



Curso de Simulação de Sistemas

CEFET-RJ – Centro Federal de Educação Tecnológica Celso Suckow da
Fonseca
Curso Técnico de Eletrotécnica

Ano 2005 - Versão 2.2

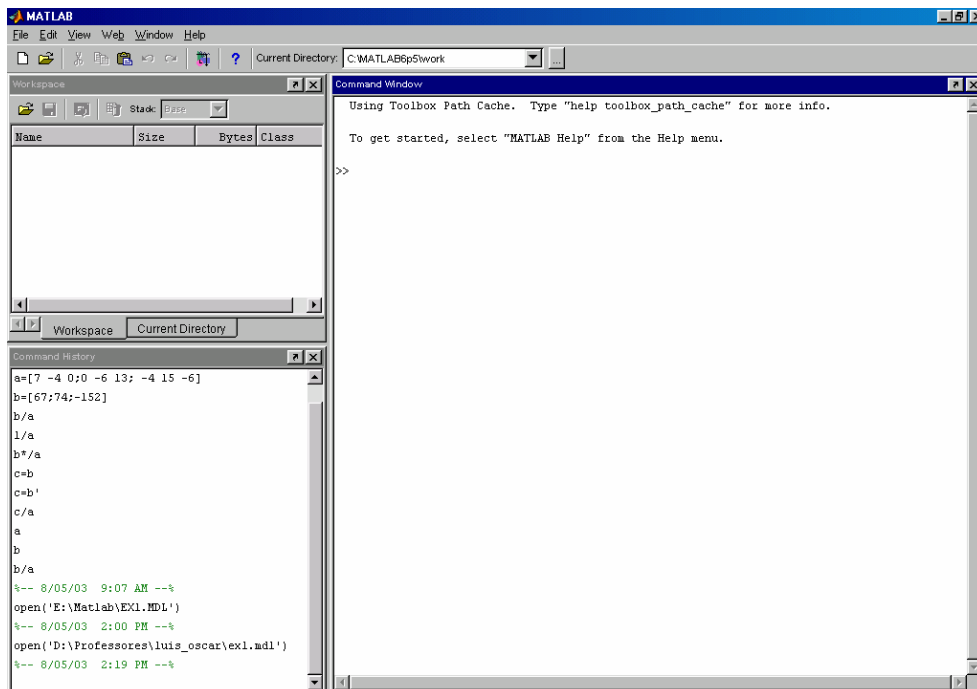
Professor : Luís Oscar de Araujo Porto Henriques

1 Introdução

1.1 O que é o MATLAB?

MATLAB é um "software" interativo de alta performance voltado para o cálculo numérico. O MATLAB integra análise numérica, cálculo com matrizes, processamento de sinais e construção de gráficos em ambiente fácil de usar onde problemas e soluções são expressos somente como eles são escritos matematicamente, ao contrário da programação tradicional.

O MATLAB é um sistema interativo cujo elemento básico de informação é uma matriz que não requer dimensionamento. Esse sistema permite a resolução de muitos problemas numéricos em apenas uma fração do tempo que se gastaria para escrever um programa semelhante em linguagem C ou Java. Além disso, as soluções dos problemas são expressas no MATLAB quase exatamente como elas são escritas matematicamente.



1.2 Carregando o MATLAB

No Gerenciador de Programas do Microsoft Windows deve-se abrir o grupo de programas do MATLAB for Windows, que contém o ícone do aplicativo MATLAB. Um duplo clique no ícone MATLAB carrega o aplicativo MATLAB.

Quando o MATLAB é carregado, 3 janelas são exibidas: a Janela de Comando (Command Windows), area de trabalho (Workspace) e Histórico (Command History). A Janela de Comando é ativada quando se inicializa o MATLAB, e o "prompt" padrão (>>) é exibido na tela.

1.3 Usando o Help

Uma vez que você está dentro do MATLAB, você pode pedir ajuda usando o comando `help`. Você pode usar os seguintes comandos:

```
>> help
```

ou

```
>> help comando
```

Usando-se o comando `Demo`, temos uma demonstração das capacidades do MATLAB.

1.4 Fazendo um exemplo simples

Suponha que temos dois pontos P1 e P2 cujas coordenadas são:

P1=(1,5) e P2=(4,7)

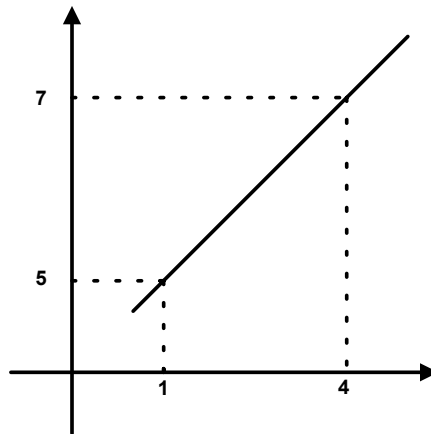
Queremos calcular a distancia entre dois pontos, que é a hipotenusa de um triângulo retângulo, conforme mostra a figura abaixo. Usando o teorema de pitágoras, podemos calcular a distancia d com a seguinte equação:

$$d = \sqrt{s_1^2 + s_2^2}$$

$$d = \sqrt{(4-1)^2 + (7-5)^2}$$

$$d = \sqrt{13}$$

$$d = 3,61$$



Como solucionaríamos este sistema no matlab?

(o sinal de percentagem serve como comentário no matlab)

```
% este programa calcula e imprime a distancia em linha reta, entre dois pontos
```

```
p1x=1; %ponto 1 – eixo x  
p1y=5; %ponto 1 – eixo y  
p2x=4; %ponto 2 – eixo x  
p2y=7; %ponto 2 – eixo y
```

```
d=sqrt((p2x-p1x)^2+(p2y-p1y)^2) % calcula a distância
```

ao executarmos estes comandos teremos o resultado:

```
d=3.6056
```

Esta saída coincide com o valor que calculamos no exemplo manual. Se a solução MATLAB não coincidir com o exemplo manual, devemos rever ambas soluções a fim de encontrar o erro.

Devemos agora aprender a montar um programa diretamente em um arquivo de extensão *.m.

Clique em file->new->M-file e escolha a abertura de um novo arquivo *.m.

1.5 Outro exemplo

Para entrar com uma matriz pequena, por exemplo usa-se

```
>> A = [1 2 3; 4 5 6; 7 8 9]
```

colocando colchetes em volta dos dados e separando as linhas por ponto e vírgula.. Quando se pressiona a tecla <enter> o MATLAB responde com

```
A =  
1     2     3  
4     5     6  
7     8     9
```

Para inverter esta matriz usa-se

```
>> B = inv(A)
```

e o MATLAB responde com o resultado.

(colocar aqui o resultado obtido) O que achou do resultado?

1.6- Funções especiais de matrizes no matlab

`zeros(n)` – Matriz quadrada de zeros de ordem n

`ones(n)` – Matriz quadrada de uns de ordem n

`eye(n)` – Matriz quadrada identidade de ordem n

2 - OPERAÇÕES COM MATRIZES

As operações com matrizes no MATLAB são as seguintes:

- Adição;
- Subtração
- Multiplicação;
- Transposta;

A seguir cada uma dessas operações é mostrada com mais detalhe.

2.1 Transposta

O caracter apóstrofo, " ' ", indica a transposta de uma matriz. A declaração

```
>> A = [1 2 3; 4 5 6; 7 8 0]
```

```
>> B = A'
```

que resulta em

```
A =      1   2   3  
      4   5   6  
      7   8   0
```

```
B =      1   4   7  
      2   5   8  
      3   6   0
```

e

```
>> x = [-1 0 2]'
```

produz

```
x =  
    -1  
     0  
     2
```

Se Z é uma matriz complexa, Z' será o conjugado complexo composto. Para obter simplesmente a transposta de Z deve-se usar $Z.'$, como mostra o exemplo

```
>> Z = [1 2; 3 4] + [5 6; 7 8]*i
```

```
>> Z1 = Z'
```

que resulta em:

```
Z =  
    1.0000 + 5.0000i    2.0000 + 6.0000i  
    3.0000 + 7.0000i    4.0000 + 8.0000i
```

```
Z1 =
      1.0000 - 5.0000i    3.0000 - 7.0000i
      2.0000 - 6.0000i    4.0000 - 8.0000i
```

2.2 Adição e Subtração

A adição e subtração de matrizes são indicadas, respectivamente, por "+" e "-". As operações são definidas somente se as matrizes as mesmas dimensões. Por exemplo, a soma com as matrizes mostradas acima, $A + x$, não é correta porque A é 3×3 e x é 3×1 . Porém,

```
>> C = A + B
```

é aceitável, e o resultado da soma é

```
C =
      2      6      10
      6      10     14
      10     14      0
```

A adição e subtração também são definidas se um dos operadores é um escalar, ou seja, uma matriz 1×1 . Neste caso, o escalar é adicionado ou subtraído de todos os elementos do outro operador. Por exemplo:

```
>> y = x - 1
```

resulta em:

```
y =
      -2
      -1
       1
```

2.3 Multiplicação

A multiplicação de matrizes é indicada por "*". A multiplicação $x*y$ é definida somente se a segunda dimensão de x for igual à primeira dimensão de y . A multiplicação

```
>> x' * y
```

é aceitável, e resulta em

```
ans =
      4
```

É evidente que o resultado da multiplicação $y'*x$ será o mesmo. Existem dois outros produtos que são transpostos um do outro.

```
>> x*y'
```

```
Ans =
      2      1     -1
      0      0      0
     -4     -2      2
```

```
>> y*x'
```

Ans =

```
2    0   -4
1    0   -2
-1   0    2
```

2.4 Divisão

Teoria matemática de divisão de matrizes

Existem dois símbolos para divisão de matrizes no MATLAB "\" e "/". Se A é uma matriz quadrada não singular, então $A \setminus B$ e B / A correspondem respectivamente à multiplicação à esquerda e à direita da matriz B pela inversa da matriz A , ou $\text{inv}(A) * B$ e $B * \text{inv}(A)$, mas o resultado é obtido diretamente. Em geral,

$x = A \setminus B$ é a solução de $A * X = B$

$x = B / A$ é a solução de $X * A = B$

Por exemplo, como o vetor B foi definido como $A * x$, a declaração

```
>> z = A \ B
```

resulta em

z =

```
-1
0
2
```

2.5 Exponenciação

A expressão A^p eleva A à p -ésima potência e é definida se A é matriz quadrada e p um escalar. Se p é um inteiro maior do que um, a exponenciação é computada como múltiplas multiplicações. Por exemplo,

```
>> A^3
```

Ans =

```
279   360   306
684   873   684
738   900   441
```

3 FUNÇÕES

As funções matemáticas são representadas no MATLAB por arquivos ".m". Por exemplo, podemos criar uma função no matlab com um nome de corcova.m:

```
function y = corcova(x)
```

```
y = 1 ./ ((x-.3).^2 + .01) + 1./((x-.9).^2 + .04) - 6;
```

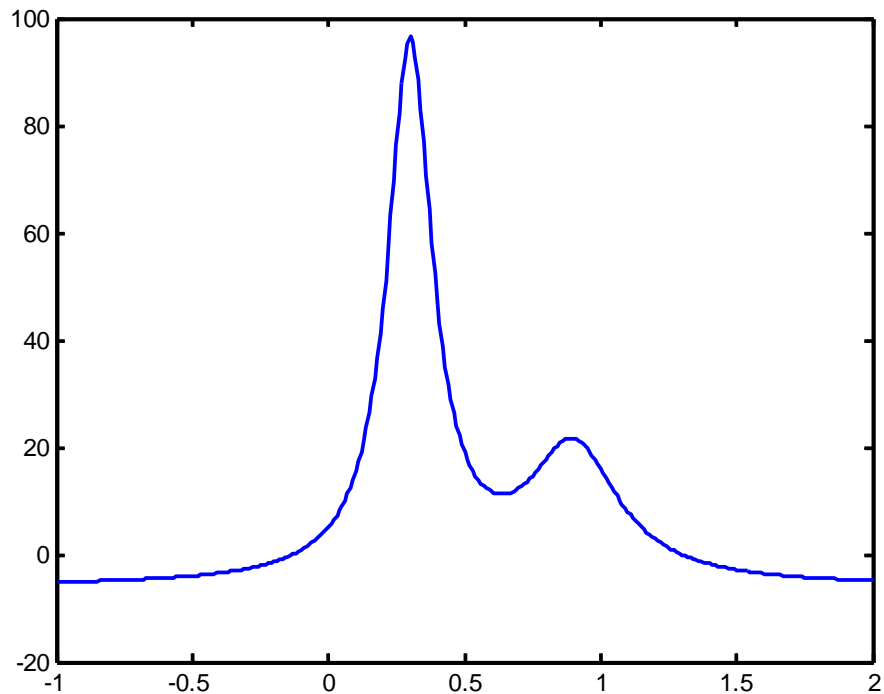
Salve esta função no diretório padrão do matlab.

$$\text{corcova}(x) = \frac{1}{(x-0.3)^2 + 0.001} + \frac{1}{(x-0.9)^2 + 0.04} - 6$$

O gráfico da função é:

```
>> x = -1:0.01:2;  
>> plot(x, corcova(x))
```

Devemos salientar que na primeira linha temos um vetor que começa em -1 e termina em 2 com passo de 0.01.



4 GRÁFICOS

A construção de gráficos no MATLAB é mais uma das facilidades do sistema. Através de comandos simples pode-se obter gráficos bidimensionais ou tridimensionais com qualquer tipo de escala e coordenada. Existe no MATLAB uma vasta biblioteca de comandos gráficos.

4.1 Gráficos Bidimensionais

Estes são os comandos para plotar gráficos bidimensionais:

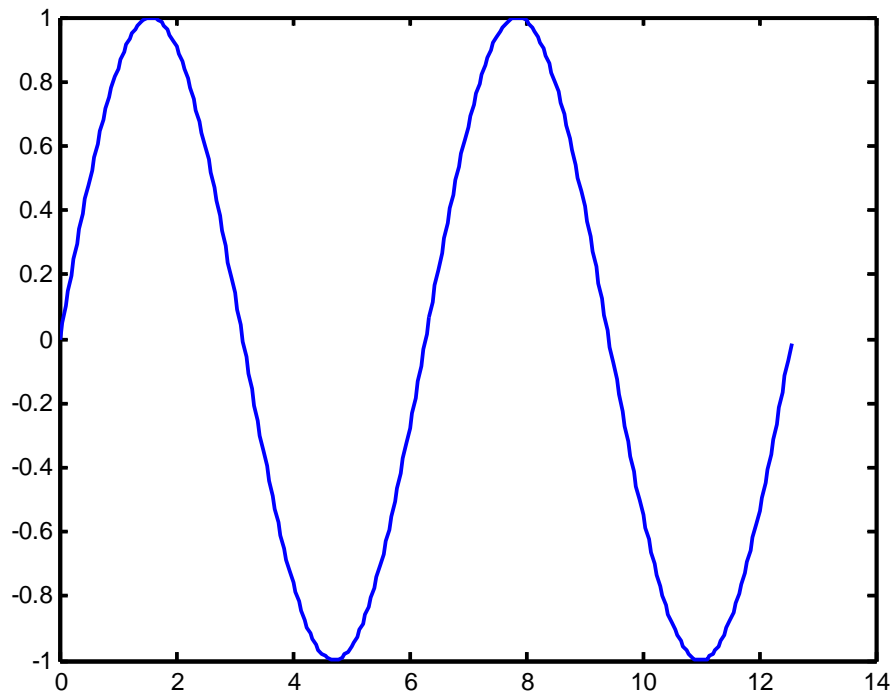
plot	Plotar linear.
loglog	Plotar em escala loglog.
semilogx	Plotar em semilog.

semilogy	Plotar em semilog.
fill	Desenhar polígono 2D.
polar	Plotar em coordenada polar.
bar	Gráfico de barras.
stem	Seqüência discreta.
stairs	Plotar em degrau.
errorbar	Plotar erro.
hist	Plotar histograma.
rose	Plotar histograma em ângulo.
compass	Plotar em forma de bússola.
feather	Plotar em forma de pena.
fplot	Plotar função.
comet	Plotar com trajetória de cometa.

Se **X** e **Y** são vetores com dimensões iguais, o comando **plot(X,Y)** produz um gráfico bidimensional dos elementos de **X** versus os elementos de **Y**, por exemplo

```
>> t = 0:0.05:4*pi;  
>> y = sin(t);  
>> plot(t,y)
```

resulta em



O MATLAB pode também plotar múltiplas linhas e apenas um gráfico. Existem duas maneiras, a primeira é usado apenas dois argumentos, como em **plot(X,Y)**, onde **X** e/ou **Y** são matrizes. Então:

Se **Y** é uma matriz e **X** um vetor, **plot(X,Y)** plota sucessivamente as linhas ou colunas de **Y** versos o vetor **X**.

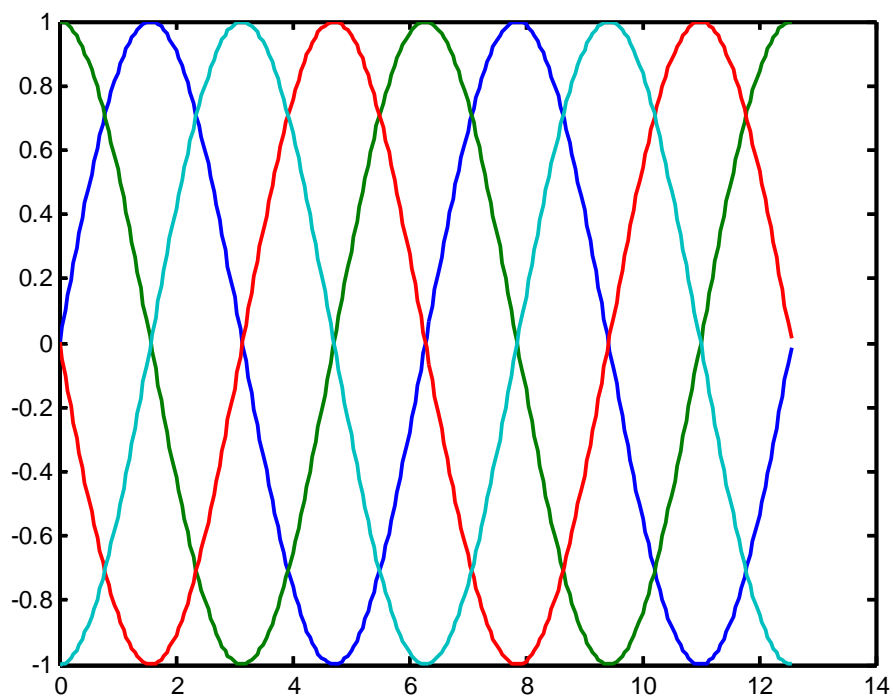
Se **X** é uma matriz e **Y** é um vetor, **plot(X,Y)** plota sucessivamente as linhas ou colunas de **X** versos o vetor **Y**.

Se **X** e **Y** são matrizes com mesma dimensão, **plot(X,Y)** plota sucessivamente as colunas de **X** versos as colunas de **Y**.

Se **Y** é uma matriz, **plot(Y)** plota sucessivamente as colunas de **Y** versos o índice de cada elemento da linha de **Y**.

A segunda, e mais fácil, maneira de plotar gráficos com múltiplas linhas é usando o comando **plot** com múltiplos argumentos. Por exemplo:

```
>> plot(t, sin(t), t, cos(t), t, sin(t + pi), t, cos(t + pi))
```



4.2 Anotações no Gráfico

O MATLAB possui comandos de fácil utilização para adicionar informações em um gráfico:

title	Título do gráfico.
xlabel	Título do eixo-X.
ylabel	Título do eixo-Y.
zlabel	Título do eixo-Z.
text	Inserir anotação no gráfico.

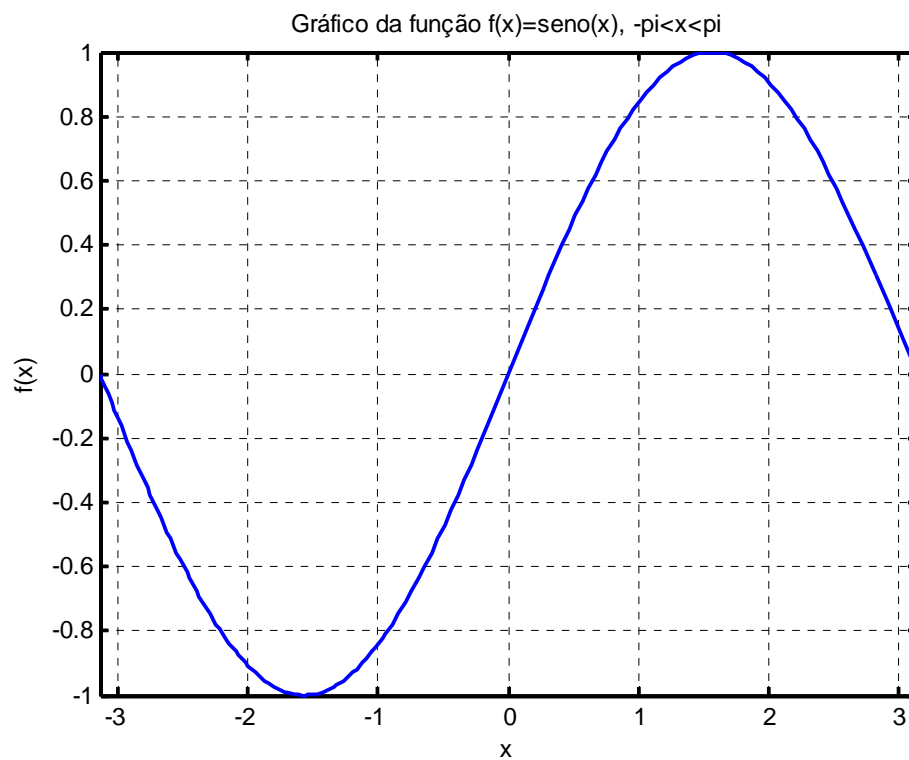
gtext	Inserir anotação com o "mouse".
grid	Linhas de grade.
legend	Cria legenda para os gráficos

Por exemplo:

Coloque título no gráfico, nos eixos e também legenda no gráfico anterior

Outro exemplo:

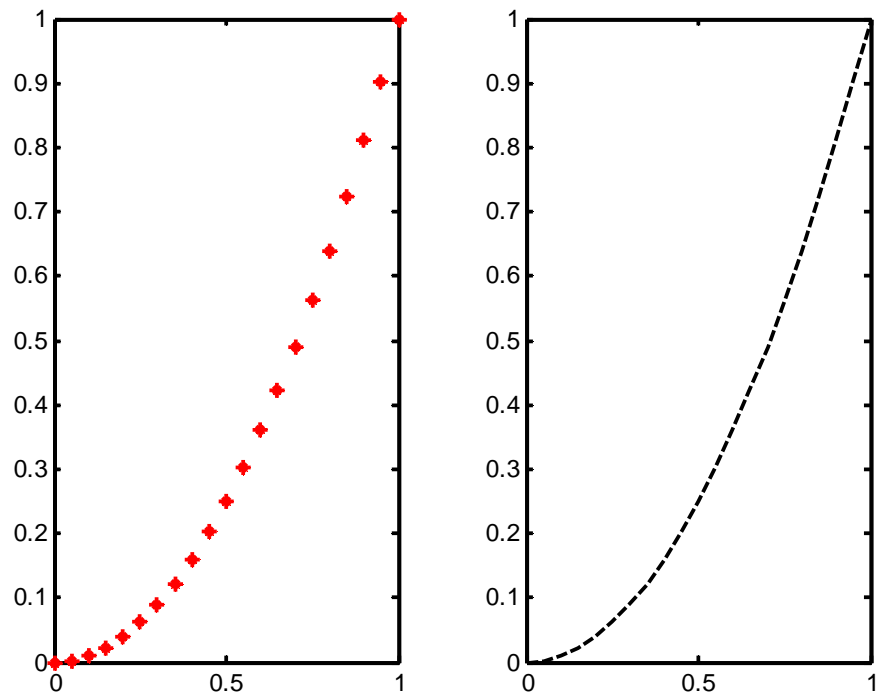
```
>> fplot('sin', [-pi pi])  
>> title('Gráfico da função f(x)=seno(x), -pi<x<pi')  
>> xlabel('x')  
>> ylabel('f(x)')  
>> grid
```



4.3 Estilos de Linha e Símbolo

Os tipos de linhas, símbolos e cores usados para plotar gráficos podem ser controlados se os padrões não são satisfatórios. Por exemplo,

```
>> x = 0:0.05:1;  
>> subplot(1,2,1), plot(x,x.^2,'r*')  
>> subplot(1,2,2), plot(x,x.^2,'k--')
```



Outros tipos de linhas, pontos e cores também podem ser usados:

TIPO DE LINHA		CORES	
-	_____	y	Amarelo
--	-----	m	Lilás
-.	-.-.-.-.-	c	Azul claro
.	r	Vermelho
		g	Verde
		b	Azul escuro
		w	Branco
		k	Preto

TIPO DE PONTO	
.
*	* * * * *
o	o o o o o o o o
+	+++++
x	xx x x x x x
v	Triângulo para baixo
^	Triângulo para cima
d	Losango
s	Quadrado
p	Pentágono
h	hexagono

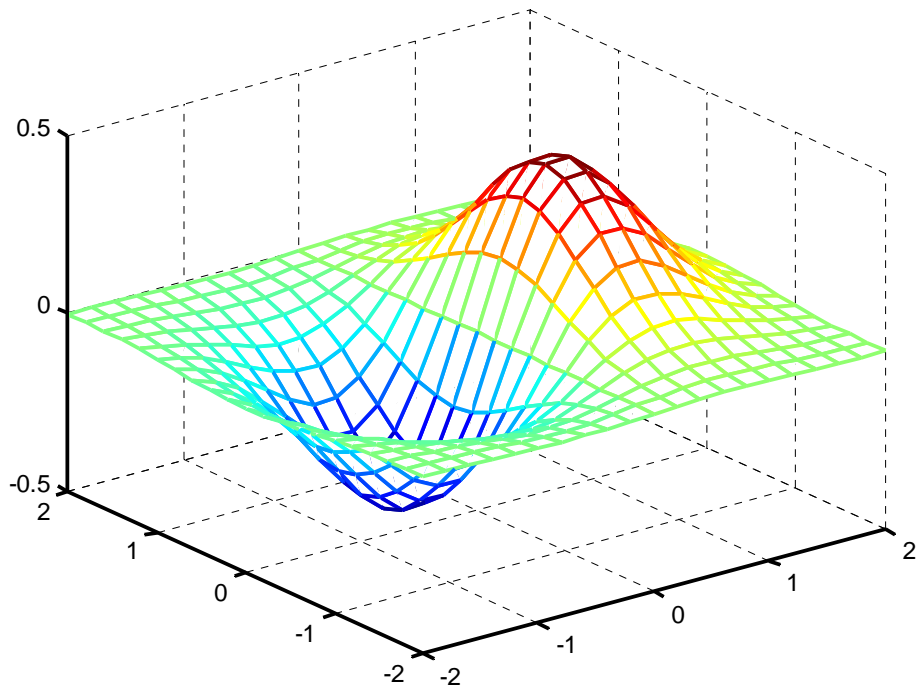
4.4 Plotando Gráficos Tridimensionais e Contornos

Estes são alguns comandos para plotar gráficos tridimensionais e contornos.

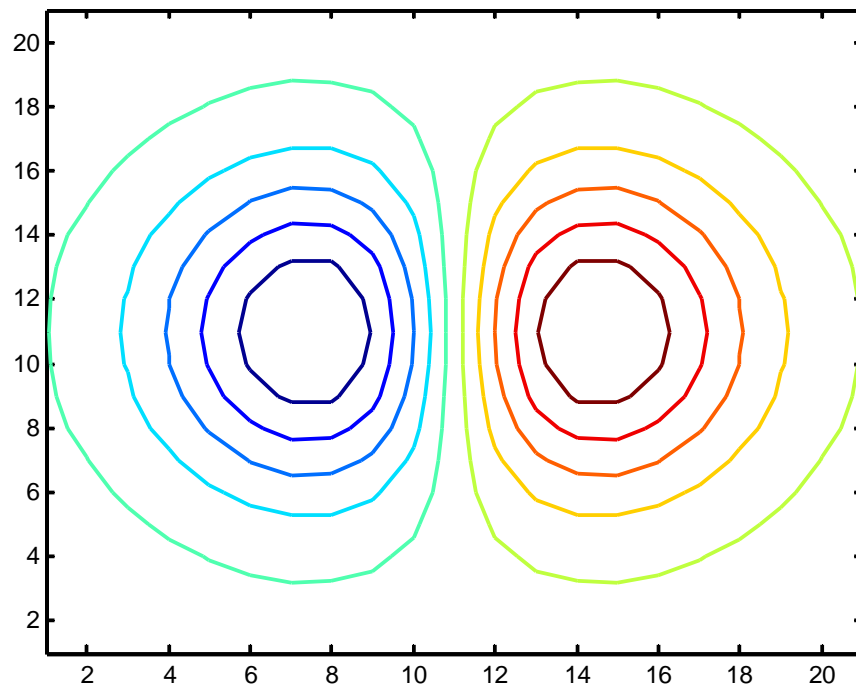
Plot3	Plotar em espaço 3D.
fill3	Desenhar polígono 3D.
comet3	Plotar em 3D com trajetória de cometa.
contour	Plotar contorno 2D.
contour3	Plotar contorno 3D.
clabel	Plotar contorno com valores.
quiver	Plotar gradiente.
mesh	Plotar malha 3D.
meshc	Combinação mesh/contour.
surf	Plotar superfície 3D.
surfc	Combinação surf/contour.
surfil	Plotar superfície 3D com iluminação.
slice	Plot visualização volumétrica.
cylinder	Gerar cilindro.
sphere	Gerar esfera.

O comando **mesh(X,Y,Z)** cria uma perspectiva tridimensional plotando os elementos da matriz **Z** em relação ao plano definido pelas matrizes **X** e **Y**. Por exemplo,

```
>> [X,Y] = meshgrid(-2:.2:2, -2:.2:2);
>> Z = X.* exp(-X.^2 - Y.^2);
>> mesh(X,Y,Z)
```



e o comando `contour(Z,10)` mostra a projeção da superfície acima no plano xy com 10 iso-linhas:

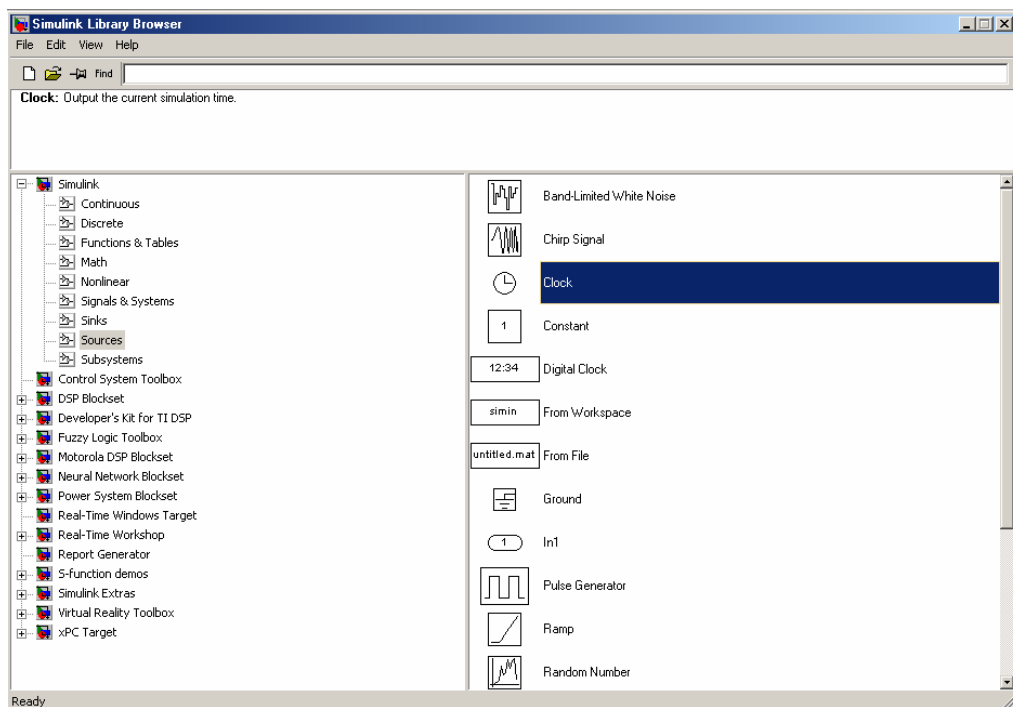


5 - Simulink

5.1 O que é Simulink?

Simulink é um pacote de software para modelar, simular, e analisar sistemas dinâmicos. Suporta os sistemas lineares e não-lineares, modela sistemas contínuos e discretos, ou um híbrido dos dois.

Para modelar, Simulink fornece uma interface gráfica (GUI) para modelagem com diagrama de blocos, usando arraste do mouse. Simulink inclui uma biblioteca detalhada do bloco dos dissipadores, as fontes, componentes lineares e não-lineares, e conectores. Você pode também customizar e criar seus próprios blocos.

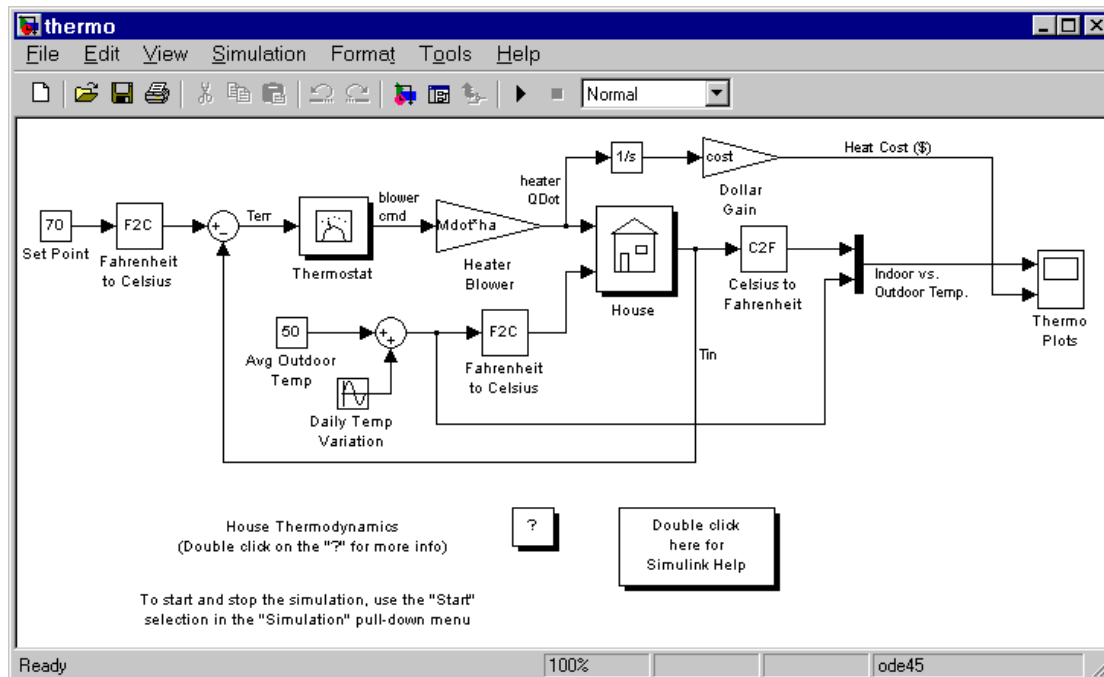


5.2 Rodando um modelo de demonstração

Um exemplo interessante de sistema no Simulink é o modelo termodinâmico de uma casa. Para rodar esta demonstração sigam os passos:

1- Inicie o Matlab

2 - Rode o sistema digitando `thermo` na janela de comando do MATLAB. Este comando inicia o Simulink e cria a janela de modelo que contém a demonstração.



3- Duplo clique no bloco de plotagem chamado de Thermo Plots.

4- o bloco de plotagem mostra 2 gráficos chamados “labeled Indoor vs. Outdoor Temp and Heat Cost (\$)” respectivamente.

5- Para começar a simulação, clique no menu SIMULATION e escolha o comando START, ou então clique o botão Start na barra de ferramenta do Simulink.

6- Enquanto a simulação roda, a temperatura interna e externa aparece no gráfico de temperatura e o preço cumulativo do sistema de aquecimento aparece no gráfico de Custo de temperatura (Heat Cost (\$) plot)

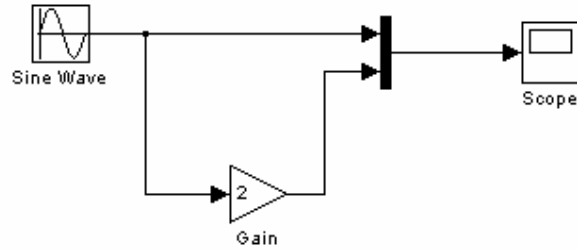
7- Para parar a simulação, escolha o comando stop no menu do simulink.

5.3 Descrição da demonstração

O modelo usado nesta demonstração é bem simples. O termostato é ajustado a 70 graus de Fahrenheit e afetado pela temperatura exterior, que varia aplicando uma onda do seno com amplitude de 15 graus a uma temperatura baixa de 50 graus. Isto simula flutuações diárias da temperatura.

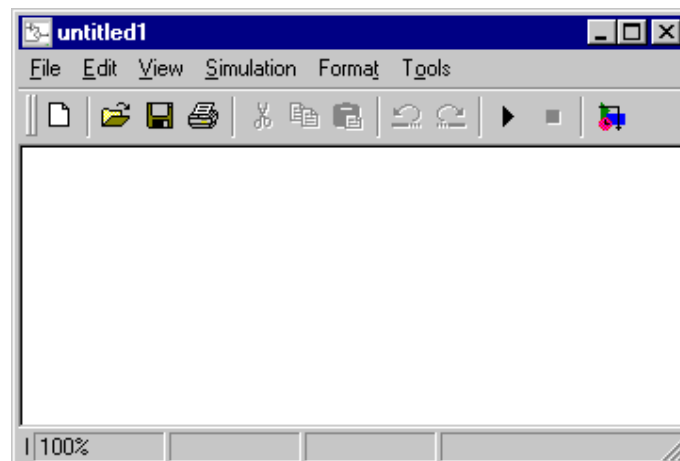
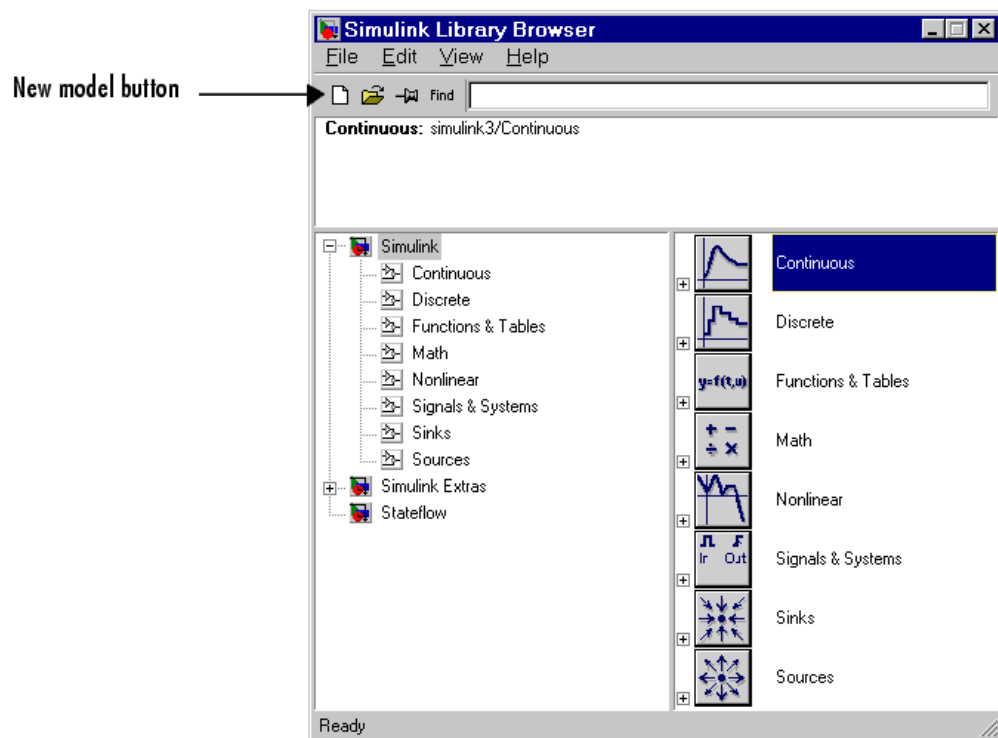
5.4 - Fazendo um modelo simples no Simulink

Este exemplo mostra-lhe como construir um modelo bem simples. . O modelo multiplica a onda do seno por 2 e indica o resultado junto com a onda do seno. O diagrama de blocos é como este:



Para criar o modelo, primeiro entre no Simulink dentro da janela de comando MATLAB. Assim, a Biblioteca de blocos do Simulink aparece.

Para criar um novo modelo novo no Windows, selecione a tecla NEW na biblioteca de blocos. O Simulink abrirá uma janela nova em branco.

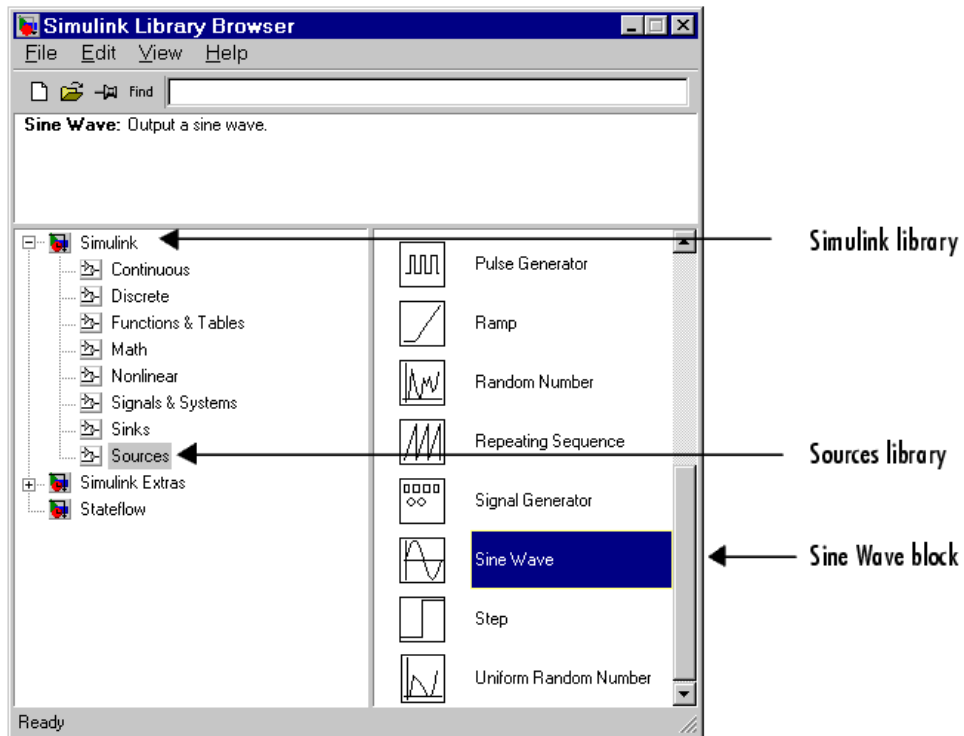


Para criar este modelo, você precisará de copiar para dentro do seu modelo os seguintes blocos das bibliotecas:

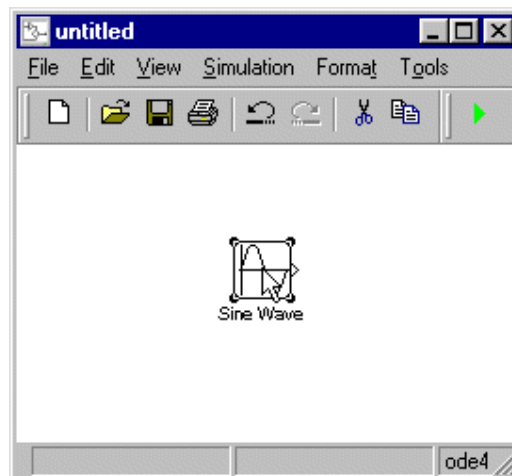
- Sources library (the Sine Wave block)
- Sinks library (the Scope block)
- Math operations library (the Gain block)
- Signal routing library (the Mux block)

Como exemplo apresentamos a busca do bloco seno.

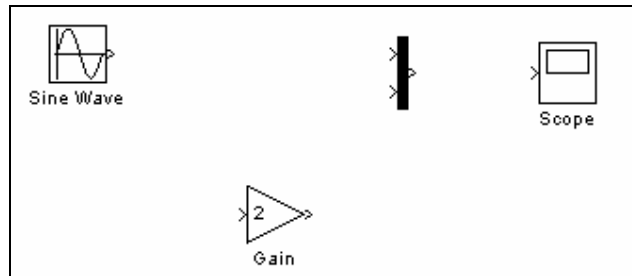
Entra-se em simulink-Sources-Sine Wave



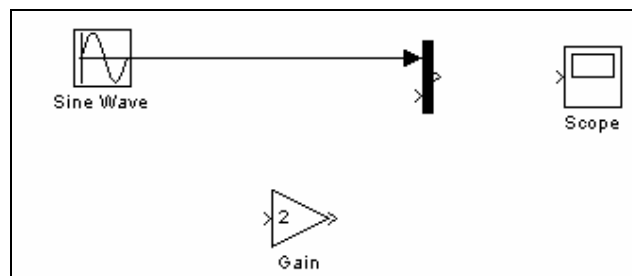
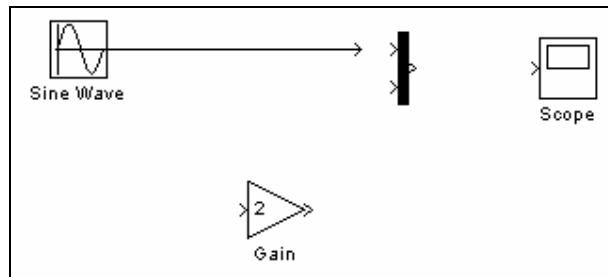
Arrasta-se o bloco da fonte senoidal para dentro da nova area.



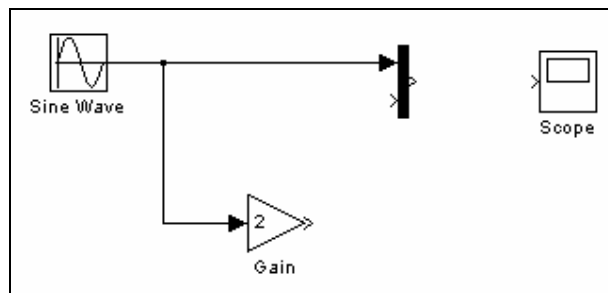
Faz-se o mesmo para todos os outros blocos. Temos agora todos os componentes na nossa área de trabalho.



O próximo passo é conectar todos os blocos. Colocando-se o mouse sobre a saída da fonte senoidal e segurando o botão arrasta-se o fio até o multiplexador.

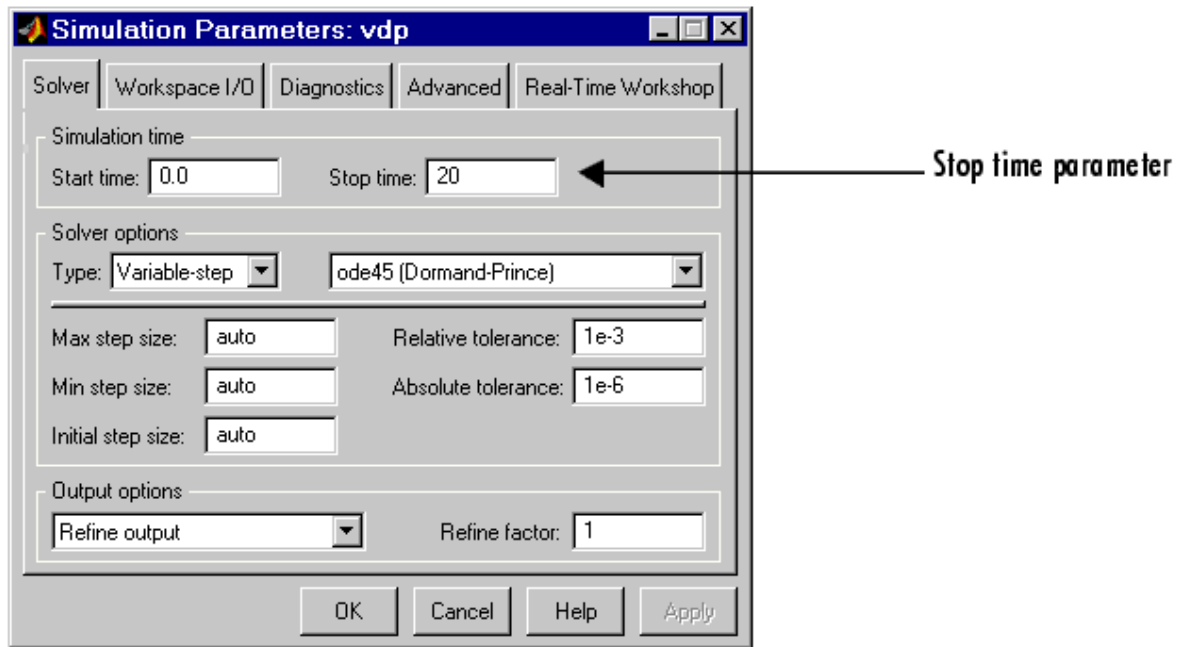


Agora termine de conectar os blocos.



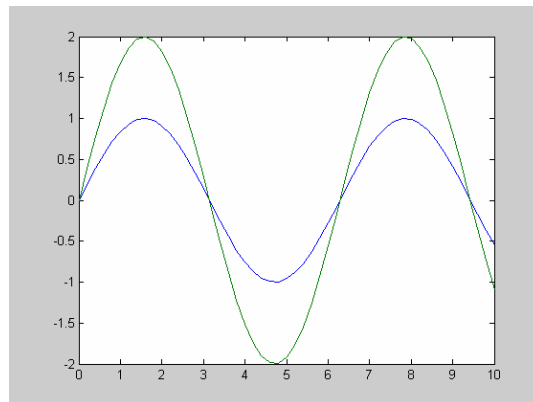
Agora devemos mexer nos parâmetros de simulação (quanto tempo durará a simulação)

Coloque no STOP TIME o valor de 20 segundos



Feche a caixa de dialogo **Simulation Parameters** clicando no botão OK

Escolha Start no menu Simulation e olhe a curva que sairá no bloco de gráfico



Exercício: Dadas 3 fontes senoidais equilibradas de amplitude 1 e defasados de 120° . Desenhar as 3 curvas e a soma delas.

6 - Sistema de Potência

O Power System Blockset (PSB) permite você construir e simular circuitos elétricos contendo elementos lineares e não lineares. Durante as próximas aulas, você será capaz de construir, simular e analisar o circuito da figura abaixo.

Nesta sessão você vai:

- Explorar as bibliotecas “**powerlib**” do PSB
- Aprender a construir um circuito simples com a biblioteca “**powerlib**”
- Interligar blocos do Simulink com o seu circuito.

O primeiro trabalho que desenvolveremos será montagem de um circuito elétrico.

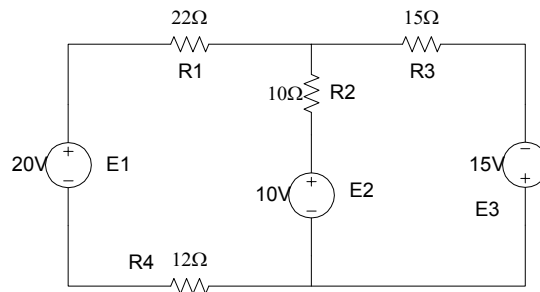


Figura 6.1 – circuito Elétrico

Devemos obter as correntes e tensões em cada resistor do circuito.

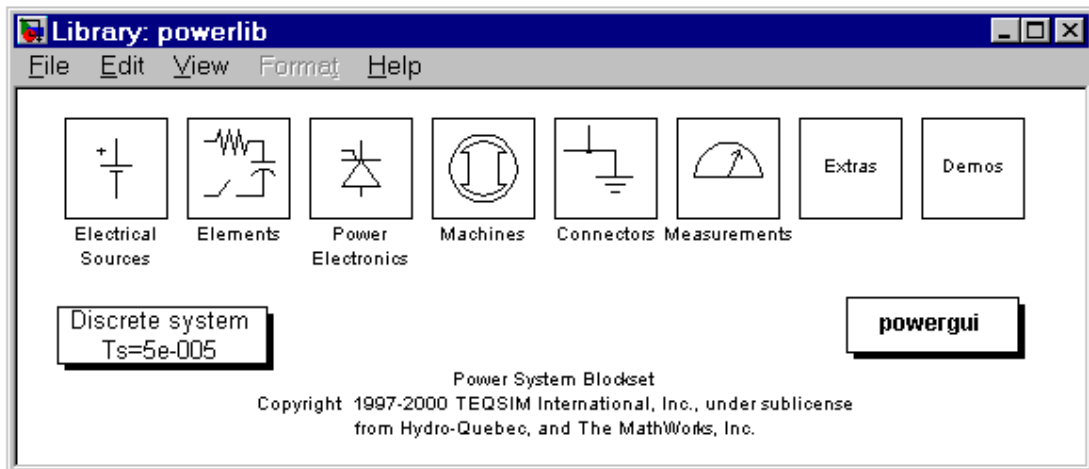
- a) Primeiro devemos resolver por maxwell este circuito.
- b) Depois vamos resolver usando o power system blockset

6.1 - Construindo um circuito elétrico com a biblioteca Powerlib

Abra a biblioteca PSB digitando o seguinte comando no prompt do matlab:

```
powerlib
```

Este comando abre a janela do Simulink mostrando inúmeros blocos diferentes.



Estas bibliotecas podem ser abertas para copiarmos os blocos necessários para a utilização em nosso circuito. Cada componente é representado por ícones que podem ter uma ou mais entradas e saídas correspondendo ao diferentes terminais do componente.

Montando o sistema:

- 1- No menu **File** da janela **powerlib**, abra uma nova janela na qual irá conter seu primeiro circuito e salve-o como: `sep_seunome`.
- 2- Abra a biblioteca de fontes elétricas (**DC Voltage Source**) e copie o bloco de tensão DC para dentro da janela `sep_seunome`

Abra a caixa de diálogo fonte de tensão DC com um duplo clique e entre com a amplitude, de acordo com os valores mostrados na figura 1-1.

Duplicate esta fonte pelo numero de fontes necessários ao circuito. E nomei as fontes segundo a figura

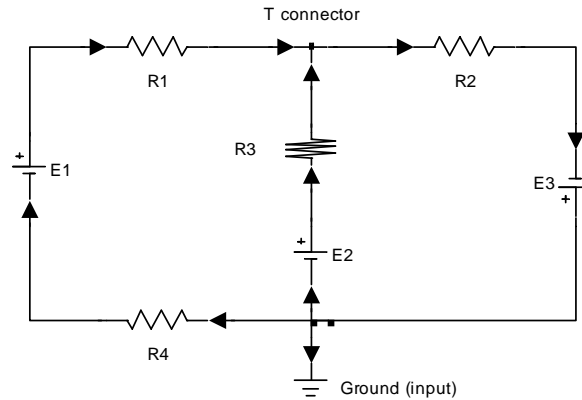
Cole o bloco RLC series Branch, que pode ser encontrado na biblioteca de elementos. Especifique R de acordo com a figura 6.1 e além disto especifique L e C respectivamente em infinito zero e (inf). Quando a caixa de diálogo é fechada, você irá perceber que os componentes L e C desapareceram e somente R está visível.

Duplicate também os resistores. E nomei os resistores segundo a figura.

Copie também o bloco de terra (Ground (Input)).

Vá a biblioteca connectors e copie um bloco T connector.

Abaixo apresentamos como deve estar nosso diagrama de blocos



Precisamos agora de medidores (voltímetros ou amperímetros) para medirmos os valores de tensão e corrente em alguns pontos do circuito.

Devemos colocar um voltímetro para medir a tensão nas resistências R1, R2, R3 e R4. Este bloco pode ser encontrado na biblioteca de medições (measurements library). Copie-o e duplique-o. Conecte sua entrada em cada lado dos resistores.

Para observar a tensão em cada um deles, necessitamos de usar um sistema de plotagem. Podemos usar os dispositivos disponíveis na biblioteca de saída do Simulink (Sinks library).

Abra o sinks library e copie o bloco de scope para a sua janela. Se o scope for conectado diretamente a saída do voltímetro, ele irá apresentar a saída em volts.

6.3 Simulando o circuito

Agora você pode começar a simulação clicando no botão start.

O que encontramos?

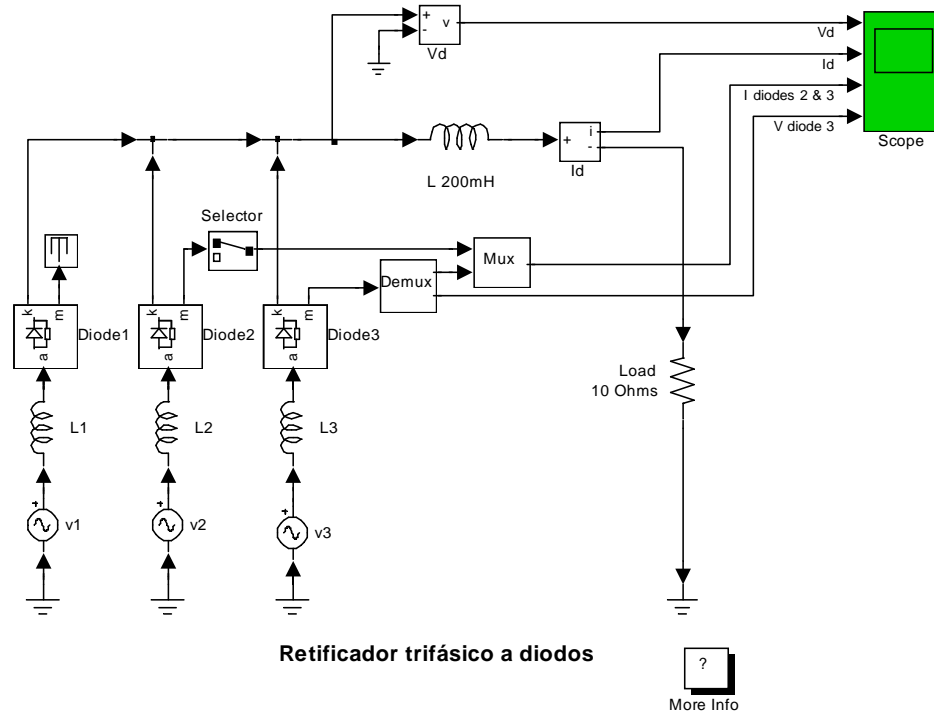
7 - Eletrônica de Potência

Nesta sessão você vai aprender:

- Como usar componentes de eletrônica de potência.

7.1 - Exercício:

Considere o circuito dado na figura abaixo. Ele representa um retificador trifásico senoidal



Uma carga de 10 ohm é alimentada por um retificador trifásico que possui um reator (5mH, 120V rms).

A corrente do retificador é filtrada por uma indutância de 200mH.

Os diodos são conectados em paralelo com os snubber (1000 ohms-0.1uF)

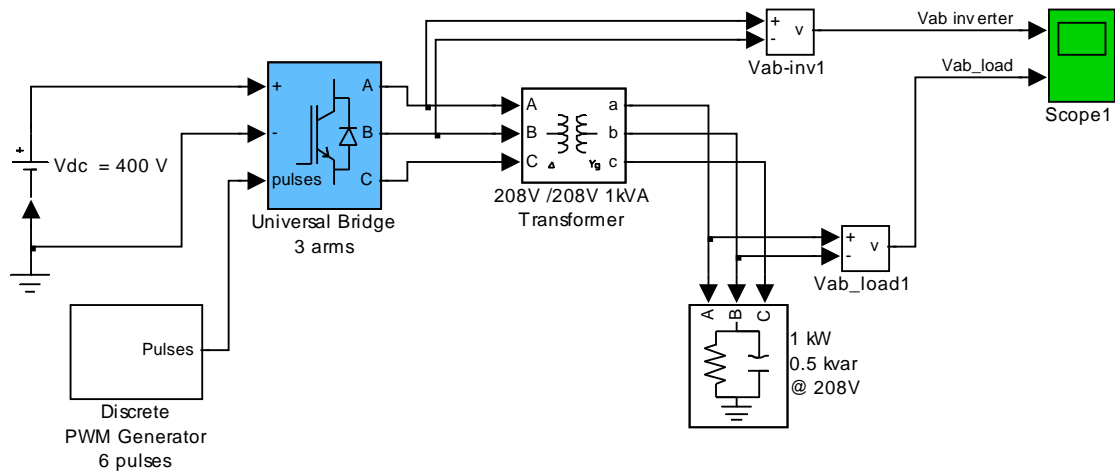
A medição das saídas dos Diodos 2 e 3 é usado para observar a tensão do diodo e corrente do diodo.

Clique no menu de parâmetros de simulação e escolha passo variável e algoritmo ode23tb com parâmetros default

Comece a simulação e observe as formas de onda no bloco de scope. Após um período de transitório a corrente I_d estabiliza em 12,7A.

Apresente as curvas encontradas usando o scope.

Outro exercício:



Three-Phase Two-Level Single Bridge

Vab fundamental inverter:
 $= m/2 * V_{dc} * \sqrt{3}$ Volts peak
 $= 0.85 * 0.866 * 400$
 $= 294 \text{ V}$

Construindo e simulando o sistema:

1. Abrir uma janela nova e salva-la com um nome.
2. Abra a biblioteca de eletrônica de potência e copie o bloco da ponte universal em seu modelo .
3. Abra o menu universal da ponte e ajuste seus parâmetros como segue:

Power Electronic device =IGBT/Diodes; Port configuration= ABC as output terminals; Snubber Rs=10000 ohms, Cs=inf; Ron=1e-4 W; Tail: Tf=1e-6s; Tt=2e-6 s).
4. Abra a biblioteca de transformadores e coloque os parâmetros indicados.
5. Coloque um three phase RLC load e complete com os parâmetros da figura.

Busque o Discrete PWM generator e assumo os seguintes parametos:
Modulation index $m = 0.85$; Frequency of output voltage = 60 Hz;
Phase of output voltage = 0 degrees; carrier frequency $F_s = 1080 \text{ Hz}$; Time step = $10e-6 \text{ s}$

Você está agora pronto para começar a simular o motor. Selecione o algoritmo da integração em passo fixo. Ajuste a tolerância relativa a $1e-4$, a tolerância absoluta e o tamanho de etapa máximo em auto, e o tempo de parada em 0.1 s.

8 - Simulando um acionamento.

Nesta sessão você vai:

- Usar máquinas elétricas e eletrônica de potência para simular um simples acionamento.
- Aprender a usar o bloco de ponte universal.
- Aprender a usar o bloco multímetro.

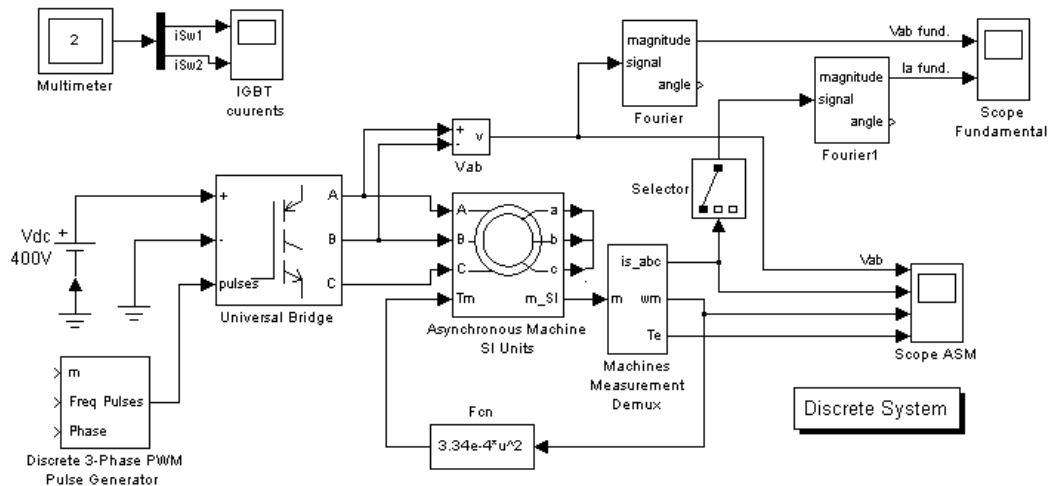
O controle variável da velocidade de máquinas elétricas da C. A. emprega chaves eletrônicas tais como IGBTs, mOSFETs e GTOs. As máquinas assíncronas alimentadas por inversores da modulação de largura de pulso (PWM) estão agora substituindo gradualmente os motores CC e as pontes de tiristor. Com o PWM associado com as técnicas modernas do controle tais como o campo orientado controle ou controle direto de torque, é possível agora obter a mesma flexibilidade no controle da velocidade e de torque do que é obtido com máquinas CC.

Nesta sessão você construirá um acionamento simples da C.C. de malha aberta que controla uma máquina assíncrona.

O circuito do sistema de potência que deve ser simulado é mostrado abaixo. Use blocos das bibliotecas de máquinas e de eletrônica de potência.

A biblioteca das máquinas contém três modelos das máquinas trifásicas mais usadas: máquinas síncronas, assíncronas ou de indução e a síncrona de ímã permanente. Cada máquina pode ser usada na modalidade do gerador ou do motor. Combinado com os elementos lineares e não-lineares tais como transformadores, linhas, cargas, disjuntores, etc., podem ser usados para simular transitórios eletromecânicos em uma rede elétrica. Podem também ser combinados com os dispositivos eletrônicos do poder para simular acionamentos.

A biblioteca de eletrônica de potência contém blocos reservando para simular diodos, tiristores, GTOs, MOSFETs, e IGBTs. Você poderia interconectar diversos blocos junto para construir uma ponte trifásica. Para o exemplo, uma ponte inversora de IGBT requereria seis IGBTs e 6 diodos em antiparalelo. A fim facilitar a execução das pontes, o bloco universal da ponte executa automaticamente estas interconexões para você.



Circuito 5: Controle PWM de um motor de Indução

8.1 -Construindo e simulando um acionamento:

4. Abrir uma janela nova e salva-la como circuit5.
5. Abra a biblioteca de eletrônica de potência e copie o bloco da ponte universal em seu modelo circuit5.
6. Abra o menu universal da ponte e ajuste seus parâmetros como segue:

Power Electronic device =IGBT/Diodes; Port configuration= ABC as output terminals; Snubber Rs=1e5 W Cs=inf; Ron=1e-3 W; Tail: Tf=1e-6s; Tt=1e-6 s).

Observe que o circuito de snubber é parte integrante de diálogo da ponte universal.

6. Abra a biblioteca das máquinas. Copie o bloco da máquina assíncrona com unidade em SI, assim como o bloco de Demux de medida de máquinas em seu modelo circuit5.
7. Abra o menu da máquina assíncrona e olhe seus parâmetros. Os parâmetros são ajustados para 3 HP, 220 V, 60 hertz, dois pares de pólos. Sua velocidade nominal é consequentemente ligeiramente inferior a velocidade síncrona de 1800 RPM ou $\omega = 188.5$ rad/s.
8. Observe que os três terminais a, b e c do rotor estão acessíveis. Durante a operação normal do motor, estes terminais devem ser curto circuitados juntos. Abra a biblioteca dos conectores. Copie o barramento vertical com duas entradas e uma saída em seu modelo circuit5.
9. Abra o menu da barramento e mude o número das entradas para três e o número de saídas para zero. Conecte suas três entradas aos três terminais do rotor como mostrado em figura 1-13.
10. Abra o menu do bloco de Demux de medida das máquinas. Quando este bloco é conectado na saída de medida do motor assíncrono, é permitido que você

tenha sinais internos específicos do motor assíncrono. Deselecione todos os sinais exceto os seguintes três sinais: i_{s_abc} (três correntes do estator), ω_m (velocidade do rotor) e T_e (torque eletromagnético).

11. Você executará agora a característica torque-velocidade carga do motor. Vamos supor uma característica quadrática da torque-velocidade (tipo carga do ventilador ou da bomba). O torque T é então proporcional ao quadrado da velocidade:

$$T = k \times \omega^2$$

O torque nominal deste motor é:

$$T_n = \frac{3 \times 746}{188.5} = 11.87 \text{ Nm}$$

Portanto, a constante K deve ser:

$$k = \frac{T_n}{\omega_s^2} = \frac{11.87}{188.5^2} = 3.34 \times 10^{-4}$$

Abra a biblioteca funções & tabelas do Simulink e copie o bloco de Fcn em seu modelo circuit5. Abra o menu do bloco e incorpore a expressão do torque como a função da velocidade:

$$3.34e-4 * u^2$$

12. Conecte a entrada do bloco Fcn de Simulink à saída da velocidade ω_m que sai do bloco Demux da máquina e a sua saída à entrada do torque do motor nomeado de T_m .
13. Abra a biblioteca de medição e copie um bloco da medida da tensão em seu modelo circuit5. Mude o nome de bloco para V_{ab} .
14. Usando os blocos de terra da biblioteca de conectores, termine de conectar os elementos de potência e as interconexões do sensor da tensão como mostrado na figura.
15. A fim controlar a ponte inversora, você necessita de um gerador de pulso. Tal gerador está disponível na biblioteca extras do powerlib. Abra a biblioteca dos blocos de Extras/Control e copie o bloco “ Discrete 3-Phase PWM Pulse Generator” em seu modelo circuit5. Conecte sua saída dos pulsos à entrada dos pulsos do bloco ponte universal.
16. Abra o menu Discrete 3-Phase PWM Pulse Generator e ajuste os parâmetros como segue:

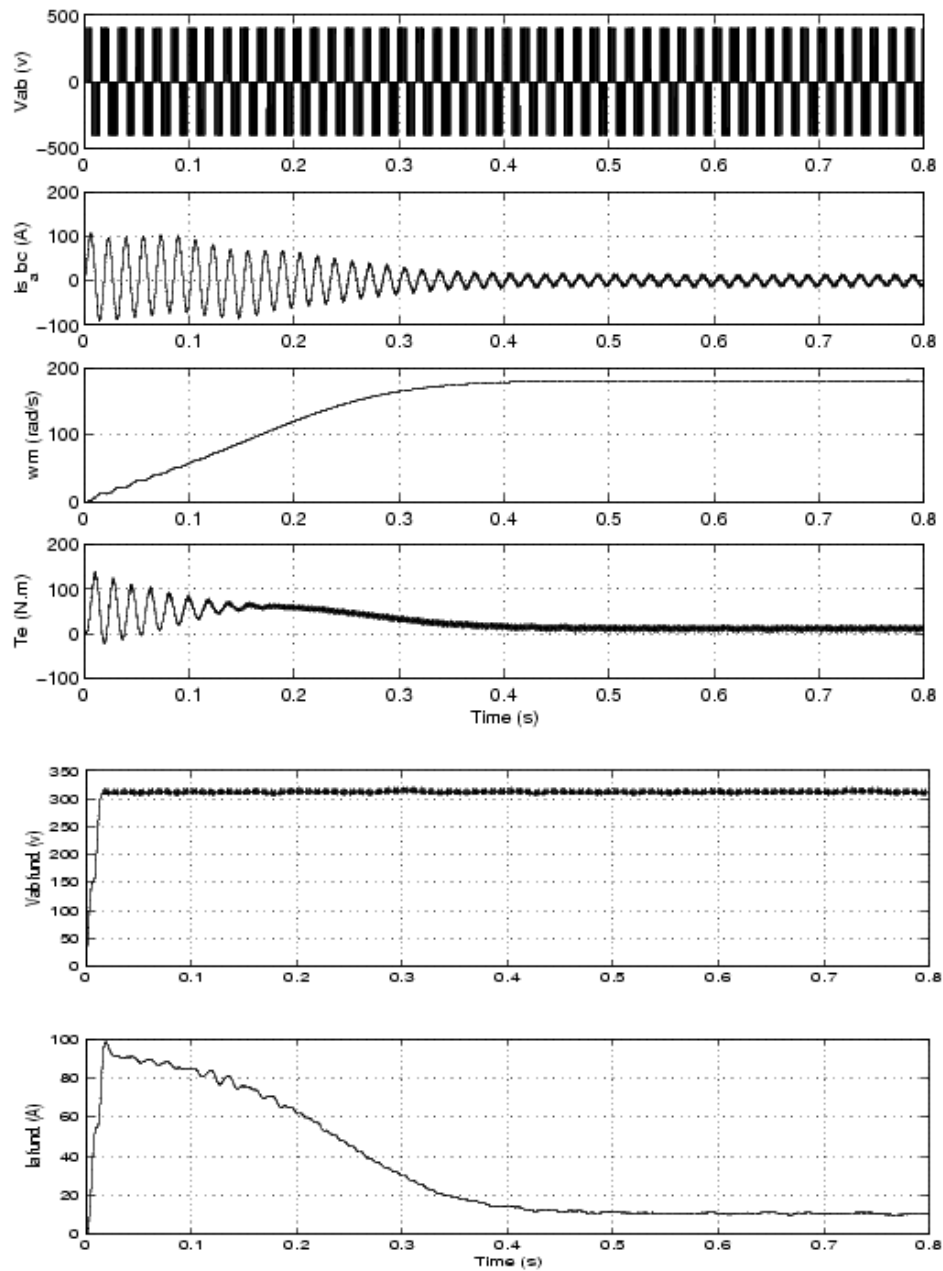
Modulation index $m = 0.90$; Frequency of output voltage = 60 Hz;
Phase of output voltage = 0 degrees; Switching frequency $F_s = 1080$ Hz; Time step = $10e-6$ s

17. Finalmente, adicione scopes a seu modelo. Copie um bloco de scope em seu circuito. Este scope será usado para indicar a tensão instantânea do motor, correntes, velocidade e torque
18. nas propriedades do scope, ajuste os seguintes parâmetros:

Number of axes=4; Time range =0.05 s; Tick labels: bottom axis only.

19. Conecte as quatro entradas conforme a figura

Você está agora pronto para começar a simular o motor. Selecione o algoritmo da integração de ode23tb. Ajuste a tolerância relativa a $1e-4$, a tolerância absoluta e o tamanho de etapa máximo em auto, e o tempo de parada em 1 s.



Acionamento PWM - Simulation Results for Motor Starting at Full Voltage