**PHÂN ĐOẠN TÍN HIỆU ÂM THANH THÀNH TIẾNG NÓI & KHOẢNG LẶNG DÙNG ĐẶC TRƯNG NĂNG LƯỢNG**

**Giảng viên: Ninh Khánh Duy**

**Trần Phước Thịnh - Nguyễn Thành Long**

**Mã lớp học phần: 1023320.2010.18.15A**

|  |  |  |  |
| --- | --- | --- | --- |
| **Điểm**  (dành cho  GV ghi) | **Bảng phân công nhiệm vụ**  (SV ghi càng cụ thể thì GV càng dễ đặt câu hỏi và cho điểm mỗi SV) | | **Chữ ký của SV**  (mỗi SV ký xác nhận trước khi nộp báo cáo) |
|  | Trần Phước Thịnh | Đọc tài liệu, cài đặt và viết báo cáo về thuật toán X (tr. 10-11) |  |
|  | Nguyễn Thành Long | Đọc tài liệu, cài đặt và viết báo cáo về thuật toán Y (tr. 6-9) |  |

Lời cam đoan: Chúng tôi, gồm các sinh viên có chữ ký ở trên, cam đoan rằng báo cáo này là do chúng tôi tự viết dựa trên các tài liệu tham khảo liệt kê ở cuối báo cáo. Các số liệu thực nghiệm và mã nguồn chương trình nếu không chỉ dẫn nguồn tham khảo đều do chúng tôi tự làm. Nếu vi phạm thì chúng tôi xin chịu trách nhiệm và tuân theo xử lý của giáo viên hướng dẫn.

TÓM TẮT— Phân đoạn tín hiệu thành tiếng nói và khoảng lặng là bài toán cần thiết trong xử lí tín hiệu tiếng nói cũng như đời sống hiện đại hiện nay. Bài toán này thực hiện việc phân đoạn tín hiệu thành tiếng nói và khoảng lặng trên miền thời gian bằng cách sử dụng thuật toán Short Time Energy kết hợp với thuật toán phân đoạn tiếng nói tự cài đặt dựa trên năng lượng . Bài thực hành sẽ thực hiện với bốn file âm thanh khác nhau là: lab\_male.wav, lab\_female.wav, studio\_male.wav, studio\_female.wav. Tín hiệu sẽ được phân nhỏ thành các khung với độ dài 10ms không chồng lên nhau, đồng thời các sẽ thực hiện tính toán năng lượng và điểm cắt zero bằng hai thuật toán nêu trên. Qua đó ta có thể phân chia được các đoạn tiếng nói và khoảng lặng thông qua đồ thị chuẩn hóa năng lượng và điểm cắt zero.

Từ khóa— Khung tín hiệu (frame), Short Time Energy (STE)

Mục lục

[I. ĐẶT VẤN ĐỀ 3](#_Toc55420587)

[II. LÝ THUYẾT XỬ LÝ TÍN HIỆU TIẾNG NÓI VÀ CÁC THUẬT TOÁN 3](#_Toc55420588)

[A. Chia tín hiệu thành các khung 3](#_Toc55420589)

[Để xử lí tính toán, ta cần phải chia tín hiệu thành các frame với độ dài thời gian mỗi frame là 10ms và không chồng lên nhau. 3](#_Toc55420590)

[B. Thuật toán Short Time Energy 3](#_Toc55420591)

[Biên độ của các đoạn không có tiếng nói thấp hơn đáng kể so với với biên độ của cá đoạn có tiếng nói. Thuật toán này phản ánh sự thay đổi chênh lệch giữa biên độ. Công thức của thuật toán Short Time Energy để tính năng lượng ngắn hạn có chỉ số trong khoảng [a,b] là: 3](#_Toc55420592)

[C. Thuật toán phân đoạn tự cài đặt 3](#_Toc55420593)

[D. Thuật toán chuẩn hóa mảng dữ liệu 3](#_Toc55420594)

[III. MÃ CHƯƠNG TRÌNH CÀI ĐẶT CÁC THUẬT TOÁN 4](#_Toc55420595)

[A. Hàm chia tín hiệu thành các khung (file divideFrame.m) 4](#_Toc55420596)

[B. Hàm tính năng lượng 4](#_Toc55420597)

[B.1. Hàm Short Time Energy (file STE.m) 4](#_Toc55420598)

[B.2. Hàm tính năng lượng (file computeEnergy.m) 4](#_Toc55420599)

[C. Hàm chuẩn hóa dữ liệu 5](#_Toc55420600)

[C.1. Hàm chuẩn hóa dữ liệu theo giá trị nhỏ nhất và lớn nhất (file nrmlEnergy1.m) 5](#_Toc55420601)

[C.2. Hàm chuẩn hóa dữ liệu 2 5](#_Toc55420602)

[D. Hàm phân đoạn tín hiệu thành tiếng nói và khoảng lặng (file automaticVoiceSpr.m) 5](#_Toc55420603)

[E. Chương trình chính (file main.m) 6](#_Toc55420604)

[IV. KẾT QUẢ THỰC NGHIỆM 7](#_Toc55420605)

[A. Hình vẽ 7](#_Toc55420606)

[A.1. Tín hiệu vào (4 files studio\_male.m , studio\_female.m , lab\_male.m, lab\_female.m) 7](#_Toc55420607)

[A.2. Đồ thị năng lượng 7](#_Toc55420608)

[A.3. Đồ thị năng lượng đã được nén 8](#_Toc55420609)

[A.4. Đồ thị năng lượng đã được nén và chuẩn hóa 8](#_Toc55420610)

[A.5 Đồ thị tín hiệu sau khi phân đoạn 8](#_Toc55420611)

[B. Kết quả sai số (RMSE) 8](#_Toc55420612)

[B.1. Công thức tính 8](#_Toc55420613)

[V. KẾT LUẬN 9](#_Toc55420614)

[VI. TÀI LIỆU THAM KHẢO 9](#_Toc55420615)

# ĐẶT VẤN ĐỀ

Tiếng nói là phương tiện giao tiếp cơ bản và quan trọng giữa con người và con người. Hiện nay, tiếng nói còn được sử dụng để giao tiếp giữa con người và máy móc hiện đại. Vì vậy, cùng với sự phát triển của khoa học kỹ thuật thì việc xử lí tiếng nói đóng vai trò quan trọng để đáp ứng các nhu cầu của xã hội. Tín hiệu tiếng nói thường được ‘số hóa’ để có thể thực hiện xử lí như nhận dạng, điều chỉnh, giảm nhiễu….

Một tín hiệu tiếng nói cơ bản thường bao gồm khoảng lặng và tiếng nói. Việc phân đoạn tín hiệu tiếng nói là việc xác định các âm vị, âm tiết trong ngôn ngữ tự nhiên được nói. Việc này xác định rõ và phân đoạn khoảng lặng và tiếng nói trong tín hiệu để có thể phân tích, giảm nhiễu, giảm số lượng tín hiệu để thực hiện các xử lí khác….Trong bài báo cáo này sẽ giới thiệu hai phương pháp phân đoạn tín hiệu tiếng nói là: Short Time Energy và Zero Crossing Rate.

Bố cục bài báo cáo sẽ được chia thành bốn phần, trong đó, phần I là ‘Đặt vấn đề’ nhằm đưa ra lời giới thiệu chung về đề tài; phần II là ‘Lý thuyết xử lí tín hiệu tiếng nói và các thuật toán’ nhằm đưa ra các thuật toán lý thuyết được sử dụng trong bài; phần III là ‘Mã chương trình và cài đặt thuật toán’, phần này là các hàm và chương trình từ matlab; phần IV là ‘ Kết quả thực nghiệm’ nhằm đưa ra kết qua bài thực hành; phần V là ‘Kết luận’.

# LÝ THUYẾT XỬ LÝ TÍN HIỆU TIẾNG NÓI VÀ CÁC THUẬT TOÁN

## Chia tín hiệu thành các khung

### Để xử lí tính toán, ta cần phải chia tín hiệu thành các frame với độ dài thời gian mỗi frame là 10ms và không chồng lên nhau.

* Thông qua thuật toán, ta sẽ nhận được:
* Số mẫu trong mỗi frame ( số mẫu trong mỗi frame sẽ bằng nhau trừ frame cuối cùng) bằng công thức: Số mẫu trong mỗi frame = độ dài thời gian của mỗi frame Tỉ lệ lấy mẫu
* Số frame được chia nhỏ = Tổng sô mẫu của tín hiệu Số mẫu trong mỗi frame

\*Lưu ý: Nếu sau khi thực hiện phép chia ta nhận được ‘Số frame được chia nhỏ’ là số thập phân thì ta sẽ làm trong lên để lấy đầy đủ tất cả các mẫu của tín hiệu

## Thuật toán Short Time Energy

### Biên độ của các đoạn không có tiếng nói thấp hơn đáng kể so với với biên độ của cá đoạn có tiếng nói. Thuật toán này phản ánh sự thay đổi chênh lệch giữa biên độ. Công thức của thuật toán Short Time Energy để tính năng lượng ngắn hạn có chỉ số trong khoảng [a,b] là:

E[m]

Trong đó:

E[m] là năng lượng ngắn hạng

n là chỉ số của mẫu trong khoảng [a,b]

m là chỉ số frame đang tính

* Ngoài ra, ta cũng có thể nén năng lượng ngắn hạn để hỗ trợ dễ dàng hơn trong việc xử lí bằng công thức:

Elog[m] = log (E[m])

Trong đó:

Elog là năng lượng ngắn hạng đã được nén lại

E[m] là năng lượng ngắn hạng

## Thuật toán phân đoạn tự cài đặt

* Thuật toán phân đoạn sẽ quét toàn bộ đồ thị hoặc mảng năng lượng, đồng thời kiểm tra mỗi giá trị của phần tử năng lượng với ngưỡng năng lượng, nêu giá trị của của phần tử lớn hơn ngưỡng thì phần tử đó thuộc đoạn tiếng nói, ngược lại nếu giá trị của phần tử đó nhỏ hơn ngưỡng thì phần tử thuộc đoạn khoảng lặng. Ngoài ra, thuật toán sử dụng một biến đếm để kiểm tra khoảng lặng (vì khoảng lặng có thời gian phải lớn hơn 200ms).

## Thuật toán chuẩn hóa mảng dữ liệu

* Để dễ dàng xác định ngưỡng năng lượng chung để phân đoạn chính xác mà không phụ thuộc vào dải biên độ của một tín hiệu cụ thể, ta cần phải chuẩn hóa các giá trị phần tử của từng mảng trong phạm vi [0,1].

Công thức chuẩn hóa:

=

Trong đó:

là mảng đã được chuẩn hóa trong phạm vi [0,1]

là tín hiệu vào

là phần tử có giá trị nhỏ nhất trong tín hiệu

là phần tử có giá trị lớn nhất trong tín hiệu

* Hoặc ta có thể sử dụng công thức chuẩn hóa sau:

=

Trong đó:

là mảng đã được chuẩn hóa trong phạm vi [0,1]

là tín hiệu vào

trung bình cộng giá trị các phần tử trong tín hiệu X

là độ lệch chuẩn của mẫu

# MÃ CHƯƠNG TRÌNH CÀI ĐẶT CÁC THUẬT TOÁN

## Hàm chia tín hiệu thành các khung (file divideFrame.m)

**function [spfr, fr] = divideFrame( y, Fs )**

**%y : tín hiệu vào**

**%Fs: tần số lấy mẫu của tín hiệu vào y**

**%spfr: số mẫu của mỗi khung**

**%fr: số khung**

**period = 0.01;%thời gian của mỗi khung**

**spfr = period\*Fs;**

**fr = ceil(length(y)/spfr);**

**end**

## Hàm tính năng lượng

### B.1. Hàm Short Time Energy (file STE.m)

**function [ e ] = STE( nR, nL, arr )**

**%arr: mảng biên độ tín hiệu**

**%e: năng lượng của mỗi khung**

**%nR: cận trái**

**%nL: cận phải**

**e = 0;**

**for i = nR:nL**

**e = e + arr(i).^2;**

**end**

**end**

### B.2. Hàm tính năng lượng (file computeEnergy.m)

**function [ E, logE ] = computeEnergy( y, spfr, fr )**

**%y: tín hiệu vào**

**%Fs: tần số mẫu của tín hiệu vào y**

**%E: mảng chứa năng lượng của các khung**

**%logE: mảng chứa năng lượng lấy logarit của các khung**

**%spfr: số mẫu của mỗi khung**

**%fr: số khung đã chia được**

**logE = zeros(1, fr);%khởi tạo mảng có fr phần tử**

**E = zeros(1, fr);% khởi tạo mảng có fr phần tử**

**j = 1; %biến đếm**

**for i = 1:spfr:length(y)**

**%tính năng lượng của khung cuối cùng**

**if i+spfr > length(y)**

**E(j) = STE(i, length(y), y);% gọi hàm tính năng lượng của khung**

**LogE(j) = log(E(j)); %lấy logarit**

**break;**

**else**

**E(j) = STE(i, i+spfr-1, y);%gọi hàm tính năng lượng của khung**

**LogE(j) = log(E(j));%lấy logarit**

**end**

**j = j + 1; %tăng biến đếm lên 1 đơn vị**

**end**

**end**

## Hàm chuẩn hóa dữ liệu

### C.1. Hàm chuẩn hóa dữ liệu theo giá trị nhỏ nhất và lớn nhất (file nrmlEnergy1.m)

**function [ x ] = nrmlEnergy1( E )**

**%nrmlEnergy1: hàm chuẩn hóa năng lượng theo giá trị nhỏ nhất và lớn nhất**

**%E: mảng năng lượng cần chuẩn hóa**

**%x: mảng năng lượng đã được chuẩn hóa**

**x = (E-min(E))/(max(E)-min(E));**

**end**

### C.2. Hàm chuẩn hóa dữ liệu 2

#### Hàm tính độ lệch chuẩn (file SD.m)

**function [ sd ] = SD( s )**

**%SD: hàm tính độ lệch chuẩn**

**%s: tập hợp dữ liệu**

**%sd: độ lệch chuẩn**

**avg = mean(s); %trung bình cộng của tập dữ liệu**

**sd = sqrt((norm(s-avg).^2)/(length(s)-1));**

**end**

#### Hàm chuẩn hóa dữ liệu 2 (file nrmlEnergy2.m)

**function [ x ] = nrmlEnergy2( E )**

**%nrmlEnergy1: hàm chuẩn hóa dữ liệu theo độ lệch chuẩn**

**%E: mảng năng lượng cần chuẩn hóa**

**%x: mảng năng lượng đã được chuẩn hóa**

**avg = mean(E); %giá trị trung bình của năng lượng**

**x = (E-avg)/SD(E);**

**end**

## Hàm phân đoạn tín hiệu thành tiếng nói và khoảng lặng (file automaticVoiceSpr.m)

**function [ point ] = automaticVoiceSpr( y, Fs, threshold, spfr, fr, logE )**

**%y: tín hiệu vào**

**%Fs: tần số lấy mẫu của tín hiệu vào y**

**%thresHold: ngưỡng năng lượng chuẩn hóa**

**%spfr: số mẫu của mỗi khung**

**%fr: số khung đã chia**

**%logE: mảng năng lượng lấy logarit**

**x = nrmlEnergy1(logE); %chuẩn hóa năng lượng**

**time = 0; %biến đếm kiểm tra thời gian**

**duration = linspace(0, length(y)/Fs, length(y));**

**point = [];%mảng chứa tọa độ phân đoạn**

**plot(duration, y);**

**d = 1; %biến đếm của mảng a**

**%xét vòng lặp từ trái sang phải của tín hiệu**

**for i = 1:fr**

**if x(i) >= thresHold %so sánh năng lượng đã chuẩn hóa của khung i**

**if time >=20 %kiểm tra có phải khoảng lặng hay không**

**p = i\*spfr/Fs - (spfr/2)/Fs;%lấy tọa độ để phân đoạn**

**point = [point p];**

**line([p p], ylim, 'Linestyle', '--', 'color', 'red');**

**end**

**time = 0; %khởi tạo lại giá trị biến time trong trường hợp gặp năng lượng vượt ngưỡng nhưng không đủ thời gian của một khoảng lặng**

**else**

**time = time +1;**

**end**

**end**

**%xét vòng lặp từ phải sang trái của tín hiệu**

**for i = length(x):-1:1**

**if x(i) >= thresHold %so sánh năng lượng đã chuẩn hóa của khung i**

**if time >=20 %kiểm tra có phải khoảng lặng hay không**

**p = i\*spfr/Fs - (spfr/2)/Fs;%lấy tọa độ để phân đoạn**

**point = [point p];**

**line([p p], ylim, 'Linestyle', '--','color', 'red');**

**end**

**time = 0;%khởi tạo lại giá trị biến time trong trường hợp gặp năng lượng vượt ngưỡng nhưng không đủ thời gian của một khoảng lặng**

**else**

**time = time +1;**

**end**

**end**

**end**

## Chương trình chính (file main.m)

**clear all;**

**[y, Fs] = audioread('studio\_female.wav');%đọc file**

**%y: mảng biên độ tín hiệu**

**%Fs: tần số lấy mẫu của tín hiệu y**

**thresHold = 0.54; %chọn ngưỡng năng lượng chuẩn hóa sau khi xem xét 4 đồ thị tín hiệu đã được chuẩn hóa**

**duration = length(y)/Fs; %thời gian của file**

**t = linspace(0, duration, length(y));**

**[spfr, fr] = divideFrame(y, Fs); %gọi hàm chia khung cho tín hiệu**

**%spfr: số mẫu của mỗi khung**

**%fr: số khung đã chia được**

**[E, logE] = computeEnergy(y, spfr, fr); %gọi hàm tính năng lượng của tín hiệu y**

**%E: mảng năng lượng**

**%logE: mảng năng lượng lấy logarit**

**x1 = nrmlEnergy1(logE); %chuẩn hóa mảng năng lượng logE với hàm chuẩn hóa theo giá trị nhỏ nhất và lớn nhất**

**x2 = nrmlEnergy2(logE);% chuẩn hóa mảng năng lượng logE với hàm chuẩn hóa theo độ lệch chuẩn**

**subplot(311);**

**plot(t, y); %vẽ tín hiệu y**

**title('Tin hieu y');**

**subplot(312)**

**plot(x1);%vẽ năng lượng đã được chuẩn hóa và lấy logarit**

**title('Nang luong chuan hoa');**

**subplot(313)**

**point = automaticVoiceSpr(y, Fs, threshold, spfr, fr, logE);%phân đoạn tiếng nói và khoảng lặng**

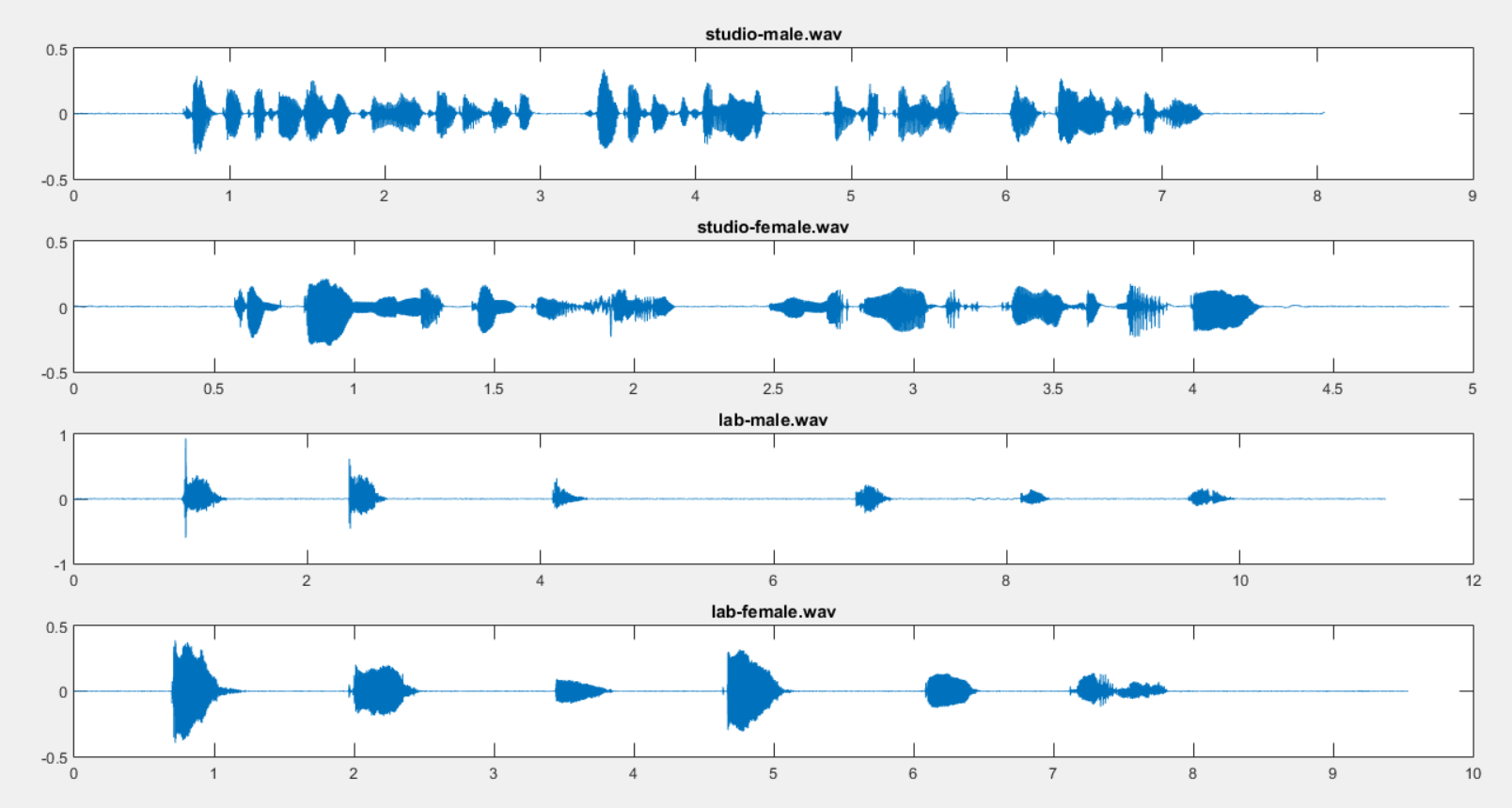
**%point: mảng tọa độ điểm phân đoạn**

**title('Phan doan tieng noi va khoang loang');**

# KẾT QUẢ THỰC NGHIỆM

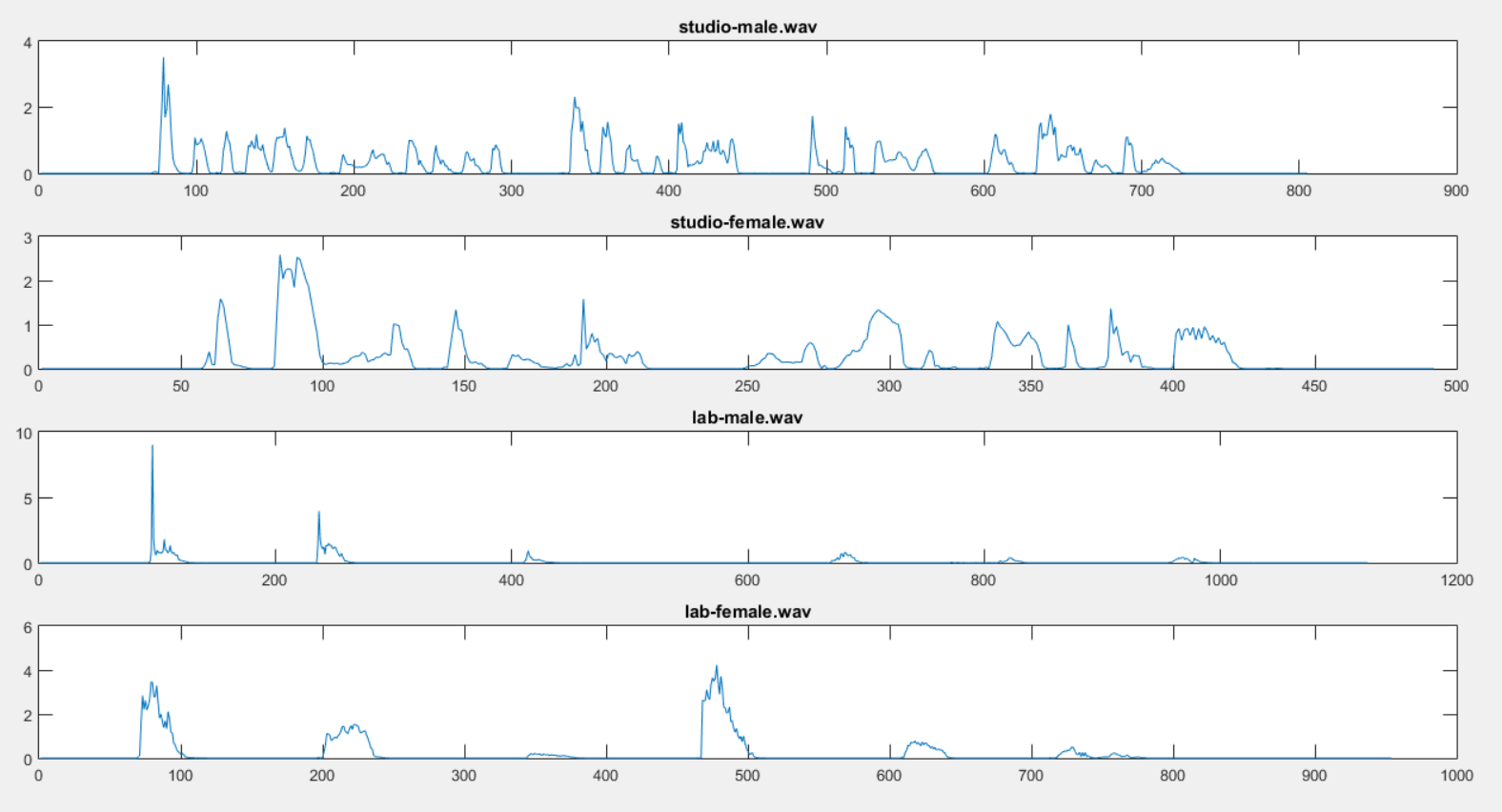
## Hình vẽ

### A.1. Tín hiệu vào (4 files studio\_male.m , studio\_female.m , lab\_male.m, lab\_female.m)

. 

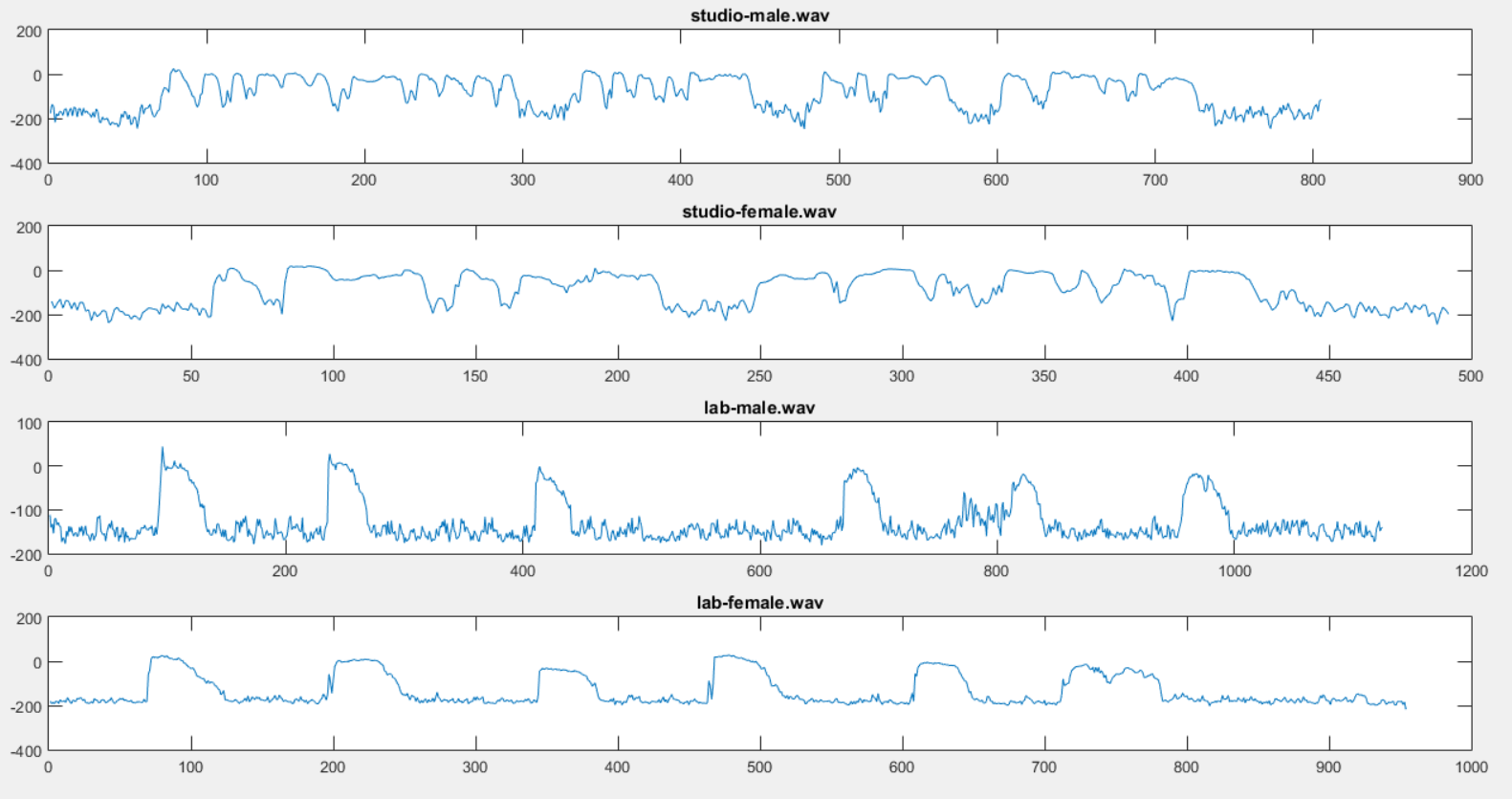
1. Đồ thị tín hiệu từ các files

### A.2. Đồ thị năng lượng



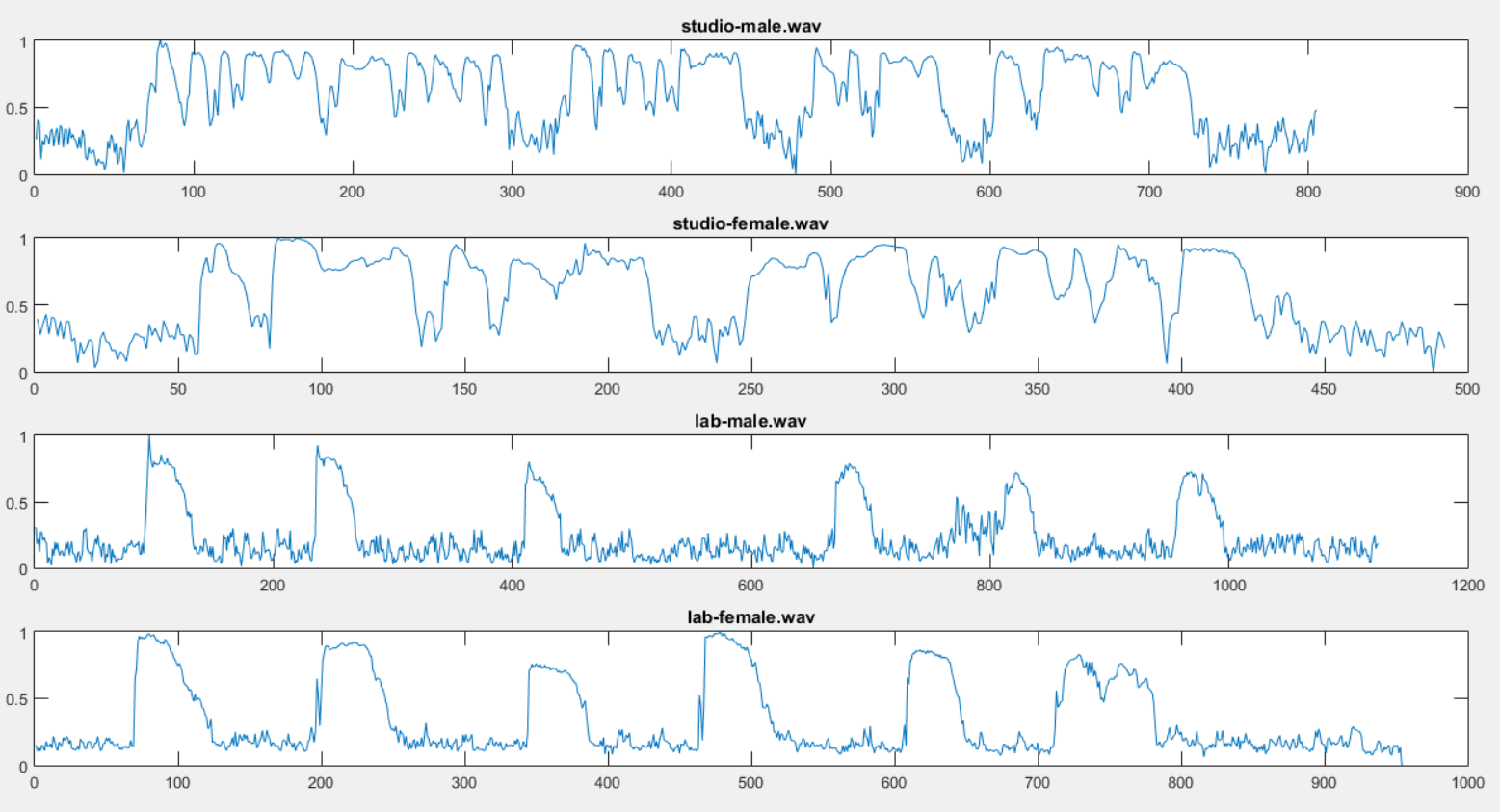
1. Đồ thị năng lượng của các tín hiệu

### A.3. Đồ thị năng lượng đã được nén



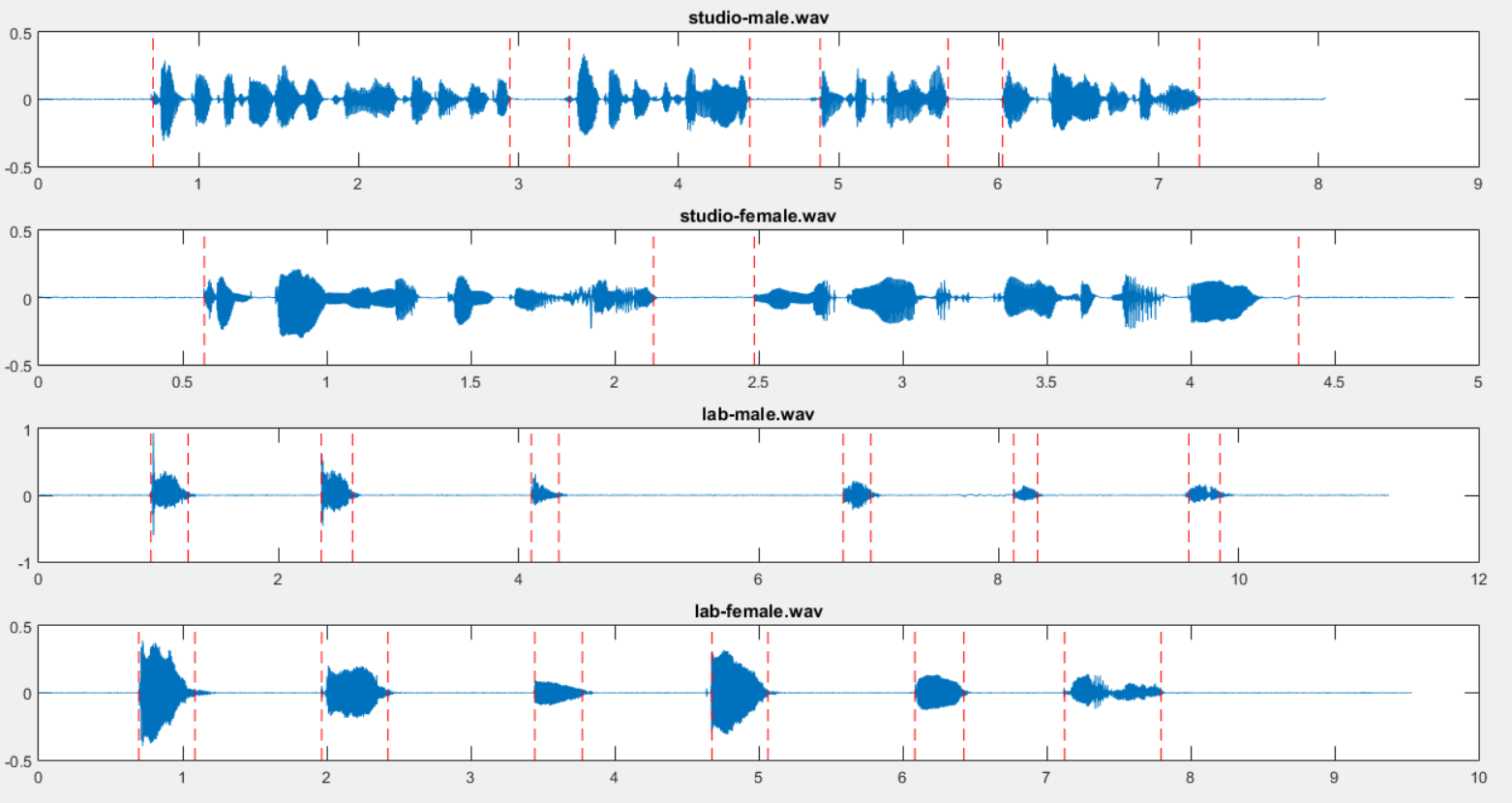
1. Đồ thị năng lượng đã được lấy logarit

### A.4. Đồ thị năng lượng đã được nén và chuẩn hóa



1. Đồ thị năng lượng đã được lấy logarit và chuẩn hóa của các tín hiệu

### A.5 Đồ thị tín hiệu sau khi phân đoạn



1. Đồ thị các tín hiệu sau khi phân đoạn

## Kết quả sai số (RMSE)

### B.1. Công thức tính

* Sai số là giá trị chênh lệch giữa giá trị ước tính và giá trị thực tế nhằm xác định độ chính xác.
* Công thức:

Trong đó:

Xlà mảng giá trị ước tính

Y là mảng giá trị lấy được sai khi tính toán/ giá trị thực tế

N là số phần tử của mảng X và mảng Y

* Tính sai số
  + ***lab\_male.wav***

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0.935 | 2.355 | 4.105 | 6.705 | 8.125 | 9.585 | 9.845 | 8.325 | 6.935 | 4.335 | 2.615 | 1.245 |
|  | 0.934 | 2.35 | 4.1 | 6.7 | 8.13 | 9.584 | 9.953 | 8.38 | 7.027 | 4.4 | 2.69 | 1.32 |

→ = 0.0568 ( N = 12 )

* + ***lab\_female.wav***

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0.696 | 1.965 | 3.445 | 4.675 | 6.085 | 7.125 | 7.795 | 6.425 | 5.065 | 3.775 | 2.425 | 1.085 |
|  | 0.7 | 1.97 | 3.45 | 4.68 | 6.09 | 7.13 | 7.827 | 6.5 | 5.15 | 3.857 | 2.473 | 1.225 |

→ = 0.0596 ( N = 12 )

* + ***studio\_male.wav***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0.715 | 3.315 | 4.885 | 6.025 | 7.255 | 5.685 | 4.445 | 2.945 |
|  | 0.7 | 3.29 | 4.82 | 6 | 7.29 | 5.725 | 4.455 | 2.973 |

→ = 0.0343 ( N = 8 )

* + ***studio\_female.wav***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0.575 | 2.485 | 4.375 | 2.135 |
|  | 0.56 | 2.48 | 4.393 | 2.155 |

→ = 0.0156 ( N = 4 )

# KẾT LUẬN

Chúng em đã tiếp cận việc phân tích tín hiệu, tính toán và chuẩn hóa năng lượng của tín hiệu, phân đoạn tiếng nói và khoảng lặng bằng những cách đơn giản và hiệu quả ( sử dụng Matlab ). Thuật toán cho kết quả tốt trong việc phân loại tiếng nói và khoảng lặng bằng cách chia nhỏ tín hiệu thành các khung tín hiệu có thời gian nhỏ hơn. Trong nghiên cứu tương lai, chúng em hy vọng sẽ cải thiện kết quả và phân biệt, loại bỏ những những tín hiệu nhiễu.

# TÀI LIỆU THAM KHẢO

1. textbook “Prentice Hall - Digital Processing Of Speech Signals\_1978.pdf”.
2. Luận văn cao học “Luận văn\_Tran Van Tam\_2019.pdf”.
3. CS425 Audio and Speech Processing\_Hodgkinson\_2012.