

**PHÂN ĐOẠN TÍN HIỆU TIẾNG NÓI VÀ KHOẢNG LẶNG  
DÙNG SHORT-TIME ZERO-CROSSING RATE, SHORT-TIME ENERGY  
VÀ PHƯƠNG PHÁP ĐẶT NGUỒN**

**Nguyễn Hải Đăng, Nguyễn Vũ Anh Duy, Trần Hữu Hồng Sơn, Nguyễn Vũ Anh Duy**

**Nhóm 12, lớp HP: 17Nh10**

<b>Điểm</b> (dành cho GV ghi)	<b>Bảng phân công nhiệm vụ</b> (SV ghi càng cụ thể thì GV càng dễ đặt câu hỏi và cho điểm mỗi SV)		<b>Chữ ký của SV</b> (mỗi SV ký xác nhận trước khi nộp báo cáo)
	Nguyễn Hải Đăng (nhóm trưởng)	Tìm kiếm, đọc tài liệu và cài đặt về phần chuẩn hóa đồ thị zcr và energy về một trục. Cài đặt và viết báo cáo về phần những điểm giao giữa 2 đồ thị, phân phân đoạn tín hiệu thành tiếng nói và khoảng lặng(tr. 10-11)	
	Nguyễn Vũ Anh Duy	Tìm kiếm, đọc tài liệu, cài đặt và viết báo cáo phần chuẩn hóa đồ thị zcr và energy về một trục. Cài đặt phần phân đoạn tín hiệu thành tiếng nói và khoảng lặng	
	Nguyễn Đình Hân	Tìm kiếm, đọc tài liệu, cài đặt và viết báo cáo phần chia khung, phần tính tổng năng lượng, tổng số lần băng qua không của mỗi khung. Làm slide và thuyết trình Powerpoint	
	Trần Hữu Hồng Sơn	Tìm kiếm, đọc tài liệu và cài đặt phần chia khung, xác định những điểm giao của 2 đồ thị zcr và energy, tổng hợp tiếng nói có trong tín hiệu. Viết báo cáo phần đặt vấn đề, lý thuyết và kết quả thực nghiệm	

**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 ghi rõ trong phần VII. 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**— Xử lý tiếng nói là sự nghiên cứu tiếng nói của con người dưới dạng tín hiệu, các tín hiệu ngoài đời thực được thể hiện ở dạng số (xử lý tín hiệu số) [1]. Có nhiều loại xử lý tiếng nói trong đó có phân đoạn tiếng nói là việc xác định ranh giới giữa những đoạn tín hiệu có tiếng nói (voiced) và không có tiếng nói (unvoiced) [2] bằng một phương pháp hay một công cụ nào đó. Trong báo cáo này chúng tôi sử dụng phương pháp xử lý dựa vào các đặc trưng của tín hiệu tiếng nói trên miền thời gian sử dụng công cụ Matlab. Kết quả thực nghiệm của việc xử lý là xóa đi những khoảng lặng của một tín hiệu âm thanh được thu âm trước đó (vừa có tiếng nói vừa có khoảng lặng).

**Từ khóa**— short-term zero-crossing rate, short-term energy, frame-based processing, signal segmentation, normalize magnitude

## Mục lục

<b>I. ĐẶT VẤN ĐỀ.....</b>	<b>3</b>
<b>II. LÝ THUYẾT XỬ LÝ TÍN HIỆU TIẾNG NÓI VÀ CÁC THUẬT TOÁN .....</b>	<b>3</b>
A.    Mối liên hệ giữa tín hiệu liên tục và tín hiệu rời rạc .....	3
B.    Zero-crossing rate (ZCR).....	3
C.    Năng lượng (Energy) .....	4
<b>III. MÃ CHƯƠNG TRÌNH CÀI ĐẶT CÁC THUẬT TOÁN .....</b>	<b>5</b>
A.    Sơ đồ thuật toán .....	5
B.    Chương trình chính .....	5
C.    Các hàm sử dụng ở chương trình trên .....	6
1. Hàm chia tín hiệu ban đầu thành các khung .....	6
2. Hàm tính tổng năng lượng của mỗi khung .....	6
3. Hàm tính tổng số lần băng qua zero của mỗi khung .....	7
4. Chuẩn hóa hai đồ thị short-time energy và short-time ZCR về cùng 1 trục (Normalized Magnitude).....	7
5. Xác định những điểm giao nhau của hai đồ thị short-time energy và short-time ZCR (Determine intersections).....	7
6. Phương pháp đặt ngưỡng biên độ để phân đoạn tín hiệu (Amplitude Threshold) .	8
<b>IV. KẾT QUẢ THỰC NGHIỆM .....</b>	<b>9</b>
A.    Kết quả định tính.....	9
B.    Kết quả định lượng.....	13
<b>V. KẾT LUẬN .....</b>	<b>14</b>
<b>VI. TÀI LIỆU THAM KHẢO.....</b>	<b>14</b>

## I. ĐẶT VẤN ĐỀ

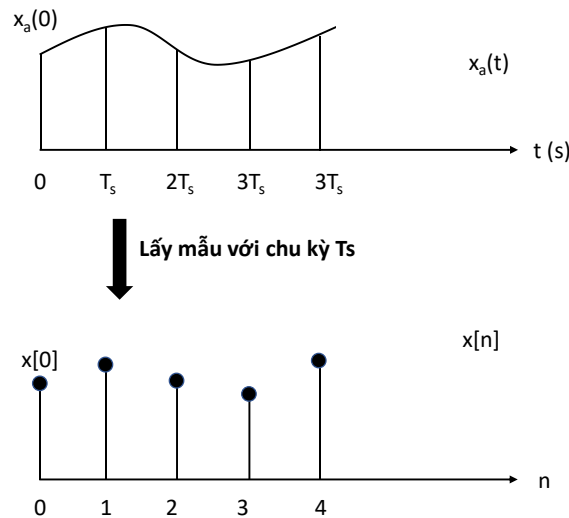
Trong xử lý tiếng nói, có thể xem như việc phân đoạn tín hiệu là việc phân tích, lọc tín hiệu thành những đoạn tín hiệu có tiếng nói (voiced) và những đoạn tín hiệu không có tiếng nói (unvoiced). Trong báo cáo này chúng tôi sẽ dùng 3 đặc trưng trên miền thời gian của một tín hiệu rời rạc để phân đoạn voiced và unvoiced của một tín hiệu âm thanh cho trước đó là đặc trưng về tốc độ băng qua không (zero crossing rate) và đặc trưng về năng lượng (energy). Và một đặc trưng quan trọng khác là tín hiệu tiếng nói có tính chất biến đổi chậm theo thời gian (tức là các đặc trưng tín hiệu dường như không đổi trong những khoảng thời gian ngắn  $10 \div 30$  ms) vì vậy chúng tôi lựa chọn phương pháp xử lý tín hiệu ngắn hạn (short-term processing) tức là xử lý tín hiệu theo từng khung (frame-based processing) kết hợp sử dụng đặc trưng zero-crossing rate thấp thì ở đó có tín hiệu tiếng nói, cao thì ở đó tín hiệu là khoảng lặng, còn đặc trưng energy thì ngược lại ở đoạn có năng lượng thấp thì ở đó là tín hiệu là khoảng lặng, năng lượng cao thì ở đó là tín hiệu tiếng nói. Với mỗi đặc trưng ta có thể vẽ được một đồ thị trên miền thời gian và việc kết hợp 2 đặc trưng nói trên (chuẩn hóa 2 đồ thị zcr và energy về cùng một thang biên độ - normalized magnitude) ta được điểm giao giữa hai đồ thị đó cũng chính là biên của đoạn tiếng nói, đó cũng chính là mục tiêu của việc phân đoạn tín hiệu tiếng nói.

Báo cáo có bố cục như sau. Phần II trình bày về lý thuyết phục vụ cho việc xử lý tín hiệu tiếng nói, lưu đồ thuật toán. Phần III trình bày mã nguồn chương trình và các thuật toán. Phần IV trình bày kết quả thực nghiệm của việc xử lý tín hiệu tiếng nói với 6 file âm thanh cho trước.

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

### A. Mối liên hệ giữa tín hiệu liên tục và tín hiệu rời rạc

Xử lý tín hiệu số là việc biểu diễn tín hiệu ngoài đời thực bằng dưới dạng số [1] đó cũng là công việc bắt buộc để chúng ta có thể thực hiện các bước tiếp theo để xử lý một tín hiệu. Vì vậy chúng ta cần tìm ra một mối liên hệ nào đó giữa tín hiệu liên tục và tín hiệu rời rạc (tín hiệu nhận được nhờ việc lấy mẫu) hay nói cách khác là tìm ra một công thức để biểu diễn mối quan hệ giữa biến thời gian liên tục và biến thời gian rời rạc.



**Hình 1.** Minh họa mối liên hệ giữa TH liên tục và TH rời rạc thông qua việc lấy mẫu.

Từ hình vẽ trên ta có được mối quan hệ như sau:

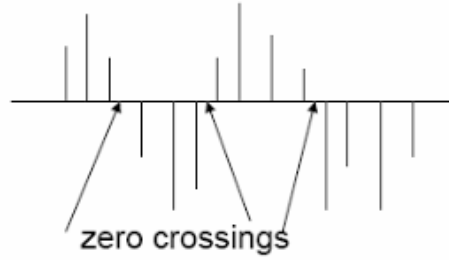
$$x[n] = x_a(t) \text{ hay } t = n * T_s \quad \text{với } t \in \mathbb{R} \text{ và } n \in \mathbb{Z}$$

Từ công thức trên ta có được mối liên hệ giữa số lượng mẫu  $N$  và độ dài  $D$  đơn vị (s) của một tín hiệu

$$D = N * T_s \quad (\text{s}) \quad \text{với } T_s = \frac{1}{F_s}$$

### B. Zero-crossing rate (ZCR)

Tốc độ băng qua không (ZCR) là một trong những đặc trưng trên miền thời gian của tín hiệu rời rạc. Zero-crossing rate là một thông số cho biết số lần mà biên độ tín hiệu băng qua điểm zero trong một khoảng thời gian nhất định.



**Hình 2.** Định nghĩa tốc độ băng qua không (zero-crossing rate) [3].

Theo đặc trưng của tín hiệu, ở những đoạn tín hiệu có zero-crossing rate cao thì ở đó tín hiệu là khoảng lặng, ngược lại ở những vị trí có zero-crossing rate thấp thì ở đó tín hiệu là có tiếng nói. Vì tín hiệu tiếng nói có tính chất biến đổi chậm theo thời gian, có nghĩa là các đặc trưng của tín hiệu gần như không thay đổi theo những khoảng thời gian ngắn hạn từ 10÷30 ms. Áp dụng vào xử lý tiếng nói ta sử dụng phương pháp xử lý ngắn hạn (short time processing) tức là chia tín hiệu ban đầu thành những khung bằng nhau (frame-based processing), mỗi khung từ 10÷30 ms. Vì vậy ta có công thức của hàm ZCR:

$$ZCR_m = \text{số lần } \{ x[n] * x[n + 1] < 0 \}$$

Với  $x[n]$  là vector biên độ lấy được từ âm thanh được thu âm trước. Trong Matlab ta có thể xác định vector  $x$  bằng cách dùng hàm `audioread("file path")`,  $m$  là chỉ số khung thứ  $m$  và  $n \in [N1m, N2m]$  với  $N1, N2$  là chỉ số mẫu bắt đầu và kết thúc của khung thứ  $m$ .

### C. Năng lượng (Energy)

Việc tính năng lượng của một tín hiệu là một tính toán thường được sử dụng trong xử lý tín hiệu. Cũng giống như zero-crossing rate, năng lượng của tín hiệu (energy) cũng là một đặc trưng của trên miền thời gian của tín hiệu rời rạc.

Năng lượng của một tín hiệu được phát biểu như sau: “Năng lượng của một tín hiệu bằng tổng bình phương biên độ tại các mẫu.”

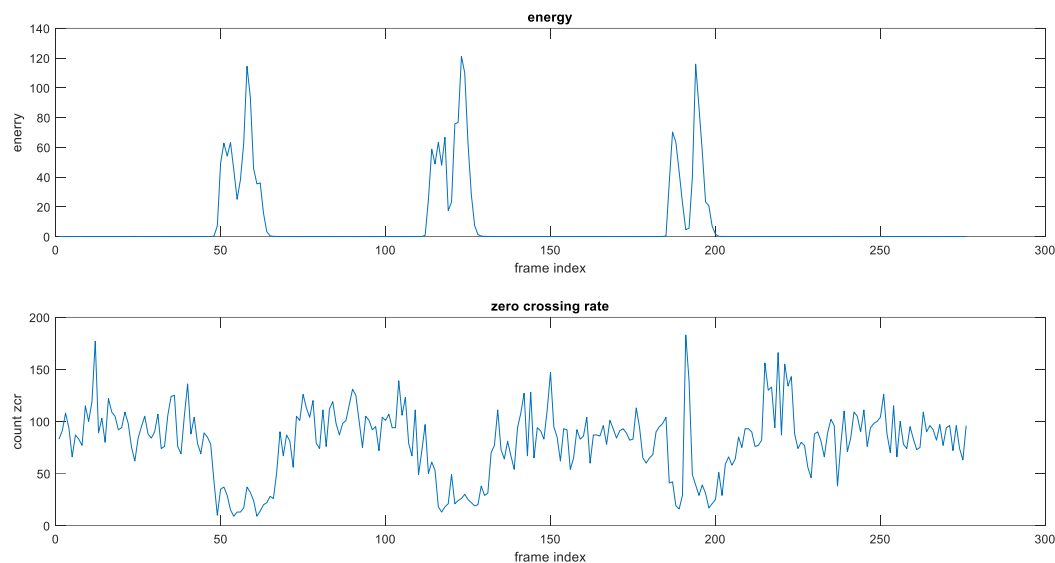
$$E = \sum_{n=-\infty}^{\infty} (x[n])^2$$

Năng lượng là một đặc trưng của tín hiệu tiếng nói mà tín hiệu tiếng nói có tính biến đổi chậm theo thời gian, tức là đặc trưng năng lượng cũng gần như không đổi theo thời gian trong những khoảng thời gian ngắn hạn từ 10÷30 ms, vì vậy ta có thể áp dụng kỹ thuật xử lý tín hiệu ngắn hạn. Bằng cách chia tín hiệu thành nhiều khung có thời gian bằng nhau (10÷30 ms), do đó ta có công thức hàm năng lượng ngắn hạn như sau:

$$E_m = \sum_{n=N1m}^{N2m} (x[n])^2$$

Với  $m$  là chỉ số khung thứ  $m$  và  $n \in [N1m, N2m]$  với  $N1, N2$  là chỉ số mẫu bắt đầu và kết thúc của khung thứ  $m$ .

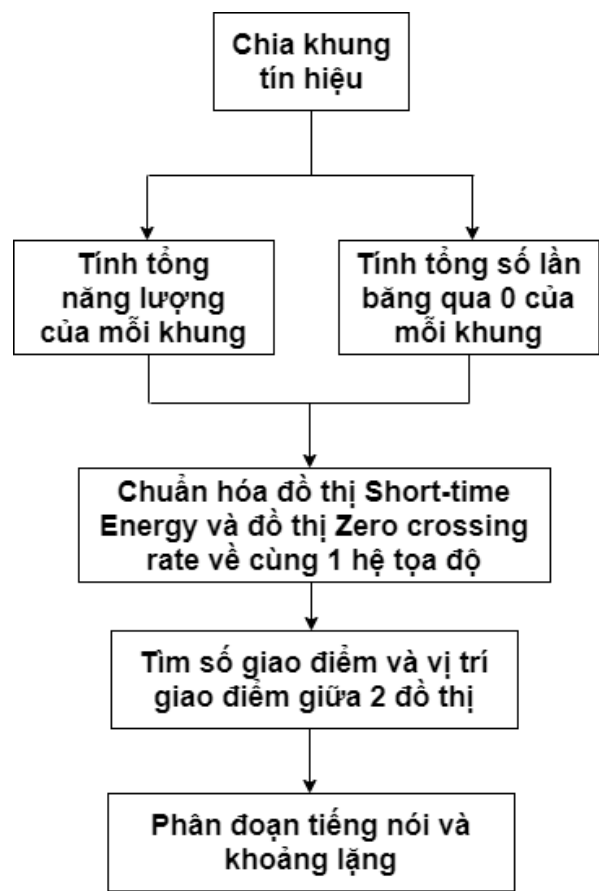
Theo đặc trưng của tín hiệu tiếng nói ở những vị trí có năng lượng cao thì ở đó là tín hiệu tiếng nói, ngược lại ở những vị trí có năng lượng thấp thì ở đó là tín hiệu khoảng lặng. Hình dưới đây gồm 2 đồ thị biểu diễn 2 hàm  $E$  và  $ZCR$  theo phương pháp short-time processing.



**Hình 3.** Hai đồ thị short-time energy và short-time zero-crossing rate của một tín hiệu âm thanh có 3 đoạn tiếng nói và 4 khoảng lặng.

**III. MÃ CHƯƠNG TRÌNH CÀI ĐẶT CÁC THUẬT TOÁN**

**A. Sơ đồ thuật toán**



**Hình 4.** Sơ đồ khối thuật toán.

**B. Chương trình chính**

% đọc tín hiệu từ file: x chứa tất cả giá trị biên độ, Fs: tần số lấy mẫu

```

[x,Fs]=audioread('C:\Users\PC\Desktop\test_signals\studio_female.wav');
D = 0.01; % D (s) độ dài của một frame
frame_len = D*Fs; % frame_len: số mẫu trên một frame
N = length(x); % N: tổng số mẫu của tín hiệu

% tính tổng số frame
num_frame = functionNumberOfFrames(N, frame_len);

% vector năng lượng của tất cả khung của tín hiệu
Energy = functionSumEnergy(x, num_frame, frame_len);

% vector chứa số lần băng qua không của tất cả khung của tín hiệu
CountZCR = functionSumZCR(x, num_frame, frame_len);

% chuẩn hóa đồ thị Energy và CountZCR về cùng 1 thang biên độ
Energy = functionNormalized(Energy, CountZCR);

% xác định số lần giao nhau và vị trí giao nhau gần đúng của 2 đồ thị
temp = functionSumGiaoNhau(num_frame, Energy, CountZCR);

% tổng hợp tất cả các tiếng nói có trong tín hiệu
result = functionSumNewSignal(x, num_frame, frame_len);

t = length(result)/Fs; % độ dài thời gian thực (giây) của đoạn âm thanh
sau khi phân đoạn
t = linspace(0, t, length(result)); % chia biến t về cùng số mẫu với
result để vẽ đồ thị
plot(t, result);
title("tín hiệu được phân đoạn");
xlabel("thời gian (s)");
ylabel("biên độ");

```

### C. Các hàm sử dụng ở chương trình trên

#### 1. Hàm chia tín hiệu ban đầu thành các khung

```

% hàm tính các tổng số khung của tín hiệu
% trả về: tổng số frame của tín hiệu
% N: tổng số mẫu của tín hiệu
% frame_len: số mẫu trên một frame
function num_frame = functionNumberOfFrames(N, frame_len)
    num_frame = floor(N/frame_len);
end

```

#### 2. Hàm tính tổng năng lượng của mỗi khung

```

% hàm tính năng lượng của các khung
% trả về: vector energy chứa năng lượng của tất cả các khung
% x: vector biên độ lấy từ âm thanh
% num_frame: tổng số frame của tín hiệu
% frame_len: số mẫu trên một frame
function energy = functionSumEnergy(x, num_frame, frame_len)
    energy = zeros(1,num_frame); % vector chứa năng lượng của các
    khung
    for k=1: num_frame % duyệt tất cả các khung
        tempEnergy = 0; % dùng biến tempEnergy để lưu giá trị
        Energy[i]
        for j=(k-1)*frame_len+1 : (frame_len*k - 1) % duyệt qua tất
        cả các mẫu có trong khung
            tempEnergy = tempEnergy + abs(x(j)^2);
        end
        energy(k) = tempEnergy; % gán lại năng lượng của khung thứ
        k và vector energy
    end
end

```

```

    end
end

```

### 3. Hàm tính tổng số lần băng qua zero của mỗi khung

```

% hàm tính tổng số lần băng qua zero của tất cả các khung
% trả về: vector chứa tổng số lần băng qua zero của tất cả các
khung
% x: vector biên độ lấy từ âm thanh
% num_frame: tổng số frame của tín hiệu
% frame_len: số mẫu trên một frame
function countZCR = functionSumZCR(x, num_frame, frame_len)
    countZCR = zeros(1,num_frame); khai báo vector chứa giá trị trả
về của hàm
    for k=1: num_frame % duyệt tất cả các khung
        tempZCR = 0; % dung biến tempZCR để lưu giá trị năng lượng
khung thứ k
        for j=(k-1)*frame_len+1 : (frame_len*k - 1) % duyệt qua tất
cả các mẫu có trong khung
            if(x(j) * x(j+1) < 0) % kiểm tra khung j và j+1 biên độ
có băng qua zero không
                tempZCR = tempZCR + 1; % tăng giá biến tempZCR
            end
        end
        countZCR(k) = tempZCR; % gán lại tổng số lần zero crossing
của khung thứ k và vector countZCR tại vị trí k
    end
end

```

### 4. Chuẩn hóa hai đồ thị short-time energy và short-time ZCR về cùng 1 trục (Normalized Magnitude)

```

% hàm chuẩn hóa đồ thị zcr và energy về 1 thang biên độ
% trả về:
% energy: vector chứa tất cả giá trị năng lượng của tất cả các
khung
% zcr: vector chứa số lần băng qua 0 của tất cả các khung
function energyNormal = functionNormalized(energy, zcr)
    ymin = min(zcr);
    ymax = max(zcr);

    % chuẩn hóa đồ thị energy theo đồ thị zcr
    energyNormal = mapminmax(energy, ymin, ymax);
end

```

### 5. Xác định những điểm giao nhau của hai đồ thị short-time energy và short-time ZCR (Determine intersections)

```

% XÁC ĐỊNH SỐ LẦN GIAO NHAU VÀ VỊ TRÍ GIAO NHAU GAN DUNG CỦA 2 DO
THI
function temp = functionSumGiaoNhanh(num_frame, energy, zcr)
    count = 0; % đếm số lần giao nhau của 2 đồ thị
    arr = zeros(1,num_frame); % tạo vector arr để lưu giá trị tính
toán
    for i=1 : (num_frame-1) % chạy hết tất cả các khung
        if((zcr(i) > energy(i) && zcr(i+1) < energy(i+1) &&
(zcr(i)-energy(i))*(zcr(i+1)-energy(i+1)) < 0) || (zcr(i) <
energy(i) && zcr(i+1) > energy(i+1) && (zcr(i)-
energy(i))*(zcr(i+1)-energy(i+1)) < 0)) % 2 trường hợp làm cho 2 đồ
thị giao nhau xét giữa 2 khung liên tiếp nhau
            arr(count+1) = i; % lưu vị trí điểm giao nhau vào arr
            count = count + 1; % tăng số điểm giao nhau lên 1
        end
    end
    temp = arr(1:count); % lưu kết quả vào vector temp có độ dài
chính là tổng số lần giao nhau của 2 đồ thị
end

```

### 6. Phương pháp đặt ngưỡng biên độ để phân đoạn tín hiệu [4] (Amplitude Threshold)

```
% PHAN DOAN TIN HIEU THANH TIENG NOI VA KHOANG LANG
function result = functionSumNewSignal(x, num_frame, frame_len, Fs)
new_sig = zeros(length(x),1); % tao vector new_sig de luu gia tri
bien do cua nhung cho tieng noi
arr = zeros(length(x),1); % neu tai mot khung la khoang lang va
khoang lang do co do dai nho hon 200ms thi luu cac gia tri bien do
cua khung do vao arr
silence = zeros(length(x),1); % tong hop cac khoang lang co do dai
lon hon 200ms
count1 = 0;
count2 = 0;
time = 0; % bien time de xac dinh khoang lang da co do dai lon hon
200ms hay chua
start = 1; % phan tu dau tien cua vector silence
for k=1 : num_frame
    frame = x((k-1)*frame_len+1 : frame_len*k); % cac gia tri bien
do cua x co trong khung k

    max_val = max(frame); % gia tri bien do lon nhat trong khung k

    if(max_val<0.03) % neu gia tri max be hon gia tri nguong bien
do cua khoang lang dat ra ban dau la 0.03
        count1 = count1 + 1;
        arr((count1-1)*frame_len+1 : frame_len*count1) = frame;
% luu cac gia tri bien do khung do vao arr
        time = time + frame_len/Fs; % tang time neu time dang be
hon 200ms
    else % neu max_val>0.03 thi co the la speech hoac noise
        if(time >= 0.2) % neu do dai silence >= 200ms thi lay,
<200ms thi bo di
            silence(start : (start + frame_len*count1 - 1)) = arr(1
: frame_len*count1); % luu khoang lang co do dai >200ms vao silence
            final = start + frame_len*count1 - 1; % final la phan
tu cuoi cung cua vector silence
            start = final + 1;
            arr = zeros(length(x),1); % reset lai arr
            count1 = 0; % reset lai count1
            time = 0; % reset lai time
        else
            arr = zeros(length(x),1); % reset lai arr

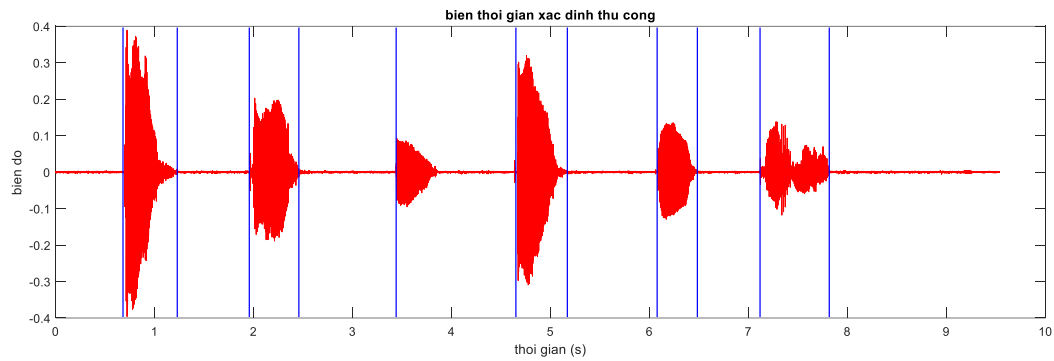
            count1 = 0; % reset lai count1
            time = 0; % reset lai time
        end
        count2 = count2 + 1;
        new_sig((count2-1)*frame_len+1 : frame_len*count2) = frame;
% luu bien do cua khung vao new_sig
    end
end
new_sig(frame_len*count2 + 1 : (frame_len*count2 + final)) =
silence(1 : final); % lay new_sig( = noise/speech) + silence de ra
am thanh ket qua
result = new_sig(1 : (frame_len*count2 + final)); % am thanh ket
qua
end
```



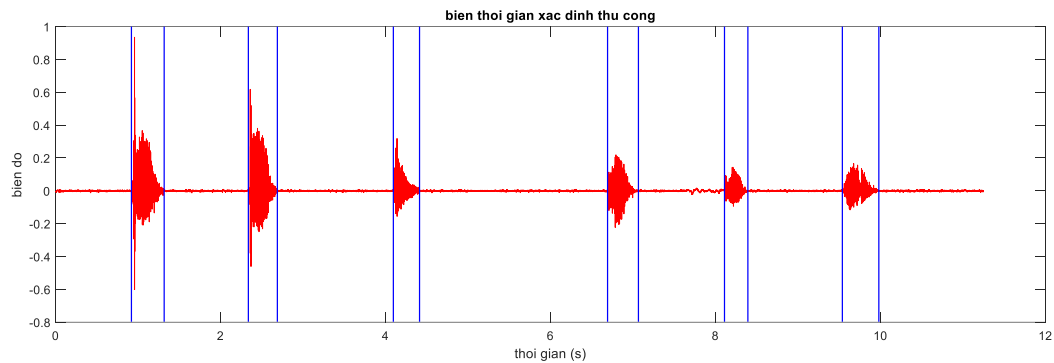
IV. KẾT QUẢ THỰC NGHIỆM

A. Kết quả định tính

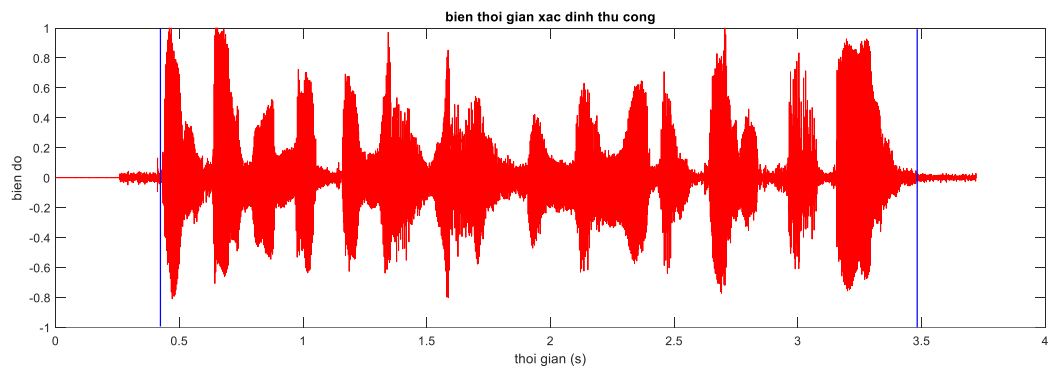
Dưới đây là các hình vẽ minh họa biên thời gian được xác định thủ công:



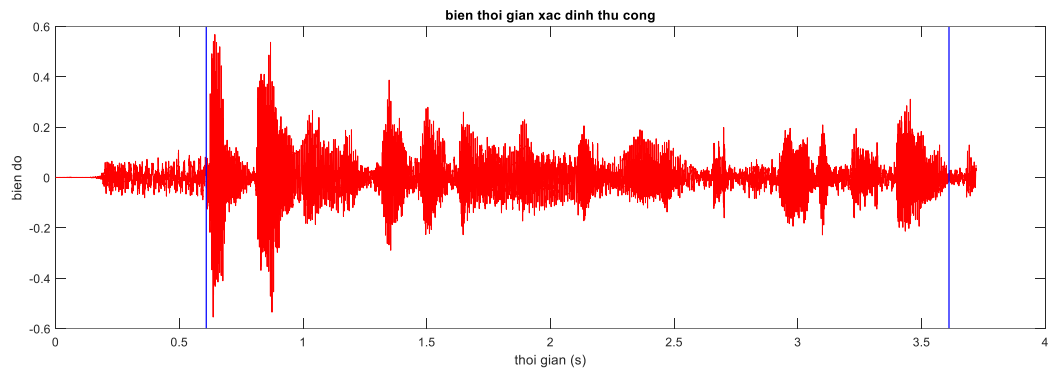
Hình 5. Biên thời gian được xác định thủ công của file lab\_female.wav



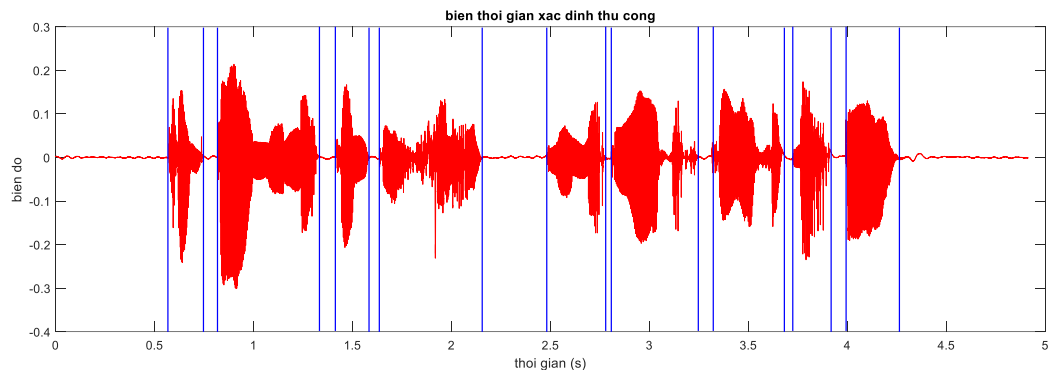
Hình 6. Biên thời gian được xác định thủ công của file lab\_male.wav



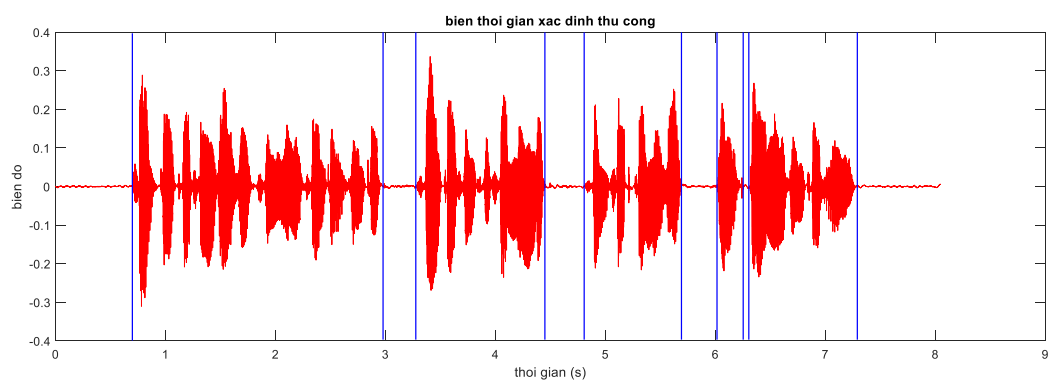
Hình 7. Biên thời gian được xác định thủ công của file phone\_female.wav



Hình 8. Biên thời gian được xác định thủ công của file phone\_male.wav



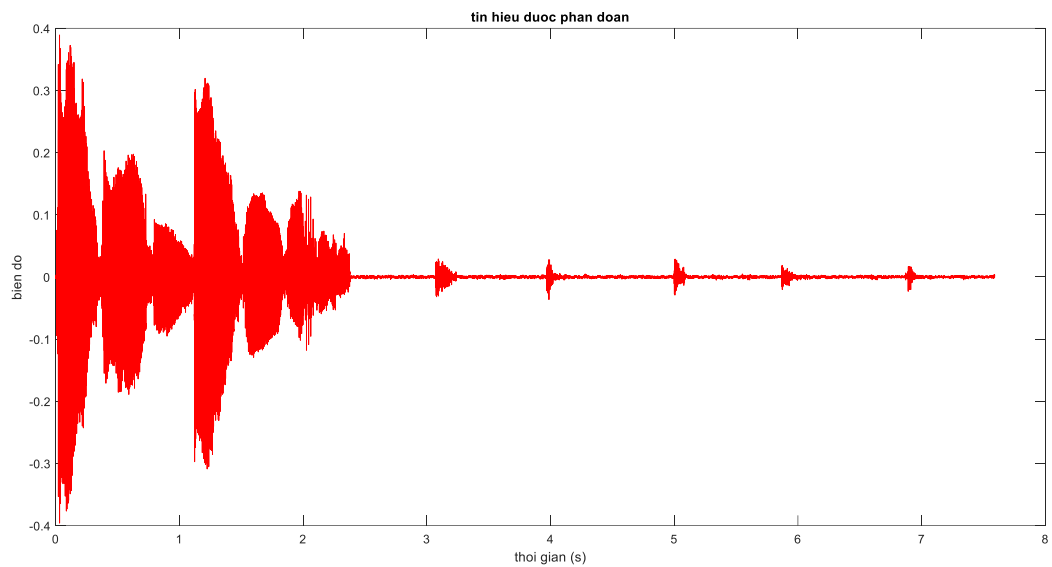
Hình 9. Biên thời gian được xác định thủ công của file studio\_female.wav



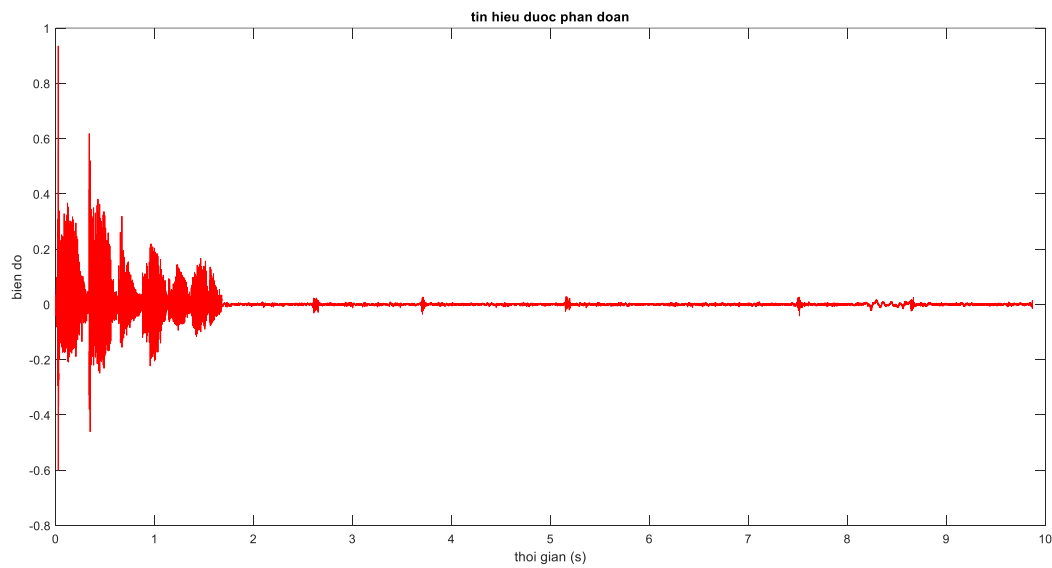
Hình 10. Biên thời gian được xác định thủ công của file studio\_male.wav

Nhận xét: Chúng tôi dùng công cụ Audacity để xác định các biên thời gian giữa voice và unvoiced một cách thủ công (bằng cách kéo và nghe thủ công). Các hình ảnh trên cho thấy việc xác định sẽ trở nên đơn giản hơn nếu âm thanh được thu trong môi trường studio - những file studio\_\*.wav và khó hơn với những file phone\_\*.wav (ranh giới giữa những đoạn có voiced và unvoiced khó được xác định một cách thủ công). Đối với file lab\_female.wav và lab\_male.wav việc xác định đơn giản hơn vì ít lẫn tạp âm và dễ phân biệt vì đoạn tín hiệu khoảng lặng dài hơn.

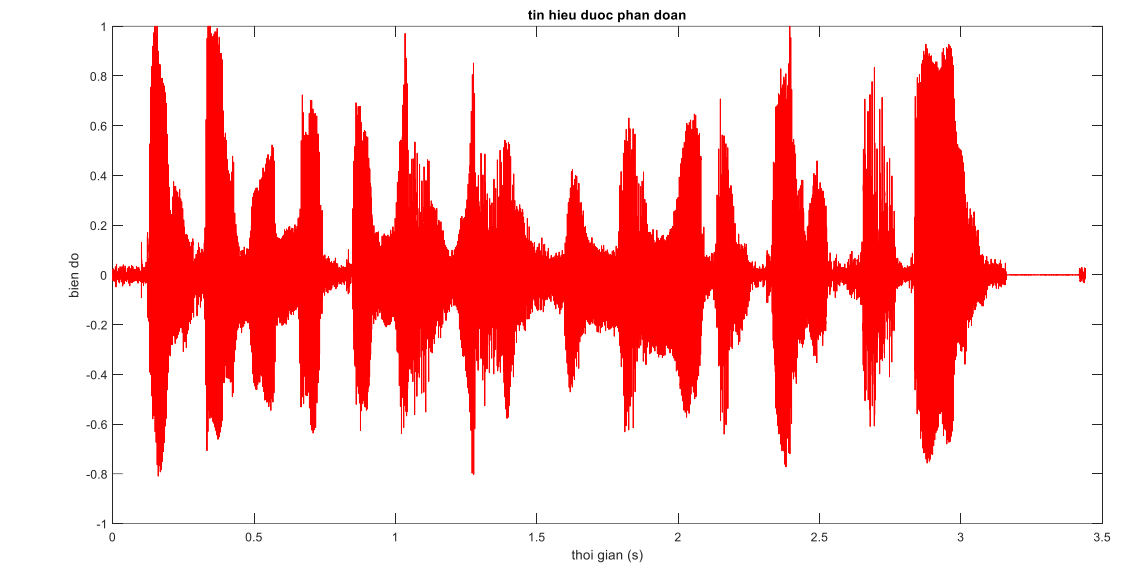
Dưới đây là các hình vẽ của tín hiệu sau khi được phân đoạn:



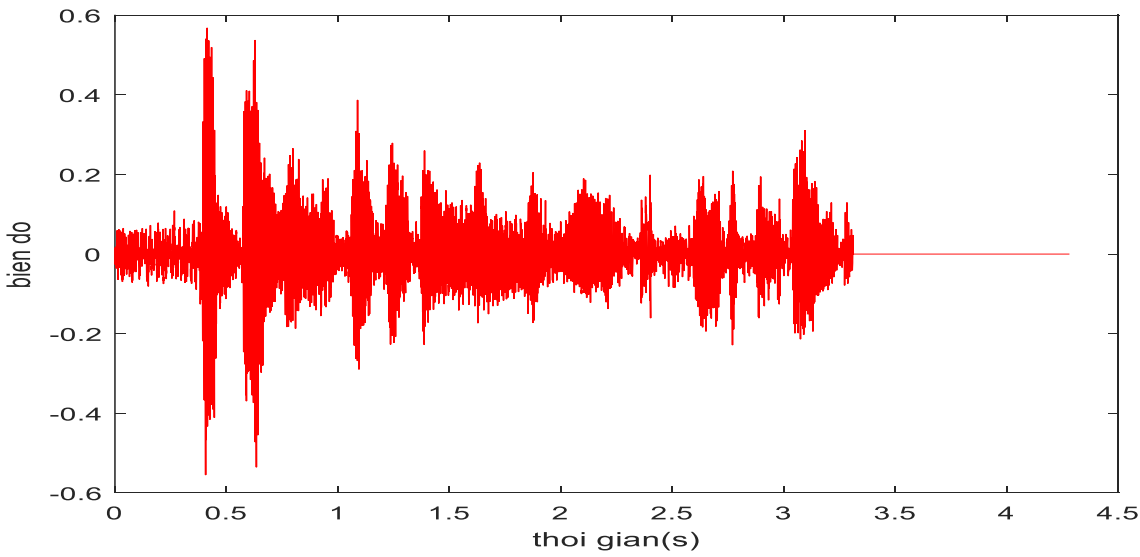
Hình 11. Tín hiệu đã được phân đoạn của file lab\_female.wav



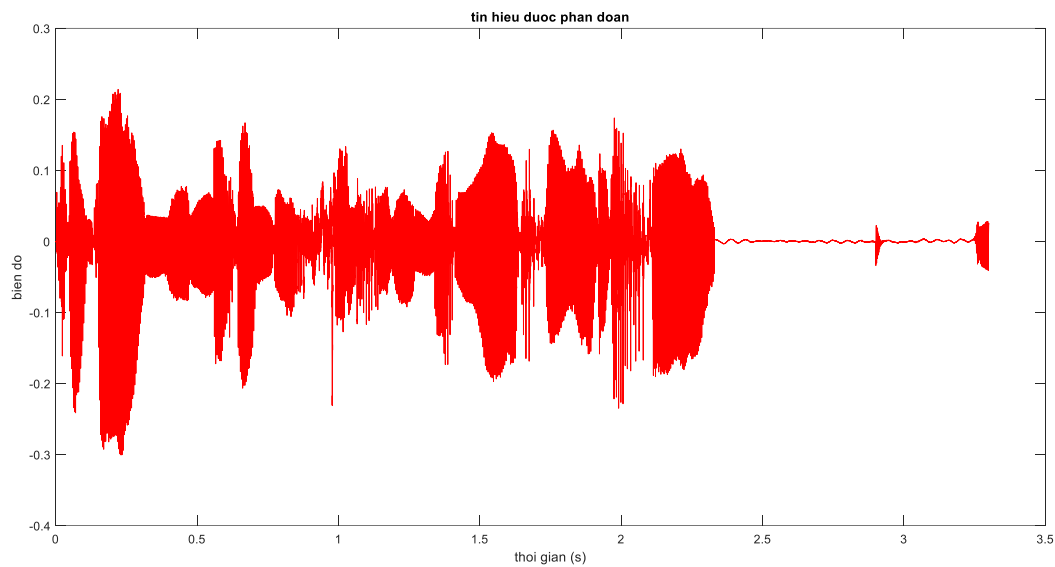
Hình 12. Tín hiệu đã được phân đoạn của file lab\_male.wav



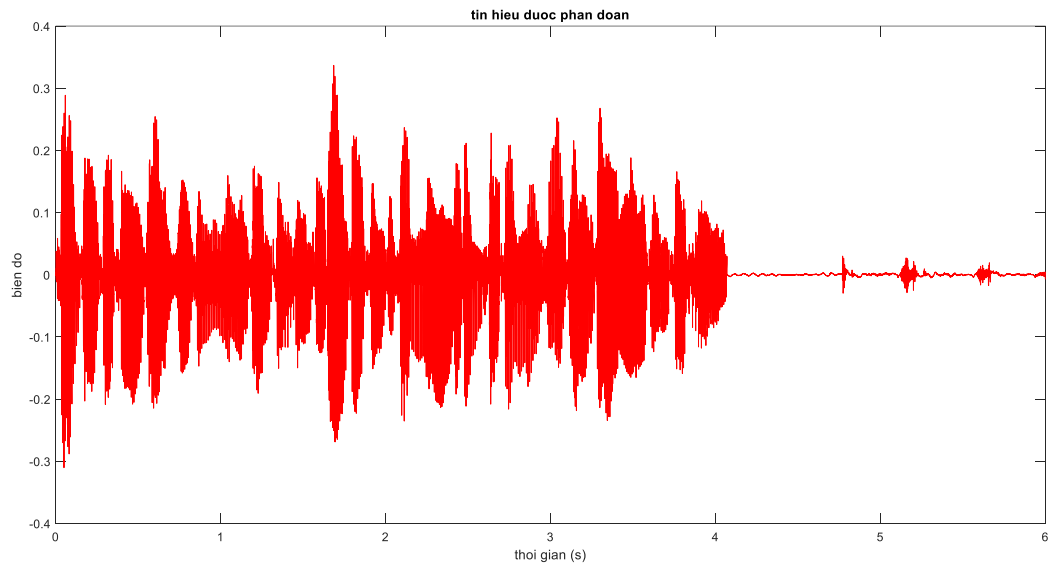
Hình 13. Tín hiệu đã được phân đoạn của file phone\_female.wav



Hình 14. Tín hiệu đã được phân đoạn của file phone\_male.wav



Hình 15. Tín hiệu đã được phân đoạn của file studio\_female.wav



Hình 16. Tín hiệu đã được phân đoạn của file studio\_male.wav

**B. Kết quả định lượng**

	Fs	Độ dài file (s)	Tổng số mẫu (mẫu)	Độ dài 1 frame(s)	Số mẫu / 1 frame (mẫu)	Tổng số frame
lab_female.wav	16000	9.5331	152529	0.01	160	953
lab_male.wav	16000	11.2491	179985	0.01	160	1124
phone_female.wav	16000	3.72	59520	0.01	160	372
phone_male.wav	16000	3.72	59520	0.01	160	372
studio_female.wav	16000	4.9136	78618	0.01	160	491

studio_male.wav	16000	8.0460	128736	0.01	160	804
-----------------	-------	--------	--------	------	-----	-----

## V. KẾT LUẬN

Bài báo cáo này thực hiện việc phân đoạn tín hiệu âm thanh, chúng em đã phân đoạn được một vài tín hiệu nhưng cũng có những trường hợp không đúng kết quả mong đợi, và vẫn chưa xác định được các đoạn nhiễu có trong tín hiệu. Trong tương lai chúng em sẽ cải thiện thuật toán để đạt được tín hiệu mong muốn.

## VI. TÀI LIỆU THAM KHẢO

- [1] Link: [https://en.wikipedia.org/wiki/Speech\\_processing](https://en.wikipedia.org/wiki/Speech_processing)
- [2] Link: [https://en.wikipedia.org/wiki/Speech\\_segmentation](https://en.wikipedia.org/wiki/Speech_segmentation)
- [3] D.S.Shete, Prof. S.B. Patil, Prof. S.B. Patil, “Zero crossing rate and Energy of the Speech Signal of Devanagari Script”, Volume 4, Issue 1, Ver. I (Jan. 2014), PP 01-05
- [4] <https://www.youtube.com/watch?v=wpXtsRpaLOA&t=185s>