|  |  |  |  |
| --- | --- | --- | --- |
| **LẬP TRÌNH ÂM THANH**  **Spring 2024-D21PT** | | | |
| Nhóm Bài tập lớn: SNDPROG24PRJ001 | | Bài Project: #01 | |
| Mai Thành Đạt **(NT)** | B21DCPT072 | | 10 |
| Trần Xuân Lâm | B21DCPT138 | | 10 |
| Trần Tuấn Linh | B21DCPT145 | | 10 |
| Mức độ hoàn thành bài tập của nhóm: 95% | | | |
| Đánh giá cụ thể | | | |
| Lớp / Hàm | Cài đặt | | File Script sử dụng |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |

**Đóng góp của các thành viên**

- Nhóm làm việc trực tiếp thông qua những buổi họp nhóm trên trường. Cả nhóm đã cùng xây dựng Project số 1 thông qua các quá trình phân tích thiết kế, cài đặt thực thi, ứng dụng xử lý từng File âm thanh và quá trình kiểm thử. Cả nhóm có mức độ hoàn thành bài tập nhóm là bằng nhau.

**Yêu cầu đề bài:**

**Phần 1: Xây dựng chương trình xử lý âm thanh bao gồm:**

**+ Xử lý file wav, (Không phải tự định nghĩa file)**

+ Các phép xử lý cơ bản: Cộng, nhân,

+ Đảo ngược thời gian

+ Dịch chuyển rời rạc

+ Nén - giảm tần số lấy mẫu

**- Yêu cầu thêm**

+ Nâng cao chất lượng - implement các bộ lọc

+ Loại bỏ nhiễu

+ Thêm một số hiệu ứng Ví dụ: tăng giảm âm lượng, lặp lại, ...

**Phần 2: file báo cáo**

Trang 1: Thành viên nhóm

+ Tên thành viên

+ Phân công sơ bộ

+ Đánh giá đóng góp tích cực

Trang 2: Trình bày vẽ các thiết kế (Phân tích thiết kế)

Trang 3: Các kết quả đạt được sau mini project 1

Trang 4: Danh sách test case:

+ Mỗi chức năng sẽ có 1-2 test case

Trang 5: Tài liệu tham khảo

**Phần 3: Phần mềm Demo**

**Phần 4: Nộp cho thầy**

+ File báo cáo

+ Source code --> zip

**PHÂN TÍCH THIẾT KẾ**

Phần tích thiết kế cho phần mềm xử lý tín hiệu âm thanh theo các bước sau:

**Bước 1: Xác định yêu cầu chức năng bao gồm như yêu bài toán đã đặt ra:**

* **Xử lý file wav, (Không phải tự định nghĩa file)**

+ Các phép xử lý cơ bản:

* Cộng, nhân,
* Đảo ngược thời gian
* +Dịch chuyển rời rạc
* Nén - giảm tần số lấy mẫu

+ Nâng cao chất lượng - implement các bộ lọc

* Bộ lọc Low Pass Filter (LPF)
* Bộ lọc High Pass Filter (HPF)
* Bộ lọc Band Pass Filter (BPF)
* Bộ lọc Band Stop Filter (BSF)

+ Loại bỏ nhiễu tín hiệu

* Sử dụng bọ lọc Wiener trong miền thời gian

+ Thêm một số hiệu ứng tăng chất lượng âm thanh

* Hiệu ứng lặp lại
* Hiệu ứng tạo tiếng vang
* Hiệu ứng tăng giảm âm thanh

**Bước 2: Thiết kế giao diện người dùng**

Xây dụng chương trình xử lý âm thanh theo menu bao gồm từng chức năng. Người dùng tương tác chương trình qua cửa sổ Console.

**Bước 3: Thiết kế cấu trúc chương trình:**

+ Xây dựng hàm Main bao gồm:

* Lựa chọn chức năng xử lý file
* Truyền dữ liệu đầu vào
* Gọi đến hàm xử lý chức năng
* Xuất file sau khi xử lý

+ Xây dựng các hàm chức năng xử lý file âm thanh

* Nhận vào các tham số là dữ liệu đầu vào nhập vào từ hàm main
* Xử lý file âm thanh theo chức năng
* Trả về file âm thanh sau khi đã xử lý

**Quá trình (kỹ thuật) cài đặt thực thi (implementation)**

**Quá trình xây dựng và cài đặt thực thi chương trình xử lý file âm thanh**

**1. Cài đặt thư viện AudioFile trên Visual Studio 2022**

* **Bước 1:** Tải file AudioFile.h trên <https://encr.pw/AudioFile> về máy tính
* **Bước 2:** Tạo Project mới 🡪 chọn chương trình Console App (C++) 🡪 Vào thư mục chứa Project vừa tạo và tạo folder mới “Include”
* **Bước 3:** Sao chép file AudioFile vào thư mục Include
* **Bước 4:** Chọn Property 🡪 chọn C/C++ 🡪 General 🡪 Additional include Directories🡪 Edit 🡪 New Line 🡪 truyền đường dẫn file Include vừa tạo 🡪 Apply
* **Bước 5:** Khai báo thư viện: #include <AudioFile.h>

**2. Xây dựng hàm Main**

* In ra Menu chức năng
* Tạo hàm While (true), nhập vào lựa chọn từ bàn phím
* Tạo hàm Switch – Case với các lựa chọn chức năng tương ứng theo Menu
* Đối với từng Case truyền vào các tham số tương ứng với các chức năng
* Gọi hàm chức năng xử lý âm thanh
* Xuất file âm thanh sau khi sử lý

**3. Xây dựng hàm cộng, nhân hai file âm thanh**

* Nhận vào đối số gồm 2 file âm thanh và toán tử (Nhân hoặc cộng).
* Tạo đối tượng AudioFile output
* Xét thông số tín hiệu đầu ra bao gồm số kênh, tần số mẫu, mẫu số liệu, số lượng mẫu số liệu
* Xử lý cộng/nhân hai file âm thanh:
  + Duyệt vòng for dãy giá trị của mẫu số liệu
  + Cộng/nhân hai dãy giá trị và gán vào dãy giá trị output
* Trả về đối tượng AudioFile

**4. Xây dựng hàm dịch chuyển thời gian**

* Nhận vào đối số gồm 1 file âm thanh và tham số m (khoảng dịch chuyển)
* Tạo đối tượng AudioFile output
* Xét thông số tín hiệu đầu ra bao gồm số kênh, tần số mẫu, mẫu số liệu, số lượng mẫu số liệu + m
* Xử lý dịch chuyển thời gian

Với trường hợp m > 0:

* Duyệt vòng for từ m đến số lượng mẫu số liệu đầu ra: Gán giá trị tín hiệu đầu ra = giá trị tín hiệu đầu vào trừ đi m.

Với trường hợp m <= 0:

* Duyệt vòng for từ 0 đến số lượng mẫu số liệu đầu ra: Gán giá trị tín hiệu đầu ra = giá trị tín hiệu đầu vào trừ đi m.
* Trả về đối tượng AudioFile sau khi đã xử lý

**5. Xây dựng hàm đảo ngược thời gian file âm thanh**

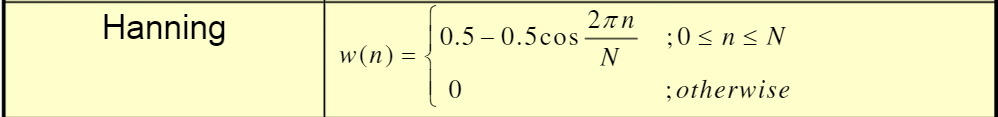
* Nhận vào đối số gồm 1 file âm thanh
* Tạo đối tượng AudioFile output
* Xét thông số tín hiệu đầu ra bao gồm số kênh, tần số mẫu, mẫu số liệu, số lượng mẫu số liệu
* Xử lý đảo ngược thời gian
  + Duyệt vòng for từ 0 đến số lượng mẫu số liệu đầu ra: Gán giá trị tín hiệu đầu ra = bằng giá trị đảo ngược tín hiệu đầu vào.
* Trả về đối tượng AudioFile sau khi đã xử lý.

**6. Xây dựng hàm nén, giảm tần số.**

* Nhận vào đối số gồm 1 file âm thanh và hệ số nén
* Tạo đối tượng AudioFile output
* Xét thông số tín hiệu đầu ra bao gồm số kênh, tần số mẫu, mẫu số liệu, số lượng mẫu số liệu
* Xử lý nén, giảm tần số
  + Duyệt vòng for từ 0 đến số lượng mẫu số liệu đầu ra: Gán giá trị tín hiệu đầu ra = bằng giá trị tín hiệu đầu vào nhân hệ số nén
* Trả về đối tượng AudioFile sau khi đã xử lý.

**7. Xây dựng hàm tính cửa sổ Hanning**

* Nhận vào đối số là bậc của bộ lọc
* Áp dụng công thức tính cửa sổ bộ lọc:



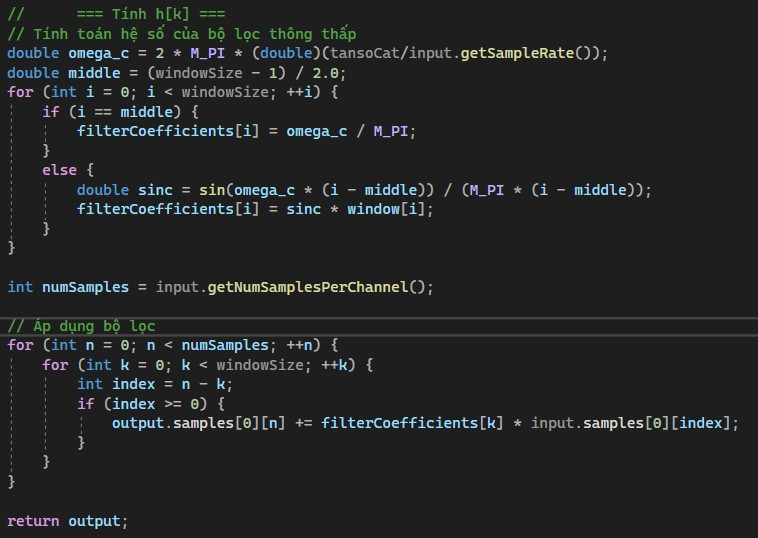
A computer screen shot of a program code

Description automatically generated.

**8.**

**Xây dựng hàm bộ lọc Low Pass Filter.**

* Nhận vào đối số gồm 1 file âm thanh, bậc của bộ lọc và tần số cắt
* Tạo đối tượng AudioFile output
* Xét thông số tín hiệu đầu ra bao gồm số kênh, tần số mẫu, mẫu số liệu, số lượng mẫu số liệu
* Gọi hàm tính cửa sổ Hanning (truyền vào tham số bậc của bộ lọc)
* Xử lý lọc Low Pass Filter



Trả về đối tượng AudioFile sau khi đã xử lý.

**9. Xây dựng hàm bộ lọc High Pass Filter**

* Nhận vào đối số gồm 1 file âm thanh, bậc của bộ lọc và tần số cắt
* Tạo đối tượng AudioFile output
* Xét thông số tín hiệu đầu ra bao gồm số kênh, tần số mẫu, mẫu số liệu, số lượng mẫu số liệu
* Gọi hàm tính cửa sổ Hanning (truyền vào tham số bậc của bộ lọc)
* Xử lý lọc High Pass Filter

A computer screen shot of a program

Description automatically generated

* Trả về đối tượng AudioFile sau khi đã xử lý.

**10. Xây dựng hàm bộ lọc Band Pass Filter**

* Nhận vào đối số gồm 1 file âm thanh, bậc của bộ lọc, tần số cắt 1 và tần số cắt 2
* Tạo đối tượng AudioFile output
* Xét thông số tín hiệu đầu ra bao gồm số kênh, tần số mẫu, mẫu số liệu, số lượng mẫu số liệu
* Gọi hàm tính cửa sổ Hanning (truyền vào tham số bậc của bộ lọc)
* A computer screen with many lines of code

  Description automatically generated with medium confidenceXử lý lọc Band Pass Filter

**10. Xây dựng hàm bộ lọc Band Stop Filter**

* Nhận vào đối số gồm 1 file âm thanh, bậc của bộ lọc, tần số cắt 1 và tần số cắt 2
* Tạo đối tượng AudioFile output
* Xét thông số tín hiệu đầu ra bao gồm số kênh, tần số mẫu, mẫu số liệu, số lượng mẫu số liệu
* Gọi hàm tính cửa sổ Hanning (truyền vào tham số bậc của bộ lọc)
* A computer screen shot of a program code

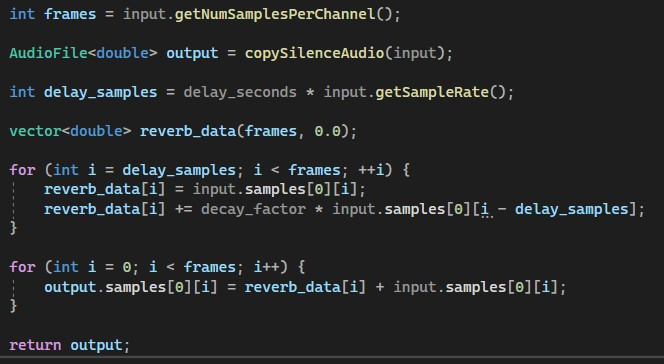
  Description automatically generatedXử lý lọc Band Stop Filter

**11. Xây dựng hàm WinnerFilter (Giảm nhiễu)**

* Nhận vào đối số gồm 1 file âm thanh gốc, 1 file âm thanh nhiễu, phương sai nhiễu
* Tạo đối tượng AudioFile output
* Xét thông số tín hiệu đầu ra bao gồm số kênh, tần số mẫu, mẫu số liệu, số lượng mẫu số liệu
* Xử lý bộ lọc Winner

**12. Xây dựng hàm tạo hiệu ứng thêm tiếng vang**

* Nhận vào đối số gồm 1 file âm thanh gốc, thời gian trễ (delay\_seconds), hệ số giảm âm thanh(decay\_factor)
* Tạo đối tượng AudioFile output
* Xét thông số tín hiệu đầu ra bao gồm số kênh, tần số mẫu, mẫu số liệu, số lượng mẫu số liệu
* Xử lý thêm hiệu ứng

****

**13. Xây dựng hàm tăng, giảm âm thanh**

* Nhận vào đối số gồm 1 file âm thanh gốc, thời gian trễ (delay\_seconds), hệ số giảm âm thanh(decay\_factor)
* Tạo đối tượng AudioFile output
* Xét thông số tín hiệu đầu ra bao gồm số kênh, tần số mẫu, mẫu số liệu, số lượng mẫu số liệu
* Xử lý tăng giảm âm thanh

A screen shot of a computer code

Description automatically generated

* Trả về đối tượng AudioFile sau khi xử lý

**14. Xây dựng hàm hiệu ứng lặp lại**

* Nhận vào đối số gồm 1 file âm thanh gốc, số lần lặp lại
* Tạo đối tượng AudioFile output
* Xét thông số tín hiệu đầu ra bao gồm số kênh, tần số mẫu, mẫu số liệu, số lượng mẫu số liệu
* Xử lý tăng giảm âm thanh

A computer screen with white and blue text

Description automatically generated

* Trả về đối tượng AudioFile sau khi xử lý

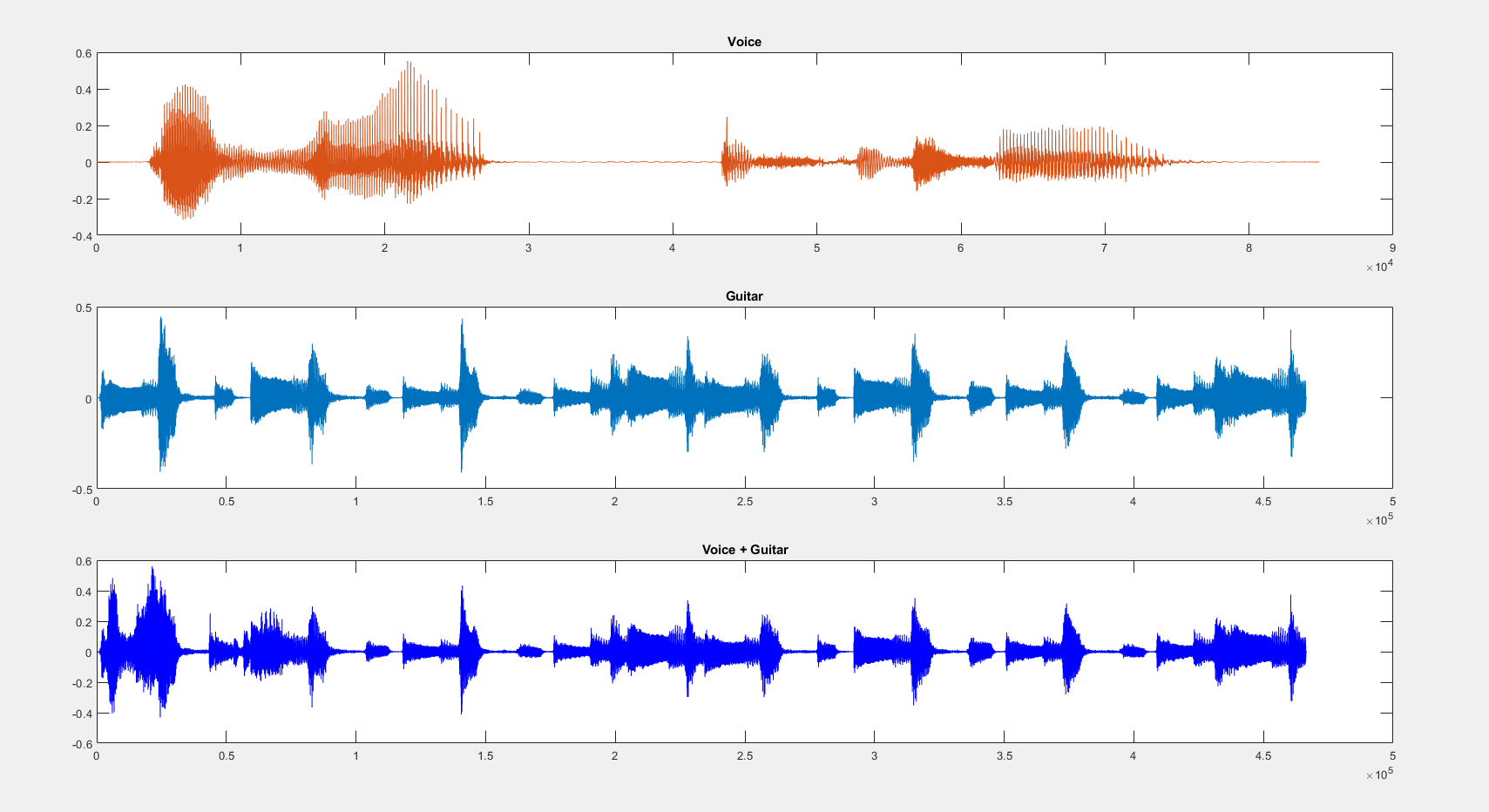
**Các kết quả chính của project đã đạt được**

**Các kết quả chính đạt được**

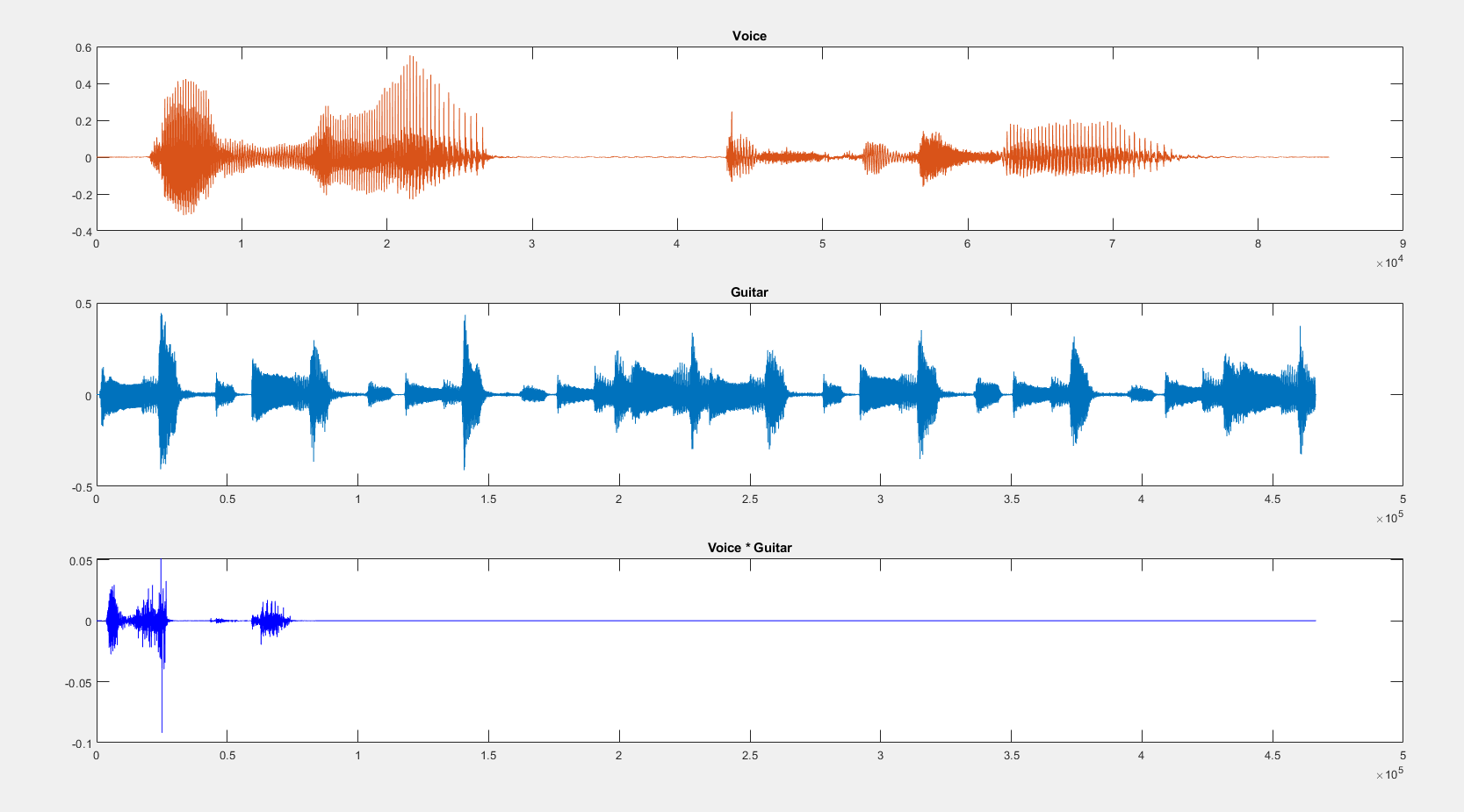
* Xây dựng thành công chương trình gồm các chức năng xử lý file âm thanh
* Hiểu thêm về các kĩ thuật xử lý file âm thanh.
* Nâng cao kỹ năng tìm kiếm thông tin, đọc hiểu tài liệu
* Nâng cao kỹ năng lập trình
* Nâng cao kỹ năng làm việc nhóm và giải quyết vấn đề.
* Có cơ hội phát triển dự án tiếp theo

**DANH SÁCH CÁC TEST CASE**

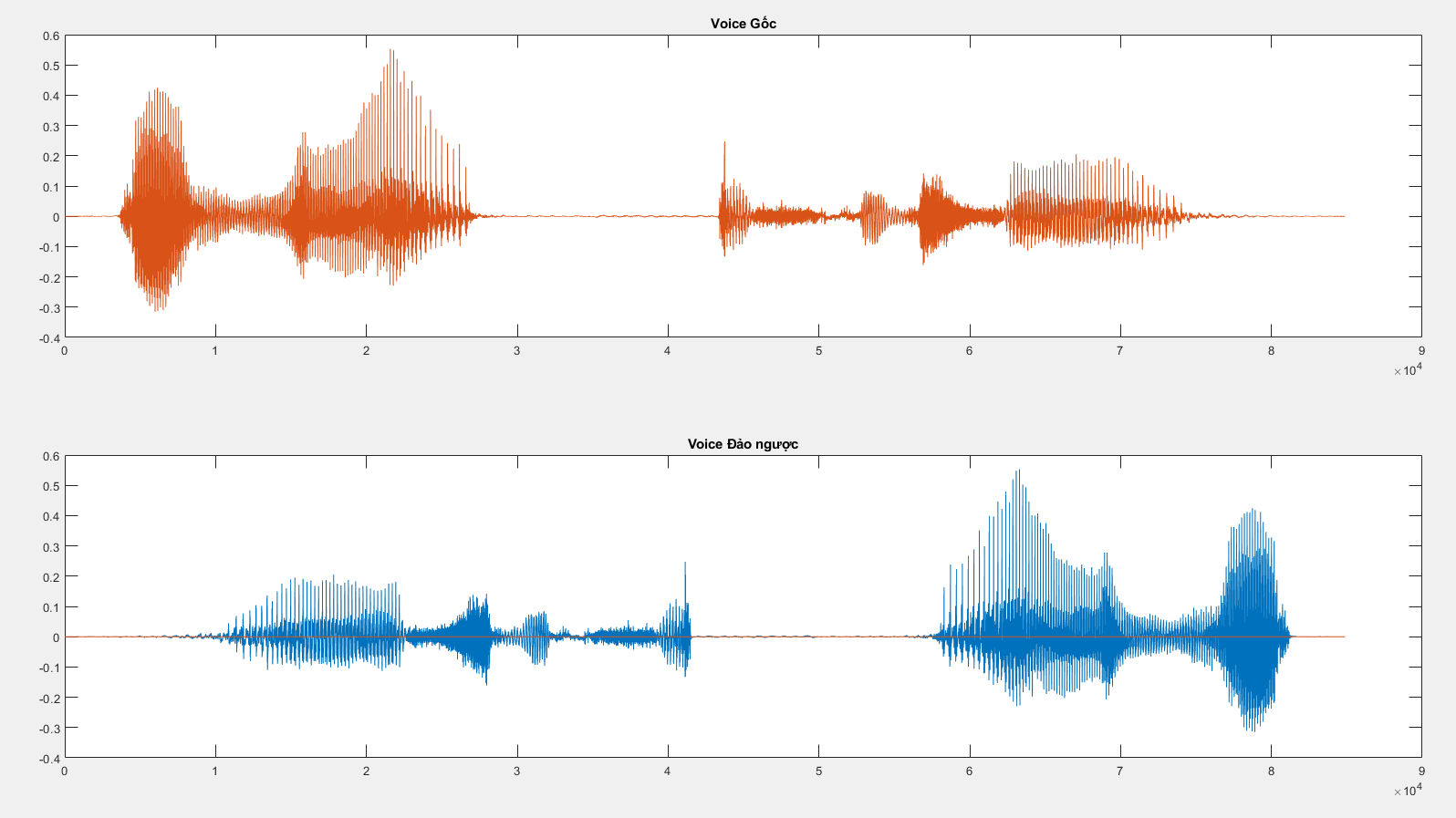
1 **Hàm cộng, nhân hai file âm thanh**



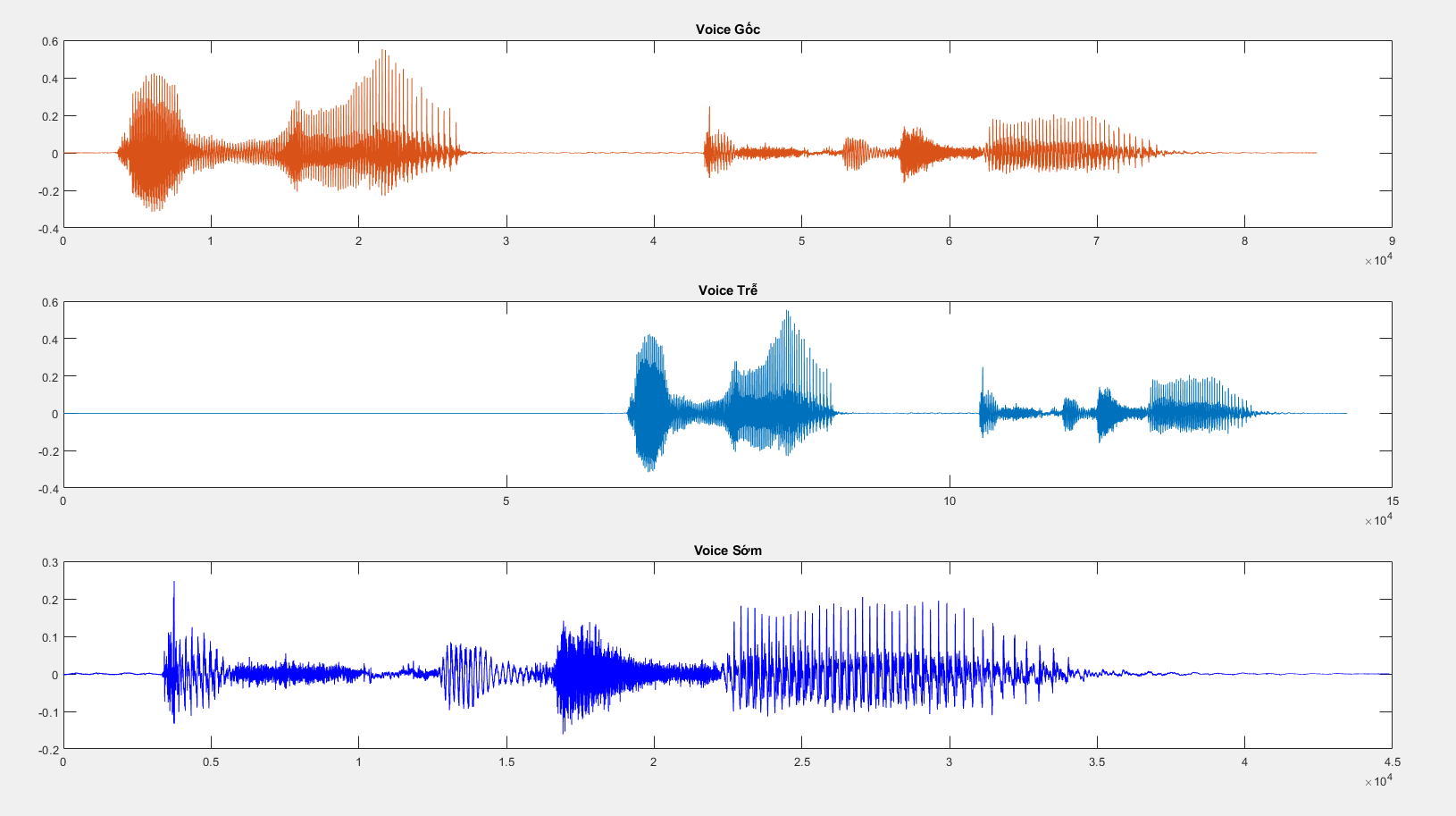
**2. Nhân hai tín hiệu**



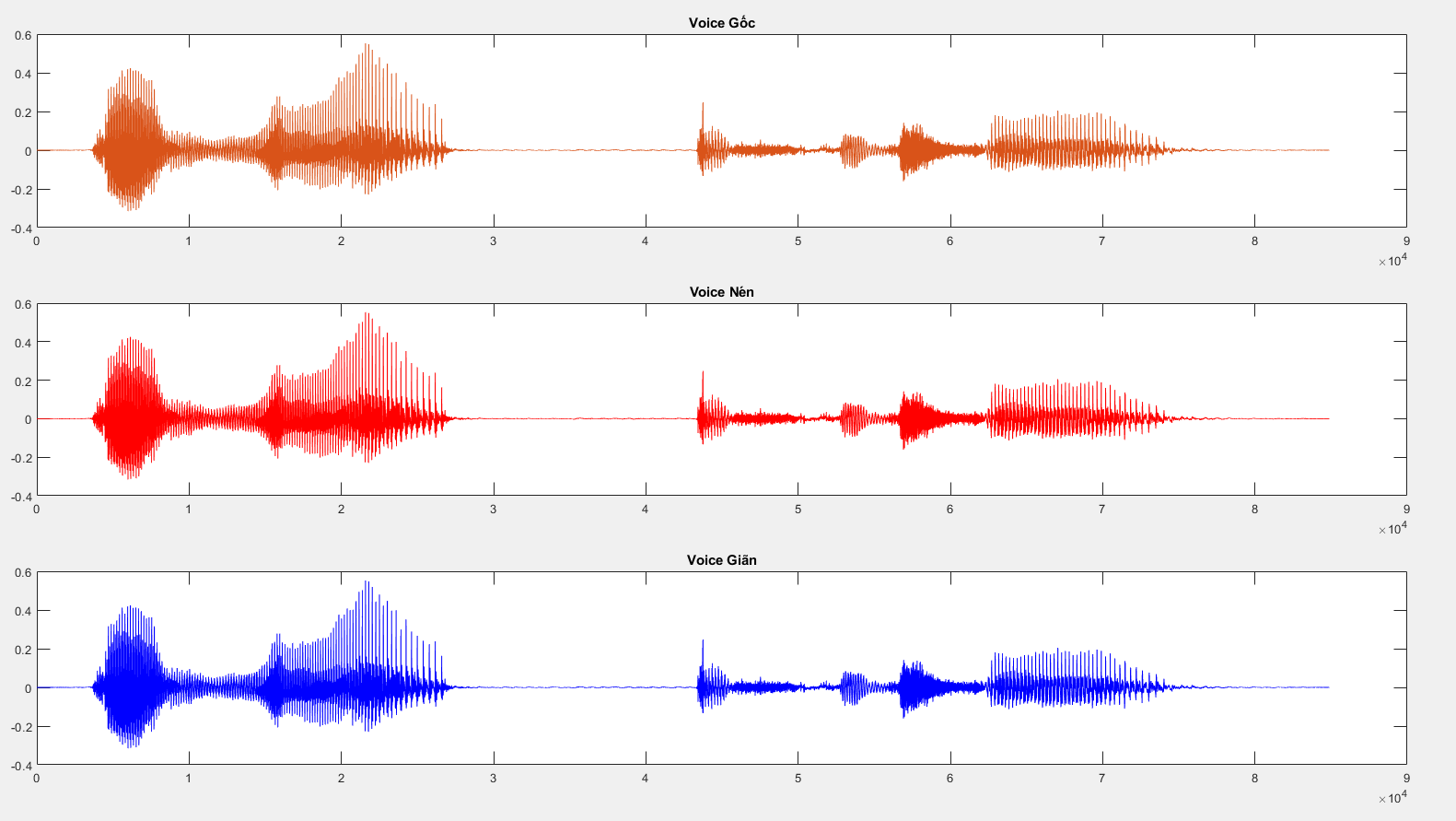
**3. Đảo ngược thời gian**



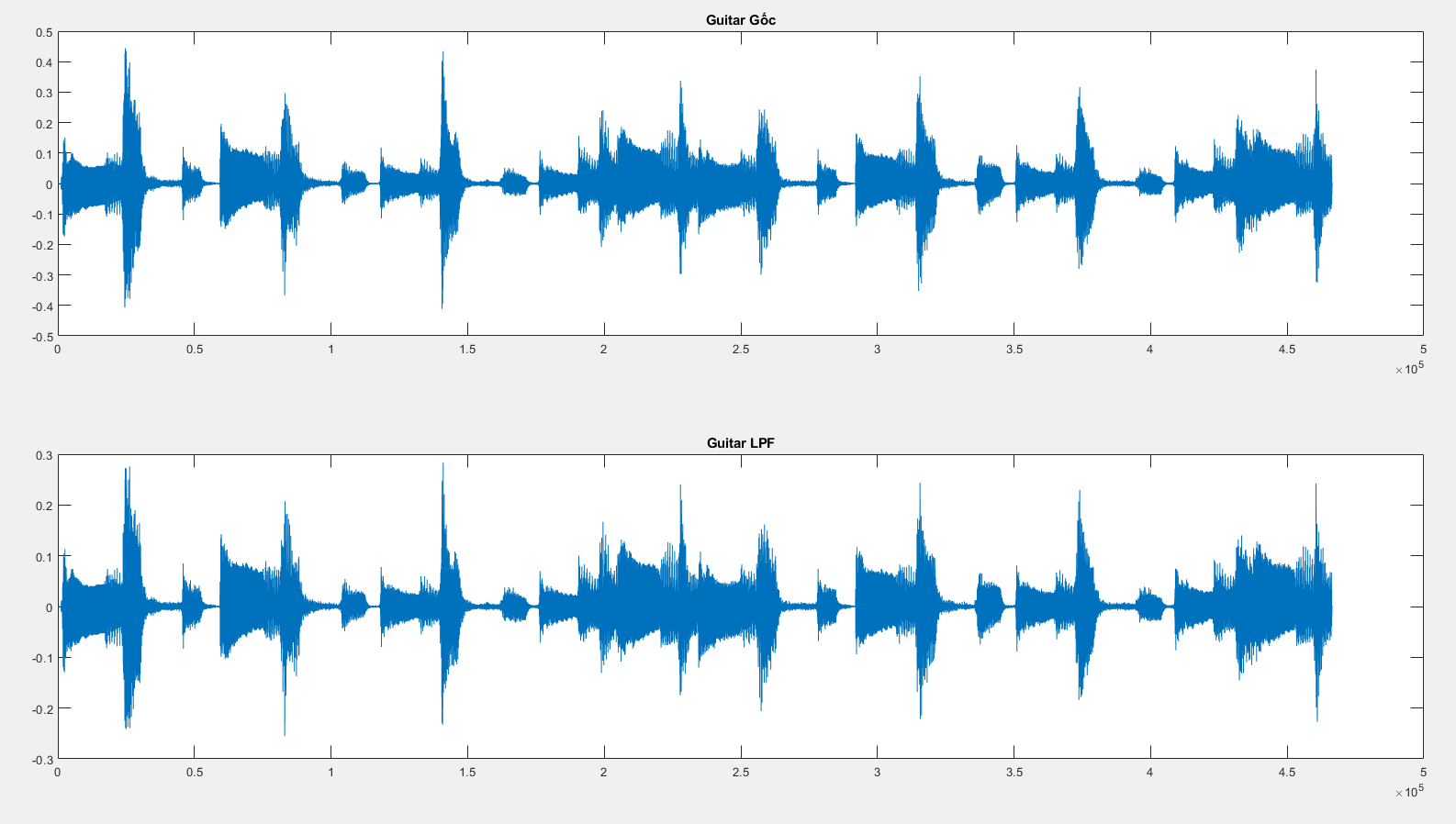
**4. Dịch chuyển rời rạc**



**5. Nén giảm tần số lấy mẫu**



**6. Bộ lọc Low Pass Filter**

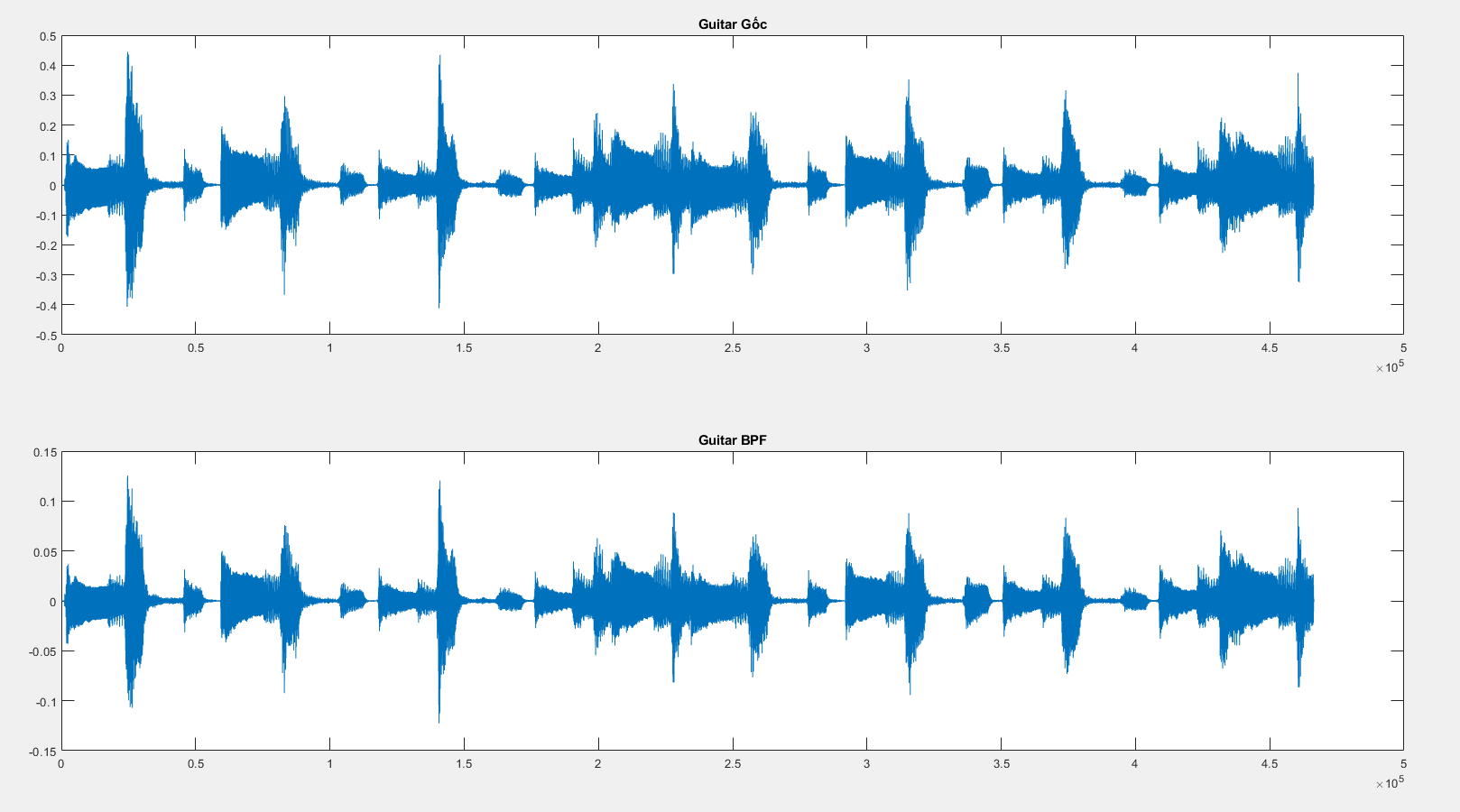


**7. Bộ lọc High Pass Filter**

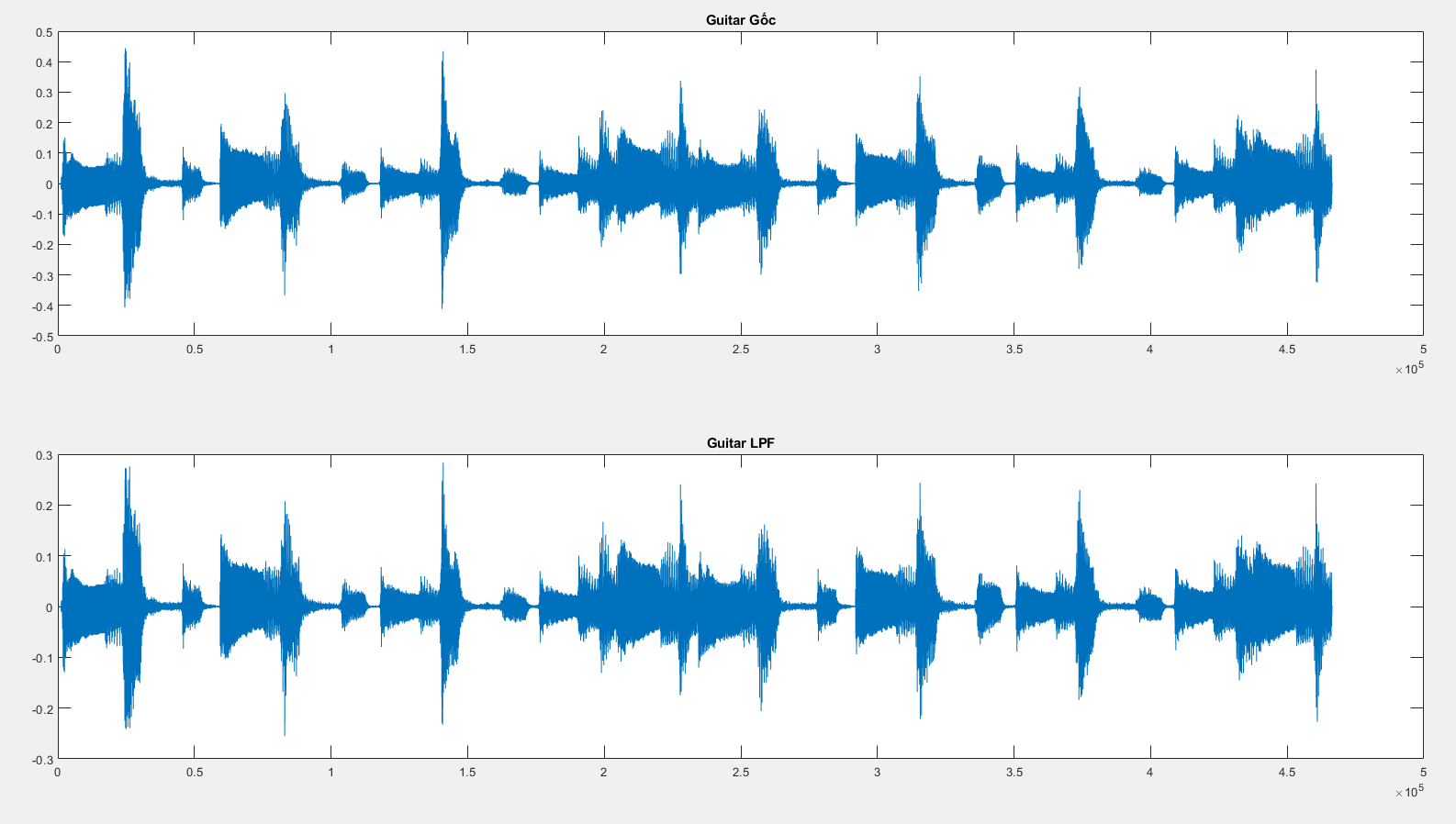
A screenshot of a computer screen

Description automatically generated

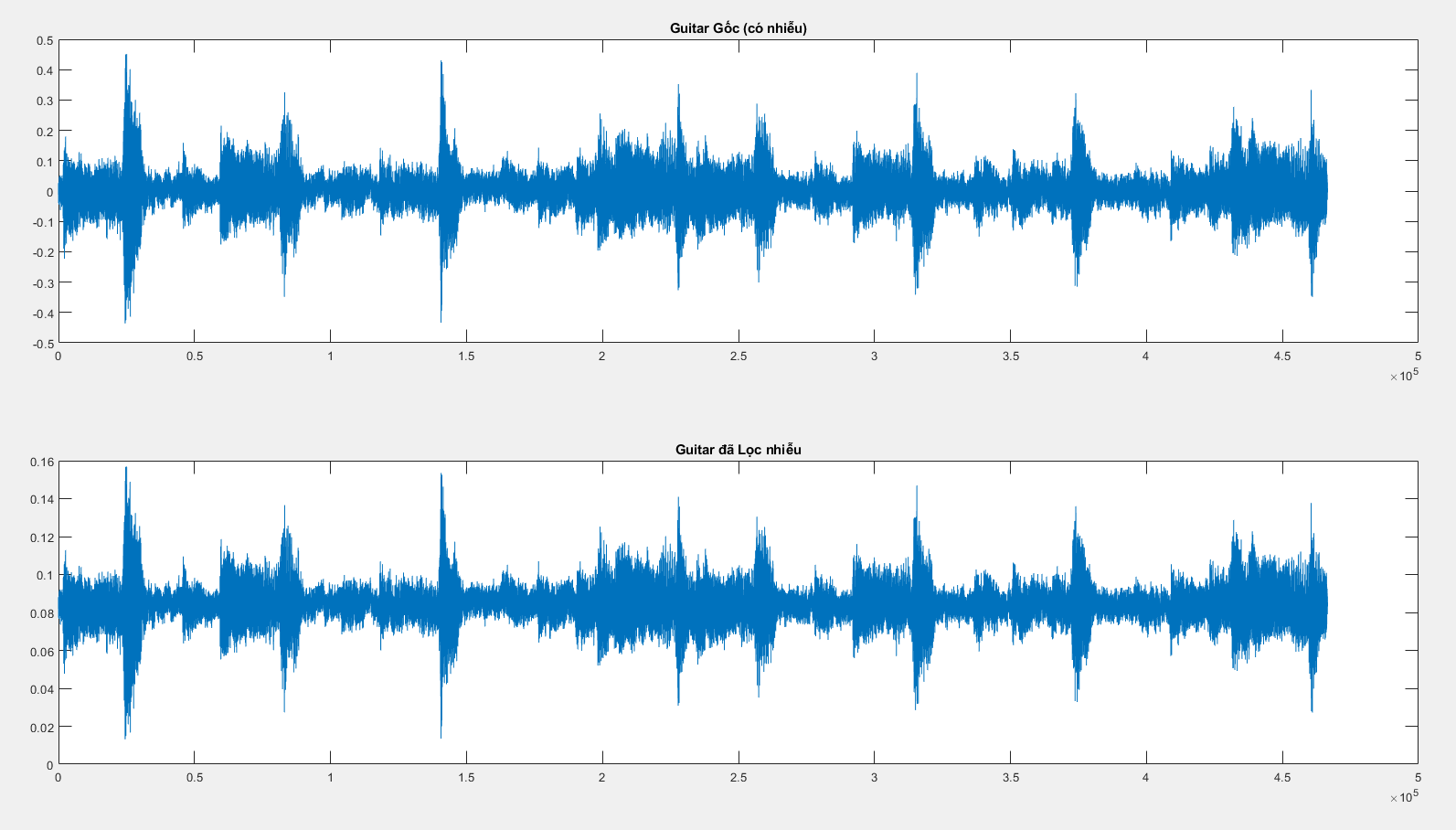
**8. Bộ lọc Band Pass Filter**



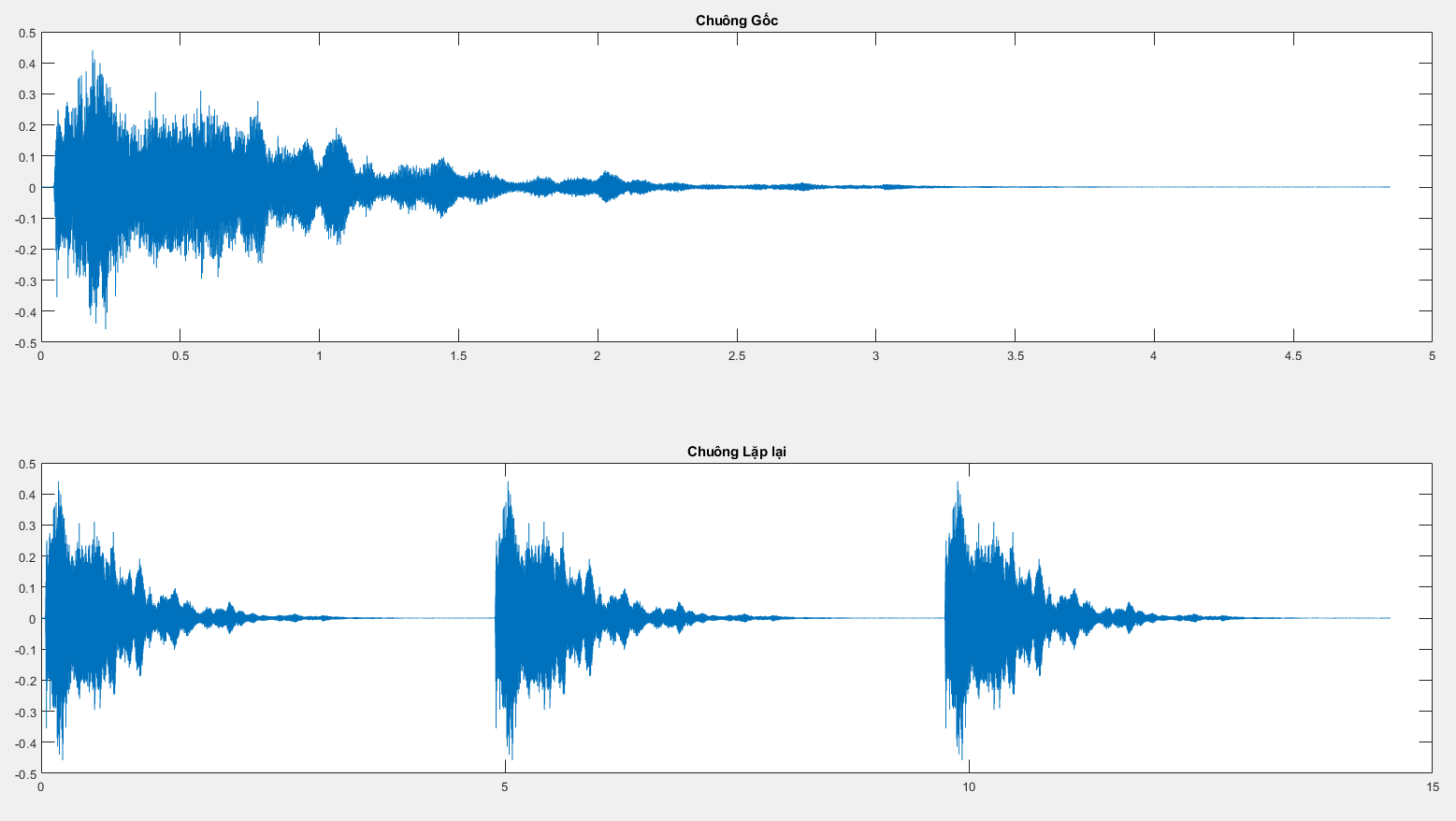
**9. Bộ lọc Band Stop Filter**



**10. Sử dụng bộ lọc Wiener trong miền thời gian**



**11. Hiệu ứng lặp lại**



**12. Hiệu ứng tạo tiếng vang**

A screenshot of a computer screen

Description automatically generated

13. Hiệu ứng tăng giảm âm thanh

A screenshot of a computer screen

Description automatically generated

**Tài liệu tham khảo**

<https://urlvn.net/5sjzj88>

<https://classroom.google.com/c/NjU0NjcyMTQzNTYy/m/NjU0Njc3MjI0NDU2/details>

<https://classroom.google.com/c/NjU0NjcyMTQzNTYy/m/NjU0Njc0ODk1OTcx/details>

<https://www.academia.edu/30258244/B%C3%A0i_gi%E1%BA%A3ng_X%E1%BB%AD_l%C3%BD_s%E1%BB%91_t%C3%ADn_hi%E1%BB%87u_Ch%C6%B0%C6%A1ng_8_THI%E1%BA%BET_K%E1%BA%BE_B%E1%BB%98_L%E1%BB%8CC_S%E1%BB%90>