

**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

---

*Estágio Supervisionado em Engenharia Elétrica*

**SIMULAÇÃO DE SISTEMAS DE COMUNICAÇÕES**

**JOSENILDO FERREIRA GALDINO**

Campina Grande - PB  
Março de 2009

**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE**  
**CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA**  
**DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

---

---

*Estágio Supervisionado em Engenharia Elétrica*

**Dados do Estagiário**

Nome: Josenildo Ferreira Galdino  
Registro Acadêmico: 20311253  
Período: 2008.2

**Dados do Local de Estágio**

Local: Laboratório de Comunicações /DEE/UFCG  
Supervisor: Edmar Candeia Gurjão

**Período de Estágio**

Início: 02 / 02 / 2009      Término: 23 / 03 / 2009  
Carga Horária: 120 horas

Campina Grande - PB  
27 de Março de 2009

**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE**  
**CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA**  
**DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

---

---

A comissão examinadora abaixo relacionada aprova o relatório de Estágio Supervisionado, elaborado pelo acadêmico,

**JOSENILDO FERREIRA GALDINO**

Como requisito parcial para a obtenção do grau de **Engenheiro Eletricista**.

**Banca Examinadora**

---

Prof. Edmar Candeia Gurjão, Dr. Eng - Orientador

---

Prof. Bruno Barbosa Albert, Dr. Eng

Campina Grande - PB  
27 de Março de 2009

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

G149e

2009 Galdino, Josenildo Ferreira.

Simulação de Sistemas de Comunicações / Josenildo Ferreira  
Galdino. — Campina Grande, 2009.

78 f. : il

Estágio (Estágio Supervisionado em Engenharia Elétrica) –  
Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e  
Informática.

Referências.

Orientador: Dr. Edmar Candeia Gurjão.

1. Comunicações Digitais. 2. PSK. 3. QAM. I. Título.

CDU- 621.376 (043)

Aos meus pais, familiares e amigos.

# Agradecimentos

Desejo expressar os meus sinceros agradecimentos.

Primeiramente, agradeço a DEUS por proporcionar-me a conclusão de mais uma etapa da vida que se consuma neste trabalho.

Aos meus pais, José Galdino Fernandes e Aurenir Freitas Fernandes, aos meus irmãos Juraci, Jurandir e Joelma, agradeço todo o amor, carinho, respeito e compreensão perante uma infinidade de finais de semana integralmente consumidos pelos estudos.

Ao Professor Edmar Candeia Gurjão, sou grato não só pela orientação, mas pela confiança em mim depositada, e pela experiência que me foi concedida como monitor do Laboratório de Princípios de Comunicações. Agradeço pelas discussões e reflexões que possibilitaram o enriquecimento e realização deste trabalho.

Aos amigos, em especial, Bruno, Célio, Cícero, Cláudio, Delleon, Elieber, Emanuel, Emanuela, Ester, Ivson, Jocélio, Josy, Karolie, Kelly, Lidiane, Luana Mércia, Nelson, Niedja, Rafaela, Rosilda, Thiago, Valentina, Vanessa e Wilker e, que incentivaram e contribuíram para o sucesso deste trabalho. Agradeço também aos meus alunos do Motiva e do Cursinho Pré-Vestibular Solidário da UFCG pelo apoio e amizade, e a todos os meus amigos da faculdade.

Aos primos, em especial, Antonio Manuel, Bárbara, Bruna, Daniela, Edinalva, Fátima, Francisco, Geraldo, Jacinta(Angel), José, Kassandra, Kelen, Lúcia, Maria José, Neide, Raquel, Reginaldo, Rowania e Paulo. Enfim, todos aqueles que me incentivaram e acreditaram em mim.

Aos professores da graduação, pelas valiosas críticas e sugestões.

Aos funcionários da coordenação de Engenharia Elétrica.

## **Resumo**

Este trabalho tem como finalidade abordar alguns sistemas de comunicações digitais e esquemas de modulação convencionais, tais como, PSK (do inglês Phase-Shift Keying e QAM (do inglês Quadrature Amplitude Modulation). Fazer um estudo comparativo por meio de simulação computacional e observar alguns parâmetros de desempenho de tais sistemas. O estágio objetiva o estudo e simulação destes sistemas com a elaboração de guias de experimentos que abordasse alguns conceitos relevantes na área de comunicações digitais. Utilizou-se a divisão de um sistema em subsistemas(blocos), com cada um destes realizando uma ação específica. A interconexão de blocos é uma maneira inteligente e prática de simulação.

**Palavras-chave:** comunicações digitais, PSK, QAM.

# SUMÁRIO

<b>1 – Introdução .....</b>	<b>01</b>
1.1 Objetivo Geral .....	02
1.2 Objetivo Específico .....	02
<b>2 – Atividades Desenvolvidas .....</b>	<b>02</b>
<b>3 – Considerações Finais.....</b>	<b>03</b>
<b>4 – Referências Bibliográficas .....</b>	<b>04</b>
<b>5 – Anexos .....</b>	<b>05</b>
<b>5.1 – Série de Fourier .....</b>	<b>06</b>
<b>5.2 – PSK Usando o Bertool .....</b>	<b>10</b>
<b>5.3 – PSK Usando o Simulink .....</b>	<b>18</b>
<b>5.4 – QAM Usando o Bertool .....</b>	<b>26</b>
<b>5.5 – QAM Usando o Simulink .....</b>	<b>32</b>
<b>5.6 – PSK e QAM .....</b>	<b>41</b>



## 1. Introdução

Neste trabalho é apresentado o relatório sobre o estágio supervisionado em engenharia elétrica, realizado no Laboratório de Comunicações - LABCOM da Universidade Federal de Campina Grande, sob a orientação e supervisão do Professor Dr. Edmar Candeia Gurjão.

Este estágio consistiu em estudar alguns dos sistemas de comunicações convencionais e elaborar guias de experimentos para algumas disciplinas voltadas para a área de telecomunicações. Os experimentos podem ser realizados por meio de simulação computacional com o uso do software MATLAB, amplamente utilizado nas engenharias. Apesar das facilidades oferecidas pelo MATLAB, iniciantes podem passar por dificuldades, se não contarem com o apoio de usuários mais experientes ou de um curso que apresente os seus princípios básicos. Neste ínterim, foi elaborado um tutorial para consulta de alguns comandos básicos do MATLAB.

O estágio teve como objetivo o estudo, a compreensão e simulação de sistemas de comunicações por meio de simulação computacional. Comparar os resultados obtidos com os teóricos. Levantar as curvas de probabilidade de erro de bit dada uma relação sinal ruído especificada. Com a elaboração dos guias pretende-se explorar os poderosos recursos do MATLAB e enfatizar sua importância na simulação de sistemas. Pretende-se contribuir para o ensino das disciplinas de Engenharia Elétrica voltadas para a área de Telecomunicações, dando um enfoque mais prático e complementando o assunto visto em sala de aula. Os guias elaborados podem ser utilizados em algumas disciplinas do curso de engenharia elétrica, tais como: Princípios de Telecomunicações, Comunicações Digitais e Sistemas de Comunicações.

## 1.1 Objetivo Geral

Este trabalho tem por objetivo simular alguns sistemas de comunicações digitais, por meio de simulação computacional. Comparar estes sistemas por meio de parâmetros de análise de desempenho, como taxa de erro de bit, que serão mencionadas no decorrer deste trabalho.

## 1.2 Objetivo Específico

- Desenvolver guias de experimentos para simulação de sistemas de comunicações digitais.
- Comparar os esquemas de modulações convencionais.
- Analisar e discutir os diversos sistemas de comunicações digitais simulados.
- Proporcionar uma melhoria no processo ensino-aprendizagem de alguns conteúdos relacionados às telecomunicações.

## 2. Atividades Desenvolvidas

Neste estágio foram elaborados seis guias para serem utilizados em laboratório com ambiente de simulação por meio do uso do software MATLAB. A seguir temos a listagem dos experimentos:

- Experimento 1: Série de Fourier
- Experimento 2: PSK Usando o Bertool
- Experimento 3: PSK Usando o Simulink
- Experimento 4: QAM Usando o Bertool
- Experimento 5: QAM usando o Simulink
- Experimento 6: PSK e QAM

### 3. Considerações Finais

Com base no estudo realizado a cerca de alguns sistemas de comunicações, houve grande aprendizado por parte do estagiário nas áreas relacionadas às telecomunicações. Entretanto, houve dificuldades por causa da pequena quantidade de material existente com a parte de simulação de sistemas de comunicações no ambiente MATLAB, grande parte do material disponível refere-se a uma abordagem mais teórica. Este material, portanto, vem a preencher esta lacuna, objetivando uma melhoria na assimilação de conceitos básicos vistos em telecomunicações.

A simulação de alguns sistemas de comunicações com o software MATLAB possibilitou uma maior compreensão de alguns conceitos fundamentais para a minha formação como engenheiro com ênfase em telecomunicações. Este trabalho, portanto, contribuiu para a formação do estagiário com o intuito de se especializar na área de comunicações digitais.

Com a realização deste trabalho espero motivar aos professores e alunos do curso de engenharia elétrica a utilizar com mais frequência o MATLAB, que é considerada uma ferramenta indispensável para as engenharias e em especial, a engenharia elétrica.

Alguns esquemas de modulação convencionais foram simulados com o software MATLAB, dentre eles, temos: PSK(Phase-Shift Keying), QAM(Quadrature Amplitude Modulation), além de um experimento sobre Série de Fourier. Com a elaboração dos guias de experimentos, pôde-se contribuir para o processo ensino-aprendizagem e visa uma melhoria da qualidade de ensino. A utilização dos guias e do software MATLAB pode permitir um novo acesso ao saber de uma maneira distinta, podendo gerar no aluno a curiosidade em aprender, e esta curiosidade, por sua vez, é a essência do conhecimento por possibilitar que o aluno pense e desenvolva o raciocínio lógico e a crítica que são indispensáveis para a aquisição do conhecimento.

#### 4. Referências Bibliográficas

Haykin. S, Sistemas de Comunicação Analógica e Digital, Bookman Companhia Ed, Porto Alegre, 2004

I.A.Glover, P.M.Grant, Digital Communications, Prentice Hall, 2000.

B. P. Lathi, Modern Digital and Analog Communications Systems. Oxford University Press, 1998.

Digital Communications, John G Proakis. McGraw-Hill, 4a Ed., 2000.

MEDEIROS, J.C.O. Princípios de Telecomunicações: teoria e prática. São Paulo: Érica, 2004.

Programação em Matlab para Engenheiros, Stephen J. Chapman, Thomson Pioneira, 2003

Hanselman D, Littlefield B. Matlab 6: Curso Completo. São Paulo, Makron Books, 2003.

Matsumoto EY. Matlab7 : Fundamentos. São Paulo, Érica, 2004.

Curso de Matlab for Windows, Departamento de Engenharia Mecânica, UNESP, Campus de Ilha Solteira. Material disponível em [www.lncc.br/~gazoni/tutorial/Apontamentos\\_de\\_MATLAB\\_INTRO.pdf](http://www.lncc.br/~gazoni/tutorial/Apontamentos_de_MATLAB_INTRO.pdf).

Acessado em 20/09/08.

Introdução a Programação no MATLAB, Curso Apresentado na 4ª Semana de Matemática, Material disponível em <http://www.dme.ufcg.edu.br/uame/arquivos/cursomatlab.ppt>, Acessado em 15/09/08.

Curso de Matlab – Grupo Pet Engenharia Elétrica da Universidade Mato Grosso do Sul Material disponível em <http://www.del.ufms.br/tutoriais/matlab/apresentacao.htm> Acessado em 25/09/08.

## **5. Anexos**

- 5.1 – Série de Fourier
- 5.2 – PSK Usando o Bertool
- 5.3 – PSK Usando o Simulink
- 5.4 – QAM Usando o Bertool
- 5.5 – QAM Usando o Simulink
- 5.6 – PSK e QAM



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
UNIDADE ACADÊMICA DE ENGENHARIA ELÉTRICA**

---

# **SÉRIE DE FOURIER**

**JOSENILDO FERREIRA GALDINO**

Este experimento objetiva implementar a onda quadrada e triangular obtida a partir da Série de Fourier.

A Série de Fourier tem esse nome em homenagem ao matemático e físico francês *Jean-Baptiste Joseph Fourier* (1768-1830) que estudou esta série que é a representação de uma função periódica como uma soma de funções periódicas. As séries de Fourier são formas de representar funções como soma de exponenciais ou senoides. A série de Fourier pode ser descrita matematicamente pela seguinte equação.

$$g(t) = a_0 + \sum_{n=1}^{\infty} [a_n \cos(n \cdot \omega_0 \cdot t) + b_n \text{sen}(n \cdot \omega_0 \cdot t)]$$

Para uma onda quadrada a Serie de Fourier é escrita da seguinte forma:

$$g(t) = a_0 + \cos(\omega_0 \cdot t) - \frac{1}{3} \cos(3 \cdot \omega_0 \cdot t) + \frac{1}{5} \cos(5 \cdot \omega_0 \cdot t) - \frac{1}{7} \cos(7 \cdot \omega_0 \cdot t) + \frac{1}{9} \cos(9 \cdot \omega_0 \cdot t) + \dots$$

O diagrama de blocos para construção do modelo no simulink é mostrado na figura 1.

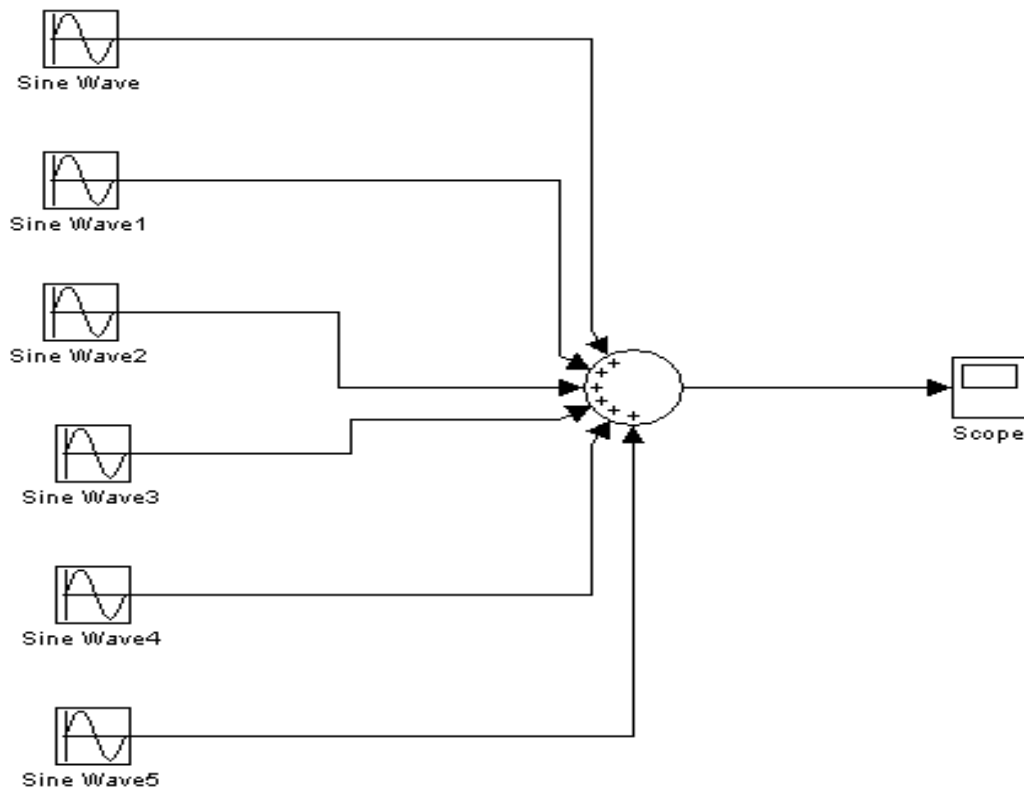


Figura 1: Série de Fourier

Para a simulação da onda quadrada obtida a partir da Serie de Fourier é necessário o uso de alguns blocos, para a busca rápida dos blocos é listada na Tabela 1 os blocos a serem utilizados, bem como a quantidade e o local de busca no simulink.

Tabela 1: Blocos do Simulink

<i>Bloco</i>	<i>Caminho</i>	Quantidade
Sine Wave	Simulink > Sources	6
Sum	Simulink > Math Operations	1
Scope	Simulink > Sinks	1

## Sine Wave

Após montar o esquema da figura 1 o próximo passo é alterar os parâmetros do bloco Sine Wave, conforme a Tabela 2 dada a seguir. Clique no bloco do Sine Wave e altere os parâmetros.

Tabela 2: Configuração do Sine Wave

<i>Bloco</i>	<i>Amplitude</i>	<i>Frequencia (rad/s)</i>	<b>Fase (radianos)</b>
Sine Wave	1	$2 \cdot \pi$	$\pi/2$
Sine Wave1	-1/3	$3 \cdot (2 \cdot \pi)$	$\pi/2$
Sine Wave2	1/5	$5 \cdot (2 \cdot \pi)$	$\pi/2$
Sine Wave3	-1/7	$7 \cdot (2 \cdot \pi)$	$\pi/2$
Sine Wave4	1/9	$9 \cdot (2 \cdot \pi)$	$\pi/2$
Sine Wave5	-1/11	$11 \cdot (2 \cdot \pi)$	$\pi/2$

## Sum



O bloco Sum do Simulink é um somador que na sua configuração padrão tem duas entradas e uma saída. Na figura 1, observe que temos que somar seis sinais, ou seja, seis senoides. Para modificar a configuração padrão do bloco Sum, dê um click com o botão esquerdo do mouse e modifique o parâmetro *List of signs* para |++++++.





Figura 2: Bloco Sum com 2 e 6 entradas

## Simulação

A simulação da onda quadrada no simulink agora pode ser executada. Para isto clique no botão *Start Simulation*, . Em seguida clique no Scope e verifique a forma de onda obtida. Para melhor visualização da curva clique no botão *Autoscale* , que faz o ajuste de escala.

## Questões

1) No simulink crie o modelo mostrado na figura 1 e ajuste os parâmetros conforme os dados da tabela 3. Simule este modelo.

A Série de Fourier de uma onda triangular é dada pela seguinte expressão:

$$g(t) = \cos(w_0 \cdot t) + \frac{1}{3^2} \cos(3 \cdot w_0 \cdot t) + \frac{1}{5^2} \cos(5 \cdot w_0 \cdot t) + \dots$$

Para gerar a onda triangular é preciso apenas modificar as amplitudes das senoides nos blocos Sine Wave. A Tabela 3 mostra os valores assumidos nas respectivas frequências.

Tabela 3: Configuração do Sine Wave

<i>Bloco</i>	<i>Amplitude</i>	<i>Frequencia (rad/s)</i>	<b>Fase (radianos)</b>
Sine Wave	1	2*pi	pi/2
Sine Wave1	1/9	3*(2*pi)	pi/2
Sine Wave2	1/25	5*(2*pi)	pi/2
Sine Wave3	1/49	7*(2*pi)	pi/2
Sine Wave4	1/81	9*(2*pi)	pi/2
Sine Wave5	1/121	11*(2*pi)	pi/2



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
UNIDADE ACADÊMICA DE ENGENHARIA ELÉTRICA**

---

**PSK  
USANDO O BERTOOL**

**JOSENILDO FERREIRA GALDINO**

Neste experimento iremos utilizar o BERTOOL *Bit Error Rate Analysis Tool* para traçar o gráfico da BER em função da relação sinal-ruído dada em decibéis do sistema PSK.

Inicialmente, abra o programa MATLAB, clicando com o botão esquerdo do mouse no ícone que aparece na Área de Trabalho, caso não tenha ícone, procure-o em Arquivos de Programas. Na janela de comandos do MATLAB, digite o comando bertool.

```
>> bertool
```

Ao digitar o comando anterior na janela de comando será exibida uma janela conforme mostrada a figura 1:

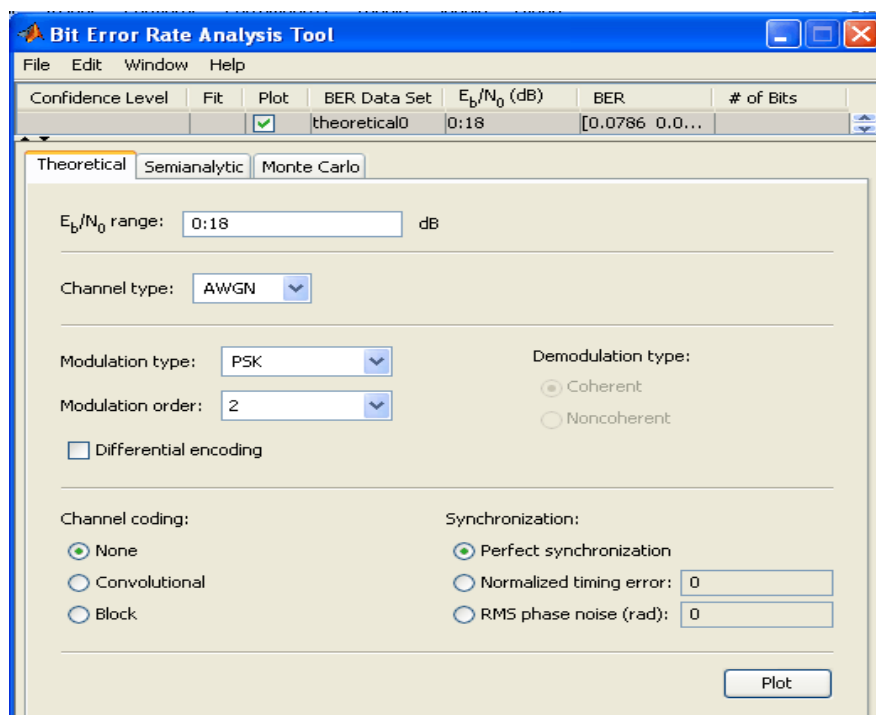


Figura 1: bertool

Na janela anterior pode-se levantar as curvas teóricas Taxa de Erro de Bit (BER – do inglês Bit Error Rate) em função da relação sinal-ruído para diversos esquemas de modulação e diversos modelos de canais de comunicação.

Na configuração padrão do MATLAB, tem-se uma modulação 2-PSK para um canal AWGN com a faixa de 20 dB, para isto modifique o parâmetro  $E_b/N_0$  de 0:18 para 0:20 da janela principal do bertool. Para visualizar a curva teórica do 2-PSK, clique com o botão esquerdo do mouse na opção Plot, localizada no canto inferior direito da janela da Figura 1. Feito o procedimento anterior será exibida uma nova janela com o gráfico conforme mostrada na Figura 2.

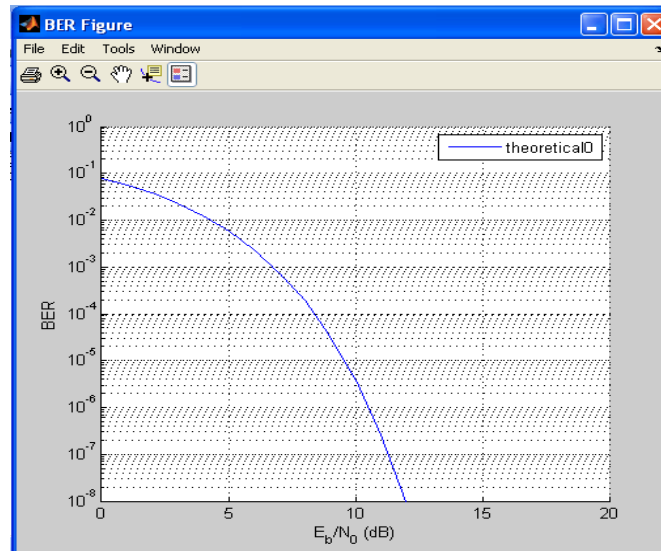


Figura 2: Curva Teórica do 2-PSK

Observe na figura 2, a legenda indica que a curva mostrada é a teórica. Esta janela gráfica possui muitos recursos, dentre eles, temos: Show Legend, Data Cursor, Pan, Zoom Out, Zoom In e Print Figure. A seguir são descritos as funções de cada uma das ferramentas mencionadas.



- Este é o símbolo da ferramenta Show Legend, que permite que seja exibida a Legenda no gráfico. Para ativar este recurso ou desativar basta apenas um clique com o botão esquerdo do mouse.



- Este é o símbolo da ferramenta Data Cursor, que permite movimentarmos na curva obtida e sabermos as suas coordenadas. Na figura 3 é mostrada o uso do Data Cursor, neste caso temos uma razão sinal ruído de 4 dB e uma BER de 0.0125.

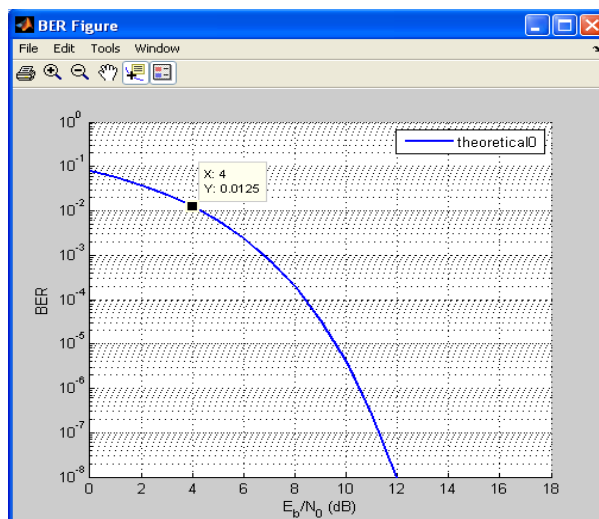






Figura 3: Uso do Data Cursor

 - Este é o símbolo da ferramenta Pan, esta opção permite que façamos um deslocamento horizontal ou vertical para visualizar melhor determinada faixa do gráfico.

 - Este é o símbolo do Zoom In, que permite que façamos uma redução a partir do ponto escolhido.

 - Este é o símbolo do Zoom Out, que permite que façamos uma ampliação a partir do ponto escolhido.

 - Este é o símbolo do Print Figure, que permite que seja feita a impressão da curva obtida caso tenha uma impressora conectada no computador.

## Curva Teórica do 4-PSK

O próximo passo é modificar os parâmetros do bertool para obtermos o desempenho da modulação 4-PSK para um canal AWGN com a faixa de 20 dB, para isto modifique o parâmetro  $E_b/N_0$  de 0:18 para 0:20 da janela principal do bertool. Para visualizar a curva teórica do 4-PSK, clique com o botão esquerdo do mouse na opção Plot, localizada no canto inferior direito da janela da Figura 1. Feito o procedimento anterior será exibida uma nova janela com o gráfico conforme mostrada na Figura 4.

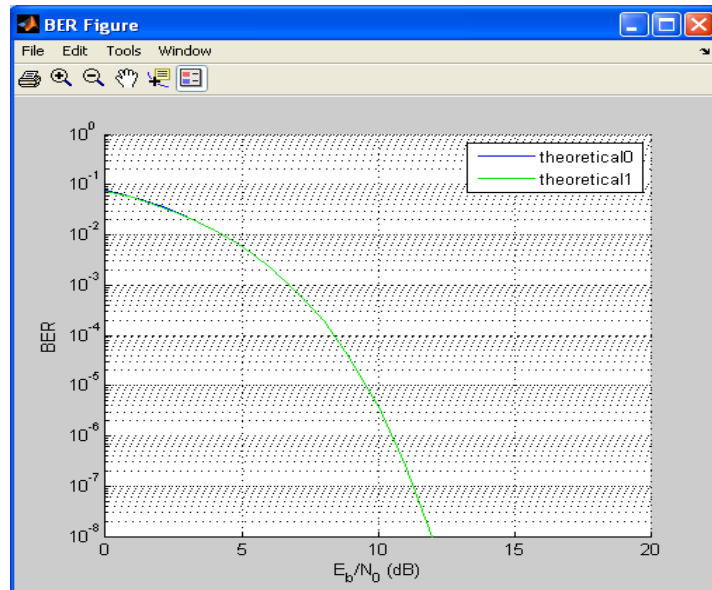


Figura 4: 2-PSK e 4-PSK

Pela figura anterior temos a curva de taxa de erro de bit em função da relação sinal-ruído dada em decibéis. De acordo com a legenda a curva em azul, refere-se ao 2-PSK e a curva em verde ao 4-PSK.

### Curva Teórica do 8-PSK

O próximo passo é modificar os parâmetros do bertool para obtermos o desempenho da modulação 8-PSK para um canal AWGN com a faixa de 20 dB, para isto modifique o parâmetro  $E_b/N_0$  de 0:18 para 0:20 da janela principal do bertool. Para visualizar a curva teórica do 8-PSK, clique com o botão esquerdo do mouse na opção Plot, localizada no canto inferior direito da janela da Figura 1. Feito o procedimento anterior será exibida uma nova janela com o gráfico conforme mostrada na Figura 5.

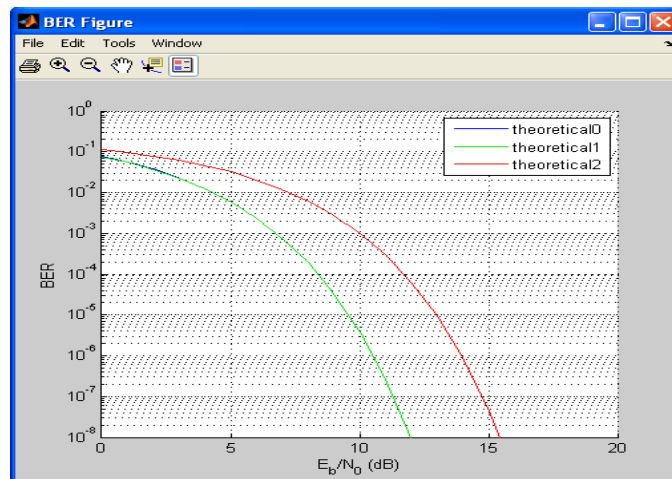


Figura 5: 2-PSK, 4-PSK e 8-PSK.

Pela figura anterior temos a curva de taxa de erro de bit em função da relação sinal-ruído dada em decibéis. De acordo com a legenda a curva em azul, refere-se ao 2-PSK, a curva em verde ao 4-PSK e a curva em vermelho refere-se ao 8-PSK.

### Curva Teórica do 16-PSK

O próximo passo é modificar os parâmetros do bertool para obtermos o desempenho da modulação 16-PSK para um canal AWGN com a faixa de 20 dB, para isto modifique o parâmetro  $E_b/N_0$  de 0:18 para 0:20 da janela principal do bertool. Para visualizar a curva teórica do 16-PSK, clique com o botão esquerdo do mouse na opção Plot, localizada no canto inferior direito da janela da Figura 1. Feito o procedimento anterior será exibida uma nova janela com o gráfico conforme mostrada na Figura 6.

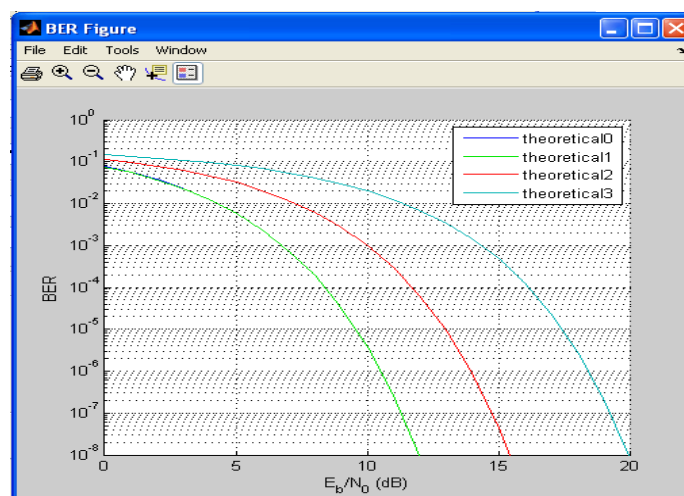


Figura 6: 2-PSK, 4-PSK, 8-PSK e 16-PSK

Pela figura anterior temos a curva de taxa de erro de bit em função da relação sinal-ruído dada em decibéis. De acordo com a legenda a curva em azul, refere-se ao 2-PSK, a curva em verde ao 4-PSK, a curva em vermelho ao 8-PSK e a curva em turquesa ao 16-PSK.

## Questões

As questões a seguir devem ser respondidas, analisando-se os gráficos gerados utilizando-se o bertool.

1) Obtenha a curva teórica para o 2-PSK em um canal AWGN e preencha a tabela dada a seguir. Dica: Use o Data Cursor.

Eb/N0	1	3	5	7	9
BER					

2) A partir da curva teórica do 2-PSK obtida anteriormente, preencha a tabela dada a seguir. Dica: Use o Data Cursor

BER	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$
Eb/N0					

3) Obtenha a curva teórica para o 8-PSK em um canal AWGN e preencha a tabela dada a seguir. Dica: Use o Data Cursor.

Eb/N0	1	3	5	7	9
BER					
Eb/N0	11	12	13	14	15
BER					

4) A partir da curva teórica do 8-PSK obtida anteriormente, preencha a tabela dada a seguir. Dica: Use o Data Cursor

BER	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$
Eb/N0					
BER	$10^{-6}$	$10^{-7}$	$10^{-8}$	$10^{-9}$	$10^{-10}$
Eb/N0					



5) A partir dos dados das tabelas anteriores e dos gráficos gerados no bertool, podemos dizer que o 2-PSK é mais eficiente do que o 4-PSK?. Justifique sua resposta em caso afirmativo ou negativo.



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
UNIDADE ACADÊMICA DE ENGENHARIA ELÉTRICA**

---

**PSK  
USANDO O SIMULINK**

**JOSENILDO FERREIRA GALDINO**

Este experimento objetiva analisar o sistema de modulação PSK por meio do Simulink. Serão feitas simulações com e sem o canal AWGN.

## Esquema de Modulação PSK

Um dos parâmetros importantes para análise de sistemas de comunicações digitais, é a taxa de erro. Iremos criar um modelo de um sistema básico BPSK no ambiente Simulink, conforme o diagrama mostrado na figura 1.

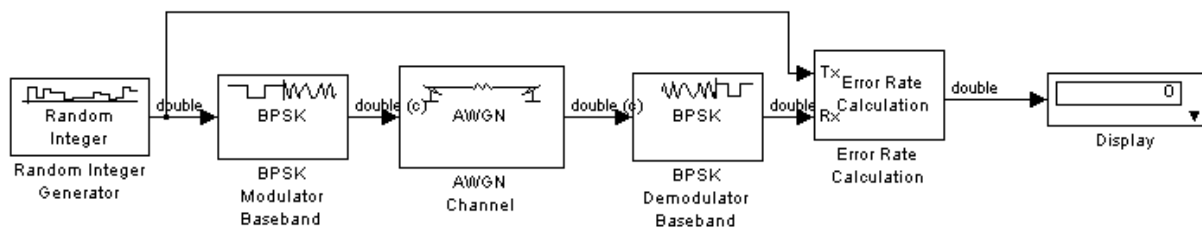


Figura 1: BPSK – Taxa de erro

Para a simulação do sistema BPSK, é necessário o uso de alguns blocos, para a busca rápida destes, é listada na Tabela 1, os blocos a serem utilizados, bem como, a quantidade e o local de busca no simulink.

Tabela 1: Blocos do Simulink

<i>Bloco</i>	<b>Caminho</b>
Random Integer Generator	Simulink > Communications Blockset > Comm Sources > Random Data Sources
BPSK Modulator Baseband	Simulink > Communications Blockset > Modulation > Digital Baseband Modulation >> PM
AWGN Channel	Simulink > Communications Blockset > Channels
BPSK Demodulator Baseband	Simulink > Communications Blockset > Modulation > Digital Baseband Modulation >> PM
Error Rate Calculation	Simulink > Communications Blockset >> Comm Sinks
Display	Simulink > Sinks

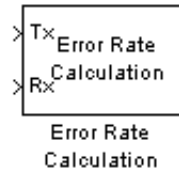
## Alterando os parâmetros

### Random Integer Generator

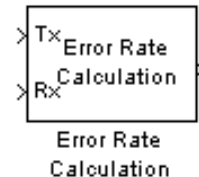
Para gerar símbolos aleatórios para simular um sistema BPSK, altere o parâmetro M-ary number para 2 e clique em OK.

### Error Rate Calculation

Error Rate Calculation exibe os resultados na área de trabalho do MATLAB, entretanto, no ambiente simulink é mais viável exibir os resultados utilizando o bloco display. Neste caso, precisamos acrescentar a porta de saída para o bloco Error, para isto dê um duplo clique com o botão esquerdo do mouse sobre o bloco e modifique o parâmetro *Output data* para *port*, feito isto o bloco modificado será semelhante ao da figura 2 dada a seguir




Este é o bloco padrão do *Error Rate Calculation*, contém 2 portas de entrada.



Este é o bloco *Error Rate Calculation* modificado, contém 2 portas de entrada e uma de saída.

Figura 2: Error Rate Calculation

A simulação do sistema BPSK pode ser executada, para isto, clique no botão *Start Simulation*, .

## Simulação do 4-PSK

Para melhor compreensão da influencia do ruído, em um sistema de comunicação digital do tipo 4-PSK iremos inicialmente simular este esquema de modulação sem o canal AWGN e observar o sinal modulado. Na etapa seguinte será adicionado o canal AWGN.

O diagrama de blocos de um sistema 4-PSK é mostrado na figura 3 dada a seguir. Neste modelo não consideraremos o canal AWGN para podermos visualizarmos como seria o sinal modulado sem a influência do ruído.

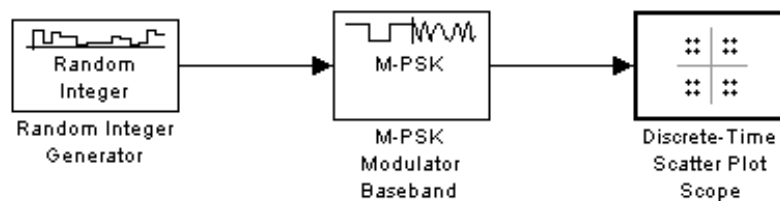


Figura 3: PSK sem canal AWGN


Para esta simulação é necessário o uso de alguns blocos, para a busca rápida dos blocos é listada na Tabela 2, os blocos a serem utilizados, e o local de busca no simulink.

Tabela 2: Blocos do Simulink

<i>Bloco</i>	<b>Caminho</b>
Random Integer Generator	Communication blockset > Comm Sources > Random Data Sources
M-PSK Modulator Baseband	Communication blockset > Modulation > Digital Baseband Modulation > PM
Discrete-Time Scatter Plot Scope	Communication blockset > Comm Sinks
AWGN Channel	Communication blockset > Channels

Para montar o esquema da figura 3 é necessário abrir o simulink. Para isto digite na janela de comandos, o comando mostrado a seguir,

```
>> simulink
```

ou simplesmente, clique no ícone , para ativar o simulink. Em seguida abra um novo documento para poder inserir os blocos neste documento.

Após montar o esquema da figura 3 o próximo passo é alterar alguns dos parâmetros do bloco, conforme a Tabela 3 dada a seguir.

Tabela 3: Parâmetro dos blocos

<i>Bloco</i>	<i>M-ary-number</i>	Phase offset (rad)
Random Integer Generator	4	--
M-PSK Modulator Baseband	4	0

O diagrama de blocos de um sistema 4-PSK sem o canal AWGN é mostrado na figura 3. Acrescente o bloco AWGN conforme modelo descrito na figura 4 e execute a simulação.

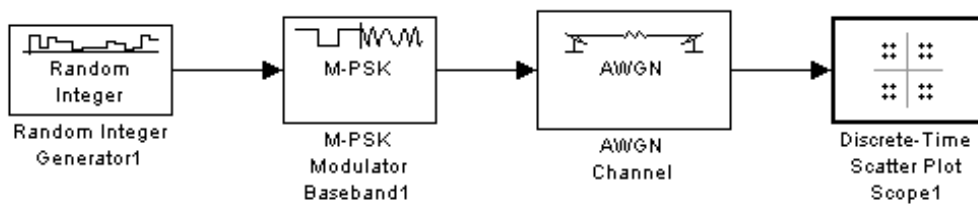



Figura 4: 4-PSK com AWGN

## Simulação

O parâmetro  $E_b/N_0$  do bloco AWGN deve ser alterado para 20 dB. A simulação do 4-PSK no simulink agora pode ser executada. Para isto clique no botão *Start Simulation*, . O resultado é mostrado nas figuras 5 e 6.

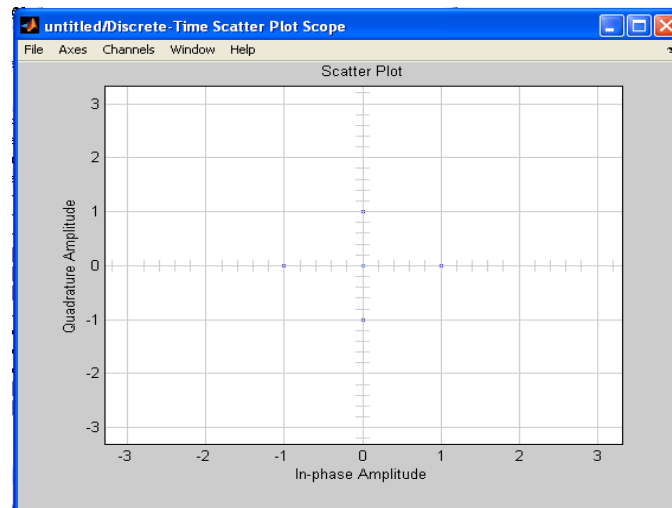


Figura 5: 4-PSK

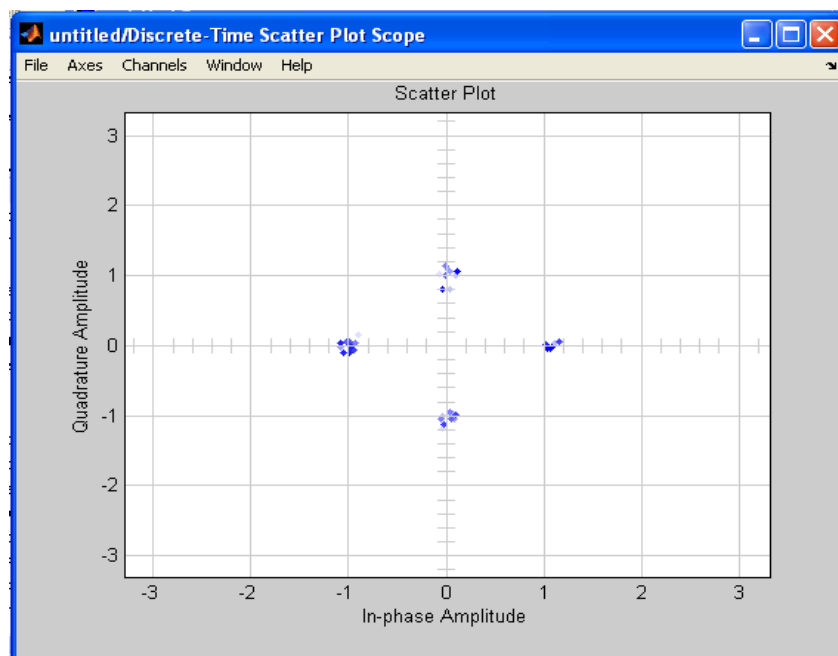


Figura 6: 4-PSK com canal AWGN

## Questões

As questões a seguir devem ser respondidas, utilizando-se o simulink.

1) Simule o esquema da figura 1, e altere o parâmetro *M-ary-number* para 2. Este parâmetro pertence ao bloco Random Integer Generator. Considere um tempo de simulação 100 segundos. Modifique o parâmetro *Symbol period* para 100 segundos. Nesta simulação o canal tem uma relação sinal-ruído de 10 dB. Anote o resultado mostrado no display.

2) Utilizando o esquema simulado anteriormente na questão 1, modifique o parâmetro  $E_b/N_0$  para 20 dB do bloco AWGN Channel, e execute o novo esquema. Anote o resultado mostrado no display e compare com o resultado obtido para o caso de 10 dB.

3) Simule o esquema da figura 7 dada abaixo. Em cima de cada um dos blocos desta figura, estão indicados as modificações a serem feitas nos parâmetros. Considere um tempo de simulação de 100 segundos. O que acontece com os pontos da constelação de sinais, quando variamos a relação sinal ruído. Explique.

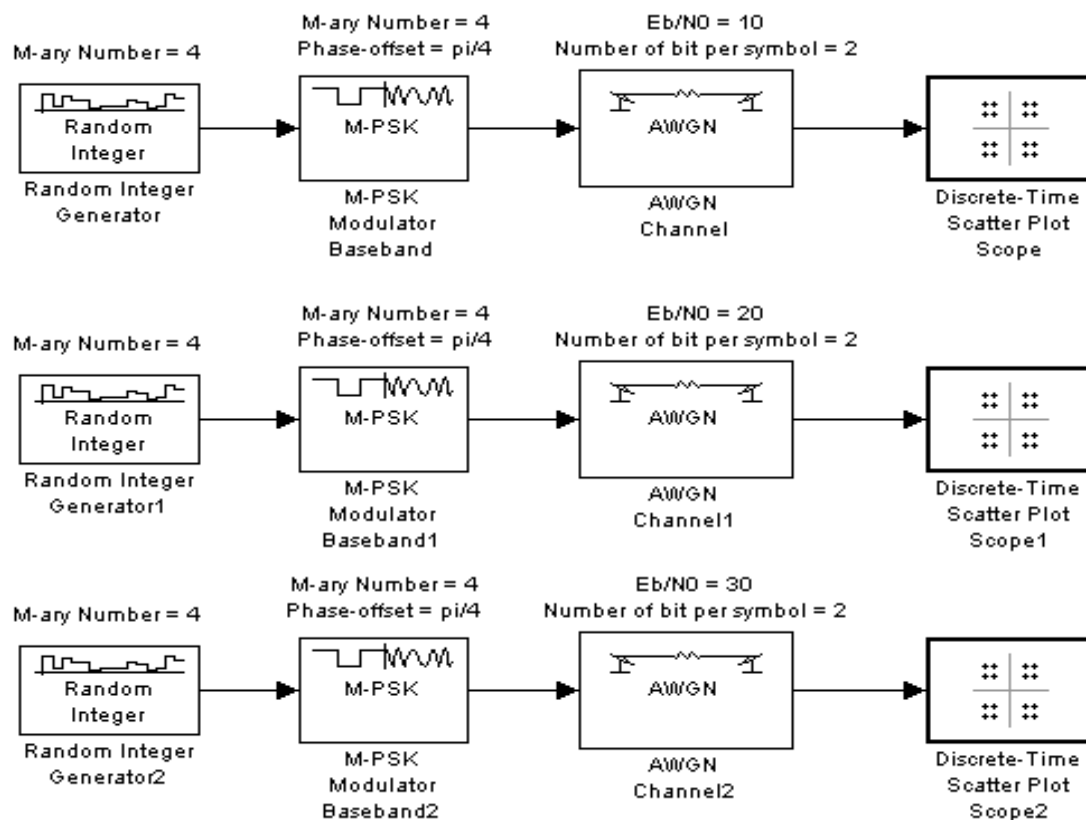


Figura 7: 4-PSK



4) Simule o esquema da figura 8 dada abaixo. Em cima de cada um dos blocos desta figura, estão indicados as modificações a serem feitas nos parâmetros. Considere um tempo de simulação de 100 segundos. O que acontece com os pontos da constelação de sinais, quando variamos a relação sinal ruído. Explique. Faça uma análise comparativa entre o 4-PSK e o 8-PSK, baseando-se nas curvas obtidas.

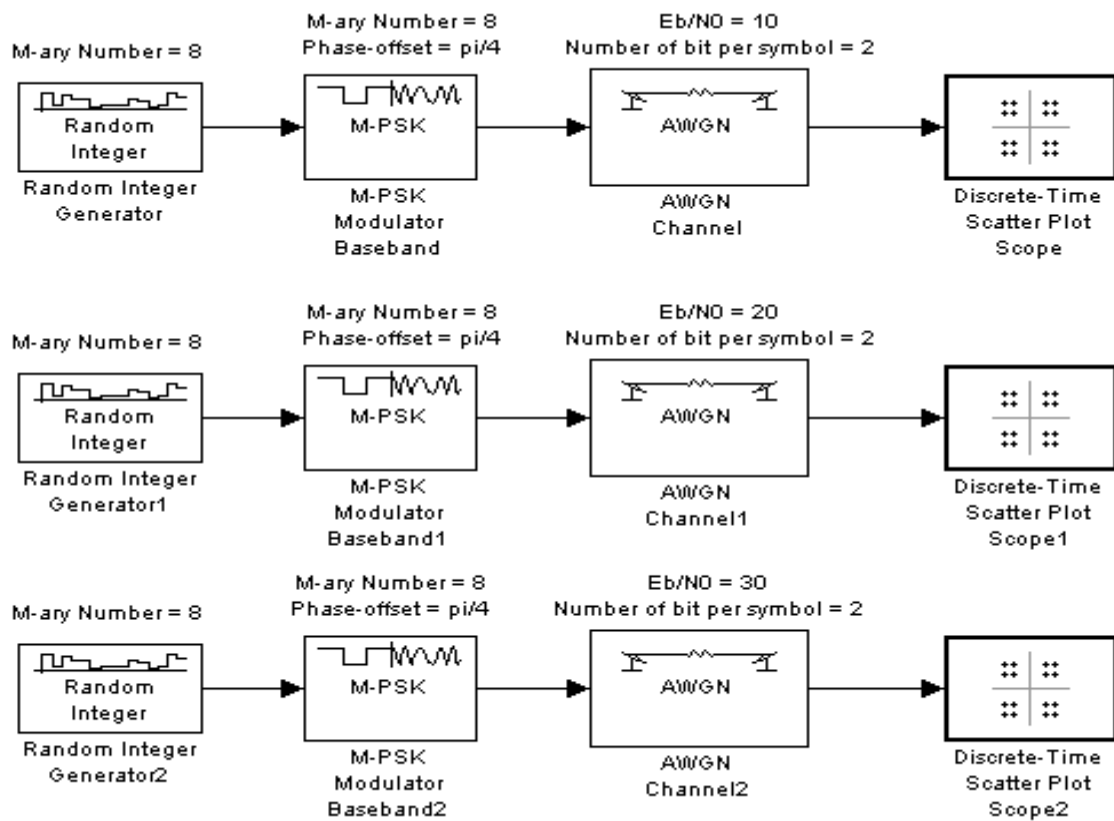


Figura 8: 8-PSK



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
UNIDADE ACADÊMICA DE ENGENHARIA ELÉTRICA**

---

**QAM  
USANDO O BERTOOL**

**JOSENILDO FERREIRA GALDINO**

Neste experimento iremos utilizar o BERTOOL *Bit Error Rate Analysis Tool* para traçar o gráfico da BER em função da relação sinal-ruído dada em decibéis do sistema QAM.

Inicialmente, abra o programa MATLAB, clicando com o botão esquerdo do mouse no ícone que aparece na Área de Trabalho, caso não tenha ícone, procure-o em Arquivos de Programas. Na janela de comandos do MATLAB, digite o comando bertool.

```
>> bertool
```

Ao digitar o comando anterior na janela de comando será exibida uma janela conforme mostrada a figura 1:

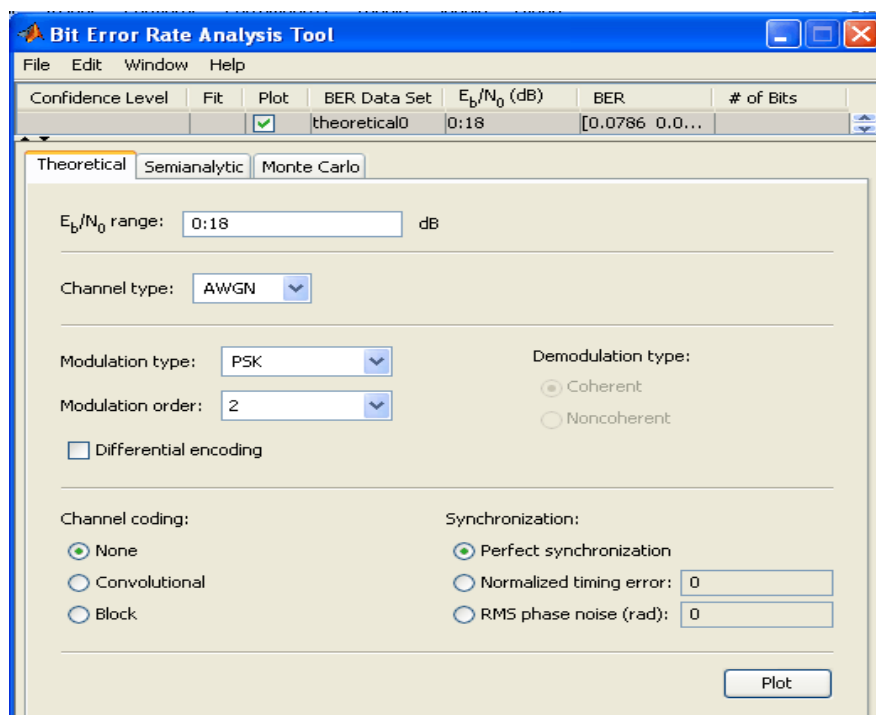


Figura 1: bertool

Na janela anterior pode-se levantar as curvas teóricas Taxa de Erro de Bit (BER – do inglês Bit Error Rate) em função da relação sinal-ruído para diversos esquemas de modulação e diversos modelos de canais de comunicação.

## Curva Teórica do 4-QAM

Para obter a curva teórica do 4-QAM para um canal AWGN em um faixa de 30 dB, modifique os parâmetros dados a seguir com os seguintes valores:  $E_b/N_0 = 0:30$  , *Channel type: AWGN*, *Modulation type: QAM*, *Modulation order: 4*. Alterando-se os parâmetros o resultado é mostrado na Figura 2, dada a seguir.

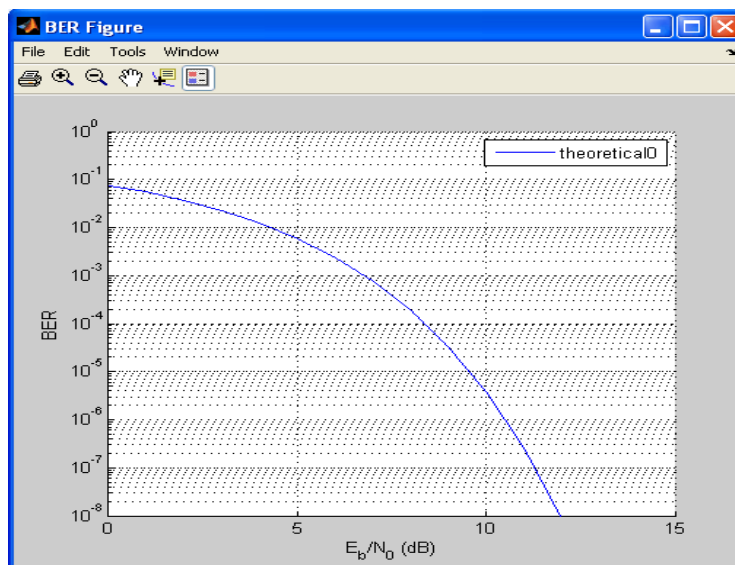


Figura 2: Curva Teórica do 4-QAM

Observe na Figura 2 a legenda indica que a curva mostrada é a theoretical0. Para melhor adequação ao sistema, modifique o nome da legenda para 4-QAM. Basta clicar com o botão esquerdo do mouse sobre a caixa da legenda e digitar “4-QAM”.

## Curva Teórica do 16-QAM

Para obter a curva teórica do 16-QAM para um canal AWGN em um faixa de 30 dB, modifique os parâmetros dados a seguir com os seguintes valores:  $E_b/N_0 = 0:30$  , *Channel type: AWGN*, *Modulation type: QAM*, *Modulation order: 16*. Alterando-se os parâmetros o resultado é mostrado na Figura 3, dada a seguir.

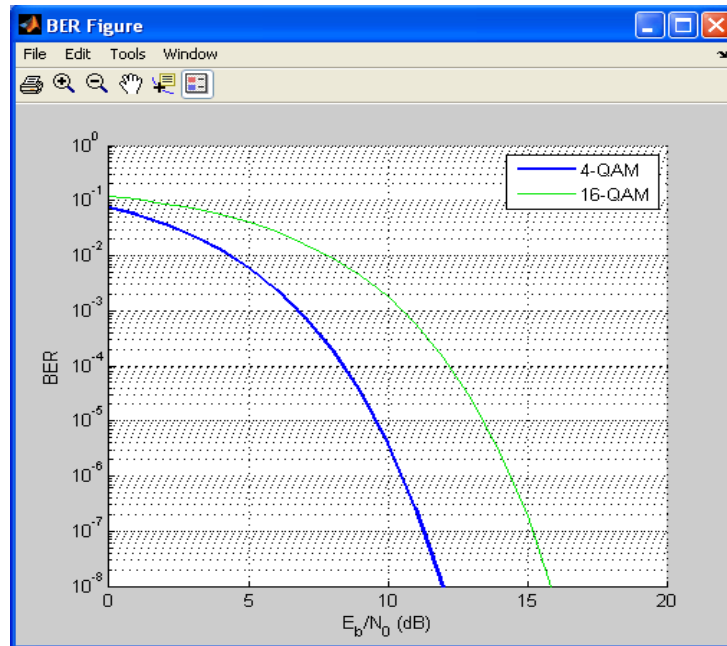


Figura 3: 4-QAM e 16-QAM

### Curva Teórica do 64-QAM

Para obter a curva teórica do 64-QAM para um canal AWGN em um faixa de 30 dB, modifique os parâmetros dados a seguir com os seguintes valores:  $E_b/N_0 = 0:30$ , *Channel type: AWGN*, *Modulation type: QAM*, *Modulation order: 64*. Alterando-se os parâmetros o resultado é mostrado na Figura 4, dada a seguir.

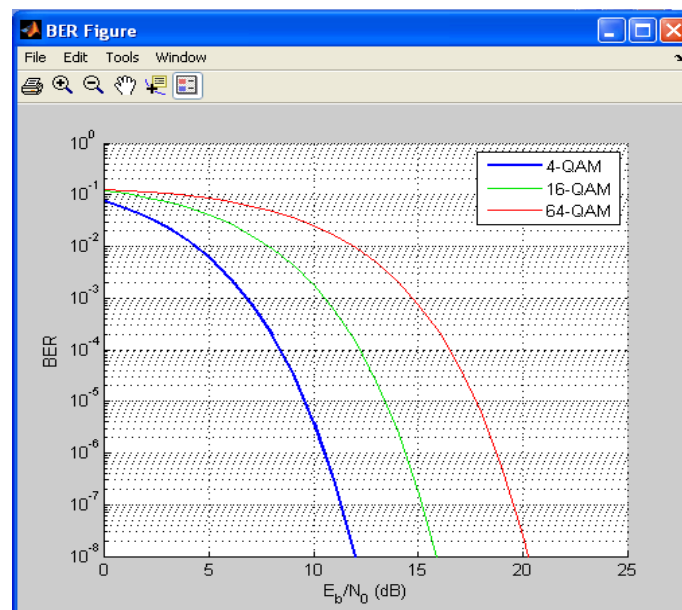


Figura 4: 4-QAM, 16-QAM e 64-QAM.

## Curva Teórica do 256-QAM

Para obter a curva teórica do 256-QAM para um canal AWGN em um faixa de 30 dB, modifique os parâmetros dados a seguir com os seguintes valores:  $E_b/N_0 = 0:30$ , *Channel type*: AWGN, *Modulation type*: QAM, *Modulation order*: 256. Alterando-se os parâmetros o resultado é mostrado na Figura 5, dada a seguir.

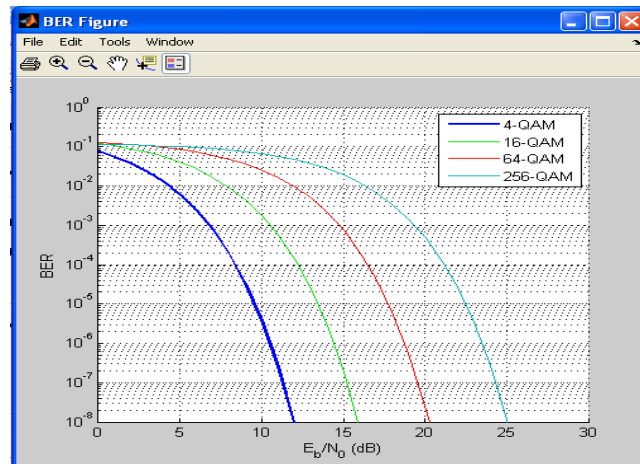


Figura 5: 4-QAM, 16-QAM, 64-QAM e 256-QAM.

## Questões

As questões a seguir devem ser respondidas, analisando-se os gráficos gerados no bertool.

1) Obtenha a curva teórica para o 4-QAM em um canal AWGN e preencha a tabela dada a seguir. Dica: Use o Data Cursor.

Eb/N0	1	2	3	4	5
BER					
Eb/N0	6	7	8	9	10
BER					

2) A partir da curva teórica do 4-QAM obtida anteriormente, preencha a tabela dada a seguir. Dica: Use o Data Cursor

BER	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$
Eb/N0					
BER	$10^{-6}$	$10^{-7}$	$10^{-8}$	$10^{-9}$	$10^{-10}$
Eb/N0					

3) Obtenha a curva teórica para o 16-QAM em um canal AWGN e preencha a tabela dada a seguir. Dica: Use o Data Cursor.

Eb/N0	1	3	5	7	9
BER					
Eb/N0	11	12	13	14	15
BER					

4) A partir da curva teórica do 16-QAM obtida anteriormente, preencha a tabela dada a seguir. Dica: Use o Data Cursor

BER	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$
Eb/N0					
BER	$10^{-6}$	$10^{-7}$	$10^{-8}$	$10^{-9}$	$10^{-10}$
Eb/N0					

5) A partir dos dados das tabelas anteriores e dos gráficos gerados no bertool, podemos dizer que o 4-QAM é mais eficiente do que o 16-QAM?. Justifique sua resposta em caso afirmativo ou negativo.



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
UNIDADE ACADÊMICA DE ENGENHARIA ELÉTRICA**

---

**QAM  
USANDO O SIMULINK**

**JOSENILDO FERREIRA GALDINO**



Este experimento objetiva analisar o sistema de modulação QAM por meio do simulink. Serão feitas simulações com e sem o canal AWGN.

## Esquema de Modulação QAM

Um dos parâmetros importantes para análise de sistemas de comunicações digitais, é a taxa de erro. Iremos criar um modelo de um sistema básico QAM no ambiente simulink, conforme o diagrama mostrado na figura 1.

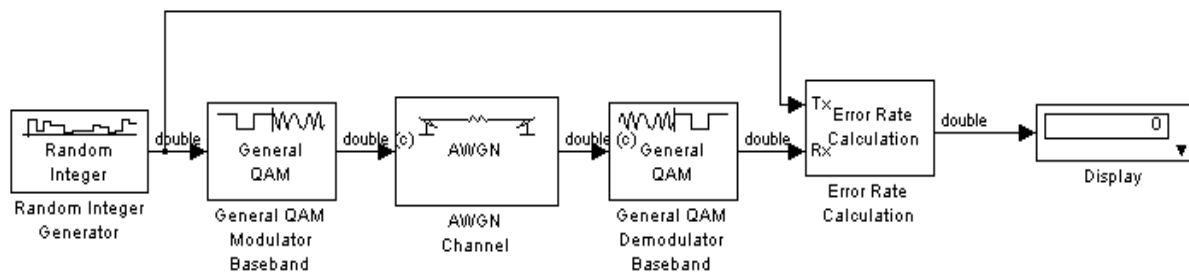


Figura 1: 4-QAM – Taxa de erro

Para a simulação do sistema BPSK, é necessário o uso de alguns blocos, para a busca rápida destes, é listada na Tabela 1, os blocos a serem utilizados, bem como, a quantidade e o local de busca no simulink.

Tabela 1: Blocos do Simulink

<i>Bloco</i>	<b>Caminho</b>
Random Integer Generator	Simulink > Communications Blockset > Comm Sources > Random Data Sources
Generator QAM Modulator Baseband	Simulink > Communications Blockset > Modulation > Digital Baseband Modulation >> AM
AWGN Channel	Simulink > Communications Blockset > Channels
Generator QAM Demodulator Baseband	Simulink > Communications Blockset > Modulation > Digital Baseband Modulation >> AM
Error Rate Calculation	Simulink > Communications Blockset >> Comm Sinks
Display	Simulink > Sinks

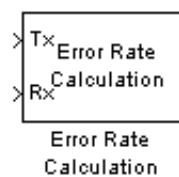
## Alterando os parâmetros

### Random Integer Generator

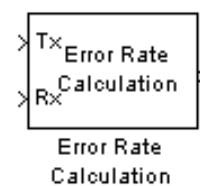
Para gerar símbolos aleatórios para simular um sistema 4-QAM, altere o parâmetro M-ary number para 4 e clique em OK.

### Error Rate Calculation

Error Rate Calculation exibe os resultados na área de trabalho do MATLAB, entretanto, no ambiente simulink é mais viável exibir os resultados utilizando o bloco display. Neste caso, precisamos acrescentar a porta de saída para o bloco Error, para isto dê um duplo clique com o botão esquerdo do mouse sobre o bloco e modifique o parâmetro *Output data* para *port*, feito isto o bloco modificado será semelhante ao da figura 2 dada a seguir




Este é o bloco padrão do *Error Rate Calculation*, contém 2 portas de entrada.



Este é o bloco *Error Rate Calculation* modificado, contém 2 portas de entrada e uma de saída.

Figura 2: Error Rate Calculation

A simulação do sistema 4-QAM pode ser executada, para isto, clique no botão *Start Simulation*, .

## Simulação do 4-QAM

Para melhor compreensão da influencia do ruído, em um sistema de comunicação digital do tipo 4-QAM iremos inicialmente simular este esquema de modulação sem o canal AWGN e observar o sinal modulado. Na etapa seguinte será adicionado o canal AWGN.

O diagrama de blocos de um sistema 4-QAM é mostrado na figura 3 dada a seguir. Neste modelo não consideraremos o canal AWGN para podermos visualizarmos como seria o sinal modulado sem a influência do ruído.

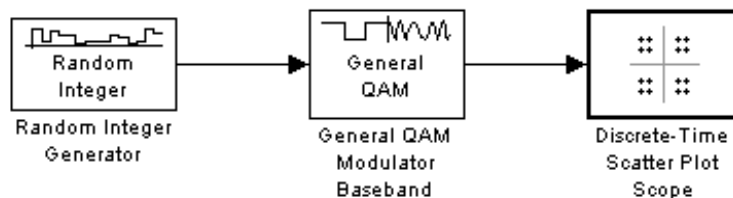


Figura 3: QAM sem canal AWGN


Para esta simulação é necessário o uso de alguns blocos, para a busca rápida dos blocos é listada na Tabela 2, os blocos a serem utilizados, e o local de busca no simulink.

Tabela 2: Blocos do Simulink

<i>Bloco</i>	<b>Caminho</b>
Random Integer Generator	Communication blockset > Comm Sources > Random Data Sources
General QAM Modulator Baseband	Communication blockset > Modulation > Digital Baseband Modulation > AM
Discrete-Time Scatter Plot Scope	Communication blockset > Comm Sinks
AWGN Channel	Communication blockset > Channels

Para montar o esquema da figura 3 é necessário abrir o simulink. Para isto digite na janela de comandos, o comando mostrado a seguir,

```
>> simulink
```

ou simplesmente, clique no ícone , para ativar o simulink. Em seguida abra um novo documento para poder inserir os blocos neste documento.

Após montar o esquema da figura 3 o próximo passo é alterar alguns dos parâmetros do bloco, conforme a Tabela 3 dada a seguir.

Tabela 3: Parâmetro dos blocos

<i>Bloco</i>	M-ary-number
Random Integer Generator	4

Execute o modelo da figura 3 e tenha como resultado os pontos da constelação conforme figura 4.

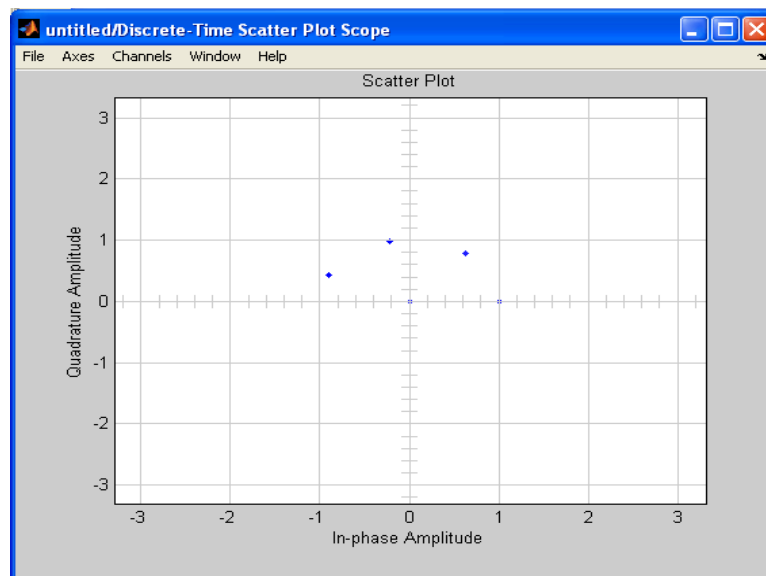


Figura 4: Constelação de Sinais

O diagrama de blocos de um sistema 4-QAM sem o canal AWGN é mostrado na figura 3. Acrescente o bloco AWGN conforme modelo descrito na figura 5 e execute a simulação.

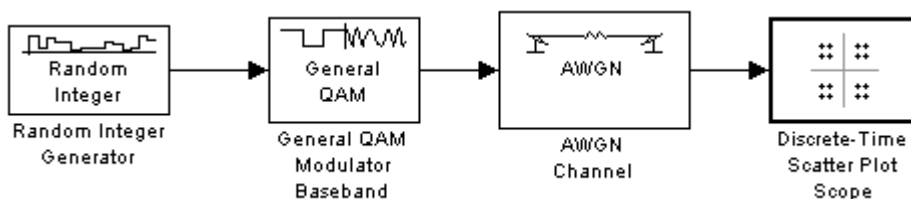



Figura 5: 4-QAM com AWGN

## Simulação

A simulação do 4-QAM no simulink agora pode ser executada. Para isto clique no botão *Start Simulation*,  e escolha um tempo de simulação de 100 segundos. O resultado é mostrado nas figuras 6 e 7.

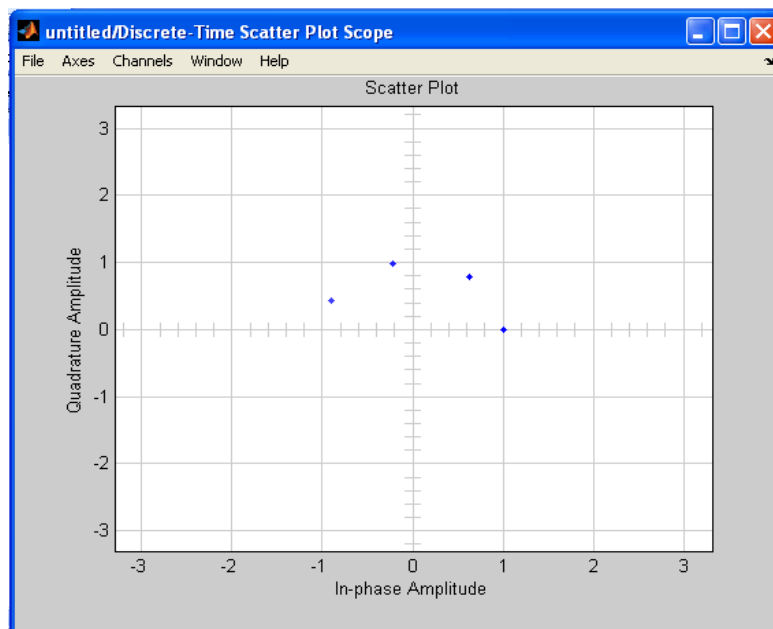


Figura 6: 4-QAM

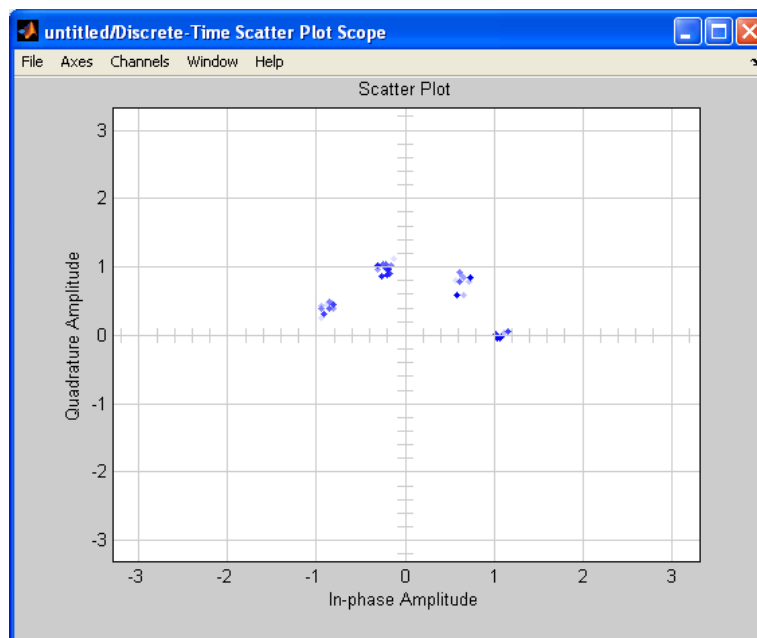


Figura 7: 4-QAM com canal AWGN.

## Questões

As questões a seguir devem ser respondidas, utilizando-se o simulink.

- 1) Simule o esquema da figura 1, e altere o parâmetro *M-ary-number* para 2. Este parâmetro pertence ao bloco Random Integer Generator. Considere um tempo de simulação 100 segundos. Modifique o parâmetro *Symbol period* para 100 segundos. Nesta simulação o canal tem uma relação sinal-ruído de 10 dB. Anote o resultado mostrado no display.
- 2) Utilizando o esquema simulado anteriormente na questão 1, modifique o parâmetro  $E_b/N_0$  para 20 dB do bloco AWGN Channel, e execute o novo esquema. Anote o resultado mostrado no display e compare com o resultado obtido para o caso de 10 dB.
- 3) Simule o esquema da figura 8 dada abaixo. Em cima de cada um dos blocos desta figura, estão indicados as modificações a serem feitas nos parâmetros. Considere um tempo de simulação de 100 segundos. O que acontece com os pontos da constelação de sinais, quando variamos a relação sinal ruído. Explique.

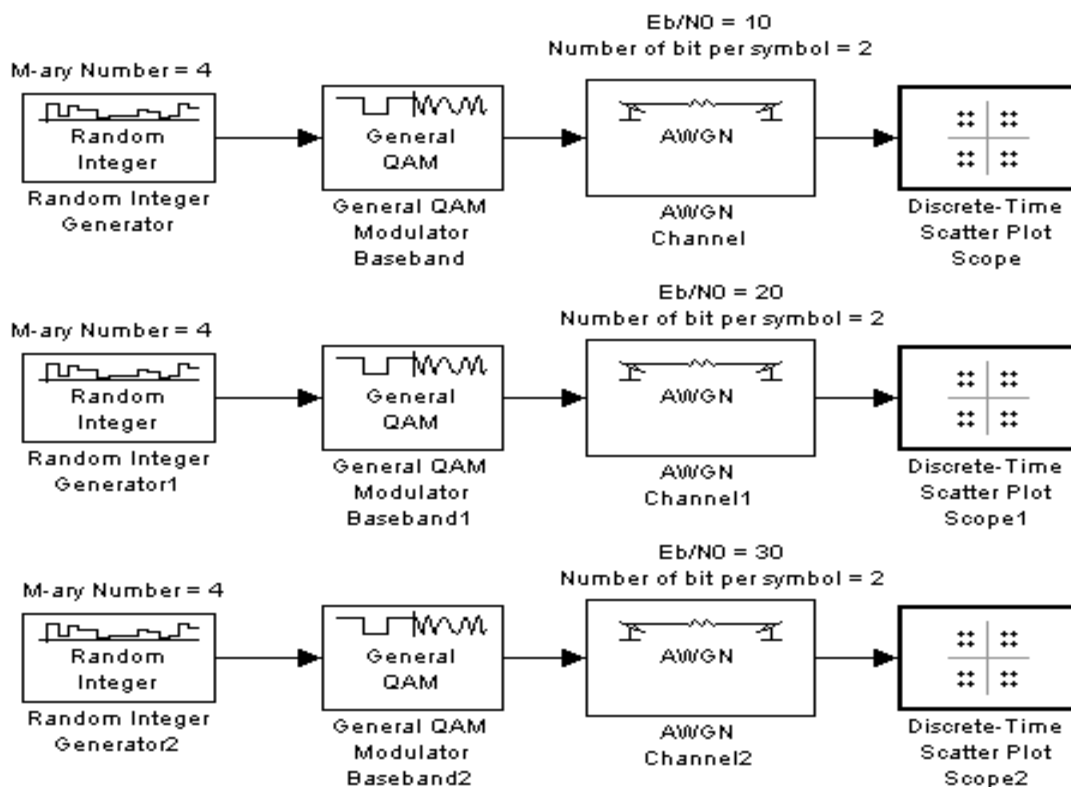


Figura 8: 4-QAM

4) Simule o esquema da figura 9 dada abaixo. Em cima de cada um dos blocos desta figura, estão indicados as modificações a serem feitas nos parâmetros. Considere um tempo de simulação de 100 segundos.

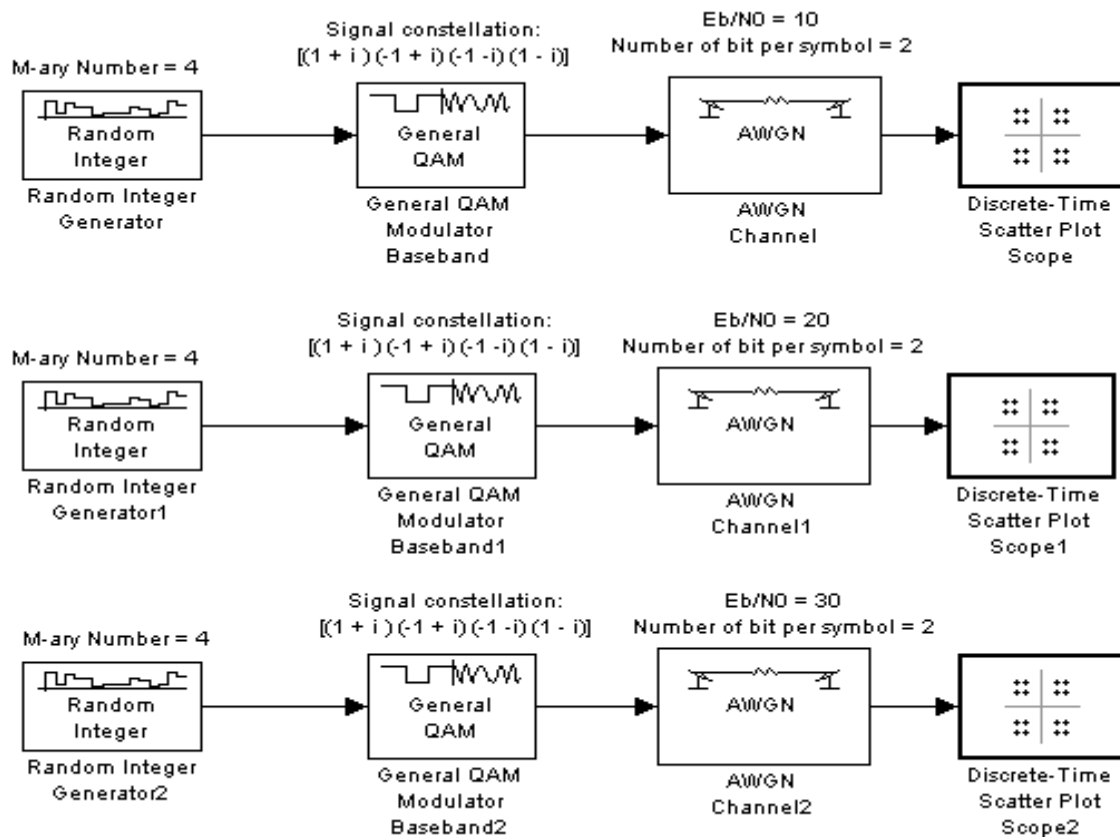


Figura 9: 4-QAM

5) Simule o esquema da figura 10. Em cima de cada um dos blocos desta figura, estão indicados as modificações a serem feitas nos parâmetros. Considere um tempo de simulação de 100 segundos.

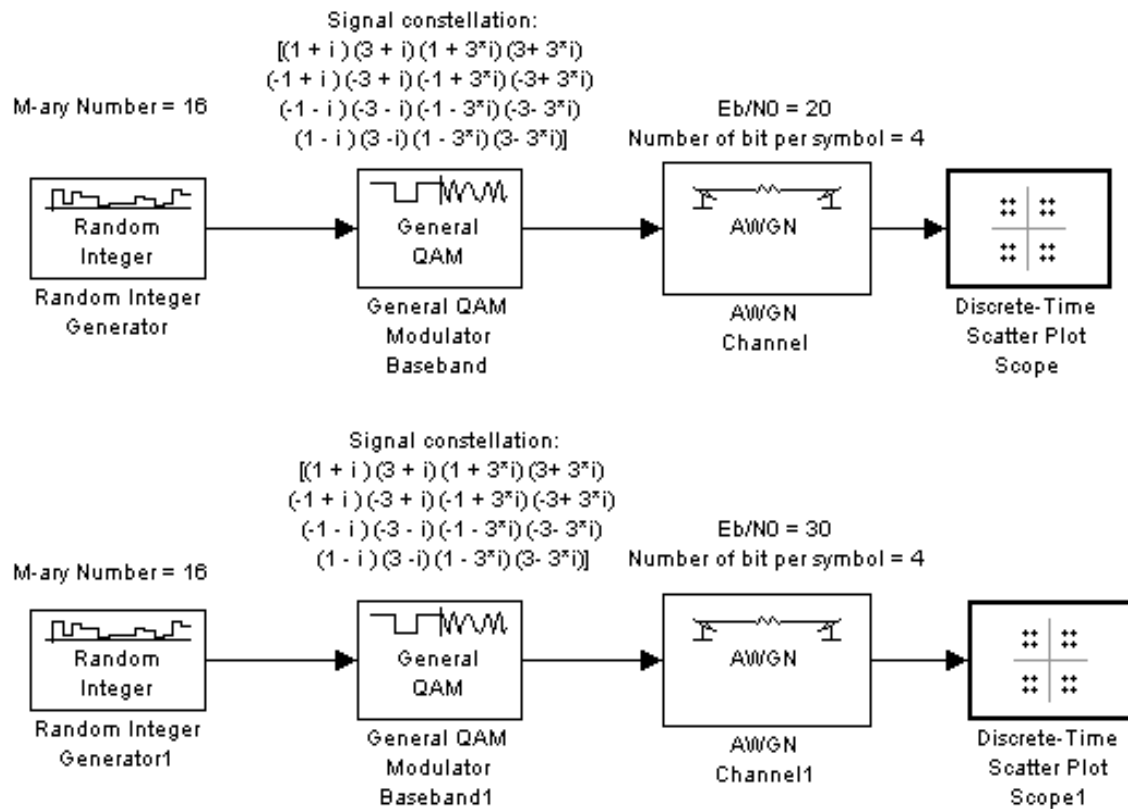


Figura 10: 16-QAM

6) Simule o sistema 16-QAM conforme mostrado na figura 11. Considere um tempo de simulação de 100 segundos.

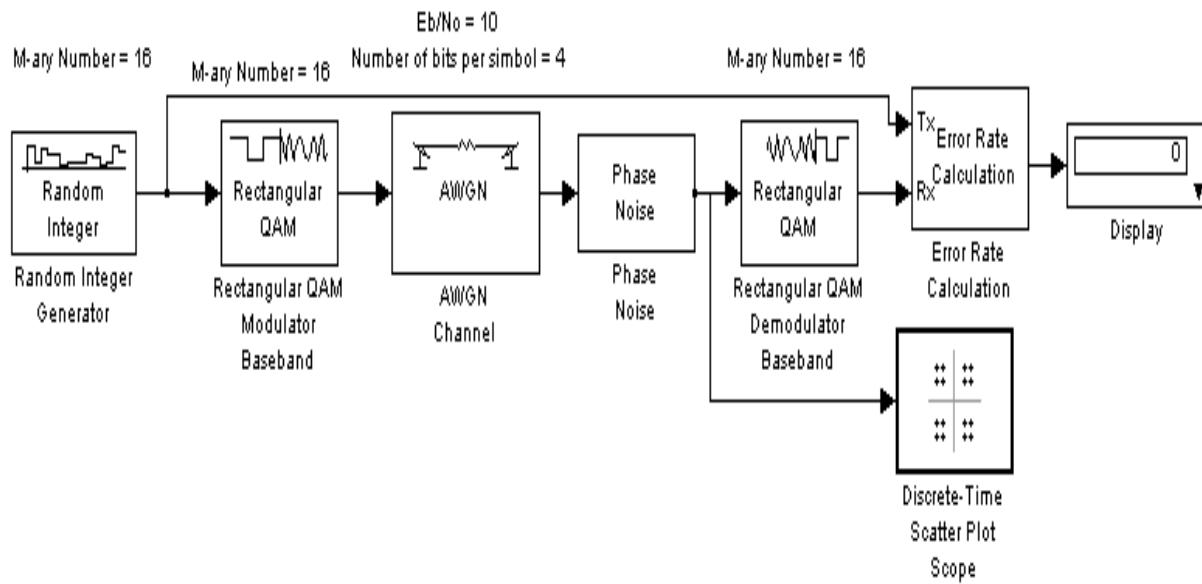


Figura 11: 16-QAM

OBSERVAÇÃO: O bloco Phase Noise encontra-se em Communications Blockset >> Modulation >> RF Impairments





**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
UNIDADE ACADÊMICA DE ENGENHARIA ELÉTRICA**

---

## **PSK E QAM**

**JOSENILDO FERREIRA GALDINO**

Este experimento objetiva implementar os sistemas 4-PSK, 8-PSK, 4-QAM e 16-QAM por meio de arquivos de comandos criados pelo usuário.

## **1. Introdução**

Este guia tem por finalidade familiarizar o estudante com o programa MATLAB e sua utilização em sistemas de comunicações. O MATLAB (**MA**TRIX **LAB**ORATORY – Laboratório de Matrizes) é um programa de computador especializado e otimizado para cálculos científicos e de engenharia. Surgiu em 1970 com o propósito de auxiliar os cursos de teoria matricial, álgebra linear e análise numérica. Entretanto, o MATLAB é considerado uma ferramenta computacional poderosa capaz de resolver diversos problemas técnicos.

Este experimento está dividido em duas partes: na primeira parte será abordado alguns comando básicos para utilização do MATLAB, bem como uso de algumas funções elementares. No final encontra-se um série de exercícios que deverá ser entregue como pré-requisito para que o aluno possa realizar o experimento no laboratório. Na segunda parte, o aluno irá realizar a simulação do experimento com a linguagem de programação MATLAB.

## **2. Introdução ao MATLAB**

Esta seção objetiva dar uma orientação ao aluno, ao iniciar neste novo ambiente, para isto será apresentado alguns comandos básicos.

### **2.1 – Ambiente MATLAB**

Para utilizar o programa MATLAB é preciso dar um duplo clique com o mouse, e aparecerá a janela padrão do MATLAB conforme mostrada na Figura 1. Caso, o ícone do MATLAB não se encontre na área de trabalho, faça o seguinte procedimento. Clique no menu Iniciar, em seguida, selecione o programa MATLAB e clique com o mouse.

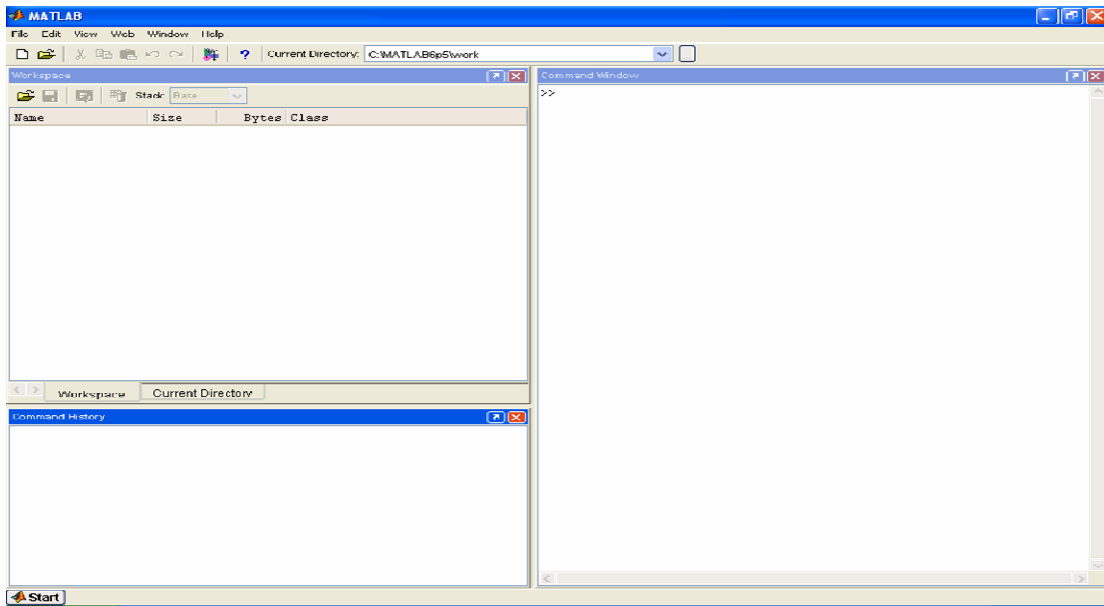


Figura 1: Área de Trabalho do MATLAB

Existem casos em que a janela exibida não é a padrão, neste caso, o procedimento para restaurar a janela padrão do MATLAB é o seguinte. Clique no Menu *Desktop*, em seguida selecione *Desktop Layout* e clique em *Default*, conforme mostra a Figura 2.

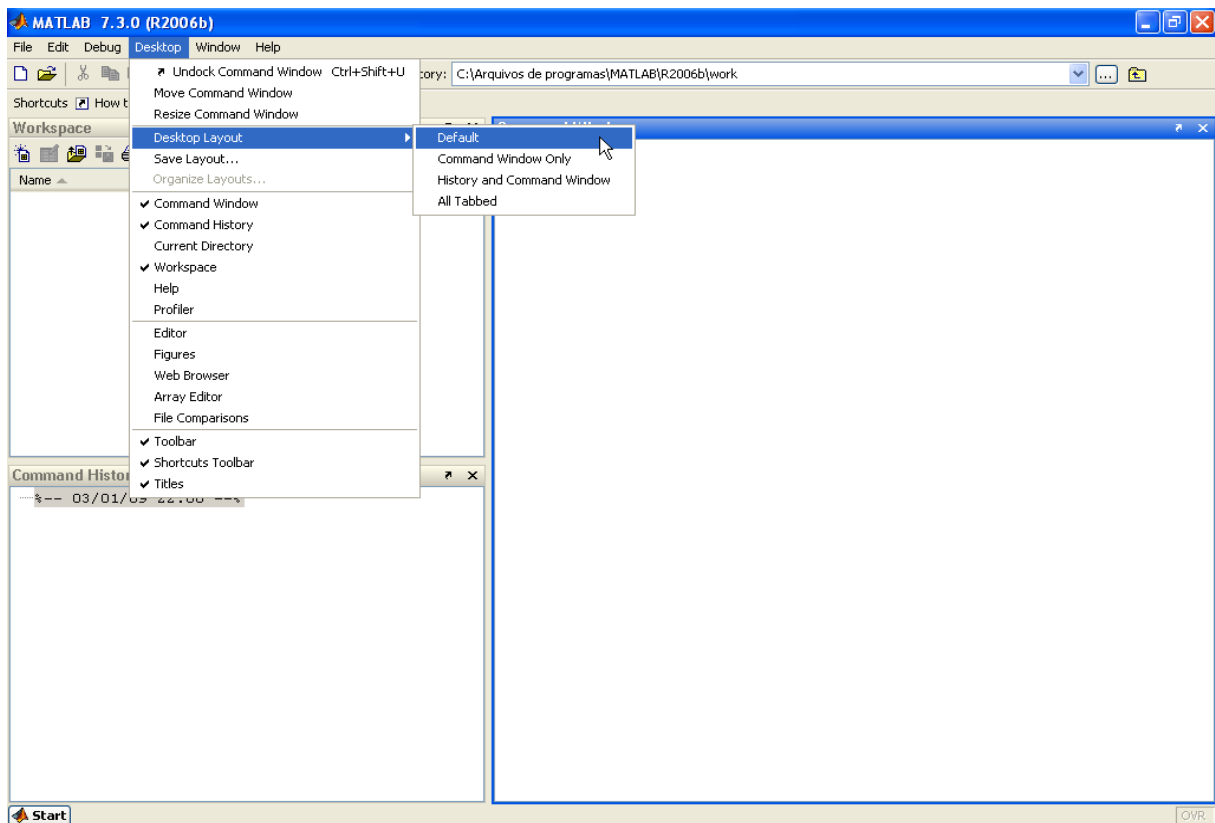


Figura 2: Restauração da Área de Trabalho padrão do MATLAB

A área de trabalho mostrada na figura 1 é obtida quando iniciamos o MATLAB versão 6.5. Esta área integra diversas ferramentas que podem ser acessadas. A figura 3 exhibe as diversas possibilidades de uso desta janela. Na área de trabalho do MATLAB, temos:

- Janela de Comandos
- Janela de Histórico de Comandos
- Espaço de Trabalho
- Navegador de Ajuda
- Navegador do Diretório Corrente

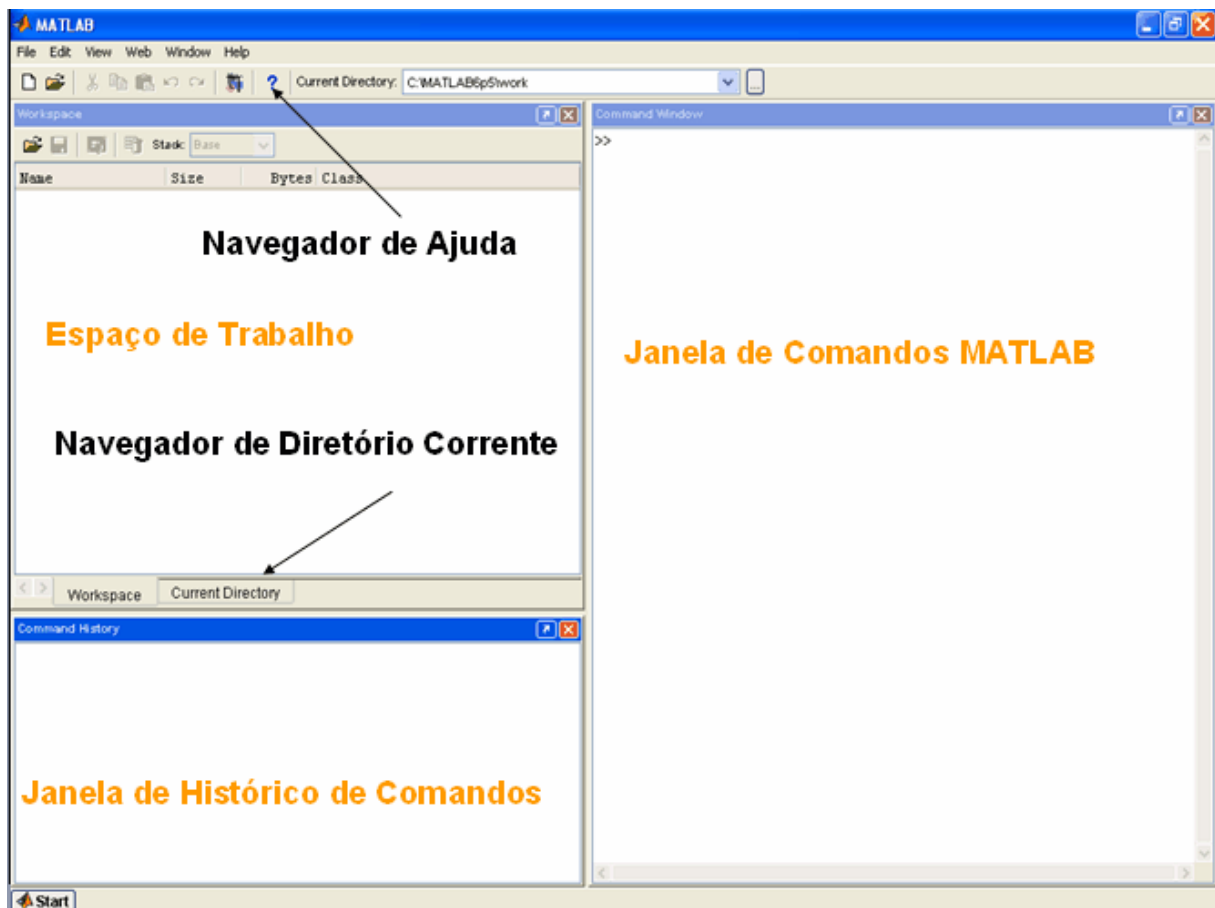


Figura 3: Área de trabalho do MATLAB, com as suas sub-divisões

### 2.1.1 Janela de Comandos

A janela de comandos é utilizada para executar comandos básicos tais como cálculos numéricos, chamada a funções, ou até mesmo executar um arquivo de comando, conhecido como M-file, arquivo.m. Por exemplo. Para efetuar o cálculo  $5 + 3 + 2$ , no MATLAB. Basta digitar `>> 5 + 3 + 2` e em seguida pressionar a tecla Enter.

```
>> 5 + 3 + 2
```

```
ans =
```

```
10
```

O MATLAB calcula a resposta após o usuário pressionar a tecla Enter, neste momento, será criada uma variável padrão denominada *ans* que é utilizada sempre que um usuário não fornecer uma variável para armazenamento do resultado. Por exemplo, suponha que queremos efetuar o cálculo  $5 + 3 + 2$  e guardar o resultado na variável *soma*.

```
>> soma = 5 + 3 + 2
```

```
soma =
```

```
10
```

Outro recurso poderoso do MATLAB é a disponibilidade de uma enorme variedade de funções, ou seja, ele possui uma grande biblioteca de funções predefinidas que podem ser utilizadas pelo usuário. Por exemplo, suponha que queiramos calcular o cosseno do ângulo de  $\frac{\pi}{3} rad$ . Para resolver este problema podemos chamar a função pré-definida *cos* e atribuir o ângulo específico em radianos.

```
>> cos(pi/3)

ans =

    0.5000
```

## 2.1.2 Janela de Histórico de Comandos

A Janela de Histórico de Comando mostra uma lista dos comandos que o usuário digitou na Janela de Comandos. Por exemplo, ao executar os cálculos anteriores a janela de histórico de Comandos foi atualizada com os novos comandos. A figura 4 mostra a janela de histórico obtida.

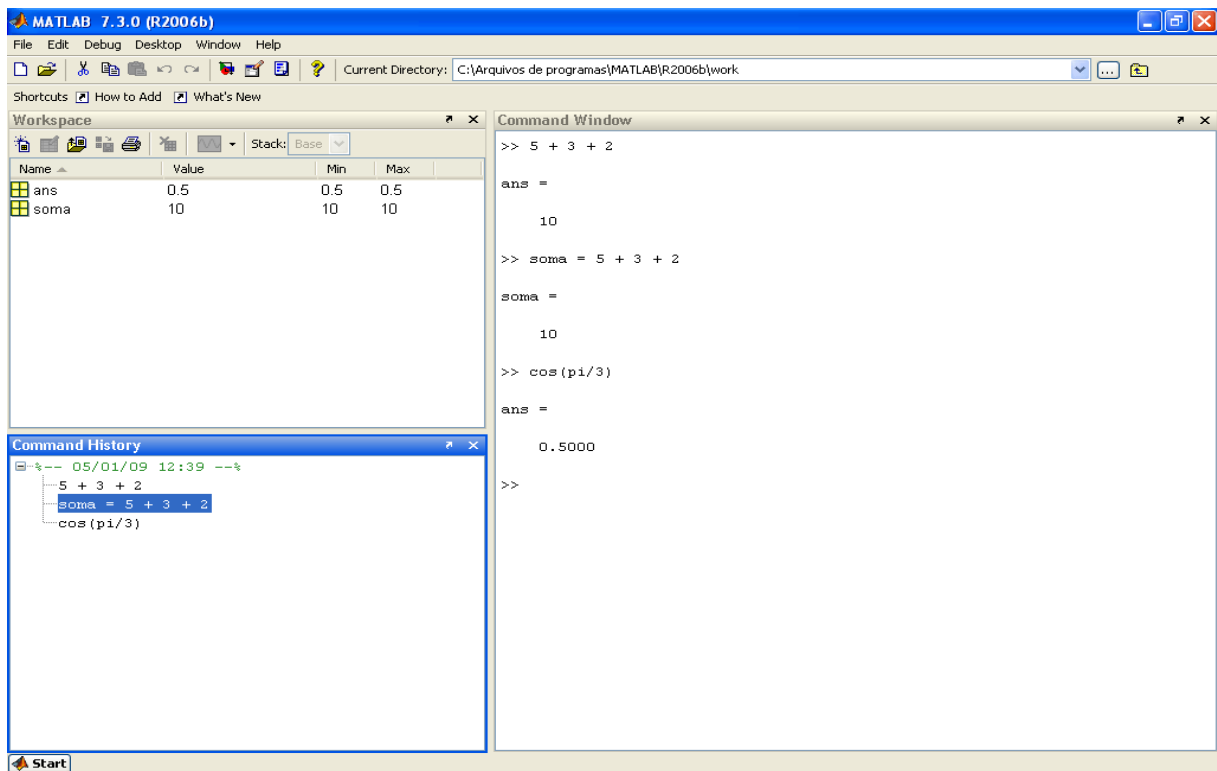


Figura 4: Janela de Histórico de Comandos

A principal vantagem de utilizar a janela de histórico de comando é a possibilidade de reuso dos comandos utilizados anteriormente. Por exemplo, se o usuário quiser realizar algum cálculo que foi realizado anteriormente, basta selecionar o comando na janela de histórico e dar um duplo clique no comando com o botão

esquerdo do mouse. Este é um recurso muito utilizado por usuários experientes no MATLAB. Além de que, os comandos na janela de histórico permanecem até mesmo quando o usuário fechar o programa e em outro momento retornar a utilizar o MATLAB, a janela de comando permanece com os comandos anteriores. Por exemplo, suponha que queiramos efetuar o cálculo  $soma = 5 + 3 + 2$ , efetuado anteriormente a partir da janela de histórico.

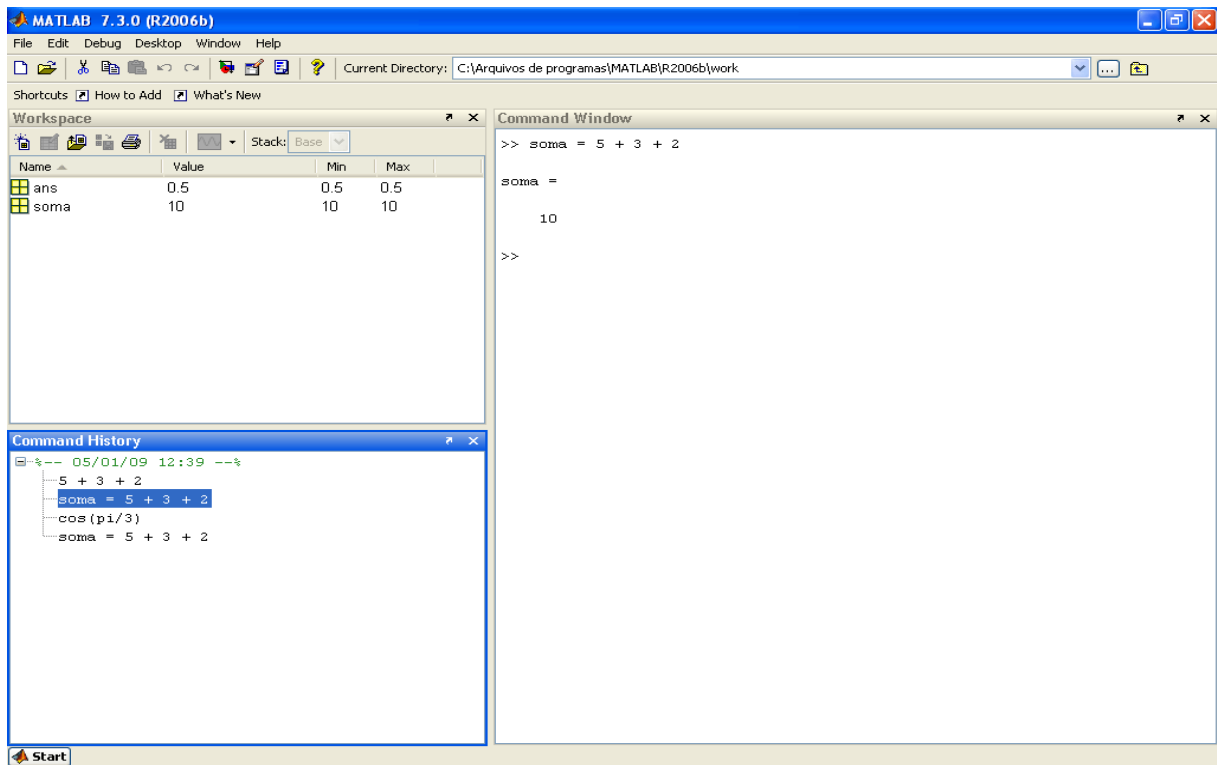


Figura 5: Utilizando a Janela de Histórico

### 2.1.3 Espaço de Trabalho

O espaço de trabalho (Workspace) exibe as variáveis criadas com os valores armazenados, é uma ferramenta importante para depuração em programas mais complexos e sua importância ficará mais clara ao trabalhar com arquivos de comandos. Ao executar os comandos anteriores as variáveis *ans* e *soma* foram criadas.

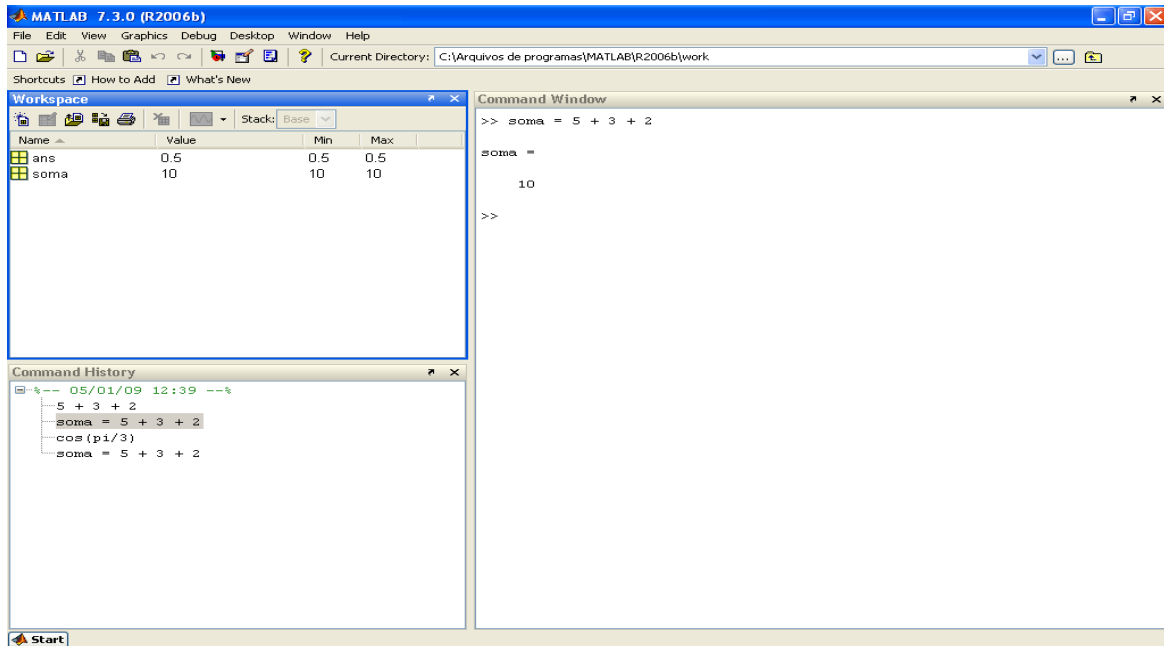



Figura 6: Espaço de Trabalho (Workspace)

## 2.1.4 Navegador de Ajuda

Para obter ajuda no MATLAB pode-se utilizar o Navegador de Ajuda da seguinte maneira, clicando com o botão esquerdo do mouse no símbolo  localizado na barra de ferramentas do MATLAB, ou simplesmente, digitando o comando *helpdesk* ou *helpwin* na janela de comandos. Outras maneiras de consultar a documentação do MATLAB é por meio da função *help*. Digita-se a comando *help* e tecla enter, ou para um consulta mais específica sobre alguma função digita-se *help* mais o nome da função. Por exemplo, suponha que queiramos saber mais detalhes da função *cos* do MATLAB usando o comando *help*. A figura 7 mostra um exemplo de como utilizar a função *help* para dar informações de uso da função e facilitando o seu uso. O comando *help* é um dos comandos básicos mais utilizados por propiciar um maior entendimento das funções predefinidas do MATLAB, além de dar dicas de funções correlatas. No caso do comando *help cos*, observe na figura 7 que o MATLAB oferece dicas de outras funções relacionadas com a função consultada e que propicia um conhecimento de um número cada vez maior de funções do MATLAB. É possível consultar as funções *acos* e *cosd*, simplesmente clicando com o botão esquerdo do mouse.



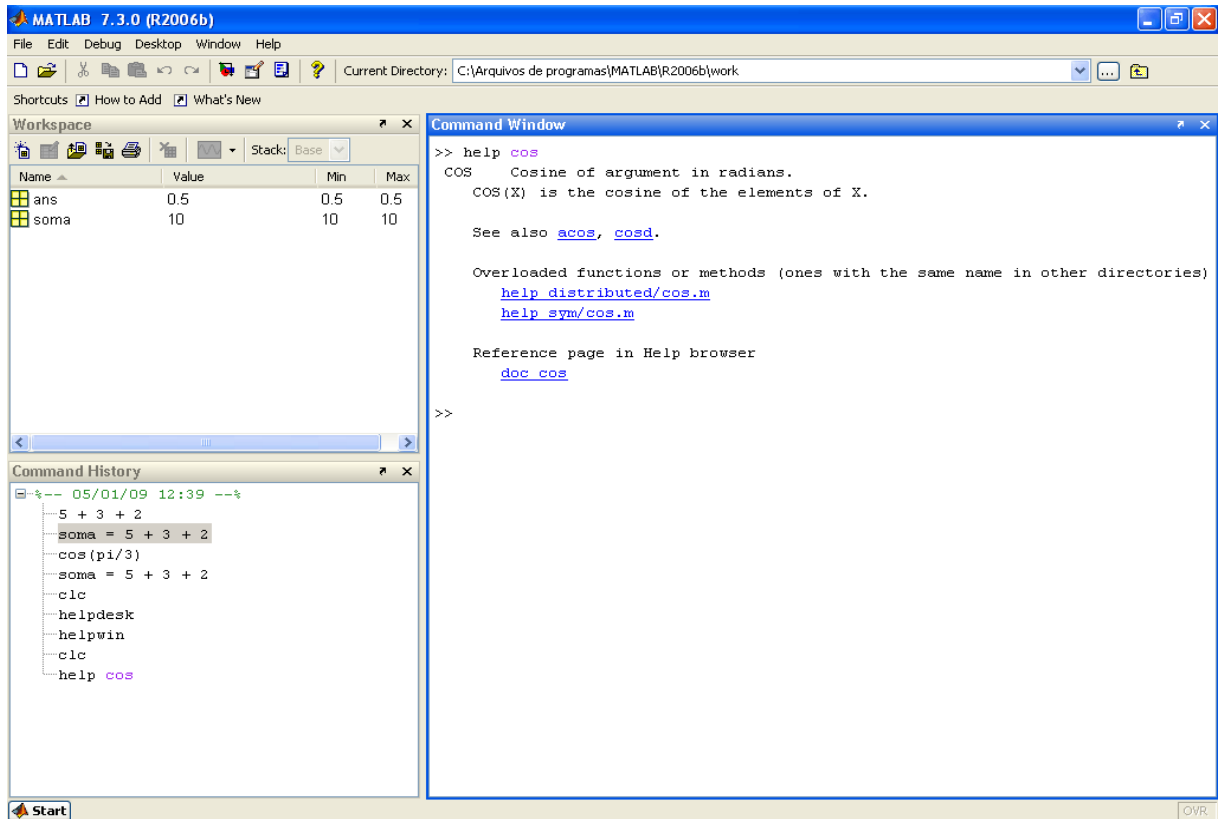


Figura 7: Uso da função help

Entretanto, a função help possui uma grande limitação, o usuário tem que saber o nome completo da função e muitas vezes isso não é possível, não tem como adivinhar. Para estes casos deve-se utilizar a função lookfor, que devolve para o usuário todas as informações da função pesquisada e de outras mais que tenha alguma relação com o nome da função.

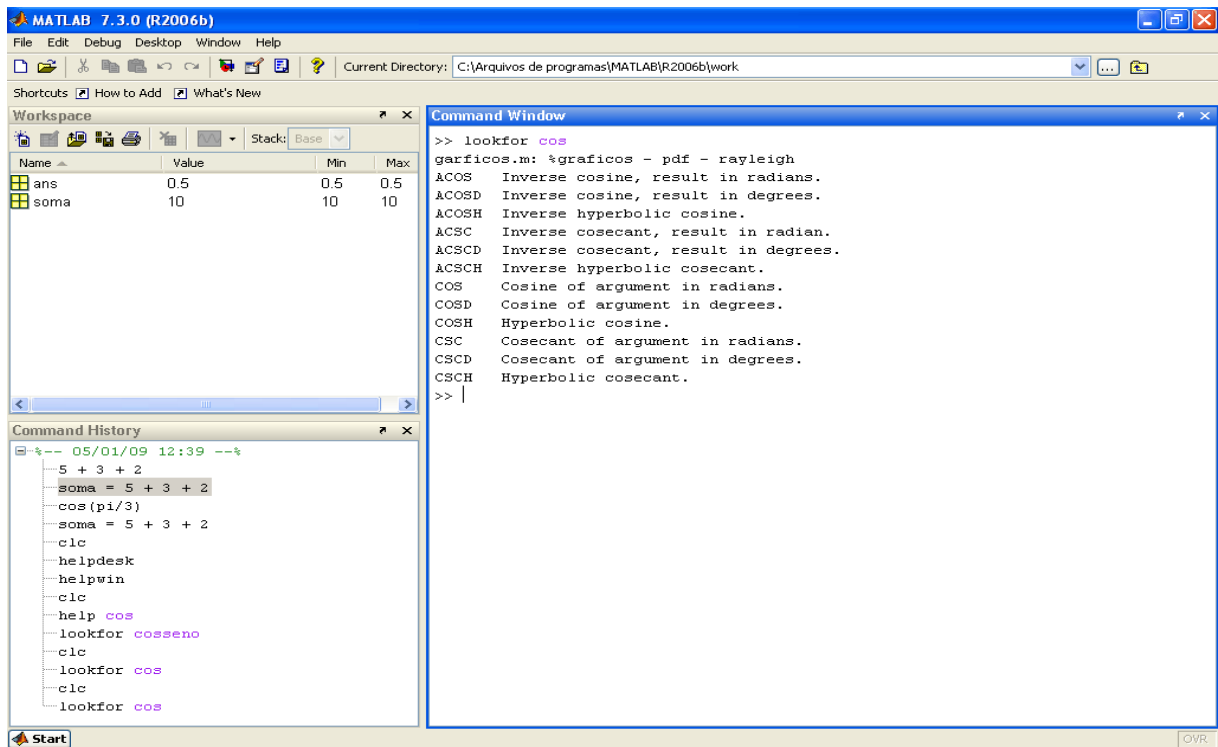


Figura 8: Uso da função lookfor

## 2.1.5 Navegador do Diretório Corrente

O Navegador de Diretório Corrente permite que o usuário tenha acesso ao arquivo do seu diretório corrente atual. Para saber em que diretório o usuário está, observe na Figura 9, ao lado no Navegador de Ajuda, está listado o diretório atual (Current Directory). Para ver os arquivos que estão neste diretório, você pode acessar diretamente do Navegador de Diretório Corrente, simplesmente clicando com o botão esquerdo do mouse, ou se preferir usar a janela de comando digite *dir* e veja a lista completa dos arquivos que estão no seu diretório atual.

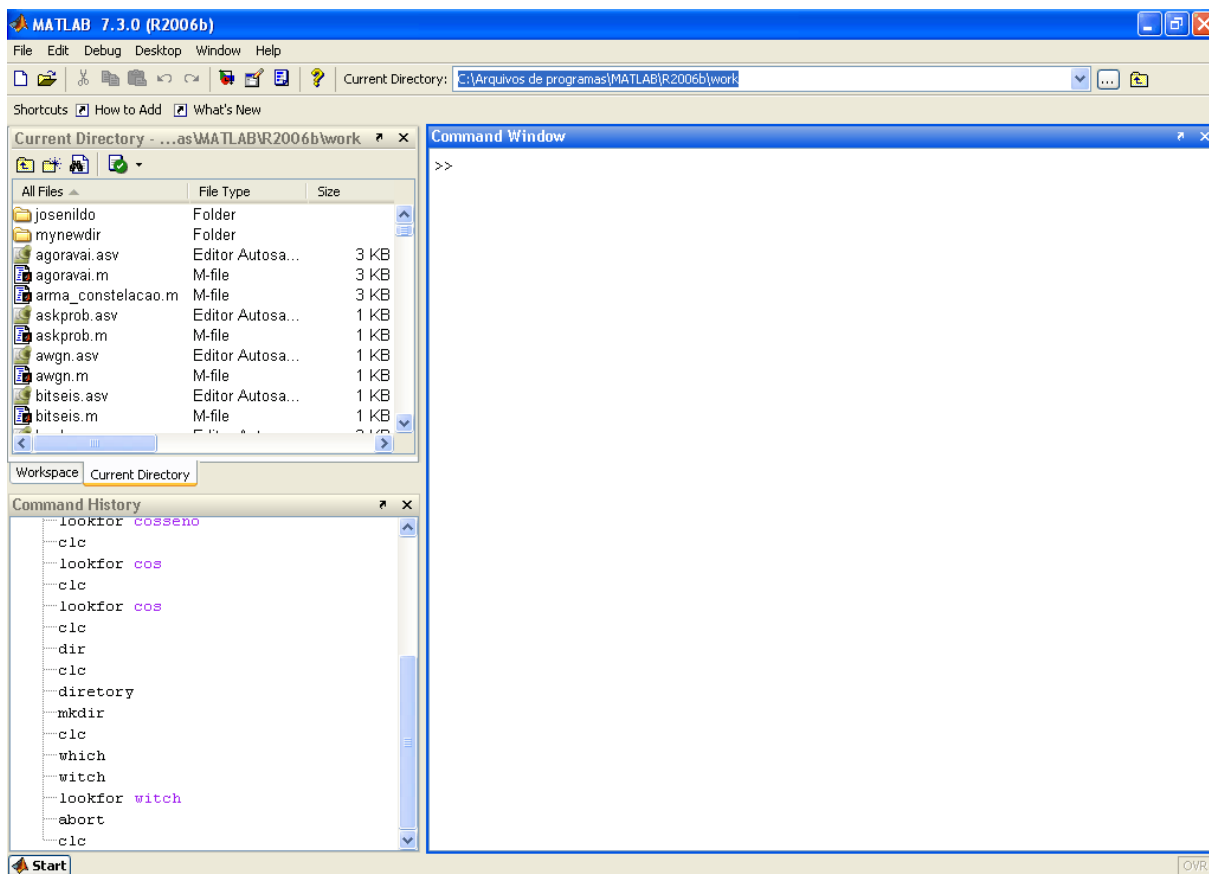


Figura 9: Navegador de Diretório Corrente

## 2.2 Variáveis Especiais

O MATLAB contém constantes pré-definidas, ou seja, por padrão elas já se encontram no MATLAB prontas para uso em programas e cálculos. A Tabela 1 lista algumas variáveis especiais do MATLAB.

Tabela 1: Variáveis Especiais do MATLAB

Variável	Função
ans	Variável especial utilizada para armazenar o resultado de uma expressão, caso esse resultado não seja explicitamente atribuído a outra variável
nan	Indica que o resultado não é um número. Geralmente, resulta de uma operação matemática indefinida.
eps	Representa a menor diferença entre dois números que pode ser representada no computador.
inf	Representa infinito, ou seja, como resultado de uma divisão por 0.

i ou j	Representa a unidade imaginaria $i (\sqrt{-1})$
pi	Representa o numero $\pi$ (3.1415927)
date	Indica a data corrente em formato de cadeia de caracteres no formato dia – mês – ano
clock	Exibe a hora atual em um vetor linha de seis elementos contendo ano, mês, dia, hora, minutos e segundos

Este é um dos conceitos preliminares que deve ser aprendido ao utilizar o ambiente MATLAB. Toda variável deve ser única e possuir um *nome* que a distingue das demais e nunca se deve dar o nome de uma variável com um nome de uma variável especial.

## 2.2.1 Iniciando Variáveis no MATLAB

### Operador dois pontos

O operador dois pontos especifica um vetor a partir de seu primeiro elemento somando-se um incremento até chegar o ultimo elemento especificado. O formato geral de um operador dois pontos é:

primeiro:incremento:ultimo

**Exemplo:** Uso do operador dois pontos:

```
x =
     1     2     3     4     5     6     7     8
>> x = 1:8
x =
     1     2     3     4     5     6     7     8
```

No exemplo anterior, observe que quando o incremento é 1, ele pode ser omitido, ou seja, os comandos  $x = 1:1:8$  e  $x = 1:8$  são equivalentes. Este é o único caso em que o incremento pode ser omitido.

## Comando disp

Este comando é utilizado para exibir o conteúdo de uma matriz ou então, para exibir uma mensagem de texto na tela. A seguir é mostrado um exemplo de uma mensagem e de uma exibição de matriz.

```
>> disp('Engenharia Eletrica - UFCG')
Engenharia Eletrica - UFCG

>> disp(x)
      1      2      3      4      5      6      7      8
```

## Comando fprintf

Este comando é utilizado para imprimir mensagens de texto ou conteúdos de matrizes. O formato geral deste comando é.

fprintf(formato, matriz)

A tabela 2 mostra os principais caracteres de formatação para o comando fprintf.

Tabela 2: Caracteres de Formatação do comando fprintf

Caracteres de Formatação	Descrição
%d	Exibe o valor como inteiro
%e	Exibe o valor em formato exponencial
%f	Exibe o valor em formato de ponto flutuante
%g	Exibe o valor em formato de ponto flutuante ou exponencial – o que for mais curto
\n	Muda de linha

No exemplo a seguir é mostrado o uso do comando `fprintf` com a opção `%d` e `\n`.

```
>> snr = 10;
>> fprintf('A relação sinal ruído eh %d dB\n', snr)
A relação sinal ruído eh 10 dB
```

Uma diferença quanto a exibição de conteúdo de matrizes usando `disp` e `fprintf` é que o comando `disp` exibe o conteúdo de um número complexo da forma  $z = a + bi$ , enquanto o comando `fprintf` exibe apenas a parte real do complexo  $z$ . Portanto, se estiver trabalhando com números complexos é recomendável exibir o conteúdo da matriz com o comando `disp`.

O exemplo a seguir iremos comparar o uso de `disp` e `fprintf` para mostrar o conteúdo de  $z = 1 + i$ .

```
>> z = 1 + i;
>> disp(z)
1.0000 + 1.0000i

>> fprintf('%d\n', z)
1
```

No exemplo anterior observasse que o comando `fprintf` não exibiu a parte imaginária do complexo  $z$ , portanto, deve-se ter cuidado ao utilizar o comando `fprintf`.

### 3. Comandos Básicos

O estudante ao iniciar o uso no MATLAB pode ter algumas dificuldades, com este intuito será apresentado alguns comandos básicos.

#### 3.1 Uso das funções `clc`, `clf` e `clear`

O MATLAB contém algumas funções úteis para uso. Algumas delas estão listadas na Tabela 2.

Tabela 2: Funções elementares

Nome da Função	Descrição
<code>clc</code>	Limpa a janela de comandos do MATLAB

clf	Limpa a janela gráfica do MATLAB
clear	Limpa o Espaço de trabalho

### 3.2 Uso das funções matemáticas elementares

As funções matemáticas elementares e algumas funções de arredondamento são mostradas na tabela 3.

Tabela 3: Funções Matemáticas Elementares

Função	Descrição
abs(x)	Calcula $ x $
angle(x)	Calcula o argumento do complexo x
real(x)	Calcula a parte real do complexo x
imag(x)	Calcula a parte imaginária do complexo x
conj(x)	Calcula o conjugado do complexo x
sin(x)	Calcula $\sin x$ , com x em radianos.
cos(x)	Calcula $\cos x$ , com x em radianos.
tan(x)	Calcula $\tan x$ , com x em radianos.
asin(x)	Calcula $\sin^{-1} x$ , com x em radianos.
acos(x)	Calcula $\cos^{-1} x$ , com x em radianos.
atan(x)	Calcula $\tan^{-1} x$ , com x em radianos.
log(x)	Calcula o logaritmo natural, $\log_e x$ .
log10(x)	Calcula o logaritmo na base 10, $\log_{10} x$ .
exp(x)	Calcula $e^x$ .
sqrt(x)	Calcula a raiz quadrada de x.
factorial(x)	Calcula o fatorial de x.
Inv(x)	Calcula a inversa da matriz x.
det(x)	Calcula o determinante da matriz x.
trace(x)	Calcula o traço da matriz x.
round(x)	Arredonda o valor de x para o inteiro mais próximo.
fix(x)	Arredonda o valor de x para o inteiro mais próximo de zero.
floor(x)	Arredonda o valor de x para o inteiro mais próximo de $-\infty$ .

ceil(x)	Arredonda o valor de x para o inteiro mais próximo de $+\infty$ .
sign(x)	Se $x < 0$ , a função retorna -1, se $x = 0$ , a função retorna 0; caso contrário, a função retorna 1.
rem(x,y)	Retorna o resto da divisão de x/y.

Diversas funções matemáticas elementares, dentre elas temos: funções trigonométricas, funções exponenciais, funções complexas e funções de arredondamento podem ser acessadas por meio do seguinte comando digitado na janela de comandos do MATLAB.

```
>> help elfun
```

### 3.3 – Formato de Exibição dos Dados de Saída

O MATLAB utiliza o formato padrão short, ou seja, exibe um número com quatro dígitos decimais depois do ponto decimal. O formato padrão pode ser modificado de acordo com a exibição que o usuário desejar de acordo com o comando da Tabela 4. Para mostrar as diversas possibilidades de exibição, usamos a variável especial pi e mostramos as várias possibilidades.

Tabela 4: Formato de Exibição de Saída

Comando	Descrição	Exemplo
format short	4 dígitos decimais (default)	3.1416
format long	14 dígitos decimais	3.141592653589793
format short e	5 dígitos mais expoente	3.1416e+000
format short g	5 dígitos no total, com ou sem expoente	3.1416
format long e	15 dígitos mais expoente	3.141592653589793e+000
format long g	15 dígitos no total, com ou sem expoente	3.14159265358979
format bank	formato monetário, 2 casas decimais	3.14
format hex	Exibição hexadecimal de bits	400921fb54442d18
format rat	Razão aproximada entre inteiros	355/113
format compact	Suprime linhas adicionais	
format loose	Restabelece as linhas adicionais	
format +	Exibe somente o sinal do número	



### 3.4 Comandos Múltiplos

O MATLAB permite escrever mais de um comando por linha, entretanto é recomendado que tenha apenas um comando por linha, isto facilita na fase de depuração de programas escritos neste ambiente. Por exemplo, podemos criar três variáveis x,y e z no MATLAB, digitando apenas uma única linha de comando, conforme mostrado a seguir.

```
>> x = 1; y = 2; z = 3;
```

### 3.5 Particionamento de uma linha de comando

Às vezes, a linha de comando que deve ser digitada pode ser extensa. Para resolver o problema pode-se particionar a linha de comando finalizando-se com ... e digitando-se o restante do comando na próxima linha. Para exemplificar temos um caso de uso do particionamento de linha de comando.

```
>> x + y + ...  
z  
  
ans =  
  
6
```

### 3.6 Comentários

Os programadores inserem comentários para documentar programas e melhorar a legibilidade para que outras pessoas possam entender seu programa. No MATLAB, os comentários começam com %, chamado de *comentário de linha única*, porque o comentário termina no fim da linha em que está. Os comentários não levam o computador a executar qualquer ação quando o programa for executado. O MATLAB utiliza a cor verde para identificar linhas de comentário, conforme mostrado no exemplo a seguir.

```
>> % Esta é uma linha de comentário.
```

## 4. Arquivo de Comando (M-File)

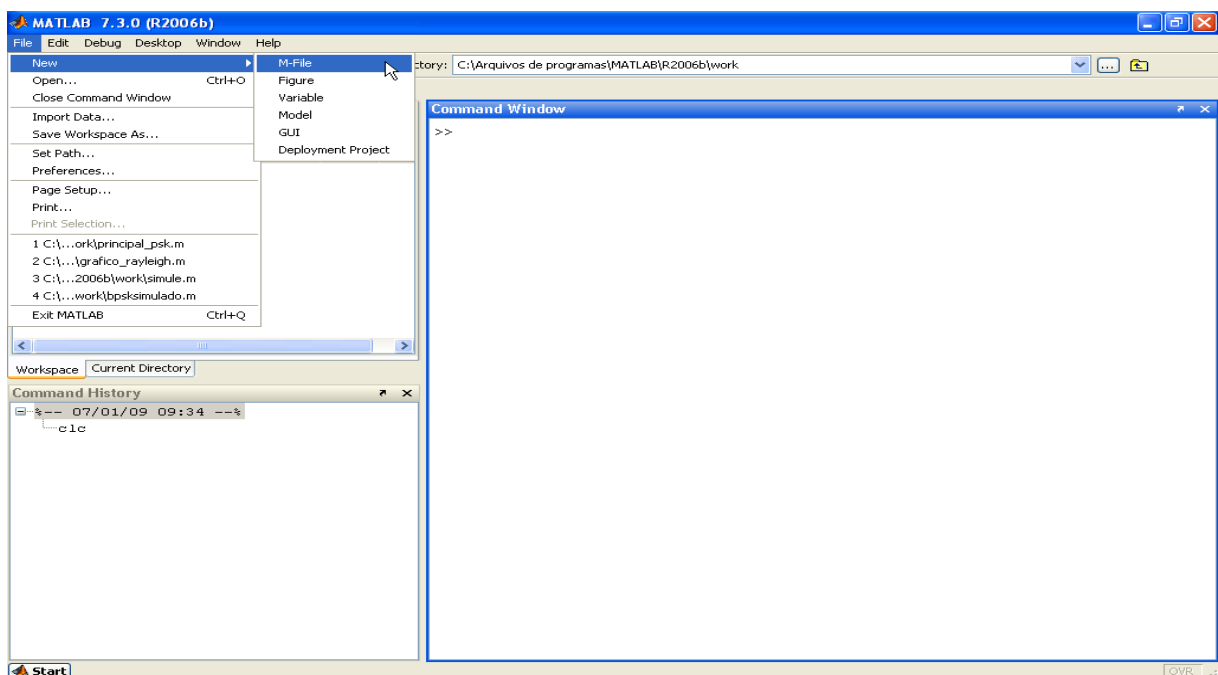
Geralmente, pode-se utilizar a janela de comandos do MATLAB para digitar os comandos. Entretanto este procedimento bastante simples para execução de comandos é ineficaz quando a complexidade do problema aumenta. Para problemas simples podemos usar a janela de comandos do MATLAB e para os mais difíceis devemos utilizar o arquivo M-file, que é também conhecido como *arquivo de comando ou arquivo M*. A extensão de um arquivo de comando é *.m*.

### 4.1 Como Criar um Arquivo M (M-file)

Para criar um arquivo M-file, siga os seguintes passos:

- ✓ Inicialmente, você deve abrir o programa MATLAB, dando um duplo click no ícone do MATLAB que está na área de trabalho.
- ✓ Selecione o menu File (dê um click).
- ✓ Selecione o item *New* e em seguida aponte para M-file e dê um click.

Executando-se os três passos anteriores será aberta uma janela do editor de texto. Nesta janela você poderá escrever todos os comandos que forem necessários para a execução do programa. Para exemplificar considere o arquivo *calculos.m* dado abaixo



### Figura 10: Arquivo de Comando (M-File)

Dica: é possível também abrir uma janela para editar o arquivo(M-file), utilizando uma linha de comando, que pode ser digitada na janela de comandos(Command window) do MATLAB. A função *edit* do MATLAB tem a seguinte sintaxe:

```
edit nome_do_arquivo
```

Exemplo: Suponha que queiramos criar um arquivo de comando (M-file), cujo nome é *calculos*.

```
>> edit calculos
```

Ao se digitar a linha de comando acima na janela de comando do MATLAB, a janela MATLAB Editor perguntará ao usuário se deseja criar uma janela de edição, clique na opção sim e será aberta uma janela de edição com o nome *calculos.m*, ou seja, utilizando o comando *edit* já é criado automaticamente um arquivo de comando. A Figura 11 mostra o uso da função *edit*.

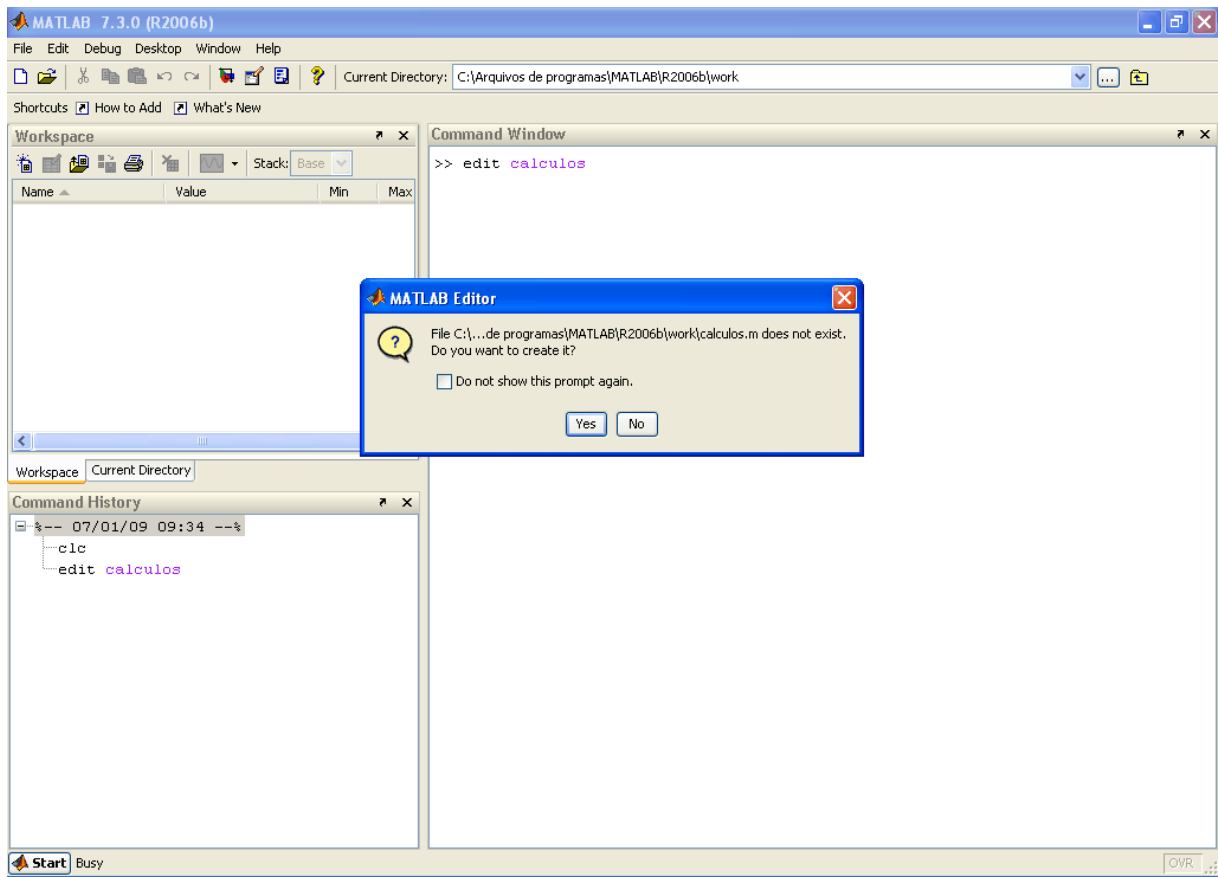


Figura 11: Uso da função edit

Na janela aberta digite os seguintes comandos

% M-file: calculos

% Este programa calcula a soma, produto e divisão de dois inteiros

% x e y

x = 6;

y = 3;

soma = x + y

produto = x\*y

divisao = x/y

## 4.2 Como Executar um Arquivo M (M-file)

Para executar um arquivo M-file no MATLAB é preciso gravar o arquivo correspondente. Para isto, selecione o item *Save Workspace As* do menu *File*. Dê preferência salvar o arquivo no diretório corrente do MATLAB, ou seja, na pasta work.

Em seguida, para executar o arquivo M, vá na janela de comandos do MATLAB e digite o nome do arquivo salvo anteriormente, sem a extensão '.m'. Feito isso o MATLAB gera o executável do programa.

Para exemplificar, considere o arquivo *calculos.m*. Ao digitarmos na janela de comandos do MATLAB o nome do arquivo *calculos* e pressionar a tecla enter. Será exibido na tela os resultados. A Figura 12 exhibe os resultados na janela de comando do MATLAB.

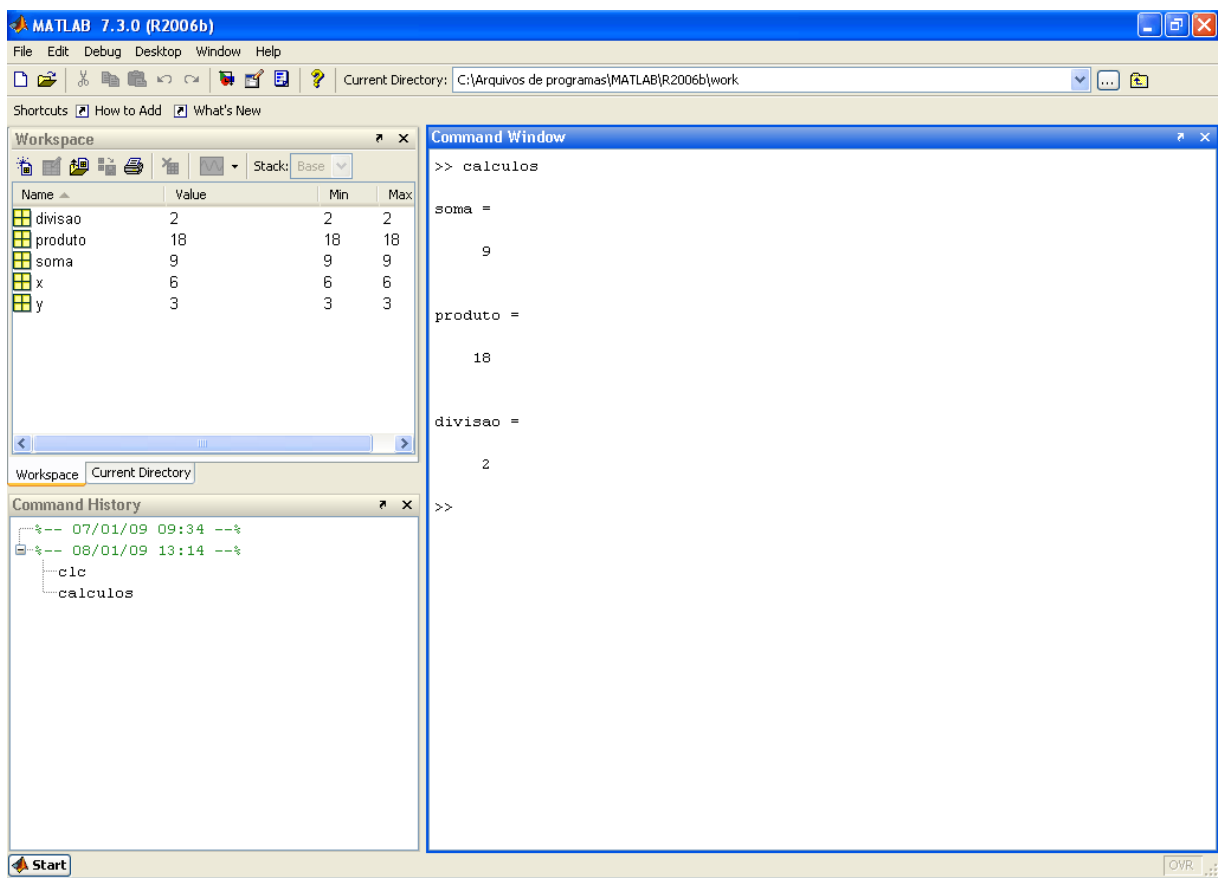



Figura 12: Executando um arquivo de comando

Outra possibilidade de executar o arquivo de comando é por meio da opção *Run*, localizada na barra de ferramentas da janela de editor de texto. Basta clicar com o botão esquerdo do mouse no símbolo,  e visualizar os dados na janela de comando. A Figura 13 mostra a execução através da opção *Run*.

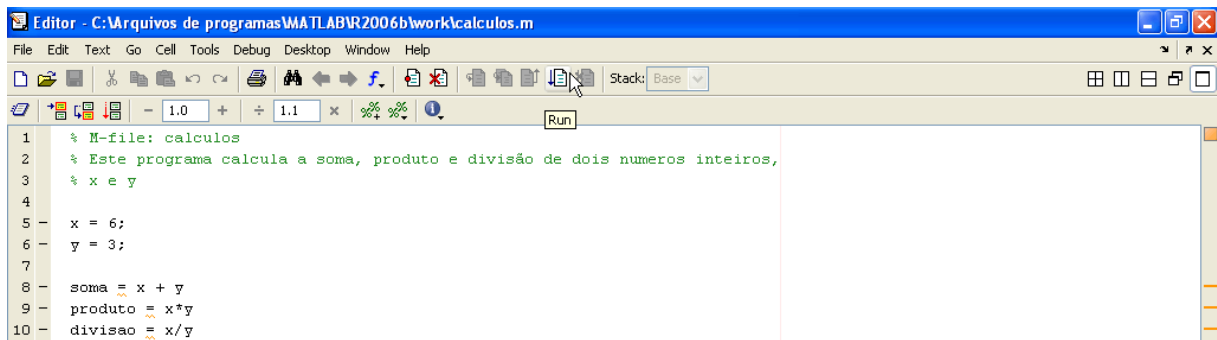


Figura 13: Execução do Arquivo de Comando calculos através do Run

Resumindo, podemos dizer de uma maneira bem simplificada que um programa em MATLAB consiste na criação do arquivo de comando e chamando o arquivo por linha de comando na janela de comando do MATLAB.

## 5 Controle de Fluxo

O MATLAB utilizado como linguagem de programação tem comando para o controle de fluxo: Algumas das estruturas são *if*, *for* e *while*. Esses comandos são equivalentes aos utilizados em outras linguagens de programação como C, C++. A seguir será descrita cada uma dessas estruturas de controle.

### if

O comando *if* é uma estrutura de seleção única que executa uma ação específica quando a condição a ser testada é verdadeira. O comando *if* tem a forma

*if* condição a ser testada

Bloco de código % executado quando a condição do *if* é verdadeira.

*end*

O diagrama de blocos mostrado na figura 14, abaixo mostra uma representação em fluxograma.

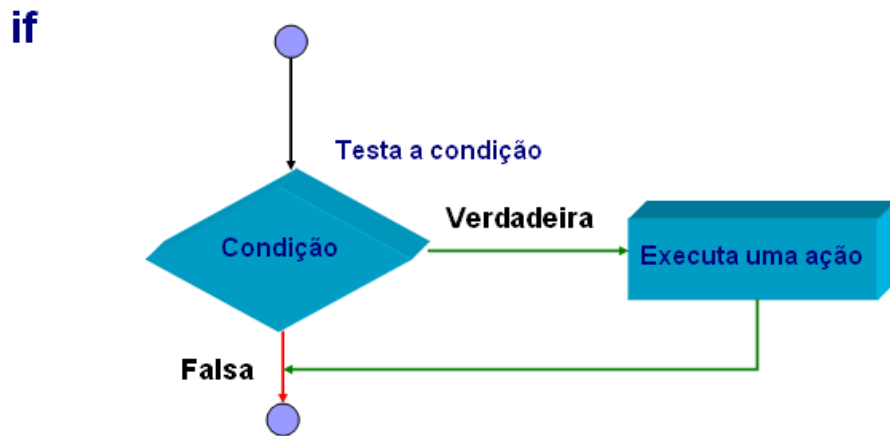


Figura 14: Fluxograma do comando If

**Exemplo:** Dada a nota do aluno, imprima a mensagem Aprovado se ele obtive nota maior ou igual a 7.

```

nota = 8
if nota >= 7
    fprintf('Aprovado');
end
  
```

### **if else**

O comando if-else é uma estrutura de seleção dupla, ou seja, executa o bloco de código do if quando a condição testada é verdadeira, e executa o bloco de código do else quando a condição a ser testada é falsa. O comando if-else tem a forma.

```

if condição a ser testada
    Bloco de código           % executado quando a condição do if é verdadeira.
else
    Bloco de código           % executado quando a condição do if é falsa.
end
  
```

O diagrama de blocos mostrado na figura 15, abaixo mostra uma representação em fluxograma.

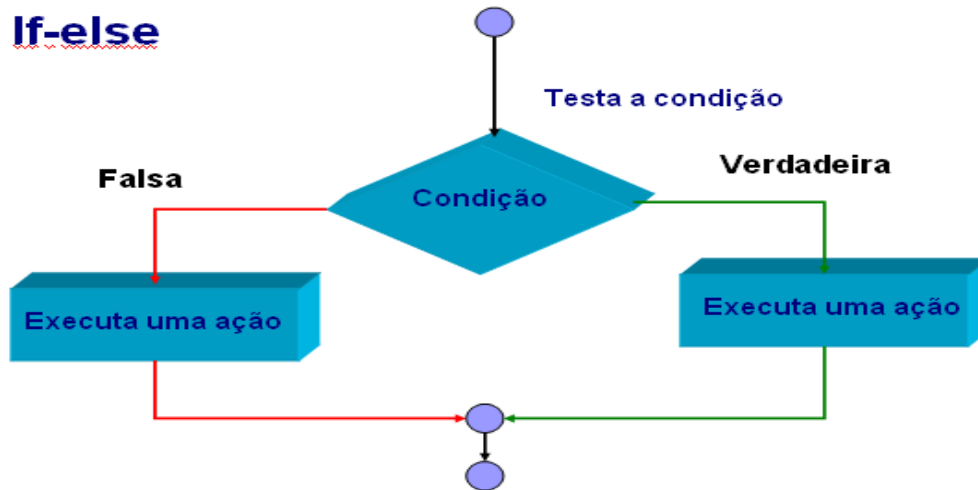


Figura 15: Fluxograma do comando If-else

**Exemplo:** Dada a nota do aluno, imprima a mensagem Aprovado se ele obtiver nota maior ou igual a 7 e imprima Reprovado se a nota for inferior a 7,0.

```

nota = 8
if nota >= 7
    fprintf('Aprovado');
else
    fprintf('Reprovado');
end
  
```

## While

O `while` é uma importante estrutura para repetição de um grupo de comandos quando a condição especificada for verdadeira. O formato geral para esta estrutura de controle é:

```

while condição a ser testada
Bloco de código           % executado quando a condição do if é verdadeira.
end
  
```



O diagrama de blocos mostrado na figura 16, mostra uma representação em fluxograma do comando while.

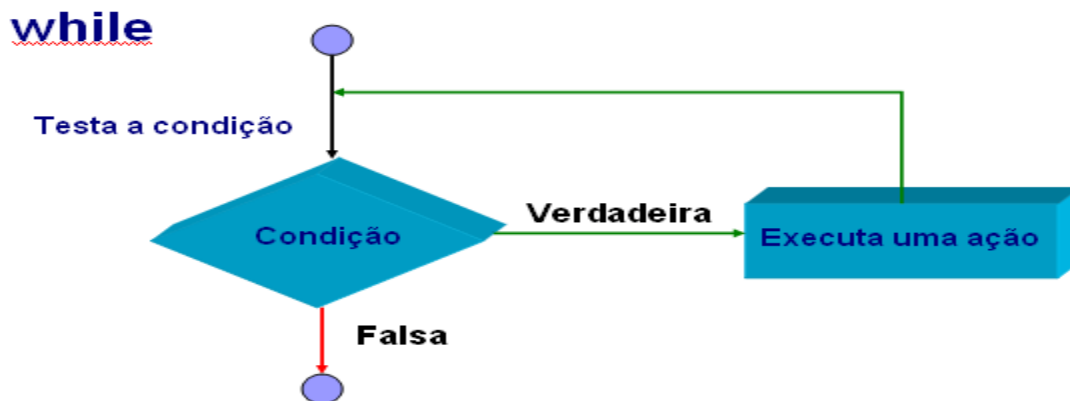


Figura 16: Fluxograma do comando while

**Exemplo:** Imprima os números inteiros de 1 a 10, usando a estrutura while

```

cont = 1                % inicializa contador com 1.
While cont <= 10       % Testa a condição
    disp(cont);
    disp('\n');
    cont = cont + 1;    % incrementa o contador em 1
end
  
```

## Switch

Este comando executa um grupo de instruções, baseado em uma condição lógica. O switch é uma estrutura de seleção múltipla, que consiste em uma série de rótulos case e um otherwise. O formato geral do comando switch é:

```

switch condição a ser testada (escalar ou string)
    case valor1
        instruções
    case valor2
        instruções
    ...
    otherwise
        instruções
end
  
```

O diagrama de blocos mostrado na figura 17, mostra uma representação em fluxograma do comando switch.

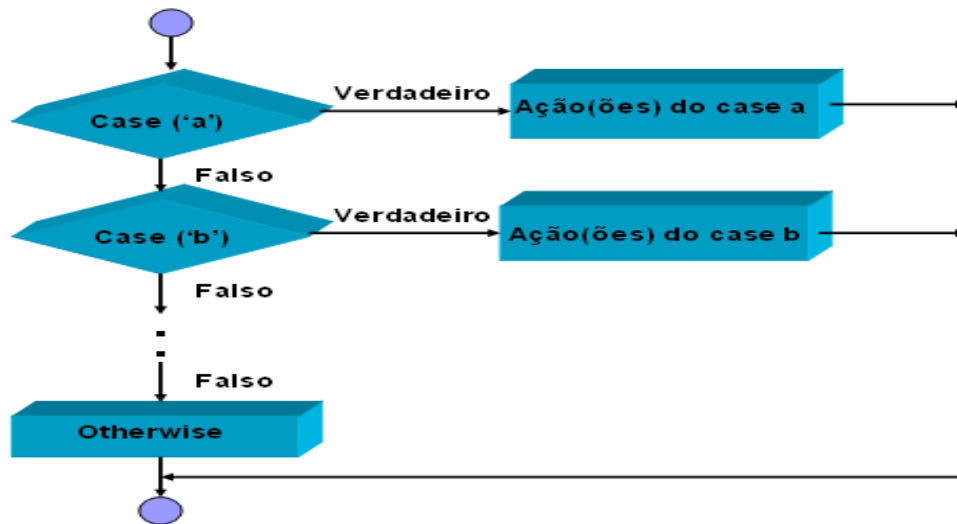


Figura 17: Fluxograma do comando switch

## For

Este comando executa um grupo de instruções um determinado número de vezes. Ao utilizar o laço for sabemos o número de iterações a ser executado, ao contrário do comando while que depende de uma condição lógica. O formato geral do comando for é:

```
for índice = início:incremento:fim
instruções
end
```

O diagrama de blocos mostrado na figura 18, mostra uma representação em fluxograma do comando for. No fluxograma abaixo, considere  $n = 3$ .

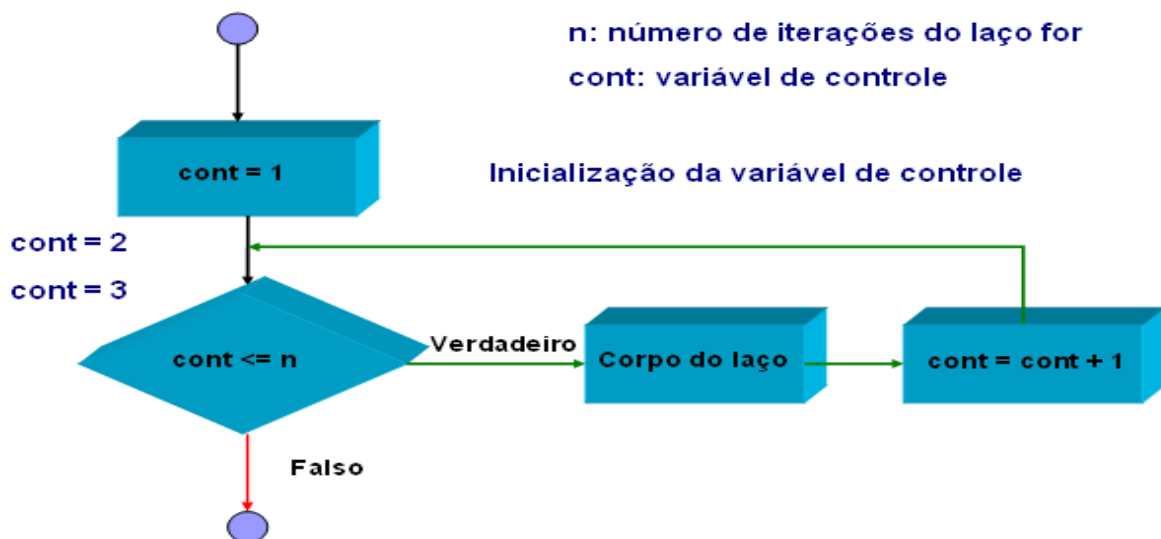


Figura 18: Fluxograma do comando for.

**Exemplo:** Calcular a soma dos 10 primeiros inteiros.

```
soma = 0; % Inicializa a variável soma com zero
for k = 1:10
    soma = soma + k;
end
```

```
fprintf('A soma dos dez primeiros inteiros eh: %.2f, soma);
```

O MATLAB também possui outros recursos para facilitar o controle do fluxo do algoritmo, dentre eles podemos destacar: *return*, *break* e *continue*. Para informações a cerca destes comando use a função *help* e faça uma consulta.

## Simulação do 4-PSK

Nesta parte iremos detalhar com arquivos de comando a simulação do 4-PSK e comparar as curvas simuladas e teóricas, de probabilidade de erro de bit em função da relação sinal-ruído.

O programa será dividido em 4 arquivos para facilitar o reuso do código para outros esquemas de modulação. Iremos criar os seguintes arquivos [bpsk.m](#), [prob\\_erro1.m](#), [gera\\_ruido.m](#), e [bpsk4.m](#)

Crie um arquivo de comando com o nome [bpsk.m](#) e digite neste arquivo os comandos mostrados a seguir. Este arquivo quando executado mostra a curva de probabilidade de erro de bit em função da relação sinal ruído.

*%Script File: bpsk.m*

```
% UNIVERSIDADE FEDERAL DE CAMPINA GRANDE - UFCG
% PROGRAMADOR: JOSENILDO FERREIRA GALDINO
% E-MAIL: JOSENILDOFG@GMAIL.COM
% DATA:20/09/08
% CURVA DE PROBABILIDADE DE ERRO EM FUNÇÃO DA RSR
%DADA EM DB PARA O 4-PSK
%
% -----
% 4-PSK
% -----
```

```
clear all
clf
snr_db = 0:0.025:35
snr = 10.^(snr_db)
P = 1 - (1 - 0.5*erfc(sqrt(snr)) ).^2
semilogy(snr_db,P,'r');
axis([0 35 10e-7 1]);
grid on;
legend('4-PSK');
hold on;
```

Crie um outro arquivo de comando com o nome [prob\\_erro1](#) e digite neste arquivo os comandos mostrados a seguir. Este arquivo cria uma função para calcular a probabilidade de erro de bit.

*% Script File: prob\_erro1.m*

```

function Pe = prob_erro1(snrdb)
% PROB_ERRO - Calcula a probabilidade de erro dada uma
% relação sinal ruído em db, para 4-PSK
M = 4;
snr = 10^(0.1*snrdb);    % Relação Sinal-Ruído
E = 1;
variancia = E/snr;    % Variância do Ruído
N = 10000;
s = randint(N,2);
erros = 0;
%-----
for k = 1:N
    a0 = s(k,2);
    a1 = s(k,1);
    a(k) = s(k,1)*2 + s(k,2)*1;
    if (a(k) == 0)
        s_mod(k) = 1 + 0*i;
    elseif (a(k) == 1)
        s_mod(k) = 0 + i;
    elseif (a(k) == 3)
        s_mod(k) = -1 + 0*i;
    else
        s_mod(k) = 0 -i;
    end
end
%-----
r = gera_ruído(N,variancia/4);    % Ruído branco
s1 = s_mod + r';    % Sinal + ruído branco
s2 = angle(s1);
%-----
for k = 1:N
    if(s2(k) < 0 )
        s2(k) = s2(k) + 2*pi ;
    end
end
%-----
for n = 1:N
    if( s2(n) < (pi/4) | s2(n) > (7*pi/4))
        s_est = 0;
    elseif( s2(n) <= (3*pi/4))
        s_est = 1;
    end
end

```

```

elseif( s2(n) <= (5*pi/4))
    s_est = 3;
else
    s_est = 2;
end
if(real(s_est) ~= real(a(n)))
    erros = erros + 1;
end
end
Pe = erros/N;          % Probabilidade de erro

```

Crie um outro arquivo de comando com o nome `gera_ruido.m` e digite neste arquivo os comandos mostrados a seguir. Este arquivo cria uma função para gerar o ruído AWGN.

*%Script File: gera\_ruido.m*

```

function ruido = gera_ruido(TAM_BLOCO,variancia)

r1 = sqrt(variancia)*(randn(TAM_BLOCO,1)); % Ruido Real Gerado
r2 = sqrt(variancia)*(randn(TAM_BLOCO,1)); % Ruido Imaginario
Gerado

ruido = r1 + i*r2;

```

Crie um arquivo de comando com o nome `bpsk4.m` e digite neste arquivo os comandos mostrados a seguir. Este é o arquivo principal e que deve ser executado para simular o 4-PSK

*%Script File: bpsk4.m*

```

% UNIVERSIDADE FEDERAL DE CAMPINA GRANDE - UFCG
% PROGRAMADOR: JOSENILDO FERREIRA GALDINO
% E-MAIL: JOSENILDOFG@GMAIL.COM
% DATA:28/09/08
% CURVA DE PROBABILIDADE DE ERRO EM FUNÇÃO DA RSR
DADA % EM DB.
% 4-PSK
% -----

```

```

% 4-PSK - CURVA TEORICA
%-----
% 4-PSK
%-----
clc
clf
clear all
snr_i = 0;
snr_inc = 1;
snr_f = 8;
snr_db = snr_i:snr_inc:snr_f
for n = 1:length(snr_db)
    snr = 10^(0.1*snr_db(n));
    P(n) = 1 - (1 - 0.5*erfc(sqrt(snr)) )^2
end;
semilogy(snr_db,P,'r');
axis([0 8 10e-7 1]);
grid on;
xlabel('snr');
ylabel('Pe');
legend('4-PSK');
hold on;
%-----
% 4-PSK - CURVA ESTIMADA
%-----
for t = 1:length(snr_db)
    probabilidade_erro(t) = prob_erro1(snr_db(t))
end
semilogy(snr_db,probabilidade_erro,'*') %Probabilidade de erro estimado.

```

Após a criação dos 4 arquivos, execute o arquivo principal (bpsk4) na área de trabalho do MATLAB e observe a curva teórica e a curva estimada. O resultado deve ser similar ao mostrado na figura 19.

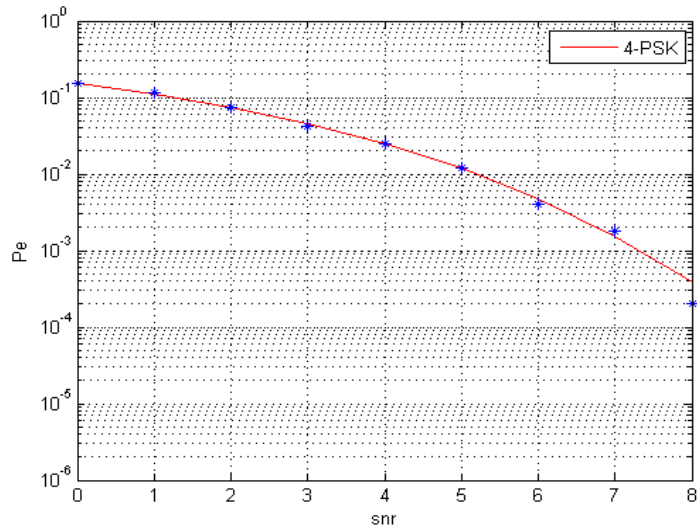


Figura 19: 4-PSK

## Simulação do 4-QAM

Nesta parte iremos detalhar com arquivos de comando a simulação do 4-QAM e comparar as curvas simuladas e teóricas, de probabilidade de erro de bit em função da relação sinal-ruído.

O programa será dividido em 3 arquivos para facilitar o reuso do código para outros esquemas de modulação. Iremos criar os seguintes arquivos [qam4.m](#), [prob\\_erro3.m](#), e [simulqam4.m](#)

Crie um arquivo de comando com o nome [qam4.m](#) e digite neste arquivo os comandos mostrados a seguir. Este arquivo quando executado mostra a curva de probabilidade de erro de bit em função da relação sinal ruído.

*Script File: qam4.m*

```
% UNIVERSIDADE FEDERAL DE CAMPINA GRANDE - UFCG
% PROGRAMADOR: JOSENILDO FERREIRA GALDINO
% E-MAIL: JOSENILDOFG@GMAIL.COM
% DATA:11/10/08
% CURVA DE PROBABILIDADE DE ERRO EM FUNÇÃO DA RSR
DADA % EM DB.
```



```

% Curva Teórica para o 4-QAM
% -----
clc
clear all
clf

% Curva Teórica do 4-QAM
% -----

snr_i = 0; snr_f = 20; snr_inc = 1;
snr_db = snr_i:snr_inc:snr_f
M = 4;
Pb = zeros(1,length(snr_db));
for indice_snr = 1:length(snr_db)
    snr = 10^(0.1*snr_db(indice_snr));
    for alpha=1:log2(sqrt(M))
        for m=0:((1-2^(-alpha))*sqrt(M)-1)
            Pb(indice_snr) = Pb(indice_snr) + ...
                ((-1)^fix(m*2^(alpha-1)/sqrt(M)) * (2^(alpha-1)-fix(m*2^(alpha-1)/sqrt(M)+0.5))*...
                erfc((2*m+1)*sqrt(3*log2(M)*snr/(2*(M-1)))))/sqrt(M);
        end
    end

    Pb(indice_snr) = Pb(indice_snr)/log2(sqrt(M));
end
semilogy(snr_db,Pb,'b');
grid on;
hold on;

```

Crie um outro arquivo de comando com o nome `prob_erro3` e digite neste arquivo os comandos mostrados a seguir. Este arquivo cria uma função para calcular a probabilidade de erro de bit.

*%Script File: prob\_erro3.m*

```

function Pe = prob_erro3(snrdb)
% PROB_ERRO - Calcula a probabilidade de erro dada uma
% relação sinal ruído em db, para 4-QAM

M = 4;
Es = 1;

```

```
No = Es*10^(-0.1*snrdb)/log2(M);
variancia = No/2; % Duas dimensões
```

```
N =10000;
```

```
%-----
s = randint(N,2); %Bits da fonte.
erros = 0;
%-----
%MODULACAO
```

```
for k = 1:N
    a0 = s(k,2);
    a1 = s(k,1);
    a(k) = s(k,1)*2 + s(k,2);
    if (a(k) == 0)
        s_mod(k) = (1+i);
    elseif (a(k) == 1)
        s_mod(k) = (-1+i);
    elseif (a(k) == 2)
        s_mod(k) = (1-i);
    else
        s_mod(k) = (-1-i);
    end
end
```

```
end
%-----
%Ajuste de Energia
```

```
s_mod = s_mod*(sqrt(2)/2);
```

```
%-----
% Canal AWGN
r = gera_ruido(N,variancia);           % Ruido branco
x = s_mod + r';                        % Sinal + ruido branco
```

```
%-----
-
%Ajuste de Energia
```

```
x = x/sqrt(2)/2;
```

```
for n = 1:N
    if( real(x(n)) >= 0 )
```

```

if ( imag(x(n)) >= 0)
    s_est(n,1) = 0; s_est(n,2) = 0;%s_est = 0;
else
    s_est(n,1) = 1; s_est(n,2) = 0;%s_est = 2;
end
else
    if imag(x(n)) > 0
        s_est(n,1) = 0; s_est(n,2) = 1;%s_est = 1;
    else
        s_est(n,1) = 1; s_est(n,2) = 1;%s_est = 3;
    end
end
end
end

for n = 1:N
    for k=1:2
        if(s_est(n,k)~=s(n,k)) erros = erros + 1;
        end
    end
end
end
Pe = erros/(N*log2(M));

```

Crie um arquivo de comando com o nome [simulqam4.m](#) e digite neste arquivo os comandos mostrados a seguir. Este é o arquivo principal e que deve ser executado para simular o 4-QAM

*%Script File: simulqam4.m*

```

% UNIVERSIDADE FEDERAL DE CAMPINA GRANDE - UFCG
% PROGRAMADOR: JOSENILDO FERREIRA GALDINO
% E-MAIL: JOSENILDOFG@GMAIL.COM
% DATA:01/10/08
clc
clear all
clf
% Curva Teórica do 4-QAM
% -----
snr_i = 0; snr_f = 8; snr_inc = 1;
snr_db = snr_i:snr_inc:snr_f

M = 4;
Pb = zeros(1,length(snr_db));

```

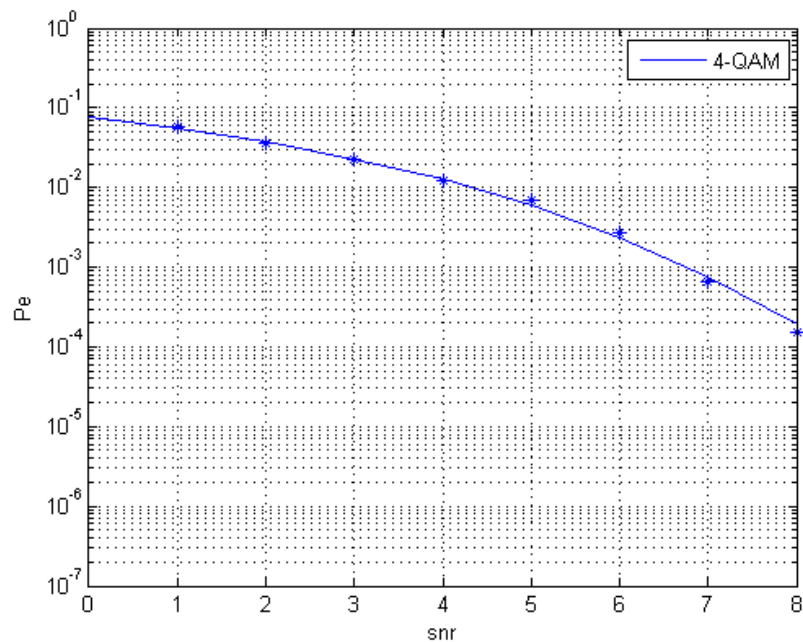
```

for indice_snr = 1:length(snr_db)
    snr = 10^(0.1*snr_db(indice_snr));
    for alpha=1:log2(sqrt(M))
        for m=0:((1-2^(-alpha))*sqrt(M)-1)
            Pb(indice_snr) = Pb(indice_snr) + ...
                ((-1)^fix(m*2^(alpha-1)/sqrt(M)) * (2^(alpha-1)-fix(m*2^(alpha-1)/sqrt(M)+0.5))*...
                erfc((2*m+1)*sqrt(3*log2(M)*snr/(2*(M-1)))))/sqrt(M);
        end
    end
    Pb(indice_snr) = Pb(indice_snr)/log2(sqrt(M));
end
semilogy(snr_db,Pb,'b');
grid on;
hold on;

% Curva Estimada do 4-QAM
for k = 1:length(snr_db)
    probabilidade_erro(k) = prob_erro3(snr_db(k))
end
semilogy(snr_db,probabilidade_erro,'*') % Probabilidade de erro
estimado.
axis([snr_i snr_f 1e-7 1]);
legend('4-QAM');
xlabel('snr');
ylabel('Pe');

```

Após a criação dos 3 arquivos, execute o arquivo principal (simulqam4) na área de trabalho do MATLAB e observe a curva teórica e a curva estimada.



## Questões

- 1) Com os arquivos de comandos criados anteriormente, crie um novo arquivo que trace as curvas teóricas do 4-PSK e 4-QAM em um mesmo gráfico para efeito de comparação. Dica: Considere uma faixa de 30 dB para a relação sinal-ruído.
- 2) Observando o gráfico gerado anteriormente, preencha as tabelas 1 e 2 dada abaixo. Dica: Use o Data cursor para observar os valores para cada um dos valores da relação sinal-ruído.

Tabela 1: 4-PSK

Eb/N0	1	3	5	7	9
BER					
Eb/N0	11	12	13	14	15
BER					

Tabela 2: 4-QAM

Eb/N0	1	3	5	7	9
BER					
Eb/N0	11	12	13	14	15
BER					

3) Analisando os dados da tabela anteriormente e o gráfico das curvas teóricas, quais as conclusões que pode-se tirar. A partir de que valor da relação sinal-ruído é mais eficiente o 4-QAM.

