MỤC LỤC

Chương 4. Cơ sở xử lý tín hiệu số.....5
4.1 Vị trí của chương trong cuốn sách.....5
4.2 Biến đổi Fourier rời rạc và thuật toán FFT.....8
4.3 Đọc một biểu đồ phổ.....14
4.4 Cửa sổ và rò phổ.....19
4.5 Biến đổi Fourier thời gian ngắn và phổ đồ.....24
4.6 Bộ lọc số.....28
4.7 Phép toán tích chập.....34
4.8 Thực hành và bài tập.....39
4.9 Tổng kết chương.....40

DANH MỤC HÌNH VẼ

Hình 4.1 Sáu công cụ của chương 4 và trình tự của chúng.....6
Hình 4.2 Cùng một dãy số, hai cách nhìn — mỗi cách trả lời một loại câu hỏi khác nhau.....7
Hình 4.3 Ẩn dụ lăng kính: phép tách không thêm gì vào cái được tách.....7
Hình 4.4 Cộng dần các hài lẻ của 100 Hz: một, hai, năm, rồi hai mươi lăm sóng sin.....8
Hình 4.5 Ý tưởng của DFT: nhân với một sóng sin mẫu rồi cộng — tổng lớn nghĩa là giống.....9
Hình 4.6 Với N = 32 mẫu ở 8.000 Hz, FFT cho 17 vạch cách đều 250 Hz.....10
Hình 4.7 Hai tần số 440 và 520 Hz với ba độ dài khung: 256, 1.024 và 8.192 mẫu.....11
Hình 4.8 Chi phí tính của DFT trực tiếp và của FFT, và tỉ lệ giữa hai cách.....12
Hình 4.9 Khung mã tối thiểu để có một biểu đồ phổ đúng đơn vị.....13
Hình 4.10 Cùng một tín hiệu, bốn cách đặt trục — và chỉ một cách đọc được.....15
Hình 4.11 Bốn nguồn âm điển hình và bốn dạng phổ đặc trưng của chúng.....15
Hình 4.12 Ba tín hiệu có phổ biên độ giống hệt nhau nhưng dạng sóng khác hẳn.....17
Hình 4.13 Hai tín hiệu cùng phổ biên độ: pha bằng không cho tiếng gõ, pha ngẫu nhiên cho tiếng nhiễu.....17
Hình 4.14 Chỗ nối giữa hai bản lặp: mượt khi số chu kỳ chẵn, gãy khi không chẵn.....19
Hình 4.15 Cùng một sóng sin sạch: trùng vạch, lệch vạch, và lệch vạch có cửa sổ Hann 20
Hình 4.16 Bốn cửa sổ thông dụng: dạng theo thời gian và đáp ứng tần số.....21
Hình 4.17 Chọn cửa sổ là chọn một điểm trên đường đánh đổi — góc tốt nhất thì trống 22
Hình 4.18 Một thành phần yếu −58 dB bị sàn rò che lấp, và hiện ra khi đổi cửa sổ.....22
Hình 4.19 Một FFT cho cả tín hiệu thấy đủ ba tần số nhưng mất sạch thứ tự thời gian 24
Hình 4.20 Cơ chế STFT: khung dài N trượt dọc tín hiệu, mỗi lần nhảy H mẫu.....25
Hình 4.21 Phổ đồ một đoạn giọng nói: bốn thứ đọc được và ý nghĩa của chúng.....26
Hình 4.22 Hai đường Δf và Δt luôn đi ngược chiều nhau — không kéo cả hai xuống được.....27
Hình 4.23 Cùng một tín hiệu, ba độ dài khung: 128, 1.024 và 8.192 mẫu.....27
Hình 4.24 Đáp tuyến tần số của bốn kiểu lọc số cơ bản, dựng từ bộ lọc thật.....29
Hình 4.25 Đáp ứng xung của một bộ lọc FIR và một bộ lọc IIR.....31
Hình 4.26 Cùng một đáp tuyến biên độ, hai kiểu trễ theo tần số hoàn toàn khác nhau. 32
Hình 4.27 Cùng một bộ echo hồi tiếp với ba hệ số: ổn định, ổn định chậm, và phân kỳ 32
Hình 4.28 Phổ đồ trước và sau khi lọc thông dải, vẽ trên cùng một thang màu.....33
Hình 4.29 Tích chập của [1; 2; 3] với [0; 1; 0,5] làm từng bước một.....35
Hình 4.30 Lọc bằng vòng lặp sai phân và chập bằng np.convolve cho kết quả trùng khít 36
Hình 4.31 Tín hiệu khô chập với đáp ứng xung của phòng cho tín hiệu mang đặc tính phòng ấy.....37
Hình 4.32 Kiểm chứng định lý bằng số: hai con đường cho cùng một kết quả.....38
Hình 4.33 Chi phí của chập trực tiếp và của chập qua FFT theo độ dài đáp ứng xung..38
Hình 4.34 Bản đồ khái niệm của chương 4 — bảy ô và các đường nối.....40
Hình 4.35 Bảng quyết định: câu hỏi nào thì dùng công cụ nào.....41

DANH MỤC BẢNG BIỂU

Bảng 4.1 Độ phân giải tần số theo độ dài khung, ở 44.100 Hz....10
Bảng 4.2 Đệm số không: làm gì và không làm gì....13
Bảng 4.3 Bốn câu hỏi bắt buộc trước khi đọc một biểu đồ phổ....14
Bảng 4.4 Bốn dấu hiệu cần nhận ra trên một biểu đồ phổ....16
Bảng 4.5 Ba cách trình bày trục tung và ý nghĩa của chúng....18
Bảng 4.6 Năng lượng rò đo trên một sóng sin sạch, khung 1.024 mẫu....20
Bảng 4.7 Bốn cửa sổ thông dụng và đặc trưng của chúng....21
Bảng 4.8 Số mẫu cần để tách hai tần số, cửa sổ Hann, ở 44.100 Hz....23
Bảng 4.9 Ba tham số của mọi phổ đồ và ảnh hưởng của chúng....25
Bảng 4.10 Bốn dấu hiệu trên phổ đồ giọng nói....26
Bảng 4.11 Chọn độ dài khung theo việc cần làm, ở 44.100 Hz....27
Bảng 4.12 Bốn kiểu lọc và việc chúng làm trong phòng thu....29
Bảng 4.13 FIR và IIR — so sánh theo năm mặt....31
Bảng 4.14 Ví dụ số tính được bằng tay....35
Bảng 4.15 Ba bài thực hành và tiêu chí hoàn thành....39
Bảng 4.16 Ba lỗi hay gặp và cách tự phát hiện....40

CHƯƠNG 4
CƠ SỞ XỬ LÝ TÍN HIỆU SỐ

Mục tiêu của chương

Sau chương này, người đọc làm được những việc sau.

- Giải thích được quan hệ giữa miền thời gian và miền tần số, và nói được mỗi miền trả lời loại câu hỏi nào.
- Nói được bằng lời biến đổi Fourier rời rạc (DFT) làm gì mà không cần tới số phức, và giải thích được vì sao thuật toán biến đổi Fourier nhanh (FFT) nhanh hơn.
- Tính được độ phân giải tần số và thời lượng khung từ N và tần số lấy mẫu, và đọc đúng bốn trục của một biểu đồ phổ.
- Giải thích được rò phổ từ nguyên nhân gốc, chọn được cửa sổ hợp lý, và tính được số mẫu cần để tách hai tần số cho trước.
- Đọc được một phổ đồ giọng nói, chọn được độ dài khung theo việc cần làm, và phát biểu được luật đánh đổi thời gian – tần số.
- Phân biệt được bốn kiểu lọc và hai họ bộ lọc đáp ứng xung hữu hạn (FIR) và đáp ứng xung vô hạn (IIR), nêu được cái giá của mỗi họ, và kiểm được điều kiện ổn định.
- Giải thích được tích chập bằng ba động tác, chứng minh được lọc FIR chính là tích chập, và nói được vì sao vang tích chập phải làm qua FFT.

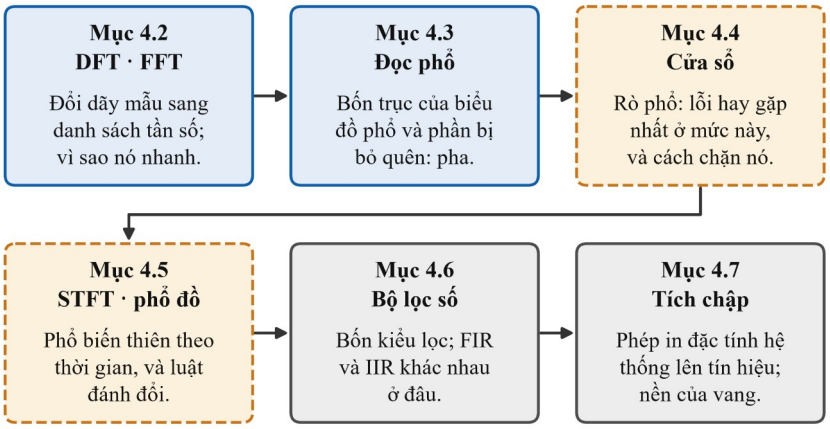
> **Chương này khác ba chương trước ở chỗ nào**
>
> Ba chương đầu trả lời câu hỏi "âm thanh đến được máy tính bằng cách nào". Chương này trả lời câu hỏi "làm được gì với nó khi nó đã ở đây".
>
> Vì thế đây là chương có nhiều toán nhất của cuốn sách — nhưng cũng là chương cuối cùng như thế. Từ chương 5 trở đi mọi chương đều kết thúc bằng mã chạy được, và chúng chỉ dùng lại đúng những công cụ dựng ở đây.
>
> Nếu chỉ có thời gian đọc kỹ *một* mục, hãy đọc mục 4.4 — nó là mục ít được dạy nhất và cũng là chỗ hay sai nhất trong thực tế.

4.1 Vị trí của chương trong cuốn sách

4.1.1 Ba chương đã qua và chỗ đứng của chương này

Ba chương đầu của cuốn sách đi theo một đường thẳng. Chương 1 nói về âm thanh như một hiện tượng vật lý: sóng, tần số, hài, decibel. Chương 2 nói về âm thanh khi nó đã thành điện áp chạy trong dây: chuỗi tín hiệu, mức chuẩn, dải động, nhiễu. Chương 3 nói về phép biến âm thanh thành số: lấy mẫu, lượng tử hoá, dBFS, định dạng.

Kết quả của ba chương ấy là một thứ duy nhất: *Một mảng số thực một chiều* nằm trong bộ nhớ máy tính, kèm theo một con số tần số lấy mẫu. Từ chương này trở đi, mọi thứ diễn ra trên mảng số ấy, và câu hỏi đổi hẳn: không còn là "âm thanh đến được máy tính bằng cách nào" mà là "làm được gì với nó khi nó đã ở đây".



Hình 4.1 Sáu công cụ của chương 4 và trình tự của chúng

Một lời về mức độ khó

Đây là chương có nhiều toán nhất của cuốn sách. Nhưng cần nói rõ ngay: chương này không dạy chứng minh, và phần bài tập cũng không hỏi chứng minh.

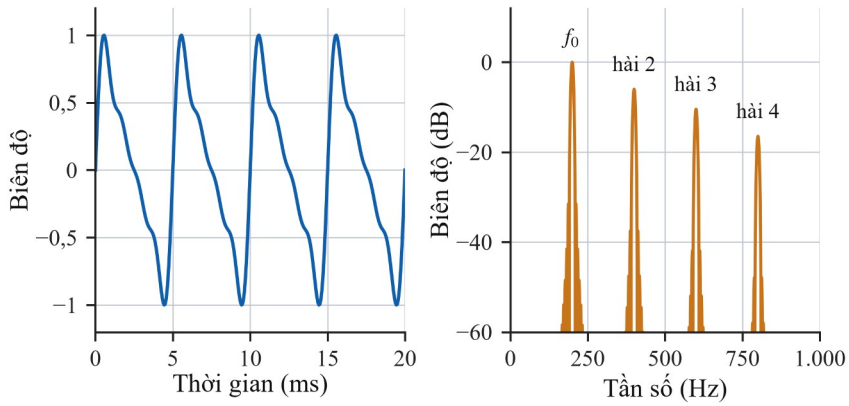
Cái được hỏi là: đọc được một biểu đồ phổ, chọn được tham số hợp lý cho một phép phân tích, giải thích được vì sao kết quả trông như thế, và nhận ra được khi nào con số máy tính đưa ra là vô nghĩa.

Mỗi công thức trong chương đều đi kèm một hình dựng từ tín hiệu thật và một con số kiểm được bằng mã. Nếu công thức khó nhớ, hãy nhớ hình.

4.1.2 Hai cách nhìn một tín hiệu

Ý tưởng nền của cả chương chỉ có một câu: *Một* tín hiệu âm thanh có *hai* cách mô tả đầy đủ và tương đương nhau. Không phải hai phần bổ sung cho nhau, mà hai bản dịch trọn vẹn của cùng một nội dung — từ bản này luôn dựng lại được bản kia mà không mất gì.

- **Miền thời gian** — dãy giá trị theo thứ tự thời gian. Trả lời câu hỏi: tại mỗi thời điểm, tín hiệu mạnh hay yếu ra sao? Đây là dạng sóng chúng ta thấy khi mở một tệp WAV trong Audacity.
- **Miền tần số** — danh sách các thành phần tần số và mức đóng góp của mỗi thành phần. Trả lời câu hỏi: âm thanh này gồm những tần số nào, mỗi tần số mạnh bao nhiêu? Đây là biểu đồ phổ.

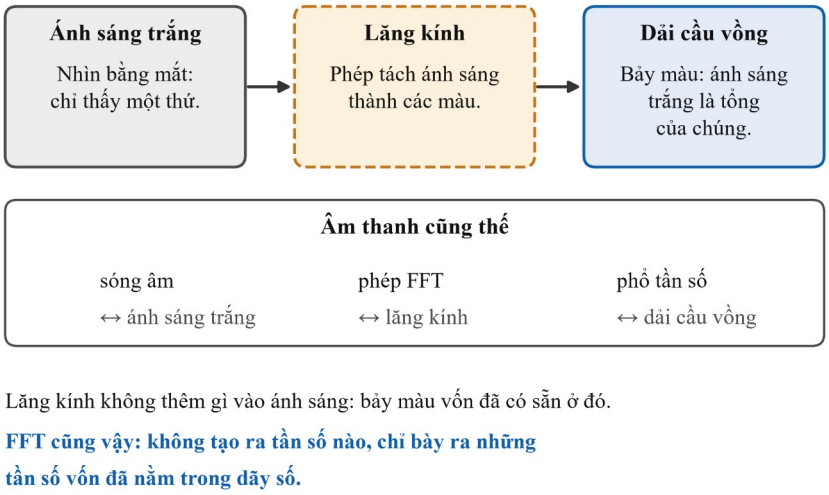


Hình 4.2 Cùng một dãy số, hai cách nhìn — mỗi cách trả lời một loại câu hỏi khác nhau

Người mới học hay hỏi cách nào đúng hơn. Câu hỏi ấy không có nghĩa, giống như hỏi bản vẽ mặt bằng đúng hơn hay bản vẽ mặt đứng đúng hơn. Người làm nghề chuyển qua lại giữa hai cách nhìn liên tục, và biết chuyển lúc nào chính là một phần lớn của kỹ năng.

4.1.3 Ẩn dụ lăng kính và giới hạn của nó

Ẩn dụ hay dùng nhất để mở đầu phần này là lăng kính: một tia sáng trắng đi qua khối thuỷ tinh thì tách ra thành dải cầu vồng. Ẩn dụ này tốt ở đúng một điểm quan trọng, và cần được dùng đúng ở điểm ấy.



Hình 4.3 Ẩn dụ lăng kính: phép tách không thêm gì vào cái được tách

Ẩn dụ này đúng ở đâu và sai ở đâu

Đúng: lăng kính không tạo ra màu nào cả. Bảy màu vốn đã nằm sẵn trong tia sáng trắng, lăng kính chỉ bày chúng ra. Phép biến đổi Fourier nhanh (FFT, tiếng Anh: Fast Fourier Transform) cũng thế — nó không sinh ra tần số nào, chỉ bày ra những tần số đã có trong dãy số.

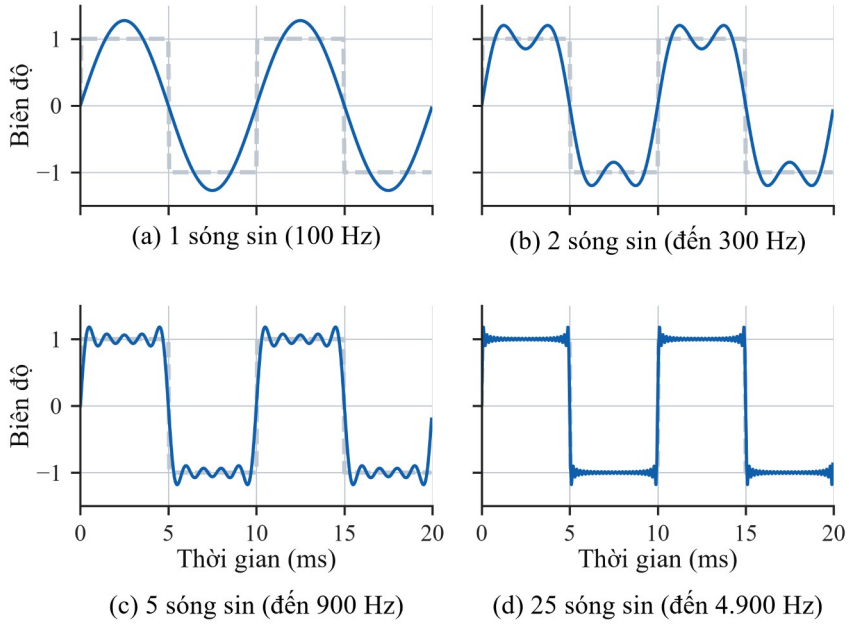
Sai: lăng kính là một vật thật, làm việc theo thời gian thực và không cần biết ánh sáng kéo dài bao lâu. FFT thì phải nhìn một *đoạn* tín hiệu có độ dài hữu hạn, và chính độ dài hữu hạn ấy sinh ra gần hết những khó khăn của chương này — rò phổ ở mục 4.4, luật đánh đổi ở mục 4.5.

Kết luận thực hành: dùng ẩn dụ để mở bài, rồi bỏ nó đi ngay khi bắt đầu bàn tới tham số.

4.2 Biến đổi Fourier rời rạc và thuật toán FFT

4.2.1 Ý tưởng gốc: mọi tín hiệu là một tổng các sóng sin

Chiều dễ hiểu trước. Nếu cộng nhiều sóng sin có tần số khác nhau lại, chúng ta được một dạng sóng phức tạp. Cộng đủ nhiều và đúng tỉ lệ thì được gần như bất kỳ hình gì.



Hình 4.4 Cộng dần các hài lẻ của 100 Hz: một, hai, năm, rồi hai mươi lăm sóng sin

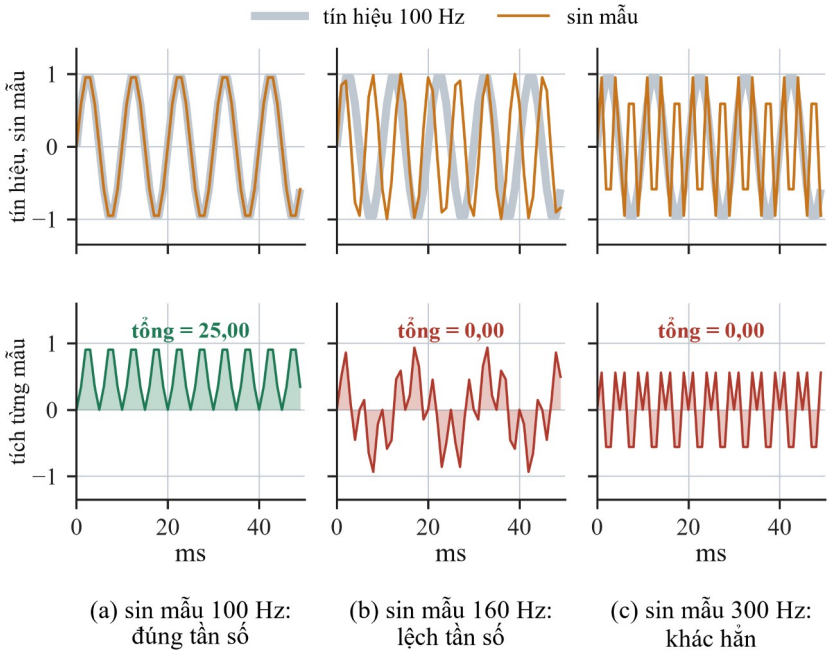
Điều Fourier phát biểu là chiều ngược lại, và nó mạnh hơn nhiều: *Mọi* tín hiệu đều tách được thành một tổng các sóng sin, và cách tách ấy là *duy nhất*. Nói cách khác, danh sách "gồm những sóng sin nào, mỗi cái biên độ bao nhiêu và lệch pha bao nhiêu" là một cách mô tả tín hiệu đầy đủ không kém gì dãy mẫu.

4.2.2 DFT làm gì — giải thích không cần số phức

Công thức của biến đổi Fourier rời rạc (DFT, tiếng Anh: Discrete Fourier Transform) trông đáng sợ vì có số phức và ký hiệu tổng. Nhưng ý tưởng đằng sau nó chỉ là một phép *đo độ giống*, và phép đo ấy hiểu được không cần số phức.

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j2\pi kn/N}, \qquad k = 0, 1, \ldots, N-1 \tag{4.1}$$

Trong công thức này, x[n] là dãy mẫu đầu vào, N là số mẫu trong khung đang xét, và X[k] là kết quả ứng với vạch tần số thứ k. Thừa số mũ phức chính là một cặp sóng sin và cosin tần số k. Phép toán thật sự đang diễn ra là: nhân tín hiệu với một sóng sin mẫu, rồi cộng tất cả các tích lại.



Hình 4.5 Ý tưởng của DFT: nhân với một sóng sin mẫu rồi cộng — tổng lớn nghĩa là giống

Hàng dưới của hình trên là toàn bộ chuyện. Khi tần số của sóng sin mẫu trùng với tần số có trong tín hiệu, tích luôn dương ở những chỗ hai bên cùng dấu nên tổng lớn. Khi hai tần số không trùng, phần dương và phần âm của tích triệt tiêu nhau gần hết nên tổng gần bằng không. Làm phép đo ấy với mọi tần số mẫu, chúng ta có cả cái phổ.

Vì sao phải có số phức

Nếu chỉ nhân với sóng sin thì có một lỗ hổng: một thành phần lệch pha đúng 90 độ so với sóng sin mẫu sẽ cho tổng bằng không, và chúng ta kết luận nhầm rằng nó không tồn tại.

Cách chữa là đo hai lần — một lần với sin, một lần với cosin — rồi ghép hai kết quả lại. Số phức chỉ là cách viết gọn của phép ghép ấy: phần thực giữ kết quả đo bằng cosin, phần ảo giữ kết quả đo bằng sin.

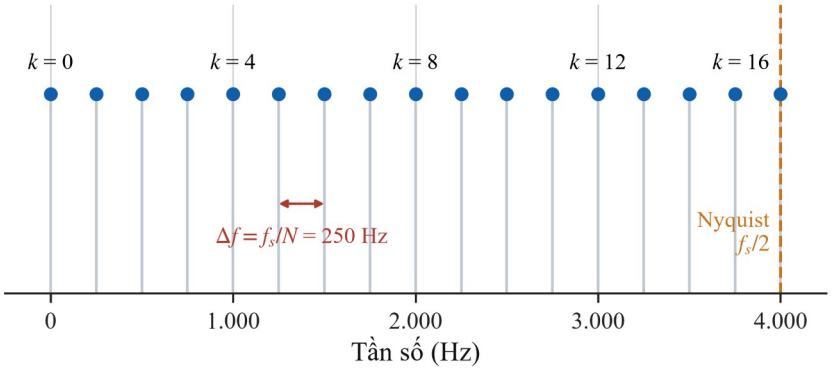
Vậy nên số phức ở đây không phải một khái niệm toán học sâu xa mà là một cách ghi chép. Ai thấy khó chịu với nó có thể nghĩ X[k] đơn giản là một cặp hai con số.

4.2.3 Trục tần số: FFT trả về những tần số nào

Đây là chỗ hay bị bỏ qua nhất, và cũng là chỗ sinh ra nhiều hiểu nhầm nhất. Kết quả của một phép FFT *không* phải một đường cong liên tục theo tần số. Nó là đúng N/2 + 1 con số, tại N/2 + 1 tần số cố định cách đều nhau.

$$f_k = k \cdot \frac{f_s}{N}; \qquad \Delta f = \frac{f_s}{N}; \qquad \Delta t = \frac{N}{f_s} \tag{4.2}$$

Trong đó f_s là tần số lấy mẫu, N là số mẫu trong khung, f_k là tần số ứng với vạch thứ k. Đại lượng Δf gọi là độ phân giải tần số: khoảng cách giữa hai vạch liền nhau. Đại lượng Δt là khoảng thời gian mà khung ấy bao phủ.



Hình 4.6 Với N = 32 mẫu ở 8.000 Hz, FFT cho 17 vạch cách đều 250 Hz

Bảng 4.1 Độ phân giải tần số theo độ dài khung, ở 44.100 Hz

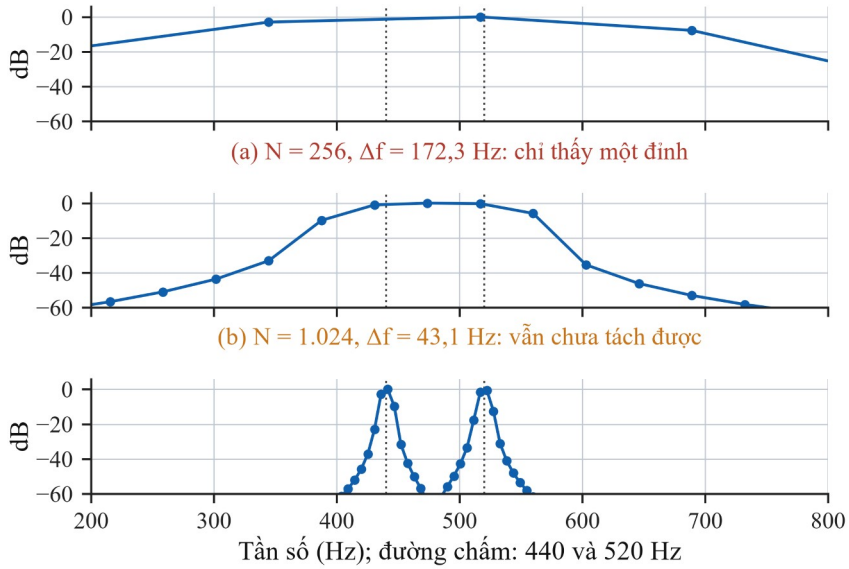
N (mẫu)	Δf (Hz)	Δt (ms)	Ghi chú thực hành
128	344,5	2,9	quá thô cho tần số, tốt cho tiếng gõ
512	86,1	11,6	dùng cho tiếng nói
1.024	43,1	23,2	giá trị mặc định phổ biến nhất
2.048	21,5	46,4	dùng cho nhạc, phân biệt được nốt
8.192	5,4	185,8	phân tích tĩnh, đo chính xác
16.384	2,7	371,5	chỉ dùng khi tín hiệu thật sự đứng yên

Một câu hay bị hiểu sai

Có người nói "FFT với N = 1.024 thì phân biệt được tới 43 Hz". Câu ấy đúng một nửa và nguy hiểm ở nửa còn lại.

Đúng: khoảng cách giữa hai vạch là 43 Hz.

Sai: điều đó *không* có nghĩa là hai tần số cách nhau 43 Hz thì tách được thành hai đỉnh. Muốn tách thật thì còn phải tính tới bề rộng búp chính của cửa sổ — với cửa sổ Hann là bốn vạch, tức 172 Hz. Mục 4.4 sẽ quay lại chuyện này.



Hình 4.7 Hai tần số 440 và 520 Hz với ba độ dài khung: 256, 1.024 và 8.192 mẫu

4.2.4 Biên độ và pha — hai nửa của kết quả

Mỗi X[k] là một cặp số. Từ cặp ấy lấy ra được hai đại lượng có ý nghĩa vật lý rõ ràng.

$$|X[k]| = \sqrt{\mathrm{Re}^2 + \mathrm{Im}^2}\,; \qquad \varphi[k] = \mathrm{atan2}\,(\mathrm{Im},\ \mathrm{Re}) \tag{4.3}$$

- **Biên độ** — thành phần tần số ấy mạnh bao nhiêu. Đây là thứ mọi biểu đồ phổ vẽ ra.
- **Pha** — thành phần tần số ấy bắt đầu ở đâu trong chu kỳ của nó. Đây là thứ gần như mọi biểu đồ phổ bỏ đi.

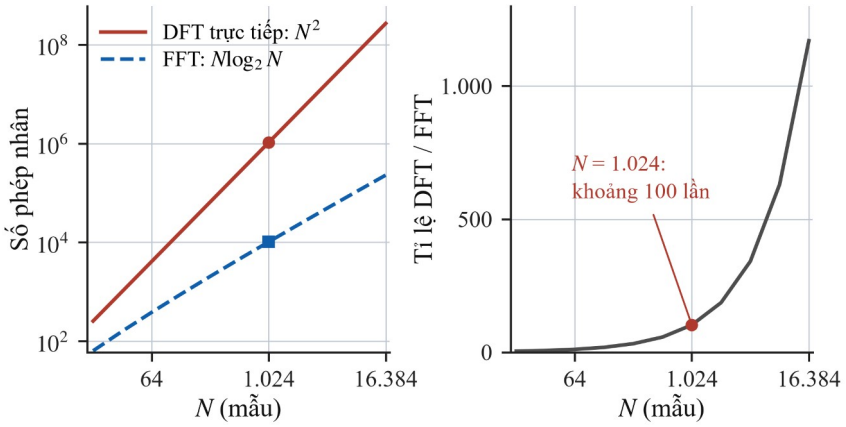
Việc bỏ pha đi là một quyết định có lý do — mắt không đọc được một biểu đồ pha, và tai người rất ít nhạy với pha ở âm ổn định. Nhưng nó là một quyết định, không phải một sự thật, và mục 4.3.4 sẽ cho thấy khi nào cái giá của nó phải trả.

4.2.5 Vì sao gọi là FFT chứ không phải DFT

DFT (Discrete Fourier Transform) là định nghĩa toán học của biến đổi Fourier rời rạc. FFT (Fast Fourier Transform, biến đổi Fourier nhanh) là một thuật toán tính ra đúng cái DFT ấy, nhưng nhanh hơn rất nhiều. Hai chữ này hay bị dùng lẫn, và trong thực hành thì lẫn cũng không sao — nhưng nên biết chúng khác nhau ở chỗ nào.

Tính DFT theo đúng định nghĩa cần N phép nhân cho mỗi vạch, và có N vạch, tức khoảng N^2 phép nhân. Thuật toán FFT — do Cooley và Tukey công bố năm 1965, dù ý tưởng đã có từ Gauss — nhận ra rằng rất nhiều phép nhân trong đó lặp lại nhau, và tổ chức lại phép tính để chỉ còn khoảng $N \cdot \log_2 N$.

$$C_{\mathrm{DFT}} \sim N^2\,; \qquad C_{\mathrm{FFT}} \sim N \log_2 N\,; \qquad \frac{C_{\mathrm{DFT}}}{C_{\mathrm{FFT}}} = \frac{N}{\log_2 N} \tag{4.4}$$



(a) Số phép nhân (b) Tỉ lệ giữa hai cách

Hình 4.8 Chi phí tính của DFT trực tiếp và của FFT, và tỉ lệ giữa hai cách

Với khung 1.024 mẫu — cỡ hay dùng nhất — chênh lệch là khoảng một trăm lần. Đó chính là khoảng cách giữa "phân tích được sau khi chờ" và "phân tích được ngay trong lúc thu". Toàn bộ ngành xử lý âm thanh thời gian thực, từ bộ chỉnh âm trong trạm làm việc âm thanh số (DAW, tiếng Anh: Digital Audio Workstation) tới bộ khử tiếng vọng trong điện thoại, dựng trên chênh lệch ấy.

Vì sao N nên là luỹ thừa của 2

Thuật toán FFT nhanh nhất khi N chia đôi được nhiều lần liên tiếp. Với N là luỹ thừa của 2 thì nó chia được tới tận cùng.

Thư viện numpy vẫn chạy được với N bất kỳ, nhưng chậm hơn, và với một số nguyên tố lớn thì chậm hơn rất nhiều.

Vì thế quy ước thực hành là: luôn chọn N trong dãy 256, 512, 1.024, 2.048, 4.096, 8.192, 16.384. Nếu đoạn tín hiệu không đủ dài thì đệm thêm số không cho đủ — thao tác gọi là zero-padding.

4.2.6 Đệm số không: cái nó làm và cái nó không làm

Vì quy ước trên, thao tác đệm số không xuất hiện thường xuyên tới mức nó sinh ra một hiểu nhầm rất phổ biến, và nên chặn ngay ở đây.

Đệm thêm số không vào cuối khung làm tăng N, mà Δf bằng f_s chia N, nên nhìn công thức (4.2) thì tưởng như độ phân giải tăng theo. Biểu đồ sau khi đệm cũng trông mượt hơn hẳn, có nhiều điểm hơn, các đỉnh trông tròn trịa hơn. Cả hai dấu hiệu ấy đều gợi ý rằng chúng ta vừa được thêm thông tin.

Nhưng không có thông tin nào được thêm cả. Số không không mang tin. Điều đệm số không thật sự làm là *nội suy* — nó tính thêm các điểm nằm giữa những vạch vốn có, dùng đúng thông tin đã có sẵn. Kết quả là một đường cong mịn hơn đi qua đúng những giá trị cũ, chứ không phải một phép đo chi tiết hơn.

Bảng 4.2 Đệm số không: làm gì và không làm gì

Điều người ta hy vọng	Điều thật sự xảy ra
Tách được hai đỉnh vốn dính vào nhau	*Không*. Hai đỉnh vẫn dính; chỉ là phần gồ lên dính nhau ấy được vẽ mịn hơn
Đo được chính xác hơn tần số của một đỉnh	*Có*, phần nào — đỉnh của đường cong nội suy gần tần số thật hơn tâm vạch gần nhất
Nhìn rõ hình dạng của một đỉnh	*Có* — đây là lý do chính đáng để đệm
Làm N thành luỹ thừa của 2 cho FFT chạy nhanh	*Có* — và đây là lý do phổ biến nhất trong thực tế

Một câu để phân biệt

Độ phân giải thật do *độ dài thực* của đoạn tín hiệu quyết định, không phải do số điểm trong mảng đưa vào FFT.

Muốn nhìn mịn hơn thì đệm số không. Muốn *phân biệt* được nhiều hơn thì phải thu dài hơn. Hai việc ấy không thay thế nhau được.

4.2.7 Khung mã tối thiểu

Toàn bộ mục này quy về chín dòng mã. Người đọc nên gõ lại chúng một lần bằng tay thay vì sao chép, vì bốn chỗ hay sai đều nằm ở những chi tiết nhỏ.

```
import numpy as np

x = doc_wav(...)                    # dãy mẫu, tần số lấy mẫu sr
n = 4096                            # độ dài khung (luỹ thừa của 2)
w = np.hanning(n)                   # cửa sổ, xem mục 4.4

X = np.fft.rfft(x[:n] * w)          # chỉ lấy nửa phổ dương
f = np.fft.rfftfreq(n, 1 / sr)      # trục tần số (Hz)

mag = np.abs(X)                     # biên độ
pha = np.angle(X)                   # pha (radian)
dB = 20 * np.log10(mag / mag.max()) # chuẩn hoá về đỉnh
```

Hình 4.9 Khung mã tối thiểu để có một biểu đồ phổ đúng đơn vị

Liệt kê 4.1 Tính phổ của một đoạn tín hiệu và in ra ba tần số mạnh nhất

```
import numpy as np

def pho_dinh(x, sr, n=4096, so_dinh=3):
    """Tra ve so_dinh tan so manh nhat trong khung dau tien."""
    n = min(n, len(x))
    w = np.hanning(n)
    X = np.abs(np.fft.rfft(x[:n] * w))
    f = np.fft.rfftfreq(n, 1 / sr)
    idx = np.argsort(X)[::-1][:so_dinh]
    return [(float(f[i]), float(X[i] / X.max())) for i in idx]
```

```
# vi du: sinh mot tin hieu ba hai roi tim lai chung
sr = 44100
t  = np.arange(sr) / sr
x  = (np.sin(2*np.pi*440*t) + 0.5*np.sin(2*np.pi*880*t)
      + 0.25*np.sin(2*np.pi*1320*t))

for f, muc in pho_dinh(x, sr):
    print(f"{f:8.1f} Hz   {20*np.log10(muc):6.1f} dB")
```

Kết quả in ra ba dòng gần đúng với 440 Hz ở 0 dB, 880 Hz ở khoảng −6 dB và 1.320 Hz ở khoảng −12 dB. Ba con số ấy đúng bằng tỉ lệ biên độ đã đưa vào, và đó là cách kiểm nhanh nhất rằng đoạn mã chạy đúng: sinh một tín hiệu đã biết trước rồi xem có tìm lại được nó không. Liệt kê 4.1 nên được chạy trước mọi bài thực hành sau của chương.

Bốn chỗ hay sai khi viết đoạn mã này

Bỏ sót việc nhân cửa sổ. Kết quả vẫn ra một cái phổ trông hợp lý, nhưng sai — xem mục 4.4.

Dùng np.fft.fft thay vì rfft (real FFT, phép FFT dành cho tín hiệu thực). Với tín hiệu thực, nửa trên của kết quả là ảnh gương của nửa dưới; rfft bỏ luôn phần thừa ấy.

Bỏ sót np.fft.rfftfreq. Khi ấy trục hoành là chỉ số k chứ không phải hertz, và mọi con số đọc ra đều vô nghĩa.

Chuẩn hoá về đỉnh mà không nói ra. Biểu đồ khi ấy luôn có đỉnh ở 0 dB dù tín hiệu to hay nhỏ — tiện để nhìn hình dạng, nhưng không so sánh được hai biểu đồ với nhau.

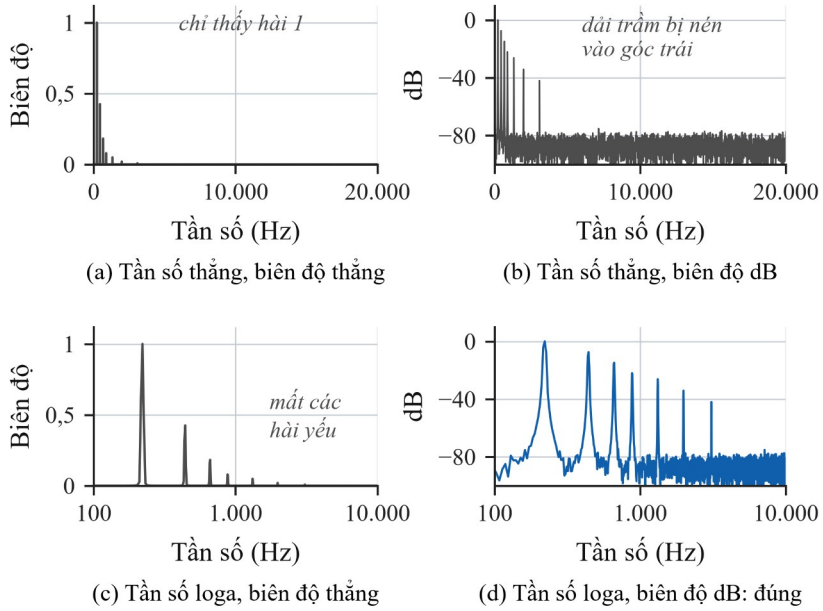
4.3 Đọc một biểu đồ phổ

4.3.1 Bốn câu hỏi phải trả lời trước khi đọc bất cứ biểu đồ phổ nào

Biểu đồ phổ là loại hình dễ bị đọc sai nhất trong cả ngành, vì hai biểu đồ nhìn khác hẳn nhau có thể cùng mô tả một tín hiệu, và hai biểu đồ nhìn giống hệt nhau có thể mô tả hai tín hiệu chẳng liên quan gì. Trước khi kết luận bất cứ điều gì từ một biểu đồ phổ, phải trả lời được bốn câu sau.

Bảng 4.3 Bốn câu hỏi bắt buộc trước khi đọc một biểu đồ phổ

Câu hỏi	Vì sao nó đổi cách đọc
Trục hoành thẳng hay loga?	Trục thẳng dồn cả dải trầm vào một góc nhỏ bên trái; trục loga dàn đều theo bội, đúng với cách tai nghe
Trục tung tuyến tính hay decibel?	Trục tuyến tính giấu mất mọi thứ dưới đỉnh khoảng hai mươi decibel; trục dB bày ra hết
Đã chuẩn hoá về đỉnh chưa?	Nếu có thì đỉnh luôn ở 0 dB dù tín hiệu to hay nhỏ — nhìn hình dạng thì được, so hai biểu đồ với nhau thì không
Khung dài bao nhiêu mẫu?	Quyết định độ phân giải tần số, tức những gì biểu đồ ấy *có thể* phân biệt được



(a) Tần số thẳng, biên độ thẳng (b) Tần số thẳng, biên độ dB

(c) Tần số loga, biên độ thẳng (d) Tần số loga, biên độ dB: đúng

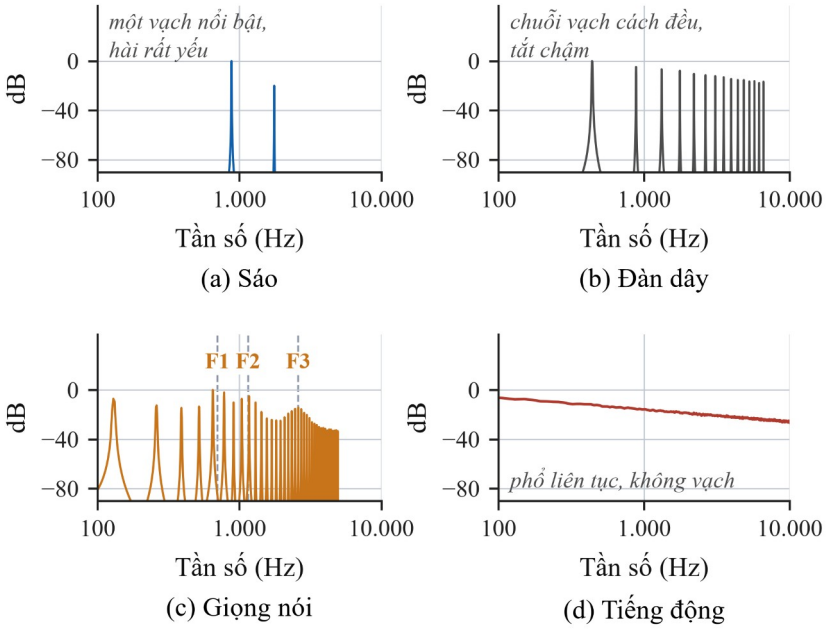
Hình 4.10 Cùng một tín hiệu, bốn cách đặt trục — và chỉ một cách đọc được

$$L[k] = 20\log_{10}\frac{|X[k]|}{X_{\text{ref}}}\ \text{[dB]}, \qquad X_{\text{ref}} = \max_k |X[k]| \tag{4.5}$$

Ô dưới bên phải của hình trên là quy ước của mọi phần mềm chuyên nghiệp, và lý do đã học ở chương 1: tai nghe tần số theo bội chứ không theo hiệu, và nghe cường độ theo loga chứ không theo tỉ lệ thẳng. Đặt trục theo cách ấy tức là vẽ cái mà tai nghe thấy, thay vì vẽ cái mà phép toán trả về.

4.3.2 Vân tay phổ — nhận nguồn âm bằng mắt

Đây là kỹ năng đáng luyện nhất của cả chương, và cũng là kỹ năng chuyển thẳng sang việc làm nghề: nhìn một biểu đồ phổ mà đoán được nó là cái gì. Bốn nguồn âm điển hình có bốn dấu vết rất khác nhau.



(a) Sáo (b) Đàn dây

(c) Giọng nói (d) Tiếng động

Hình 4.11 Bốn nguồn âm điển hình và bốn dạng phổ đặc trưng của chúng

Bảng 4.4 Bốn dấu hiệu cần nhận ra trên một biểu đồ phổ

Dấu hiệu nhìn thấy	Kết luận về nguồn
Có các vạch nhọn, cách đều nhau	Nguồn tuần hoàn có cao độ — nhạc cụ, giọng hát, tiếng máy quay đều tốc độ
Chỉ một vạch nổi bật, các vạch sau rất yếu	Âm gần thuần — sáo, âm chuẩn của bộ đo, sóng sin do máy sinh
Chuỗi vạch cách đều nhưng có vài vùng năng lượng gồ lên nằm ngang	Giọng nói hoặc nhạc cụ có khoang cộng hưởng; các vùng gồ lên ấy là formant
Phổ liên tục, không có vạch nào	Tiếng động, gió, nhiễu nền, tiếng bước chân — không có cao độ

Khoảng cách giữa các vạch cho biết tần số cơ bản, tức cao độ. Tốc độ tắt dần của chuỗi vạch cho biết âm sắc: tắt nhanh thì nghe tối và mượt, tắt chậm thì nghe sáng và gắt. Đây chính là phần phổ hài đã học ở chương 1, giờ nhìn được bằng công cụ thay vì chỉ nghe.

4.3.3 Sàn nhiễu — phần dưới của biểu đồ cũng có nghĩa

Người mới đọc phổ thường chỉ nhìn các đỉnh. Nhưng phần *thấp* của biểu đồ — cái mặt gồ ghề nằm dưới cùng — mang thông tin quan trọng không kém, và ở nhiều bài toán thực tế thì nó mới là thứ cần đo.

- Sàn nhiễu cao hay thấp so với đỉnh cho biết tỉ số tín hiệu trên nhiễu — chính là đại lượng đã học ở chương 2, giờ đo được bằng mắt.
- Sàn nhiễu phẳng đều theo tần số là nhiễu trắng, thường của mạch điện. Sàn nghiêng xuống về phía cao là nhiễu hồng hoặc tiếng ồn môi trường.
- Sàn nhiễu có vạch nhô lên ở 50 Hz và các bội của nó là nhiễu ù lưới điện — vấn đề của phần cứng, không sửa được bằng phần mềm một cách sạch sẽ.

Một bẫy thường gặp khi đọc sàn nhiễu

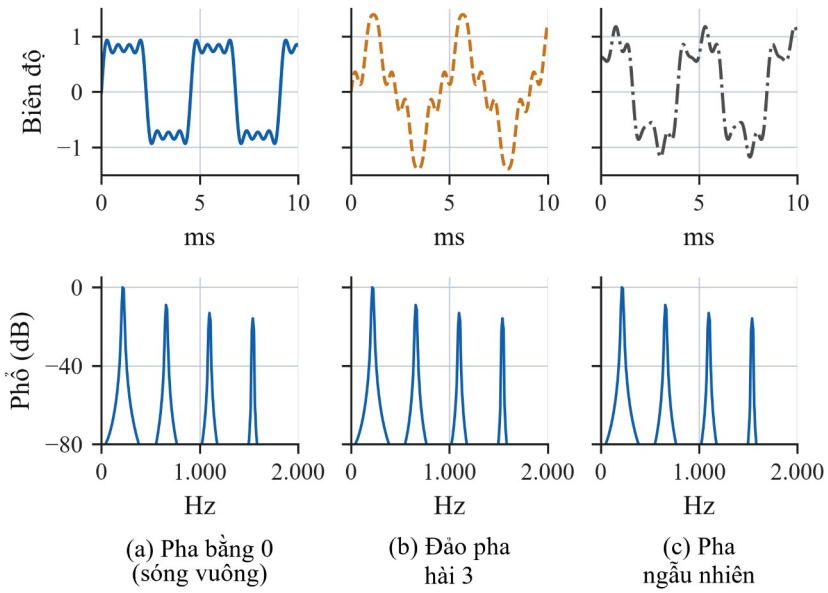
Không phải mọi thứ ở đáy biểu đồ đều là nhiễu thật của tín hiệu.

Nếu bỏ sót việc nhân cửa sổ, phần rò phổ của chính đỉnh mạnh sẽ nằm rải khắp đáy biểu đồ và trông giống hệt một sàn nhiễu. Nhiều người đã kết luận nhầm rằng thiết bị của họ bị nhiễu, trong khi lỗi nằm ở đoạn mã phân tích.

Cách phân biệt: đổi độ dài khung. Sàn nhiễu thật gần như không đổi hình dạng; sàn do rò phổ thì đổi theo.

4.3.4 Pha — nửa bị bỏ đi

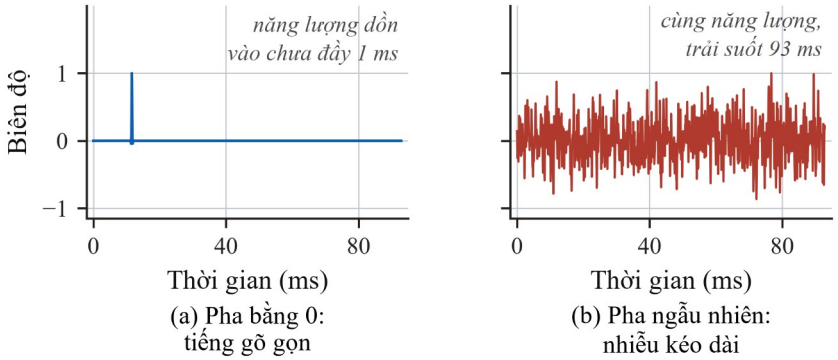
Như đã nói ở mục 4.2.4, mỗi X[k] có hai phần và biểu đồ phổ chỉ vẽ một. Câu hỏi tự nhiên là: bỏ đi như thế có mất gì không? Câu trả lời là *có*, và mất rất nhiều — chỉ là tai người phần lớn thời gian không nhận ra.



Hình 4.12 Ba tín hiệu có phổ biên độ giống hệt nhau nhưng dạng sóng khác hẳn

Ba dạng sóng ở hàng trên khác nhau rõ ràng, nhưng ba biểu đồ phổ ở hàng dưới giống nhau tới từng vạch. Toàn bộ khác biệt nằm ở pha. Với tai người nghe ba tín hiệu này ở mức ổn định, chúng gần như không phân biệt được — đó là lý do việc bỏ pha thường không gây hậu quả.

Nhưng "gần như" không phải "hoàn toàn", và có một loại tín hiệu mà pha quyết định tất cả: âm chuyển tiếp, tức tiếng gõ, tiếng đập, phụ âm bật.



Hình 4.13 Hai tín hiệu cùng phổ biên độ: pha bằng không cho tiếng gõ, pha ngẫu nhiên cho tiếng nhiễu

Điều cần nhớ về pha

Pha không nghe được ở âm ổn định — một nốt đàn giữ lâu, một tiếng ù, một sóng sin.

Pha quyết định *hoàn toàn* ở âm chuyển tiếp. Cùng một phổ biên độ, pha đúng cho một tiếng trống sắc gọn, pha sai cho một tiếng nhiễu nhoè kéo dài.

Đó là lý do mọi thuật toán co giãn thời gian và dịch cao độ đều gặp khó khăn với tiếng trống, và vì sao chất lượng của chúng thường được đánh giá bằng đúng loại tín hiệu ấy. Chương 10 sẽ quay lại chuyện này khi bàn về hiệu ứng.

4.3.5 Con số trên trục tung thật ra nghĩa là gì

Còn một chi tiết nữa hay bị bỏ qua, và nó khiến nhiều người kết luận sai về *mức*. Giá trị |X[k]| mà FFT trả về không phải biên độ của thành phần tần số ấy. Nó phụ thuộc vào ba thứ: biên độ thật, số mẫu N trong khung, và cửa sổ đã dùng.

Cụ thể, cùng một sóng sin biên độ 1, nếu đưa vào FFT với N = 1.024 thì |X[k]| ở đỉnh lớn gấp bốn lần so với khi N = 256. Và nếu nhân cửa sổ Hann thì con số ấy lại nhỏ đi khoảng một nửa, vì cửa sổ đã kéo phần lớn khung xuống dưới 1.

Bảng 4.5 Ba cách trình bày trục tung và ý nghĩa của chúng

Cách trình bày	Cách tính	Đọc được gì
Chuẩn hoá về đỉnh	chia cho max của chính nó	Chỉ đọc được *hình dạng* phổ. Không so được hai biểu đồ với nhau. Đây là cách dùng trong hầu hết hình của bài này
Biên độ thật	nhân 2, chia N, chia hệ số cửa sổ	Đọc được biên độ của từng thành phần đúng như trong tín hiệu
dBFS	từ biên độ thật, áp công thức (3.7) của chương 3	So được với mức toàn thang — dùng khi cần biết có bị cắt đỉnh hay không

Hệ quả thực hành

Nếu chỉ cần nhìn hình dạng phổ — tìm hài, nhận nguồn, so tương quan giữa các đỉnh — thì chuẩn hoá về đỉnh là đủ, và đó là mặc định của gần hết công cụ.

Nếu cần một *con số* để báo cáo hoặc để so với một phép đo khác thì phải chuẩn hoá đầy đủ. Bỏ qua bước này là lý do phổ biến nhất khiến hai người đo cùng một tín hiệu mà ra hai con số lệch nhau cả chục decibel.

Cách tự kiểm nhanh: sinh một sóng sin biên độ đúng 1, chạy đoạn mã của mình, và xem nó có trả về đúng 1 (hoặc 0 dBFS) không. Nếu không thì hệ số chuẩn hoá đang sai.

4.3.6 Bốn bước đọc một biểu đồ phổ lạ

Gộp cả mục 4.3 lại thành một quy trình. Khi gặp một biểu đồ phổ chưa biết xuất xứ — trong một bài báo, một tài liệu kỹ thuật, hay do chính đồng nghiệp gửi tới — làm theo bốn bước dưới đây, đúng thứ tự.

- **Bước 1 — đọc hai trục.** Thẳng hay loga, dB hay tuyến tính, đã chuẩn hoá chưa, khung dài bao nhiêu. Nếu hình không ghi rõ những điều này thì mọi kết luận rút ra từ nó đều là phỏng đoán.
- **Bước 2 — tìm cấu trúc.** Có vạch hay không? Vạch có cách đều không? Khoảng cách giữa hai vạch liền nhau là bao nhiêu — đó chính là tần số cơ bản.

– **Bước 3 — nhìn sàn.** Sàn nhiễu cách đỉnh bao nhiêu decibel? Sàn phẳng hay nghiêng? Có vạch ù nhô lên không? Và quan trọng nhất: đây là sàn thật hay là rò phổ?
– **Bước 4 — đặt câu hỏi ngược.** Với độ dài khung ấy, biểu đồ này *có thể* phân biệt được gì? Nếu ai đó kết luận hai đỉnh cách nhau 30 Hz là hai nguồn khác nhau, mà khung chỉ dài 512 mẫu, thì kết luận ấy không có cơ sở.

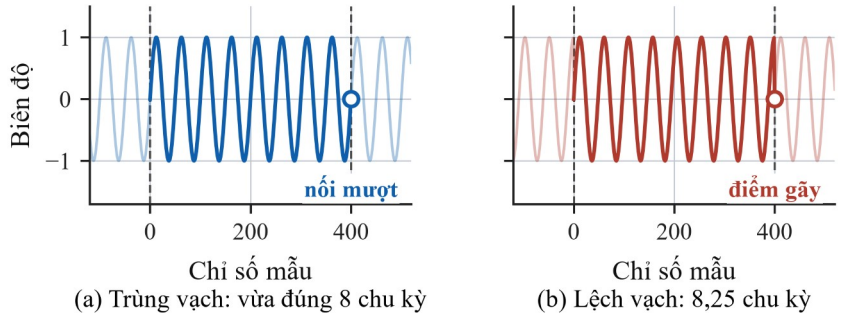
Bước 4 là bước hay bị bỏ nhất và cũng là bước phân biệt người đọc phổ có chuyên môn với người chỉ nhìn hình. Nó cần đúng một phép chia — và mục tiếp theo giải thích vì sao phép chia ấy chưa đủ.

4.4 Cửa sổ và rò phổ

Mục này không có trong hầu hết các giáo trình nhập môn, hoặc chỉ được nhắc bằng một câu. Nhưng nó xứng đáng một mục riêng vì hai lý do: đây là lỗi hay gặp nhất ở người mới bắt đầu phân tích phổ, và nó là loại lỗi *không báo* — chương trình chạy bình thường, biểu đồ vẽ ra có vẻ hợp lý, nhưng kết quả sai.

4.4.1 Nguyên nhân gốc: FFT ngầm nối đầu khung với cuối khung

Phép DFT ở công thức (4.1) chỉ nhìn N mẫu. Nó hoàn toàn không biết tín hiệu còn tiếp diễn ở hai bên khung ấy, và cách nó xử lý điều đó là ngầm giả định rằng khung *lặp lại vô hạn* — nghĩa là mẫu cuối khung nối thẳng vào mẫu đầu khung.



Hình 4.14 Chỗ nối giữa hai bản lặp: mượt khi số chu kỳ chẵn, gãy khi không chẵn

Nếu khung chứa vừa đúng một số nguyên chu kỳ thì chỗ nối liền mượt, và phép biến đổi trả về đúng một vạch sắc nét. Nếu không thì chỗ nối sinh ra một điểm gãy đột ngột — và một điểm gãy đột ngột, theo đúng lý thuyết Fourier, chứa vô số tần số. Những tần số ấy hiện lên biểu đồ dù chúng không hề có trong tín hiệu. Đó là *rò phổ*.

Vì sao gần như luôn luôn bị rò

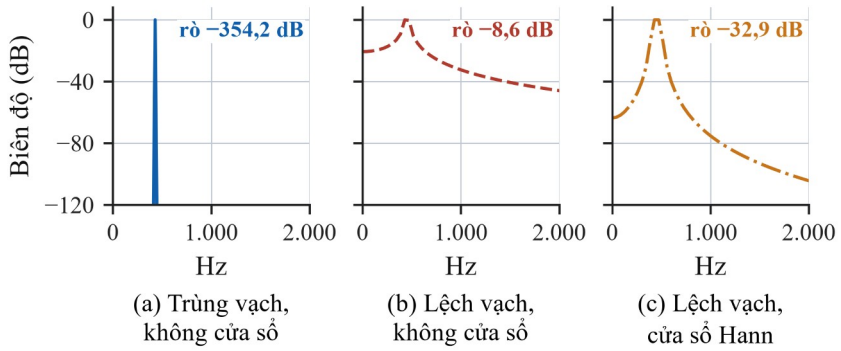
Để khung chứa đúng số nguyên chu kỳ, tần số của tín hiệu phải trùng đúng tâm một vạch, tức phải là bội đúng của f_s / N.

Với tín hiệu thật — một giọng người, một nốt đàn, tiếng máy — điều đó gần như không bao giờ xảy ra. Xác suất trùng đúng là gần bằng không.

Nói cách khác: rò phổ không phải trường hợp ngoại lệ mà là trường hợp *mặc định*. Trường hợp không rò mới là ngoại lệ, và chỉ gặp khi chúng ta tự sinh tín hiệu để thử.

4.4.2 Đo thử mức rò trên tín hiệu thật

Ba ô dưới đây đều là một sóng sin duy nhất, sạch tuyệt đối, không có nhiễu nào. Chỉ khác nhau ở tần số lệch vạch hay không, và ở chỗ có nhân cửa sổ hay không.



Hình 4.15 Cùng một sóng sin sạch: trùng vạch, lệch vạch, và lệch vạch có cửa sổ Hann

Bảng 4.6 Năng lượng rò đo trên một sóng sin sạch, khung 1.024 mẫu

Trường hợp	Cửa sổ	Năng lượng ngoài búp chính
trùng đúng tâm vạch (430,7 Hz)	không	−354,2 dB
lệch nửa vạch (452,2 Hz)	không	−8,6 dB
lệch nửa vạch (452,2 Hz)	Hann	−32,9 dB
lệch nửa vạch (452,2 Hz)	Blackman	−54,8 dB

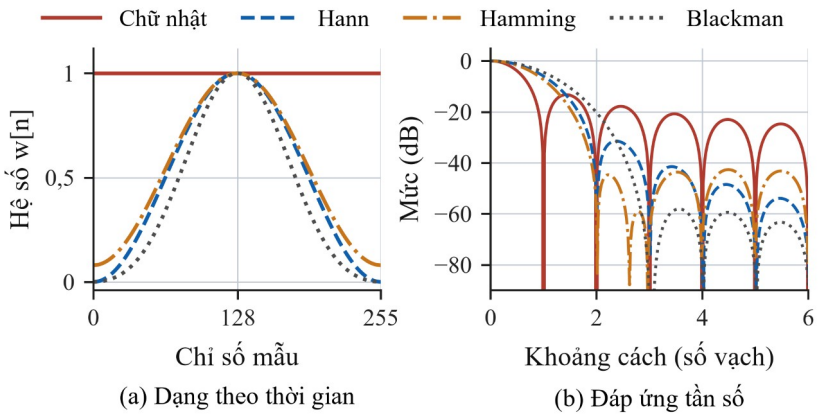
Dòng thứ nhất là trường hợp lý tưởng — năng lượng ngoài búp chính nhỏ tới mức chỉ còn là sai số làm tròn của máy. Dòng thứ hai là điều xảy ra trong thực tế nếu không làm gì cả: chỉ vì tần số lệch nửa vạch, một tín hiệu hoàn hảo hiện lên như một tín hiệu bẩn. Hai dòng cuối cho thấy một phép nhân đơn giản chữa được phần lớn chuyện đó.

4.4.3 Cửa sổ là gì và nó chữa bằng cách nào

Cửa sổ là một dãy hệ số cùng độ dài với khung, nhân từng phần tử vào tín hiệu trước khi đưa vào FFT. Hình dạng của nó là một đường cong dạng chuông: bằng 1 ở giữa và thon dần về 0 ở hai mép.

$$x_{\mathrm{w}}[n] = w[n] \cdot x[n]\,; \qquad w_{\mathrm{hann}}[n] = \frac{1}{2}\left[1 - \cos\frac{2\pi n}{N-1}\right] \tag{4.6}$$

Cách chữa rất trực tiếp: nếu hai mép của khung đều đã bị kéo về không thì chỗ nối giữa hai bản lặp không còn điểm gãy nào nữa, bất kể tần số của tín hiệu là bao nhiêu. Điểm gãy biến mất thì các tần số giả do nó sinh ra cũng biến mất theo.



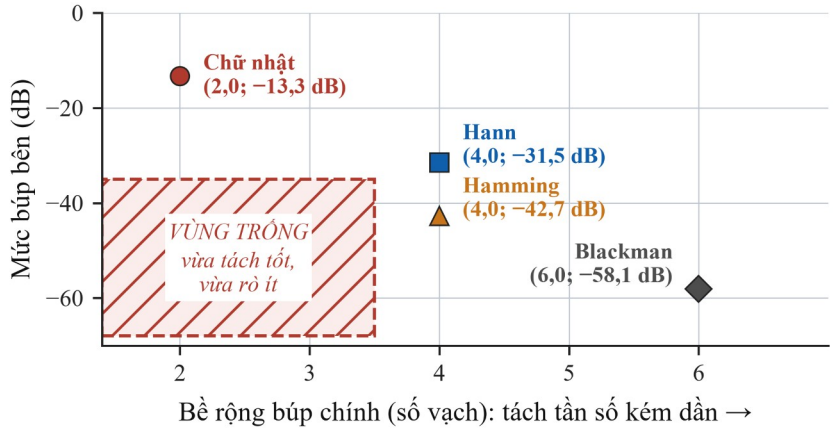
Hình 4.16 Bốn cửa sổ thông dụng: dạng theo thời gian và đáp ứng tần số

Nhưng phép chữa ấy có giá của nó, và giá phải trả nằm ở ô bên phải của hình trên. Mỗi cửa sổ có hai đặc trưng đối nghịch nhau.

– **Bề rộng búp chính** — một sóng sin đơn lẻ, thay vì hiện lên thành một vạch, sẽ hiện lên thành một búp rộng chừng ấy vạch. Búp càng rộng thì càng khó tách hai tần số gần nhau.
– **Mức búp bên** — phần năng lượng rơi ra ngoài búp chính. Búp bên càng thấp thì càng ít rò, tức càng nhìn được các thành phần yếu.

Bảng 4.7 Bốn cửa sổ thông dụng và đặc trưng của chúng

Cửa sổ	Búp chính (số vạch)	Búp bên (dB)	Dùng khi nào
Chữ nhật (không cửa sổ)	2,0	−13,3	chỉ khi biết chắc tín hiệu trùng vạch, hoặc khi phân tích một xung ngắn nằm gọn trong khung
Hann	4,0	−31,5	mặc định — hợp với gần hết mọi việc
Hamming	4,0	−42,7	khi cần nhìn thành phần yếu bên cạnh một đỉnh mạnh
Blackman	6,0	−58,1	khi cần đo chính xác mức của một đỉnh, hoặc soi méo hài rất nhỏ



Hình 4.17 Chọn cửa sổ là chọn một điểm trên đường đánh đổi — góc tốt nhất thì trống

Không có cửa sổ nào tốt nhất

Vùng gạch đỏ trong hình trên — vừa tách tần số tốt vừa rò ít — là vùng *trống*. Không cửa sổ nào nằm ở đó, và sẽ không bao giờ có.

Đây không phải một thiếu sót của người thiết kế mà là một định luật: một hàm càng thon trong miền thời gian thì càng bè trong miền tần số, và ngược lại. Đó là cùng một định luật với luật đánh đổi ở mục 4.5.

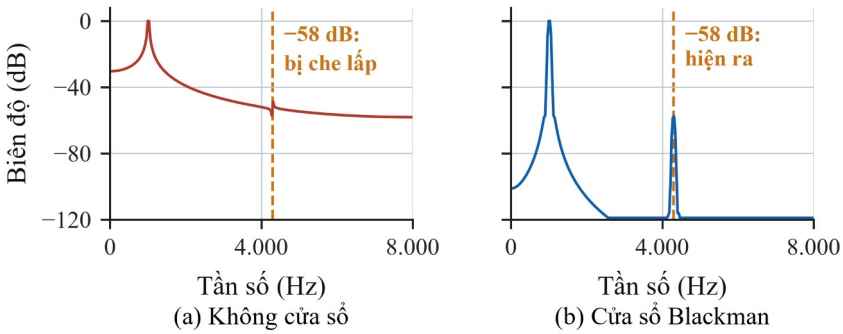
Vì thế câu hỏi đúng không phải "cửa sổ nào tốt nhất" mà "việc này sợ cái gì hơn: dính hai đỉnh, hay che lấp thành phần yếu".

4.4.4 Hệ quả thứ nhất: thành phần yếu bị che lấp

Rò phổ không chỉ làm biểu đồ kém rõ ràng. Nó *che mất* thông tin, và thông tin bị che luôn là loại chúng ta đang cần tìm nhất.

Lý do nằm ở hình học của vấn đề. Phần rò của một đỉnh trải ra hai bên nó và tạo thành một cái sàn giả. Nếu sàn giả ấy nằm cao hơn một thành phần yếu thật thì thành phần ấy chìm hẳn xuống dưới và không còn cách nào nhìn thấy. Với cửa sổ chữ nhật, sàn giả ấy nằm quanh mức −13,3 dB so với đỉnh; với Blackman thì nó xuống tới −58,1 dB — thấp hơn hơn bốn mươi decibel.

Hình dưới đây dựng đúng tình huống ấy: một đỉnh mạnh ở 1 kHz và một thành phần yếu hơn nó 58 decibel. Con số 58 không được chọn tuỳ ý — nó nằm đúng giữa hai mức búp bên vừa nói, nên bị che lấp với cửa sổ này và hiện ra với cửa sổ kia.



Hình 4.18 Một thành phần yếu −58 dB bị sàn rò che lấp, và hiện ra khi đổi cửa sổ

Tình huống ở hình trên là tình huống thật hằng ngày: đo méo hài của một bộ khuếch đại, tìm một tiếng ù nhỏ lẫn trong bản thu, kiểm tra sàn nhiễu của một micro. Trong cả ba việc, cái cần tìm luôn yếu hơn cái đang phát rất nhiều — và đó chính là điều kiện khiến rò phổ trở nên nghiêm trọng.

4.4.5 Hệ quả thứ hai: muốn tách hai tần số thì cần bao nhiêu mẫu

Ở mục 4.2.3 đã cảnh báo rằng độ phân giải Δf không phải khoảng cách tối thiểu tách được. Giờ đã đủ dữ kiện để nói con số đúng.

$$\Delta f_{\text{tách}} \geq B_{\text{w}} \cdot \frac{f_{\text{s}}}{N} \iff N \geq B_{\text{w}} \cdot \frac{f_{\text{s}}}{\Delta f_{\text{tách}}} \tag{4.7}$$

Trong đó B_w là bề rộng búp chính của cửa sổ, tính bằng số vạch — lấy từ bảng ở mục trên. Vế phải là dạng dùng được nhiều hơn: biết cần phân biệt hai tần số cách nhau bao nhiêu, tính ngược ra phải thu dài bao nhiêu mẫu.

Bảng 4.8 Số mẫu cần để tách hai tần số, cửa sổ Hann, ở 44.100 Hz

Khoảng cách cần tách	Số mẫu tối thiểu	Thời lượng thu	Tình huống
500 Hz	353	8 ms	tách hai nhạc cụ ở dải trung
100 Hz	1.764	40 ms	tách hai nốt cách nhau một cung ở quãng tám giữa
20 Hz	8.820	200 ms	tách hài của hai giọng hát cùng lúc
5 Hz	35.280	800 ms	tách hai nốt lệch nhau rất ít — đo độ lạc dây
1 Hz	176.400	4.000 ms	đo trôi tần số của một máy phát

Con số đáng nhớ nhất của mục này

Muốn phân biệt hai tần số cách nhau 5 Hz với cửa sổ Hann, cần 35.280 mẫu — tức phải thu liên tục 0,8 giây.

Con số này thường làm người đọc ngạc nhiên, vì trực giác cho rằng chỉ cần một khoảnh khắc là đủ. Nhưng nó có nghĩa vật lý rất rõ: hai sóng lệch nhau 5 Hz thì phải chờ 0,2 giây chúng mới lệch pha đủ để phân biệt được — và cửa sổ Hann cần bốn lần khoảng ấy.

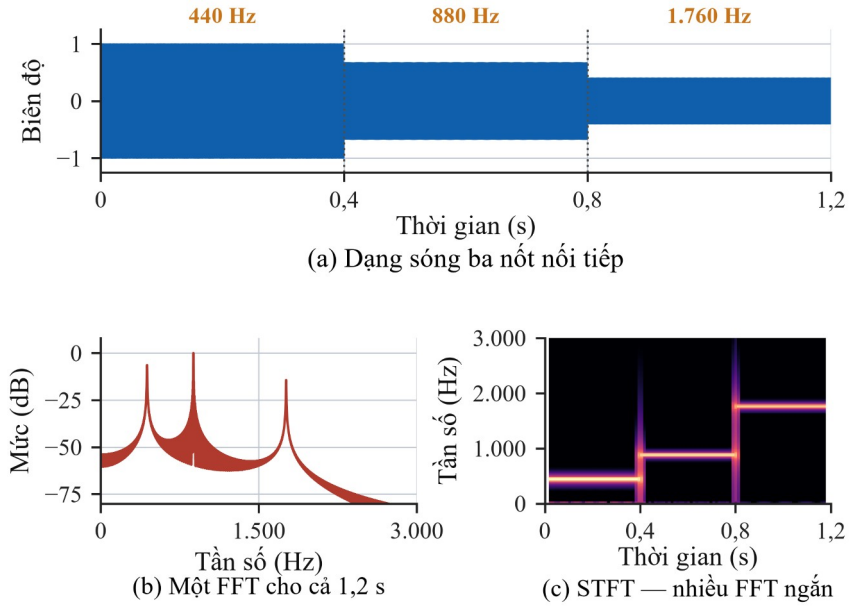
Hệ quả thực hành: nếu bài toán đòi phân giải tần số cao thì nó *bắt buộc* là bài toán trên tín hiệu ổn định. Không có cách nào phân tích rất mịn một thứ thay đổi rất nhanh.

Câu cuối của khung trên chính là cửa vào của mục tiếp theo. Cái ràng buộc vừa nói không phải chuyện riêng của cửa sổ, mà là một định luật chung của mọi phép phân tích tín hiệu.

4.5 Biến đổi Fourier thời gian ngắn và phổ đồ

4.5.1 Vì sao một phép FFT là không đủ

Mọi thứ đã bàn ở ba mục trước đều dựa trên một giả định ngầm: tín hiệu *đứng yên* trong suốt khung đang xét. Với một sóng sin do máy sinh thì giả định ấy đúng. Với âm nhạc, giọng nói hay tiếng động thì nó sai hoàn toàn — thay đổi theo thời gian chính là bản chất của chúng.



Hình 4.19 Một FFT cho cả tín hiệu thấy đủ ba tần số nhưng mất sạch thứ tự thời gian

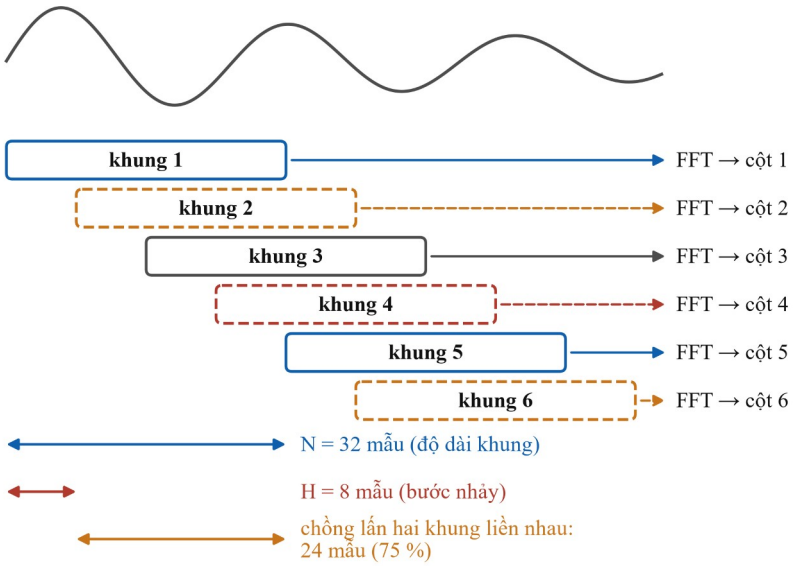
Ô dưới bên trái của hình trên là một biểu đồ phổ hoàn toàn đúng — ba tần số đều có mặt, đúng tỉ lệ. Nhưng nó không nói được rằng chúng đến lần lượt chứ không cùng lúc. Với một bản nhạc dài ba phút, một biểu đồ phổ như thế gần như không dùng được: nó chỉ nói bản nhạc gồm những tần số nào, tức nói rằng nó gồm gần như mọi tần số.

4.5.2 Ý tưởng: cắt nhỏ rồi làm FFT cho từng miếng

Cách chữa đơn giản tới mức đáng ngạc nhiên: thay vì làm một phép FFT cho cả tín hiệu, cắt tín hiệu thành nhiều khung ngắn chồng lấn nhau, làm FFT cho từng khung, rồi xếp các kết quả cạnh nhau theo thứ tự thời gian. Kết quả là một *bảng hai chiều*: hàng là tần số, cột là thời gian, giá trị trong ô là mức. Đó là biến đổi Fourier thời gian ngắn (STFT, tiếng Anh: Short-Time Fourier Transform), và khi vẽ bảng ấy ra bằng màu thì gọi là *phổ đồ*.

$$X[m,k] = \sum_{n=0}^{N-1} w[n]\,x[n+mH]\,e^{-j2\pi kn/N} \tag{4.8}$$

Trong đó m là chỉ số khung, H là bước nhảy giữa hai khung liền nhau, w là cửa sổ. So với công thức (4.1), chỗ khác duy nhất là thêm một chỉ số m và một phép dịch mH — nghĩa là STFT không phải một phép toán mới, chỉ là phép FFT cũ lặp lại nhiều lần ở nhiều vị trí.



Hình 4.20 Cơ chế STFT: khung dài N trượt dọc tín hiệu, mỗi lần nhảy H mẫu

Bảng 4.9 Ba tham số của mọi phổ đồ và ảnh hưởng của chúng

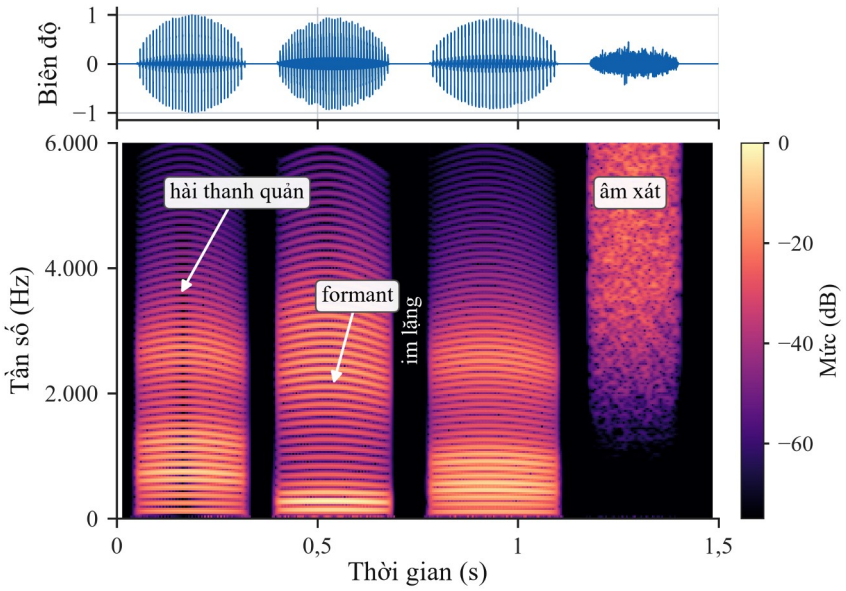
Tham số	Điều khiển cái gì	Chọn thế nào
N — độ dài khung	Đánh đổi giữa độ mịn tần số và độ mịn thời gian	Quyết định quan trọng nhất; xem mục 4.5.4
H — bước nhảy	Phổ đồ mịn hay giật theo trục ngang, và số phép tính	Thường lấy N/4, tức chồng lấn 75 %
w — cửa sổ	Mức rò phổ trong từng khung	Hann, gần như luôn luôn

Vì sao phải chồng lấn: cửa sổ kéo hai mép của mỗi khung về không, nên nếu các khung nối đuôi nhau mà không chồng lấn thì phần tín hiệu rơi vào mép khung sẽ bị làm yếu đi và có thể mất hẳn. Chồng lấn 75 phần trăm — tức H bằng một phần tư N — bảo đảm mọi mẫu đều nằm ở giữa ít nhất một khung.

4.5.3 Đọc một phổ đồ

Phổ đồ có ba trục chứ không phải hai, và trục thứ ba là màu. Đây là điểm hay gây lúng túng nhất khi mới tiếp xúc.

- Trục hoành: thời gian, đọc như dạng sóng.
- Trục tung: tần số, thấp ở dưới cao ở trên.
- Màu: mức năng lượng, thường tối là yếu và sáng là mạnh. Thang màu gần như luôn tính bằng decibel.



(a) Dạng sóng (trên) · (b) Phổ đồ cùng đoạn ấy (dưới)

Hình 4.21 Phổ đồ một đoạn giọng nói: bốn thứ đọc được và ý nghĩa của chúng

Bảng 4.10 Bốn dấu hiệu trên phổ đồ giọng nói

Nhìn thấy	Nghĩa là
Các vạch *ngang* cách đều nhau, chồng lên nhau	Hài của thanh quản. Khoảng cách giữa chúng là tần số cơ bản, tức cao độ giọng
Dải sáng nằm ngang, rộng, ở một vài tần số cố định	Formant — cộng hưởng của khoang miệng. Chúng quyết định đó là nguyên âm nào
Vùng tối, không có gì	Im lặng hoặc phụ âm câm
Mảng sáng ở dải cao, không có vạch	Âm xát kiểu "s", "x", "h" — nhiễu băng rộng, không có cao độ

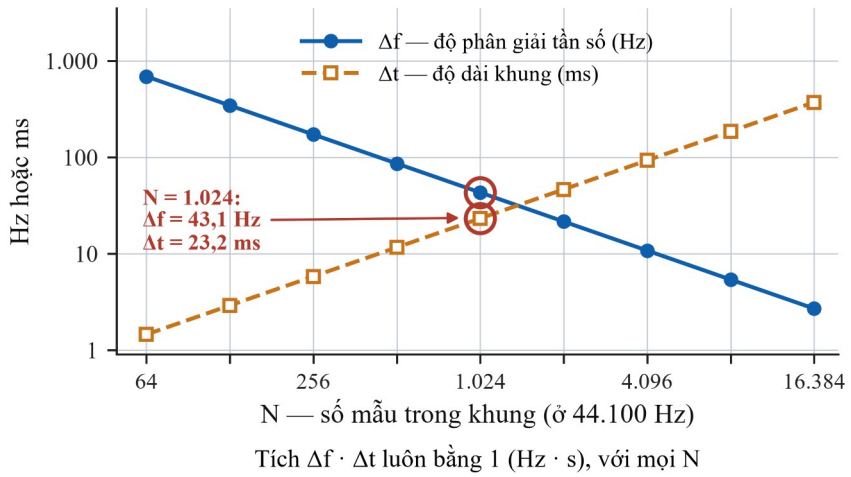
Đọc được bốn dấu hiệu này là đủ để làm phần lớn công việc soát lỗi hằng ngày: tìm chỗ có tiếng ù, xác định đoạn nào bị cắt, kiểm tra xem bộ khử tạp có làm mất phụ âm không, hay nhận ra một bản thu đã bị nén có tổn hao — dấu vết của nó chính là vạch cắt ngang đã học ở chương 3.

4.5.4 Luật đánh đổi thời gian và tần số

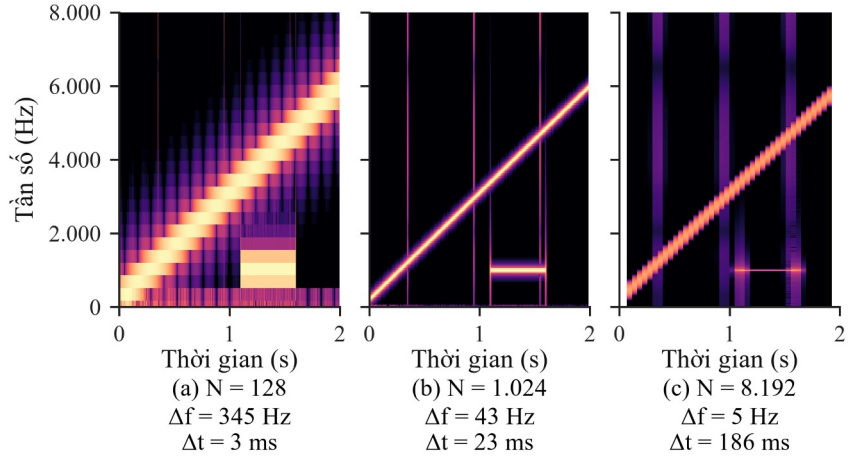
Đây là điều quan trọng nhất của cả mục, và là một trong ba điều quan trọng nhất của cả chương. Nó phát biểu được bằng một dòng.

$$\Delta f \cdot \Delta t = \frac{f_s}{N} \cdot \frac{N}{f_s} = 1 \quad [\text{Hz} \cdot \text{s}] \tag{4.9}$$

Tích của độ phân giải tần số và độ dài khung *luôn* bằng 1. Đó không phải một xấp xỉ hay một quy tắc kinh nghiệm mà là một hệ quả trực tiếp của định nghĩa: Δf bằng f_s chia N, Δt bằng N chia f_s, nhân vào thì N triệt tiêu.



Hình 4.22 Hai đường Δf và Δt luôn đi ngược chiều nhau — không kéo cả hai xuống được



Hình 4.23 Cùng một tín hiệu, ba độ dài khung: 128, 1.024 và 8.192 mẫu

Bảng 4.11 Chọn độ dài khung theo việc cần làm, ở 44.100 Hz

N	Δf	Δt	Việc hợp với nó
128	345 Hz	3 ms	định vị chính xác tiếng gõ, tiếng trống
512	86 Hz	12 ms	phân tích tiếng nói — bắt được phụ âm
1.024	43 Hz	23 ms	mặc định cho việc chung
2.048	22 Hz	46 ms	phân tích nhạc — phân biệt được các nốt
8.192	5 Hz	186 ms	đo tần số chính xác trên tín hiệu ổn định

Đây là một định luật, không phải một hạn chế kỹ thuật

Không có thuật toán nào vượt qua được luật này, kể cả những phương pháp hiện đại nhất. Chúng có thể phân bố sai số khác đi, nhưng tổng thì không đổi.

Đây chính là cùng một định luật với nguyên lý bất định trong vật lý lượng tử, chỉ phát biểu trên tín hiệu thay vì trên hạt. Vị trí và động lượng ở đó, thời gian và tần số ở đây.

Ý nghĩa thực hành: khi ai đó hứa một công cụ "phân tích vừa mịn về tần số vừa chính xác về thời gian", câu hỏi cần đặt ra là nó *đánh đổi* bằng cái gì — chứ không phải nó làm thế nào.

4.5.5 Khung mã cho một phổ đồ

Liệt kê 4.2 Dựng và vẽ phổ đồ bằng scipy

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal

n_fft, hop = 1024, 256          # chong lan 75 %

f, t, Z = signal.stft(x, fs=sr, window="hann",
                      nperseg=n_fft, noverlap=n_fft - hop)

A  = np.abs(Z)                           # bo pha di
dB = 20 * np.log10(np.maximum(A / A.max(), 1e-6))

plt.pcolormesh(t, f, dB, vmin=-80, vmax=0,
               shading="auto", cmap="magma")
plt.ylim(0, 8000)               # chi nhin dai co noi dung
plt.xlabel("Thoi gian (s)"); plt.ylabel("Tan so (Hz)")
plt.colorbar(label="dB")
```

Ba chỗ hay sai khi vẽ phổ đồ

Bỏ sót việc đổi sang decibel. Vẽ thẳng biên độ tuyến tính thì phổ đồ gần như đen kịt, chỉ còn thấy vài đỉnh mạnh nhất.

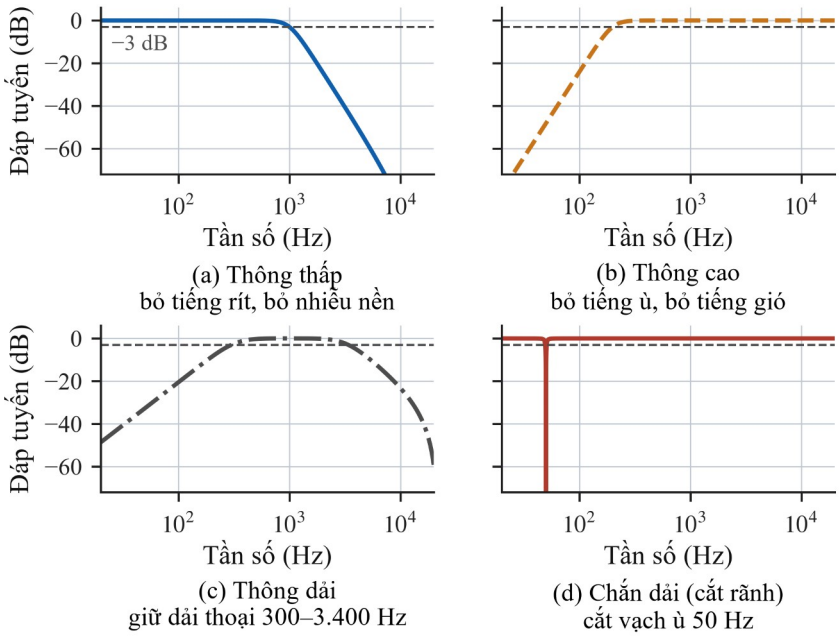
Không đặt vmin. Khi ấy thang màu tự co giãn theo dữ liệu, nên hai phổ đồ của hai tệp khác nhau không so sánh được với nhau.

Để nguyên trục tung tới Nyquist. Với nhạc và tiếng nói thì phần trên 8 kHz thường gần trống, và để nguyên khiến phần có nội dung bị nén vào một dải hẹp ở đáy.

4.6 Bộ lọc số

4.6.1 Bốn kiểu lọc — bảng chữ cái của mọi phép xử lý tần số

Lọc là việc thay đổi mức của các dải tần số khác nhau. Dù phần mềm có phức tạp tới đâu, mọi bộ chỉnh âm, mọi bộ khử tạp, mọi bộ tách dải đều ghép lại từ bốn kiểu cơ bản dưới đây.



Hình 4.24 Đáp tuyến tần số của bốn kiểu lọc số cơ bản, dựng từ bộ lọc thật

Bảng 4.12 Bốn kiểu lọc và việc chúng làm trong phòng thu

Kiểu lọc	Giữ lại	Việc điển hình
Thông thấp	Phần dưới tần số cắt	Bỏ tiếng rít cao tần, bỏ nhiễu nền của mạch, làm mềm một nguồn quá gắt
Thông cao	Phần trên tần số cắt	Bỏ tiếng ù lưới điện, bỏ tiếng gió vào micro, bỏ rung nền của sàn
Thông dải	Một dải ở giữa	Giữ dải thoại 300 – 3.400 Hz, tách một nhạc cụ khỏi bản hoà tấu
Chắn dải hoặc cắt rãnh	Mọi thứ trừ một dải hẹp	Cắt đúng vạch ù 50 Hz mà giữ nguyên phần còn lại; triệt một tiếng cộng hưởng của phòng

Đường đứt ở mức −3 dB trong hình trên là chỗ quy ước gọi là tần số cắt. Cần nhớ rằng đó *không* phải nơi tín hiệu biến mất: ở đúng tần số cắt, tín hiệu còn khoảng bảy mươi phần trăm biên độ, tức vẫn nghe rất rõ. Chỗ tín hiệu thật sự mất nằm xa hơn nhiều, và cách xa bao nhiêu là do *bậc* của bộ lọc quyết định.

4.6.2 Phương trình sai phân — một dòng lặp làm được tất cả

Bốn đáp tuyến trong hình trên trông rất khác nhau, nhưng chúng đều sinh ra từ đúng một dạng phép tính. Toàn bộ lý thuyết bộ lọc số quy về một công thức duy nhất.

$$y[n] = \sum_{i=0}^{P} b_i x[n-i] - \sum_{j=1}^{Q} a_j y[n-j] \tag{4.10}$$

Đọc bằng lời: giá trị đầu ra tại mẫu n bằng một tổ hợp có trọng số của vài mẫu *đầu vào* gần đây, trừ đi một tổ hợp có trọng số của vài mẫu *đầu ra* gần đây. Dãy b gọi là hệ số thuận, dãy a gọi là hệ số hồi tiếp. Chọn hai dãy ấy khác nhau thì được các bộ lọc khác nhau — và đó là toàn bộ nghề thiết kế bộ lọc.

Liệt kê 4.3 Phương trình sai phân viết thẳng thành một vòng lặp

```
def loc(b, a, x):
    """Phuong trinh sai phan (4.10) viet ra tuong minh."""
    y = [0.0] * len(x)
    for n in range(len(x)):
        acc = 0.0
        for i in range(len(b)):            # phan thuan
            if n - i >= 0:
                acc += b[i] * x[n - i]
        for j in range(1, len(a)):         # phan hoi tiep
            if n - j >= 0:
                acc -= a[j] * y[n - j]
        y[n] = acc / a[0]
    return y
```

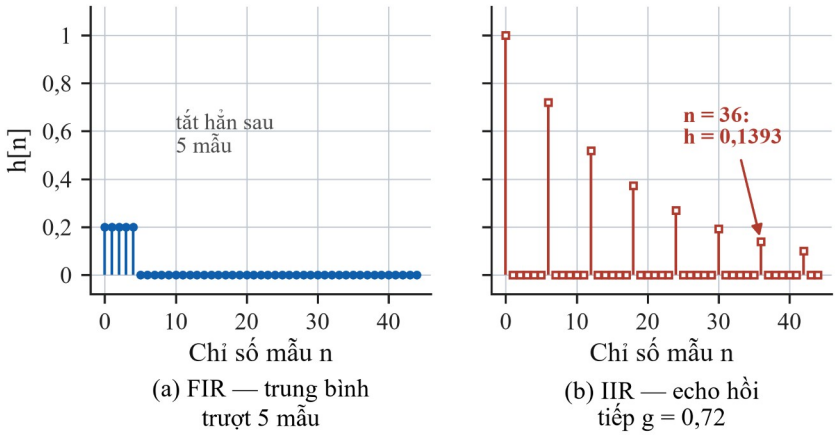
Liệt kê 4.3 chậm hơn hàm thư viện khoảng vài trăm lần và không nên dùng trong sản phẩm. Nhưng nó đáng gõ lại một lần, vì nó cho thấy rằng bên trong scipy.signal.lfilter không có phép màu nào — chỉ có đúng hai vòng lặp lồng nhau. Bản trong mô-đun sigtools.py đã được kiểm và cho kết quả trùng thư viện tới cỡ 10^{-13}.

4.6.3 Hai họ bộ lọc: FIR và IIR

Nhìn vào công thức (4.10), có một câu hỏi tự nhiên: nếu bỏ hẳn phần hồi tiếp đi thì sao? Câu trả lời chia mọi bộ lọc số thành hai họ, và hai họ ấy khác nhau ở gần như mọi mặt.

Nếu dãy a chỉ có một phần tử bằng 1 thì vòng lặp thứ hai trong phương trình sai phân không chạy lần nào: đầu ra chỉ phụ thuộc vào đầu vào. Đó là họ FIR (Finite Impulse Response, đáp ứng xung hữu hạn). Nếu dãy a dài hơn thì đầu ra quay lại làm đầu vào cho chính nó, và đó là họ IIR (Infinite Impulse Response, đáp ứng xung vô hạn).

Cách phân biệt nhanh nhất khi nhìn một bộ lọc trong mã người khác: xem có mảng hệ số thứ hai không. Trong scipy, lệnh firwin trả về một mảng — FIR; lệnh butter trả về hai mảng — IIR. Sự khác biệt một dòng mã ấy kéo theo năm khác biệt trong bảng dưới đây.



Hình 4.25 Đáp ứng xung của một bộ lọc FIR và một bộ lọc IIR

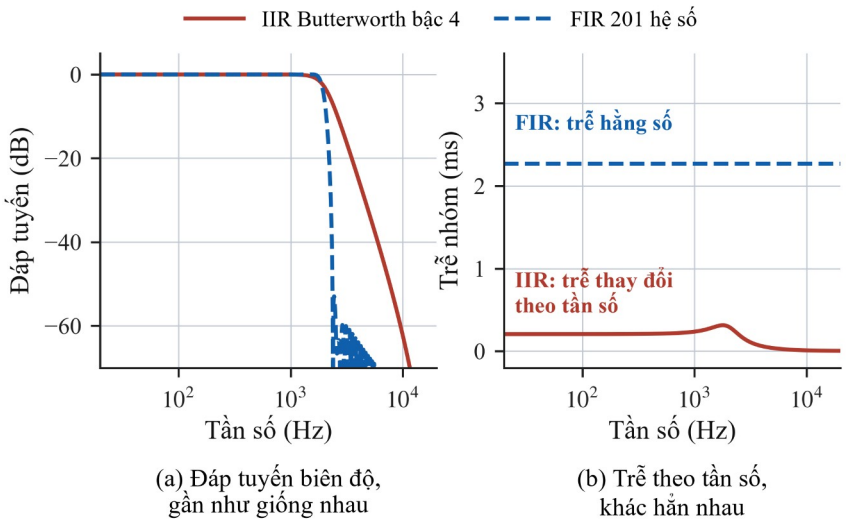
Đáp ứng xung là cách mô tả một bộ lọc đầy đủ nhất: đưa vào đúng một mẫu bằng 1 rồi xem nó cho ra cái gì. FIR cho ra một chuỗi hữu hạn rồi tắt hẳn; IIR cho ra một chuỗi vô hạn nhỏ dần mãi. Đó là toàn bộ nghĩa của hai chữ viết tắt: Finite và Infinite Impulse Response.

Bảng 4.13 FIR và IIR — so sánh theo năm mặt

Mặt	FIR (không hồi tiếp)	IIR (có hồi tiếp)
Đáp ứng xung	Hữu hạn, tắt sau P mẫu	Vô hạn, tắt dần mãi
Pha	Tuyến tính được — mọi tần số trễ bằng nhau	Không tuyến tính — mỗi tần số trễ một kiểu
Ổn định	Luôn ổn định, không cần kiểm	Có thể phân kỳ; phải kiểm trước khi chạy
Chi phí	Cần nhiều hệ số cho một đáp tuyến dốc	Ít hệ số, hiệu quả hơn nhiều
Độ trễ	Bằng nửa số hệ số — có thể đáng kể	Rất nhỏ

4.6.4 Cái giá của IIR: méo pha

Dòng "pha" trong bảng trên là dòng hay bị bỏ qua nhất, và cũng là dòng khiến người ta chọn sai bộ lọc nhiều nhất. Hai bộ lọc có thể có đáp tuyến biên độ gần như giống hệt nhau mà vẫn cho ra hai kết quả nghe khác nhau.



(a) Đáp tuyến biên độ, gần như giống nhau

(b) Trễ theo tần số, khác hẳn nhau

Hình 4.26 Cùng một đáp tuyến biên độ, hai kiểu trễ theo tần số hoàn toàn khác nhau

Ô bên trái của hình cho thấy hai bộ lọc cắt gần như giống nhau. Ô bên phải cho thấy khác biệt thật: bộ FIR đối xứng làm trễ *mọi* tần số đúng bằng nhau, nên tín hiệu ra chỉ là tín hiệu vào bị dời chỗ chứ không đổi hình. Bộ IIR làm trễ mỗi tần số một khác, nên các thành phần bị xô lệch tương đối với nhau và dạng sóng biến dạng.

Chọn FIR hay IIR — ba câu hỏi

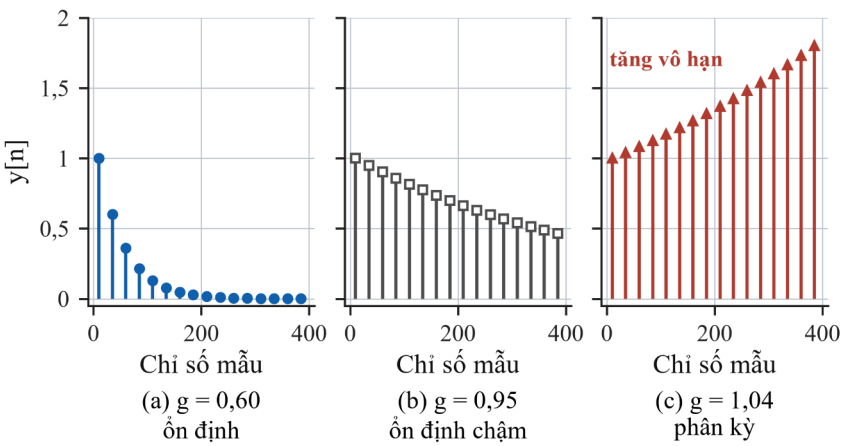
Việc này có sợ méo dạng sóng không? Nếu là xử lý hàng loạt trong hậu kỳ, nhất là với tiếng trống và âm chuyển tiếp, thì chọn FIR.

Việc này có phải chạy thời gian thực với độ trễ rất nhỏ không? Nếu là theo dõi trực tiếp lúc thu, hoặc chạy trên vi điều khiển, thì chọn IIR.

Không chắc? Dùng IIR bậc thấp — đó là mặc định của gần hết phần mềm, và với phần lớn công việc thì méo pha của nó không nghe thấy.

4.6.5 Cái giá của IIR: có thể phân kỳ

Vì đầu ra của bộ lọc IIR quay lại làm đầu vào cho chính nó, có khả năng vòng lặp ấy tự khuếch đại mãi. Khi đó biên độ tăng không giới hạn và tín hiệu ra thành vô nghĩa.



(a) g = 0,60 ổn định

(b) g = 0,95 ổn định chậm

(c) g = 1,04 phân kỳ

Hình 4.27 Cùng một bộ echo hồi tiếp với ba hệ số: ổn định, ổn định chậm, và phân kỳ

Đây không phải chuyện lý thuyết. Đó chính là tiếng hú khi micro hướng vào loa — cùng một vòng hồi tiếp, chỉ khác là nó xảy ra trong không khí thay vì trong mảng số. Và đó cũng là lỗi hay gặp nhất khi tự viết một hiệu ứng vang: đặt hệ số hồi tiếp quá cao để "vang lâu hơn", rồi nghe tiếng loa bị méo nặng.

Liệt kê 4.4 Kiểm điều kiện ổn định trước khi cho bộ lọc chạy

```
import numpy as np

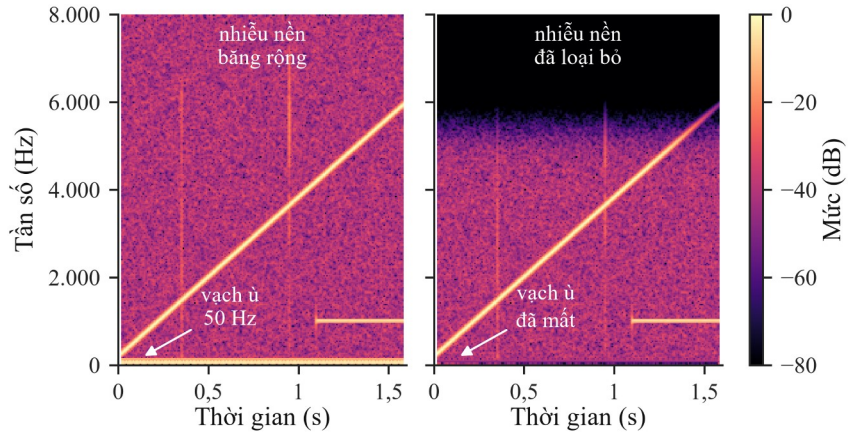
def on_dinh(a):
    """He IIR on dinh khi moi nghiem cua a nam trong duong tron"""
    if len(a) <= 1:
        return True                 # FIR: luon on dinh
    nghiem = np.roots(a)
    return bool(np.all(np.abs(nghiem) < 1.0))

if not on_dinh(a):
    raise ValueError("bo loc khong on dinh — se vo")
```

Liệt kê 4.4 nên được gọi một lần mỗi khi hệ số bộ lọc thay đổi. Nó tốn vài micro giây và tránh được cả một loại lỗi mà nếu không kiểm thì chỉ phát hiện ra lúc nghe thấy tiếng méo nặng trong tai nghe.

4.6.6 Soát lỗi bộ lọc bằng phổ đồ

Hai mục 4.5 và 4.6 gặp nhau ở đây. Khi đã chỉnh xong một bộ lọc, cách kiểm tốt nhất không phải nghe mà là nhìn: vẽ phổ đồ của tín hiệu trước và sau khi lọc, đặt cạnh nhau, dùng *chung* một thang màu.



(a) Trước khi lọc

(b) Sau khi lọc thông dải 130 Hz – 5 kHz

Hình 4.28 Phổ đồ trước và sau khi lọc thông dải, vẽ trên cùng một thang màu

Nghe bằng tai chỉ nói được "nghe sạch hơn". Phổ đồ nói được "vạch ù 50 Hz đã hết, nhưng dải trên 5 kHz cũng mất theo" — và câu thứ hai mới là câu giúp chỉnh lại tham số. Nó cũng cho thấy bộ lọc tác động ở *đoạn nào* của tín hiệu, điều mà một biểu đồ đáp tuyến không nói được.

Dùng chung thang màu là bắt buộc

Nếu để mỗi phổ đồ tự chuẩn hoá theo đỉnh của chính nó thì sau khi lọc, phần còn lại được kéo sáng lên cho vừa thang — và hai hình trông gần như giống nhau dù bộ lọc đã cắt rất mạnh.

Cách làm đúng: tính mốc chuẩn từ tín hiệu *trước* khi lọc, rồi dùng đúng mốc ấy cho cả hai hình. Đây là cùng một nguyên tắc với việc đặt vmin cố định đã nói ở mục 4.5.5.

Gộp lại thành một quy trình soát lỗi bộ lọc gồm bốn bước, dùng được cho mọi bài toán lọc trong phần còn lại của cuốn sách.

- **Bước 1.** Vẽ phổ đồ tín hiệu gốc và chỉ ra bằng mắt chỗ có vấn đề — vạch ù, nhiễu nền, tiếng rít — kèm tần số ước lượng của nó.
- **Bước 2.** Chọn kiểu lọc và tần số cắt từ con số vừa đọc, chứ không từ trực giác. Kiểm điều kiện ổn định nếu dùng IIR.
- **Bước 3.** Lọc, rồi vẽ lại phổ đồ trên *cùng* một thang màu và đặt cạnh hình cũ.
- **Bước 4.** Trả lời hai câu: cái cần bỏ đã bỏ được chưa, và cái không được đụng tới có bị đụng không? Chỉ khi cả hai câu đều xong mới nghe để xác nhận lại bằng tai.

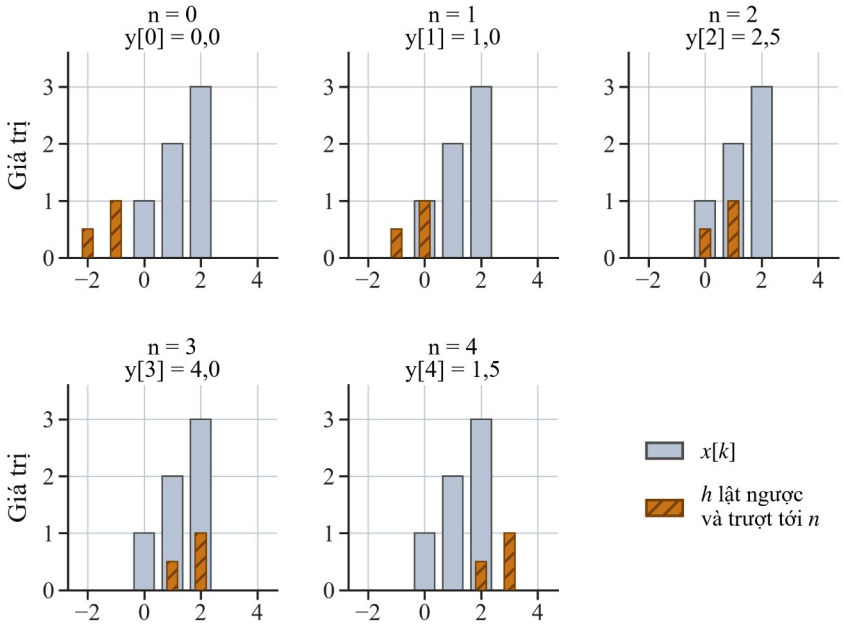
Thứ tự bốn bước này quan trọng. Nghe trước rồi mới chỉnh là cách làm dễ dẫn tới việc chỉnh quá mức mà không nhận ra, vì tai quen rất nhanh với thứ vừa nghe. Nhìn trước, chỉnh theo con số, rồi mới nghe để xác nhận là thứ tự an toàn hơn nhiều.

4.7 Phép toán tích chập

4.7.1 Ba động tác

Tích chập là phép toán mang tên đáng sợ nhất của chương, nhưng lại là phép có định nghĩa đơn giản nhất. Nó chỉ gồm ba động tác lặp đi lặp lại: *Lật ngược* dãy thứ hai, *trượt* nó tới vị trí đang xét, rồi *nhân* từng cặp trùng nhau và *cộng* lại.

$$y[n] = (x * h)[n] = \sum_{k=0}^{L_h - 1} h[k]\, x[n-k] \tag{4.11}$$



Hình 4.29 Tích chập của [1; 2; 3] với [0; 1; 0,5] làm từng bước một

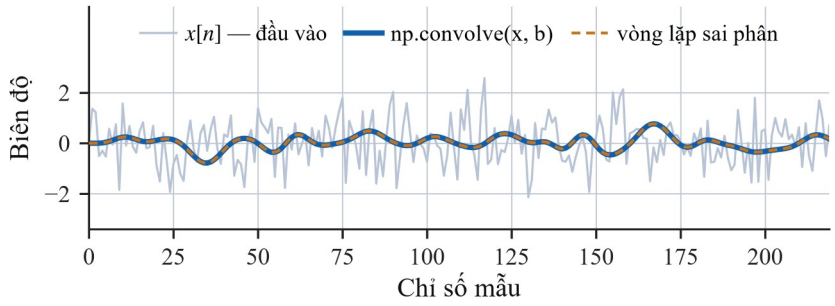
Bảng 4.14 Ví dụ số tính được bằng tay

n	Các cặp nhân vào nhau	y[n]
0	1 × 0	0,0
1	1 × 1 + 2 × 0	1,0
2	1 × 0,5 + 2 × 1 + 3 × 0	2,5
3	2 × 0,5 + 3 × 1	4,0
4	3 × 0,5	1,5

Kết quả dài 5 phần tử, đúng bằng 3 + 3 − 1. Đây là một quy tắc chung: chập một dãy dài L_x với một dãy dài L_h cho ra dãy dài $L_x + L_h - 1$. Điều đó có nghĩa vật lý rõ ràng — nếu h là đáp ứng xung của một phòng vang hai giây thì tín hiệu ra dài hơn tín hiệu vào đúng hai giây, chính là phần đuôi vang.

4.7.2 Lọc FIR và tích chập là cùng một phép toán

Đặt công thức (4.11) cạnh công thức (4.10) với dãy hồi tiếp rỗng, sẽ thấy chúng là một. Dãy hệ số b của một bộ lọc FIR *chính là* đáp ứng xung của nó, và lọc bằng dãy ấy chính là chập tín hiệu với đáp ứng xung ấy.



(a) Hai cách tính cho hai đường trùng khít



(b) Chênh lệch giữa hai cách tính

Hình 4.30 Lọc bằng vòng lặp sai phân và chập bằng np.convolve cho kết quả trùng khít

Một câu đáng nhớ với người làm nghề

Lọc là chập. Chập với đáp ứng xung của một bộ lọc thì được hiệu ứng lọc. Chập với đáp ứng xung của một cái phòng thì được hiệu ứng vang. Chập với đáp ứng xung của một cái loa cũ thì được âm sắc của cái loa ấy.

Nói cách khác: mọi hệ thống tuyến tính bất biến theo thời gian — bộ lọc, phòng, loa, micro, băng từ — đều được mô tả *hoàn toàn* bởi đáp ứng xung của nó, và việc cho tín hiệu đi qua nó chính là phép chập.

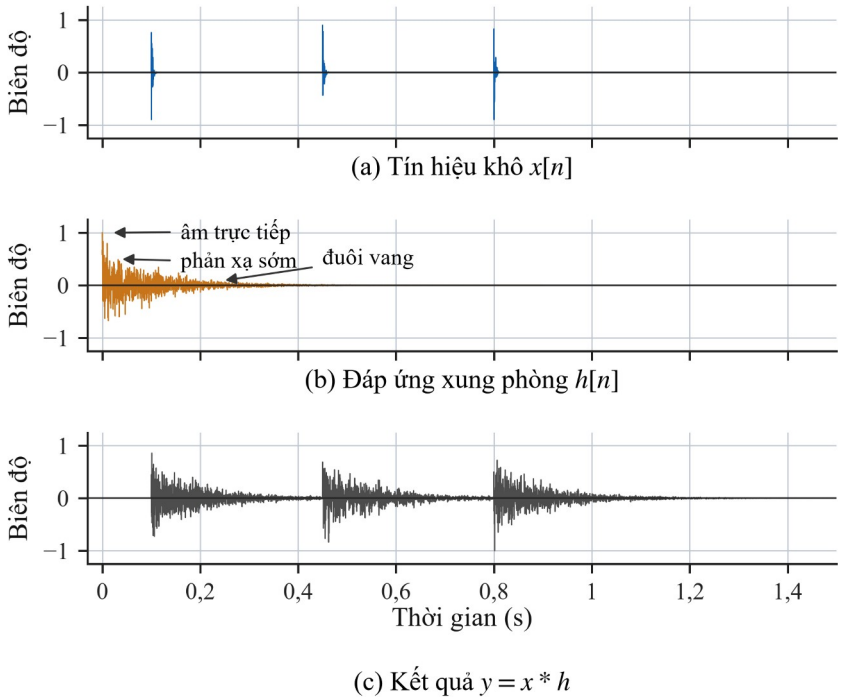
Đó là lý do một phép toán trông tầm thường lại là nền của cả một ngành công nghiệp.

4.7.3 Vang tích chập

Ứng dụng nổi tiếng nhất của tích chập trong hậu kỳ điện ảnh là vang tích chập. Nguyên lý gồm hai bước, và cả hai đều đã học xong.

– Bước một: ghi lại đáp ứng xung của một không gian thật. Đặt loa và micro trong nhà thờ, hầm, phòng thu, rồi phát một tiếng nổ ngắn hoặc một tiếng quét tần số và thu lại toàn bộ phần đuôi. Kết quả là một tệp WAV dài vài giây.

– Bước hai: chập giọng thu khô với tệp ấy. Kết quả nghe đúng như được thu trong chính không gian đó.



Hình 4.31 Tín hiệu khô chập với đáp ứng xung của phòng cho tín hiệu mang đặc tính phòng ấy

Điều đáng nhấn mạnh: đây *không* phải một phép mô phỏng hay xấp xỉ. Nếu căn phòng là một hệ tuyến tính bất biến — và với âm thanh ở mức bình thường thì nó đúng là thế — thì phép chập cho ra chính xác cái mà micro sẽ thu được nếu chúng ta thật sự đem nguồn âm ấy vào phòng ấy. Đó là lý do vang tích chập nghe thật hơn hẳn các bộ vang tổng hợp bằng thuật toán.

Giới hạn của vang tích chập

Đáp ứng xung mô tả *một* vị trí nguồn và *một* vị trí micro trong phòng. Muốn nhân vật đi lại trong không gian thì cần nhiều đáp ứng xung và phải chuyển tiếp mượt giữa chúng.

Phép chập giả định hệ tuyến tính bất biến. Nó không mô tả được những gì phi tuyến — méo của loa khi mở to, tiếng rung của vật thể trong phòng.

Đổi tham số của vang tích chập rất khó. Muốn "vang ngắn hơn một chút" thì phải sửa chính tệp đáp ứng xung, chứ không có núm điều chỉnh nào. Vang tổng hợp bằng thuật toán kém về độ chân thực nhưng vượt trội về khả năng điều chỉnh — nên trong thực tế người ta dùng cả hai.

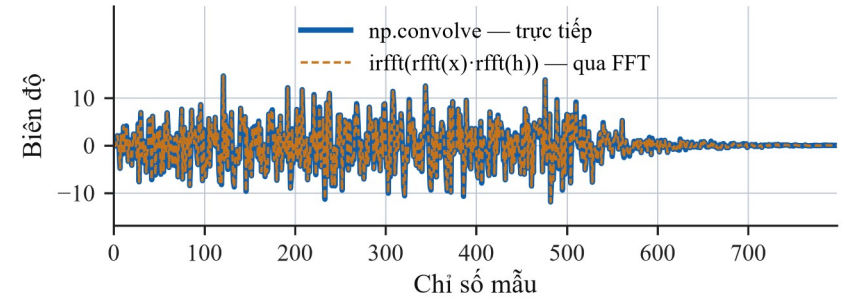
4.7.4 Định lý tích chập — và vì sao nó cứu cả ứng dụng

Có một vấn đề rất thực tế với những gì vừa nói. Chập trực tiếp theo công thức (4.11) tốn $L_x \times L_h$ phép nhân. Với một đáp ứng xung hai giây ở 44.100 Hz, xử lý một giây tín hiệu cần khoảng bốn tỉ phép nhân — vượt xa khả năng chạy thời gian thực.

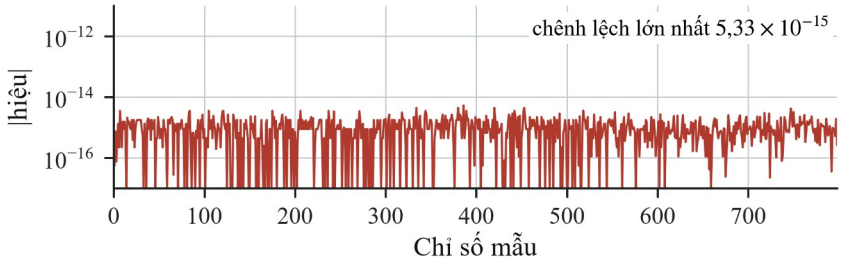
Lối thoát nằm ở một trong những kết quả đẹp nhất của lý thuyết tín hiệu.

$$y = x * h \iff Y[k] = X[k] \cdot H[k]\,; \qquad L_y = L_x + L_h - 1 \tag{4.12}$$

Phép chập trong miền thời gian tương ứng với phép *nhân từng phần tử* trong miền tần số. Nhờ đó, thay vì chập trực tiếp, chúng ta làm FFT cả hai vế, nhân hai kết quả với nhau, rồi làm FFT ngược (IFFT, tiếng Anh: Inverse FFT) — ba bước đều rẻ.

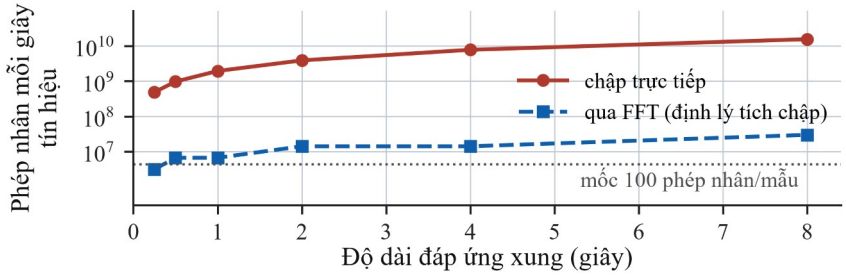


(a) Hai con đường, một kết quả

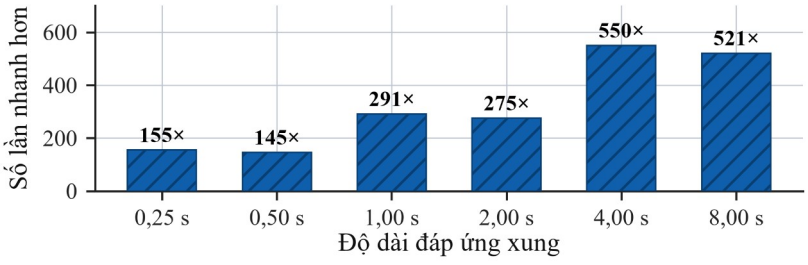


(b) Chênh lệch giữa hai con đường

Hình 4.32 Kiểm chứng định lý bằng số: hai con đường cho cùng một kết quả



(a) Chi phí xử lý một giây âm thanh



(b) Tỉ lệ giữa hai cách

Hình 4.33 Chi phí của chập trực tiếp và của chập qua FFT theo độ dài đáp ứng xung

Với đáp ứng xung hai giây, cách qua FFT nhanh hơn khoảng 275 lần. Đó chính là khoảng cách giữa "phải chờ kết xuất" và "nghe được ngay trong lúc dựng". Mọi phần mềm vang tích chập trên thị trường đều làm theo cách này, thường kết hợp thêm kỹ thuật chia khối để giảm độ trễ.

4.8 Thực hành và bài tập

4.8.1 Ba bài thực hành

Ba bài dưới đây làm theo thứ tự. Mỗi bài xây trên kết quả của bài trước.

Bảng 4.15 Ba bài thực hành và tiêu chí hoàn thành

Bài	Việc phải làm	Coi là xong khi
TH1 — Phổ	Đọc một tệp WAV, vẽ dạng sóng và phổ của một đoạn 4.096 mẫu, có nhân cửa sổ Hann và trục hoành đúng đơn vị hertz	Chỉ ra được ba tần số mạnh nhất và nói được chúng có phải hài của nhau không
TH2 — Cửa sổ	Vẽ lại đúng phổ ấy nhưng bỏ cửa sổ đi, đặt hai hình cạnh nhau	Chỉ ra được trên hình phần nào là rò phổ và giải thích được nó từ đâu ra
TH3 — Phổ đồ và lọc	Vẽ phổ đồ của cả tệp với ba giá trị N khác nhau, rồi lọc thông dải và vẽ lại	Nói được mỗi giá trị N hợp với câu hỏi nào, và chỉ ra được trên phổ đồ chỗ bộ lọc đã cắt

4.8.2 Bài tập

Bảy bài dưới đây. Mọi đáp số đều kiểm được bằng mô-đun sigtools.py; bài làm phải trình bày cách tính chứ không chỉ ghi kết quả.

Bài 1. Một khung FFT dài 2.048 mẫu ở tần số lấy mẫu 48.000 Hz. Tính độ phân giải tần số, thời lượng khung, và tần số ứng với vạch thứ 100.

Bài 2. Cần phân biệt hai nhạc cụ chơi hai nốt cách nhau 12 Hz, ở 44.100 Hz, dùng cửa sổ Hann. Tính số mẫu tối thiểu và thời lượng thu tương ứng. Nếu chỉ có 0,2 giây tín hiệu thì có làm được không, và vì sao?

Bài 3. Một người vẽ phổ của một sóng sin 1.000 Hz sạch nhưng thấy năng lượng trải rộng khắp biểu đồ. Nêu ba nguyên nhân có thể, xếp theo thứ tự khả năng, và nêu cách kiểm từng nguyên nhân.

Bài 4. Với STFT có N = 4.096 và H = 1.024 ở 44.100 Hz: tính độ phân giải tần số, thời lượng khung, phần trăm chồng lấn, và số khung cho một đoạn 5 giây.

Bài 5. Cho bộ lọc có b = [0,5; 0,5] và a = [1]. Đây là FIR hay IIR? Tính đáp ứng xung tám mẫu đầu. Bộ lọc này là thông thấp hay thông cao, và giải thích bằng lời tại sao — không cần vẽ đáp tuyến.

Bài 6. Chập x = [2; 0; −1] với h = [1; 1; 1] bằng tay. Kết quả dài bao nhiêu? Nếu h là một bộ lọc thì nó làm gì với tín hiệu?

Bài 7. Một đoạn thu phỏng vấn ngoài trời dài 3 phút có ba vấn đề: tiếng ù 50 Hz, tiếng gió ở dải rất trầm, và một tiếng rít cao tần không đổi ở khoảng 15 kHz. Hãy trình bày quy trình xử lý: dùng công cụ nào để *chẩn đoán* từng vấn đề, dùng bộ lọc gì để chữa, theo thứ tự nào, và nêu một rủi ro của mỗi bước.

Cách trình bày bài làm

Mỗi bài tính toán phải có ba phần: công thức dùng (ghi rõ số hiệu), thay số, kết quả kèm đơn vị.

Số thập phân dùng dấu phẩy; nghìn dùng dấu chấm. Kết quả làm tròn tới một chữ số thập phân trừ khi đề nói khác.

Bài 3 và bài 7 không có đáp số duy nhất. Hãy tự đánh giá theo tính hợp lý của lập luận và theo việc có nêu được rủi ro hay không, chứ không theo việc trùng với đáp án mẫu.

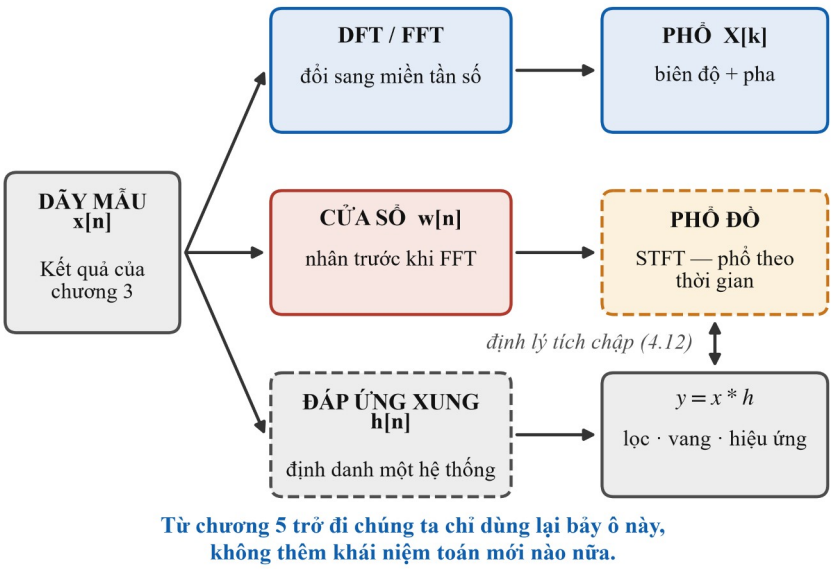
4.8.3 Ba lỗi hay gặp nhất khi làm bài tập chương này

Bảng 4.16 Ba lỗi hay gặp và cách tự phát hiện

Lỗi	Dấu hiệu nhận ra	Cách chữa
Lẫn Δf với khoảng cách tối thiểu tách được	Kết luận rằng N = 1.024 tách được hai tần số cách 43 Hz	Luôn nhân thêm bề rộng búp chính của cửa sổ — công thức (4.7)
Bỏ sót việc nhân cửa sổ rồi kết luận có nhiễu	Sàn biểu đồ cao bất thường ở một tín hiệu sạch	Đổi độ dài khung; nếu sàn đổi theo thì đó là rò chứ không phải nhiễu
Lẫn tần số cắt với nơi tín hiệu biến mất	Nói "lọc thông cao 100 Hz thì dưới 100 Hz mất hết"	Nhớ rằng ở tần số cắt tín hiệu còn khoảng 70 % biên độ; độ dốc do bậc lọc quyết định

4.9 Tổng kết chương

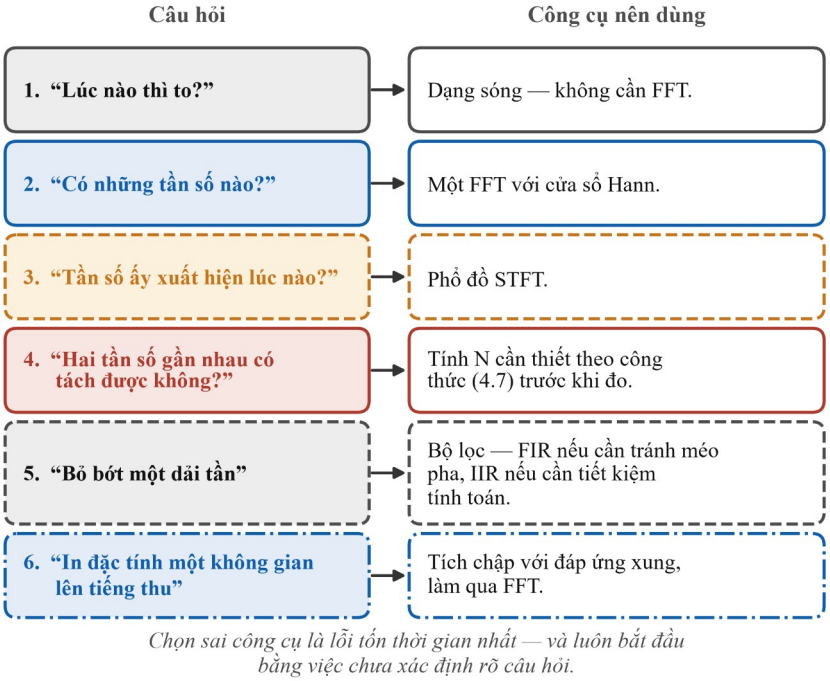
4.9.1 Bảy khái niệm và cách chúng nối với nhau



Hình 4.34 Bản đồ khái niệm của chương 4 — bảy ô và các đường nối

Bản đồ trên là toàn bộ nền toán của cuốn sách. Từ chương 5 trở đi, khi bắt đầu viết mã xử lý thật, chúng ta chỉ dùng lại bảy ô này chứ không thêm khái niệm toán mới nào nữa. Người đọc nên vẽ lại bản đồ này bằng trí nhớ một lần trước chương sau.

4.9.2 Chọn công cụ theo câu hỏi



Hình 4.35 Bảng quyết định: câu hỏi nào thì dùng công cụ nào

Lỗi tốn thời gian nhất trong xử lý tín hiệu không phải lỗi tính sai, mà là dùng sai công cụ — và nó luôn bắt đầu từ việc chưa xác định rõ câu hỏi cần trả lời. Bảng trên nên được xem lại mỗi khi bắt đầu một bài toán mới.

4.9.3 Năm điều phải nhớ

- **Một tín hiệu có hai mô tả tương đương.** Miền thời gian trả lời "lúc nào", miền tần số trả lời "cái gì". Chuyển qua lại được là một phần lớn của kỹ năng.
- **FFT cho N/2 + 1 con số, không cho một đường cong.** Khoảng cách giữa hai vạch là f_s chia N. Muốn vạch dày hơn thì chỉ có một cách: lấy thêm mẫu.
- **Luôn nhân cửa sổ.** Hann là mặc định. Không nhân cửa sổ là lỗi không báo — chương trình vẫn chạy, biểu đồ vẫn có vẻ hợp lý, nhưng kết quả sai.
- **$\Delta f \times \Delta t = 1$, không thể vượt qua.** Mịn về tần số thì nhoè về thời gian. Chọn N là chọn đại lượng nào được ưu tiên hơn.
- **Lọc là chập.** Mọi hệ tuyến tính bất biến đều được mô tả đầy đủ bởi đáp ứng xung của nó, và cho tín hiệu đi qua nó chính là phép chập với đáp ứng xung ấy.

4.9.4 Tự kiểm trước khi sang chương 5

Tám câu dưới đây nên trả lời được không cần mở tài liệu. Câu nào chưa trả lời được thì đọc lại đúng mục ghi trong ngoặc.

- Nói bằng lời DFT làm gì, không dùng chữ "số phức". (4.2.2)
- Với N = 2.048 ở 44.100 Hz thì Δf bằng bao nhiêu? (4.2.3)
- Vì sao một sóng sin sạch lại có thể hiện lên như tín hiệu bẩn? (4.4.1)
- Cửa sổ Hann đánh đổi cái gì lấy cái gì? (4.4.3)
- Muốn tách hai tần số cách 10 Hz thì cần thu bao lâu? (4.4.5)
- Trên phổ đồ giọng nói, formant trông thế nào? (4.5.3)
- FIR và IIR khác nhau ở ba điểm nào? (4.6.3)
- Vì sao vang tích chập phải làm qua FFT? (4.7.4)

4.9.5 Chương sau

Chương 5 mở đầu phần lập trình thật. Từ đó trở đi, mỗi chương kết thúc bằng một đoạn mã chạy được chứ không phải một công thức. Nội dung chương 5 gồm: mảng numpy dành cho âm thanh, đọc và ghi tệp WAV, chỉnh gain theo decibel, trộn nhiều nguồn, cắt ghép, và chuẩn hoá đỉnh.

Chuẩn bị trước chương sau: cài xong môi trường Python với numpy, scipy, soundfile và matplotlib; chuẩn bị hai tệp WAV của riêng mình — một giọng đọc và một đoạn nhạc, mỗi tệp khoảng ba mươi giây; và chạy được đoạn mã ở mục 4.2.7 trên hai tệp ấy.

Tài liệu tham khảo của chương

Sách và giáo trình

- Smith, J. O. — Mathematics of the Discrete Fourier Transform (DFT), with Audio Applications. Nguồn nền của phần DFT, cửa sổ và rò phổ; bản trực tuyến của CCRMA, Đại học Stanford.
- Lyons, R. G. — Understanding Digital Signal Processing, ấn bản 3. Nguồn của phần FFT, bộ lọc số và tích chập; là sách nên đọc tiếp cho ai muốn đi sâu.
- Oppenheim, A. V., Schafer, R. W. — Discrete-Time Signal Processing. Nguồn chuẩn mực về phương trình sai phân, FIR và IIR, điều kiện ổn định.
- Harris, F. J. — On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform, Proceedings of the IEEE, 1978. Nguồn gốc của các con số về búp chính và búp bên ở mục 4.4.
- Zölzer, U. (chủ biên) — DAFX: Digital Audio Effects, ấn bản 2. Nguồn của phần STFT ứng dụng và vang tích chập; sẽ dùng lại nhiều ở chương 9 – 10.

Tài liệu kỹ thuật và công cụ

- Tài liệu numpy.fft và scipy.signal — hai mô-đun được dùng suốt chương này và toàn bộ phần lập trình của cuốn sách.
- Cooley, J. W., Tukey, J. W. — An Algorithm for the Machine Calculation of Complex Fourier Series, 1965. Bài báo công bố thuật toán FFT.
- Sonic Visualiser — phần mềm mã nguồn mở của Queen Mary University of London, dùng để xem phổ đồ ở bài thực hành TH3.

Đọc thêm theo hướng

- Muốn hiểu sâu phần cửa sổ và rò phổ: đọc Harris (1978), rồi thử dựng lại bảng của ông bằng mã của mình — đó là bài luyện tốt nhất cho cả mục 4.4.
- Muốn đi sâu vào thiết kế bộ lọc: đọc Lyons chương về FIR và IIR, rồi thử thiết kế một bộ chắn dải 50 Hz đủ hẹp để không ăn vào giọng nói.
- Muốn đi tiếp về STFT ứng dụng — dịch cao độ, co giãn thời gian, tách nguồn: đọc DAFX chương 7 và 8. Chương 10 của cuốn sách sẽ chạm tới phần này.

Một lưu ý về các con số trong bài

> **Cách đọc các con số**
>
> Các con số đặc trưng của cửa sổ — bề rộng búp chính, mức búp bên — là giá trị tiêu chuẩn lấy từ Harris (1978), làm tròn tới một chữ số thập phân.
>
> Các con số về chi phí tính toán là *ước lượng* bậc độ lớn, dùng để so sánh hai cách làm chứ không phải để dự đoán thời gian chạy thật trên một máy cụ thể.
>
> Mọi mức rò phổ, mọi độ lệch giữa hai cách tính, và mọi đáp số bài tập trong bài đều sinh ra từ mô-đun sigtools.py bằng cách đo trên tín hiệu thật, đã qua 157 phép tự kiểm.