

BỘ HIỆU CHỈNH ÂM THANH

Phạm Minh Thắng - 21021638

Vũ Văn Nghĩa - 21021621

Mai Thanh Tùng - 21021645

Tóm tắt nội dung—Dự án này nhằm xây dựng một bộ hiệu chỉnh âm thanh có đồ họa sử dụng MATLAB. Bộ hiệu chỉnh này được thiết kế để xử lý âm thanh theo thời gian thực từ nhiều loại tệp âm thanh. Mục tiêu cuối cùng là tạo ra một bộ hiệu chỉnh âm thanh với các băng tần ứng với một phần ba quãng tám, được điều khiển thông qua 31 tần số, đảm bảo sự chính xác và linh hoạt.

I. GIỚI THIỆU CHUNG

Hiệu chỉnh âm thanh là việc điều chỉnh các tần số trong một tín hiệu điện sao cho phù hợp với nhu cầu sử dụng. Công cụ hỗ trợ quá trình này được gọi là Bộ hiệu chỉnh âm thanh (**Equalizer**). Trong thời đại mà các nền tảng nhạc số đang phát triển mạnh mẽ, nhu cầu tinh chỉnh âm thanh để phù hợp với không gian nghe nhạc hoặc sở thích cá nhân ngày càng trở nên phổ biến. Vì vậy, Equalizer đã trở thành một công cụ quen thuộc, đặc biệt được sử dụng rộng rãi trong ngành công nghiệp âm nhạc.

Có hai cách để phân loại Equalizer. Cách thứ nhất là phân loại dựa trên loại tín hiệu mà Equalizer hoạt động (tín hiệu số hoặc tín hiệu tương tự). Cách thứ hai là phân loại thành Equalizer có tham số (**Parametric Equalizer**) và có đồ họa (**Graphic Equalizer**). Cách phân loại này liên quan đến cách và mức độ mà các bands có thể được điều khiển.

Khi thu âm hoặc phát lại một bản nhạc, loại Equalizer được sử dụng phổ biến nhất là Equalizer đồ họa (Graphic Equalizer). Loại này bao gồm các thanh trượt (crossfader) có thể điều chỉnh thủ công, mỗi thanh đại diện cho một tần số cụ thể. Người dùng có thể sử dụng công cụ này để tùy chỉnh âm thanh theo ý muốn. Ví dụ, khi tăng cường các tần số thấp (bass), biên độ của chúng sẽ được nâng lên, làm cho âm bass trở nên rõ ràng và mạnh mẽ hơn. Hiệu ứng này sẽ càng rõ nét hơn nếu đồng thời giảm biên độ của các tần số cao trong cùng một tín hiệu.

Các băng tần của bộ hiệu chỉnh âm thanh bằng tham số (**Parametric Equalizer**) có thể thay đổi các thông số như âm lượng, tần số trung tâm và băng thông trong khi Equalizer có đồ họa chỉ cho phép thay đổi âm lượng. Chính điều này làm cho bộ hiệu chỉnh âm thanh bằng tham số vượt trội hơn bộ hiệu chỉnh âm thanh đồ họa về khả năng và cả độ linh hoạt [1].

Tuy nhiên, nếu xét đến yếu tố thân thiện với người dùng, bộ hiệu chỉnh âm thanh có đồ họa được coi là ưu việt hơn. Đặc biệt đối với những người không phải chuyên nghiệp, bộ hiệu chỉnh âm thanh bằng tham số sẽ rất khó để sử dụng. Điều này

một phần do sự đánh đổi cơ bản giữa sự đơn giản và khả năng kiểm soát các chi tiết, ta chỉ có thể đạt được một trong hai.

II. LÝ THUYẾT

A. Tín hiệu số và tín hiệu tương tự

Tín hiệu âm thanh mà tai người cảm nhận được có thể được mô hình hóa thành tín hiệu tương tự, liên tục cả về thời gian và biên độ. Tuy nhiên, máy tính không thể lưu trữ vô hạn các giá trị, điều này không cho phép máy tính lưu trữ tín hiệu liên tục theo thời gian. Do đó, các tín hiệu tương tự phải được biến đổi thành tín hiệu số tương ứng bằng cách rời rạc hóa tín hiệu tương tự rồi sau đó xấp xỉ các giá trị rời rạc. Về cơ bản việc rời rạc hóa và tính xấp xỉ là điều tất yếu khi lưu trữ tín hiệu vào trong một máy tính và là điều kiện tiên quyết để xử lý tín hiệu số [2].

B. Phương pháp lấy mẫu

Quá trình biến đổi tín hiệu tương tự thành tín hiệu số được gọi là lấy mẫu. Lấy mẫu của một tín hiệu liên tục $x(t)$ nghĩa là xấp xỉ và ghi lại giá trị của tín hiệu tại các thời điểm $t = nT$. Khi xử lý tín hiệu âm thanh, phải chọn tần số lấy mẫu thích hợp để tránh hiện tượng chồng phổ. Định lý lấy mẫu Nyquist - Shannon phát biểu rằng tốc độ lấy mẫu tối thiểu cần thiết phải lớn hơn gấp đôi thành phần tần số lớn nhất của tín hiệu được lấy mẫu [2]. Theo hệ quả của định lý này, tín hiệu âm thanh thường được lấy mẫu trên 44kHz để lớn hơn hai lần ngưỡng mà tai người có thể nghe được (khoảng 20kHz).

C. Biến đổi Z

Bất kể là tín hiệu số hay tương tự, các bộ lọc đều không được thiết kế trong miền thời gian, thay vào đó tín hiệu được chuyển đổi sang miền tần số để truy cập và thao tác với các thành phần tần số. Đối với tín hiệu số, việc chuyển đổi này được thực hiện bằng phép biến đổi Z. Đối với chuỗi rời rạc $x(k)$, biến đổi Z của nó được định nghĩa là

$$X(z) = \sum_{k=-\infty}^{\infty} x[k]z^{-k} \quad (1)$$

Trong đó, k là số nguyên và $z = Ae^{jw}$ với mọi số thực A và w sao cho tổng trong phương trình (1) hội tụ [2]. A và w lần lượt biểu thị cho độ lớn và đối số phức của z . Ta có $x[k-n]$ trong miền thời gian được biểu diễn bằng $z^{-n}X(z)$ trong miền z . Do đó, các phương trình sai phân trong miền thời gian được chuyển thành phương trình đại số trong miền z .

D. Bộ lọc FIR

Thuật ngữ FIR (đáp ứng xung hữu hạn) dùng để chỉ các bộ lọc số mà hàm truyền của nó chỉ phụ thuộc vào sự kết hợp tuyến tính của các tín hiệu đầu vào [2]. Hàm truyền $H(z)$ của bộ lọc FIR được cho bởi

$$H(z) = \sum_{n=0}^{N-1} b_n z^{-n} \quad (2)$$

Trong đó, b_n là hệ số cho mỗi n và $N < \infty$. Do đó, phương trình đầu ra của bộ lọc FIR là

$$y[k] = \sum_{n=0}^{N-1} b_n x[k-n] \quad (3)$$

E. Bộ lọc IIR

Ngược lại với bộ lọc FIR, hàm truyền của bộ lọc IIR phụ thuộc vào cả tín hiệu đầu vào và đầu ra trước đó [2]. Hàm truyền của bộ lọc IIR được cho bởi

$$H(z) = \frac{\sum_{n=0}^{N-1} b_n z^{-n}}{1 + \sum_{m=1}^M a_m z^{-m}} \quad (4)$$

Phương trình đầu ra tương ứng thu được bằng cách chuyển phương trình (4) sang miền thời gian bằng biến đổi Z ngược

$$y[k] = \sum_{n=0}^{N-1} b_n x[k-n] - \sum_{m=1}^M a_m y[k-m] \quad (5)$$

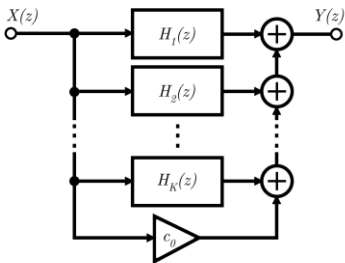
III. PHƯƠNG PHÁP

A. Thiết kế bộ lọc

1) *Cấu trúc bộ lọc*: Hàm truyền của bộ lọc được xác định bởi công thức

$$H(z) = c_0 + \sum_{k=1}^K \frac{b_{k,0} + b_{k,1}z^{-1}}{1 + a_{k,1}z^{-1} + a_{k,2}z^{-2}} \quad (6)$$

Trong đó, K là số bộ lọc được sắp xếp song song và c_0 biểu diễn hệ số khuếch đại. Công thức này được minh họa bằng sơ đồ trong Hình 1, trong đó $X(z)$ là tín hiệu vào, $Y(z)$ là tín hiệu ra.



Hình 1. Cấu trúc bộ lọc song song

2) Xác định hệ số của mẫu số:

$$\theta_k = \frac{2\pi f_k}{f_s}, \quad k = 1, 2, \dots, K \quad (7a)$$

$$p_k = e^{-\frac{\Delta\theta_k}{2}} e^{\pm j\theta_k}. \quad (7b)$$

Trong đó, θ_k là các tần số cực được xác định bởi các tần số tương tự f_k và tần số lấy mẫu f_s cho trước. Bảng thông $\Delta\theta_k$ của bộ lọc bậc hai thứ k^{th} thu được từ hai tần số cực liền kề với nó [3] và được tính theo công thức

$$\Delta\theta_k = \frac{\theta_{k+1} - \theta_{k-1}}{2}, \quad (8)$$

$$\Delta\theta_1 = \theta_2 - \theta_1,$$

$$\Delta\theta_K = \theta_K - \theta_{K-1}.$$

Hệ số của mẫu trong mỗi hàm truyền được xác định bởi

$$a_{k,1} = -2|p_k| \cos \theta_k, \quad (9a)$$

$$a_{k,2} = |p_k|^2 \quad (9b)$$

3) *Xác định hệ số của tử số*: Khi đã xác định được các hệ số của mẫu từ các nghiệm cực, bài toán sẽ trở nên tuyến tính

$$h = Mb, \quad (10)$$

Các cột của ma trận M bao gồm đáp ứng tần số của các mẫu số có bậc hai $1/(1 + a_{k,1}e^{-jw_n} + a_{k,2}e^{-j2w_n})$ và độ trễ tương ứng của chúng $e^{-jw_n}/1 + a_{k,1}e^{-jw_n} + a_{k,2}e^{-j2w_n}$ với tần số góc w_n ($2\pi \text{rad/sample}$). Cột cuối cùng của ma trận M gồm hệ số khuếch đại c_0 , do đó các phần tử của nó đều là 1. $h = [H(w_1) \dots H(w_n)]^T$ là vector cột chứa các kết quả của đáp ứng tần số.

$$M = \begin{pmatrix} \frac{1}{\text{den}(1,1)} & \frac{e^{-j\omega_1}}{\text{den}(1,1)} & \dots & \frac{1}{\text{den}(1,K)} & \frac{e^{-j\omega_1}}{\text{den}(1,K)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{1}{\text{den}(N,1)} & \frac{e^{-j\omega_N}}{\text{den}(N,1)} & \dots & \frac{1}{\text{den}(N,K)} & \frac{e^{-j\omega_N}}{\text{den}(N,K)} \end{pmatrix} \quad (11)$$

Trong trường hợp một hệ thống chưa được xác định, có ít nhất một cách chọn b sao cho khi nhân với M sẽ thu được kết quả là một vector đáp ứng tần số chính xác. Tuy nhiên, khi có nhiều phương trình hơn số ẩn, điều này có thể dẫn đến phương trình vô nghiệm.

Trong trường hợp đó, tập hợp các hệ số tối ưu b_{opt} của tử số sẽ tạo ra sự trùng khớp nhất có thể giữa vector đáp ứng tần số h và vector đáp ứng mục tiêu h_t . Vector này thu được bằng phương pháp Bình phương nhỏ nhất (Least squares)

$$b_{opt} = M^+ h_t \quad (12a)$$

$$M^+ = (M^H M)^{-1} M^H \quad (12b)$$

Với ”+” ký hiệu cho nghịch đảo giả Moore-Penrose và ”H” ký hiệu cho ma trận chuyển vị liên hợp. Lúc này, phương pháp Bình phương nhỏ nhất sẽ tối thiểu hóa sai số bằng công thức

$$e_{LS} = \sum_{n=1}^N |H(e^{jw_n}) - H_t(w_n)|^2 \quad (13)$$

Vì M không bị ảnh hưởng khi thay đổi h_t nên nó có thể được tính toán trước và lưu trữ, còn lại phép nhân ma trận là thao tác duy nhất cần thiết để xác định b_{opt} . Điều này làm giảm thiểu đáng kể khối lượng tính toán.

Tất cả các phần tử trong phương trình (12a) đều là số phức nên đòi hỏi các phép toán ma trận phức tạp. Tuy nhiên, phương trình (12a) có thể được xử lý sao cho chỉ chứa các phần tử thực và thu được các hệ số thực ở tử số. Bằng cách tách riêng phần thực và phần ảo trong ma trận M^+ và h_t khi xây dựng đáp ứng mục tiêu, các phương trình được tạo ra cho riêng từng phần thực và ảo. Điều này mang lại một hệ thống tuyến tính như sau

$$b_{opt} = M_r^+ h_{t,r} \quad (14)$$

Trong đó,

$$M_r = \begin{pmatrix} Re\{M\} \\ Im\{M\} \end{pmatrix}, \quad (15a)$$

$$h_{t,r} = \begin{pmatrix} Re\{h_t\} \\ Im\{h_t\} \end{pmatrix}. \quad (15b)$$

Cả hai ma trận này đều chỉ chứa các phần tử là số thực. Hàm truyền với hệ số thực có đáp ứng xung thực, làm cho đáp ứng tần số đối xứng liên hợp [3]. Vì ma trận M có phần thực và phần ảo được đặt song song nên hệ thống tuyến tính mới sẽ thu được số phương trình (commands) gấp đôi trong khi vẫn giữ nguyên các hệ số ở mẫu (số ẩn). Do đó, kích thước của ma trận mới là $(2N, 2K+1)$.

4) *Trọng số*: Các điểm tần số cũng có thể được gán trọng số khác nhau trong quá trình giảm thiểu sai số LS, trong trường hợp đó lỗi là

$$e_{WLS} = \sum_{n=1}^N W(w_n) |H(e^{jw_n}) - H_t(w_n)|^2 \quad (16)$$

Trong đó, $W(w_n)$ là trọng số không âm đối với tần số w_n [3]. Việc triển khai trọng số trong việc ước lượng tham số hiệu quả nhất về mặt tính toán là nhân tất cả các phần tử của ma trận M , và vector mục tiêu $h_{t,r}$ tương ứng với tần số w_n , với $\sqrt{W(w_n)}$ trước khi thực hiện các phép toán ma trận. Tuy nhiên, nhược điểm của việc tính trọng số, do hàm trọng số phụ thuộc vào đáp ứng mục tiêu, là ma trận M không thể được tính toán trước và lưu trữ nữa. Bởi vì khi nó phụ thuộc vào đáp ứng mục tiêu, việc thay đổi đáp ứng mục tiêu yêu cầu phải tính toán lại ma trận M .

B. Thiết kế bộ hiệu chỉnh âm thanh

1) *Phân bố và bậc của bộ lọc*: Sự tương đồng giữa tần số và đáp ứng mục tiêu chủ yếu được tạo ra bởi lượng lớn các tần số cực. Tuy nhiên, để không ảnh hưởng đến hiệu quả tính

toán, con số này phải được giữ trong một giới hạn hợp lý. Bộ hiệu chỉnh âm thanh được trình bày trong báo cáo này có 31 bands được phân bố theo thang logarit trên phổ tần số từ 20Hz đến 20kHz. Tuy nhiên, con số 31 này không thể được chọn một cách tùy tiện, điều này xuất phát từ việc mỗi band tương ứng với một phần ba của quãng tám với hệ số phân tách giữa các band là $\sqrt[3]{2}$. Do đó, số lượng phép nhân để có thể bao quát toàn bộ phổ tần số từ 20Hz đến 20kHz là 31.

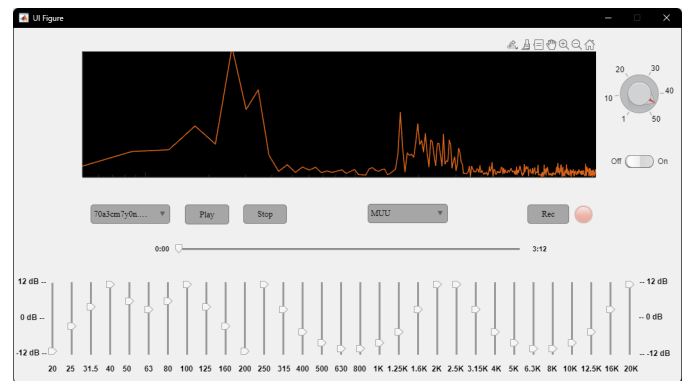
Sau khi xác định số mức tăng âm lượng, số điểm cực sẽ được giữ cố định. Con số này sẽ được đặt gấp đôi so với mức tăng âm lượng. Trong bộ hiệu chỉnh âm thanh này, tổng số bậc của bộ lọc sẽ nhiều hơn 1 so với số điểm tần số, tương ứng là 124 và 123. Thông thường, để tránh hiện tượng đáp ứng dao động giữa điểm tần số và điểm tần số đích, bậc của bộ lọc sẽ được giữ thấp hơn so với số điểm tần số [4].

2) *Trọng số của các điểm tần số*: Đáp ứng tần số được tăng cường bằng cách gán cho các điểm tần số một hàm trọng số thích hợp $W_n = 1/|H_t(w)|^2$ [3]. Ma trận M và đáp ứng mục tiêu được nhân với hệ số trọng số $\sqrt{W(w_n)} = 1/|H_t(w)|$ trước khi thực hiện các phép toán ma trận. Điều này giúp giảm thiểu sai số liên quan đến độ lệch tương đối của đáp ứng biên độ tần số với đáp ứng biên độ mục tiêu.

IV. KẾT QUẢ

A. Giao diện và tính năng

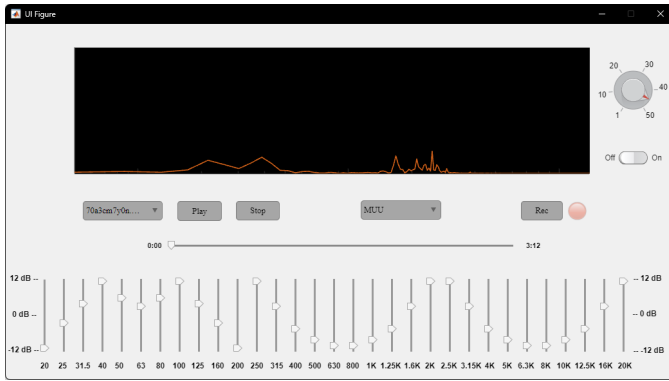
Bộ hiệu chỉnh âm thanh sẽ có giao diện như hình dưới. Ngoài 31 thanh trượt (crossfader), bộ hiệu chỉnh âm thanh được thiết kế thêm một vài tính năng khác để làm cho nó dễ sử dụng và trông bắt mắt hơn. Đoạn nhạc có thể bị tạm dừng, dừng hoặc chọn một file âm thanh khác bất kỳ lúc nào. Trong khi đang được phát, biên độ của các tần số đều được biểu diễn trong thời gian thực. Ngoài ra, có 13 tùy chọn khác được cài đặt sẵn cho phép người dùng lựa chọn để cá nhân hóa theo từng thể loại.



Hình 2. Giao diện chính của bộ hiệu chỉnh âm thanh

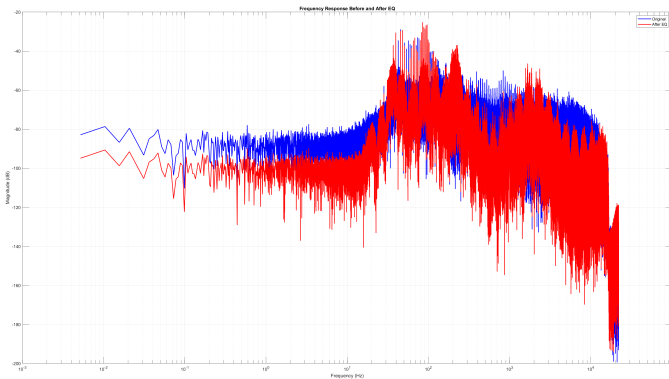
B. Đánh giá khả năng hoạt động của bộ hiệu chỉnh âm thanh EQ

Để đánh giá khả năng hoạt động của bộ hiệu chỉnh âm thanh EQ, các thanh trượt đã được đặt theo giá trị như sau đây:

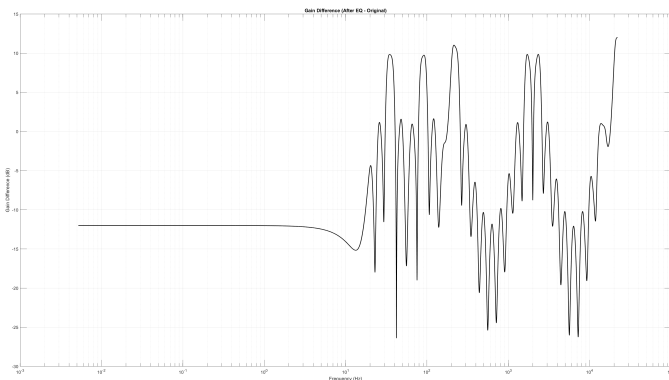


Hình 3. Giá trị đáp ứng biên độ của các băng tần được đánh giá

Đường màu xanh thể hiện phổ tần số của tín hiệu gốc (trước EQ). Đường màu đỏ thể hiện phổ tần số của tín hiệu sau khi áp dụng EQ. Rõ ràng, sau khi áp dụng EQ, phổ tín hiệu đã có sự thay đổi tại một số dải tần cụ thể. Điều này phản ánh rằng các bộ lọc của hệ thống EQ đã tác động đến các băng tần. Phổ tín hiệu sau EQ phù hợp với các giá trị gain mong muốn, nhưng vẫn còn sự dao động nhỏ do hiệu ứng nhiễu hoặc các đặc tính của bộ lọc.



Hình 4. Đáp ứng tần số trước và sau khi thay đổi giá trị EQ



Hình 5. Độ chênh lệch biên độ tín hiệu trước và sau khi qua EQ

Về độ chính xác, bộ hiệu chỉnh âm thanh này đem lại hiệu suất khá tốt khi tại các mức tần số được điều chỉnh biên độ

thì đều đã có sự thay đổi so với tín hiệu gốc ban đầu. Tuy nhiên, thông qua 5, ta có thể thấy sai số về độ lợi (gain) giữa giá trị được đặt và giá trị thực tế nhận được vẫn khá cao, giá trị sai số độ lợi trung bình là 4.6068dB.

TÀI LIỆU

- [1] Vesa Välimäki, Joshua D. Reiss. All About Audio Equalization: Solutions and Frontier, Department of Signal Processing and Acoustics, Aalto University, May 2016.
- [2] Steffi Knorn, Signals and Systems, Signals and Systems at department of engineering sciences at Uppsala University, 2018.
- [3] J. Rämö, V. Välimäki and B. Bank. High-Precision Parallel Graphic Equalizer. IEEE/ACM Transactions on Audio, Speech and Language Processing, Vol. 22, No. 12, pp. 1894-1904, December 2014. DOI: 10.1109/TASLP.2014.2354241.
- [4] T. W. Parks and C. S. Burrus, Digital Filter Design. New York: John Wiley & Sons, Inc., 1987.

Link github sản phẩm: [Tại đây](#)