1. **Giới thiệu chung**

Ngày nay công nghệ ngày càng phát triển mọi người hướng tới tự động hóa những công việc chân tay, mà trên thực tế thì năng suất và hiệu suất của máy móc hơn hẳn con người trong tất cả các lĩnh vực như chế tạo máy, chế tạo linh kiện, vật liệu, điều khiển các thiết bị gia dụng. Một cách tự nhiên đó là vấn đề nảy sinh trong việc này đó là nhu cầu giao tiếp của con người với máy móc, đã có rất nhiều những nhà nghiên cứu đầu tư công sức và thời gian để xây dựng những hệ thống nhận dạng tiếng nói khác nhau. Đặc điểm chung của nhận dạng tiếng nói đó là sử dụng những đặc trưng của tiếng nói, hay là quá trình “trích đoạn đặc trưng” tiếp tục với những nghiên cứu đó em sẽ đi làm rõ đề tài “Tìm hiểu về nhận dạng người nói và áp dụng sử dụng tiếng nói như một khóa bí mật”.

Đề tài sử dụng ngôn ngữ lập trình là Matlab, công nghệ sử dụng ở đây là Matlab Guide để tạo giao diện cho ứng dụng. Các kiến thức XLTHS đã sử dụng là thuật toán trích đoạn đặc trưng MFCC (Cửa sổ, FFT, MEL, Cepstrum), VQ – Lượng tử hóa vector (`Thuật toán tạo chùm LBG), sử dụng các bộ lọc.

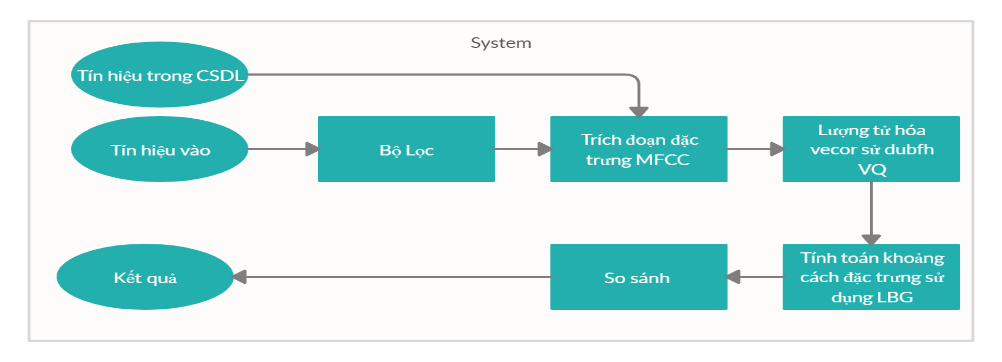
1. **Mô tả tài**

**Tính ứng dụng và tính khả thi của đề tài:** Ứng dụng có tính thực tiễn khá cao với nhu cầu về công nghệ vào cuộc sống ngày càng cao như bây giờ. Hệ thống nhận dạng tiếng nói (Automatic Speech Recognition – ASR) sẽ có những ứng dụng tuyệt vời trong tất cả các lĩnh vực của đời sống, nếu được áp dụng thành công sẽ là một cuộc cách mạng trong giao tiếp người máy (Human Machine Interface). Với đề tài này thì tính dụng khá thực tiễn tại các trường học, lớp học, công ty, xí nghiệp đông người sẽ giúp giảm thiểu nhân lực con người về việc kiểm tra, rà soát như. Ví dụ Giáo viên/ Giảng viên có thể biết được danh tính của học sinh thông qua giọng nói trong trường hợp học hộ, thi hộ tốt hơn.

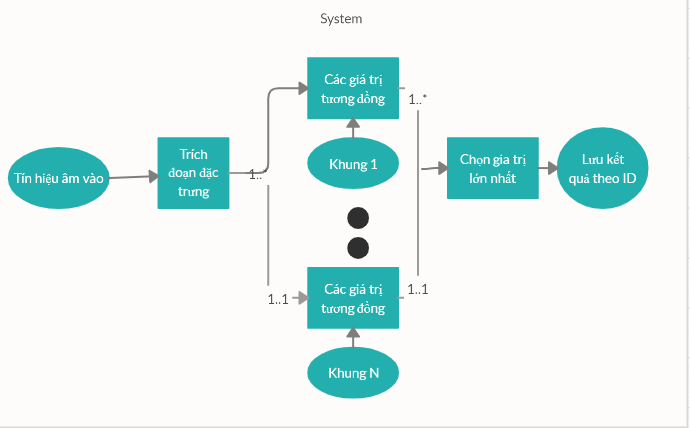
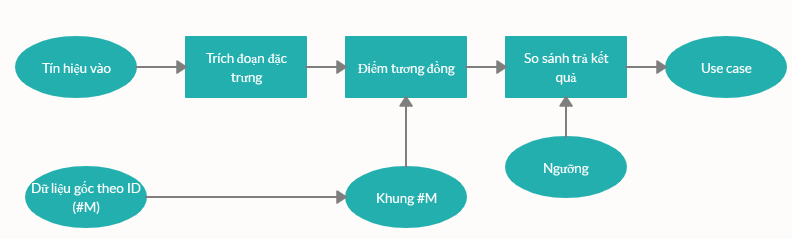
* Trong lĩnh vực điều khiển: Như Google Home, Google Assistant, điều khiển robot…
* Trong lĩnh vực chuyển đổi tín hiệu: Tự động biên dịch cuộc hội thoại ra văn bản, phiên dịch trong các cuộc họp có sự bất đồng về ngôn ngữ…
* Trong lĩnh vực nhân diện: Nhận diện danh tính, nhận diện tội phạm, làm khóa bảo mật, điểm danh, chấm công…

**Mục tiêu của đề tài:**

* Xây dựng được một hệ thống nhận diện tiếng nói, để tạo khóa bảo mật.
* Nghiên cứu và tìm hiểu về thuật toán mới MFCC, VQ.
* Biết cách phân tích và xây dựng một hệ thống.
* Tìm hiểu và hiểu được các khâu của một hệ thống sẽ đóng vai trò như thê nào.



**Hình 2: Sơ đồ khối của hệ thống**

**Hình 3: Tách đặc trưng Hình 4: Nhận diện**

* Như ở hai hình trên, hệ thống bao gồm 2 module chính: tách đặc trưng và nhận dạng đặc trưng
* Hình 3: tách các dữ liệu từ tiếng nói đặc trưng nhất của người nói vào.
* Hình 4: là quá trình nhận dạng người đang giao tiếp với hệ thống bằng cách so sánh các đặc trưng của người này với những người đã được huấn luyện.
* **Các thiết bị và công cụ cần sử dụng:**
* Thiết bị: Laptop, mic.
* Công cụ: Matlab, MySQL.
* **Các thuật toán xử lý tín hiệu chính sử dụng là:** MFCC, VQ, LBG, các bộ lọc fir, các hàm đều được tìm hiểu (đầu vào, đầu ra, công thức, khái niệm…) và được viết lại các hàm bằng Matlab.

**Tách đặc trưng:**

* Tách đặc trưng của mẫu là một phần quan trọng của bất cứ hệ thống nhận dạng nào. Một cách lý tưởng, một đối tượng khác nhau sẽ có một hoặc nhiều đặc trưng. Các đặc trưng càng khác nhau giữa các đối tượng thì việc nhận dạng càng chính xác.
* Việc nhận dạng sẽ dựa trên các đặc trưng này, có thể sử dụng 1 đặc trưng hoặc kết hợp nhiều đặc trưng lại với nhau. Với các hệ thống nhận diện giọng nói hiện nay, thường chỉ sử dụng một đặc trưng của tín hiệu âm thanh.
* Cho tới thời điểm hiện nay, các phương pháp chủ yếu để tách đặc trưng có thể kể đến như: Linear Prediction Coding (LPC), Mel – Frequency Cepstrum Coefficients (MFCC), Principle Components Analysis (PCA) và các phương pháp khác.

**Huấn luyện và nhận dạng:**

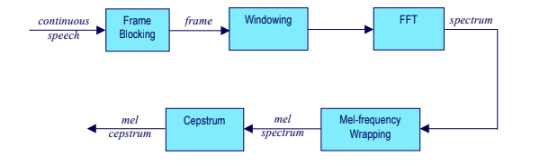
* Các đặc trưng sau khi được tạo thành, dù bằng bất cứ phương pháp nào cũng sẽ được dùng để huấn luyện tạo cơ sở dữ liệu và nhận dạng về sau.
* Các kĩ thuật chính được dùng trong việc nhận dạng âm thanh có thể kể đến là:

1. Dynamic Time Warping – DTW. Phương pháp này phổ biến trong thấp niên 1970 & 1980.
2. Vector Quantization – VQ. Phương pháp này cho kết quả khả quan hơn so với phương pháp DTW, trong phạm vi từ vựng nhỏ (khoảng 20 từ), đây có thể là phương pháp được quan tâm.
3. Hidden Markov Modeling/Artifitial Neural Network – HMM/ANN. HMM là kĩ thuật mới nhất, được dùng trong các hệ thống đòi hỏi số từ vựng rất lớn, lên đến hàng ngàn từ.

* Sau khi tìm hiểu các thuật toán phổ biến hiện nay, tôi đã chọn MFCC để tách các đặc trưng và

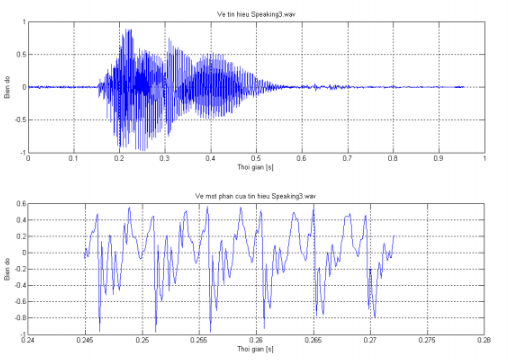
VQ là phương pháp huấn luyện và nhận dạng. Vì sự hiệu quả của nó, và không đòi hỏi khối lượng tính toán lớn.

**MFCC**

****

**Hình 5: Sơ đồ khối quá trình tách đặc trưng bằng MFCC**

1. **Record / Sampling**: Tai người thính nhất với các tín hiệu có tần số trong khoảng 100Hz – 5Khz, và thông thường với tín hiệu âm thanh, khu vực phổ này chiếm phần lớn năng lượng của âm thanh được phát ra. Để có được các năng lượng chủ yếu của tiếng nói con người, hệ thống của tôi sẽ lấy mẫu âm thanh ở tốc độ 12Khz, như vậy tín hiệu thu được sẽ mang tần số lên đến 6KHz.
2. **Frame Blocking**: Đặc điểm của tín hiệu tiếng nói là tín hiệu chậm biến đổi theo thời gian, hay còn được gọi là “quasi – stationary”. Khi xem xét một khoảng thời gian đủ ngắn, khoảng từ 5 – 100ms, tính chất của nó gần như là ổn định. Tuy nhiên, trong một khoảng thời gian dài, tính chất của nó bị thay đổi, phản ánh những âm thanh khác nhau được nói.



**Hình 6: Một ví dụ về tín hiệu tiếng nói**

**Hình trên là biểu diễn thời gian dài**

**Hình dưới là biểu diễn thời gian ngắn**

* Vì vậy phân tích khoảng thời gian ngắn (short time spectral analysis) thường được sử dụng

trong phân tích tính chất của tín hiệu tiếng nói. Trong khi một từ được phát ra có thể dài đến 1s, nên cần thiết phải chia các tín hiệu thu được thành các frame nhỏ, các frame này có độ dài tương ứng 5 –10ms.

* Để tránh sự thay đổi đột ngột giữa các frame, sẽ có sự lặp lại giữa các frame liên tiếp. Ví dụ, mỗi frame đầu tiên có N mẫu; frame kế tiếp cũng sẽ có N mẫu, nhưng chỉ có M ( M < N ) mẫu là “mới”, còn lại N – M mẫu, đầu tiên là N - M mẫu cuối cùng của frame đầu tiên, quá trình như vậy tiếp diễn cho những frame sau.
* Thông thường chọn N = 128, 256, 512 ... để thuận tiện cho việc tính FFT đằng sau này. Và chọn M ~ N/3.

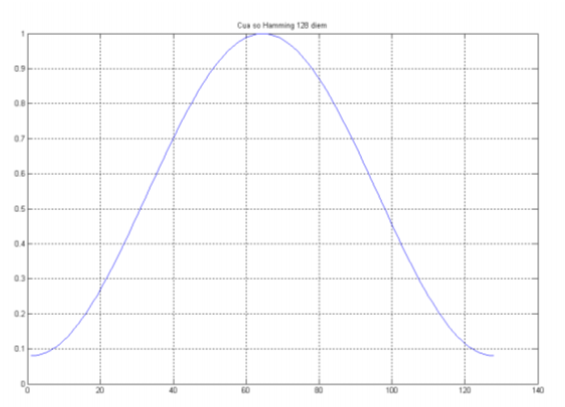
1. **Windowing.**

* Bước tiếp theo là cửa sổ hóa mỗi frame để giảm sự không liên tục giữa điểm đầu và điểm cuối của mỗi frame. Nguyên tắc của việc cửa sổ hóa là giảm thiểu méo dạng phổ bằng cách làm hẹp dần ở phía đầu và cuối của mỗi frame.
* Gọi cửa sổ là w(n), 0 ≤ n ≤ N – 1, với N là số phần tử trong mỗi frame. Tín hiệu sau khi cửa số hóa như sau:

yl(n) = xl(n) \* w(n) 0 ≤ n ≤ N – 1.

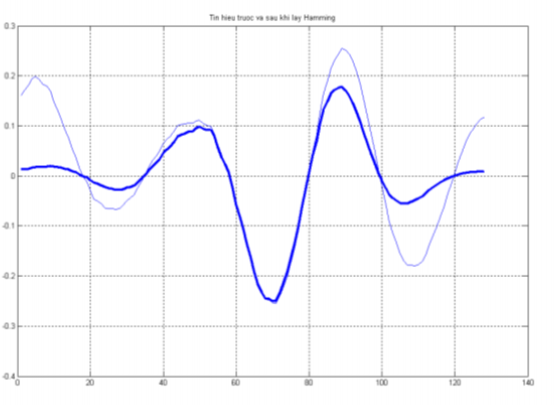
* Thông thường cửa sổ Hamming được dùng, có dạng như sau:

w(n)=0.54−0.46cos(2Πn/(N−1)), 0≤n≤N−1



**Hình 7: Cửa sổ Hamming 128 điểm**

* Tín hiệu sau khi Hamming hóa sẽ bị “thu nhỏ” lại ở hai đầu như hình vẽ 7.

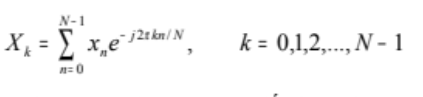


**Hình 8: Ảnh hưởng của cửa sổ Hamming lên tín hiệu**

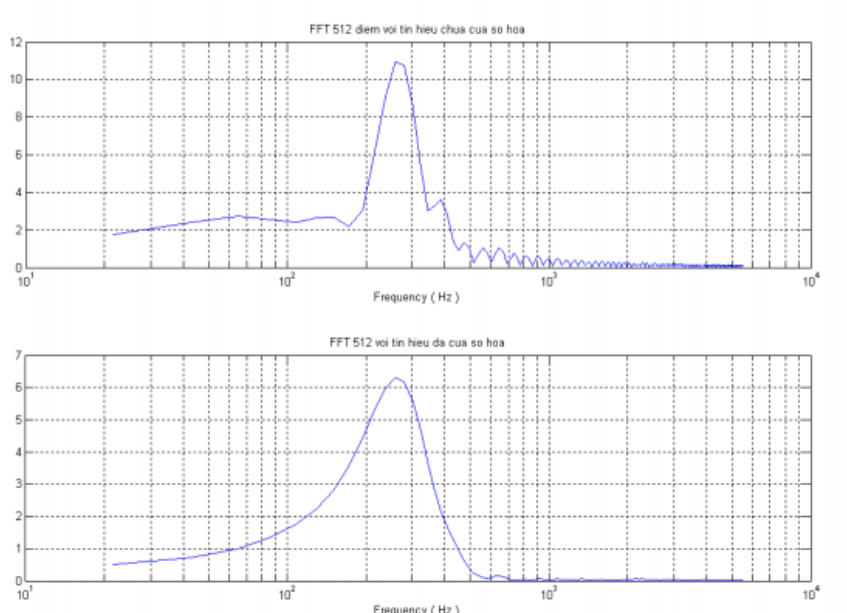
**thu nhỏ ở hai đầu**

1. **FFT**

* Mục đích của quá trình này là chuyển đổi tín hiệu thời gian thành tín hiệu tần số. FFT là giải thuật nhanh của phép biến đổi Fourier rời rạc (DFT).
* Công thức của FFT với N mẫu được tính như sau:



* Xk được tính từ công thức trên là số ảo, nhưng thông thường ta chỉ quan tâm đến biên độ của nó.
* Xk có thể chia làm 2 phần: phần tần số dương ứng với 0 ≤ f < Fs/ 2 tương đương với 0 ≤ n ≤ N / 2 − 1, và thành phần tần số âm – Fs / 2 < f < 0 tương đương với N / 2 + 1 ≤ n ≤ N − 1. Sau đây là kết quả khi sau bước này, đồng thời cũng thấy được tác dụng của việc cửa số hóa frame.



**Hình 9: Tín hiệu sau biến đổi FFT V và cửa sổ**

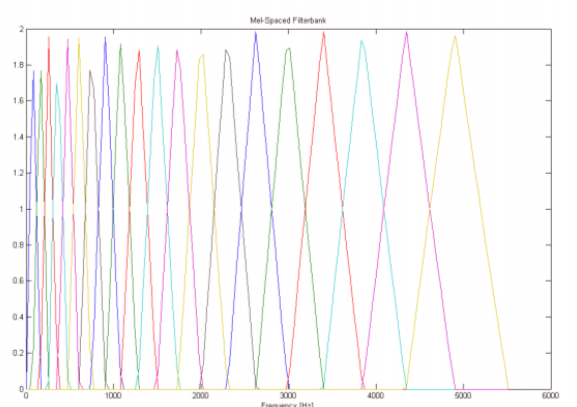
* Kết quả từ bước này là một dãy các biên độ phổ tần số của các frame liên tiếp nhau.

1. **Mel – frequency – Wrapping.**

* Một số nghiên cứu vật lý về tai người cho thấy phản ứng của tai người với tín hiệu tiếng nói không tuân theo quy luật tuyến tính về tần số. Vậy một cách tiếp cận chủ quan, mỗi tín hiệu âm thanh được phát ra sẽ đươc chuyển đổi lại cho phù hợp, lúc này tần số mel (mel frequency) được sử dụng. Mel – frequency tuyến tính ở tần số dưới 1Khz và logarithmic ở tần số trên 1Khz. Để tính toán mel – spectrum thì người ta dùng những cửa sổ lọc – filterbank Kết quả thu được đến bước này là một ma trận [số điểm hệ số mel – spectrum][số frame]. Các hệ số này được đưa vào bước cuối cùng để tìm Acoustic vector, là đặc trưng của giọng nói

.

* Trên thực tế, để tính toán mel – spectrum thì người ta dùng những cửa sổ lọc – filterbank, được xắp xếp một cách đồng điều như Hình 10 bên dưới. Filterbank là những bộ lọc thông dãy hình chữ nhật, khoảng cách cũng như băng thông của nó là những hằng số.

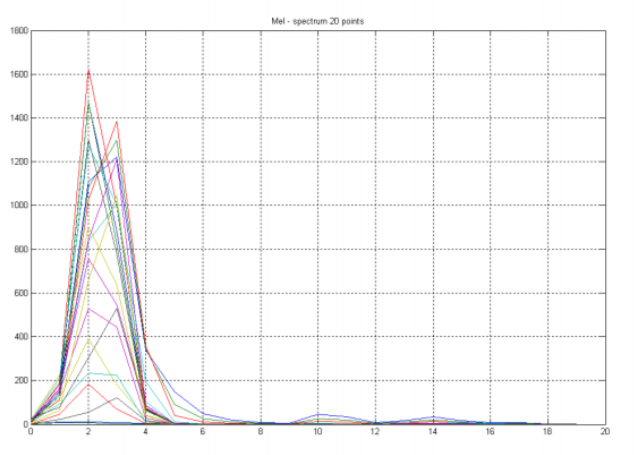


**Hình 10. Mel – spaced filterbank với 20 hệ số mel – spectrum.**

* Để xác định mel – spectrum, cho biên độ phổ tần số sau bươc FFT ở trên qua bộ lọc mel, với công thức tính như sau:



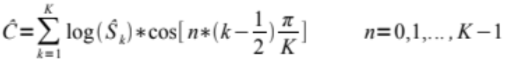
* Giả sử với L = 20, mel – spectrum sẽ có hình dạng như sau, với đầu vào khoảng 20 frames:



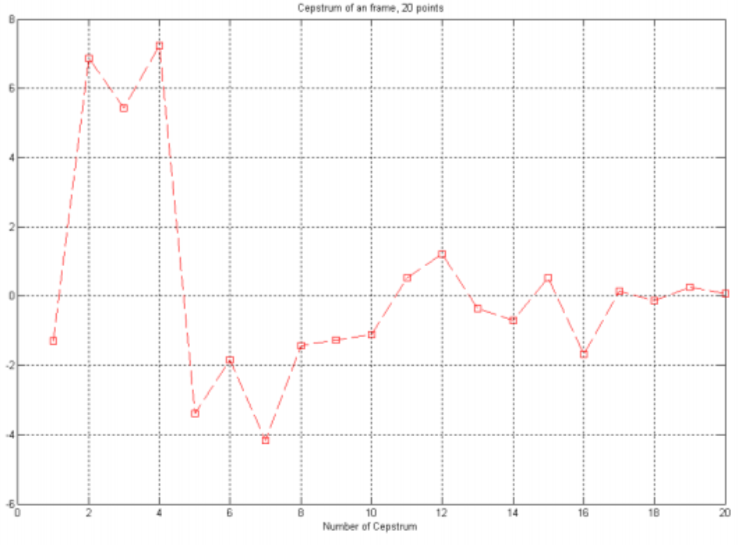
**Hình 11. Mel – spectrum 20 điểm của 20 frame liên tiếp.**

1. **Cepstrum**

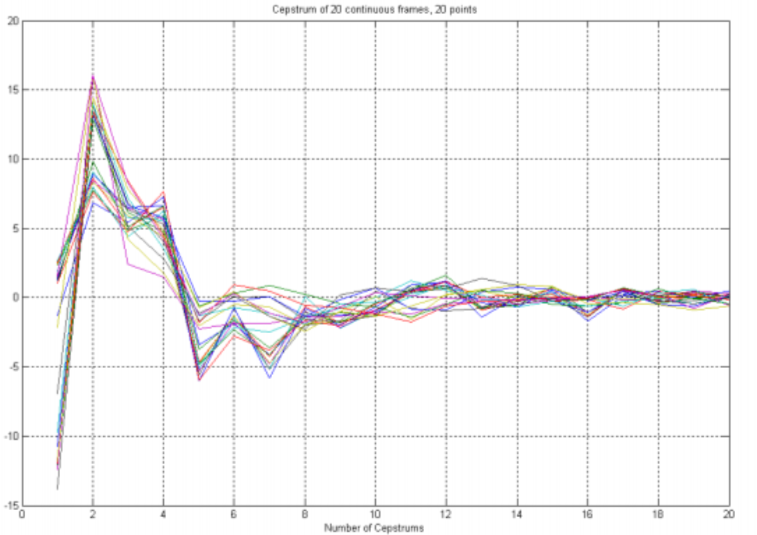
* Bước này chuyển đổi logarit của mel spectrum về trục thời gian, kết quả được gọi là “mel frequency cepstrum coefficients (MFCC) “. MFCC của phổ tiếng nói là đặc trưng cho đặc tính phổ biên độ của tín hiệu từ các frame.
* Phương pháp biến đổi ở đây dùng DCT (discrite cosin transform) thay vì IFFT, vì những đặc điểm sau:
* Tín hiệu là thực
* IFFT áp dụng cho tín hiệu là số phức, trong khi DCT là số thực.
* DCT có chức năng tương tự như IFFT, nhưng hiệu quả hơn do sử dụng số thực.
* Công thức tính:



* Kết quả sau bước này, ta có tín hiệu được gọi là Acoustic vector, mang đặc trưng của giọng nói thu được. Hình dạng Acoustic vector của 1 frame có thể như sau:



**Hình 12: Một ví dụ về Acoustic vector của 1 frame. Cepstrum có 20 điểm.**

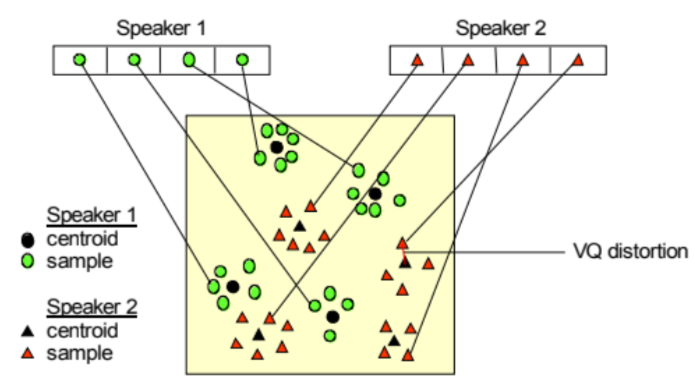
****

**Hình 13: Acoustic vector của 20 frame liên tiếp.**

* Đến đây, ta đã thu được đặc trưng của tiếng nói thu được, là một chuỗi các acoustic vector của các frame liên tiếp nhau. Các acoustic vector này sẽ được dùng để huấn luyện và nhận dạng giọng nói.

**VQ**

|  |  |
| --- | --- |
| VQ là phương pháp ánh xạ những vector trong một không gian lớn thành một số lượng hữu hạn các vector cũng nằm trong không gian đó. Mỗi vùng các không gian rộng lớn gọi là một bó (cluster) có thể đặc trưng bởi tâm của nó gọi là “codeworld”. Tập hợp các code word này gọi là “codebook”. | Tiếng nói nào tương ứng với tổng các khoảng cách Euclid đến một VQ codebook nào đó nhỏ nhất thì tương ứng với tiếng nói đã tạo nên VQ codebook đó. Khoảng cách Euclid giữa 2 vector n chiều a (a1, a2, …, an) và b (b1, b2, …, bn) được tính như sau:  C:\Users\Tu Nguyen\AppData\Local\Microsoft\Windows\INetCache\Content.Word\aqwe.jpg |

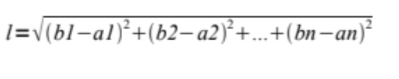
****

**Hình 14. Sơ đồ nguyên lý miêu tả thông tin của codebook. Âm thanh này có thể phân biệt với âm thanh khác nhờ vào vị trí của centroid**

* Trong hình vẽ này, chỉ có 2 giọng nói và 2 chiều của không gian acoustic vector được trình bày. Vòng tròn chỉ acoustic vector của người thứ nhất, hình chữ nhật chỉ acoustic vector của người thứ 2.
* Để nhận dạng, khoảng cách Euclid được dùng để tính khoảng cách từ mỗi acoustic vector đến codeword gần nhất của mỗi codebook đã được huấn luyện (nằm trong database). Tiếng nói nào tương ứng với tổng các khoảng cách Euclid đến một VQ codebook nào đó nhỏ nhất thì tương ứng với tiếng nói đã tạo nên VQ codebook đó.

**Khoảng cách Euclid.**

* Khoảng cách Euclid giữa 2 vector n chiều a (a1, a2, ..., an) và b (b1, b2, ..., bn) được tính như sau:



**Thuật toán tạo chùm LBG:** Mục đích của thuật toán tạo chùm là tạo mỗi VQ codebook cho mỗi thông tin thu được từ acoustic vector đã tạo ra từ phần trước. Các bước của thuật toán.

1. Khởi tạo 1 vector codebook, là trung tâm của tất cả các acoustic vector thu nhận được từ phần trước.
2. Nhân đôi kích thước của codebook bằng cách chia đôi codebook hiện hành yn theo quy tắc sau: (Yn)^+ = Yn (1 + ε)

(Yn)^- = Yn (1 - ε)

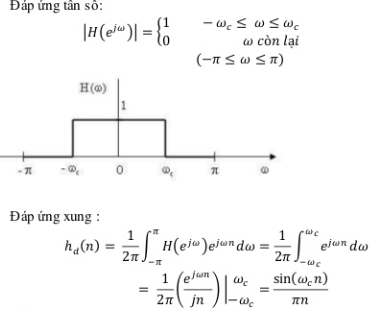
Với các n = 1 … (kích thước của codebook). ε: hệ số chia. Ví dụ ε = 0.01.

1. Nearest Neighbor Search: với các vector huấn luyện, ở đây là acoustic vector, tìm các codeword trong codebook hiện thời gần với nó nhất (theo khoảng cách Euclid), xắp xếp các vector này vào trong các cell tương ứng.
2. Centroid Update: cập nhật codeword trong mỗi cell ở bước 3, dùng centroid của mỗi acoustic vector tương ứng với cell đó, thông thường nhất là dùng trung bình cộng các acoustic vector của cell đó.
3. Lặp lại bước 3 & 4 cho đến khi khoảng cách trung bình nhỏ hơn ngưỡng cho trước.
4. Lặp lại bước 2, 3 & 4 cho đến khi kích thước của codebook đến giá trị mong muốn.

**Các bộ lọc:**

**Bộ lọc thông thấp:**

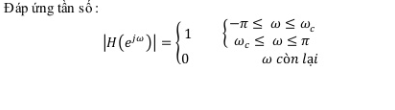
* Bộ lọc thông thấp là bộ lọc chuyển tín hiệu tần số thấp và chặn hoặc trở kháng, tín hiệu tần số cao. Nói cách khác, các tín hiệu tần số thấp đi qua dễ dàng hơn nhiều và với các tín hiệu tần số cao và điện trở thấp hơn sẽ khó khăn hơn nhiều, đó là lý do tại sao nó là một bộ lọc thông thấp.

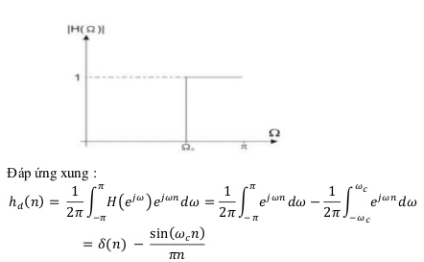
****

**(Trích dẫn trong giáo trình xử lý tín hiệu)**

**Bộ lọc thông cao:**

* Bộ lọc thông cao là bộ lọc chuyển tín hiệu tần số cao và chặn hoặc trở kháng, tín hiệu tần số thấp. Nói cách khác, các tín hiệu tần số cao đi qua dễ dàng hơn nhiều và với các tín hiệu tần số thấp và điện trở cao hơn sẽ khó khăn hơn nhiều, đó là lý do tại sao nó là một bộ lọc thông cao.

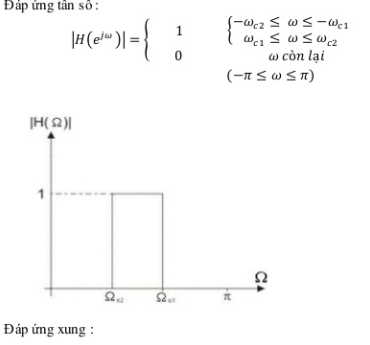
****

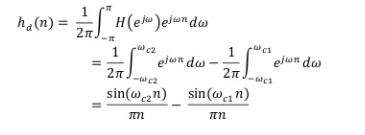
****

**(Trích dẫn trong giáo trình xử lý tín hiệu)**

**Bộ lọc thông dải:**

* Là [bộ lọc](https://vi.wikipedia.org/w/index.php?title=B%E1%BB%99_l%E1%BB%8Dc&action=edit&redlink=1) cho qua các [tần số](https://vi.wikipedia.org/wiki/T%E1%BA%A7n_s%E1%BB%91) trong một phạm vi nhất định và loại bỏ các [tần số](https://vi.wikipedia.org/wiki/T%E1%BA%A7n_s%E1%BB%91) bên ngoài phạm vi đó.





**(Trích dẫn trong giáo trình xử lý tín hiệu)**

* **Những khó khăn khi nhận dạng tiếng nói:**

**Sự biến động của người nói trong việc phát âm:**

* Tiếng nói thay đổi theo thời gian, theo độ tuổi.
* Tình trạng sức khỏe. Một người khi khỏe mạnh sẽ phát âm khác hẳn so với khi gặp ốm đau. Ví dụ như cảm cúm chẳng hạn.
* Tốc độ nói.
* Với một người, trong một khoảng thời gian ngắn, việc phát âm một từ trong nhiều lần khác nhau có thể khác nhau.

**Ảnh hưởng của ngoại cảnh:**

* Nhiễu, tiếng ồn của môi trường xung quanh. Ví dụ một người nói trong không gian yên tĩnh sẽ dễ nghe hơn là ở ngoài đường phố
* Handset để thu âm có thể khác nhau trong những tình huống khác nhau. Khoảng cách từ miệng người nói đến Handset.
* **Điều kiện lý tưởng cho việc thực hiện nhận dạng tiếng nói nói chung và âm thanh nói riêng là tiếng nói sẽ ổn định kể cả trong lúc huấn luyện và lúc nhận dạng. Tiếng nói của mỗi người là duy nhất, không trùng lẫn với những người khác. Do đó, cho đến thời điểm hiện tại, việc nhận dạng âm thanh, tiếng nói là một công việc rất khó khăn.**

**Các tham số đã được lựa chọn:**

* Tai người thính nhất với các tín hiệu có tần số trong khoảng 100Hz – 5Khz, và thông thường với tín hiệu âm thanh, khu vực phổ này chiếm phần lớn năng lượng của âm thanh được phát ra. Để có được các năng lượng chủ yếu của tiếng nói con người, hệ thống của tôi sẽ lấy mẫu âm thanh ở tốc độ 12Khz, như vậy tín hiệu thu được sẽ mang tần số lên đến 6Khz.
  + Fs= 12khz.
  + Dải thông của bộ lọc [100hz – 5Khz].
* Thử nghiệm để chọn kích thước của CodeBook

|  |  |  |
| --- | --- | --- |
| Kích thước của CodeBook | Hamming  % | Rectangular% |
| 1 | 57,14 | 57,14 |
| 2 | 75,4 | 66,4 |
| 4 | 81,5 | 76,5 |
| 8 | 88,7 | 80,7 |
| 16 | 95,78 | 85,78 |
| 32 | 100 | 90 |
| 64 | 100 | 95 |

* + Chọn lựa kích thước từ 32 là hợp lý và sử dụng cửa sổ Hamming đem lại kết quả cao hơn.

**Ưu điểm và các hạn chế của kết quả đồ án:**

**Ưu điểm:**

* Ưu điểm của bài toán là tính thời gian thực, chương trình có khả năng huấn luyện giọng nói mới.
* Chương trình có thể cấu hình để chạy trong các môi trường nhiễu khác nhau.
* Đơn giản và độ chính xác khá cao.
* Không yêu cầu cao về bộ nhớ thiết bị và trình xử lý cao.
* Phù hợp khi nhận dạng với các đoạn âm thanh ngắn.

**Nhược điểm:**

* Số lượng từ ngữ bị hạn chế, không nhận dạng được với những đoạn âm thanh dài.
* Trong môi trường nhiều tạp âm bị hạn chế.
  + Để nâng cao chất lương ta cần có bước tiền xử lý tốt hơn, giọng người nói và huấn luyện không có sự biến động nhiều, lựa chọn giải thuật nhận diện tiên tiến hơn như HMM, CNN.

1. **Danh sách tài liệu tham khảo**

[1] Y. Linde, A. Buzo &amp; R. Gray, “An algorithm for vector quantizer design”, IEEE Transactions on Communications, Vol. 28, pp.84-95, 1980.

[2] Lê Tiến Thường (2002). Xử lý số tín hiệu và Wavelets-Tập 1, NXB Đại học Quốc gia TP Hồ Chí Minh.

[3] Phạm Văn Tuấn (2011). Bài giảng nhận dạng tiếng nói, Đại học Bách khoa Đà Nẵng.

[4] Nguyễn Hồng Quang. Nhận dạng tiếng nói tiếng Việt, tìm hiểu và ứng dụng.

[5] Trương Trung Kiên. Slide bài giảng Xử lý tín hiệu số, Học viện công nghệ bưu chính viễn thông.

[6] Lê Duy Tân (2015). Nhận dạng giọng nói MFCC, Đại học Bách Khoa Hà Nội.

[7] Nguyễn Đức Thành (2004). MATLAB và ứng dụng trong điều khiển.NXB ĐHQG TPHCM.

[8] Hà Thúc Phùng. Voice Recognition System*.*