> Relação sinal-ruído (SNR);
- EbNo -> Relação sinal-ruído (Eb/No);
- Ativar/Desativar interferencia não-linear;
- fci -> Frequência do oscilador de interferência (Hz);
- fc -> Frequência do oscilador (Hz);
- samplesPerSymbol -> Amostras por símbolo;

Ver também

Dicas para o uso e configuração do Analisador de Espectros do Simulink

## Testes que podem ser feitos

Os testes foram realizados nas versões 2014a e 2015a do software Matlab, funcionando perfeitamente nas mesmas. A seguir alguns testes que podem ser feitos com este modelo.

- Analise no domínio da frequência do sinal pós Upconverter e Downconverter;
- Podemos ver o mapeamento dos bits modulados após o mesmo sofrer as degradações impostas pelo canal através de pontos em uma constelação pelo bloco Received Constellation;
- Outro teste válido é alterar o tipo de modulação utilizada para fins de comparação;
- Observar a taxa de erro de bit, bem como o número de bits errados e transmitidos;
- Observar o diagrama de olho dos símbolos transmitidos após os mesmos terem passado pelo canal;
- Analisar a constelação dos símbolos transmitidos alterando da relação sinal-ruído com a simulação rodando para fins de comparação;
- Observar o valor RMS do EVM(Error Vector Magnitude), que corresponde a diferença em vetores entre o sinal de entrada e o sinal recebido.

Podemos ativar ou desativar a interferência co-canal e observar as seguintes características:

### OFF

- Podemos ver após o processo de Upconverter, que no espectro esta interferência cai completamente fora da banda do sinal desejado.

### ON

- Diante da não-linearidade, a terceira harmônica da interferência cai na banda do sinal desejado.
- Podemos observar após o processo de Downconverter, que no espectro você pode ver o tom de interferência presente com o sinal de banda base.

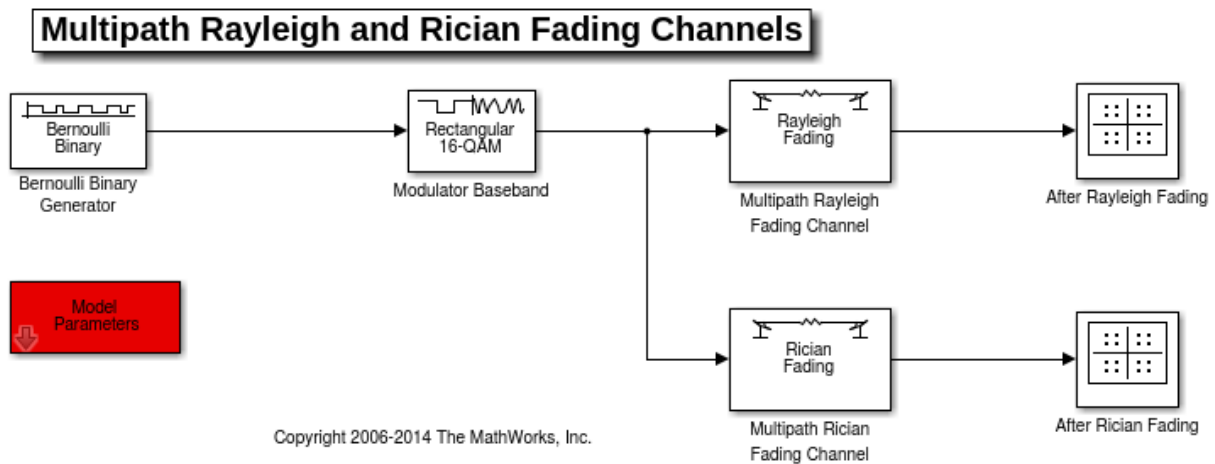
## References

- [1] <http://www.mathworks.com/help/comm/ref/qpskmodulatorbaseband.html>
- [2] <http://www.mathworks.com/help/comm/ref/raisedcosinetransmitfilter.html>
- [3] <http://www.mathworks.com/help/simulink/slref/subsystem.html>
- [4] <http://www.mathworks.com/help/comm/ref/qpskdemodulatorbaseband.html>
- [5] <http://www.mathworks.com/help/simulink/slref/inport.html>
- [6] <http://www.mathworks.com/help/simulink/slref/outport.html>
- [7] <http://www.mathworks.com/help/simulink/slref/mathfunction.html>
- [8] <http://www.mathworks.com/help/simulink/slref/transportdelay.html>
- [9] <http://www.mathworks.com/help/comm/ref/evmmeasurement.html>

# Multipath Fading Channel

## Multipath Fading Channel - Simulink

Este modelo mostra como utilizar os blocos de desvanecimento Rayleigh e Rician do Communications System Toolbox™ fornecido pelo próprio Simulink™. Os canais Rayleigh e Rician são modelos úteis de fenômenos do mundo real em comunicações sem fio. Mais informações [1]



Primeiramente baixe o arquivo a seguir Sistema.zip (que é uma versão parametrizável daquele encontrado na pasta em /opt/MATLAB/R20xxx/toolbox/comm/commdemos). Descompacte e certifique-se que no Matlab você esteja no diretório onde descompactou o arquivo. Digite no terminal do Matlab:

```
open_system('multipathfading')
```

ou

```
multipathfading
```

## Toolbox e blocos necessários

Para realização da simulação, é necessário o **Communications System Toolbox™** html <sup>[1]</sup>, pdf <sup>[2]</sup> fornecido pelo próprio Simulink e os seguintes blocos:

Communications System Toolbox

- Bernoulli Binary Generator <sup>[2]</sup>;
- Digital Baseband Modulation <sup>[3]</sup>;
- Multipath Rayleigh Fading Channel <sup>[2]</sup>;
- Multipath Rician Fading Channel <sup>[3]</sup>;
- Constellation Diagram <sup>[7]</sup>.

Para uma melhor visualização dos blocos e subsistemas usar [Tools> Model Explorer <sup>[11]</sup>]

## Modelo e parâmetros

O modelo simula uma transmissão por frames utilizando modulações digitais em um canal com desvanecimento multipercurso de Rayleigh e Rician. Neste modelo é possível controlar através das variáveis do bloco "Parâmetros do modelo", os parâmetros para realização do mesmo bem como eventual testes. Os parâmetros fornecidos por este bloco são:

Bloco Bernoulli Binary Generator

- bitRate -> Transmission rate (b/s);
- symbolPerFrame -> Number of symbols per transmitted frame.

Bloco Modulador Baseband

- Modulation used -> Choice of modulation used (M-QAM,BPSK,QPSK,M-FSK,M-DPSK);
- M\_ary -> Order modulation.

Blocos Multipath Rayleigh Fading Channel e Multipath Rician Fading Channel

- maxDopplerShift -> Maximum Doppler shift of diffuse components (Hz);
- delayVector -> Discrete delays of channel (s);
- gainVector -> Average path gains (dB);
- LOSDopplerShift -> Doppler shift of line-of-sight component (Hz);
- KFactor -> Ratio of specular power to diffuse power.

## Testes que podem ser feitos

Os testes foram realizados nas versões 2014a e 2015a do software Matlab, funcionando perfeitamente nas mesmas. A seguir alguns testes que podem ser feitos com este modelo.

- Podemos ver o mapeamento dos bits modulados após o mesmo sofrer as degradações impostas pelo canal através de pontos em uma constelação pelos blocos After Rayleigh Fading e After Rician Fading;
- Outro teste válido é alterar o tipo de modulação utilizada para fins de comparação;

Para visualizar os resultados e características do canal Rayleigh simulado, digite no Matlab os seguintes comando:

```
multipathfading
modelname = 'multipathfading';
rayleigh_block = [modelname '/Multipath Rayleigh Fading Channel'];
rician_block = [modelname '/Multipath Rician Fading Channel'];
set_param(rayleigh_block, 'openVisAtStart', 'on');
set_param(rician_block, 'openVisAtStart', 'off');
Tsim = 20;
sim(modelname, Tsim);
```

Resultados e características visualizadas:

- Resposta ao impulso do canal;
- Resposta em frequência do canal;
- Resposta ao impulso do canal em cascata;
- Função de espalhamento;
- Trajetória fasorial;
- Ganhos de cada percurso;
- Componentes de multipercurso;
- Espectro Doppler;

Para visualizar os mesmos resultados e características do canal Rician simulado, digite no Matlab os seguintes comando:

```
multipathfading
modelname = 'multipathfading';
rayleigh_block = [modelname '/Multipath Rayleigh Fading Channel'];
rician_block = [modelname '/Multipath Rician Fading Channel'];
set_param(rayleigh_block, 'openVisAtStart', 'off');
set_param(rician_block, 'openVisAtStart', 'on');
Tsim = 20;

sim(modelname, Tsim);
```

## References

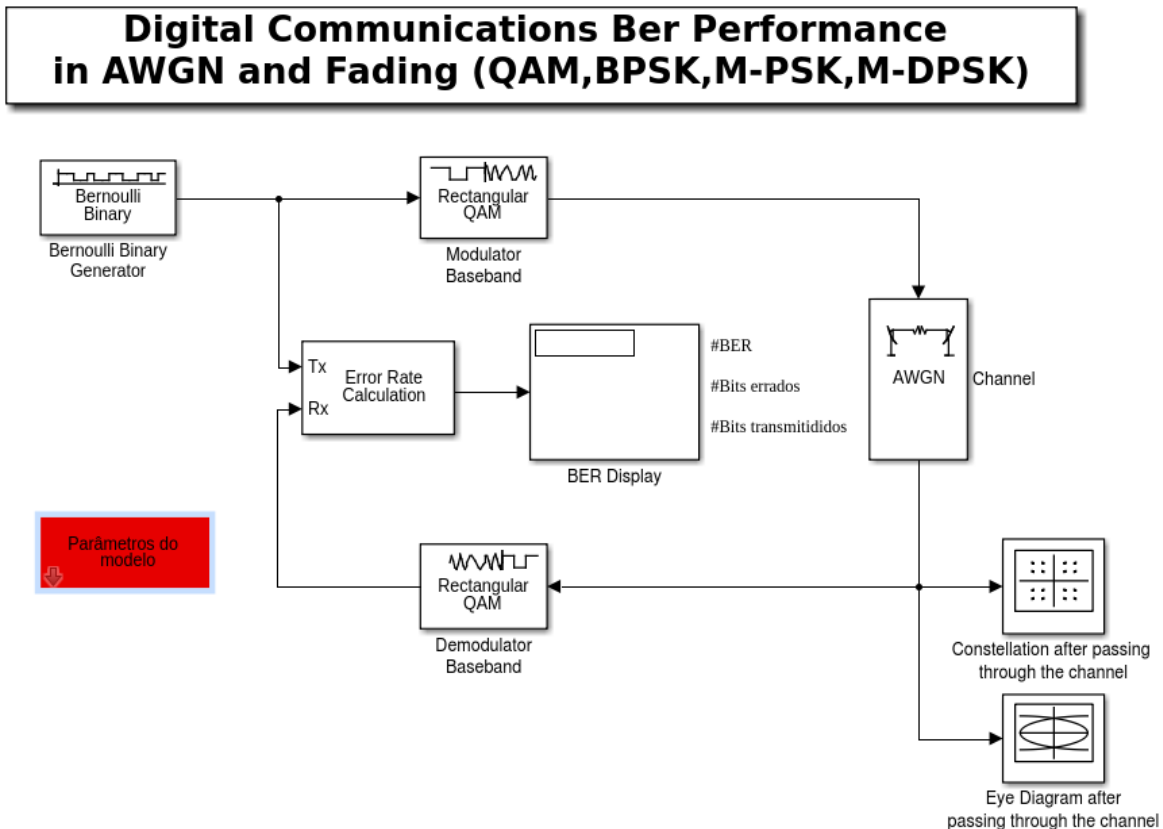
- [1] <http://www.mathworks.com/examples/simulink-communications/2800-multipath-fading-channel>
- [2] <http://www.mathworks.com/help/comm/ref/multipathrayleighfadingchannel.html>
- [3] <http://www.mathworks.com/help/comm/ref/multipathricianfadingchannel.html>

# Digital Communications Ber Performance in AWGN or Fading

---

## BER Performance - Simulink

Um dos critérios mais utilizados para analisar o desempenho de sistemas digitais é a taxa de bits errados (BER - Bit Error Rate) definida como a probabilidade de identificação incorreta de um bit pelo circuito de decisão do receptor. A BER faz uma comparação dos bits transmitidos com os recebidos, computando o número de erros e a taxa de erros dos mesmos. Trata-se de uma medida bastante importante para situações de perda de potência e distorções no sinal transmitido devido principalmente ao meio de propagação. Este modelo considera o desempenho de BER de algumas das principais técnicas de modulação digital utilizadas em canais AWGN e canais com desvanecimento por multipercurso (Rayleigh e Rician). São elas: M-QAM, BPSK, QPSK, M-PSK e M-DPSK.



Primeiramente baixe o arquivo a seguir Sistema.zip. Descompacte e certifique-se que no Matlab você esteja no diretório onde descompactou o arquivo. Digite no terminal do Matlab:

```
open_system('ber_performance')
```

ou

```
ber_performance
```

## Toolbox e blocos necessários

Para realização da simulação, é necessário o **Communications System Toolbox™** html <sup>[1]</sup>, pdf <sup>[2]</sup> fornecido pelo próprio Simulink e os seguintes blocos:

Simulink

- Display <sup>[1]</sup>.

Communications System Toolbox

- AWGN Channel <sup>[2]</sup>;
- Bernoulli Binary Generator <sup>[2]</sup>;
- Digital Baseband Modulation <sup>[3]</sup>;
- Multipath Rayleigh Fading Channel <sup>[2]</sup>;
- Multipath Rician Fading Channel <sup>[3]</sup>;
- Error Rate Calculation <sup>[5]</sup>;
- Constellation Diagram <sup>[7]</sup>;
- Eye Diagram <sup>[3]</sup>.

Para uma melhor visualização dos blocos e subsistemas usar [Tools> Model Explorer <sup>[11]</sup>]



## Modelo e parâmetros

O modelo simula uma transmissão por frames utilizando modulações digitais em um canal com ruído AWGN ou com desvanecimento por multipercurso (Rayleigh ou Rician). Neste modelo é possível controlar através das variáveis do bloco "Parâmetros do modelo", os parâmetros para realização do mesmo bem como eventual testes. Os parâmetros fornecidos por este bloco são:

### Bloco Bernoulli Binary Generator

- bitRate -> Taxa de transmissão (b/s);
- symbolPerFrame -> Número de símbolos por frame.

### Bloco Modulador Baseband and Demodulator Baseband

- Modulação digital utilizada -> Escolha da modulação usada (M-QAM,BPSK,QPSK,M-FSK,M-DPSK);
- M\_ary -> Ordem da modulação.

### Bloco Channel

- maxDopplerShift -> Maximum Doppler shift of diffuse components (Hz);
- delayVector -> Discrete delays of channel (s);
- gainVector -> Average path gains (dB);
- LOSDopplerShift -> Doppler shift of line-of-sight component (Hz);
- KFactor -> Ratio of specular power to diffuse power;
- inputSignal -> Potência do sinal de entrada (watts);
- snr -> Relação sinal-ruído (SNR);
- EbNo -> Relação sinal-ruído (Eb/No).

## Testes que podem ser feitos

Os testes foram realizados nas versões 2014a e 2015a do software Matlab, funcionando perfeitamente nas mesmas. A seguir alguns testes que podem ser feitos com este modelo. Você pode alterar alguns parâmetros mesmo que a simulação esteja rodando, basta acessar novamente o bloco parâmetros do modelo.

- Podemos ver o mapeamento dos bits modulados após o mesmo sofrer as degradações impostas pelo canal através de pontos em uma constelação pelo bloco Constellation after passing through the channel;
  - Outro teste válido é alterar o tipo de modulação utilizada para fins de comparação;
  - Observar a taxa de erro de bit, bem como o número de bits errados e transmitidos;
  - Observar o diagrama de olho dos símbolos transmitidos após os mesmos terem passado pelo canal;
  - Analisar a constelação dos símbolos transmitidos alterando da relação sinal-ruído com a simulação rodando para fins de comparação.
-

# Orthogonal Frequency Multiplex Division (OFDM)

## Orthogonal Frequency Multiplex Division (OFDM) - Simulink

A OFDM (*Orthogonal Frequency Division Multiplexing*) é uma técnica multi-portadora bastante difundida em sistemas digitais, principalmente nos sistemas sem fio devido sua robustez contra a seletividade em frequência do canal. A técnica consiste na divisão da largura de banda total em subcanais menores, utilizando subportadoras ortogonais entre si, possibilitando a sobreposição espectral das mesmas. A divisão em subcanais faz com que a taxa de transmissão seja reduzida em cada subcanal. Assim tem-se um aumento da duração do símbolo transmitido em cada subportadora, o que implica em uma diminuição da sensibilidade ao desvanecimento seletivo em frequência.

Primeiramente baixe o arquivo a seguir Sistema.zip. Descompacte e certifique-se que no Matlab você esteja no diretório onde descompactou o arquivo. Digite no terminal do Matlab:

```
open_system('System_OFDM')
```

ou

```
System_OFDM
```

## Toolbox e blocos necessários

Para realização da simulação, é necessário o **Communications System Toolbox™** [html](#) <sup>[1]</sup>, [pdf](#) <sup>[2]</sup> fornecido pelo próprio Simulink e os seguintes blocos:

Simulink

- Reshape <sup>[1]</sup>;
- Display <sup>[1]</sup>.

Communications System Toolbox

- AWGN Channel <sup>[2]</sup>;
- Random Integer Generator <sup>[1]</sup>;
- Digital Baseband Modulation <sup>[3]</sup>;
- Multipath Rayleigh Fading Channel <sup>[2]</sup>;
- Multipath Rician Fading Channel <sup>[3]</sup>;
- OFDM Modulator Baseband <sup>[2]</sup>;
- OFDM Demodulator Baseband <sup>[3]</sup>;
- Error Rate Calculation <sup>[5]</sup>;
- Constellation Diagram <sup>[7]</sup>;
- Eye Diagram <sup>[3]</sup>.

Para uma melhor visualização dos blocos e subsistemas usar [**Tools> Model Explorer** <sup>[11]</sup>]

## Modelo e parâmetros

O modelo simula um sistema OFDM genérico, a partir de uma sequência de inteiros transmitidos. Primeiramente, o sinal de informação é modulado digitalmente de acordo com o tipo de modulação específica, seja ela QAM ou PSK. Em seguida, é realizada uma conversão do sinal serial-paralelo (S/P) em NFFT feixes de dados onde cada sinal modula uma subportadora gerando um vetor de NFFT sub-símbolos que formam o símbolo OFDM. Essas subportadoras passam pelo bloco *OFDM Modulator* onde é realizada a geração dos símbolos OFDM, a inserção do

prefixo cíclico e a conversão paralelo-serial para transmissão pelo canal. Após a passagem pelo canal combinado com ruído AWGN o sinal chega ao receptor. Nele ocorre o processo inverso do transmissor, o sinal é novamente convertido para serial-paralelo, é removido o prefixo cíclico e a detecção dos símbolos OFDM. Por fim é realizado a demodulação dos sub-símbolos e a recepção do sinal. Neste modelo é possível controlar através das variáveis do bloco "Parâmetros do modelo", os parâmetros para realização do mesmo bem como eventual testes. Os parâmetros fornecidos por este bloco são:

- NFFT -> Número de subportadoras de dados do sistema;
- Cp -> Comprimento do prefixo cíclico;
- Simbolos\_OFDM -> Número de símbolos OFDM transmitidos por *frame*;
- Rs -> Taxa de símbolo (b/s);
- Modulação digital utilizada -> Escolha da modulação usada (M-QAM,BPSK,QPSK);
- M\_ary -> Ordem da modulação;
- maxDopplerShift -> Maximum Doppler shift of diffuse components (Hz);
- delayVector -> Discrete delays of channel (s);
- gainVector -> Average path gains (dB);
- snr -> Relação sinal-ruído (SNR);
- EbNo -> Relação sinal-ruído (Eb/No).

## Testes que podem ser feitos

Os testes foram realizados nas versões 2014a e 2015a do software Matlab, funcionando perfeitamente nas mesmas.

- Análise de sistemas OFDM com altas taxas de transmissão e com canais com múltiplos caminhos;
- Comparação do Sistema OFDM com sistemas de unica portadora, tanto em canais apenas com ruído AWGN, quanto para canais com múltiplos caminhos;
- Alteração do número de subportadoras do sistema;
- Comparação de resultados para diferentes modulações (ou ordem de modulação);
- Alteração da taxa de transmissão para valores menores e maiores relatando os resultados;
- Seria interessante a utilização de subportadoras pilotos para estimação e equalização de canal. Você pode habilitar essas subportadoras pelo bloco *OFDM Modulator Baseband*.

## References

- [1] <http://www.mathworks.com/help/simulink/slref/reshape.html>
- [2] <http://www.mathworks.com/help/comm/ref/ofdmmodulatorbaseband.html>
- [3] <http://www.mathworks.com/help/comm/ref/ofdmmodulatorbaseband.html>

# Article Sources and Contributors

**INCENTIVANDO O USO DO SIMULINK NOS CURSOS DE TELECOMUNICAÇÕES** *Source:* <http://wiki.sj.ifsc.edu.br/wiki/index.php?oldid=93342> *Contributors:* Moecke

**Modulações Analógicas** *Source:* <http://wiki.sj.ifsc.edu.br/wiki/index.php?oldid=98272> *Contributors:* Rogerio.j

**Amostragem de sinais** *Source:* <http://wiki.sj.ifsc.edu.br/wiki/index.php?oldid=98300> *Contributors:* Moecke, Rogerio.j

**Quantização** *Source:* <http://wiki.sj.ifsc.edu.br/wiki/index.php?oldid=98283> *Contributors:* Rogerio.j

**Modulação por Codificação de Pulso (PCM)** *Source:* <http://wiki.sj.ifsc.edu.br/wiki/index.php?oldid=98275> *Contributors:* Rogerio.j

**Modulação Delta (DM)** *Source:* <http://wiki.sj.ifsc.edu.br/wiki/index.php?oldid=98276> *Contributors:* Rogerio.j

**Modulação Delta-Sigma** *Source:* <http://wiki.sj.ifsc.edu.br/wiki/index.php?oldid=98277> *Contributors:* Rogerio.j

**Modulação por Codificação Diferencial de Pulso(DPCM)** *Source:* <http://wiki.sj.ifsc.edu.br/wiki/index.php?oldid=98278> *Contributors:* Rogerio.j

**Códigos de linha** *Source:* <http://wiki.sj.ifsc.edu.br/wiki/index.php?oldid=98279> *Contributors:* Rogerio.j

**Modulação por Amplitude de Pulso M-ária (PAM-M)** *Source:* <http://wiki.sj.ifsc.edu.br/wiki/index.php?oldid=98029> *Contributors:* Rogerio.j

**Transmissão digital em banda base** *Source:* <http://wiki.sj.ifsc.edu.br/wiki/index.php?oldid=98280> *Contributors:* Rogerio.j

**Modulações digitais binárias** *Source:* <http://wiki.sj.ifsc.edu.br/wiki/index.php?oldid=98212> *Contributors:* Rogerio.j

**Modulações digitais M-árias** *Source:* <http://wiki.sj.ifsc.edu.br/wiki/index.php?oldid=98271> *Contributors:* Rogerio.j

**Passband Modulation** *Source:* <http://wiki.sj.ifsc.edu.br/wiki/index.php?oldid=98281> *Contributors:* Rogerio.j

**Multipath Fading Channel** *Source:* <http://wiki.sj.ifsc.edu.br/wiki/index.php?oldid=97774> *Contributors:* Moecke, Rogerio.j

**Digital Communications Ber Performance in AWGN or Fading** *Source:* <http://wiki.sj.ifsc.edu.br/wiki/index.php?oldid=97775> *Contributors:* Rogerio.j

**Orthogonal Frequency Multiplex Division (OFDM)** *Source:* <http://wiki.sj.ifsc.edu.br/wiki/index.php?oldid=98268> *Contributors:* Rogerio.j

# Image Sources, Licenses and Contributors

**Arquivo:Analog\_modulation\_simulink.png** *Source:* [http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Analog\\_modulation\\_simulink.png](http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Analog_modulation_simulink.png) *License:* unknown *Contributors:* Rogerio.j

**Arquivo:amostragem\_sinais\_simulink.png** *Source:* [http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Amostragem\\_sinais\\_simulink.png](http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Amostragem_sinais_simulink.png) *License:* unknown *Contributors:* Rogerio.j

**Arquivo:quantizer.png** *Source:* <http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Quantizer.png> *License:* unknown *Contributors:* Rogerio.j

**Arquivo:sistema\_pcm\_simulink.png** *Source:* [http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Sistema\\_pcm\\_simulink.png](http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Sistema_pcm_simulink.png) *License:* unknown *Contributors:* Rogerio.j

**Arquivo:dm\_simulink.png** *Source:* [http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Dm\\_simulink.png](http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Dm_simulink.png) *License:* unknown *Contributors:* Rogerio.j

**Arquivo:sdm\_simulink.png** *Source:* [http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Sdm\\_simulink.png](http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Sdm_simulink.png) *License:* unknown *Contributors:* Rogerio.j

**Arquivo:dpcm\_simulink.png** *Source:* [http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Dpcm\\_simulink.png](http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Dpcm_simulink.png) *License:* unknown *Contributors:* Rogerio.j

**Arquivo:código\_linha\_simulink.png** *Source:* [http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Código\\_linha\\_simulink.png](http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Código_linha_simulink.png) *License:* unknown *Contributors:* Rogerio.j

**Arquivo:M\_PAM\_simulink.png** *Source:* [http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:M\\_PAM\\_simulink.png](http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:M_PAM_simulink.png) *License:* unknown *Contributors:* Rogerio.j

**Arquivo:banda\_base.png** *Source:* [http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Banda\\_base.png](http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Banda_base.png) *License:* unknown *Contributors:* Rogerio.j

**Arquivo:modulation\_binary.png** *Source:* [http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Modulation\\_binary.png](http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Modulation_binary.png) *License:* unknown *Contributors:* Rogerio.j

**Arquivo:digital\_mod\_M.png** *Source:* [http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Digital\\_mod\\_M.png](http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Digital_mod_M.png) *License:* unknown *Contributors:* Rogerio.j

**Arquivo:modulação\_banda\_passante\_simulink.png** *Source:* [http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Modulação\\_banda\\_passante\\_simulink.png](http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Modulação_banda_passante_simulink.png) *License:* unknown *Contributors:* Rogerio.j

**Arquivo:Multipath\_Fading\_Channels\_simulink.png** *Source:* [http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Multipath\\_Fading\\_Channels\\_simulink.png](http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Multipath_Fading_Channels_simulink.png) *License:* unknown *Contributors:* Rogerio.j

**Arquivo:Ber\_performance\_simulink.png** *Source:* [http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Ber\\_performance\\_simulink.png](http://wiki.sj.ifsc.edu.br/wiki/index.php?title=Arquivo:Ber_performance_simulink.png) *License:* unknown *Contributors:* Rogerio.j

# Licença

---

Attribution-NonCommercial-ShareAlike 3.0 Unported  
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

---