



BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH

Khoa: Công Nghệ Thông Tin



LAB REPORT 09

Student's ID :
Student's name : Hồ Phúc Lâm
Subject : PTHTDPT
Instructor : Nguyễn Thành Thái
Faculty : Công Nghệ Thông Tin
Completed Date : 30/10/2024

[illegible]

LAB 06 CM3106: MULTIMEDIA COURSEWORK HANDOUT TIME-FREQUENCY ANALYSIS

1. Tổng quan về Coursework và mục tiêu:

- Xây dựng một bộ tổng hợp dựa trên Fourier, có tính năng tổng hợp dạng hạt (granular synthesis) trên MATLAB, với chức năng chỉnh sửa và phát lại âm thanh dựa trên STFT.
- Ứng dụng cần cho phép chỉnh sửa phổ (spectrogram) của âm thanh và phát lại âm thanh đã chỉnh sửa theo các yêu cầu về cao độ và nhịp độ.
- Đảm bảo các chức năng cơ bản như điều chỉnh cao độ, nhịp độ và bổ sung một số hiệu ứng âm thanh cơ bản.

2. Các yêu cầu cơ bản của Coursework:

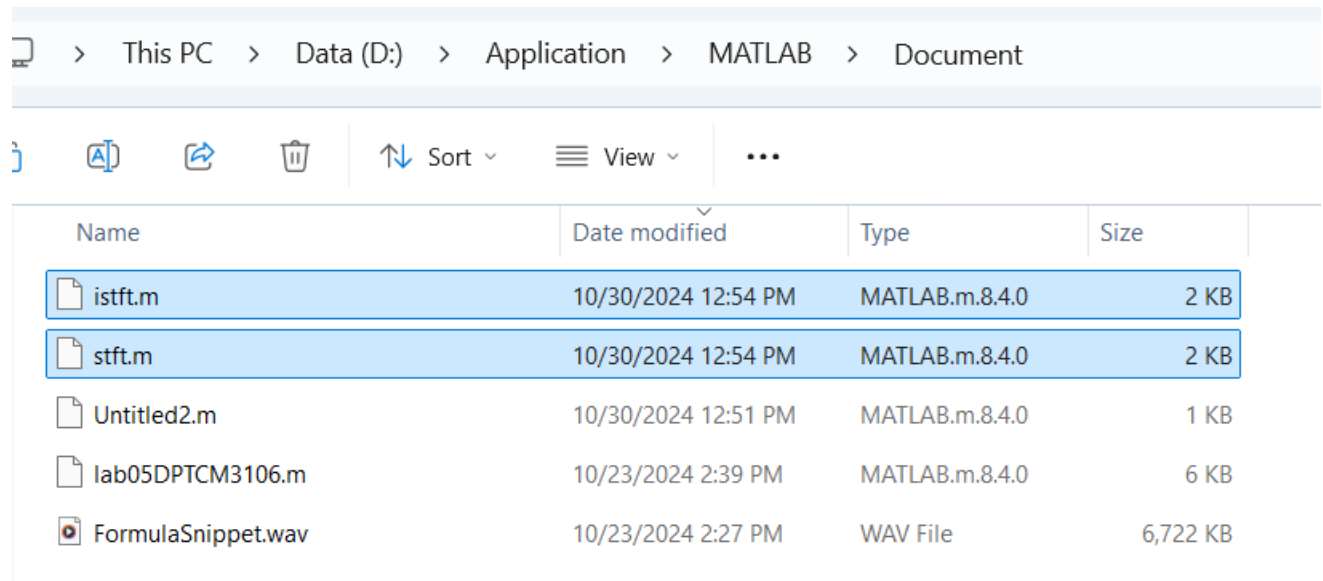
- **Input âm thanh:** Đọc tập tin âm thanh và tính toán STFT để tạo phổ.
- **Chỉnh sửa phổ:** Hiển thị và cho phép chỉnh sửa phổ thời gian-tần số.
- **Phát lại âm thanh:** Thực hiện biến đổi ngược STFT để phát lại âm thanh đã chỉnh sửa, có thể kiểm soát được cao độ và nhịp độ.
- **Bổ sung Granular Convolution:** Ứng dụng kỹ thuật tổng hợp dạng hạt để cải thiện chất lượng âm thanh.

3. Các yêu cầu bổ sung để đạt điểm cao:

- **Chức năng phức tạp hơn** như thêm lớp âm thanh, lặp lại và đảo chiều các đoạn âm thanh, hoặc áp dụng các hiệu ứng âm thanh nâng cao.
- **Cải thiện giao diện người dùng:** GUI có thể bao gồm các điều khiển âm thanh, hiệu ứng và tính năng xuất âm thanh.
- **Hỗ trợ nhiều lớp âm thanh (polyphonic)** và MIDI cho đầu vào dữ liệu.

4. Thực hành

Tải và cài đặt *stft.m* and *istft.m*



4.1. Tính toán STFT

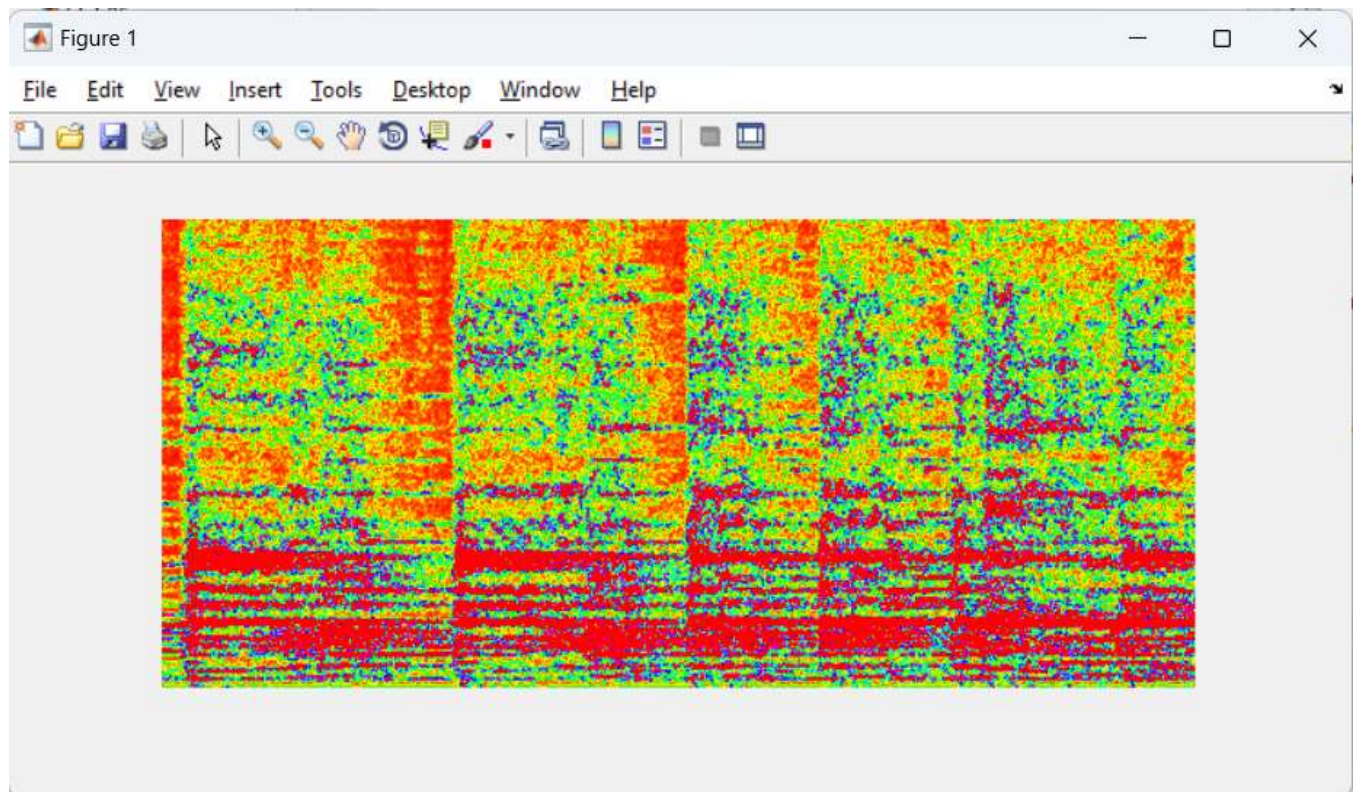
```
function D = stft(x, f, w, h, sr)
    % D = stft(X, F, W, H, SR) Short-time Fourier transform.
    % x: tín hi?u ??u vào, f: ?i?m FFT, w: ?i?m c?a s?, h: ?i?m nh?y
    % Tr? v? khung STFT, m?i c?t là FFT c?a m?t khung tín hi?u.
end
```

4.2. Ví dụ STFT và ISTFT

```
%4.2. Ví d? STFT và ISTFT
load handel.mat; % T?i d? li?u âm thanh
n = 512; % S? ?i?m FFT
nhop = n/4; % S? ?i?m nh?y
Y = stft(y, n, n, nhop); % Tính STFT
yback = istft(Y, n, n, nhop); % Tái t?o l?i tín hi?u âm thanh
sound(yback, Fs); % Phát âm thanh tái t?o
```

4.3. Xây dựng Spectrogram

```
load handel; % Get some audio
% stft parameters (can vary)
n = 512;
nhop = n/4;
Y = stft(y, n, n, nhop);
% Make Spectrogram
specy = abs(Y)/n;
% set left-hand coordinate origin
imshow(flipud(255*specy));
colormap(hsv); %color display
```



4.4. Thay đổi nhịp độ với Phase Vocoder

```
% Get some audio
load handel;
% Half Speed
yslow = pvoc(y, .5, 1024);
% Compare original and resynthesis
sound(y, Fs);
sound(yslow, Fs);
% Twice as Fast
yfast = pvoc(y, 2, 1024);
% Compare original and resynthesis
sound(y, Fs);
sound(yfast, Fs)
```

4.5. Thay đổi cao độ với Phase Vocoder

```
% Get some audio
load handel;
% Pitch up a Fifth
ypvoc = pvoc(y, 0.66666);
ypitch = resample(ypvoc, 2, 3); % NB: 0.666 = 2/3
sound(y, Fs);
sound(ypitch, Fs);
sound(y(1:length(ypitch)) + ypitch, Fs);
% Pitch up an octave
ypvoc = pvoc(y, 0.5);
ypitch = resample(ypvoc, 1, 2);
...
% Pitch down an octave
ypvoc = pvoc(y, 2);
ypitch = resample(ypvoc, 2, 1);
```

4.6. Loại bỏ tạp âm (Hum Removal)

```
close all, clear all;
```

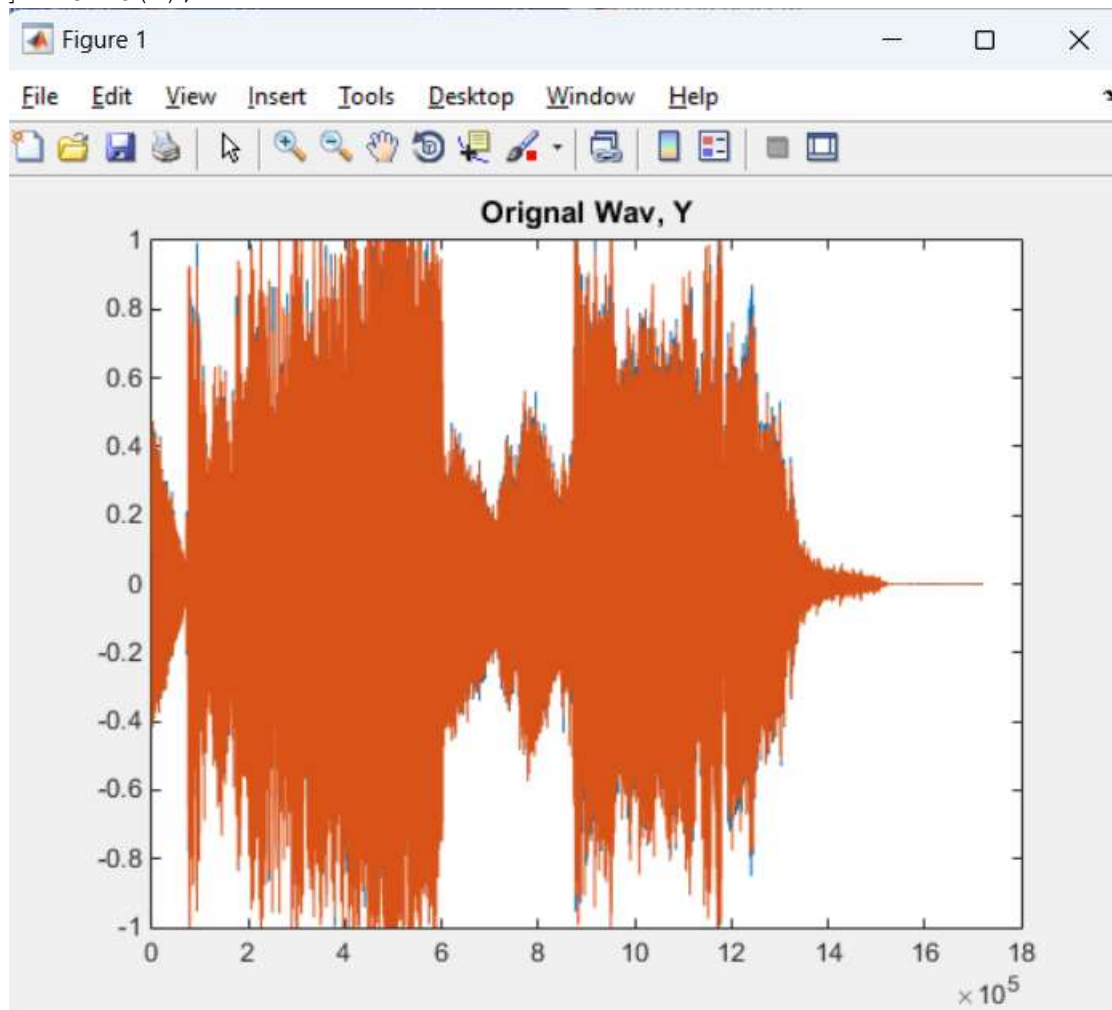
```
[y,Fs] = audioread('FormulaSnippet.wav'); % Get some audio  
reply = input('Press return to play sound? ');
```

```
figure(1)  
plot(y);  
title('Original Wav, Y');
```

```
sound(y, Fs)
```

```
% stft parameters (can vary)  
n = 1024;  
nhop = n/4;  
Y = stft(y,n,n,nhop);
```

```
[yn, ym] = size(Y);
```

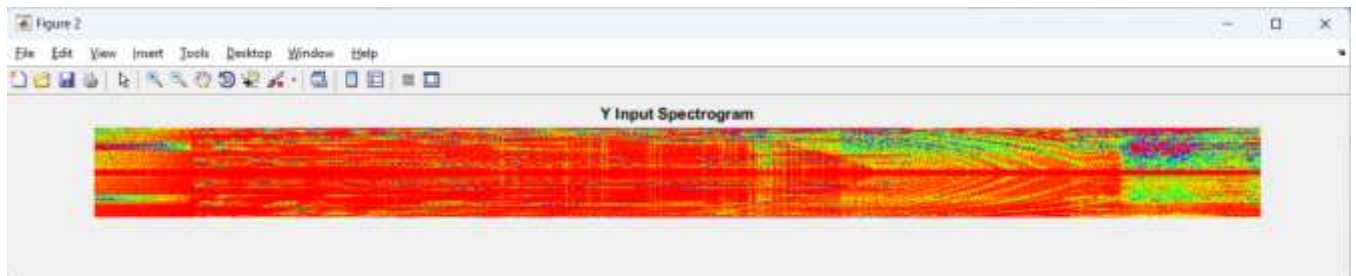


```
figure(2)
imshow(255*abs(Y)/n)
colormap('hsv')
title('Y Input Spectrogram')

% Edit example cut a chunk out of Y Hum in
%Rows 20:22 around 160HZ (= 20*Fs (= 8192)/ n (=1024) )

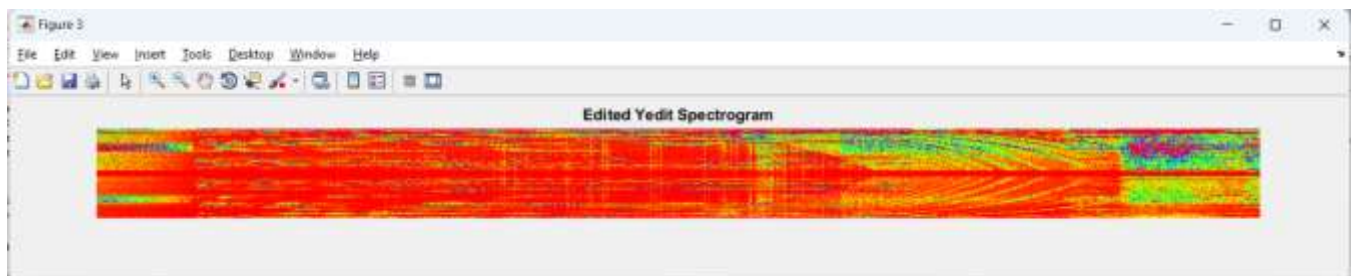
Yedit = Y;

Yresidual = zeros(yn,ym);
Yresidual(20:22,:) = Y(20:22,:);
Yedit(20:22,:) = 0;
```

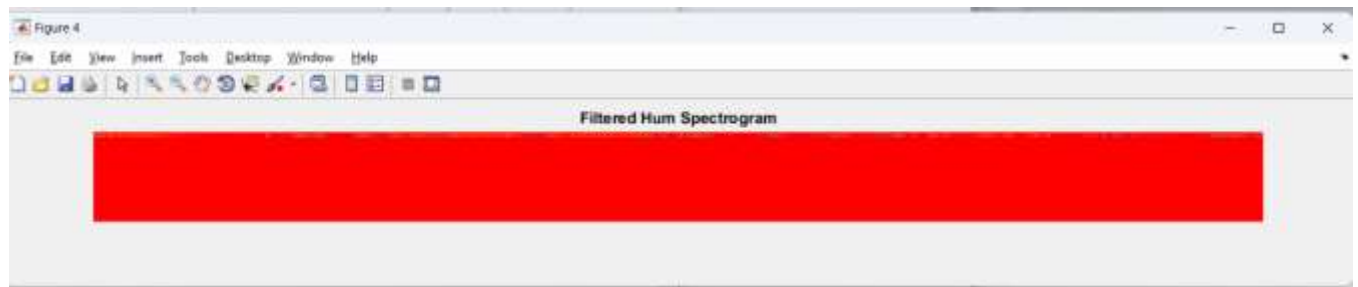


```
reply = input('Press return to play edited sound? ');

figure(3)
imshow(255*abs(Yedit)/n)
colormap('hsv')
title('Edited Yedit Spectrogram')
```



```
figure(4)
imshow(255*abs(Yresidual)/n)
colormap('hsv')
title('Filtered Hum Spectrogram')
```




```
% Remake sound
yedit = istft(Yedit,n,n, nhop);

sound(yedit,Fs)

reply = input('Press return to play residual sound? ');

figure(5)
plot(yedit)
title('Yedit')

figure(6)

plot(yedit(1:length(yedit)) - y(1:length(yedit))')
title('Yedit residual')

% Play residual
sound(yedit(1:length(yedit)) - y(1:length(yedit))',Fs);
```

