

TẦN SỐ CƠ BẢN TRÊN MIỀN THỜI GIAN
Võ Đức Hùng Sơn, Nguyễn Thái Quyên, Nguyễn Duy Nhân, Nguyễn Phú Tân
Nhóm 19, lớp HP : 16.11

Điểm	Bảng phân chia nhiệm vụ		Chữ ký của SV
	Võ Đức Hùng Sơn	Cài đặt thuật toán tính F0 trên miền thời gian bằng hàm tự tương quan.	
	Nguyễn Thái Quyên	Viết báo cáo.	
	Nguyễn Duy Nhân	Cài đặt thuật toán lọc trung vị.	
	Nguyễn Phú Tân	Cài đặt thuật toán tính F0 trên miền tần số.	

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—Báo cáo này báo cáo lại quá trình thực hiện việc cài đặt thuật toán tìm tần số cơ bản của tín hiệu âm thanh với những đoạn ghi âm nguyên âm chính của mỗi sinh viên, trên miền thời gian và trên miền tần số. Sử dụng hàm tự tương quan để cài đặt thuật toán tìm F0 trên miền thời gian. Để làm trơn kết quả tính F0, và thu được F0 mong muốn là giới hạn tần số đồng thời sử dụng lọc trung vị để thu kết quả F0 tốt nhất. Còn với thuật toán tìm tần số cơ bản F0 trên miền tần số bằng cách biến đổi Fourier. Để làm tăng độ chính xác của thuật toán cần lựa chọn tín hiệu tốt không bị nhiễu. Khi tín hiệu đưa vào ổn định thì việc tính F0 sẽ chính xác hơn và sai số cũng nhỏ hơn

Từ khóa: Tần số cơ bản, chu kì, hàm tự tương quan, lọc trung vị, tín hiệu tiếng nói.

Mục lục

A. ĐẶT VẤN ĐỀ.....1

B. TÍNH FO TRÊN MIỀN THỜI GIAN

 I. CƠ SỞ LÝ THUYẾT.....4

 II. CÀI ĐẶT THUẬT TOÁN.....8

 III. KẾT QUẢ THỰC NGHIỆM.....12

C. TÍNH FO TRÊN MIỀN TẦN SỐ

 I. CƠ SỞ LÝ THUYẾT.....14

 II. CÀI ĐẶT THUẬT TOÁN.....17

 III. KẾT QUẢ THỰC NGHIỆM.....19

D. KẾT LUẬN.....21

E. NHỮNG ĐIỀU ĐÃ HỌC ĐƯỢC.....21

F. TÀI LIỆU THAM KHẢO.....21

A. ĐẶT VẤN ĐỀ

Tiếng nói mà con người giao tiếp hằng ngày có bản chất là sóng âm lan truyền trong không khí, hay gọi cách khoa học là tín hiệu âm thanh tiếng nói là tín hiệu biến thiên liên tục về thời gian. Mỗi một tín hiệu tiếng nói có một dải tần riêng, nó trải rộng từ 20 đến 20000Hz. Và trong khoảng 300 đến 3400Hz là ta có thể nghe rõ nhất. Mỗi một âm do con người phát ra nó mang một tần số cơ bản riêng, đặc trưng cho mỗi tín hiệu. [1]

Tín hiệu tiếng nói là tiếng nói của con người dưới dạng tín hiệu. Tín hiệu tiếng nói thường được thể hiện dưới dạng số, tức là được "số hóa", và do đó xử lý tiếng nói có thể được coi là giao của xử lý tín hiệu số.

Vậy quá trình đi tìm tần số cơ bản của mỗi tín hiệu thực hiện ra sao. Bài báo cáo này sẽ trình bày các cơ sở lý thuyết, các thuật toán, thực nghiệm để tìm F_0 .

Trước hết ta tìm hiểu một số khái niệm để phục vụ cho việc cài đặt thuật toán.

Tần số cơ bản (còn gọi là F_0 hoặc cao độ) của một tín hiệu tuần hoàn bằng nghịch đảo của chu kỳ tín hiệu đó [1], nó đặc trưng cơ bản cho 1 tính chất của tín hiệu, tùy theo mỗi tín hiệu mà có tính chất khác nhau.

Chu kỳ được xác định bằng khoảng thời gian ngắn nhất mà tín hiệu lặp lại trên miền thời gian. [1]

Thuật toán tìm cao độ của tín hiệu dùng hàm tự tương quan là một trong những thuật toán thông dụng, sử dụng miền thời gian để tìm ra chu kỳ của tín hiệu [1]. Thuật toán này tìm giá trị lớn nhất của hàm tự tương quan trên một mẫu tín hiệu xác định. Thuật toán này tìm giá trị lớn nhất của hàm tự tương quan trên một mẫu tín hiệu xác định.

Lọc trung vị là một kỹ thuật lọc số phi tuyến, thường được sử dụng để loại bỏ nhiễu khỏi hình ảnh hoặc tín hiệu. Giảm tiếng ồn như vậy là một bước xử lý trước điển hình để cải thiện kết quả xử lý sau này. [1]

Thuật toán tìm tần số cơ bản trên miền tần số dùng biến đổi Fourier [2] là phép biến đổi một hàm số hoặc một tín hiệu theo miền thời gian sang miền tần số. Biến đổi Fourier thường được nghĩ đến như sự chuyển đổi tín hiệu thành các thành phần biên độ và tần số.

Trong phân tích tín hiệu tiếng nói, thay vì sử dụng trực tiếp tín hiệu tiếng nói trong miền thời gian, người ta thường hay sử dụng các đặc trưng phổ của tiếng nói. Phổ biên độ [3] của tín hiệu tiếng nói được định nghĩa là biến đổi Fourier của khung tín hiệu với khoảng thời gian phân tích cố định.

Bài báo có bố cục như sau. Phần A trình bày phương pháp tìm F_0 trên miền thời gian, Phần B trình bày phương pháp tìm F_0 trên miền tần số. Trong mỗi phần A, B sẽ trình bày về tổng quan về cơ sở lý thuyết, sơ đồ thuật toán (kể cả những vấn đề phát sinh và đề xuất các giải pháp khắc phục), cụ thể các thuật toán và kết quả thực nghiệm. Phần C trình bày bài học đạt được. Cuối cùng là phần tài liệu tham khảo dùng trong báo cáo này

A. TÍNH FO TRÊN MIỀN THỜI GIAN:

I. CƠ SỞ LÝ THUYẾT XỬ LÝ TÍN HIỆU TIẾNG NÓI VÀ CÁC THUẬT TOÁN:

1. Cơ sở lý thuyết:

Hàm tự tương quan của tín hiệu $x(n)$

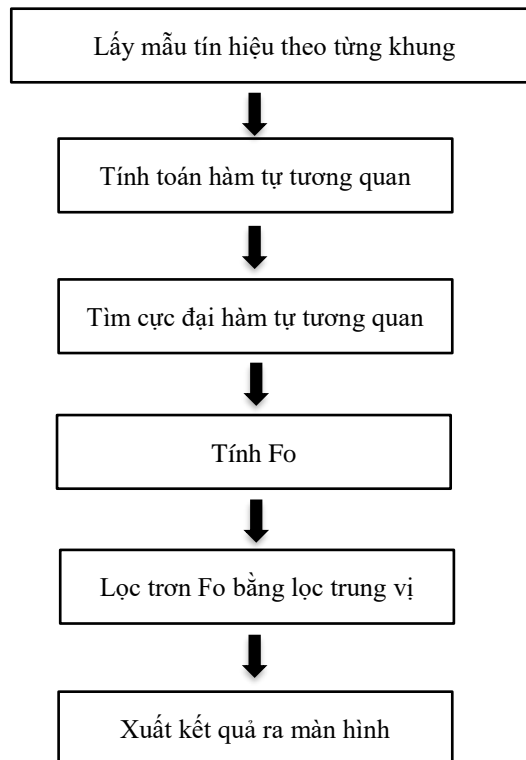
$$r(k) = \sum_{m=-\infty}^{\infty} x(n).x(n+k) \quad [4]$$

- Một số tính chất của hàm tự tương quan:

* $r(k)$ là hàm chẵn

* $r(k)$ đạt cực đại tại những điểm tương ứng là bội số của chu kì cơ bản của tín hiệu. [5]

2. Sơ đồ thuật toán:

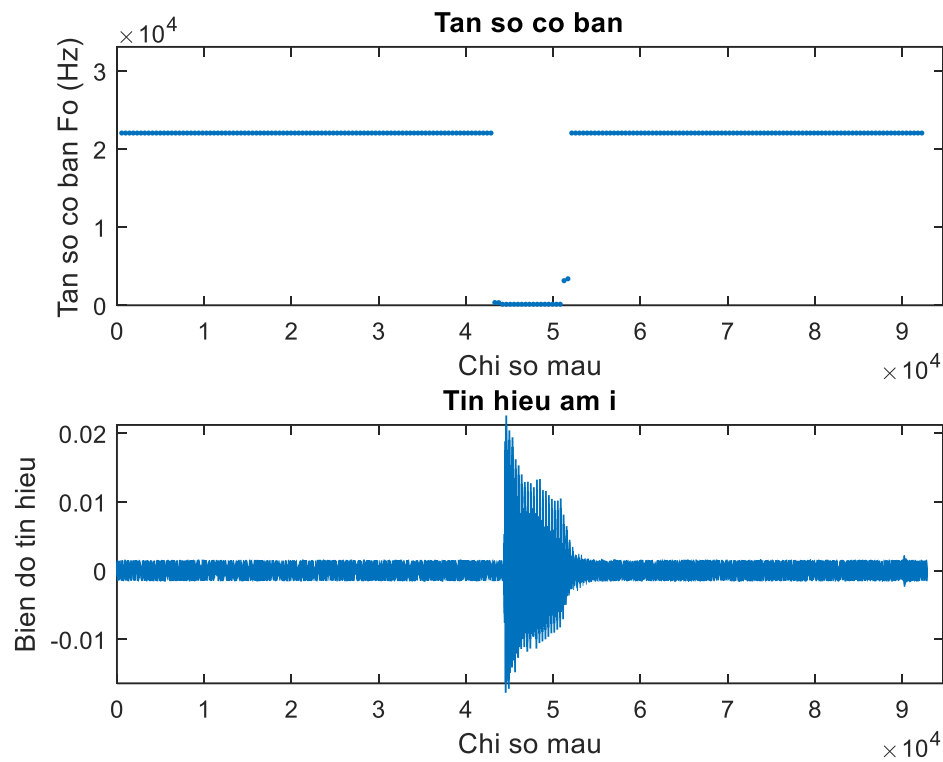


3. Các vấn đề phát sinh và hướng khắc phục:

3.1. Xuất hiện nhiều giá trị tần số F0 rất không chính xác:

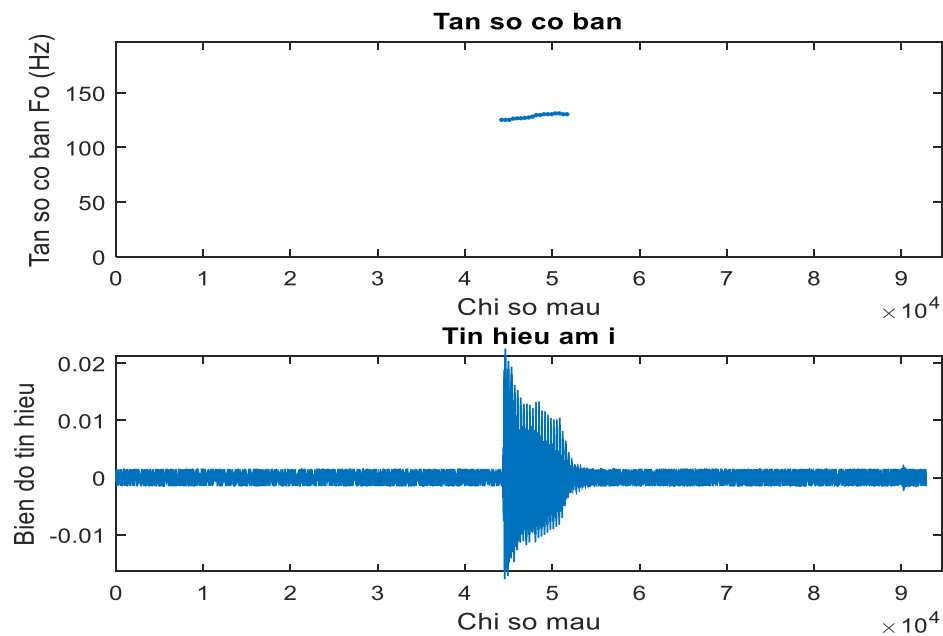
3.1.1. Vấn đề :

- Xuất hiện những giá trị Fo rất không chính xác, có biên độ rất cao



Hình 1. Kết quả xuất hiện nhiều F0 rất không chính xác.

3.1.2. Hướng khắc phục:
- Giới hạn tần số F0, vì tần số cơ bản của con người dao động từ 80Hz đến 400Hz [3] nên ta chỉ chấp nhận những tần số tính được trong đoạn giới hạn trên .

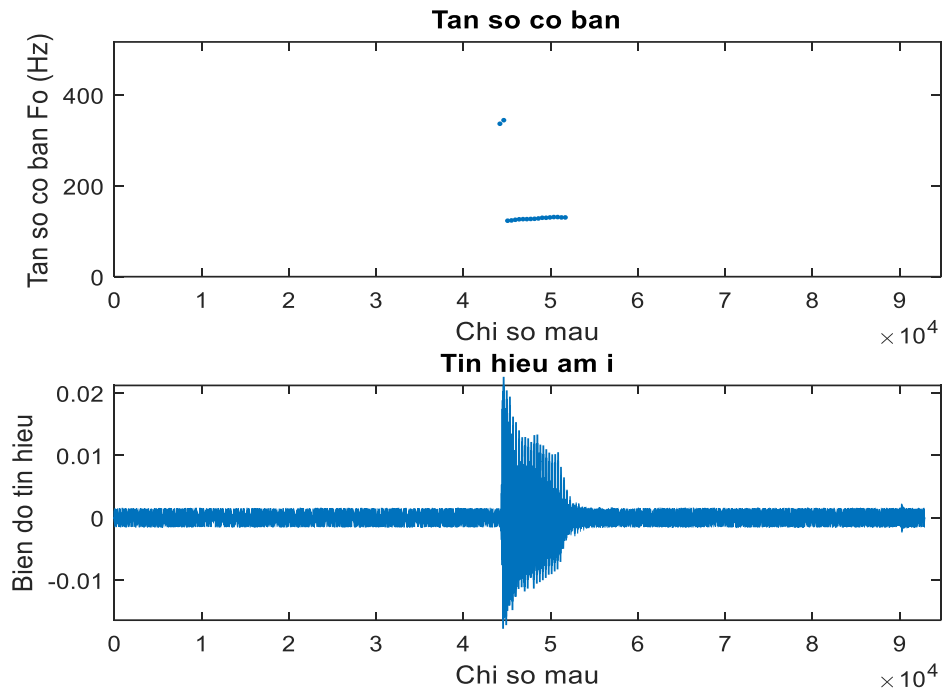


Hình 2. Kết quả F0 sau khi đã giới hạn.

3.2. Giá trị đột biến của F0 (vẫn nằm trong miền giới hạn):

3.2.1. Vấn đề :

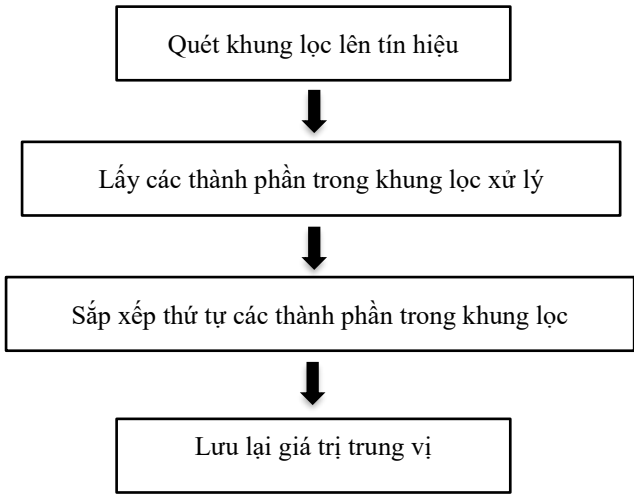
- Kết quả tính ra chứa một vài giá trị F0 cá biệt (quá cao hay thấp so với các giá trị bình thường nhưng vẫn nằm trong miền giới hạn)



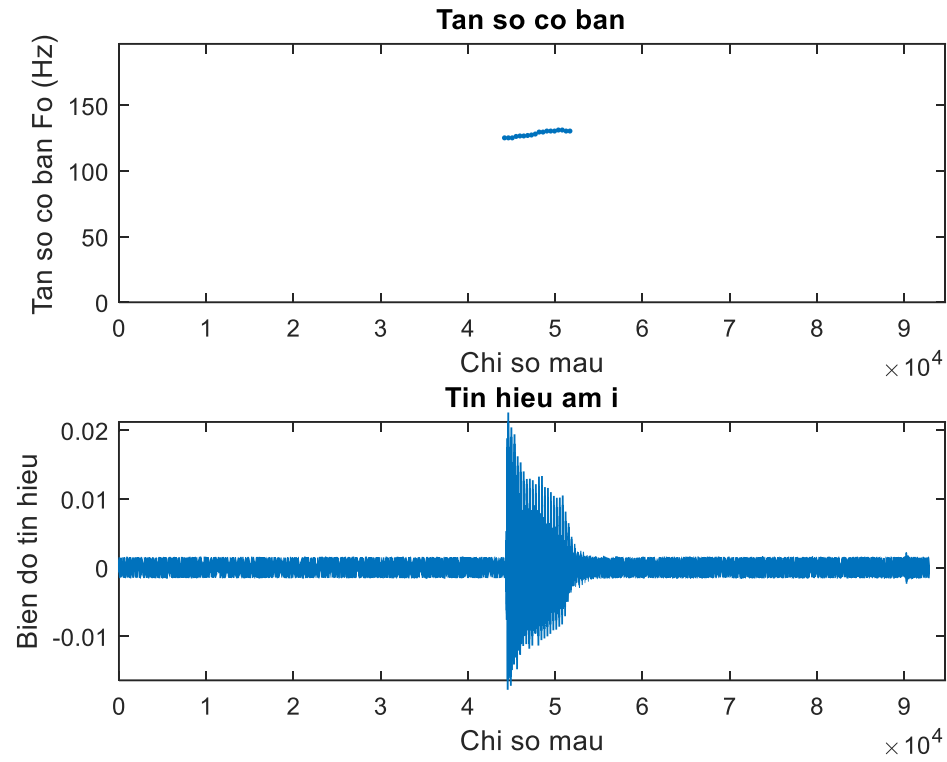
Hình 3. Kết quả F0 khi chưa lọc những đột biến tần số.

3.2.2. Hướng khắc phục:

- Dùng lọc trung vị để loại bỏ các giá trị F_0 tăng, giảm đột biến.
- a. Cơ sở lý thuyết:
- Ta sử dụng một khung lọc (kích thước 3x1) quét qua lần lượt từng điểm tín hiệu của tín hiệu đầu vào.
 - Tại vị trí mỗi điểm tín hiệu lấy giá trị của các điểm tín hiệu tương ứng trong vùng 3x1 của tín hiệu gốc “lấp” vào ma trận lọc.
 - Sắp xếp các điểm tín hiệu trong cửa sổ này theo thứ tự (tăng hoặc giảm dần tùy ý).
 - Cuối cùng gán điểm tín hiệu nằm chính giữa (trung vị) cho giá trị điểm tín hiệu đang xét của tín hiệu ra.
- b. Sơ đồ thuật toán:

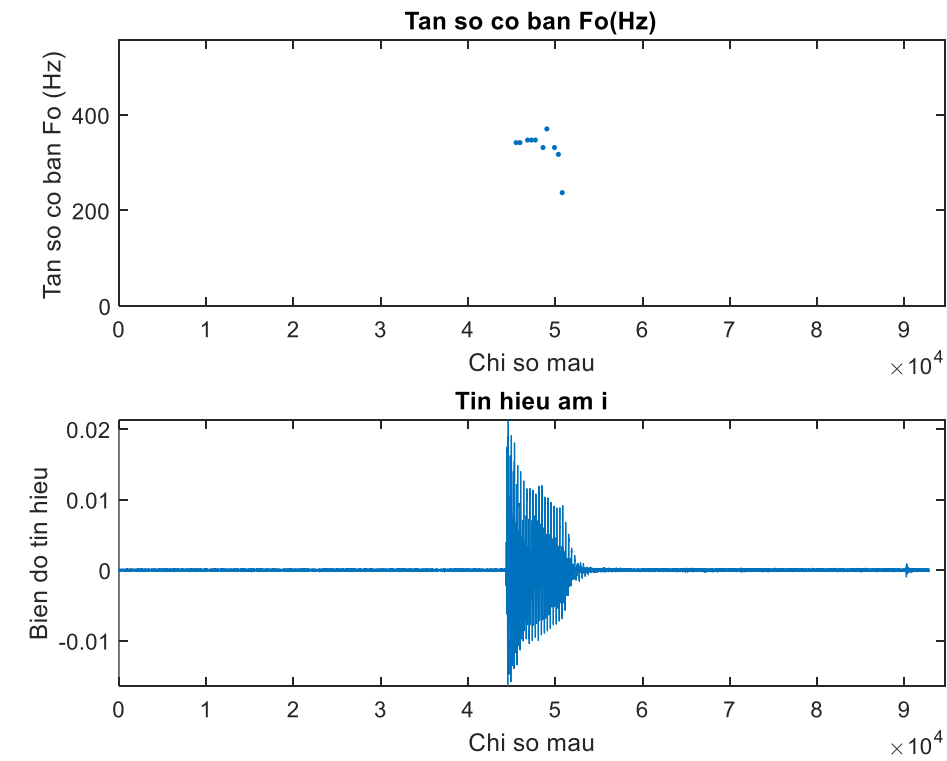


c. Kết quả



Hình 4. Kết quả F0 khi đã lọc những đột biến tần số.

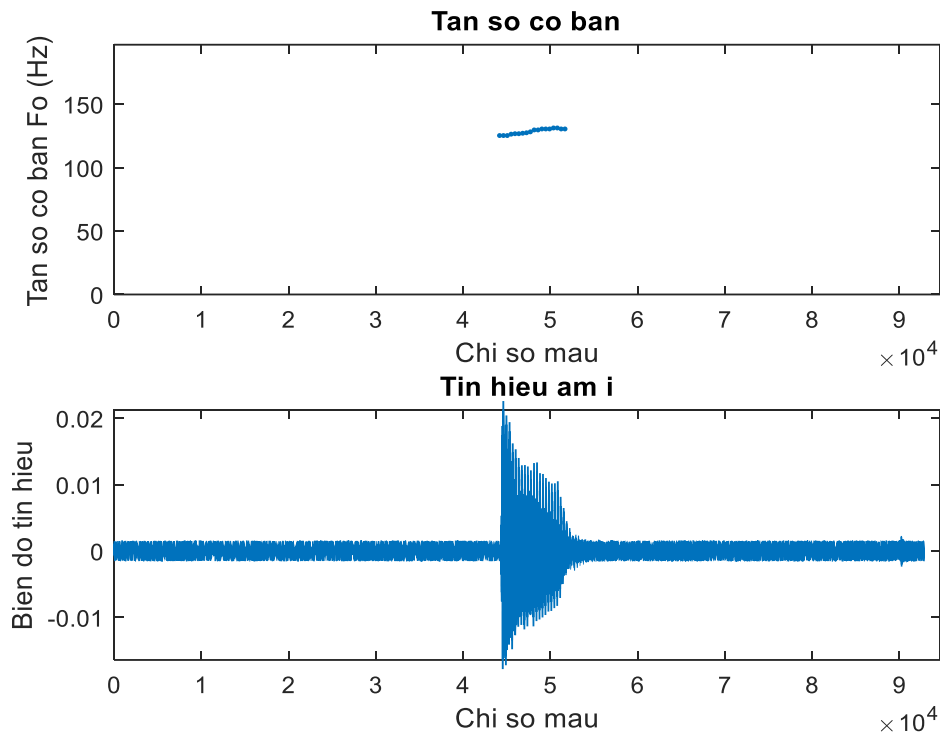
- 3.3. Chọn 2 cực đại kề nhau không chính xác
- 3.3.1. Vấn đề : Hai cực đại được chọn để tính độ trễ không chính xác, trùng cực đại của nhiều



Hình 5. Kết quả F0 khi chọn sai cực đại

3.3.2. Hướng khắc phục:

Ta sẽ chọn 2 cực đại gồm: max của hàm tự tương quan và 1 cực đại có giá lớn nhất và kề với max chính nhất, khi đó sẽ không chọn nhầm cực đại của nhiễu, độ trễ cần tìm được chính xác.



Hình 6. Kết quả F0 khi chọn đúng cực đại

II. CÀI ĐẶT THUẬT TOÁN:

```
clear();

% ***** LẤY TÍN HIỆU
[a, fs] = audioread('./am/son/i.wav'); % đọc file ghi âm

% ***** MỘT SỐ HẲNG SỐ
a_len = length(a); % chiều dài tín hiệu
frame_len = floor(0.025 * fs); % chiều dài mỗi khung, mỗi khung dài 25ms (thời gian mà đặc tính giọng nói không đổi)
number_frame = floor((a_len-frame_len)/(0.4*frame_len))+1; % số lượng khung, overlap 60% (15ms so với 25ms)
index = 1; % chỉ số của mảng lưu fo từng khung
number_filter = 5; % số lượng mẫu trong 1 khung lọc
max_period = floor(fs/80); % max số lượng mẫu trong 1 chu kỳ cơ bản
min_period = floor(fs/400); % min số lượng mẫu trong 1 chu kỳ cơ bản

% ***** VẼ TÍN HIỆU VÀO
subplot(2,1,2);
plot(a);
title('Tín hiệu âm i');
xlabel('Chỉ số mẫu');
ylabel('Biên độ tín hiệu');
axis([0 a_len*1.02 -inf inf]);

% ***** DUYỆT VÀ TÍNH FO TRONG TỪNG KHUNG
for k = 1 : number_frame
    % lấy mẫu tín hiệu theo từng khung
```

```

        frame = getFrame(k, a, frame_len);
    % tính toán hàm tự tương quan
        fr_xcorr = xcorr(frame, frame);
    % tìm cực đại hàm tự tương quan và tính fo
        [fo, loc] = getFo(k, frame_len, fr_xcorr, min_period, max_period, fs);
    % lưu fo vào mảng kết quả
        if(fo ~= 0) % kiểm có tồn tại fo
            fo_vecto(index) = fo; % lưu Fo của khung thứ k vào mảng kết quả
            loc_vecto(index) = loc; % lưu chỉ số mẫu chính giữa của khung
            index = index + 1; % tăng chỉ số của mảng kết quả
        end
    end

end

% ***** LỌC VÀ XUẤT KẾT QUẢ
if(fo_vecto ~= 0) % kiểm tra mảng fo_vecto có tồn tại
    subplot(2,1,1);
    % làm trơn mảng kết quả fo bằng lọc trung vị
        fo_vecto = MyFunctionFilter(fo_vecto, number_filter);
    % xuất kết quả ra màn hình
        plot(loc_vecto, fo_vecto, '.');
        title('tần số cơ bản');
        axis([0 a_len*1.02 0 max(fo_vecto)*1.5]);
        xlabel('chỉ số mẫu');
        ylabel('tần số cơ bản fo (Hz)');
        fo_mean = mean(fo_vecto);
end

```

TRONG ĐÓ:

- ***getFrame(k, a, frame_len)***: hàm lấy khung thứ *k* với độ dài là *frame_len* của tín hiệu *a*

```

function [frame] = getFrame(k, a, frame_len)
    %***** Tham số
        % k là vị trí khung cần lấy
        % a là tín hiệu vào
        % frame_len là chiều dài mỗi khung
    %***** Kết quả trả về
        % frame: khung thứ k của tín hiệu
    %***** Code
        % xác định vị trí khung k, nạp chồng 60% (15ms so với 25ms)
            range = floor((k - 1)*0.4*frame_len) + 1 : floor((k - 1)*0.4*frame_len) + frame_len;
        % lấy khung thứ k
            frame = a(range);
end

```

- ***getFo(k, frame_len, fr_xcorr, min_period, max_period, fs)***: hàm tính *Fo* của khung thứ *k*

```

function [fo, loc] = getFo(k, frame_len, fr_xcorr, min_period, max_period, fs)
    %***** Tham số
        %k là vị trí của khung
        %frame_len là chiều dài mỗi khung
        %fr_xcorr là hàm tự tương quan của khung thứ k
        %min_period là số lượng mẫu nhỏ nhất trong 1 chu kì cơ bản
        %max_period là số lượng mẫu lớn nhất trong 1 chu kì cơ bản

```

```

%***** Kết quả trả về
fo = 0; % tần số cơ bản của khung thứ k
loc = 0; % chỉ số mẫu chính giữa của khung thứ k

%***** Code
% Tìm cực đại cục bộ hàm tự tương quan
[pks,lag] = findpeaks(fr_xcorr );

% Tính Fo
if(~isempty(pks)) %kiểm tra hàm tự tương quan có cực đại không
    % Tìm độ trễ của 2 cực đại kề nhau của hàm tự tương quan
    % Hai cực đại gồm: max hàm tự tương quan và cực đại lớn nhất và kề max nhất
    first_loc_max = find(pks == max(pks)); % tìm chỉ số max hàm tự tương quan
    if(first_loc_max > 1) % kiểm tra chỉ số của max có lớn hơn 1
        second_loc_max = 1; % chỉ số của max nửa đầu (kề max chính nhất) hàm tự tương quan
        second_max = pks(first_loc_max - 1); % giá trị max nửa đầu hàm tự tương quan
    chính nhất
    % Tìm max và chỉ số max của nửa đầu hàm tự tương quan
    for i = 1 : first_loc_max - 1
        if(pks(i) >= second_max)
            second_loc_max = i;
            second_max = pks(i);
        end
    end
    % Tính độ trễ của 2 cực đại đã tìm
    delta = lag(first_loc_max) - lag(second_loc_max);

    % ***** Tính fo
    %kiểm tra delta có thỏa điều kiện max, min số lượng mẫu trong 1 chu kì

    if(delta >= min_period && delta <= max_period)
        fo = fs/delta;
        loc = floor((k-1)*0.4*frame_len + frame_len/2);
    end
end
end
end
end

```

- ***MyFunctionFilter(fo_vecto, number_filter): loại bỏ những giá trị Fo tăng, giảm đột biến trong mảng kết quả***

```

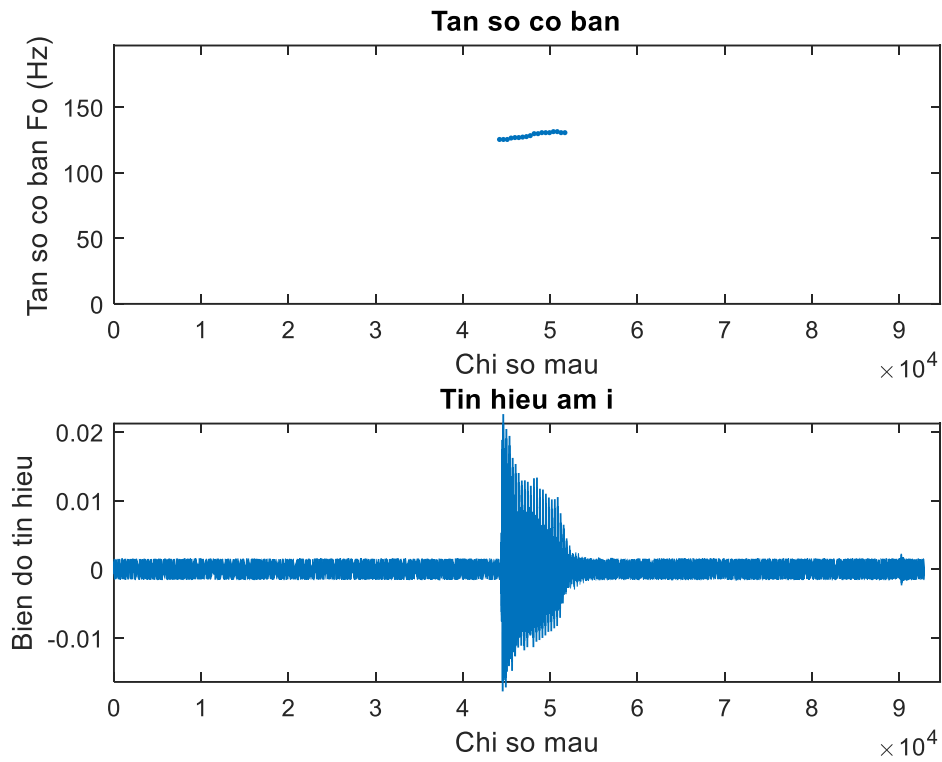
function y = FunctionMedianFilter(y,m) %y tín hiệu cần lọc, m là kích thước của khung lọc
khung = 1:m;
for j = 1:length(y)+1-m % quét khung lọc lên tín hiệu vào
    for i = 1:m % gán các điểm giá trị của tín hiệu vào khung lọc
        khung(i) = y(j-1+i);
    end
    khung = sort(khung); % sắp xếp các điểm giá trị trong khung lọc
    y(j) = khung((length(khung)+1)/2); % gán giá trị trung vị cho tín hiệu ra
end
end
end

```

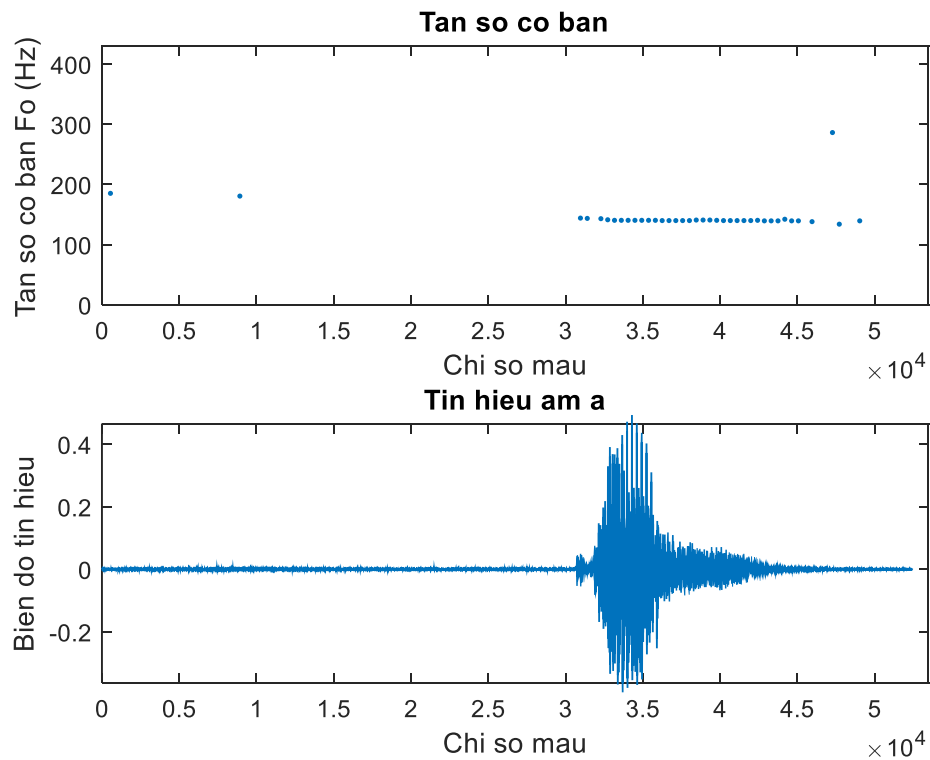
-
- *$xcorr(x,x)$: tìm hàm tự tương quan tín hiệu $x(n)$ – hàm có sẵn trong Matlab*
 - *$findpeaks(x)$: tìm cực đại của tín hiệu $x(n)$ – hàm có sẵn trong Matlab*
 - *$max(x)$: tìm giá trị max của $x(n)$ – hàm có sẵn trong Matlab*
 - *$find(x = k)$: trả về vị trí trong x mà tại đó x có giá trị bằng k – hàm có sẵn trong Matlab*

III. KẾT QUẢ THỰC NGHIỆM:

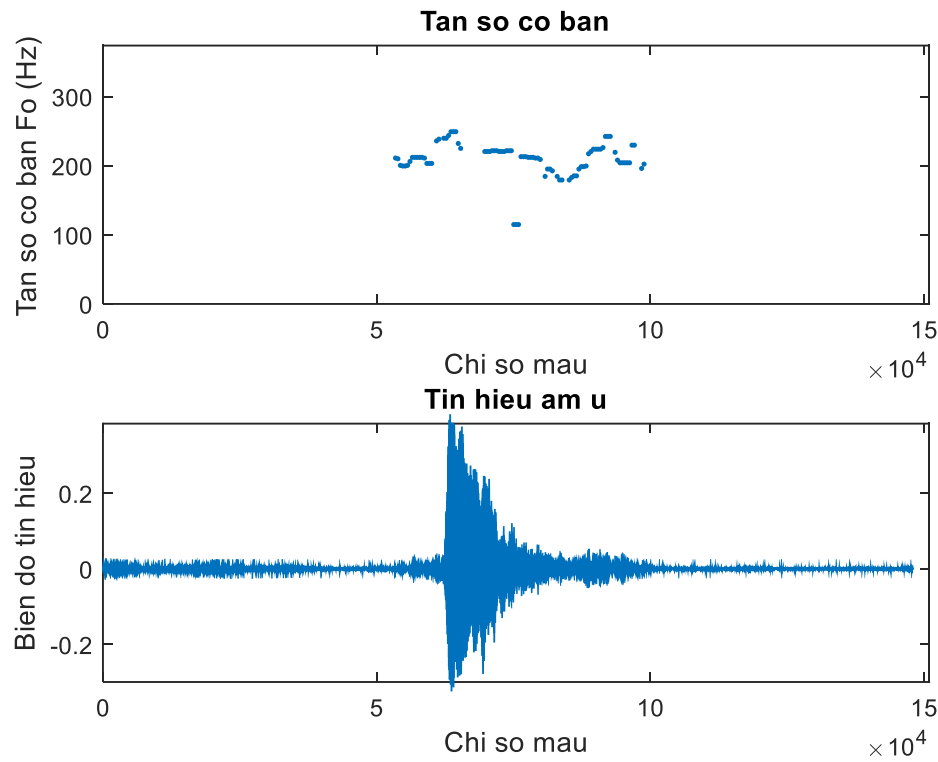
1. Hình ảnh tốt nhất, trung bình và xấu nhất khi tính F0 với thuật toán trên :



Hình 7. Kết quả tính F0 tốt nhất.



Hình 8. Kết quả tính F0 trung bình



Hình 9. Kết quả tính F0 xấu nhất.

2. Bảng so sánh giữa kết quả F0 bằng cách tính thủ công và cách tính theo thuật toán trên:

Bảng 1. Bảng so sánh giữa kết quả F0 bằng cách tính thủ công và cách tính theo thuật toán hàm tự tương quan

	a		e		i		o		u	
	Thủ công	Tự động	Thủ công	Tự động	Thủ công	Tự động	Thủ công	Tự động	Thủ công	Tự động
Son	138.88	129.44	142.85	119.46	153.84	128.51	153.15	127.25	163.9	145.48
Nhân	158.85	161.78	147.06	156.49	156.25	157.7	153.85	157.44	149.25	167.8
Quyên	235.83	227.52	230.89	252.41	233.33	251.79	234.57	223.46	252	244.8
Tân	137.7	139.89	142.2	165.13	152	185.6	147.1	150.07	158	155.49

	Fo mean		Fo std	
	Thủ công	Tự động	Thủ công	Tự động
Son	146.9	130.03	8.89	9.5
Nhân	153.05	160.24	4.87	4.69
Quyên	237.33	240	8.4	13.65
Tân	147.4	159.24	7.98	17.34

3. Các yếu tố ảnh hưởng đến kết quả tính Fo:
- Tín hiệu vào ít nhiễu thì sẽ cho kết quả tính Fo chính xác hơn
 - Độ dài mỗi khung nằm trong khoảng 25ms – 30ms sẽ cho kết quả chính xác hơn (thời gian mà được tính giọng nói không đổi)
 - Giới hạn đúng miền giá trị của Fo, để lấy những giá trị được chính xác hơn

B. TÍNH FO TRÊN MIỀN TẦN SỐ

I. CƠ SỞ LÝ THUYẾT XỬ LÝ TÍN HIỆU TIẾNG NÓI VÀ CÁC THUẬT TOÁN

1. Cơ sở lý thuyết

- a) Phép biến đổi Fourier DTFT (tín hiệu rời rạc vô hạn và không tuần hoàn)
- Biểu diễn tín hiệu rời rạc bằng cách lấy mẫu tín hiệu tương tự dưới dạng:

$$x_s(t) = \sum_{k=-\infty}^{\infty} x(kT)\delta(t-kT)$$

- Biến đổi Fourier cho tín hiệu:

+ Tính biến đổi Fourier của $\delta(t - kt)$.

+ Sử dụng nguyên lý xếp chồng, tìm biến đổi Fourier của $(x_s(t))$

$$x_s(t) \xleftrightarrow{F} \sum_{n=-\infty}^{\infty} x(nT)e^{-jn\omega T}$$

Đặt $x(nT)=x[n]$ và thay biến $\Omega = \omega T$ (lưu ý đơn vị của Ω [rad] và ω [rad/s]), ta được:

$$\text{DTFT} : X(\Omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n}$$

- Tín hiệu rời rạc trong miền thời gian nhưng DTFT lại liên tục và tuần hoàn trong miền tần số. DTFT chính là hàm phức theo biến tần số thực. Ta gọi DTFT là phổ của tín hiệu rời rạc.
- Phổ của tín hiệu gồm có hai phần: phổ biên độ (magnitude spectrum) và phổ pha (phase spectrum).

$$X(\Omega) = |X(\Omega)|e^{j\theta(\Omega)}$$

Phổ biên độ: $|X(\Omega)|$

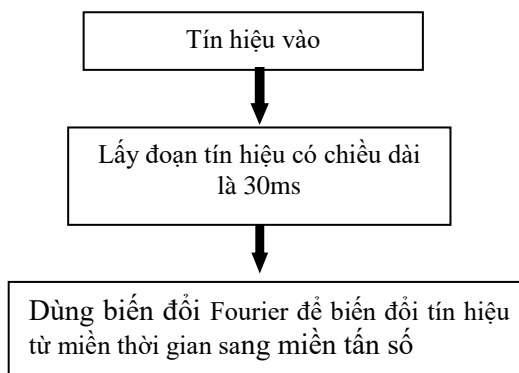
Phổ pha: $\theta(\Omega)$.

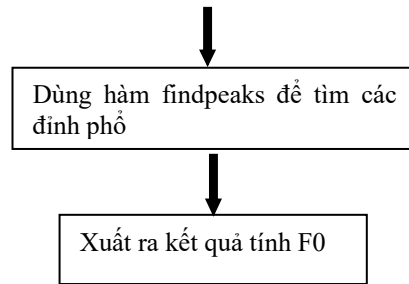
- Trường hợp dãy tín hiệu hữu hạn : dùng phép biến đổi Fourier rời rạc DFT.
- Các phép tính chuỗi Fourier được thực hiện bằng cách sử dụng phép biến đổi Fourier rời rạc (DFT), do đó được thực hiện bằng cách sử dụng thuật toán hiệu quả được gọi là chuyển đổi Fourier(FFT).
- b) Cửa sổ hamming (kiểm tra kết quả tính được Fo trên miền tần số):
 - Biểu diễn tín hiệu rời rạc bằng cách lấy mẫu tín hiệu tương tự dưới dạng:

$$w(n) = 0.52 - 0.46\cos(2\pi\frac{n}{N}) \quad [2]$$
 - Làm ngắn đoạn tín hiệu, trả về tần số gần đúng của tín hiệu

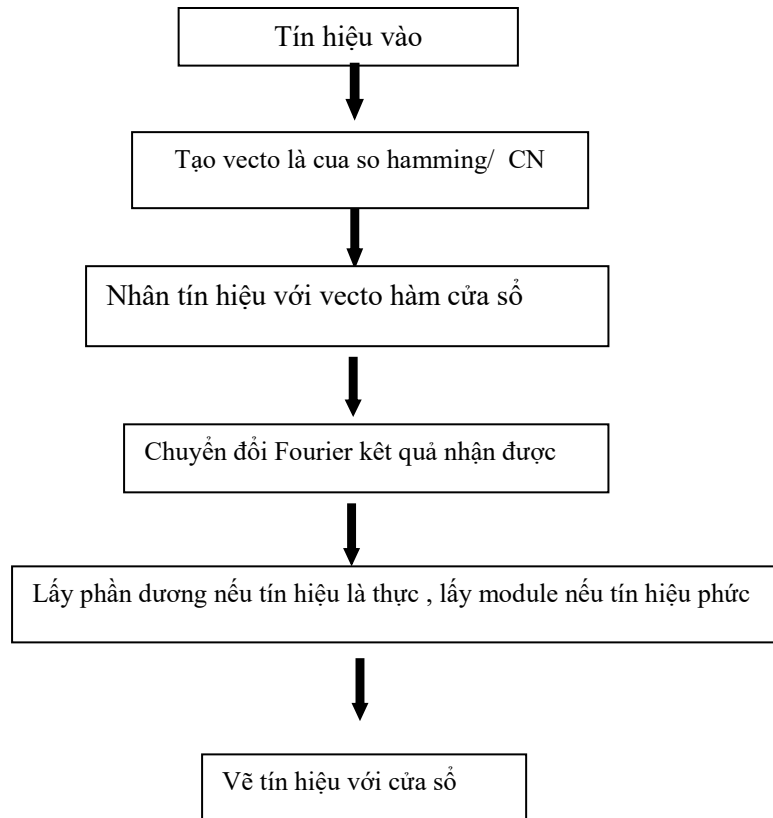
2. Sơ đồ thuật toán :

2.2 Biến đổi Fourier:





2.2. Hàm sửa số kiểm tra kết quả:

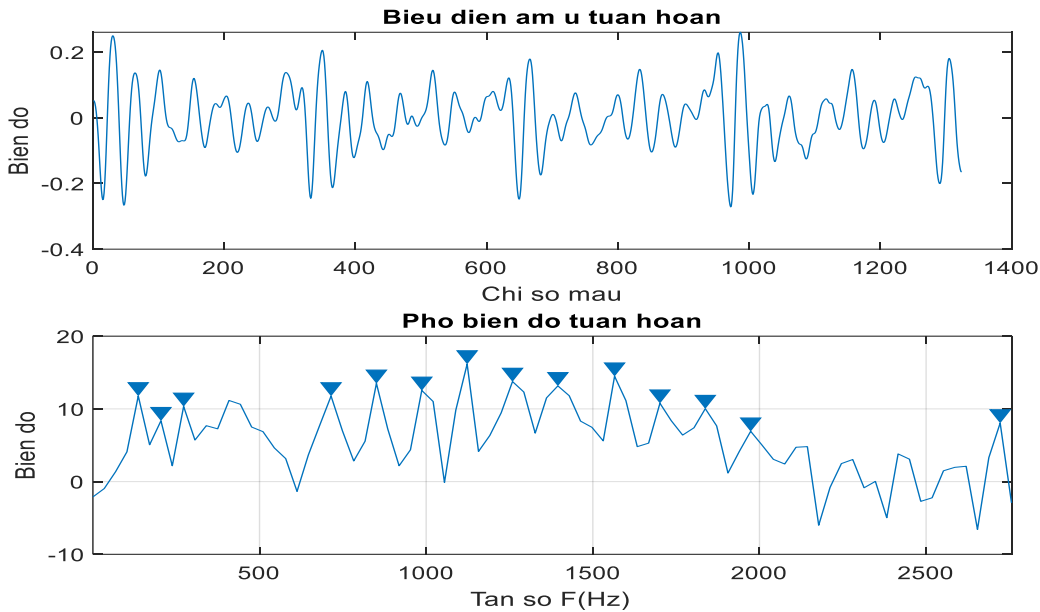


3. Các vấn đề phát sinh và hướng giải quyết

3.1. Vấn đề:

Hàm findpeaks tìm số đỉnh của phổ không chính xác tuyệt đối, từ đó dẫn đến sai số trong quá trình tính tần số cơ bản F0.

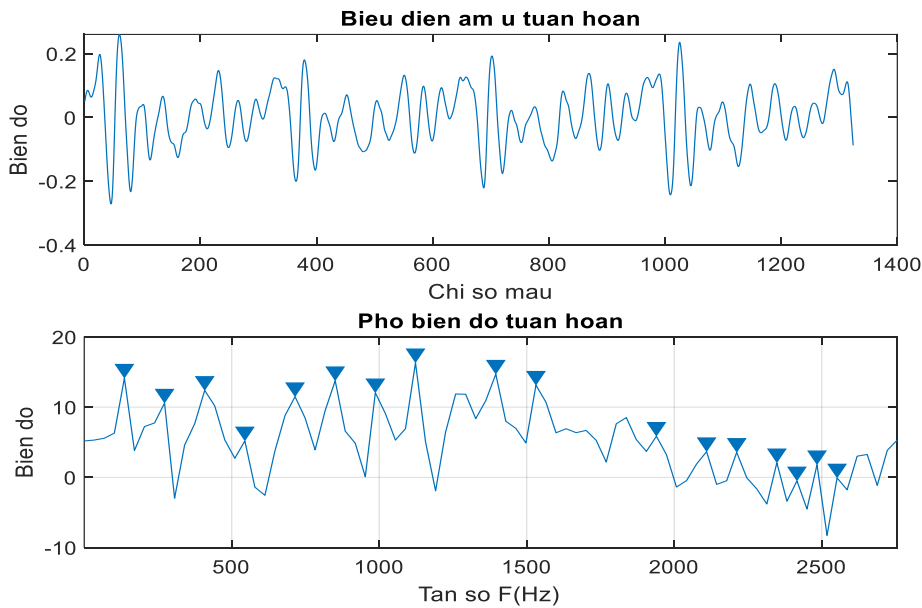
Ví dụ minh họa



Hình 10. Hình ảnh tính Fo khi lọc tín hiệu

3.2. Hướng giải quyết:

Do quá trình thu âm tín hiệu tiếng nói bị nhiễu hay lẫn nhiều hợp âm nên để xác định các đỉnh phổ một cách chính xác thì phải lọc và loại bỏ các tín hiệu dư thừa.



Hình 11. Hình ảnh tính Fo khi đã lọc tín hiệu

II. CÀI ĐẶT CÁC THUẬT TOÁN

```
[y1,Fs1]=audioread('a.wav');
y2 = y1(0.401*Fs1:0.431*Fs1);
subplot(321);
plot(y2);
title('Bieu dien am a tuan hoan');
xlabel('Chi so mau');
ylabel('Bien do');

subplot(323);
[dftylg1, tt1] = convertFrquency(y2, Fs1);
plot(tt1,dftylg1);

%tim so dinh
[pks1,locs1] = findpeaks(dftylg1,tt1,'Threshold',1);%tim cac dinh cua tin hieu va vi tri cua chung
findpeaks(dftylg1,tt1,'Threshold',1);%ve cac dinh cua tin hieu
title('Pho bien do tuan hoan');
xlabel('Tan so F(Hz)');
ylabel('Bien do');
do=length(locs1) ;    %so dinh

%tim do chenh lech giua hai dinh
for i=2:1:do
    f1(i,1)= locs1(i)- locs1(i-1);
end
F1=sum(f1)/(length(f1)-1);    %tinh F0

subplot(322);
[y1,Fs1]=audioread('a.wav');
y2 = y1(3000:4323);
plot(y2);
title('Bieu dien am a khong tuan hoan');
xlabel('Chi so mau');
ylabel('Bien do');

subplot(324);
[dftylg2, tt2] = convertFrquency(y2, Fs1);
plot(tt2,dftylg2);

%tim so dinh
[pks2,locs2] = findpeaks(dftylg2,tt2,'Threshold',1);%tim cac dinh cua tin hieu va vi tri cua chung
findpeaks(dftylg2,tt2,'Threshold',1);%ve cac dinh cua tin hieu
title('Pho bien do khong tuan hoan');
xlabel('Tan so F(Hz)');
ylabel('Bien do');
do=length(locs2) ;    %so dinh
%tim do chenh lech giua hai dinh
for i=2:1:do
    f2(i,1)= locs2(i)- locs2(i-1);
end
F2=sum(f2)/(length(f2)-1);    %tinh F0

% kiem tra tan so
N= 1024;
subplot(325);
[xabs] = check_hamm(y2,N);
plot(( xabs((1:N)))));
axis([0 1200 0 inf ]);
xlabel('Tan so (Hz)');
ylabel('Bien do |x|');
title('Check with Hamming Window N = 1024');
```

*Các hàm được sử dụng trong bài

a. Hàm chuyển đổi

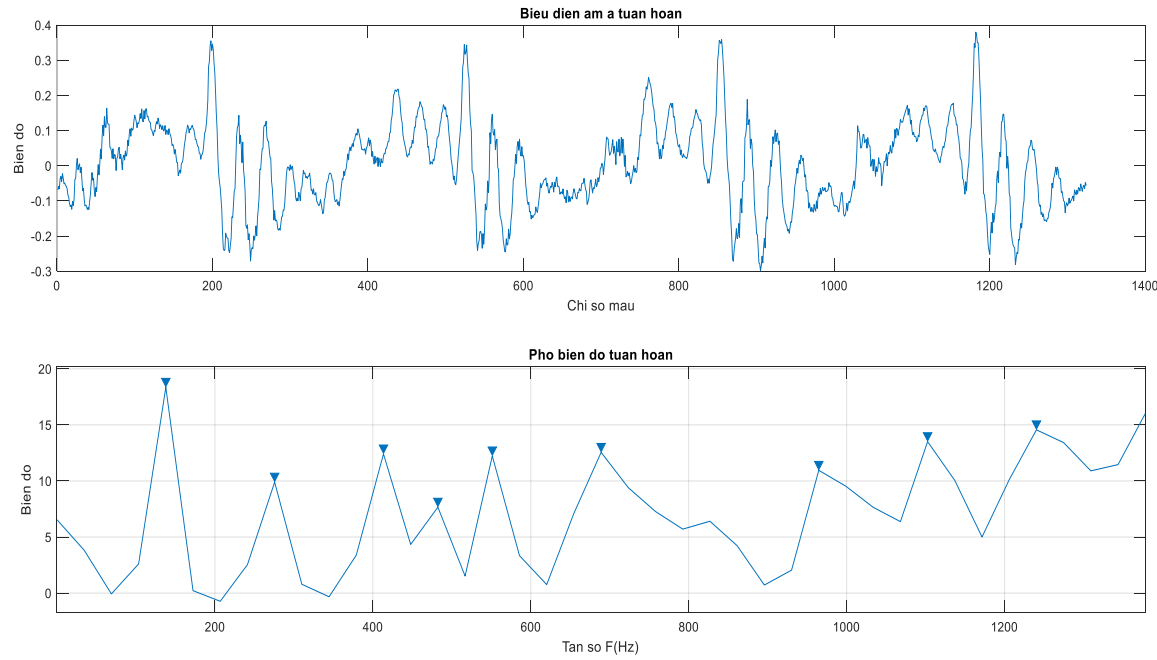
```
% đưa tín hiệu vào và tần số tín hiệu
% do phân gia của Fs1 Hz thì lấy mẫu 1 lần
% lấy phần dương nếu là thực, lấy module nếu phức, đồng thời biến đổi fourier dft1=abs(fft(signal));
% tạo length(dft1) điểm, khoảng cách giữa các điểm; tt1=linspace(1/Fs1,Fs1/16,length(dft1))
% dft1log1=10*log10(dft1); % chuyển sang miền logarit
function [dft1log1, tt1] = convertFrequency(signal, Fs)
Fs1 = Fs;
t1=1/Fs1:1/Fs1:(length(signal)/Fs1);
dft1=abs(fft(signal));
dft1=dft1(1:(length(dft1)/16));
tt1=linspace(1/Fs1,Fs1/16,length(dft1));
dft1log1=10*log10(dft1);
end
```

b. Hàm cửa sổ kiểm tra tần số tính được:

```
% đưa vào tín hiệu
% tạo vectơ của số hamming
% nhân tín hiệu với hàm của số hamming
% chuyển đổi fourier
% lấy phần dương nếu là thực, lấy module nếu phức
function [xabs] = check_hamm(signal, Fs, N)
x= signal; % tín hiệu vào là đoạn ghi âm giống nội
hamm = [hamming(N)' zeros(1, length(x) - N)]; % hàm hamming + padding zero để có chiều dài vectơ là 5000
mau
x_hamm = hamm.* x; % lấy đoạn tín hiệu với chiều dài khung của số function hamming
dftx = fft(x_hamm); % biến đổi Fourier của x_hamm
xabs = abs(dftx); % lấy phần dương, phần âm là âm nên không lấy,
end
```

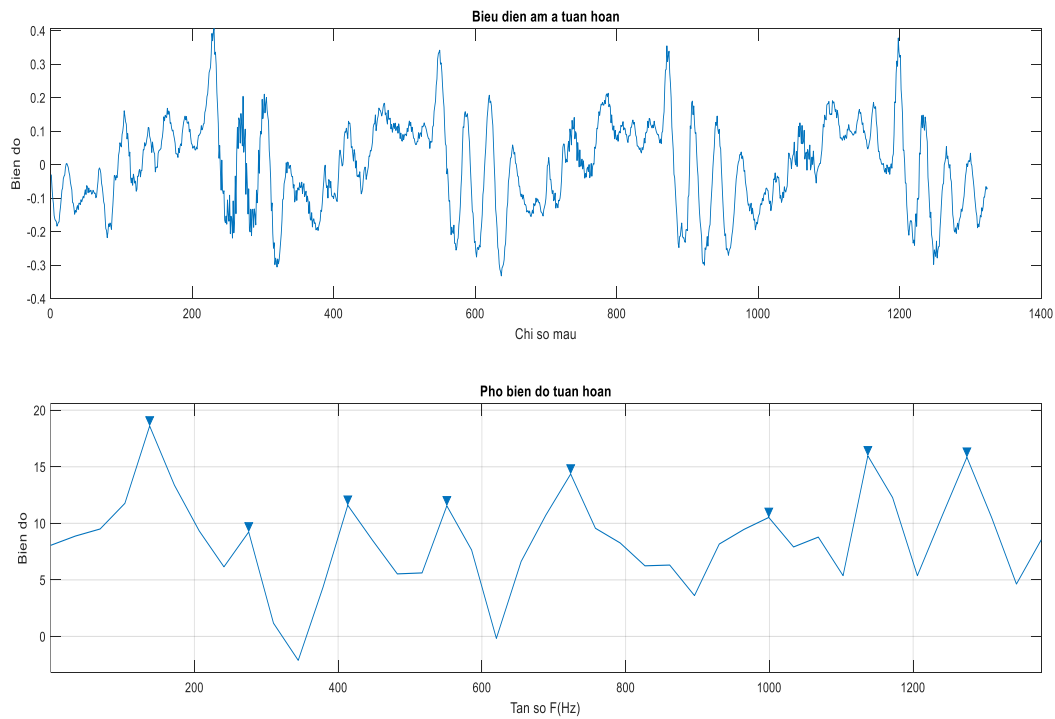
III. KẾT QUẢ THỰC NGHIỆM:
A. Trên miền thời gian:

1. Hình ảnh tốt nhất, trung bình và xấu nhất khi tính F0 với thuật toán trên :
- Trong trường hợp F0 cho kết quả tốt nhất khi chọn tín hiệu tại đoạn tuần hoàn thì $F_0 = 142 \text{ (Hz)}$



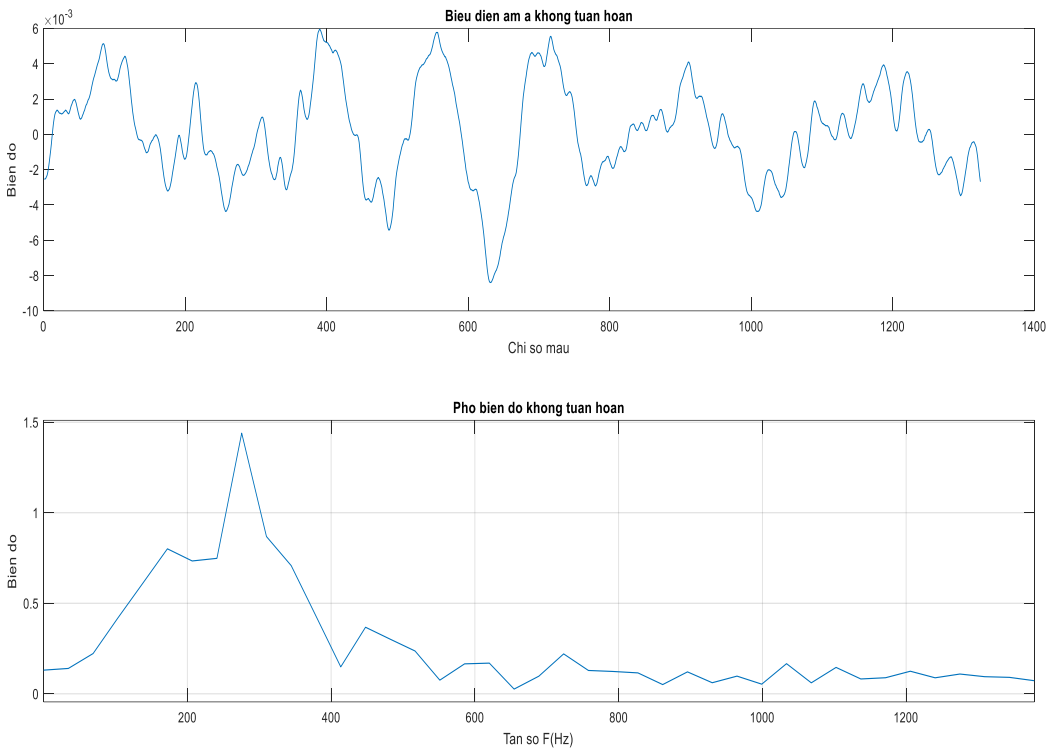
Hình 12: Trường hợp tín hiệu cho kết quả tính F0 chính xác nhất

- Trường hợp F0 cho kết quả trung bình khi chọn tín hiệu tại đoạn gần tuần hoàn thì $F_0 = 183 \text{ (Hz)}$

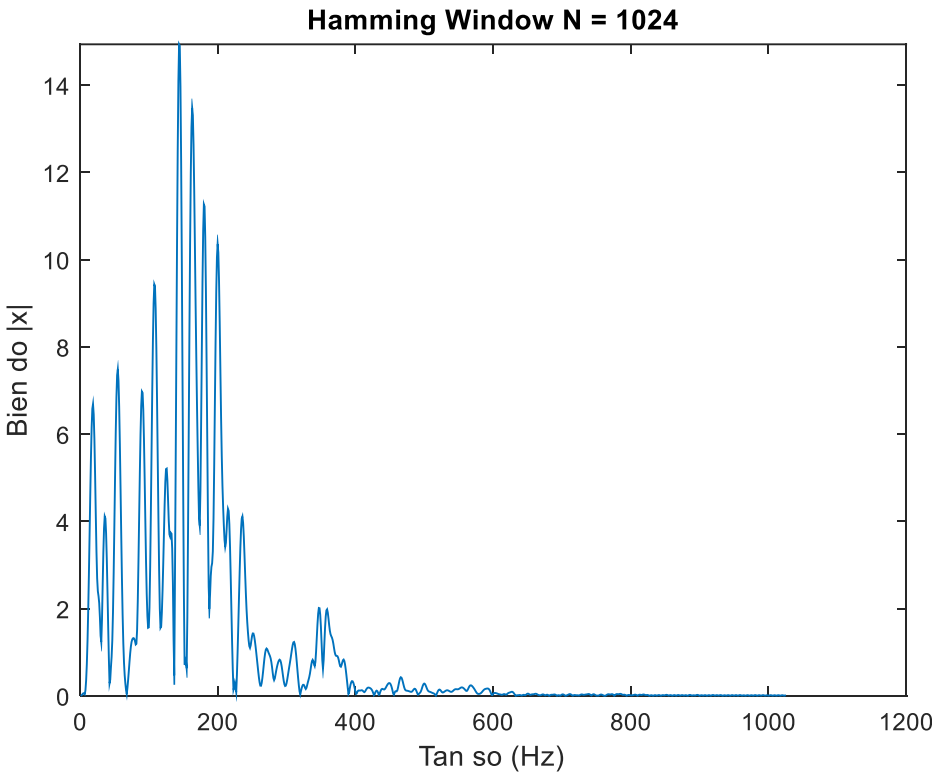


Hình 13: Trường hợp tín hiệu cho kết quả tính F0 trung bình

- Trong trường hợp F0 cho kết quả xấu nhất khi chọn tín hiệu tại đoạn không tuần hoàn thì chương trình báo lỗi vì không xác định được F0.



Hình 14: Trường hợp tín hiệu cho kết quả tính F0 kém chính xác



Hình 15: Hình ảnh dùng của sổ hamming kiểm tra kết quả F0 = 183 (Hz)

2. Bảng so sánh kết quả tính F0:

Bảng 2: Bảng so sánh kết quả tính Fo tự động và thủ công trên miền tần số

	a		e		i		o		u	
	Thủ công	Tự động	Thủ công	Tự động	Thủ công	Tự động	Thủ công	Tự động	Thủ công	Tự động
Son	138.88	152.5	142.85	160.5	153.84	157	153.15	138.7	163.9	163.9
Nhân	158.85	155	147.06	157.5	156.25	178	153.85	167.5	149.25	149.25
Quyên	235.83	231.2	230.89	215.7	233.33	229.6	234.57	250.4	252	252
Tân	137.7	142.3	142.2	137.8	152	149.7	147.1	151	158	158

	Fo mean		Fo std	
	Thủ công	Tự động	Thủ công	Tự động
Son	146.9	151.56	8.89	6.15
Nhân	153.05	163.56	4.87	7.35
Quyên	237.33	236.14	8.4	16.768
Tân	147.4	151.84	7.98	6.8

3. Yếu tố ảnh hưởng đến kết quả tính F0:
- Khi tính tần số cơ bản F0 trên miền tần số, yếu tố ảnh hưởng đến độ chính xác của thuật toán là tín hiệu vào phải không bị nhiễu và phải chọn được những đoạn tuần hoàn của tín hiệu để xuất ra được kết quả F0 chính xác nhất. Khi chọn những đoạn tín hiệu bị nhiễu hay không tuần hoàn thì sẽ không xác định được các đỉnh của phổ biên độ, từ đó thuật toán tính F0 cũng trở nên sai.
 - Chọn những đoạn tín hiệu tuần hoàn thì kết quả tính F0 sẽ chính xác hơn.

D. KẾT LUẬN

Bài báo này thực hiện việc cài đặt thuật toán tìm tần số cơ bản của tín hiệu trên miền tần số và miền thời gian. Để làm tăng độ chính xác của thuật toán cần lựa chọn tín hiệu tốt không bị nhiễu. Khi tín hiệu đưa vào ổn định thì việc tính F0 sẽ chính xác hơn và sai số cũng nhỏ hơn. Trong tương lai sẽ thử nghiệm các thuật toán tính tần số cơ bản trên miền tần số cũng như miền thời gian tiên tiến hơn để cải thiện độ chính xác của giá trị F0.

E. NHỮNG ĐIỀU ĐÃ HỌC ĐƯỢC

- Bài báo cáo giúp em hiểu rõ hơn về một số kiến thức (cách tìm tần số cơ bản trên miền tần số và thời gian)
- Kỹ năng làm việc nhóm với các bạn được nâng cao
- Tăng khả năng tự tìm và đọc tài liệu để hoàn thành bài báo cáo
- Tăng khả năng viết báo, trình bày slide, thuyết trình
- Cảm ơn thầy đã cung cấp cho em các kiến thức cần thiết để làm bài báo cáo này.

F. TÀI LIỆU THAM KHẢO

[1] Wikipedia (<https://vi.wikipedia.org/wiki>)

[2] Oberhettinger F., "Tables of Fourier Transforms and Fourier Transforms of Distributions." Springer, 1990.

[3] *Phạm Xuân Sự, Lê Xuân Thành, Bài giảng xử lý tiếng nói, Học viện công nghệ bưu chính viễn thông, 2010*

[4] Xử lý tín hiệu số - Vũ Văn Biên

[5] Wikipedia
(https://vi.wikipedia.org/wiki/T%E1%BA%A7n_s%E1%BB%91_%C3%A2m_c%C6%A1_b%E1%BA%A3n)