

TÌM TẦN SỐ CƠ BẢN CỦA TÍN HIỆU TRÊN MIỀN THỜI GIAN DÙNG HÀM TỰ TƯƠNG QUAN VÀ TRÊN MIỀN TẦN SỐ KẾT HỢP LỌC TRUNG VỊ

Nhóm 2:

Hồ Gia Khánh, Đinh Quang Duy, Hồ Trọng Hiền, Nguyễn Xuân Lực.

Lớp HP: 1022103.1810.16.10

| Điểm | Bảng phân chia nhiệm vụ | | Chữ ký sinh viên |
|------|-----------------------------|--|------------------|
| | Hồ Gia Khánh | Đọc tài liệu, cài đặt thuật toán, viết báo cáo, làm slide và thuyết trình về hàm lọc trung vị. (tr. 7-9, 12-13) | |
| | Đinh Quang Duy | Đọc tài liệu, cài đặt thuật toán, viết báo cáo, làm slide và thuyết trình về hàm tự tương quan. Cài đặt hàm sai số RMSE (tr. 3-5, 8-10, 14) | |
| | Nguyễn Xuân Lực | Đọc tài liệu phân tích phổ, tìm F0 trên miền tần số, làm slide và thuyết trình về F0 trên miền tần số. Định dạng lại báo cáo. (tr. 5-6, 10-12) | |
| | Hồ Trọng Hiền (nhóm trưởng) | | |

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: Tìm tần số cơ bản của tín hiệu là bài toán cần thiết trong xử lý tín hiệu âm thanh, đặc biệt là tín hiệu tiếng nói. Bài thực hành này thực hiện việc xác định tần số cơ bản của tín hiệu tiếng nói trên miền thời gian dùng hàm tự tương quan và trên miền tần số. Kết hợp lọc trung vị để làm tròn kết quả thu được. Kết quả thực nghiệm cũng cho thấy ta có thể xác định được tần số cơ bản của tín hiệu tiếng nói trên miền thời gian và miền tần số. Tuy nhiên, việc phát triển một thuật toán để tính chính xác tần số cơ bản là không dễ vì thường gặp những lỗi như tín hiệu có nhiễu...

Từ khóa: Tính/đo tần số cơ bản, tính/đo chu kỳ pitch, biểu diễn trên miền tần số, hàm tự tương quan, lọc trung vị

MỤC LỤC

I. ĐẶT VẤN ĐỀ3

II. LÝ THUYẾT XỬ LÝ TÍN HIỆU TIẾNG NÓI VÀ CÁC THUẬT TOÁN TÍNH F0 VÀ LÀM TRON F0 THU ĐƯỢC BẰNG LỌC TRUNG VỊ.....3

 A. Lý thuyết.....3

 B. Thuật toán tìm tần số cơ bản tự tương quan.....3

 1. Đặt vấn đề.....3

 2. Cơ sở lý thuyết.....3

 3. Sơ đồ khối.....4

 4. Các tham số quan trọng của hàm tự tương quan.....4

 5. Vấn đề phát sinh5

 C. Thuật toán tìm tần số cơ bản trên miền tần số.....5

 1. Đặt vấn đề.....5

 2. Cơ sở lý thuyết.....5

 3. Sơ đồ khối.....6

 4. Các tham số quan trọng6

 5. Các vấn đề phát sinh.....6

 D. Thuật toán lọc trung vị để làm tron F0 thu được.....7

 1. Đặt vấn đề.....7

 2. Cơ sở lý thuyết.....7

 3. Các tham số quan trọng của hàm lọc trung vị.....8

III. CÀI ĐẶT CÁC THUẬT TOÁN.....8

 A. Thuật toán tự tương quan.....8

 B. Thuật toán trên miền tần số.....10

 C. Thuật toán lọc trung vị để làm tron F0 thu được.....12

 D. Thuật toán tính sai số RMSE14

IV. KẾT QUẢ THỰC NGHIỆM14

 A. Hình vẽ.....14

 1. Hàm tự tương quan.....14

 2. F0 trên miền tần số15

 B. Bảng biểu.....17

 C. Nhận xét.....19

V. KẾT LUẬN.....20

VI. NHỮNG ĐIỀU ĐÃ HỌC ĐƯỢC20

VII. LỜI CẢM ƠN20

VIII. TÀI LIỆU THAM KHẢO.....20

I. ĐẶT VẤN ĐỀ

Xử lý và phân tích tiếng nói hiện nay đóng vai trò quan trọng trong các ngành khoa học và kỹ thuật. Phân tích tiếng nói thực hiện giải quyết các vấn đề để tìm ra một dạng thức tối ưu biểu diễn được tiếng nói một cách hiệu quả. Nó là cơ sở cho việc phát triển các kỹ thuật, công nghệ tổng hợp, nhận dạng và nâng cao chất lượng tín hiệu tiếng nói. Phân tích tiếng nói thường thực hiện việc trích chọn hoặc chuyển đổi tín hiệu tiếng nói sang một dạng thức biểu diễn khác sao cho có thể biểu diễn thông tin tiếng nói tốt hơn theo cách mà chúng ta cần [1]. Để làm được điều đó, việc xác định tần số cơ bản là rất quan trọng. Có nhiều phương pháp khác nhau như sử dụng hàm sai biên độ trung bình (AMDF), LPC, xử lý đồng hình (homomorphic), hàm tự tương quan (autocorrelation), sử dụng bộ lọc đạo và hàm tự tương quan, phân tích phổ trên miền tần số... để xác định được tần số cơ bản của tiếng nói [1].

II. LÝ THUYẾT XỬ LÝ TÍN HIỆU TIẾNG NÓI VÀ CÁC THUẬT TOÁN TÍNH F_0 VÀ LÀM TRƠN F_0 THU ĐƯỢC BẰNG LỌC TRUNG VỊ

A. Lý thuyết

Xử lý tín hiệu số đang bùng nổ nhanh chóng trong ngành công nghiệp điện tử và viễn thông hiện nay bởi nhiều lợi thế hơn so với xử lý tín hiệu liên tục. Xử lý tín hiệu số có nhiều ứng dụng đa dạng, đặc biệt trong việc xử lý tiếng nói hay xử lý âm thanh.

Tần số cơ bản (còn gọi là F_0 hoặc cao độ) của một tín hiệu tuần hoàn bằng nghịch đảo của chu kỳ cơ bản của tín hiệu đó. F_0 phụ thuộc vào giới tính, lứa tuổi của người phát âm, F_0 thay đổi theo thanh điệu và F_0 cũng ảnh hưởng đến ngữ điệu của câu nói [1]. Chu kỳ cơ bản là khoảng thời gian nhỏ nhất của tín hiệu tuần hoàn trên miền thời gian [3]. Tần số cơ bản mang thông tin có ý nghĩa vật lý đặc trưng cho tín hiệu tuần hoàn nên việc xác định nó là rất quan trọng trong xử lý tín hiệu số nói chung và tín hiệu giọng nói nói riêng. Trong bài báo cáo này, hàm tự tương quan được sử dụng để tính tần số cơ bản trên miền thời gian và sử dụng phép biến đổi nhanh Fourier (FFT) để phân tích phổ và tính tần số cơ bản trên miền tần số để tìm được tần số cơ bản của tiếng nói thông qua 5 nguyên âm /u/ /e/ /o/ /a/ /i/. Lọc trung vị (median smoothing) sử dụng để làm trơn kết quả F_0 thu được.

B. Thuật toán tìm tần số cơ bản tự tương quan

1. Đặt vấn đề

Việc xác định tần số cơ bản của tín hiệu là rất quan trọng trong việc phân tích và xử lý tín hiệu.

Trong xử lý tín hiệu số, có rất nhiều phương pháp để xác định tần số cơ bản của tín hiệu tiếng nói, tuy nhiên trong báo cáo này, chúng ta sẽ sử dụng một trong những thuật toán phổ biến để xác định chu kỳ cơ bản của tín hiệu trên miền thời gian bằng cách ứng dụng hàm tự tương quan, từ đó tính được tần số cơ bản của tín hiệu tiếng nói.

2. Cơ sở lý thuyết

a) Phân tích tiếng nói trong miền thời gian

Các đặc điểm của tín hiệu tiếng nói luôn biến đổi theo thời gian. Chẳng hạn như cao độ (pitch), mức độ kích thích giữa âm vô thanh, âm hữu thanh và khoảng lặng, các biên độ khác nhau của tín hiệu, hay sự biến thiên tần số cơ bản trong miền tín hiệu tuần hoàn. Chính vì các đặc điểm trên luôn luôn biến đổi nên giả định trong hầu hết các phương pháp xử lý tín hiệu tiếng nói là các thành phần trong tín hiệu tiếng nói thay đổi không đáng kể hoặc cố định so với thời gian. Để áp dụng giả định trên, trong lý thuyết phân tích, các phân tích phải được tiến hành trong một khoảng thời gian giới hạn. Vì vậy nên khi đi phân tích tín hiệu tiếng nói có khoảng thời gian tương đối dài, phải áp dụng các phương pháp xử lý ngắn hạn nhằm chia tín hiệu thành từng đoạn (segment) hay từng khung nhỏ (frame) trong một khoảng thời gian tương đối nhỏ (thường khoảng từ 10-30ms). Các đoạn này thường được lấy chồng lên nhau nhằm tránh việc mất mát tín hiệu.

b) Hàm tự tương quan (Autocorrelation Function)

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 [8].

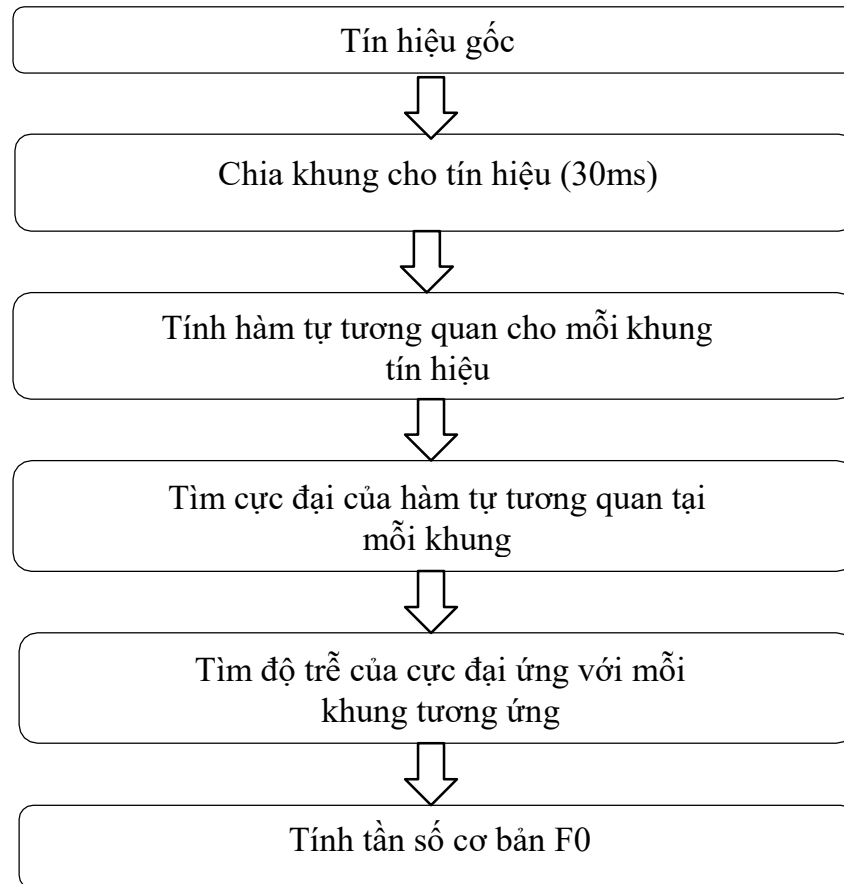
Hàm tự tương quan - AutoCorrelation Function (ACF) trên khung tiếng nói có độ dài N [9] là

Với N là độ rộng của cửa sổ, l là độ trễ được tính tại thời điểm n .

$$r_{xx}(l) = \frac{1}{n} \sum_{n=0}^{N-1} (x(n)x(n-l))$$

Hàm tự tương quan là hàm chẵn, đạt giá trị cực đại tại $l = 0$. Đại lượng $r_{xx}(0)$ chính bằng năng lượng của tín hiệu. Tầm quan trọng hàm tự tương quan nằm ở việc hàm sẽ đạt các giá trị cực đại tương ứng tại các điểm là bội của chu kỳ cơ bản của tín hiệu. Khi đó các tần số cơ bản là tần số xuất hiện của các cực đại đó. Tính chất này khiến hàm tự tương quan trở thành cơ sở cho việc tính toán chu kỳ của tất cả các loại tín hiệu, bao gồm cả tín hiệu tiếng nói.

3. Sơ đồ khối



Hình 1. Sơ đồ khối thuật toán tìm F_0 trên miền thời gian dùng hàm tự tương quan

4. Các tham số quan trọng của hàm tự tương quan

Công thức tính tần số cơ bản của tín hiệu: $F_0 = F_s / L$, trong đó F_0 là tần số cơ bản, F_s là tần số lấy mẫu và L là độ trễ mà tại đó hàm tự tương quan đạt cực đại.

Vì tần số lấy mẫu F_s đã biết, nên để tính tần số cơ bản, ta phải đi tìm độ trễ L .

5. Vấn đề phát sinh

• Số lượng cực đại của khung tín hiệu

Vì khung tín hiệu có chiều dài hữu hạn nên năng lượng cũng hữu hạn, điều này dẫn đến biên độ các đỉnh trong đồ thị tương quan sẽ giảm dần về 2 bên, với trục đối xứng là điểm chính giữa đồ thị. Điều này khiến cho việc xác định cực đại sẽ khó khăn, do đó ta chỉ lấy cực đại kế cận cực đại trung tâm, qua đó việc tính toán tần số cơ bản sẽ bớt phức tạp hơn.

C. Thuật toán tìm tần số cơ bản trên miền tần số

1. Đặt vấn đề

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 [5]. Chu kỳ được xác định bằng khoảng thời gian ngắn nhất mà tín hiệu lặp lại trên miền thời gian. Tần số F_0 là tần số giao động của dây thanh. Trong bài báo cáo này, tín hiệu được xử lý là tín hiệu tiếng nói nên tần số cơ bản phụ thuộc vào giới tính và độ tuổi. F_0 của nữ thường cao hơn của nam, F_0 của người trẻ thường cao hơn của người già. Thường với giọng của nam, F_0 nằm trong khoảng từ 80-250Hz, với giọng của nữ, F_0 trong khoảng 120-400Hz. Sự biến đổi của F_0 có tính quyết định đến thanh điệu của từ cũng như ngữ điệu của câu. Câu hỏi đặt ra là làm thế nào để xác định tần số cơ bản (fundamental frequency).

Vào những năm thập kỷ 60, khi công nghệ vi xử lý phát triển chưa mạnh thì thời gian xử lý phép toán DFT (Discrete Fourier Transform: Biến đổi Fourier rời rạc: liên kết thời gian rời rạc và tần số rời rạc) trên máy tương đối chậm, do số phép nhân phức tương đối lớn. Để khắc phục về mặt tốc độ xử lý của phép tính DFT, nhiều tác giả đã đưa ra các thuật toán riêng dựa trên DFT gọi là FFT (Fast Fourier Transform: Biến đổi Fourier nhanh).

Phương pháp được sử dụng trong nghiên cứu này là phương pháp biến đổi nhanh Fourier (FFT) để xác định tần số cơ bản (F_0) trên miền tần số kết hợp với lọc trung vị để làm trơn kết quả F_0 thu được.

2. Cơ sở lý thuyết

a) Phép biến đổi Fourier rời rạc (Discrete Fourier Transform –DFT)

Trong toán học, phép biến đổi Fourier rời rạc, đôi khi còn được gọi là biến đổi Fourier hữu hạn, là một phép biến đổi trong giải tích Fourier cho các tín hiệu thời gian rời rạc. Đầu vào của biến đổi này là một chuỗi hữu hạn các số phức hoặc các số thực. Đặc biệt, biến đổi này được sử dụng rộng rãi trong xử lý tín hiệu và các ngành liên quan đến phân tích tần số chứa trong một tín hiệu. Biến đổi này được tính nhanh bởi thuật toán biến đổi Fourier nhanh (FFT)[5].

Biến đổi Fourier rời rạc cho phép tính phiên bản tần số rời rạc của biến đổi Fourier của tín hiệu rời rạc (Discrete-time Fourier Transform – DTFT).

Công thức DFT N-điểm

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\omega k} \text{ với } \omega = \frac{k2\pi}{N} \text{ và } 0 \leq k \leq N-1 (*)$$

$X[k]$ đại diện cho biên độ và pha ở các bước sóng khác nhau của tín hiệu vào $x[n]$. Phép biến đổi DFT tính các giá trị $X[k]$ từ các giá trị $x[n]$ [5].

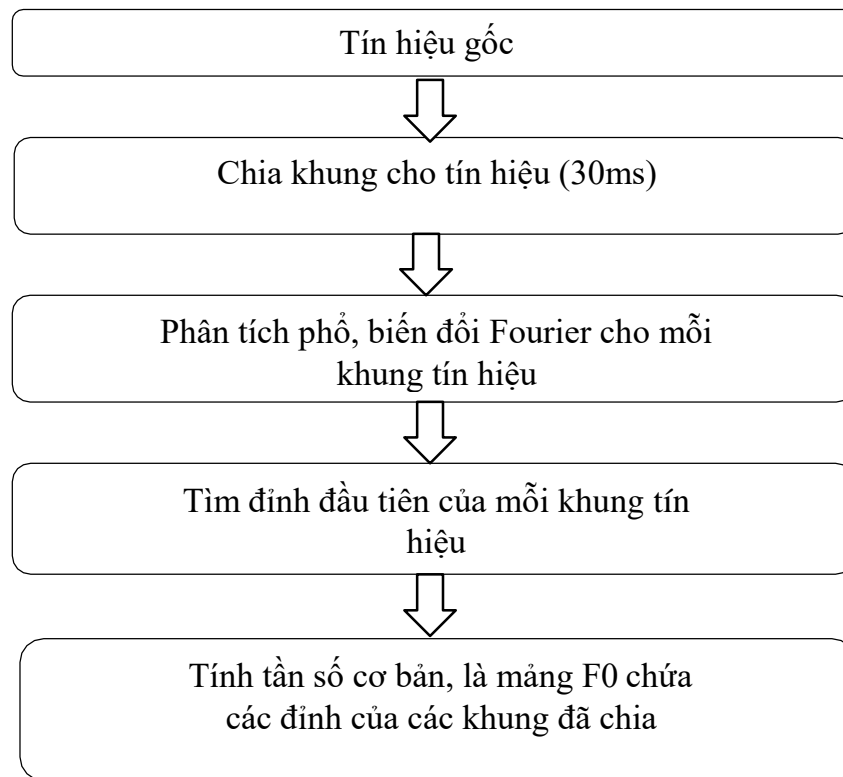
b) Phép biến đổi Fourier nhanh (Fast Fourier Transform - FFT)

Một biến đổi Fourier nhanh (FFT) là một thuật toán hiệu quả để tính biến đổi Fourier rời rạc (DFT) và biến đổi ngược. Có nhiều thuật toán FFT khác nhau sử dụng kiến thức từ nhiều mảng khác nhau của toán học [7].

Các thuật toán FFT tính nhanh DFT N-điểm của tín hiệu rời rạc $x[n]$, trong đó, $N = 2^m \geq \text{length}(x[n])$. Tính trực tiếp từ định nghĩa DFT (*) đòi hỏi $O(N^2)$ phép tính: có N số X_k cần tính, để tính mỗi số cần tính một tổng N số hạng. Một FFT là một phương pháp để tính cùng kết quả đó trong $O(N \log N)$ phép tính.

Trong bài báo cáo này, sử dụng hàm `fft()` trong thư viện Matlab để phân tích phổ kết hợp với hàm `findpeaks()` để tìm tần số cơ bản.

3. Sơ đồ khối



Hình 2. Sơ đồ khối thuật toán tìm F0 trên miền tần số

4. Các tham số quan trọng

Hàm `fft()` trong thư viện Matlab:

- Cú pháp $Y = \text{fft}(X, N)$ với X là tín hiệu vào (trên miền thời gian rời rạc) và N là số điểm trong phép biến đổi DFT- N điểm.
- Nếu $\text{length}(X) > N$ thì X được thêm vào các giá trị 0 cho tới khi $\text{length}(X) = N$ để thực hiện phép biến đổi.
- Nếu $\text{length}(X) < N$ thì X được cắt ngắn sao cho $\text{length}(X) = N$.
- Kết quả trả về (Y) là mảng chứa các số phức biểu diễn tín phổ (spectrum) của $x[n]$.

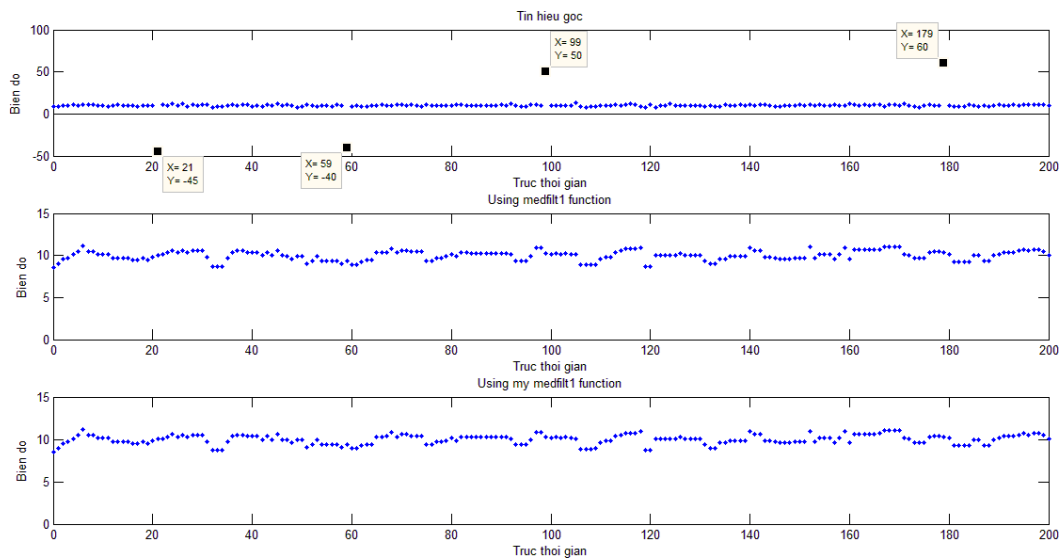
5. Các vấn đề phát sinh

Trong việc tìm phổ của tín hiệu liên tục cũng như tín hiệu rời rạc, tất cả các giá trị của tín hiệu trên miền thời gian đều rất cần thiết. Tuy nhiên, trên thực tế, chúng ta chỉ có thể tính toán và quan sát trong khoảng thời gian giới hạn. Do vậy, phổ tín hiệu chỉ có thể được tính xấp xỉ từ lượng dữ liệu giới hạn thông qua cửa sổ. Từ đó, ta sẽ bị hạn chế khả năng phân biệt hai thành phần tần số được tách ra nhỏ hơn tần số cơ bản. Phổ trong cửa sổ sẽ trải dài ra khắp miền tần số nên năng lượng của tín hiệu gốc tập trung tại một tần số sẽ bị trải ra khắp miền tần số bởi hàm cửa sổ. Hiện tượng này được gọi là rò phổ, đây là hiện tượng rất phổ biến khi sử dụng hàm cửa sổ [6].

D. Thuật toán lọc trung vị để làm trơn F0 thu được

1. Đặt vấn đề

Bộ lọc trung vị là một kỹ thuật lọc số phi tuyến tính, thường được sử dụng để loại bỏ nhiễu khỏi hình ảnh hoặc tín hiệu, giảm tiếng ồn. Đây là một bước xử lý trước điển hình để cải thiện kết quả sau khi xử lý. Lọc trung vị được sử dụng rộng rãi trong xử lý ảnh kỹ thuật số bởi vì, trong điều kiện nhất định, nó bảo đảm việc loại bỏ nhiễu nhưng tín hiệu hay hình ảnh vẫn không bị mất nhiều thông tin. Lọc trung vị có nhiều ứng dụng trong xử lý tín hiệu. Bài báo này dùng lọc trung vị để làm trơn F_0 thu được sau khi trả qua quá trình tìm F_0 bằng hàm tự tương quan hay trên miền tần số [3]. Sau đây là một tính hiệu bất kì bị nhiễu nhiễu, có vài điểm có biên độ bất thường cần phải lọc.



Hình 3. Ví dụ cụ thể về tác dụng của lọc trung vị.

Nhận xét: Ta thấy những điểm có biên độ bất thường được làm trơn về phù hợp với tín hiệu gốc

2. Cơ sở lý thuyết

Trong lý thuyết xác suất và thống kê, **số trung vị** (tiếng Anh: *median*) là một số tách giữa nửa lớn hơn và nửa bé hơn của một mẫu, một quần thể, hay một phân bố xác suất. Nó là giá trị giữa trong một phân bố, mà số các số nằm trên hay dưới con số đó là bằng nhau. Điều đó có nghĩa rằng 1/2 quần thể sẽ có các giá trị nhỏ hơn hay bằng số trung vị, và một nửa quần thể sẽ có giá trị bằng hoặc lớn hơn số trung vị [4].

Nếu số phần tử của dãy số là lẻ ($n = 2k + 1$ phần tử), thì số trung vị sẽ là số ở vị trí thứ $n + 1$. Ví dụ trong dãy số 2, 5, 8, 10, 14, 16, 22 thì số trung vị sẽ là 10.

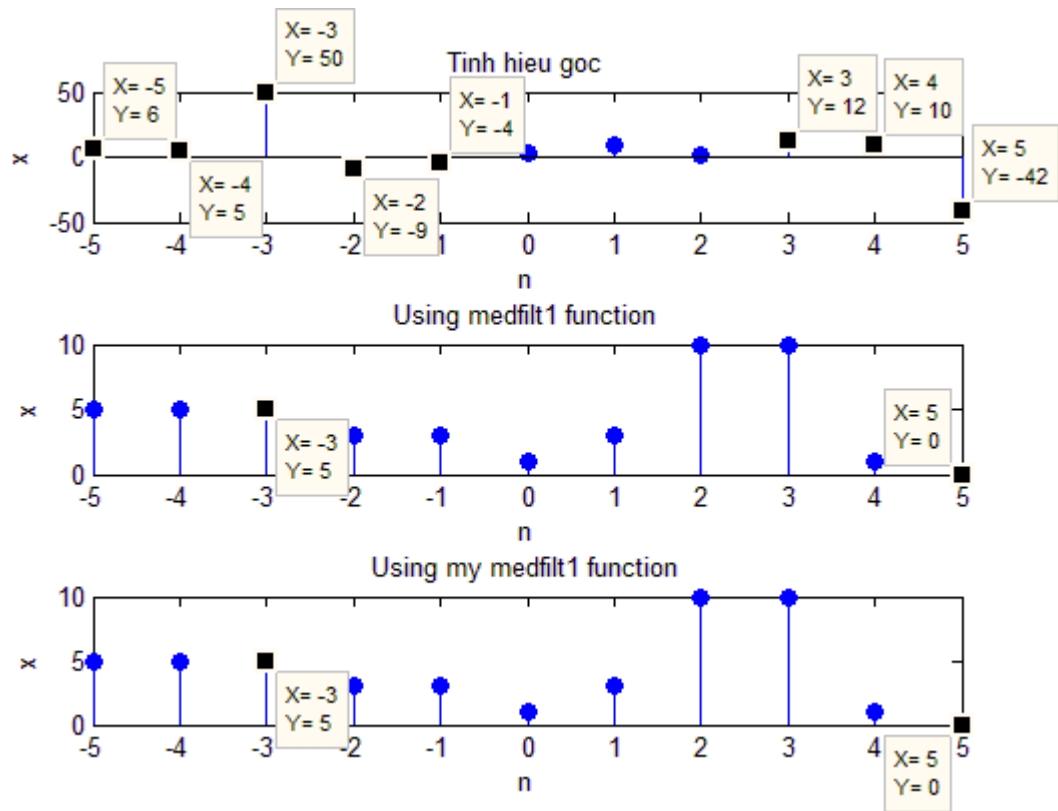
Nếu số phần tử của dãy là số chẵn ($n = 2k$ phần tử), thì số trung vị sẽ là trung bình cộng của phần tử thứ n và thứ $n + 1$. Ví dụ dãy số 5, 6, 9, 11, 15, 17, 20, 24 thì số trung vị sẽ là trung bình cộng của 11 và 15 là 13.

Ví dụ 1:

| | | | | | | | | | | | |
|-------------------------|--------------|---|---|--|--|--|----|----|----|---|---|
| Ví dụ của cửa sổ bằng 3 | | | | | | | | | | | |
| | Mảng ban đầu | 3 | 8 | 6 | 9 | 7 | 12 | 11 | 60 | 6 | |
| Cửa sổ tại điểm 1 | 0 | 3 | 8 | Sắp xếp tăng dần ta được 0 3 8, số ở giữa là số 3 sẽ được gán lại vị trí số 1 của mảng | | | | | | | |
| Cửa sổ tại điểm 2 | | 3 | 8 | 6 | Sắp xếp tăng dần ta được 3 6 8, số ở giữa là số 6 sẽ được gán lại vị trí số 2 của mảng | | | | | | |
| Cửa sổ tại điểm 3 | | | 8 | 6 | 9 | Sắp xếp tăng dần ta được 6 8 9, số ở giữa là số 8 sẽ được gán lại vị trí số 3 của mảng | | | | | |
| Cửa sổ tại điểm 4 | | | | 6 | 9 | 7 | | | | | |
| Cửa sổ tại điểm 5 | | | | | 9 | 7 | 12 | | | | |
| Cửa sổ tại điểm 6 | | | | | | 7 | 12 | 11 | | | |
| Cửa sổ tại điểm 7 | | | | | | | 12 | 11 | 60 | | |
| Cửa sổ tại điểm 8 | | | | | | | | 11 | 60 | 6 | |
| Cửa sổ tại điểm 9 | | | | | | | | | 60 | 6 | 0 |

Hình 4. Cách thuật toán hoạt động

Ví dụ 2:



Hình 5. Tín hiệu trước và sau khi lọc

Xét tín hiệu ban đầu, tại $n = -3$ thì giá trị của x là 50. Áp dụng lọc trung vị cho tín hiệu x với bậc là 5, tức là ta xét các giá trị của x tại $n = -5, -4, -3, -2, -1$. Mỗi giá trị tại đó tương ứng với 6, 5, 50, -9, -4. Sau khi sắp xếp tăng dần ta được -9, -4, 5, 6, 50 và chọn trung vị, ta được số trung vị là 5. Và 5 sẽ được chọn là giá trị cho tín hiệu x sau khi lọc tại $n = -3$.

Một ví dụ khác, tại $n = 5$, giá trị của x sẽ là -42, sau khi lọc tín hiệu x với bậc là 5 (xét các giá trị $n = 3, -4, 5, 6, 7$ tương ứng với các giá trị là -9, 4, -50, 10, 1) thì giá trị của x tại $n = 0$ là -4.

Nhận xét: Từ ví dụ trên, có thể thấy rằng tác dụng của lọc trung vị là lọc những điểm có biên độ bất thường của tín hiệu, hay nói cách khác là làm mịn tín hiệu. Bậc N của bộ lọc cần phải được xem xét để tín hiệu sau khi lọc không được còn những điểm bất thường hoặc là quá mịn dẫn đến sự sai lệch so với kết quả thực tiễn.

3. Các tham số quan trọng của hàm lọc trung vị

Trong matlab, đã có sẵn hàm thư viện để lọc trung vị của tín hiệu cần lọc. Đó là hàm `medfilt1` (lọc trung vị 1 chiều) để lọc trung vị các tín hiệu một chiều. Cú pháp của hàm là `medfilt1(x,N)`. Với x là tín hiệu đầu vào sau khi tính F_0 bằng hàm tự tương quan (đã được rời rạc hóa) hoặc trên miền tần số, N là số chiều (bậc) của tín hiệu hay chiều dài hàm cửa sổ. Để lấy chính xác điểm trung vị, N thường là số lẻ. Bài báo này sẽ sử dụng hàm `my_medfilt(x, N)` có chức năng giống như hàm thư viện `medfilt1(x, N)`.

III. CÀI ĐẶT CÁC THUẬT TOÁN

A. Thuật toán tự tương quan

Chương trình sẽ đọc tín hiệu từ các file nguyên âm /a/, /i/, /u/, /e/, /o/ đã thu từ 2 thí nghiệm đã làm trước đó, và tính toán được chu kỳ cơ bản của các nguyên âm trên, từ đó rút ra tần số cơ bản.

Trong thư viện Matlab, hàm `length()` trả về độ dài của mảng, sử dụng hàm `xcorr()` với đối số truyền vào là tín hiệu, sử dụng hàm `plot` vẽ được đồ thị hàm tự tương quan, tiếp theo xác định các cực đại địa phương sử dụng hàm `findpeaks()`, sau đó tìm độ trễ tại các cực đại đó sử dụng hàm `max()`, từ độ trễ ta tính được tần số cơ bản F_0 .

Kết quả thu được sẽ đi qua bộ lọc trung vị (được trình bày Phần IV), từ đó sẽ tính được tần số cơ bản của tín hiệu một cách tương đối chính xác. Cuối cùng tính sai số so với âm thử công đã đo được sử dụng hàm `RMSE()` với đối số truyền vào là tần số sau khi lọc và tần số đo thử công được sinh viên tự cài đặt. hàm `round()` làm tròn số nguyên, hàm `mean()` để tính trung bình, hàm `std()` để tính độ lệch chuẩn

```
[y,Fs] = audioread('oluc.wav');
t=1/Fs:1/Fs:(length(y)/Fs);
subplot(3,1,3);
plot(t,y);
title('Tín hiệu input');
xlabel('Chi số thời gian');
ylabel('Biên độ');
frame_leng=0.03 * Fs;
half = round(frame_leng/2);
number_frame = round(length(y)/half) - 1;
result = [];
F0 = [];
k = 1;
% doc file am thanh, tao mien thoi gian t, ve hinh tin hieu am thanh voi
% bien do la truc doc va chi so thoi gian la truc ngang
% chia khung (frame_leng) cho moi tin hieu 30ms, tinh so luong khung
% khoi tao mang result va F0 rong de nhan gia tri, bien dem k de dem so
% phan tu trong mang F0

for i = 1 : number_frame - 1
    range = ((i-1)*half + 1) : (frame_leng + (i-1)*half);
    frame = y(range);
    [rxx, lags] = xcorr(frame);
    lags1 = lags((length(lags) - 1)/2 + 1 : length(lags));
    rxx1 = rxx((length(lags) - 1)/2 + 1 : length(lags));
    lags11 = lags1(1 : length(lags1) - 1);
    rx1 = rxx1(1 : length(rxx1) - 1);
    % khung i+1 chạy chồng lên khung i nửa khung để không bị mất tín hiệu
    % lấy hàm tự tương quan cho khung tín hiệu frame
    % lấy một nửa của hàm tự tương quan để tính kết quả vì nó đối xứng

    [pksa, locs] = findpeaks(rx1);
    [a,x] = max(pksa);
    result(i) = Fs/locs(x);
    if(result(i) > 400 || result(i) < 80)
        result(i) = 0;
    else
        F0(k) = result(i);
        k=k+1;
    end
    % tìm đỉnh trong một nửa của hàm tự tương quan lấy ở trên lưu vào mảng
    pksa
    % tìm cực đại của mỗi khung từ các đỉnh trong pksa
    % tìm độ trễ mà tại đó có cực đại
    % kiểm tra điều kiện nam trong khoảng âm thanh cho phép của nam 80-250
    % và của nữ là 120-400 nếu đúng lưu vào mảng F0, nếu sai gán result(i) = 0
end

subplot(3,1,2);
stem(result, 'b', 'LineStyle', 'none', 'MarkerFaceColor', 'w');
title('Trước khi lọc');
xlabel('n');
ylabel('F');
```

```

F0_mean = mean(F0);
F0_std = std(F0);
%F0 tren ca tin hieu truoc khi loc
% ve ket qua result vua tim duoc sau khi ket thuc vong lap
%F0_mean trung binh trong doan tuan hoan
%F0_std do lech chuan trong doan tuan hoan

F2_result = my_medfilt1(result, 5);
subplot(3, 1, 1);
stem(F2_result, '.', 'LineStyle','none', 'MarkerFaceColor','w');
title('Sau khi loc voi N = 5');
xlabel('n');
ylabel('F');
%TIEN HANH LOC TRUNG VI VOI N = 5 tu mang result de ve tan so

F2 = my_medfilt1(F0, 5);
F2_mean = mean(F2);
F2_std = std(F2);
%Loc F0 doan tuan hoan
%F2_mean trung binh trong doan tuan hoan
%F2_std do lech chuan trong doan tuan hoan

SaiSo = RMSE(F2, 141.35);
%Sai so tren F0 sau khi loc trung vi so voi F0 thu cong tinh o buoc 1
%voi RMSE(F2, 141.35) la ham tinh sai so, tham so truyen vao la tan so
sau khi loc F2 va tan so F0 tinh thu cong 141.35

```

Kết quả F0 sẽ được trình bày trong phần IV

B. Thuật toán trên miền tần số

Trong matlab sử dụng Hàm length() trả về độ dài của mảng, biến đổi Fourier fft() tham số truyền vào là tín hiệu ban đầu, findpeaks() trả về các đỉnh trong tín hiệu, linspace() trả về các điểm cách đều nhau. Cuối cùng tính sai số so với âm thử công đã đo được sử dụng hàm RMSE() với đối số truyền vào là tần số sau khi lọc và tần số đo thử công được sinh viên tự cài đặt

```

function [frame_len,half] = create(Fs)
    frame_len = round(0.03*Fs);
    half = round(frame_len/2);
end
function [P2,P1] = biendoi(frame)
    P2 = abs(fft(frame,1024));
    P1 = P2(1:length(P2)/2+1);
    P1(2:end-1) = 2*P1(2:end-1);
end
clear;clc;clf;

[y,Fs] = audioread('ukhanh.wav');

%Su dung ham create de tao do dai khung tin hieu (30ms)
[frame_len,half] = create(Fs);

temp = [];
F0 = [];
i=1;

```

```

%Vong lap chay tu 1 -> so khung
for k = 1 : (length(y)/half) - 1
    range = ((k-1)*half + 1):(frame_len + (k-1)*half);
    frame = y(range);

%Hien thi tin hieu dau vao la mau xanh va khung duoc chon mau do
    subplot(3,1,1)
    plot(y);
    title('Tin Hieu');
    xlabel('Chi so thoi gian');
    ylabel('Bien Do');
    hold on;
    yn = ones(size(y))*NaN;
    yn(range) = frame;
    plot(yn,'r');
hold off

%Hien thi tung khung
    subplot(3,1,2)
    plot(frame);
    title('Pho bien do');
    xlabel('Tan So');
    ylabel('Bien Do');

%Su dung ham bien doi de tinh bien doi Fourier roi rac (DFT)
    [P2,P1] = biendoi(frame);

    freq=linspace(1/Fs,Fs/2,length(P1));
    subplot(3,1,3)
    findpeaks(P1,freq,'MinPeakDistance',80);
    title('Hien thi cac dinh vua tim duoc');
    xlabel('Tan So');
    ylabel('Bien Do');

%Su dung ham findpeak de tim cac dinh
    [pks,locs] = findpeaks(P1,freq, 'MinPeakDistance',80);
    temp(k) = locs(1);
    if(temp(k) > 400 || temp(k) < 80)
        temp(k) =0;
    end
end

```

```

        else
            F0(i) = temp(k);
            i= i+1;
        end
    pause;
end

figure(2);
subplot(4,1,1);
stem(temp, '.', 'LineStyle','none', 'MarkerFaceColor','w');
title('Truoc khi loc');
xlabel('n');
ylabel('F');

F0_mean = mean(F0); %F0 trung bình trong doan tuan hoan
F0_std = std(F0);   %F0 do lech chuan trong doan tuan hoan
%tien hanh loc trung vi
%Loc F0 tren ca tin hieu sau khi loc voi N = 5
F2_temp = my_medfilt1(temp, 5);
subplot(4, 1, 3);
%ve F0 tren ca tin hieu sau khi loc voi N = 5
stem(F2_temp, '.', 'LineStyle','none', 'MarkerFaceColor','w');
title('Sau khi loc voi N = 5');
xlabel('n');
ylabel('F');
%Loc F0 doan tuan hoan
F2 = my_medfilt1(F0, 5);
F2_mean = mean(F2); %F2 trung bình trong doan tuan hoan
F2_std = std(F2);   %F2 do lech chuan trong doan tuan hoan
SaiSo = RMSE(F2, 175);
%Sai so tren F0 sau khi loc trung vi so voi F0 thu cong tinh o buoc 1
%voi RMSE(F2, 141.35) la ham tinh sai so, tham so truyen vao la tan so
sau khi loc F2 va tan so F0 tinh thu cong 175

```

Kết quả F0 sẽ được trình bày trong phần IV

C. Thuật toán lọc trung vị để làm trơn F0 thu được

Sử dụng hàm `ceil()` làm tròn trên với các đối số truyền vào là số cần làm tròn và trả về số sau khi thực hiện. Hàm `length()` trả về độ dài của mảng

- **Hàm sắp xếp tăng dần Sort mảng 1 chiều để phục vụ cho hàm my_medfilt1**

```
function y = Sort(y)
    for i = 1:length(y)-1
        for j = i+1:length(y)
            if (y(i) > y(j))
                temp = y(i);
                y(i) = y(j);
                y(j) = temp;
            end
        end
    end
end
```

%y là mảng cần sắp xếp, hàm Sort trả về một mảng đã sắp xếp

- **Hàm lọc trung vị 1 chiều my_medfilt1 giống với hàm thư viện của matlab medfilt1**

```
function z = my_medfilt1(y, m)
    z = y; %Do dài z bằng do dài y
    temp = 1:m; %Mang của so
    for x = 1:length(y)
        for i = 1:m
            if(x < ceil(m/2))
                if(i < ceil(m/2) - x+1)
                    %Neu thoa 2 dieu kien tren thi bo sung them so
                    %0 vao dau mang temp cho phu hop
                    temp(i) = 0;
                else
                    %Gan gia tri phu hop cua mang y qua temp
                    temp(i) = y(x+i-ceil(m/2));
                end
            else
                if(x > length(y) - ceil(m/2) + 1)
                    if(i < length(y) + ceil(m/2) - x + 1)
                        %Gan gia tri phu hop cua mang y qua temp
                        temp(i) = y(x-ceil(m/2) + i);
                    else
                        %Bo sung them so 0 vao cuoi mang temp cho
                        %du chieu dài của temp
                        temp(i) = 0;
                    end
                else
                    %Neuphan tu của y không thuộc 2 dau mang de
                    %them so 0vao temp thì gan gia tri phu hop của
                    %y qua temp
                    temp(i) = y(x-ceil(m/2)+i);
                end
            end
        end
    end

    %sáp xếp và gan gia tri o giữa vào một mảng mới để trả về
    %ket qua
    temp = Sort(temp);
    z(x) = temp((length(temp)+1)/2);
end

end

%Gia tri đầu vào m là chieu dài khung của so
%Giá trị đầu vào y là tín hiệu cần lọc
%Giá trị trả về z là tín hiệu sau khi lọc
```

D. Thuật toán tính sai số RMSE

Trong matlab sử dụng hàm `length()` để lấy độ dài, hàm `abs()` để lấy trị tuyệt đối, hàm `mean()` để tính trung bình, hàm `sqrt()` để lấy căn bậc 2.

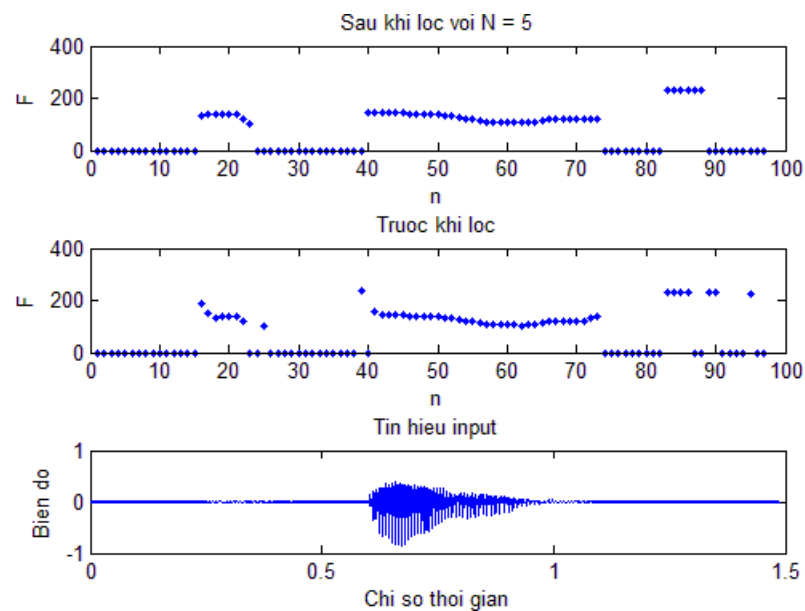
```
function z = RMSE(y,f)
    temp = y;
    for i = 1:length(y)
        temp =abs (y(i)-f)*abs (y(i)-f) ;
    end
    z = sqrt( mean(temp) );
end
% y la mang tan so chua cac tan so cua cac khung, f la tan so chuan
% tinh thu cong
% z la gia tri sai so
```

IV. KẾT QUẢ THỰC NGHIỆM

A. Hình vẽ

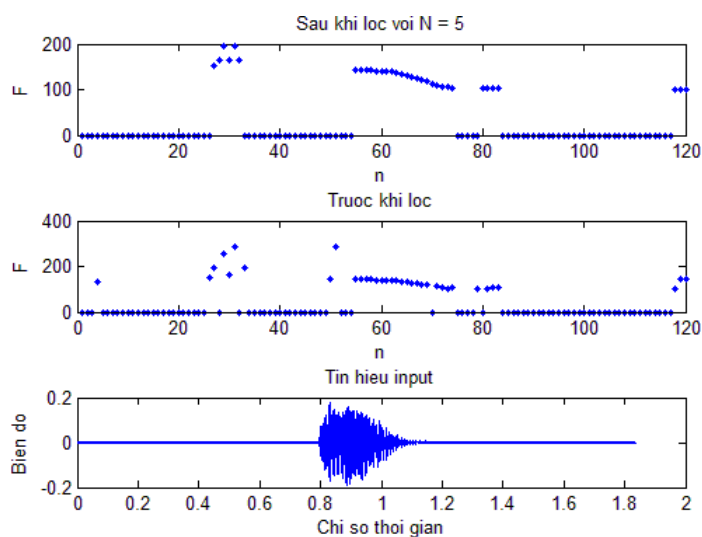
1. Hàm tự tương quan

- Trường hợp xấu nhất /u/ của Đinh Quang Duy



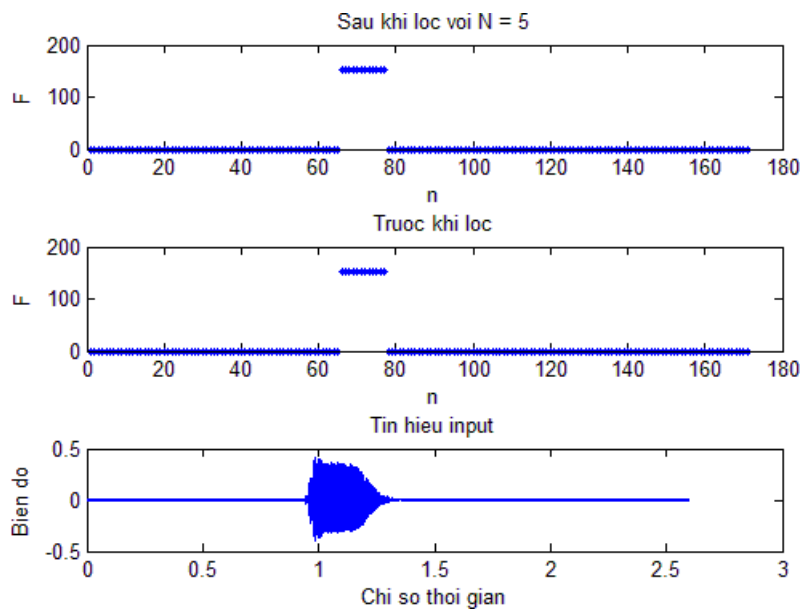
Hình 6. F0 của trường hợp xấu nhất (.bmp)

- Trường hợp trung bình /o/ của Đinh Quang Duy



Hình 7. F0 của trường hợp trung bình (.bmp)

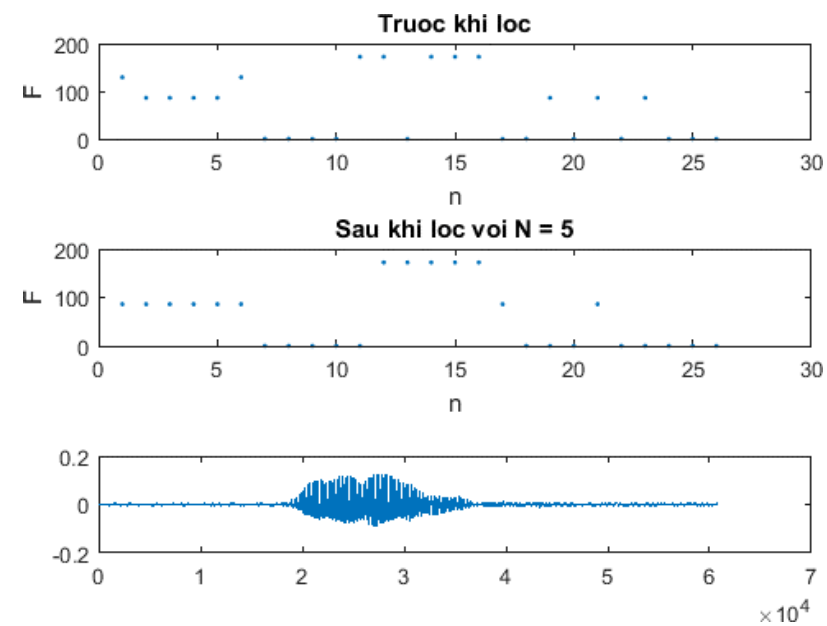
- Trường hợp tốt nhất /o/ của Nguyễn Xuân Lực



Hình 8. F0 của trường hợp tốt nhất (.bmp)

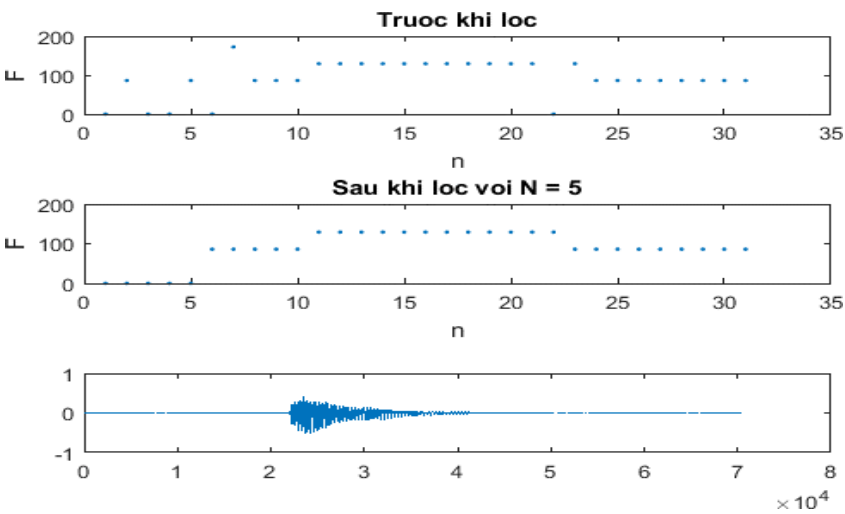
2. F0 trên miền tần số

- Trường hợp xấu /u/ của Hồ Gia Khánh



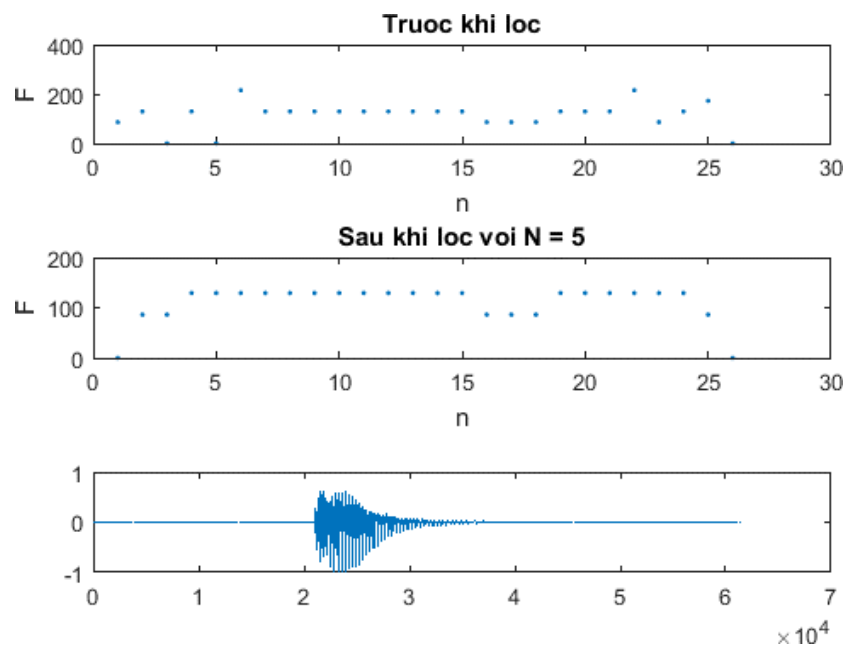
Hình 9. F0 của trường hợp xấu nhất (.bmp)

- Trường hợp trung bình /e/ của Đinh Quang Duy



Hình 10. F0 của trường hợp trung bình (.bmp)

- Trường hợp tốt nhất /a/ của Đinh Quang Duy



Hình 11. F0 của trường hợp tốt nhất (.bmp)

B. Bảng biểu

- Âm của Hồ Gia Khánh

Bảng 1: Kết quả đo được của Hồ Gia Khánh

| Âm | F0 chuẩn (Hz) | F0 trước lọc (Hz) | TTQ lọc | Sai số RMSE trước lọc | F0 sau lọc (Hz) | TTQ lọc | Sai số RMSE sau lọc | F0 FFT trước lọc (Hz) | Sai số RMSE trước lọc | F0 FFT sau lọc (Hz) | Sai số RMSE sau lọc |
|-----|---------------|-------------------|---------|-----------------------|-----------------|---------|---------------------|-----------------------|-----------------------|---------------------|---------------------|
| /i/ | 170.3 | 165.71 | | 9.7 | 165.57 | | 16.64 | 135.07 | 33.5 | 134.09 | 84.18 |
| /u/ | 175 | 182.29 | | 5 | 183.66 | | 71.23 | 133.19 | 84.17 | 133.19 | 88.87 |
| /o/ | 164.6 | 167.13 | | 7 | 167.04 | | 6.36 | 124.8 | 87.47 | 122.19 | 78.48 |
| /a/ | 162.7 | 178.53 | | 28.25 | 178.72 | | 43.83 | 139.97 | 45.8 | 139.97 | 76.57 |
| /e/ | 173.6 | 188.99 | | 193.9 | 166.82 | | 64.17 | 112.64 | 78.47 | 104.35 | 87.47 |

• Âm của Đinh Quang Duy

Bảng 2: Kết quả đo được của Đinh Quang Duy

| Âm | F0 chuẩn (Hz) | F0 TTQ trước lọc (Hz) | Sai số RMSE trước lọc | F0 TTQ sau lọc (Hz) | Sai số RMSE sau lọc | F0 FFT trước lọc (Hz) | Sai số RMSE trước lọc | F0 FFT sau lọc (Hz) | Sai số RMSE sau lọc |
|-----|------------------|-----------------------------|-----------------------------|---------------------------|---------------------------|-----------------------------|-----------------------------|---------------------------|---------------------------|
| /i/ | 134.86 | 154.5 | 12.14 | 152.84 | 11.65 | 140.26 | 5.56 | 139.87 | 5.25 |
| /u/ | 143.18 | 144.79 | 79.55 | 142.42 | 79.55 | 123.88 | 57.05 | 121.46 | 57.05 |
| /o/ | 138.68 | 145.56 | 7.83 | 136.47 | 39.13 | 123.77 | 9.48 | 121.83 | 9.48 |
| /a/ | 128.95 | 117.68 | 8.13 | 117.18 | 18.42 | 122.45 | 0.25 | 122.97 | 0.25 |
| /e/ | 134.45 | 130.65 | 64.2 | 126.64 | 44.9 | 120.78 | 48.32 | 119.3 | 48.72 |

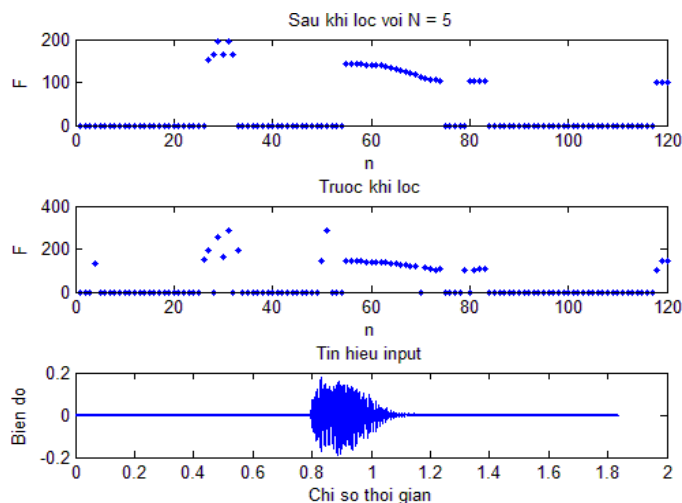
• Âm của Nguyễn Xuân Lực

Bảng 3: kết quả đo được của Nguyễn Xuân Lực

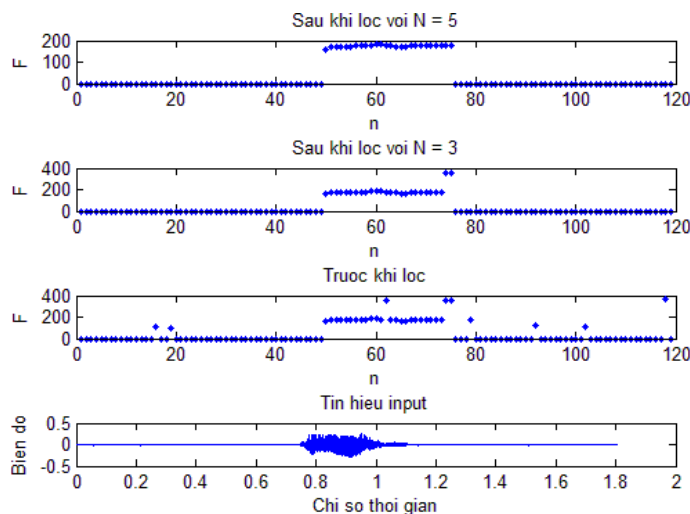
| Âm | F0 chuẩn (Hz) | F0 TTQ trước lọc (Hz) | Sai số RMSE trước lọc | F0 TTQ sau lọc (Hz) | Sai số RMSE sau lọc | F0 FFT trước lọc (Hz) | Sai số RMSE trước lọc | F0 FFT sau lọc (Hz) | Sai số RMSE sau lọc |
|-----|------------------|-----------------------------|-----------------------------|---------------------------|---------------------------|-----------------------------|-----------------------------|---------------------------|---------------------------|
| /i/ | 154.19 | 133.8 | 42.54 | 133.44 | 42.54 | 161.44 | 18.08 | 157.41 | 68.06 |
| /u/ | 160.95 | 152.36 | 60.95 | 145.16 | 60.95 | 149.62 | 74.82 | 138.52 | 31.75 |
| /o/ | 153.66 | 152.78 | 2.63 | 152.78 | 2.63 | 138.74 | 67.52 | 132.89 | 67.53 |
| /a/ | 141.35 | 133.03 | 27.69 | 132.96 | 27.69 | 114.41 | 12.15 | 100.27 | 55.22 |
| /e/ | 146.03 | 130.25 | 44.65 | 130.25 | 62.66 | 172.27 | 16.83 | 162.64 | 59.9 |

C. Nhận xét

- Thuật toán Lọc trung vị



Hình 12. Trường hợp trung bình của hàm tự tương quang (/o/ của Đinh Quang Duy) (.bmp)



Hình 13. Sự khác nhau giữa lọc $N=3$ và Lọc $N=5$ (/e/ của Hồ Gia Khánh) (.bmp)

Tham số ảnh hưởng nhất là chiều dài của cửa sổ. Hình 13 cho thấy khi chiều dài cửa sổ càng lớn thì mức độ làm trơn của tín hiệu càng lớn, nhưng sẽ làm tín hiệu sai lệch quá nhiều so với trường hợp chưa lọc.

Đối với hình 12. Trước khi lọc có nhiều điểm bị lệch ra. Sau khi lọc ta thấy những điểm bị lệch ra đã được đưa về sao cho phù hợp với tín hiệu.

Các bảng số liệu ở trên cho thấy, trước khi lọc đa phần âm có tần số cao hơn sau khi lọc, vì lọc trung vị đã lọc bớt các thành phần có biên độ cao.

- So sánh thuật toán tìm F0 trên miền thời gian dùng hàm tự tương quan và tìm F0 trên miền tần số

Khi sử dụng thuật toán tìm F0 trên miền thời gian sử dụng hàm tự tương quan cho thấy kết quả tần số F0 thu được gần giống với kết quả F0 tìm được ở bước thủ công, kết quả này cũng khá tốt so với khi tìm trên miền tần số

Ưu điểm:

- + Sử dụng hàm tự tương quan: trên kết quả thu được ta thấy thuật toán này có độ chính xác tốt hơn và sai số cũng nhỏ hơn
- + Trên miền tần số : đơn giản, dễ cài đặt, loại bỏ được tần số ảo

Nhược điểm:

- + Sử dụng hàm tự tương quan: tuy nhiên cũng có một vài trường hợp cho thấy kết quả sai số vẫn chưa được tốt vì tín hiệu thu vào thường bị nhiễu bởi âm thanh xung quanh, có nhiều tần số ảo
- + Trên miền tần số: sai số lớn, trên kết quả thu được ta thấy thuật toán này có độ chính xác không cao

V. KẾT LUẬN

Bài báo này thực hiện việc cài đặt thuật toán tìm tần số cơ bản của tín hiệu tiếng nói dùng hàm tự tương quan được cài đặt trên Matlab. Để làm tròn tần số cơ bản được xác định, lọc trung vị được sử dụng. Các thử nghiệm với tín hiệu của 5 nguyên âm (/a/, /e/, /i/, /o/, /u/) cho thấy tần số tính được từ thuật toán tìm tần số cơ bản F0 trên miền thời gian dùng hàm tự tương quan có kết quả gần giống với tần số chuẩn đo thủ công hơn thuật toán tìm tần số cơ bản F0 trên miền tần số.

VI. NHỮNG ĐIỀU ĐÃ HỌC ĐƯỢC

Qua bài thực hành trên chúng em một phần hiểu được cách sử dụng phần mềm Matlab cho việc tính toán và xử lý tín hiệu, chúng em còn biết thêm được 2 cách để tìm tần số cơ bản của một tín hiệu tiếng nói bất kì đó là: thuật toán tìm tần số cơ bản F0 trên miền thời gian sử dụng hàm tự tương quan và thuật toán tìm tần số cơ bản F0 trên miền tần số, và cách cài đặt chúng trong Matlab.

Ngoài ra chúng em còn học được các kỹ năng làm việc nhóm sao cho hợp lý.

VII. LỜI CẢM ƠN

Cảm ơn TS. Ninh Khánh Duy, giảng viên bộ môn Hệ thống nhúng, khoa Công nghệ thông tin, trường Đại học Bách Khoa, Đại học Đà Nẵng đã hướng dẫn và hỗ trợ chúng em trong quá trình học tập và tìm hiểu cách làm bài thực hành và báo cáo.

VIII. TÀI LIỆU THAM KHẢO

- [1] Bài giảng Xử lý tiếng nói, Học viện bưu chính viễn thông, 2010.
- [2] Prentice Hall –Digital Processing Of Speech Signals
- [3] Median filter, Wikipedia: https://en.wikipedia.org/wiki/Median_filter
- [4] Số trung vị, Wikipedia: https://vi.wikipedia.org/wiki/S%E1%BB%91_trung_v%E1%BB%8B
- [5] Tần số âm cơ bản, Wikipedia https://vi.wikipedia.org/wiki/Tần_số_âm_cơ_bản
- [6] Slide bài giảng xử lý tín hiệu số : <http://www.stu.edu.vn/uploads/documents/310810-112932.pdf>
- [7] Biến đổi Fourier nhanh : https://vi.wikipedia.org/wiki/B%E1%BA%BFn_%C4%91%E1%BB%95i_Fourier_nhanh
- [8] Cải tiến thuật toán tự tương quan tìm cao độ của tín hiệu đàn ghi-ta_2017.pdf -Nguyễn Bình Thiên, Ninh Khánh Duy, bài báo Fair 2017
- [9] “Digital signal processing - principles, algorithms and applications_1995.pdf”: