Aalto University
School of Electrical
Engineering

Using MATLAB in Audio and Acoustics

Javier Gómez Bolaños

Department of Signal Processing and Acoustics
Aalto University, School of Electrical Engineering
javier.gomez.bolanos@aalto.fi

January 29, 2016

Contents

Using Matlab

- Mathematical simulator (Algorithms, plots, programs, measurements, ...)
- It works better for matrix operations
- Vectors and Matrices can represent whatever we want (signals, scores, frequency/time axes, ...)
- Continuous functions and polynomials are represented as vectors in Matlab

$$\mathbf{a} = \{a(1)\, a(2) \cdots a(n) \cdots a(N)\};$$

Using Matlab

- Vector/Matrix conjugate transpose (Hermitian)
  $\mathbf{b} = \mathbf{a}'; \quad \mathbf{a}^{M,N}, \mathbf{b}^{N,M}$
- Vector/Matrix transpose
  $\mathbf{b} = \mathbf{a}.'; \quad \mathbf{a}^{M,N}, \mathbf{b}^{N,M}$
- Vector/Matrix product
  $\mathbf{c} = \mathbf{a} * \mathbf{b}; \quad \mathbf{a}^{M,N}, \mathbf{b}^{N,M}, \mathbf{c}^{M,M}$
- Vector/Matrix element-wise product
  $\mathbf{c} = \mathbf{a}. * \mathbf{b}; \quad \mathbf{c}(n) = \mathbf{a}(n) * \mathbf{b}(n) \quad \mathbf{a}^{M,N}, \mathbf{b}^{M,N}, \mathbf{c}^{M,N}$
- Vector/Matrix division
  $\mathbf{c} = \mathbf{a}/\mathbf{b}; \quad \mathbf{c} = \mathbf{a} * \mathbf{b}^{-1}$
  $\mathbf{c} = \mathbf{a}\backslash\mathbf{b}; \quad \mathbf{c} = \mathbf{a}^{-1} * \mathbf{b}$
- Vector/Matrix element-wise division
  $\mathbf{c} = \mathbf{a}./\mathbf{b}; \quad \mathbf{c}(n) = \mathbf{a}(n)/\mathbf{b}(n)$

Parseval's theorem

- General form:

$$\int_{-\inf}^{\inf} |x(t)|^2 \, dt = \int_{-\inf}^{\inf} |X(f)|^2 \, df$$

- For Discrete Fourier Transform: (if N=Nfft)

```
X = fft(x);
```

$$\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2$$

- In a more general way: (N<=Nfft $\equiv$ zero-padding)

```
X = fft(x,Nfft);
```

$$\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{Nfft} \sum_{k=0}^{Nfft-1} |X(k)|^2$$

Filtering

$$H(z) = \frac{b_0 + b_1(z^{-1}) + \cdots + b_n(z^{-N})}{1 + a_1(z^{-1}) + \cdots + a_n(z^{-L})}$$

$$H(z) = \frac{B(z)}{A(z)}$$

$$y(n) = \sum_{i=0}^{N} b_i x(n-i) - \sum_{j=1}^{L} a_j x(n-j)$$

Filtering: Example

$$H(z) = \frac{-0.5 + z^{-1}}{1 + 0.5z^{-1}}$$

```
b = [-0.5 1];
a = [1 0.5];
y =[0 b(1).*x];
for i=2:size(x,1)
y(i) = sum(b(end:-1:1).*x(i-1:i))-a(2)*y(i-1);
end;
```

or

```
y = filter(b,a,x);
```

To design filters check `fdesign()` command in Matlab or whatever other implemented method (`fir1()`, `firpm()`,...).

Processing in frequency domain

In the frequency domain, calculations are the same for each frequency bin (element-wise)

$$U(f) = \frac{P(f)}{Z(f)} \quad ; \quad Z(f) = Z_o \frac{1}{\tanh(\gamma L)} \quad ; \quad \gamma = j\frac{\omega}{c} = j\frac{2\pi f}{c}$$

```
% Particle velocity at the entrance of a lossless
% closed-end tube of length L
P = fft(p,Nfft);
w = 2*pi.*(0:Nfft/2+1)/Nfft*fs;
c = 343;
L = 0.010;
Z0 = 413;
gamma = 1i.*w./c;
Zaux = Z0./tanh(gamma);
Z = [Zaux.'; Zaux(end-1:-1:2)'];
U = P./Z;
```