

Fazendo Músicas Simples no MATLAB

Uma nota musical consiste e pode ser descrita em termos de sua frequência (o quanto é aguda), sua amplitude (volume) e sua forma (características). Para um modelo simples de som nós iremos começar com uma simples onda senóide. Uma simples nota, pode ser representada pela função:

$$A \sin(2\pi ft)$$

onde: f é a frequência (ciclos por segundo)

t é o tempo decorrido (em segundos)

A é a amplitude (de dimensões pouco importantes para nosso uso)

Criando um tom simples com o MATLAB: O conjunto de comandos abaixo vão criar uma onda senóide para a frequência de 440Hz (meio A ou A4), com uma frequência de amostragem de 8000 e duração de $\frac{1}{2}$ de segundo.

```
>> % primeiramente vamos definir variáveis de entrada para a frequência de amostragem
(fa) e duração (dur)
>> fa = 8000
>> dur = 0.5
>> % definimos o vetor tempo o qual é 'dur' segundos e
>> % possui intervalos para todo 1/fa seg.
>> t = 0:1/fa:dur;
>> % então criamos uma série de amplitudes para uma onda senóide.
>> a=sin(2*pi*440*t);
>> % finalmente podemos mexer no tom que criamos.
>> soundsc(a,fa)
```

Você também pode usar o comando *wavwrite* para salvar o arquivo como um arquivo windows wav. A forma para este comando seria:

wavwrite(song, 8000, "nome_do_arquivo.wav")

Notas e frequências para uma escala igualmente moderada: (adaptado de Bryan H. Suits, "Frequencies for equal-tempered scale", n.d. <http://www.phy.mtu.edu/~suits/notefreqs.html>, acessado em Agosto de 2004).

Octave	0	1	2	3	4	5	6	7
C	16.35	32.7	65.41	130.81	261.63	523.25	1046.5	2093
C [#] /D ^b	17.32	34.65	69.3	138.59	277.18	554.37	1108.73	2217.46
D	18.35	36.71	73.42	146.83	293.66	587.33	1174.66	2349.32
D [#] /E ^b	19.45	38.89	77.78	155.56	311.13	622.25	1244.51	2489.02
E	20.6	41.2	82.41	164.81	329.63	659.26	1318.51	2637.02
F	21.83	43.65	87.31	174.61	349.23	698.46	1396.91	2793.83
F [#] /G ^b	23.12	46.25	92.5	185	369.99	739.99	1479.98	2959.96
G	24.5	49	98	196	392	783.99	1567.98	3135.96
G [#] /A ^b	25.96	51.91	103.83	207.65	415.3	830.61	1661.22	3322.44
A	27.5	55	110	220	440	880	1760	3520

ENGCOMP - UFPA – Aldebaro – 2011 - Baseado em ENGR121, SS Moor, obtido na Web:

Tradução: Alexandre Van der ven.

<http://users.rowan.edu/~shreek/networks1/music.html>

A [#] /B ^b	29.14	58.27	116.54	233.08	466.16	932.33	1864.66	3729.31
B	30.87	61.74	123.47	246.94	493.88	987.77	1975.53	3951.07

Script do arquivo de uma música simples: Note a natureza dos comentários no programa.

```
% Programa musical.m
% Preparado por S. Scott Moor, IPFW August 2004
% Baseado em sugestões de Shreekanth Mandayam,
% Departamento de Engenharias Elétrica e da Computação, Rowan University
% veja http://users.rowan.edu/~shreek/networks1/music.html
%
% Esse programa cria ondas senóides para séries de notas padrões.
% Cada nota é configurada para ter uma duração de 0.5 segundos
% e taxa de amostragem de 8000 Hz. As notas são juntadas em uma
% música que então é reproduzido pelo computador.

% variáveis usadas:
%   fa      = frequência de amostragem (amostragens/seg)
%   x       = vetor tempo (seg.)
%   a, b, cs, d, e, & fs = a amplitude de uma série de notas usadas na música.
%   linha1, linha2, linha3 = a série de amplitudes usadas para cada linha da música.
%   song    = the amplitude series for the entire song.

% configuração das séries de tempo.

fa= 8000;
x = [0:1/sf:0.5];

% define each note

a=sen(2*pi*440*x);
b=sen(2*pi*493.88*x);
cs=sen(2*pi*554.37*x);
d=sen(2*pi*587.33*x);
e=sen(2*pi*659.26*x);
fs=sen(2*pi*739.99*x);

% montando as notas em uma música.

linha1 = [a,a,e,e,fs,fs,e,e];
linha2 = [d,d,cs,cs,b,b,a,a];
linha3 = [e,e,d,d,cs,cs,b,b];
musica = [linha1,linha2,linha3,linha3,linha1,linha2];

% agora tocando a música

sound(musica, fa)
```

Comece com a palavra 'programa' e o nome do arquivo usado. Este é seguido pelos nomes dos autores e data.

Descreve a proposta básica do programa e o que ele faz.

Comentários antes da primeira linha em branco vão ser mostrados na janela de comando se você digitar >> **help musica1**

Define as variáveis usadas dentro do escript. As vezes essa lista é dividiída em entrada, saída e variáveis intermediárias.

Através do programa é incluído comentários que descrevem a função por cada parte do programa.