# MỤC LỤC

**Chương 4. Cơ sở xử lý tín hiệu số**.....7
4.1 Hai cách nhìn một tín hiệu: miền thời gian và miền tần số.....8
4.2 Biến đổi Fourier (DFT/FFT).....10
4.3 Biến đổi Fourier thời gian ngắn và phổ đồ.....14
4.4 Bộ lọc số.....17
4.5 Phép toán tích chập.....20
4.6 Hướng dẫn thực hành.....23
4.7 Hệ thống bài tập và câu hỏi ôn tập.....26
4.8 Thuật ngữ chuyên ngành Anh – Việt.....29

# DANH MỤC HÌNH VẼ

Hình 4.1 Sóng sin 10 kHz hiển thị trên màn hình dao động ký tương tự (thang 100 µs/ô): cách nhìn tín hiệu theo miền thời gian....9
Hình 4.2 Hai cách nhìn cùng một tín hiệu: dạng sóng theo thời gian và phổ vạch hài Fourier theo tần số....10
Hình 4.3 Chân dung Joseph Fourier (1768–1830), tác giả của lý thuyết phân tích chuỗi lượng giác....11
Hình 4.4 Phổ vạch hài của một nốt nhạc (cơ bản khoảng 147 Hz), trục tần số lôgarit, cửa sổ Hann, FFT 4096 (Audacity)....13
Hình 4.5 Thiết lập phổ đồ trong Audacity: kích thước cửa sổ 2048, hàm cửa sổ Hann (Edit > Preferences > Spectrograms)....15
Hình 4.6 Spectral Frequency Display (trái) và Frequency Analysis (phải) của cùng một đoạn thoại mẫu tổng hợp trong Adobe Audition....15
Hình 4.7 Phổ đồ thời gian của đoạn tiếng nói tổng hợp trong Audacity (trục ngang: thời gian, trục dọc: tần số)....16
Hình 4.8 Phổ đồ đoạn nhạc violin: các nốt nhạc thể hiện qua tần số cơ bản và chuỗi hài âm (trục ngang: thời gian, trục dọc: tần số)....16
Hình 4.9 Phổ đồ thời gian (Spectrogram) và quy luật đánh đổi giữa độ phân giải thời gian và độ phân giải tần số....17
Hình 4.10 Bộ cân bằng tham số (Parametric Equalizer) của Adobe Audition khi cắt thông cao ở khoảng 80 Hz....18
Hình 4.11 Bộ cân bằng đồ thị (graphic equalizer) hai kênh, mỗi kênh 10 dải tần từ 30 Hz đến 16 kHz, kèm bộ hiển thị phổ....18
Hình 4.12 Đáp tuyến tần số của bốn loại bộ lọc số cơ bản: thông thấp, thông cao, thông dải và triệt dải....19
Hình 4.13 Phép tích chập: tín hiệu khô kết hợp với đáp ứng xung không gian tạo ra tín hiệu mang đặc tính vang dội tự nhiên....21
Hình 4.14 Đáp ứng xung phòng (tệp mẫu tổng hợp) và hộp thoại Convolution Reverb của Adobe Audition....21
Hình 4.15 Gian giữa nhà thờ lớn St. Vitus (Praha), một không gian có độ vang dài thường được dùng để thu đáp ứng xung....22

# DANH MỤC BẢNG BIỂU

Bảng 4.1 Ký hiệu và thuật ngữ viết tắt chuyên ngành.....7
Bảng 4.2 Đánh đổi độ phân giải thời gian và tần số ở tần số lấy mẫu chuẩn CD 44.100 Hz.....14
Bảng 4.3 Bảng thuật ngữ chuyên ngành Xử lý tín hiệu âm thanh số.....29

# DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

| Ký hiệu viết tắt | Thuật ngữ tiếng Anh đầy đủ | Ý nghĩa chuyên môn trong kỹ thuật Audio - Video |
|---|---|---|
| ADSR | Attack - Decay - Sustain - Release | Mô hình đường bao biên độ 4 giai đoạn theo thời gian trong thiết kế tiếng động |
| AES | Audio Engineering Society | Hội Kỹ sư Âm thanh Quốc tế — tổ chức ban hành các tiêu chuẩn âm thanh chuyên nghiệp |
| DAW | Digital Audio Workstation | Trạm làm việc âm thanh số (Pro Tools, DaVinci Resolve Fairlight, Nuendo, Reaper) |
| dB | Decibel | Đơn vị logarit biểu diễn tỷ số giữa hai đại lượng biên độ hoặc công suất |
| dB SPL | Decibel Sound Pressure Level | Mức áp suất âm thanh trong không khí quy chiếu theo ngưỡng nghe 20 µPa |
| dBFS | Decibels Relative to Full Scale | Thang đo decibel số toàn dải; 0 dBFS là trần tuyệt đối của hệ thống kỹ thuật số |
| dBTP | Decibels True Peak | Mức đỉnh thực của tín hiệu số có tính toán hiện tượng liên mẫu (inter-sample peaks) |
| dBA / dBC | A-weighted / C-weighted Decibels | Mức đo decibel qua mạch lọc bù trừ đặc tuyến độ nhạy của tai người |
| EBU | European Broadcasting Union | Hiệp hội Phát thanh Truyền hình Châu Âu (tác giả tiêu chuẩn EBU R128) |
| HRTF | Head-Related Transfer Function | Hàm truyền liên quan đến đầu người — cơ sở định vị không gian 3 chiều và âm thanh nhị nhĩ |
| ILD | Interaural Level Difference | Độ chênh lệch mức áp suất âm thanh giữa hai tai (chủ đạo ở tần số cao > 1,5 kHz) |
| ITD | Interaural Time Difference | Độ chênh lệch thời gian âm thanh truyền đến hai tai (chủ đạo ở tần số thấp < 1,5 kHz) |
| ITU-R | ITU Radiocommunication Sector | Phân ban Vô tuyến của Liên minh Viễn thông Quốc tế (tác giả chuẩn BS.1770) |
| LKFS / LUFS | Loudness K-weighted / Units Full Scale | Đơn vị đo lường độ to cảm nhận chuẩn hóa theo ITU-R BS.1770 |
| LRA | Loudness Range | Dải độ to — chỉ số mô tả độ động giữa đoạn to và đoạn nhỏ của chương trình (đơn vị LU) |
| NC / NR | Noise Criteria / Noise Rating | Đường cong tiêu chí đánh giá mức độ ồn nền của phòng thu |

| Ký hiệu viết tắt | Thuật ngữ tiếng Anh đầy đủ | Ý nghĩa chuyên môn trong kỹ thuật Audio - Video |
|---|---|---|
| NIOSH | National Institute for Occupational Safety and Health | Viện Quốc gia về An toàn và Sức khỏe Nghề nghiệp Hoa Kỳ |
| RMS | Root Mean Square | Giá trị căn bậc hai trung bình bình phương (năng lượng hiệu dụng của tín hiệu) |
| RT60 | Reverberation Time 60 dB | Thời gian để năng lượng âm vang suy giảm 60 dB sau khi ngắt nguồn âm |
| SMPTE | Society of Motion Picture and Television Engineers | Hiệp hội Kỹ sư Điện ảnh và Truyền hình Quốc tế |
| SNR | Signal-to-Noise Ratio | Tỷ số tín hiệu trên tạp âm (đo bằng dB) |

# CHƯƠNG 4
# CƠ SỞ XỬ LÝ TÍN HIỆU SỐ

Chương 4 giới thiệu công cụ toán học nằm sau mọi phần mềm xử lý âm thanh, gồm hai cách nhìn một tín hiệu trong miền thời gian và miền tần số, biến đổi Fourier và thuật toán FFT, phổ đồ thời gian STFT cùng quy luật đánh đổi độ phân giải, bốn loại bộ lọc số cơ bản và phép toán tích chập, nền tảng của công nghệ vang tích chập.

### *Mục tiêu của chương*

Sau khi đọc xong chương này, người đọc có thể:

1. **Về kiến thức lý thuyết:** Giải thích được mối quan hệ chặt chẽ giữa miền thời gian và miền tần số; phát biểu ý tưởng cốt lõi của phép biến đổi Fourier (DFT/FFT) và STFT; đọc đúng các trục tọa độ của biểu đồ phổ và phổ đồ thời gian (spectrogram); trình bày bản chất của quy luật đánh đổi độ phân giải thời gian - tần số; nêu nguyên lý hoạt động của bốn loại bộ lọc số và thuật toán tích chập (convolution).
2. **Về kỹ năng chuyên môn:** Sử dụng ngôn ngữ Python và thư viện khoa học để lập trình vẽ dạng sóng, tính toán phổ tần số (FFT) và trực quan hóa spectrogram; nhận diện 'vân tay phổ' đặc trưng của các nhạc cụ, tiếng thoại và tiếng động; áp dụng bộ lọc Butterworth để lọc tạp âm rít cao tần hoặc tiếng ù xoay chiều trầm.
3. **Về tư duy làm việc:** Hình thành tư duy 'hai cách nhìn một thực thể' trong thiết kế phần mềm đa phương tiện; chủ động lựa chọn công cụ phân tích và thông số kích thước khung tối ưu cho từng bài toán âm thanh cụ thể.

**Môi trường thực hành:** máy tính cài Python 3.8 trở lên (numpy, scipy, librosa, matplotlib) và phần mềm Audacity.

### *Ký hiệu và chữ viết tắt dùng trong chương*

*Bảng 4.1 Ký hiệu và thuật ngữ viết tắt chuyên ngành*

| Ký hiệu / Viết tắt | Thuật ngữ tiếng Anh nguyên gốc | Ý nghĩa chuyên môn trong xử lý tín hiệu số |
|---|---|---|
| DFT | Discrete Fourier Transform | Phép biến đổi Fourier rời rạc dùng cho tín hiệu số rời rạc |
| FFT | Fast Fourier Transform | Thuật toán biến đổi Fourier nhanh giúp tính DFT tốc độ cao |

| Ký hiệu / Viết tắt | Thuật ngữ tiếng Anh nguyên gốc | Ý nghĩa chuyên môn trong xử lý tín hiệu số |
|---|---|---|
| STFT | Short-Time Fourier Transform | Biến đổi Fourier thời gian ngắn trượt cửa sổ theo trục thời gian |
| FIR | Finite Impulse Response | Bộ lọc số có đáp ứng xung hữu hạn (bộ lọc không hồi tiếp) |
| IIR | Infinite Impulse Response | Bộ lọc số có đáp ứng xung vô hạn (bộ lọc có hồi tiếp) |
| dBFS | Decibels Relative to Full Scale | Đơn vị đo decibel toàn dải quy chuẩn trong môi trường số |
| DSP | Digital Signal Processing | Lĩnh vực Khoa học kỹ thuật Xử lý tín hiệu số |
| WAV / PCM | Waveform Audio File Format / PCM | Định dạng âm thanh số không nén lưu trữ chuỗi mẫu điều chế xung mã |
| ADC / DAC | Analog-to-Digital / Digital-to-Analog | Chuỗi bộ chuyển đổi tương tự - số và số - tương tự phần cứng |
| Hz / kHz | Hertz / Kilohertz | Đơn vị đo tần số dao động vật lý (chu kỳ trên giây) |

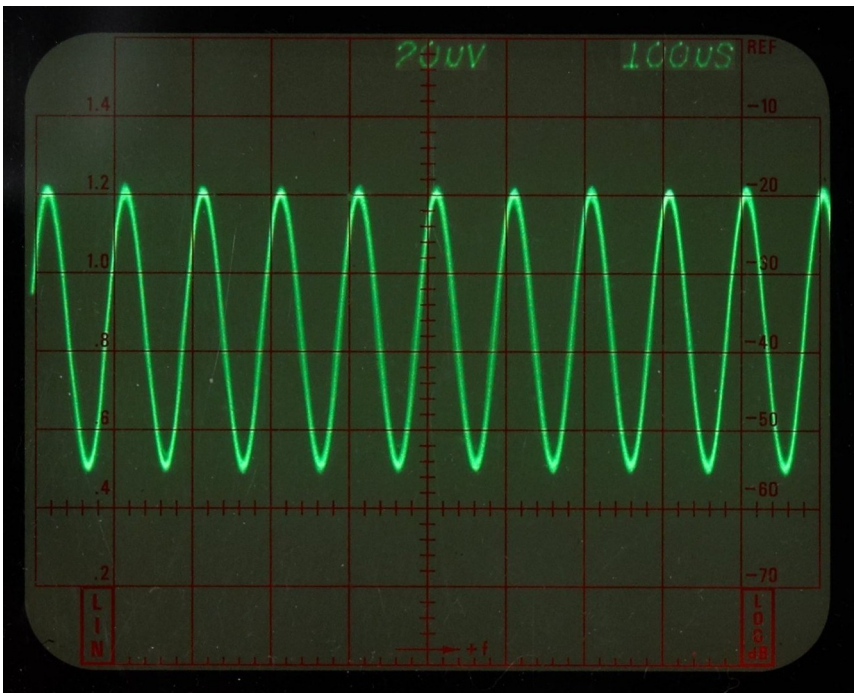
## 4.1 Hai cách nhìn một tín hiệu: miền thời gian và miền tần số

Ở Chương 3, âm thanh vật lý đã được số hóa và chuyển đổi thành một mảng số thực một chiều (numpy.ndarray) [10] trong bộ nhớ máy tính. Từ chương này, người đọc bắt đầu nghiên cứu phương pháp quan sát, giải nghĩa và can thiệp trực tiếp vào mảng dữ liệu số đó. Điểm mấu chốt mang tính nguyên lý là cùng một tín hiệu âm thanh có thể được mô tả bằng hai cách nhìn bổ sung lẫn nhau, tương tự như việc chụp một bức tượng nghệ thuật từ hai góc nhìn khác nhau để có hình dung toàn diện nhất [4].

### 4.1.1 Khái niệm miền thời gian và miền tần số

Miền thời gian (time domain) là phương pháp quan sát truyền thống và trực quan nhất: biểu diễn sự biến thiên của biên độ áp suất sóng âm (hoặc dải điện thế tín hiệu) theo trục thời gian tuyến tính. Đó chính là dạng sóng (waveform) hiển thị khi mở một tệp WAV trong phần mềm Audacity hoặc các DAW chuyên dụng, và miền thời gian trả lời trực tiếp câu hỏi: 'Tại mỗi thời điểm chính xác, tín hiệu âm thanh mạnh hay yếu ra sao?'.

Hình 4.1 là màn hình dao động ký tương tự hiển thị sóng sin 10 kHz với thang thời gian 100 µs/ô, một ví dụ điển hình của cách quan sát tín hiệu trong miền thời gian.




*Hình 4.1 Sóng sin 10 kHz hiển thị trên màn hình dao động ký tương tự (thang 100 µs/ô): cách nhìn tín hiệu theo miền thời gian*

*Nguồn: Pittigrilli, CC BY-SA 4.0, Wikimedia Commons (truy cập 05/11/2026).*

Ngược lại, miền tần số (frequency domain) là một phương pháp quan sát hoàn toàn khác: cùng một đoạn tín hiệu trong miền thời gian được phân rã thành các thành phần tần số cơ bản cấu thành nên nó, chỉ rõ mỗi tần số thành phần đóng góp bao nhiêu phần trăm năng lượng vào tổng thể tín hiệu. Miền tần số trả lời trực tiếp câu hỏi: 'Âm thanh này được cấu tạo từ những tần số nào, và mức năng lượng của mỗi tần số là bao nhiêu?'. Biểu đồ biểu diễn sự phân bổ năng lượng theo tần số được gọi là phổ âm thanh (spectrum).

**VÍ DỤ 4.1 — ẨN DỤ: LĂNG KÍNH TÁCH ÁNH SÁNG TRẮNG**

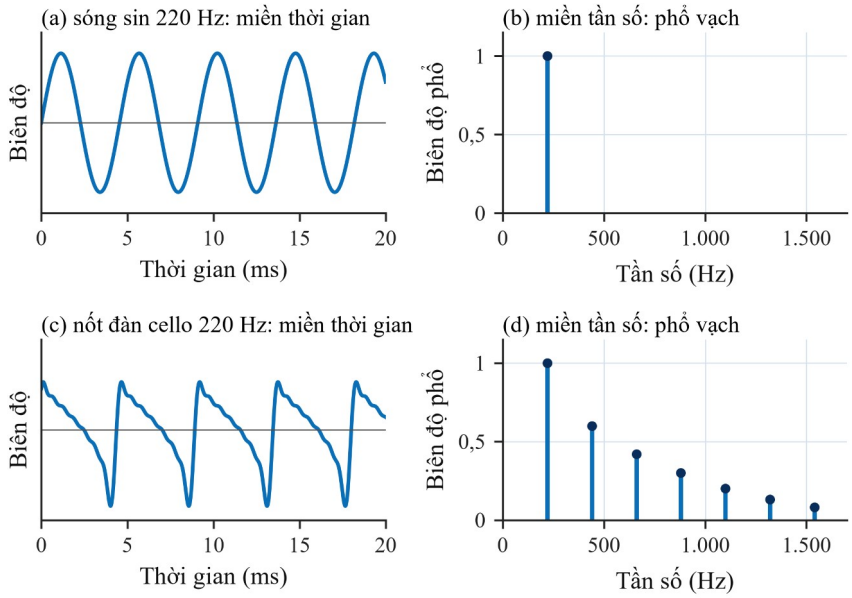
Một tia sáng trắng tự nhiên chiếu qua một khối lăng kính thủy tinh bị khúc xạ và tách thành một dải cầu vồng gồm bảy sắc màu đơn sắc (đỏ, cam, vàng, lục, lam, chàm, tím), trong đó mỗi màu tương ứng với một 'tần số' bước sóng ánh sáng cụ thể.

Phép biến đổi tín hiệu sang miền tần số hoạt động tương tự một lăng kính toán học: nó phân rã một tín hiệu âm thanh phức tạp thành các tần số sóng sin đơn âm cấu thành. Chẳng hạn, tai người nghe một hợp âm piano như một khối âm thanh hài hòa thống nhất, nhưng ở miền tần số có thể thấy rõ hợp âm đó gồm những nốt nhạc đơn lẻ nào (nốt Đồ, nốt Mi, nốt Sol), xuất hiện dưới dạng các vạch năng lượng nhọn.

### 4.1.2 Vân tay phổ của nhạc cụ và tiếng động

Nội dung Chương 1 đã làm rõ lý do một cây sáo trúc và một cây kèn trumpet cùng chơi một nốt La 440 Hz (A4) ở cùng mức âm lượng nhưng tai người vẫn phân biệt được hai nhạc cụ đó: sự khác biệt nằm ở cấu trúc chuỗi phổ hài Fourier và tỷ lệ biên độ năng lượng giữa các hài âm. Miền tần số chính là nơi 'vân tay phổ' đặc trưng này được hiển thị rõ ràng và khoa học nhất.

Một sóng sin đơn âm tinh khiết, khi quan sát trên biểu đồ phổ miền tần số, chỉ xuất hiện một vạch đứng nhọn tại tần số tương ứng. Một nhạc cụ thật phát ra âm có cao độ rõ ràng hiển thị một vạch cao nhất tại tần số cơ bản ($f_0$) cùng một chuỗi các vạch hài âm cách đều nhau ($2f_0$, $3f_0$, $4f_0$...). Ngược lại, một tiếng động tạp nhiễu phức tạp không có cao độ rõ rệt (như tiếng xì của tivi hỏng, tiếng mưa rơi, tiếng gõ trống) có biểu đồ phổ trải rộng liên tục trên nhiều dải tần và không xuất hiện các đỉnh nhọn cách đều. Nhờ quan sát biểu đồ phổ, kỹ sư âm thanh có thể nhận dạng loại nhạc cụ hay nguồn âm mà không cần nghe thử bằng tai nghe.




*Hình 4.2 Hai cách nhìn cùng một tín hiệu: dạng sóng theo thời gian và phổ vạch hài Fourier theo tần số*

## 4.2 Biến đổi Fourier (DFT/FFT)

Nền tảng toán học của việc chuyển đổi tín hiệu từ miền thời gian sang miền tần số là phép biến đổi Fourier. Phép toán này xuất phát từ ý tưởng của nhà toán học Jean-Baptiste Joseph Fourier: mọi tín hiệu trong tự nhiên, dù phức tạp đến đâu, đều có thể được phân tích thành tổng của nhiều sóng sin đơn tần, trong đó mỗi sóng sin thành phần có tần số, biên độ và góc pha hoàn toàn xác định [1], [2].

### 4.2.1 Ý tưởng cốt lõi của thuật toán FFT

Biến đổi Fourier là công cụ giúp tìm ra 'công thức pha trộn' ẩn bên trong tín hiệu, cho biết chính xác cần những sóng sin nào, với tỷ lệ biên độ bao nhiêu, để khi cộng tất cả lại thì thu được đúng tín hiệu gốc. Đối với tín hiệu âm thanh số (là một chuỗi các mẫu biên độ rời rạc nT), phiên bản toán học tương ứng là phép biến đổi Fourier rời rạc (DFT - Discrete Fourier Transform).

Hình 4.3 là chân dung Joseph Fourier (1768–1830), nhà toán học người Pháp, người đặt nền móng cho việc biểu diễn một hàm tuần hoàn bằng tổng các thành phần hình sin mà thuật toán FFT kế thừa và tính toán hiệu quả.




*Hình 4.3 Chân dung Joseph Fourier (1768–1830), tác giả của lý thuyết phân tích chuỗi lượng giác*

*Nguồn: Amédée Félix Barthélemy Geille (theo Julien-Léopold Boilly), Public domain, Wikimedia Commons (truy cập 05/11/2026).*

Tuy nhiên, việc tính DFT theo công thức định nghĩa trực tiếp đòi hỏi độ phức tạp thuật toán rất lớn là $O(N^2)$, nên máy tính không thể thực thi theo thời gian thực đối với các mảng âm thanh dài. Thuật toán biến đổi Fourier nhanh (FFT - Fast Fourier Transform) [5] giảm độ phức tạp xuống còn O(N log N), nhờ đó tăng tốc độ tính toán lên hàng nghìn lần. Trong kỹ thuật lập trình âm thanh thực tế, FFT là công cụ tiêu chuẩn được tích hợp trong mọi thư viện xử lý tín hiệu: đầu vào là một đoạn mảng số biên độ trong miền thời gian, đầu ra là mảng kết quả biểu diễn phổ biên độ trong miền tần số.

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi kn/N} \tag{4.1}$$

### 4.2.2 Phương pháp đọc hiểu biểu đồ phổ tần số

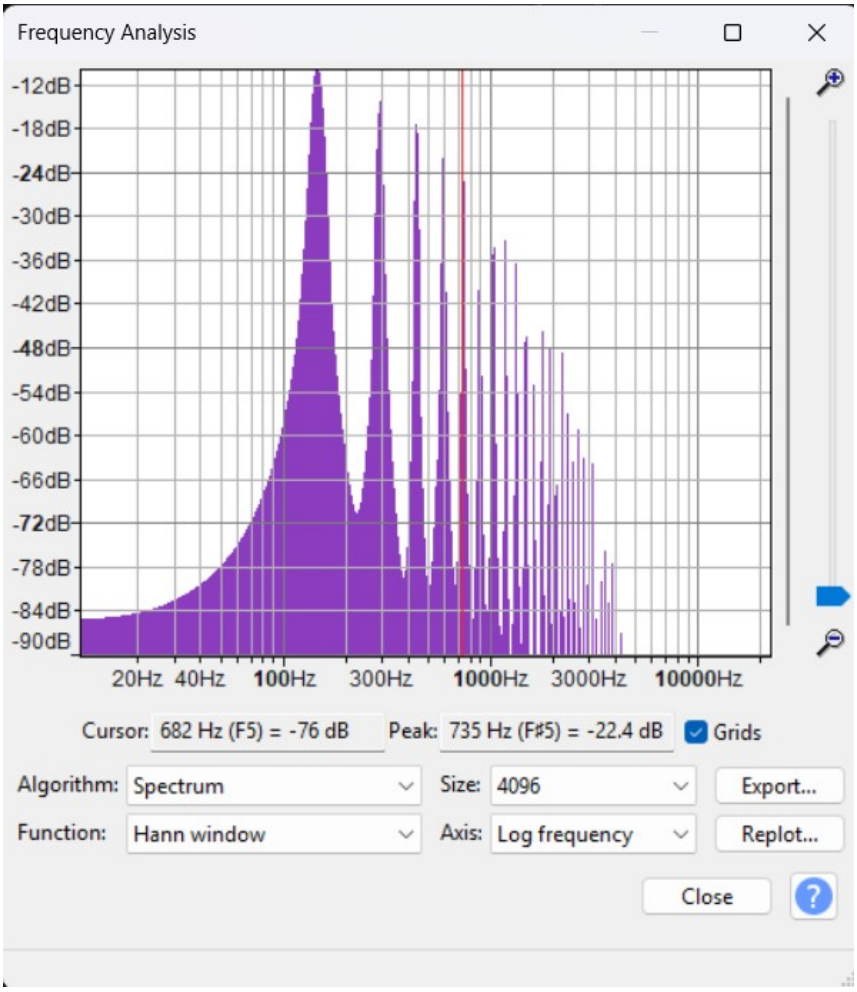
Một biểu đồ phổ tần số tiêu chuẩn được hiển thị với hai trục tọa độ cụ thể:

- **Trục hoành (trục X):** Biểu diễn tần số dao động vật lý tính bằng đơn vị Hertz (Hz) hoặc Kilohertz (kHz); thang đo thường theo dạng logarit để tương thích với cảm nhận cao độ của thính giác người.
- **Trục tung (trục Y):** Biểu diễn biên độ phổ hoặc mức năng lượng của tần số tương ứng, thường đo bằng thang Decibel (dB) hoặc dBFS.

Khi phân tích phổ, kỹ sư cần ghi nhớ ba dạng đồ thị đặc trưng sau:

1. **Xuất hiện một đỉnh đứng nhọn duy nhất:** Đặc trưng cho một thành phần sóng sin thuần, tức một đơn âm tinh khiết ở tần số tương ứng.
2. **Xuất hiện nhiều đỉnh nhọn cách đều nhau:** Đặc trưng cho một âm thanh có cao độ rõ ràng kèm chuỗi hài âm Fourier (như giọng nói con người, tiếng nhạc cụ violin, flute); đỉnh nhọn đầu tiên ở tần số thấp nhất là tần số cơ bản $f_0$ quyết định cao độ nốt nhạc, các đỉnh sau là hài bậc 2, 3, 4 cách đều theo bội số nguyên.
3. **Đồ thị phổ trải rộng liên tục, gồ ghề và không có đỉnh rõ rệt:** Đặc trưng cho các tiếng động phi điều hòa (tiếng va đập, tiếng nổ) hoặc tiếng ù nền, tiếng xì trắng (white noise) có năng lượng phân bổ đều ở mọi tần số.

Hình 4.4 là phổ của một nốt nhạc tổng hợp có tần số cơ bản khoảng 147 Hz, đo bằng Plot Spectrum của Audacity với cửa sổ Hann và FFT 4096: các vạch hài nằm cách đều nhau ở các bội số của tần số cơ bản và giảm dần về biên độ.




*Hình 4.4 Phổ vạch hài của một nốt nhạc (cơ bản khoảng 147 Hz), trục tần số lôgarit, cửa sổ Hann, FFT 4096 (Audacity)*

*Nguồn: ảnh chụp màn hình do người soạn thực hiện (Audacity 3.7.7), 07/11/2026.*

### 4.2.3 Quy luật đánh đổi độ phân giải thời gian và tần số

Trong xử lý tín hiệu số tồn tại một quy luật bất biến mang tính vật lý: không thể đạt đồng thời độ phân giải thời gian và độ phân giải tần số tối đa. Để phân biệt rõ hai tần số nằm rất sát nhau ở miền tần số, thuật toán FFT phải phân tích trên một đoạn tín hiệu mẫu có độ dài đủ lớn trong miền thời gian. Đoạn tín hiệu đưa vào phân tích càng dài (kích thước khung N lớn) thì độ phân giải tần số càng mịn và chính xác, nhưng độ phân giải thời gian bị 'nhòe' đi, khiến không thể biết chính xác nốt nhạc hay tiếng động đó vang lên ở mili-giây nào.

Ngược lại, nếu chọn một khung tín hiệu rất ngắn để định vị chính xác thời điểm xuất hiện tiếng động, thì dải tần số thu được sẽ thô và mờ, nên không thể phân biệt được các nốt nhạc gần nhau.

*Bảng 4.2 Đánh đổi độ phân giải thời gian và tần số ở tần số lấy mẫu chuẩn CD 44.100 Hz*

| Kích thước khung N (mẫu) | Độ dài thời gian của khung (ms) | Độ phân giải tần số Δf (Hz) | Định vị thời gian / Ứng dụng |
|---|---|---|---|
| 256 mẫu | 256 / 44.100 ≈ 5,8 ms | 44.100 / 256 ≈ 172,3 Hz | Rất tốt. Phù hợp phân tích tiếng động gõ, bộ gõ (Transient) |
| 1.024 mẫu | 1.024 / 44.100 ≈ 23,2 ms | 44.100 / 1.024 ≈ 43,1 Hz | Cân bằng. Sử dụng phổ biến cho hầu hết các tín hiệu nhạc và lời thoại |
| 4.096 mẫu | 4.096 / 44.100 ≈ 92,9 ms | 44.100 / 4.096 ≈ 10,8 Hz | Kém. Phù hợp phân tích cao độ (Pitch), hòa âm nốt trầm phức tạp |

$$\Delta f = \frac{f_s}{N} \quad \text{(Hz)} \tag{4.2}$$

## 4.3 Biến đổi Fourier thời gian ngắn và phổ đồ

Một bản nhạc hay một đoạn ghi âm luôn biến đổi: âm thanh, lời thoại, tiếng động thay đổi liên tục, nốt nhạc này nối tiếp nốt nhạc kia dọc theo trục thời gian. Nếu áp dụng thuật toán FFT cho toàn bộ tệp âm thanh dài, kết quả chỉ là một biểu đồ phổ trung bình của cả bài, cho biết bài có những tần số nào nhưng không xác định được tần số đó vang lên tại thời điểm nào. Để giải quyết bài toán biến thiên này, cần đến phép biến đổi Fourier thời gian ngắn (STFT - Short-Time Fourier Transform).

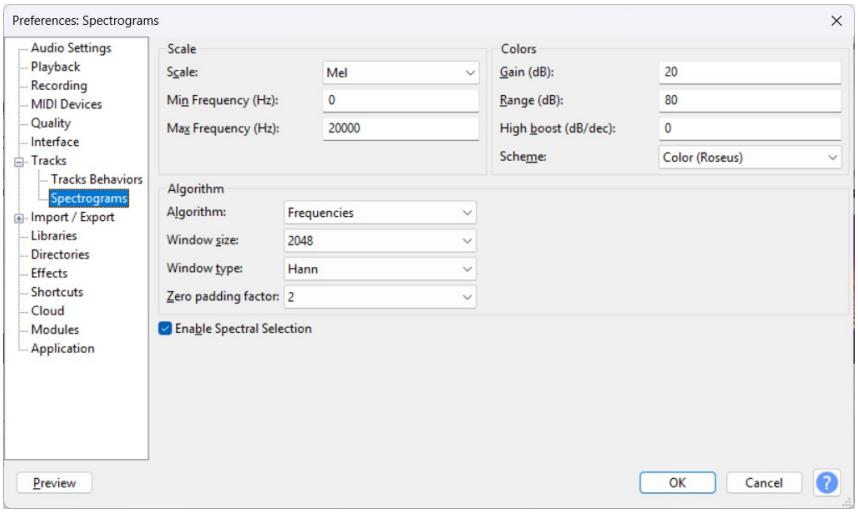
### 4.3.1 Nguyên lý hoạt động của STFT và cửa sổ trượt

Thay vì phân tích toàn bộ tín hiệu một lần, thuật toán STFT sử dụng một 'cửa sổ trượt' (sliding window) có kích thước ngắn xác định (vài chục mili-giây) trượt dọc theo tín hiệu miền thời gian. Tại mỗi vị trí dừng, STFT trích xuất đoạn tín hiệu nằm trong cửa sổ và thực hiện phép biến đổi Fourier (FFT) trên đoạn ngắn đó. Lặp lại quy trình này từ đầu đến cuối tệp, STFT cho ra một chuỗi các biểu đồ phổ biến thiên liên tục theo dòng thời gian [7].

### 4.3.2 Chồng lấn khung và hàm cửa sổ làm mềm

Để quá trình phân tích diễn ra liên tục và không mất thông tin tại ranh giới cắt giữa các khung, các khung phân tích kề nhau được thiết kế gối lên nhau một khoảng nhất định gọi là độ chồng lấn (overlap, thường chọn bằng 50% hoặc 75% kích thước khung).

Hình 4.5 là trang Spectrograms trong Preferences của Audacity, nơi chọn thuật toán, kích thước cửa sổ (Window size 2048), loại cửa sổ (Hann), hệ số đệm không và dải tần hiển thị của phổ đồ.




*Hình 4.5 Thiết lập phổ đồ trong Audacity: kích thước cửa sổ 2048, hàm cửa sổ Hann (Edit > Preferences > Spectrograms)*

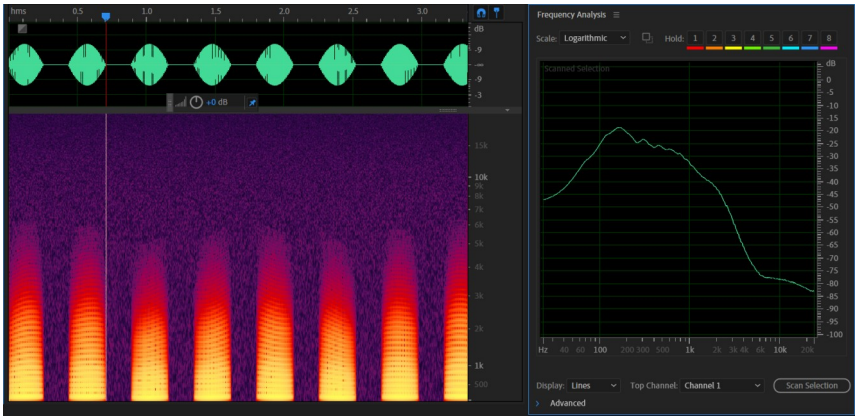
*Nguồn: ảnh chụp màn hình do người soạn thực hiện (Audacity 3.7.7), 07/11/2026.*

Bên cạnh đó, việc cắt cụt đột ngột một đoạn tín hiệu tại hai đầu khung gây ra hiện tượng méo toán học gọi là rò rỉ phổ (spectral leakage), làm xuất hiện nhiều tần số giả không có thật ở biên phổ. Để khắc phục, đoạn tín hiệu trong mỗi khung phải được nhân với một hàm cửa sổ làm mềm (Window Function, phổ biến là cửa sổ Hann, Hamming hoặc Blackman) [6]; hàm này có đặc tính triệt tiêu dần biên độ về 0 tại hai đầu khung, giúp làm mịn biên tín hiệu trước khi chạy FFT.

### 4.3.3 Phương pháp đọc hiểu biểu đồ phổ đồ thời gian

Kết quả đầu ra của phép biến đổi STFT là một ma trận dữ liệu hai chiều biểu diễn phổ theo thời gian, được trực quan hóa thành biểu đồ phổ đồ thời gian (Spectrogram) quen thuộc trong các phần mềm âm thanh chuyên nghiệp. Một spectrogram biểu diễn đồng thời ba trục thông tin cốt lõi:

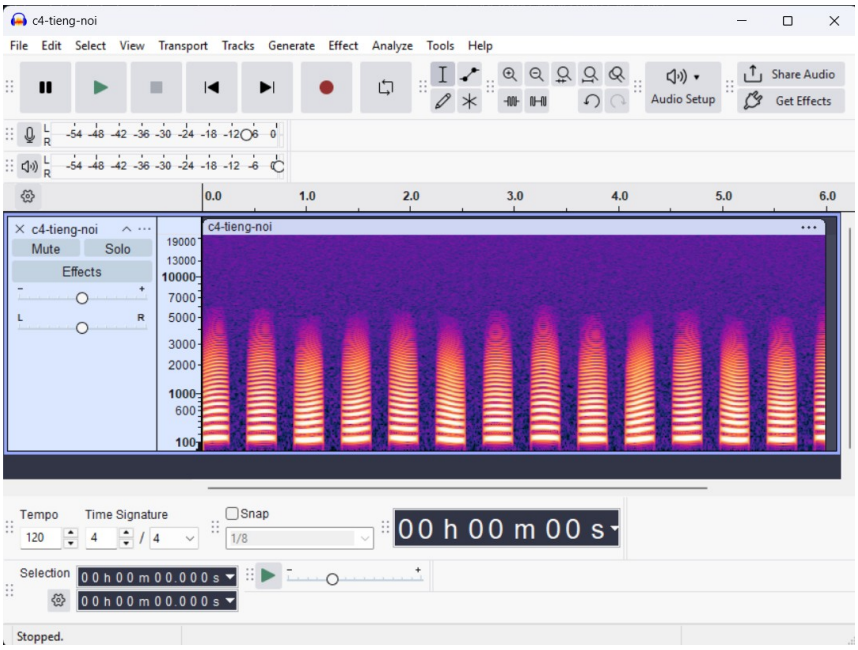
Hình 4.6 đặt cạnh nhau hai cách xem cùng một đoạn âm thanh trong Adobe Audition: phổ đồ thời gian ở bên trái cho thấy năng lượng phân bố theo cả thời gian lẫn tần số, còn cửa sổ Frequency Analysis ở bên phải cho thấy phổ trung bình của toàn đoạn trên thang tần số logarit.




*Hình 4.6 Spectral Frequency Display (trái) và Frequency Analysis (phải) của cùng một đoạn thoại mẫu tổng hợp trong Adobe Audition*

*Nguồn: ảnh chụp màn hình do người soạn thực hiện (Adobe Audition 2026, bản 26.5), 07/11/2026.*

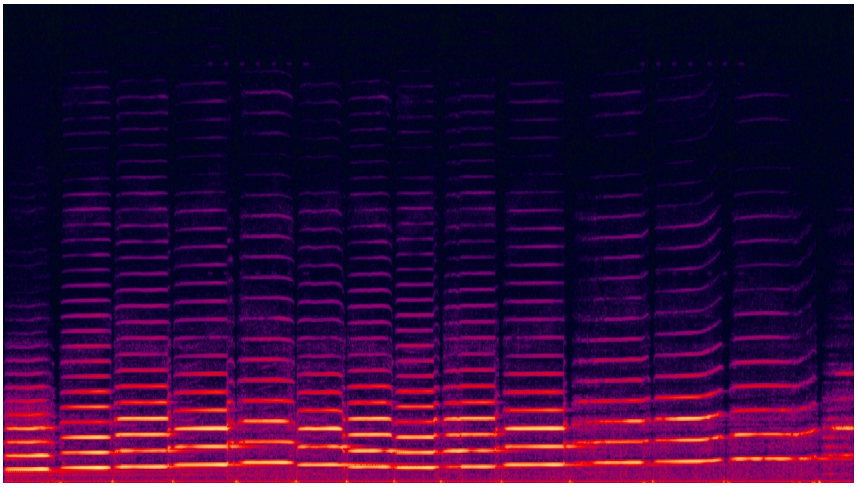
Hình 4.7 là phổ đồ của một đoạn tiếng nói tổng hợp trong Audacity: mỗi âm tiết hiện thành một cột sáng, các vạch ngang cách đều là các hài của tần số cơ bản, còn vùng năng lượng tập trung phản ánh các vùng cộng hưởng.




*Hình 4.7 Phổ đồ thời gian của đoạn tiếng nói tổng hợp trong Audacity (trục ngang: thời gian, trục dọc: tần số)*

*Nguồn: ảnh chụp màn hình do người soạn thực hiện (Audacity 3.7.7), 07/11/2026.*

Hình 4.8 là phổ đồ của một đoạn nhạc đàn violin: mỗi nốt hiện ra thành một cột gồm các vạch ngang cách đều nhau, tương ứng với tần số cơ bản và các hài âm, trong khi trục ngang thể hiện sự thay đổi theo thời gian.



*Hình 4.8 Phổ đồ đoạn nhạc violin: các nốt nhạc thể hiện qua tần số cơ bản và chuỗi hài âm (trục ngang: thời gian, trục dọc: tần số)*

*Nguồn: Antilived (người tải lên; tác giả gốc không ghi rõ), CC BY-SA 3.0, Wikimedia Commons (truy cập 05/11/2026).*

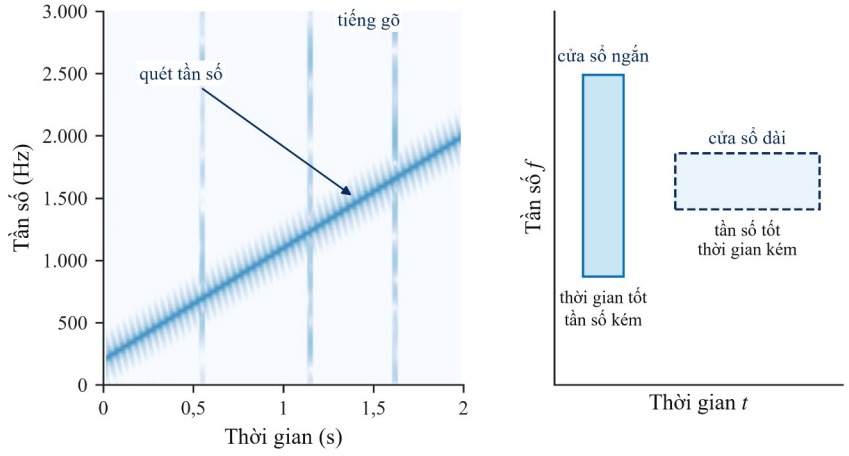
- **Trục hoành (trục ngang):** Biểu diễn trục thời gian thực tế chạy từ đầu đến cuối đoạn âm thanh (giây).
- **Trục dọc (trục đứng):** Biểu diễn dải tần số âm thanh từ 0 Hz đến tần số giới hạn Nyquist ($f_s/2$).

– **Trục màu sắc (thang màu sắc):** Biểu diễn mức năng lượng hoặc biên độ của âm thanh tại mốc thời gian và tần số tương ứng (thường quy đổi sang thang đo dB); màu tối (đen, xanh lam sẫm) biểu diễn năng lượng rất yếu hoặc không có âm thanh, còn màu nóng, sáng (vàng, đỏ, trắng) biểu diễn năng lượng lớn.

Nhờ cấu trúc hiển thị 3 chiều trực quan này, mọi hành động âm thanh đều để lại một dấu vết hình học đặc trưng trên spectrogram:

1. **Một nốt nhạc trầm kéo dài:** hiện thành một vạch sáng nằm ngang, phẳng, ở phân vùng tần số thấp phía dưới đồ thị.
2. **Một tiếng động gõ, tiếng tát, tiếng nổ cực ngắn:** hiện thành một vệt sáng thẳng đứng dứt khoát kéo dài từ tần số thấp lên tần số cao, do năng lượng được giải phóng tức thời ở mọi dải tần.
3. **Tiếng còi hú hoặc tiếng vuốt tần số (Chirp) từ trầm lên cao:** hiện thành một đường sáng uốn lượn đi chéo từ góc dưới bên trái lên góc trên bên phải.

**Nối lại Chương 3:** spectrogram là công cụ phát hiện hiện tượng 'Fake Lossless', tức bản nhạc được ghi nhãn chất lượng cao FLAC nhưng thực chất được chuyển đổi từ MP3 128 kbps. Trên spectrogram, tệp fake lossless lộ rõ một vạch cắt phẳng, dốc đứng như bức tường gạch (brick-wall cutoff) tại mốc 16.000 Hz, và dải cao phía trên hoàn toàn không còn năng lượng vì thuật toán nén MP3 đã loại bỏ phần này từ trước.




(a) phổ đồ: chirp và ba tiếng gõ (b) ô phân giải thời gian – tần số

*Hình 4.9 Phổ đồ thời gian (Spectrogram) và quy luật đánh đổi giữa độ phân giải thời gian và độ phân giải tần số*

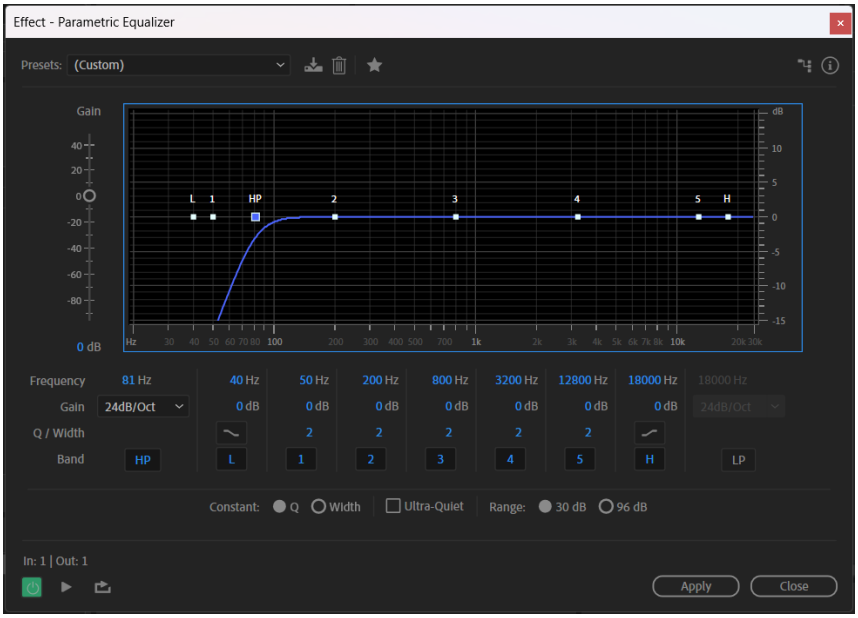
## 4.4 Bộ lọc số

Khi đã nắm được cách nhìn tín hiệu theo hai miền và có thể quan sát âm thanh dưới dạng phổ tần số, việc can thiệp để tinh chỉnh âm sắc trở nên rõ ràng và chủ động hơn. Bộ lọc số (digital filter) là một hệ thống toán học được lập trình nhằm giữ lại một số dải tần số mong muốn và làm suy hao các dải tần số không mong muốn.

### 4.4.1 Bốn loại bộ lọc số cơ bản và ứng dụng

Trong xử lý tín hiệu và hòa âm hậu kỳ, bốn loại bộ lọc nền tảng sau được sử dụng thường xuyên:

Hình 4.10 là hộp thoại Parametric Equalizer của Adobe Audition với băng HP (thông cao) bật ở khoảng 80 Hz, độ dốc 24 dB/oct; đường cong đáp ứng cho thấy phần dưới tần số cắt bị suy giảm mạnh còn dải trên 100 Hz giữ nguyên.




*Hình 4.10 Bộ cân bằng tham số (Parametric Equalizer) của Adobe Audition khi cắt thông cao ở khoảng 80 Hz*

*Nguồn: ảnh chụp màn hình do người soạn thực hiện (Adobe Audition 2026, bản 26.5), 07/11/2026.*

Hình 4.11 cho thấy một bộ cân bằng đồ thị (graphic equalizer) với các thanh trượt cố định theo từng dải tần từ 30 Hz đến 16 kHz, là ứng dụng trực quan của việc tăng hoặc cắt từng dải tần bằng bộ lọc.

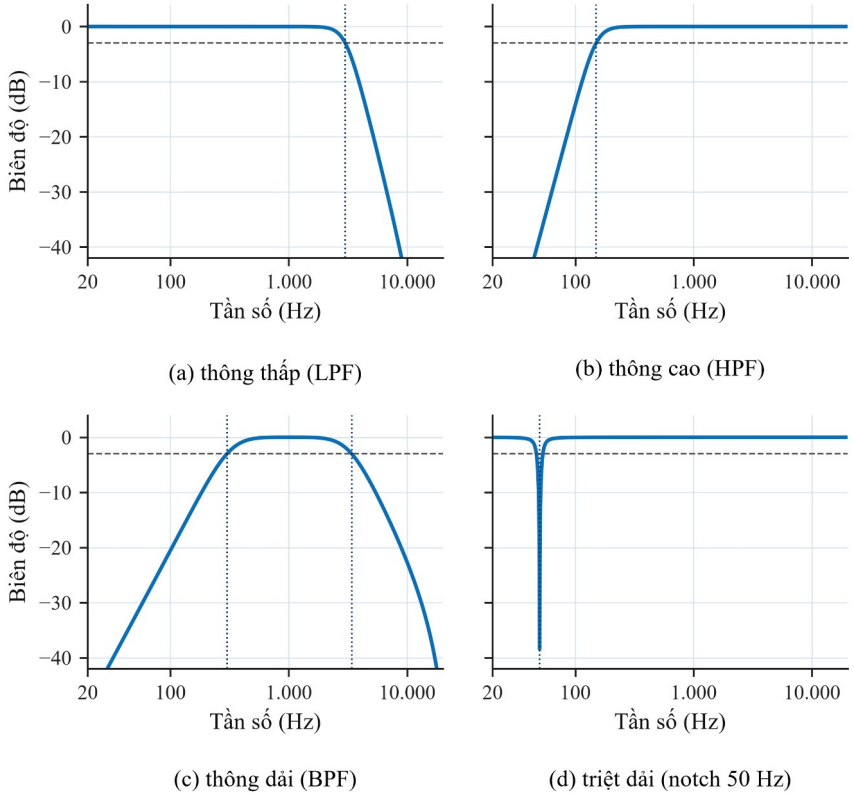



*Hình 4.11 Bộ cân bằng đồ thị (graphic equalizer) hai kênh, mỗi kênh 10 dải tần từ 30 Hz đến 16 kHz, kèm bộ hiển thị phổ*

*Nguồn: Touho_T, CC BY 2.0, Wikimedia Commons (truy cập 05/11/2026).*

1. **Bộ lọc thông thấp (Low-pass Filter - LPF):** chỉ cho phép các tần số thấp hơn tần số cắt ($f_c$) đi qua và làm suy hao mạnh các tần số cao hơn $f_c$. Ứng dụng: lọc bỏ tiếng xì cao tần của gió máy lạnh và làm dịu âm thanh chói; đây cũng là bộ lọc chống răng cưa (Anti-Aliasing Filter) bắt buộc phải đặt trước bộ chuyển đổi ADC.

2. **Bộ lọc thông cao (High-pass Filter - HPF):** chỉ cho phép các tần số cao hơn tần số cắt ($f_c$) đi qua và triệt tiêu các tần số trầm dưới $f_c$. Ứng dụng: cắt bỏ tiếng ù nền dải trầm (rumble), tiếng rền của động cơ xe và tiếng gió khi thu âm ngoài trời.
3. **Bộ lọc thông dải (Band-pass Filter - BPF):** chỉ cho phép một dải tần số nằm giữa hai tần số cắt biên ($f_L$ và $f_H$) đi qua và triệt tiêu toàn bộ tần số cực trầm và cực cao. Ứng dụng: cô lập giọng nói con người và mô phỏng hiệu ứng 'giọng nói truyền qua điện thoại' (chỉ giữ dải thông thoại chuẩn từ 300 Hz đến 3.400 Hz).
4. **Bộ lọc chắn dải / bộ lọc cắt rãnh (Band-stop / Notch Filter):** triệt tiêu hoàn toàn một dải tần số rất hẹp quanh tần số trung tâm và cho phép mọi tần số khác đi qua bình thường. Ứng dụng: cắt bỏ chính xác tiếng ù xoay chiều ở 50 Hz hoặc 60 Hz do lỗi tiếp đất lưới điện mà không ảnh hưởng đến dải tần lời thoại lân cận.




*Hình 4.12 Đáp tuyến tần số của bốn loại bộ lọc số cơ bản: thông thấp, thông cao, thông dải và triệt dải*

### 4.4.2 Phương trình sai phân và hai họ bộ lọc FIR / IIR

Về mặt toán học, một bộ lọc số hoạt động trực tiếp trong miền thời gian bằng cách tính giá trị của từng mẫu đầu ra y[n] tại thời điểm n thông qua một tổ hợp tuyến tính có trọng số của các mẫu đầu vào hiện tại x[n], các mẫu đầu vào quá khứ x[n-k] và có thể cả các mẫu đầu ra vừa tính trước đó y[n-m]. Công thức tổng quát này được gọi là phương trình sai phân [1], [3]:

$$y(n) = \sum_{k=0}^{M} b_k\, x(n-k) - \sum_{m=1}^{L} a_m\, y(n-m) \qquad (4.3)$$

Trong đó, các hệ số $\{a_m\}$ và $\{b_k\}$ là các trọng số toán học; việc lựa chọn các bộ trọng số khác nhau quyết định đáp tuyến tần số và loại bộ lọc thu được. Từ phương trình sai phân này, các bộ lọc số được chia thành hai họ cốt lõi:

- **Bộ lọc FIR (Finite Impulse Response - Bộ lọc có đáp ứng xung hữu hạn):** là họ bộ lọc không có nhánh hồi tiếp (tất cả các hệ số $\{a_m\}$ bằng không), nên mỗi mẫu đầu ra y[n] chỉ phụ thuộc vào mẫu đầu vào hiện tại và quá khứ. Đặc tính: luôn ổn định tuyệt đối và dễ thiết kế tuyến tính pha, nhưng đòi hỏi số bậc lọc (số phép nhân cộng) rất lớn để đạt độ dốc cắt sắc nét, do đó gây trễ tín hiệu.
- **Bộ lọc IIR (Infinite Impulse Response - Bộ lọc có đáp ứng xung vô hạn):** là họ bộ lọc có nhánh hồi tiếp (feedback loop, các hệ số $\{a_m\}$ khác không), nên mẫu đầu ra y[n] phụ thuộc ngược lại vào các mẫu đầu ra vừa tính trước đó. Đặc tính: hiệu suất rất cao, chỉ cần vài phép tính (bậc lọc nhỏ) đã đạt độ dốc cắt rất sắc nét. Tuy nhiên, bộ lọc IIR có nguy cơ mất ổn định (hệ thống tự dao động kéo dài vô hạn) nếu các hệ số thiết kế không chính xác.

> **MỐI LIÊN KẾT 4.1 — HIỆU ỨNG TIẾNG VỌNG ECHO CHÍNH LÀ BỘ LỌC IIR**
>
> Ý tưởng mẫu đầu ra phụ thuộc vào mẫu đầu ra trước đó có vẻ trừu tượng về mặt toán học, nhưng thực chất đó là sơ đồ khối của hiệu ứng tiếng vọng (Echo) có hồi tiếp, vốn sẽ được lập trình ở chương về hiệu ứng không gian.
>
> Mỗi tiếng vọng dội lại sau một khoảng thời gian trễ là một bản sao của mẫu đầu ra trước đó, được giảm nhẹ âm lượng (Feedback Gain < 1) và cộng ngược vào tín hiệu đầu vào hiện tại. Cấu trúc lặp lại vô hạn này chính là mô hình toán học của bộ lọc IIR. Như vậy, các hiệu ứng âm thanh không phải là những khái niệm độc lập mà đều được xây dựng trên cùng bộ khung toán học của bộ lọc số.

## 4.5 Phép toán tích chập

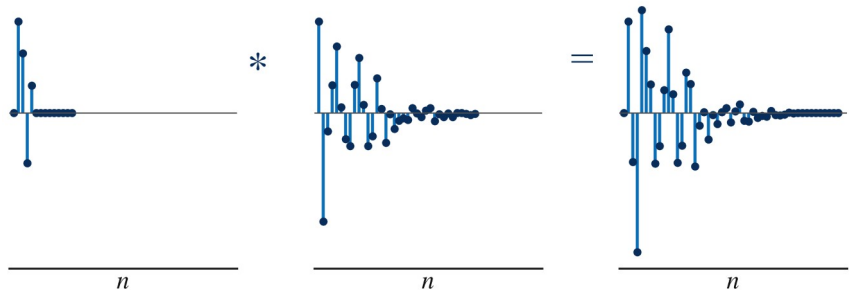
Tích chập (convolution, ký hiệu là phép toán *) là phép toán nền tảng của xử lý tín hiệu số và âm học phòng. Mặc dù công thức tổng chập trên dãy mẫu rời rạc trông phức tạp, ý nghĩa bản chất của nó đối với người làm âm thanh lại dễ hiểu và mang tính trực quan.

### 4.5.1 Bản chất: in dấu đặc tính hệ thống lên tín hiệu

Mỗi hệ thống âm học (một căn phòng, một hành lang bê tông, một thùng loa hay một bộ lọc số) đều có một 'dấu vân tay' vật lý đặc trưng gọi là đáp ứng xung (Impulse Response - viết tắt là IR) [1]. Đáp ứng xung là toàn bộ sóng âm dội lại mà hệ thống phát ra khi được kích thích bằng một xung điện hoặc xung áp suất tức thời rất ngắn, tương đương hàm delta Dirac lý tưởng; trong thực tế, đó là một tiếng 'tách' hoặc tiếng nổ rất đanh chứa năng lượng phân bổ đều ở mọi tần số.

**Ý nghĩa cốt lõi của phép tích chập:** tích chập một tín hiệu âm thanh đầu vào bất kỳ x[n] với đáp ứng xung h[n] của một hệ thống là hành vi 'in dấu' toàn bộ đặc tính âm học của hệ thống đó lên tín hiệu đầu vào, tạo ra tín hiệu đầu ra y[n] hoàn chỉnh:

$$y(n) = x(n) * h(n) = \sum_{k} x(k)\, h(n-k) \tag{4.4}$$



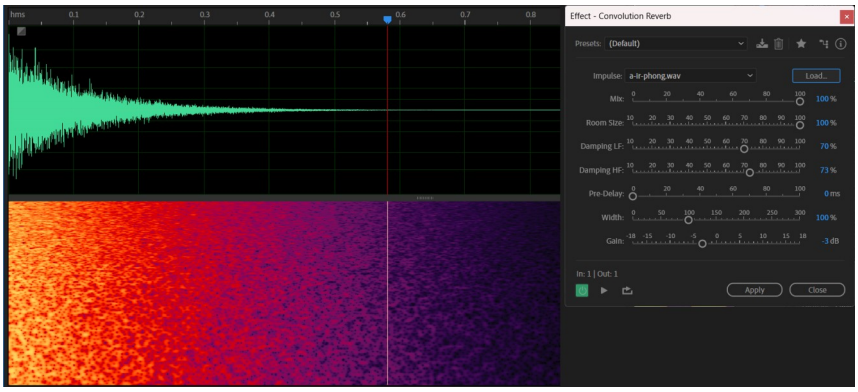

(a) tín hiệu khô $x[n]$ (b) đáp ứng xung $h[n]$ (c) tín hiệu ướt $y[n]$

*Hình 4.13 Phép tích chập: tín hiệu khô kết hợp với đáp ứng xung không gian tạo ra tín hiệu mang đặc tính vang dội tự nhiên*

### 4.5.2 Công nghệ vang tích chập

Xét trường hợp người thu âm bước vào một nhà thờ lớn cổ kính hay một hang đá rộng và vỗ tay thật mạnh một cái. Tiếng vỗ tay đanh gọn (xung kích thích) kích hoạt toàn bộ không gian phản xạ dội lại vang rền, kéo dài nhiều giây rồi tắt dần; đó chính là đáp ứng xung (IR) mô tả trọn vẹn đặc tính vang của nhà thờ. Đáp ứng xung này được thu lại bằng một bộ micro âm thanh nổi chất lượng cao.

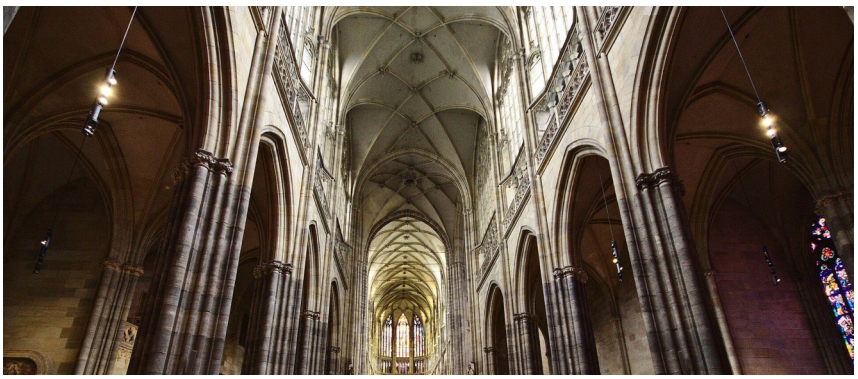
Hình 4.14 ghép dạng sóng và phổ đồ của tệp đáp ứng xung một căn phòng (bên trái) với hộp thoại Convolution Reverb đã nạp chính tệp ấy (bên phải): phần đuôi dạng sóng tắt dần chính là hồi âm mà bộ tích chập sẽ in lên tín hiệu đi qua.




*Hình 4.14 Đáp ứng xung phòng (tệp mẫu tổng hợp) và hộp thoại Convolution Reverb của Adobe Audition*

*Nguồn: ảnh chụp màn hình do người soạn thực hiện (Adobe Audition 2026, bản 26.5), 07/11/2026.*

Hình 4.15 là không gian bên trong một nhà thờ lớn kiểu Gothic với trần vòm cao và bề mặt đá phản xạ mạnh, loại không gian có đáp ứng xung kéo dài nhiều giây và thường được thu lại để làm vang tích chập.



*Hình 4.15 Gian giữa nhà thờ lớn St. Vitus (Praha), một không gian có độ vang dài thường được dùng để thu đáp ứng xung*

*Nguồn: Sage Ross, CC BY-SA 3.0, Wikimedia Commons (đã cắt) (truy cập 05/11/2026).*

Giả sử cần làm cho một giọng nói (được thu khô, sạch trong phòng thu cách âm) nghe như đang được cất lên giữa không gian nhà thờ cổ kính đó. Người làm âm thanh chỉ cần thực hiện phép tích chập giữa giọng nói khô và tệp đáp ứng xung nhà thờ đã thu được. Kết quả là giọng nói mang theo toàn bộ độ sâu, các đợt phản xạ sớm và đuôi vang tự nhiên của nhà thờ. Đây là nguyên lý hoạt động của hiệu ứng vang tích chập (Convolution Reverb).

### 4.5.3 Ví dụ số tính toán tích chập tối giản

Để hiểu bản chất phép tính số học diễn ra bên trong máy tính, người đọc thực hiện phép tích chập bằng tay với hai dãy số ngắn sau đây:

- **Tín hiệu vào miền thời gian:** x = [1, 2, 3]
- **Đáp ứng xung của một hệ thống trễ:** h = [0, 1, 0,5] (hệ thống làm trễ tín hiệu một mẫu, kèm theo một tiếng vọng nhỏ có biên độ bằng một nửa).

Phép tính tích chập y = x * h được thực hiện bằng cách trượt ngược dãy h qua dãy x và tính tổng tích các phần tử chồng lấn tại mỗi bước trượt:

- **Bước 0 (n = 0):** $y[0] = x[0]\cdot h[0] = 1 \cdot 0 = 0$
- **Bước 1 (n = 1):** $y[1] = x[1]\cdot h[0] + x[0]\cdot h[1] = 2\cdot 0 + 1\cdot 1 = 1$
- **Bước 2 (n = 2):** $y[2] = x[2]\cdot h[0] + x[1]\cdot h[1] + x[0]\cdot h[2] = 3\cdot 0 + 2\cdot 1 + 1\cdot 0{,}5 = 2{,}5$
- **Bước 3 (n = 3):** $y[3] = x[2]\cdot h[1] + x[1]\cdot h[2] = 3\cdot 1 + 2\cdot 0{,}5 = 4$
- **Bước 4 (n = 4):** $y[4] = x[2]\cdot h[2] = 3 \cdot 0{,}5 = 1{,}5$

Kết quả dãy đầu ra sau tích chập: y = [0, 1, 2,5, 4, 1,5]

**Nhận xét kỹ thuật:** kết quả đầu ra y có độ dài bằng (len(x) + len(h) - 1) = 3 + 3 - 1 = 5 mẫu. Dãy kết quả dài hơn tín hiệu vào vì mang theo cả phần 'đuôi' của đáp ứng xung dội lại. Về trực giác, đáp ứng xung của một phòng thực tế càng dài và phức tạp (gồm hàng chục nghìn mẫu biên độ phản xạ) thì đuôi tích chập càng dày; đây là lý do vang tích chập đòi hỏi năng lực tính toán lớn của máy tính.

## 4.6 Hướng dẫn thực hành

Ba bài thực hành dưới đây giúp người đọc chuyển các khái niệm lý thuyết trừu tượng về miền tần số, phổ Fourier, spectrogram và bộ lọc số thành các dòng lệnh Python thực tế. Người đọc tạo thư mục dự án `src/` và thực thi đầy đủ các đoạn mã nguồn dưới đây, vốn đã được kiểm chứng chạy đúng [9], [10].

### 4.6.1 Vẽ dạng sóng và phổ vạch hài Fourier

**Mục tiêu:** lập trình sinh tín hiệu sóng sin thuần và sóng vuông, thực hiện phép FFT để so sánh trực quan 'vân tay phổ' của hai loại sóng cơ bản này theo đúng lý thuyết chuỗi hài.

*Liệt kê 4.1 src/spectrum.py — Vẽ dạng sóng và phổ vạch hài Fourier của sóng sin và sóng vuông*

```
# -*- coding: utf-8 -*-
# src/spectrum.py
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal

fs = 44100  # Tần số lấy mẫu tiêu chuẩn CD
t = np.linspace(0, 1, fs, endpoint=False) # Thời lượng 1 giây

# 1. Sinh sóng sin thuần khiết 440 Hz (Nốt La chuẩn)
sine = 0.5 * np.sin(2 * np.pi * 440 * t)

# 2. Sinh sóng vuông 220 Hz (Chứa dải hài lẻ phong phú)
square = 0.4 * signal.square(2 * np.pi * 220 * t)

def show_spectrum(x, name):
    # Thực hiện phép biến đổi Fourier nhanh cho số thực rfft
    X = np.abs(np.fft.rfft(x)) / (len(x)/2)
    f = np.fft.rfftfreq(len(x), 1/fs)

    # Tìm đỉnh phổ mạnh nhất
    peak_idx = np.argmax(X)
    peak_freq = f[peak_idx]

    print(f"{name} -> Đỉnh phổ mạnh nhất tại: {peak_freq:.1f} Hz (Biên độ:
{X[peak_idx]:.4f})")

    # Vẽ đồ thị phổ
    plt.figure(figsize=(7, 3.5))
    plt.stem(f, X, linefmt='b', markerfmt='o', basefmt='gray')
    plt.title(f"Phổ miền tần số - {name}")
    plt.xlabel("Tần số (Hz)")
    plt.ylabel("Biên độ phổ")
    plt.xlim(0, 2000) # Chỉ quan sát dải dưới 2 kHz
    plt.ylim(0, 0.6)
    plt.grid(True)
    plt.tight_layout()
    plt.show()

if __name__ == "__main__":
    print("--- BÀI THỰC HÀNH 1: PHÂN TÍCH PHỔ FOURIER ---")
    show_spectrum(sine, "Sóng Sin thuần (440 Hz)")
    show_spectrum(square, "Sóng vuông (220 Hz)")
```

**Hướng dẫn quan sát:** khi chạy chương trình, sóng sin thuần khiết chỉ cho một vạch phổ duy nhất tại mốc 440 Hz, còn sóng vuông 220 Hz hiển thị một chuỗi vạch phổ cách đều nhau tại các tần số hài lẻ: 220 Hz (hài cơ bản $f_0$), 660 Hz (hài 3), 1.100 Hz (hài 5), 1.540 Hz (hài 7)... Kết quả này chứng minh sóng vuông được cấu thành từ tổng vô hạn các sóng sin hài lẻ, đúng với lý thuyết Fourier.

### 4.6.2 Vẽ phổ đồ thời gian của tín hiệu biến thiên

**Mục tiêu:** lập trình tạo tín hiệu quét tần số (Chirp) và vẽ biểu đồ spectrogram bằng hàm STFT để quan sát sự thay đổi liên tục của tần số theo thời gian.

*Liệt kê 4.2 src/spectro.py — Vẽ phổ đồ thời gian của tín hiệu quét tần (chirp)*

```
# -*- coding: utf-8 -*-
# src/spectro.py
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal

fs = 44100
t = np.linspace(0, 3, 3 * fs, endpoint=False) # Thời lượng 3 giây

# Sinh tín hiệu quét tần (Chirp) chạy tuyến tính từ 200 Hz lên 4000 Hz trong 3
giây
x = signal.chirp(t, f0=200, f1=4000, t1=3, method='linear')

# Tính toán STFT và vẽ Spectrogram
f, tt, Sxx = signal.spectrogram(x, fs, nperseg=1024, noverlap=512)

plt.figure(figsize=(7, 4))
# Chuyển đổi năng lượng phổ sang thang đo logarit Decibel (dB)
plt.pcolormesh(tt, f, 10 * np.log10(Sxx + 1e-12), shading='gouraud',
cmap='viridis')
plt.ylabel("Tần số (Hz)")
plt.xlabel("Thời gian (giây)")
plt.ylim(0, 6000) # Giới hạn quan sát dải dưới 6 kHz
plt.title("Spectrogram tín hiệu Chirp biến thiên (200 Hz -> 4000 Hz)")
plt.colorbar(label="Mức năng lượng (dB)")
plt.tight_layout()
plt.show()
```

**Hướng dẫn quan sát:** biểu đồ spectrogram hiển thị một đường chéo sáng đi từ tọa độ (0s, 200Hz) lên tọa độ (3s, 4000Hz). Người đọc nên thử thay đổi tham số kích thước khung `nperseg` (ví dụ giảm xuống 256 mẫu hoặc tăng lên 4.096 mẫu) để quan sát sự thay đổi của độ phân giải thời gian và tần số theo quy luật đánh đổi Heisenberg.

### 4.6.3 Thiết kế bộ lọc số và kiểm chứng sự thay đổi âm sắc

**Mục tiêu:** lập trình sử dụng bộ lọc Butterworth [8] để lọc thông thấp và thông cao trên tệp âm thanh thực tế `input.wav`, xuất ra các tệp kết quả để nghe và thẩm âm sự thay đổi vật lý.

*Liệt kê 4.3 src/filter_demo.py — Lọc thông thấp và thông cao bằng bộ lọc Butterworth*

```
# -*- coding: utf-8 -*-
# src/filter_demo.py
import numpy as np
import soundfile as sf
from scipy import signal

def run_filter_demo():
    try:
        # Đọc tệp âm thanh đầu vào (yêu cầu chuẩn bị sẵn file input.wav trong
thư mục)
        x, fs = sf.read('input.wav')
    except Exception as e:
        print("Lỗi: Không tìm thấy file input.wav! Đang tự động sinh tín hiệu
nhiễu mô phỏng...")
        fs = 44100
        x = np.random.normal(0, 0.2, fs * 3) # Nhiễu trắng dài 3 giây
        sf.write('input.wav', x, fs)

    if x.ndim > 1:
        x = x[:, 0]  # Lấy 1 kênh mono cho đơn giản

    # 1. Thiết kế bộ lọc thông thấp (Butterworth bậc 4, cắt tại 1000 Hz)
    # Tần số cắt được chuẩn hóa theo dải tần Nyquist (fs/2)
    fc_low = 1000.0
    sos_lp = signal.butter(4, fc_low / (fs / 2), btype='low', output='sos')
    y_lp = signal.sosfilt(sos_lp, x)
    sf.write('out_lowpass.wav', y_lp, fs)

    # 2. Thiết kế bộ lọc thông cao (Butterworth bậc 4, cắt tại 3000 Hz)
    fc_high = 3000.0
    sos_hp = signal.butter(4, fc_high / (fs / 2), btype='high', output='sos')
    y_hp = signal.sosfilt(sos_hp, x)
    sf.write('out_highpass.wav', y_hp, fs)

    print("--- HOÀN THÀNH XỬ LÝ LỌC SỐ ---")
    print("Ghi thành công out_lowpass.wav -> Âm sắc tối, trầm ấm (mất hoàn
toàn dải treble).")
    print("Ghi thành công out_highpass.wav -> Âm sắc mỏng, sáng mảnh (mất hoàn
toàn dải bass).")

if __name__ == "__main__":
    run_filter_demo()
```

**Hướng dẫn thẩm âm:** người đọc đeo tai nghe kiểm âm để nghe hai tệp kết quả. Tệp `out_lowpass.wav` nghe tối, ngột ngạt và ấm, như âm thanh truyền qua một bức tường gạch dày. Ngược lại, tệp `out_highpass.wav` nghe mỏng, sắc và gợi cảm giác xa xăm như tiếng phát ra từ một chiếc loa phát thanh di động.

## TÓM TẮT CHƯƠNG 4

1. **Hai miền bổ sung:** Một mảng âm thanh số có thể quan sát theo hai cách bổ sung mật thiết cho nhau: miền thời gian (dạng sóng biên độ theo thời gian) và miền tần số (phân bổ năng lượng phổ theo tần số).
2. **Phép biến đổi Fourier:** Biến đổi Fourier nhanh (FFT) là thuật toán tối ưu giúp máy tính phân rã tín hiệu thành tổng các sóng sin đơn tần thành phần, và đỉnh cao nhất của phổ biên độ chỉ rõ tần số mạnh nhất của tín hiệu.

3. **Đánh đổi độ phân giải:** Kích thước khung phân tích FFT càng dài thì độ phân giải tần số càng mịn (xác định chính xác cao độ nốt nhạc) nhưng độ phân giải thời gian càng thô (thời điểm bị nhòe), và ngược lại.
4. **STFT và Spectrogram:** STFT trượt một cửa sổ ngắn có chồng lấn dọc theo tín hiệu và chạy FFT trên từng khung để tạo ra biểu đồ phổ theo thời gian (Spectrogram), trong đó tần số được biểu diễn trên trục dọc, thời gian trên trục ngang và năng lượng bằng màu sắc.
5. **Bộ lọc số:** Là hệ thống toán học biến đổi phổ tần (LPF cắt treble, HPF cắt tiếng ù bass, BPF lọc thông dải thoại điện thoại, Notch triệt tiếng ù 50 Hz); bộ lọc FIR không hồi tiếp nên ổn định tuyệt đối, còn bộ lọc IIR có vòng hồi tiếp nên đạt hiệu suất cao.
6. **Tích chập và vang tích chập:** Phép tích chập (Convolution) cho phép in dấu đáp ứng xung (đặc trưng âm học) của một căn phòng lên tín hiệu thoại khô sạch, tạo ra hiệu ứng vang tích chập (Convolution Reverb) có độ trung thực cao nhất trong các loại vang.

## 4.7 Hệ thống bài tập và câu hỏi ôn tập

### 4.7.1 Câu hỏi ôn tập lý thuyết

**Câu hỏi 1:** Hãy phân biệt miền thời gian và miền tần số, đồng thời cho biết mỗi cách quan sát cung cấp thông tin cốt lõi nào cho kỹ sư lập trình âm thanh.

**Câu hỏi 2:** Hãy phát biểu ý tưởng toán học cốt lõi của phép biến đổi Fourier và chỉ ra sự khác nhau giữa phép toán DFT với thuật toán FFT.

**Câu hỏi 3:** Vì sao không thể đồng thời đạt độ phân giải thời gian và độ phân giải tần số cực đại? Hãy lấy một ví dụ minh họa thực tế.

**Câu hỏi 4:** Hãy trình bày nguyên lý hoạt động của biến đổi STFT và giải thích vì sao việc nhân khung tín hiệu với hàm cửa sổ (Window Function) là bắt buộc.

**Câu hỏi 5:** Hãy phân tích ưu và nhược điểm của hai họ bộ lọc số FIR và IIR, đồng thời giải thích vì sao hiệu ứng tiếng vọng Echo có hồi tiếp thực chất là một bộ lọc IIR.

**Câu hỏi 6:** Hãy giải thích ý nghĩa của phép toán tích chập và khái niệm đáp ứng xung (Impulse Response) bằng ẩn dụ về tiếng vang dội lại trong một nhà thờ.

### 4.7.2 Bài tập vận dụng

Các bài tập dưới đây mô phỏng một số tình huống kỹ thuật thường gặp khi phân tích và xử lý âm thanh:

**Bài tập 1 (Phân tích và đọc hiểu biểu đồ phổ tần số):**

Người làm âm thanh nhận được ba tệp âm thanh không dán nhãn; khi chạy phép phân tích FFT, thu được ba biểu đồ phổ đặc trưng sau đây:

– **Phổ A:** Xuất hiện duy nhất một đỉnh nhọn, dứt khoát tại tần số 1.000 Hz.

- **Phổ B:** Xuất hiện một chuỗi các vạch đỉnh nhọn cách đều nhau tại các mốc tần số 150 Hz, 300 Hz, 450 Hz, 600 Hz, 750 Hz...
- **Phổ C:** Đồ thị gồ ghề, năng lượng phân bổ liên tục và trải rộng khắp toàn bộ dải thông từ 20 Hz đến 20 kHz, không xuất hiện đỉnh nhọn rõ rệt nào.

Hãy phân tích bản chất vật lý và xác định loại nguồn âm thanh tương ứng với mỗi biểu đồ phổ trên.

**BÀI TẬP 4.1 — HƯỚNG DẪN GIẢI**

**Biểu đồ phổ A:** Tương ứng với một sóng sin đơn âm thuần khiết (Pure Tone) có tần số f = 1.000 Hz. Đây là đơn âm chuẩn quy chiếu, thường được sử dụng trong phòng thu để cân chuẩn thiết bị đo lường.

**Biểu đồ phổ B:** Tương ứng với một âm thanh có cao độ rõ ràng và giàu âm sắc hài âm (Harmonic sound). Tần số cơ bản $f_0$ quyết định cao độ nốt nhạc chính là đỉnh đầu tiên $f_0$ = 150 Hz (giọng nam trầm, nốt trầm của đàn cello), còn các đỉnh tiếp theo là chuỗi hài âm cách đều theo bội số nguyên (300Hz = $2f_0$, 450Hz = $3f_0$, 600Hz = $4f_0$...).

**Biểu đồ phổ C:** Tương ứng với một tiếng động phi điều hòa phức tạp, không có cao độ rõ ràng (Noise / Non-harmonic transient), ví dụ như tiếng mưa rơi, tiếng gió rít dải rộng hoặc tiếng xì trắng (white noise) của hệ thống loa.

**Bài tập 2 (Tính toán và đánh đổi độ phân giải thời gian - tần số):**

Một hệ thống xử lý âm thanh được thiết lập với tần số lấy mẫu $f_s$ = 48.000 Hz. Hãy thực hiện các tính toán kỹ thuật sau:

a) Tính độ phân giải tần số Δf (Hz) và độ dài thời gian của một khung phân tích (ms) khi kỹ sư cấu hình kích thước khung FFT là N = 512 mẫu.

b) Tính các thông số tương ứng khi tăng kích thước khung lên N = 4.800 mẫu.

c) Trong trường hợp cần phân biệt chính xác hai nốt nhạc trầm chỉ cách nhau 5 Hz trong bản hòa âm, kỹ sư nên lựa chọn cấu hình khung nào, và cái giá phải trả về mặt định vị thời gian là gì?

**BÀI TẬP 4.2 — HƯỚNG DẪN GIẢI**

a) Với kích thước khung N = 512 mẫu ở fs = 48.000 Hz:

• **Độ phân giải tần số:** $\Delta f = fs / N = 48.000 / 512 = 93,75$ Hz (thang phân giải thô).

• **Độ dài thời gian của khung:** $T_{khung} = N / fs = 512 / 48.000 \approx 0,0107$ giây = 10,7 ms (định vị thời gian tốt).

b) Với kích thước khung N = 4.800 mẫu ở fs = 48.000 Hz:

• **Độ phân giải tần số:** $\Delta f = fs / N = 48.000 / 4.800 = 10$ Hz (thang phân giải mịn hơn nhiều).

• **Độ dài thời gian của khung:** $T_{khung} = N / fs = 4.800 / 4.800 = 0,1$ giây = 100 ms (định vị thời gian thô).

c) Phân tích lựa chọn cho bài toán phân biệt 5 Hz:

• Để phân biệt được độ lệch tần hẹp 5 Hz, độ phân giải tần số bắt buộc phải mịn hơn 5 Hz ($\Delta f \leq 5$ Hz), do đó cả hai cấu hình trên đều không đáp ứng được (93,75 Hz và 10 Hz đều lớn hơn 5 Hz). Kỹ sư cần tăng kích thước khung lên tối thiểu N = 9.600 mẫu để đạt độ phân giải $\Delta f = 48.000 / 9.600 = 5$ Hz.

• **Cái giá phải trả:** khung dài 9.600 mẫu tương ứng với độ dài thời gian lên tới $T_{khung} = 9.600 / 48.000 = 200$ ms. Khoảng thời gian 200 ms là quá lớn đối với thính giác, khiến thông tin biên độ bị nhòe rộng theo trục thời gian, nên kỹ sư mất hoàn toàn khả năng định vị chính xác đến mili-giây thời điểm xuất hiện của các tiếng gõ đập (transient) nhanh.

**Bài tập 3 (Lựa chọn bộ lọc số cho các tình huống xử lý âm thanh):**

Hãy lựa chọn loại bộ lọc số phù hợp nhất (Thông thấp, Thông cao, Thông dải, Triệt dải/Notch) để xử lý tối ưu cho bốn tình huống âm thanh thực tế sau:

- **Tình huống 1:** Bản thu tiếng thoại ngoài trời bị lẫn tạp âm gió trầm, rền, gây khó chịu khi nghe.
- **Tình huống 2:** Cần mô phỏng giọng nói phát qua một chiếc điện thoại di động đời cũ, yêu cầu âm sắc mỏng và nghẹt.
- **Tình huống 3:** Bản thu thoại trong phòng thu bị lẫn tiếng ù xoay chiều tại tần số 50 Hz, rất đều đặn, do lưới điện.
- **Tình huống 4:** Thiết lập lọc bỏ dải tần số cao vượt ngưỡng nghe của con người để chống hiện tượng méo chồng phổ (răng cưa) trước khi tín hiệu đưa vào bộ chuyển đổi số ADC.

**BÀI TẬP 4.3 — HƯỚNG DẪN GIẢI**

**Tình huống 1:** Sử dụng bộ lọc Thông cao (High-pass Filter). Cài đặt tần số cắt fc khoảng 80 Hz - 100 Hz để loại bỏ toàn bộ năng lượng rền ở dải cực trầm của tiếng gió mà không gây tổn hao đến dải tần formant chính của giọng nói.

**Tình huống 2:** Sử dụng bộ lọc Thông dải (Band-pass Filter). Cài đặt dải thông hẹp, chỉ cho phép tần số từ 300 Hz đến 3.400 Hz đi qua và triệt tiêu hoàn toàn dải trầm lẫn dải treble cao, nhờ đó tái hiện đúng màu sắc nghẹt đặc trưng của loa điện thoại.

**Tình huống 3:** Sử dụng bộ lọc Triệt dải / Cắt rãnh (Notch Filter). Thiết lập rãnh cắt cực hẹp tại tần số trung tâm đúng 50 Hz để triệt tiêu hoàn toàn tiếng ù của điện lưới mà không ảnh hưởng đến năng lượng lời thoại ở các dải tần lân cận.

**Tình huống 4:** Sử dụng bộ lọc Thông thấp (Low-pass Filter). Cài đặt tần số cắt fc sát mốc tần số Nyquist (thường chọn fc = 20 kHz ở fs = 44.1 kHz) để triệt tiêu toàn bộ nhiễu cao tần vượt ngưỡng nghe trước khi lấy mẫu.

**Bài tập 4 (Tính toán tích chập bằng tay và giải nghĩa vật lý):**

Thực hiện phép tính tích chập miền thời gian của tín hiệu thoại ngắn x = [2, 0, 1] với đáp ứng xung của hệ thống h = [1, 1], sau đó cho biết dãy số đầu ra phản ánh hệ thống này thực hiện hiệu ứng vật lý gì trên tín hiệu.

**BÀI TẬP 4.4 — HƯỚNG DẪN GIẢI**

Thực hiện thuật toán tích chập tay y = x * h:

• y[0] = x[0]·h[0] = 2 · 1 = 2

• y[1] = x[1]·h[0] + x[0]·h[1] = 0·1 + 2·1 = 2

• y[2] = x[2]·h[0] + x[1]·h[1] + x[0]·h[2] (h[2]=0) = 1·1 + 0·1 = 1

• y[3] = x[2]·h[1] + x[1]·h[2] (h[2]=0) = 1 · 1 = 1

**Kết quả dãy đầu ra sau tích chập:** y = [2, 2, 1, 1]

**Giải nghĩa vật lý hệ thống h = [1, 1]:**

Đáp ứng xung h = [1, 1] có nghĩa là hệ thống cộng mẫu tín hiệu hiện tại với chính mẫu đó trễ đi một bước thời gian (y[n] = x[n] + x[n-1]). Đây là mô hình toán học tối giản của một bộ lọc làm mịn dải cao (một bộ lọc thông thấp đơn giản), tạo ra một tiếng dội tức thời rất ngắn gắn liền với âm thanh gốc.

## 4.8 Thuật ngữ chuyên ngành Anh – Việt

Bảng tra cứu thuật ngữ kỹ thuật chuyên sâu phục vụ nghiên cứu và lập trình âm thanh quốc tế:

*Bảng 4.3 Bảng thuật ngữ chuyên ngành Xử lý tín hiệu âm thanh số*

| Thuật ngữ tiếng Anh | Thuật ngữ tiếng Việt | Ý nghĩa chuyên môn và Ngữ cảnh ứng dụng |
|---|---|---|
| Time domain | Miền thời gian | Biểu diễn biên độ áp suất biến thiên theo thời gian tuyến tính |
| Frequency domain | Miền tần số | Biểu diễn dải năng lượng phân bổ theo tần số dao động |

| Thuật ngữ tiếng Anh | Thuật ngữ tiếng Việt | Ý nghĩa chuyên môn và Ngữ cảnh ứng dụng |
|---|---|---|
| Spectrum | Phổ âm thanh | Đồ thị biểu diễn biên độ hoặc công suất theo trục tần số (Hz) |
| Fourier Transform | Biến đổi Fourier | Phép toán phân rã tín hiệu thành tổng vô hạn các sóng sin |
| Fast Fourier Transform (FFT) | Biến đổi Fourier nhanh | Thuật toán tối ưu tính toán DFT tốc độ cao trong thực tế |
| Short-Time Fourier Transform | Biến đổi Fourier thời gian ngắn | Chạy FFT trên các khung trượt cửa sổ gối chồng lấn nhau |
| Spectrogram | Phổ đồ / Biểu đồ phổ-thời gian | Bức tranh trực quan hóa tần số theo thời gian (trục màu là dB) |
| Spectral Leakage | Hiện tượng rò rỉ phổ | Méo toán học làm xuất hiện tần số giả do cắt khung đột ngột |
| Window Function | Hàm cửa sổ làm mềm | Hàm nhân làm mượt hai đầu khung tín hiệu (Hann, Hamming) |
| Digital Filter | Bộ lọc số | Hệ thống thuật toán giữ lại hoặc làm yếu các dải tần chọn lọc |
| FIR Filter | Bộ lọc đáp ứng xung hữu hạn | Bộ lọc không hồi tiếp, chỉ phụ thuộc mẫu vào, ổn định tuyệt đối |
| IIR Filter | Bộ lọc đáp ứng xung vô hạn | Bộ lọc có nhánh hồi tiếp, phụ thuộc cả mẫu ra trước đó |
| Difference Equation | Phương trình sai phân | Công thức toán học tính mẫu đầu ra từ mẫu vào và quá khứ |
| Convolution | Phép toán Tích chập | Phép toán in đặc tính đáp ứng xung của hệ thống lên tín hiệu |
| Impulse Response (IR) | Đáp ứng xung của hệ thống | Toàn bộ âm dội hệ thống phát ra khi nhận một xung kích thích |

## TÀI LIỆU THAM KHẢO CHƯƠNG 4

[1] Oppenheim, A. V., & Schafer, R. W. (2010). *Discrete-Time Signal Processing* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

[2] Smith, J. O. (2007). *Mathematics of the Discrete Fourier Transform (DFT) with Audio Applications* (2nd ed.). Stanford University: W3K Publishing.

[3] Lyons, R. G. (2010). *Understanding Digital Signal Processing* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

[4] Nguyễn An Sỹ. (2018). *Giáo trình Kỹ thuật âm thanh số và Truyền hình*. Hà Nội: Nhà xuất bản Khoa học và Kỹ thuật.

[5] Cooley, J. W., & Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90), 297–301.

[6] Harris, F. J. (1978). On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*, 66(1), 51–83.

[7] Allen, J. B., & Rabiner, L. R. (1977). A unified approach to short-time Fourier analysis and synthesis. *Proceedings of the IEEE*, 65(11), 1558–1564.

[8] Butterworth, S. (1930). On the theory of filter amplifiers. *Experimental Wireless and the Wireless Engineer*, 7, 536–541.

[9] Virtanen, P., Gommers, R., Oliphant, T. E., et al. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17, 261–272.

[10] Harris, C. R., Millman, K. J., van der Walt, S. J., et al. (2020). Array programming with NumPy. *Nature*, 585, 357–362.