Universidade Federal Fluminense – UFF
Escola de Engenharia – TCE
Curso de Engenharia de Telecomunicações – TGT
Programa de Educação Tutorial – PET

Dicas PET-Tele

Manipulação de dados de som no Matlab/Octave

(Versão: A2013M06D20)

Autores: Carina Ribeiro Barbio Corrêa

Tutor: Alexandre Santos de la Vega

Niterói – RJ
Junho / 2013

1 Introdução

Este tutorial tem como objetivo apresentar as funções necessárias para capturar e reproduzir sons no Matlab/Octave.

O Matlab (abreviatura de MATrix LABoratory - Laboratório de Matrizes) é um aplicativo de simulação matemática comercial que realiza operações matriciais, constrói gráficos em duas ou três dimensões, auxilia no estudo e na implementação de técnicas de processamento de sinais, além de manipular outras funções especializadas.

O Octave é um software livre, equivalente ao Matlab, desenvolvido por Eaton (1997) e por vários outros colaboradores. Foi originalmente criado como um livro texto de graduação, no projeto de um reator químico que estava sendo escrito por James B. Rawlings, da Universidade Wisconsin-Madison, e John G. Ekerdt da Universidade do Texas. É um programa de código aberta. Assim, muitas pessoas contribuem com centenas de comandos que são adicionadas às versões em fase teste.

As funções apresentadas neste tutorial estão separadas em duas seções, pois nem todas elas funcionam em ambos os aplicativos.

Todas as funções apresentadas são encontradas no Apêndice A.

2 Captura e reprodução de sons

O Matlabe/Octave dispõe de algumas funções para tratamento de áudio. Os dados de áudio manipulados pelo aplicativo são amostras do sinal original de áudio, organizadas em uma sequência indexada, os quais podem ser caracterizadas por três parâmetros: a frequência de amostragem do sinal de áudio, o número de *bits* utilizados na representação de cada amostra e o número de canais usados na gravação (1 para mono e 2 para estéreo).

As amostras do sinal de áudio são armazenados em vetores.

Supõe-se que os arquivos que usam codificação linear têm extensão *lin* ou *raw*, enquanto os arquivos que armazenam dados em *mu-law* têm um nome com extensão *au*, *mu* ou *snd*.

3 Captura e reprodução de sons no Matlab

O Matlab permite o uso de som em plataforma PC e em qualquer plataforma UNIX que tenha um dispositivo '*/dev/audio*'.

Existem funções para ler arquivos de som para um vetor, gravar um vetor para um arquivo e para reproduzir sons a partir de arquivos.

Dois formatos de som comerciais são aceitos no Matlab, o formato de audio da NeXT/Sun (file.au) e o formato da Microsoft (file.wav). Esses formatos podem ser gravados e lidos. O formato NeXT/Sun Audio aceita dados de múltiplos canais sendo eles no formato 'mu-law' de 8 *bits* e linear de 8 ou 16 *bits*.

3.1 Função *sound*

A função *sound* permite reproduzir um som armazenado em um vetor. A sua sintaxe é representada abaixo, onde x é o vetor linha ou coluna, *fa* é a frequência de amostragem e b é o número de *bits*/segundo.

```
sound(x,fa,b)
```

Essa função envia o vetor x para o altofalante, com frequência de amostragem *fa*. Os valores em x que estão fora do intervalo $[-1, 1]$ são comprimidos. Se *fa* é omitida, a frequência padrão de 8.192 Hz é usada. O Matlab executará o som usando b, *bits*/segundo, se possível. A maioria das plataformas aceita $b = 8$ ou $b = 16$. Se b for omitido, será utilizado $b = 16$.

No exemplo abaixo, será gerada uma senóide com frequência de 1000 Hz e com a duração de 1 segundo. Para gerar este sinal, deve ser utilizada a função seno do Matlab. Quando o argumento desta função varia entre 0 e 2π, a função perfaz um período. Se quisermos 1000 períodos, o argumento terá de variar entre 0 e 2000π, durante o segundo de duração do sinal. Para gerar os instantes de tempo, será escolhida uma frequência de amostragem de 8 kHz, o que dará 8000 instantes de amostragem no intervalo $[0 \cdots 1]$. O código Matlab para gerar o sinal é

```
>> Fs = 8000;
>> t = 0:(1/Fs):1;
>> x = sin(2*pi*1000*t);
```

Para ouvir este sinal, deve-se utilizar a função *sound* com a frequência de amostragem correta, da seguinte forma:

```
>> sound(x,Fs)
```

3.2 Função *soundsc*

A função *soundsc* é equivalente à função *sound*, porém os valores de x são escalados para o intervalo $[-1.1]$, em vez de cortados. Há um argumento adicional que permite relacionar um intervalo de valores em x a todo intervalo sonoro. A sintaxe dessa função é

```
soundsc(x,fa,b, [smin smax])
```

Caso seja omitido, o intervalo padrão utilizado será [min(x) max(x)].
O exemplo utilizado é o mesmo da função *sound*, cujo código é

```
>> Fs = 8000;
>> t = 0:(1/Fs):1;
>> x = sin(2*pi*1000*t);
>> soundsc(x,Fs)
```

3.3 Função *wavrecord*

A função *wavrecord* grava um som usando o dispositivo de entrada do *Windows*. A sua sintaxe pode ser de três formas:

```
x = wavrecord(n,fa)
x = wavrecord(n,fa,ch)
x = wavrecord(n,fa,ch,'dtype')
```

A sintaxe x = *wavrecord(n,fa)* grava n amostras de um sinal de áudio, com uma frequência de amostragem *fa*. O valor padrão para *fa* é 11025 Hz.

A sintaxe x = *wavrecord(n,fa,ch)* usa o número *ch* de canais de entrada do dispositivo de áudio. O valor de *ch* pode ser 1 ou 2, para mono ou estéreo, respectivamente. O valor padrão para *ch* é 1.

A sintaxe *x = wavrecord(n,fa,ch,'dtype')* usa o tipo de dado especificado pela *string 'dtype'* para gravar o som. A tabela a seguir lista os valores da *string* junto com o número de *bits* por amostra correspondentes.

dtype	*bits*/amostra
'double'	16
'single'	16
'int16'	16
'uint8'	8

A função *wavrecord* é usada apenas para o sistema operacional *Microsoft Windows 32-bits*. Para gravar arquivos de áudio em outras plataformas, deve-se usar a função *audiorecorder*.

O exemplo utilizado grava cinco segundos de uma amostra de áudio de 16 *bits*, com uma frequência de 11025 Hz. Enquanto o comando *wavrecord* estiver executando, deve-se falar no microfone para gravar o áudio.

```
>> Fs = 11025;
>> y = wavrecord(5*Fs,Fs,'int16')
>> wavplay(y,Fs);
```

3.4 Função *audiorecorder*

A função *audiorecorder* é similar à função *wavrecord*. Ela grava um som usando o dispositivo de entrada do *Windows*. Sua sintaxe é

```
y = audiorecorder
y = audiorecorder(fa,nbits,ncanais)
```

O parâmetro *nbits* é o número de *bits* por amostra, cujo valor pode ser 8, 16 ou 24, sendo 8 o valor padrão. O parâmetro *ncanais* é o número de canais, que pode ser 1 (mono) ou 2 (estéreo).

3.5 Função *wavplay*

A função *wavplay* executa um arquivo de som no formato 'file.wav'. A sua sintaxe é

```
wavplay(y,fa)
```

Para executar esta função, deve-se, primeiramente, gravar um som no Matlab e, depois, executá-la.

```
>> Fs = 11025;
>> y = wavrecord(5*Fs,Fs,'int16')
>> wavplay(y,Fs);
```

3.6 Função *wavread*

A função *wavread* permite inserir, no ambiente Matlab, o conteúdo de um arquivo de som. A sua sintaxe é

```
[x,fa,Formato] = wavread(wavefile)
```

Esta função carrega um arquivo '*wavefile*', com contúdo no formato 'file.wav', retornando os dados amostrados na variável x, a taxa de amostragem na variável fa e a informação de formato do arquivo 'file.wav' na variável *Formato*. A variável *Formato* é um vetor de 6 elementos, dado por

1. Formato dos dados (sempre PCM)
2. Número de canais
3. Taxa de amostragem (fa)
4. *Bytes* por segundo, (amostragem média)
5. Alinhamento em bloco dos dados
6. *Bits* por amostra

Primeiramente deve-se criar um arquivo 'file.wav' no diretório corrente. Com o arquivo criado, basta executar o seguinte comando:

```
>>[x, fa] = wavread('nome do arquivo.wav');
```

Para ouvir o som do vetor x, basta executar o comando:

```
>> sound(x, Fs);
```

Para pausar antes do som tocar novamente, deve-se executar os seguintes comandos:

```
>> duration = numel(y) / Fs;
>> pause(duration + 2);
```

Para ler apenas os 2 primeiros segundos do som armazenado, deve ser usado o seguinte código:

```
>> nsamples = 2 * Fs;
>> [y2, Fs] = wavread(hfile, nsamples);
>> sound(y2, Fs);
>> pause(4)
```

3.7 Função *wavwrite*

A função *wavwrite* grava um arquivo de som no formato 'file.wav'. A sua sintaxe é

```
wavwrite(x,fa,'nome do arquivo')
```

Primeiramente um som deve ser criado no Matlab, para depois a função ser usada.

```
>> Fs = 8000;
>> t = 0:(1/Fs):1;
>> x = sin(2*pi*1000*t);
>> sound(x,Fs)
>> wavwrite(x,8000,'teste')
```

3.8 Função *auread*

A função *auread* lê um arquivo de som no formato NEXT/SUN. Sua sintaxe é a mesma da função *wavread*.

```
[x,fa,Formato] = auread('nome do arquivo.au')
```

3.9 Função *auwrite*

A função *auwrite* grava um arquivo de som no formato NEXT/SUN. Sua sintaxe é igual à da função *wavwrite*.

```
auwrite(x,fa,'nome do arquivo')
```

3.10 Função *lin2mu*

A função *lin2mu* converte sinais de áudio com formato linear em sinais codificados no formato *mu-law*. Tem como sintaxe:

```
lin2mu(x,n)
```

Se n não for especificado, ele assume um dos valores padrões 0, 8 ou 16.

```
>> Fs = 8000;
>> t = 0:(1/Fs):1;
>> x = sin(2*pi*1000*t);
>> sound(x,Fs)
>> lin2mu(x)
```

3.11 Função *mu2lin*

A função *mu2lin* converte sinais de áudio codificados no formato *mu-law* em sinais com formato linear. Sua sintaxe é

```
mu2lin(x,n)
```

Se n não for especificado, ele assume o valor padrão 0.

4 Captura e reprodução de sons no Octave

Uma vez que Octave é um aplicativo de uso livre, equivalente ao aplicativo comercial Matlab, ele apresenta funções similares para manipulação de áudio. As funções definidas no Octave são apresentadas a seguir.

4.1 Função *sound*

A função *sound* permite reproduzir um som armazenado em um vetor. A sintaxe da função é a mesma usada no Matlab:

```
sound(x,fa,b)
```

Esta função foi testada na distribuição Ubuntu, versão 12.04, do Linux, e foi encontrado o seguinte erro:

```
dd: opening ‘/dev/dsp’: No such file or directory
    cmd = dd if=/dev/dsp of="/tmp/oct-3iR8tF" bs=8000 count=2
error: fread: invalid stream number = -1
error: called from
error: /usr/share/octave/3.2.4/m/audio/record.m at line 60, column 5
octave3.2:1>
```

Para solucionar esse problema, foi criada uma nova função, que deve ser inserida no diretório *usr/share/octave/3.2.4/m/audio/* ou no diretório corrente. O código dessa nova função é

```
function sound (name)
  if (nargin == 1 && isvector(name) && ! ischar(name))
    ## play a vector
    [nr, nc] = size(name);
    if (nc != 1)
      if (nr == 1)
        name = name';
        nr = nc;
      else
        error("sound: X must be a vector");
      endif
    endif
    X = name + 127;
    unwind_protect
      file = tmpnam();
      wavwrite(name,8000,8,[file,".wav"]);
      system(sprintf ("aplay -t%s -r%d %s","wav",8000,[file,".wav"]));
      unwind_protect_cleanup
      unlink(file);
    end_unwind_protect
  else
    print_usage();
  endif
end function
```

4.2 Função *record*

A função *record* grava uma quantidade de segundos em um vetor x usando o dispositivo de entrada do *Windows*. A sintaxe da função é

```
record(sec,bps)
```

Esta função foi testada na distribuição Ubuntu, versão 12.04, do Linux, e foi encontrado o seguinte erro:

```
dd: opening ‘/dev/dsp’: No such file or directory
    cmd = dd if=/dev/dsp of="/tmp/oct-3iR8tF" bs=8000 count=2
error: fread: invalid stream number = -1
error: called from
error: /usr/share/octave/3.2.4/m/audio/record.m at line 60, column 5
octave3.2:1>
```

Para solucionar esse problema, foi criada uma nova função, que deve ser inserida no diretório *usr/share/octave/3.2.4/m/audio/* ou no diretório corrente. O código dessa nova função é

```
function X = rec(sec, sampling_rate)
  if (nargin == 1)
    sampling_rate = 8000;
  elseif (nargin != 2)
    print_usage();
  endif

  file = tmpnam();
  file= [file,".wav"];

  input("Please hit ENTER and speak afterwards!!!!!\n",1);

  cmd = sprintf("arecord --duration=%d --rate=%d %s",sec,sampling_rate,file);
  system (cmd);

  X = wavread(file);
end
```

4.3 Função *wavread*

Como citado anteriormente, a função *wavread* permite inserir no ambiente Matlab o conteúdo de um arquivo de som. A sua sintaxe é

```
[x,fa,Formato] = wavread(wavefile)
```

4.4 Função *wavwrite*

Como citado anteriormente, a função *wavwrite* grava um arquivo de som no formato .wav. A sua sintaxe é

```
wavwrite(x,fa,’nome do arquivo’)
```

4.5 Função *loadaudio*

A função *loadaudio* carrega um arquivo de áudio para um vetor. Sua sintaxe é

```
loadaudio("nomedoarquivo","extensao",bps)
```

4.6 Função *saveaudio*

A função *saveaudio* salva um vetor de áudio em um arquivo *nome.extensao*. A sintaxe é

```
saveaudio("nome",x,"extensao", bps)
```

Os padrões para os parâmetros *extensao* e *bps*, são *lin* e 8, respectivamente.

4.7 Função *setaudio*

A função *setaudio* executa o comando *mixer* do sistema operacional em questão. A sintaxe é

```
setaudio(w_tipo,valor)
```

4.8 Função *lin2mu*

Como citado anteriormente, a função *lin2mu* converte sinais de áudio com formato linear em sinais codificados no formato *mu-law*. Ela tem como sintaxe

```
lin2mu(x,n)
```

4.9 Função *mu2lin*

Como citado anteriormente, a função *mu2lin* converte os sinais de áudio codificados no formato *mu-law* em sinais com formato linear. Sua sintaxe é

```
mu2lin(x,n)
```

Apêndice A
Resumo das funções

A1 - Resumo das funções do Matlab

Função	Descrição	Sintaxe
sound	Reproduz um som armazenado em um vetor	sound(x,fa,b)
soundsc	Equivalente a função *sound*, porém os valores de x são escalados para o intervalo $[-1.1]$, em vez de cortados	soundsc(x,8000)
wavrecord	Grava um som usando o dispositivo de entrada do Windows	x=wavrecord(n,fa)
audiorecorder	Similar à função *wavrecord*, ela grava um som usando o dispositivo de entrada do *Windows*	y=audiorecorder(fa,nbits,ncanais)
wavplay	Executa um arquivo de som no formato 'file.wav'	wavplay(y,fa)
wavread	Carrega para o ambiente Matlab um arquivo de som	[x,fa,Formato]=wavread(wavefile)
wavwrite	Grava um arquivo de som no formato 'file.wav'	wavwrite(x,fa,'nome do arquivo')
auread	Lê um arquivo de som no formato NeXT/Sun	[x,fa,Formato] = auread(wavefile)
auwrite	Grava um arquivo de som no formato NeXT/Sun	auwrite(x,fa,'nome do arquivo')
lin2mu	Converte sinais de áudio lineares em sinais codificados no formato *mu-law*	lin2mu(x,n)
mu2lin	Converte os sinais de áudio codificados no formato *mu-law* em sinais lineares	mu2lin(x,n)

A2 - Resumo das funções do Octave

Função	Descrição	Sintaxe
sound	Reproduz um som armazenado em um vetor	sound(x,fa,b)
record	Grava uma quantidade de segundos em um vetor x usando o dispositivo de entrada do *Windows*	record(sec,bps)
wavread	Permite inserir no ambiente Matlab um arquivo de som	[x,fa,Formato]=wavread(wavefile)
wavwrite	Grava um arquivo de som no formato 'file.wav'	wavwrite(x,fa,'nome do arquivo')
loadaudio	Carrega um arquivo de áudio para um vetor	loadaudio("arquivo","extensao",bps)
saveaudio	Salva um vetor de áudio em um arquivo	saveaudio("arquivo",x,"extensao", bps)
setaudio	Executa o comando *mixer*	setaudio(w_tipo,valor)
lin2mu	Converte sinais de áudio lineares em sinais codificados no formato *mu-law*	lin2mu(x,n)
mu2lin	Converte os sinais de áudio codificados no formato *mu-law* em sinais lineares	mu2lin(x,n)

Referências

[1] http://zone.ni.com/reference/en-XX/help/371361D-01/lvtextmath/msfunc_wavrecord/} Acessado em: 20/02/2013.

[2] http://www.mathworks.com/help/matlab/ref/wavrecord.html Acessado em: 20/02/2013.

[3] http://www.gnu.org/software/octave/doc/interpreter/Audio-Processing.html Acessado em: 20/02/2013.

[4] LITTLEFIELD, Bruce; HANSELMAN, Duane. *Matlab6 - Curso Completo*, Pearson-Prentice Hall.