

## Chương 3

### LẤY MẪU VÀ LƯỢNG TỬ

#### 3.1. Mở đầu

Mục đích chính của cuốn sách là nghiên cứu các kỹ thuật cơ bản để mô phỏng hệ thống truyền thông trên máy tính điện tử số ở mức chính xác nhất có thể. Hầu hết các ứng dụng, đều phải tạo và xử lý dạng sóng. Máy tính chỉ có thể xử lý các con số (biểu diễn các mẫu tín hiệu). Ngoài ra, do tính hữu hạn về độ dài từ mã của máy tính, nên các giá trị mẫu có độ chính xác hữu hạn. Nói cách khác, các giá trị của mẫu được lượng tử hóa. Do vậy, lấy mẫu và lượng tử hóa là các hoạt động căn bản trong quá trình mô phỏng, và chúng gây thêm sai số vào kết quả mô phỏng. Việc khử hoàn toàn các sai số này là không thể, và cần phải có giải pháp dung hòa. Có thể thấy rằng: cách tốt nhất là giảm thiểu các ảnh hưởng gây sai số của quá trình lấy mẫu và lượng tử. Cũng cần lưu ý rằng, các hệ thống vật lý sử dụng kỹ thuật xử lý tín hiệu số DSP cũng phải chịu các ảnh hưởng của sai số lượng tử và lấy mẫu.

Vì vậy, chương này xét một số chủ đề lấy mẫu và trình bày các giá trị mẫu trong mô phỏng hệ thống truyền thông. Xét vấn đề hai định lý lấy mẫu cơ bản là định lý lấy mẫu thông thấp và định lý lấy mẫu thông dải. Vì các tín hiệu thông dải thường được biểu diễn bởi tín hiệu thông thấp trong các mô phỏng hệ thống, nên định lý lấy mẫu thông thấp là định lý quan trọng nhất. Ta sẽ thấy rằng, có thể lấy mẫu tín hiệu thông thấp bằng tần hạn chế, và có thể khôi phục tín hiệu thông dải từ các giá trị mẫu nếu tần số lấy mẫu lớn hơn 2 lần độ rộng băng tần của tín hiệu thông thấp hạn băng. Định lý lấy mẫu thông dải, mặc dù không hữu dụng lắm so với định lý thông thấp trong mô phỏng, nhưng mang lại kết quả tương đương. Tín hiệu thông dải có thể được lấy mẫu và được tái tạo nếu tần số lấy mẫu nằm trong khoảng  $2B$  và  $4B$  trong đó  $B$  là độ rộng băng tần của tín hiệu thông dải được lấy mẫu.

Tiếp theo là xét quá trình lượng tử. Sai số lượng tử xuất hiện trong mọi mô phỏng, vì giá trị mẫu chỉ được biểu diễn bởi các từ mã chiều dài hữu hạn. Xét hai loại sai số lượng tử: sai số do biểu diễn số với dấu phẩy tĩnh và sai số lượng tử do biểu diễn số bằng dấu phẩy động. Ta sẽ thấy rằng, khi sử dụng biểu diễn số dấu phẩy tĩnh, thì tỉ số tín hiệu trên tạp âm lượng tử tăng 6 dB mỗi khi chiều dài từ mã tăng thêm một bit. Nói chung, khi mô phỏng trên các máy tính sử dụng biểu diễn dấu phẩy động, thì thường bỏ qua tạp âm lượng tử được. Tuy nhiên, tạp âm này không bao giờ bằng 0, tích lũy dần và làm giảm đáng kể độ chính xác kết quả mô phỏng. Ta nên cảnh giác với nguồn nhiễu này.

Sau đó là đề cập vấn đề *tái tạo* và *nội suy*. Ta sẽ thấy rằng, nếu lấy mẫu tín hiệu thông thấp hạn băng với tần số lấy mẫu lớn hơn 2 lần độ rộng băng thông của tín hiệu đó, thì có thể khôi phục (tái tạo) tín hiệu liên tục cơ bản đó mà không có sai số bằng cách đánh trọng số cho mỗi mẫu với một sóng dạng  $\sin(x)/x$ , tương đương với việc cho các mẫu tín hiệu đó qua bộ lọc thông thấp lý tưởng. Kết quả là dạng sóng được xác định cho tất cả các giá trị theo thời

gian và bằng cách tách các mẫu "mới" giữa các mẫu gốc, ta có thể tạo các mẫu nội suy. Quá trình này gọi là tăng mẫu, làm tăng tần số lấy mẫu hiệu quả. Quá trình ngược lại là giảm mẫu thực hiện bằng cách trích (tách) mỗi mẫu thứ  $M$  từ một tập các mẫu gốc. Triển khai sử dụng tăng mẫu (upsampling) và giảm mẫu (downsampling) trong mô phỏng dẫn đến xuất hiện nhiều tần số lấy mẫu, chúng hữu hiệu khi mô phỏng hệ thống các tín hiệu băng tần khác nhau lớn ví dụ như hệ thống thông tin trải phổ.

Chủ đề quan trọng là quan hệ giữa tần số lấy mẫu với dạng xung sử dụng để truyền dẫn dạng sóng. Dạng xung được giả thiết là giới hạn về thời gian, và do vậy nó không thể được giới hạn băng (*băng tần vô hạn*). Vì vậy, xuất hiện các sai số chồng phổ. Tiêu chuẩn để lựa chọn tần số lấy mẫu là xác định tỉ số tín hiệu trên tạp âm cần thiết, ở đó nguồn tạp âm gây ra sai số chồng phổ (sai số chồng phổ tạo thành nguồn tạp âm). Phương pháp chung để xác định PSD của tín hiệu điều chế và tích phân số của PSD này xác định tỉ số tín hiệu trên tạp âm chồng phổ.

### 3.2. Lấy mẫu

Quá trình biến đổi tín hiệu tương tự thành tín hiệu số theo trình tự: lấy mẫu, lượng tử hóa và mã hóa được minh họa ở hình 3.1. Tín hiệu tương tự  $x(t)$  liên tục cả về thời gian và độ lớn. Kết quả của quá trình lấy mẫu là các tín hiệu rời rạc về thời gian nhưng liên tục về biên độ được gọi là tín hiệu số liệu mẫu. Tín hiệu số được tạo ra từ tín hiệu dữ liệu mẫu bằng cách mã hóa các giá trị mẫu thành tập các giá trị hữu hạn. Ta sẽ thấy, các lỗi sẽ xảy ra tại mỗi bước của quá trình này.

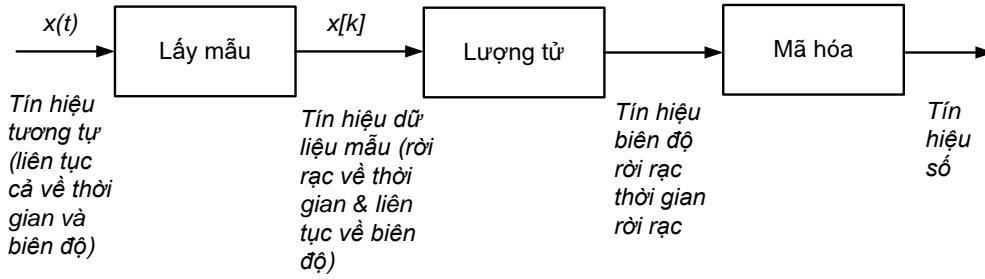
#### 3.2.1. Định lý lấy mẫu thông thấp

Bước đầu tiên trong quá trình tạo tín hiệu số từ tín hiệu liên tục  $x(t)$  là lấy mẫu  $x(t)$  tại các điểm cách đều theo thời gian để tạo ra các giá trị mẫu  $x_s(t) = x(kT_s) = x[k]$ . Tham số  $T_s$  là chu kỳ lấy mẫu, là nghịch đảo của tần số lấy mẫu  $f_s$ .

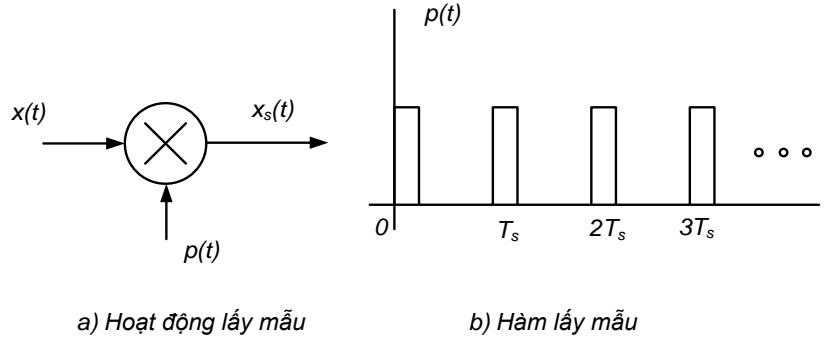
Mô hình lấy mẫu được minh họa ở hình 3.2. Tín hiệu  $x(t)$  được nhân với một xung tuần hoàn  $p(t)$  tạo ra tín hiệu mẫu  $x_s(t)$ . Nói cách khác:

$$x_s(t) = x(t)p(t) \quad (3.1)$$

Hàm lấy mẫu  $p(t)$  được giả thiết là một xung hẹp nhận giá trị 1 hoặc 0. Theo đó,  $x_s(t) = x(t)$  khi  $p(t) = 1$  và  $x_s(t) = 0$  khi  $p(t) = 0$ . Cần đặc biệt lưu ý đến chu kỳ của hàm lấy mẫu  $p(t)$  còn dạng sóng của  $p(t)$  là tùy chọn. Hàm lấy mẫu được minh họa trong hình 3.2, trực giác, nó thể hiện một cái chuyển mạch tại các thời điểm lấy mẫu.



Hình 3.1: Lấy mẫu, lượng tử, mã hóa



Hình 3.2: Lấy mẫu và hàm lấy mẫu

Vì  $p(t)$  là hàm tuần hoàn, nên được biểu diễn bằng chuỗi Fourier:

$$p(t) = \sum_{n=-\infty}^{\infty} C_n \cdot e^{j2\pi n f_s t} \quad (3.2)$$

Trong đó, các hệ số chuỗi Fourier được cho bởi:

$$C_n = \frac{1}{T_s} \int_{-T_s/2}^{T_s/2} p(t) \cdot e^{-j2\pi n f_s t} dt \quad (3.3)$$

Thay (3.2) vào (3.1) ta có:

$$x_s(t) = x(t) \sum_{n=-\infty}^{\infty} C_n e^{j2\pi n f_s t} \quad (3.4)$$

Để rút ra định lý lấy mẫu và các điều kiện để khôi phục tín hiệu  $x(t)$  từ các mẫu  $x(kT_s)$ , cần phải xác định phổ của  $x_s(t)$  và chỉ ra rằng  $x(t)$  có thể khôi phục từ  $x_s(t)$  với các điều kiện cần thiết. Muốn vậy, cần phải xác định: (i) phổ của tín hiệu mẫu  $x_s(t)$ ; (ii) phổ của tín hiệu thông tin  $x(t)$ ; (iii) phổ của xung lấy mẫu  $p(t)$ ; (iv) các mối quan hệ từ đó tìm được điều kiện lấy mẫu hay định lý lấy mẫu. Theo đó, ta tiến hành như sau:

*Phổ của tín hiệu  $x_s(t)$ :* Theo định nghĩa, biến đổi Fourier của tín hiệu mẫu là:

$$X_s(f) = \int_{-\infty}^{\infty} x(t) \underbrace{\sum_{n=-\infty}^{\infty} C_n e^{j2\pi n f_s t}}_{x_s(t)} e^{-j2\pi f t} dt \quad (3.5)$$

Chuyển đổi thứ tự lấy tích phân và lấy tổng ta được:

$$X_s(f) = \sum_{n=-\infty}^{\infty} C_n \int_{-\infty}^{\infty} x(t) e^{-j2\pi(f-nf_s)t} dt \quad (3.6)$$

*Phổ của tín hiệu tương tự  $x(t)$* : Phổ của  $x(t)$  được xác định theo định nghĩa biến đổi Fourier là:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt \quad (3.7)$$

Từ (3.6), ta có biến đổi Fourier của tín hiệu mẫu được viết như sau:

$$X_s(f) = \sum_{n=-\infty}^{\infty} C_n X(f - nf_s) \quad (3.8)$$

Từ (3.8), thấy rõ quan hệ giữa phổ của tín hiệu mẫu  $x_s(t)$  và phổ của tín hiệu tương tự  $x(t)$ . Theo đó, ảnh hưởng của việc lấy mẫu tín hiệu liên tục là tái sinh phổ của tín hiệu được lấy mẫu xung quanh tần số 0 ( $f = 0$ ) và tất cả các hài của tần số lấy mẫu ( $f = nf_s$ ). Phổ được dịch, được đánh trọng số bởi hệ số chuỗi Fourier  $C_n$  của xung lấy mẫu  $p(t)$  tương ứng.

*Phổ của xung lấy mẫu  $p(t)$* : Vì các mẫu dữ liệu phải có giá trị bằng với giá trị của tín hiệu được lấy mẫu tại các thời điểm lấy mẫu và phải có chu kỳ lặp lại bằng với chu kỳ của xung lấy mẫu, nên định phù hợp cho xung lấy mẫu  $p(t)$  là:

$$p(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT_s) \quad (3.9)$$

được coi là lấy mẫu hàm xung kim trong đó các giá trị mẫu được thể hiện bởi các trọng số của các hàm xung kim. Thế (3.9) vào (3.3), nhận được:

$$C_n = \frac{1}{T_s} \int_{-T_s/2}^{T_s/2} \delta(t) e^{-j2\pi n f_s t} dt \quad (3.10)$$

Áp dụng tính chất dịch của hàm delta, ta có:

$$C_n = \frac{1}{T_s} = f_s \quad (3.11)$$

Sử dụng kết quả này vào (3.2) và biến đổi Fourier cho  $p(t)$  nhận được:

$$P(f) = f_s \sum_{n=-\infty}^{\infty} \delta(f - nf_s) \quad (3.12)$$

*Rút ra điều kiện lấy mẫu:* Vì lấy mẫu hàm xung kim  $C_n = f_s, \forall n$ . Nên phổ của tín hiệu mẫu  $X_s(f)$  theo (3.8) trở thành:

$$X_s(f) = f_s \sum_{n=-\infty}^{\infty} X(f - nf_s) \quad (3.13)$$

Lưu ý, có thể tìm được phổ tín hiệu mẫu  $X_s(f)$  theo biểu thức sau:

$$X_s(f) = X(f) \otimes P(f) \quad (3.14)$$

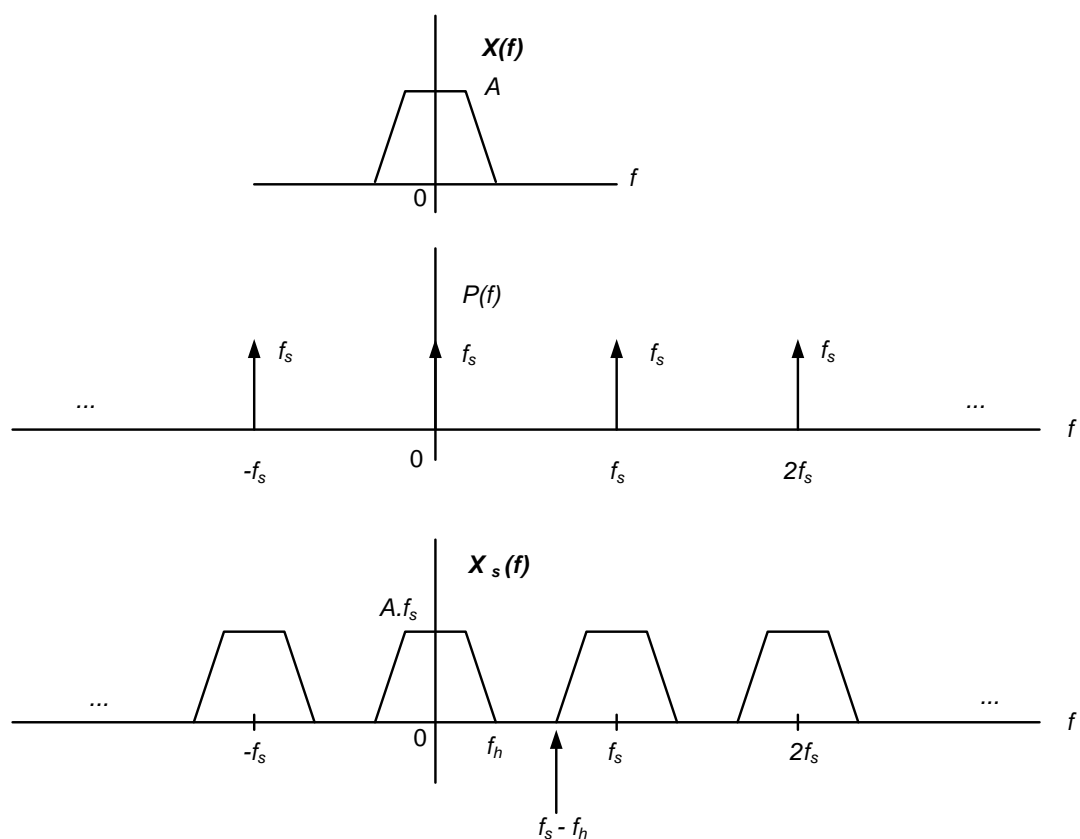
trong đó  $\otimes$  ký hiệu cho tích chập, tạo  $X_s(f)$  theo (3.14) được cho ở hình 3.3 đối với trường hợp tín hiệu giới hạn băng.

Định lý lấy mẫu có thể được triển khai từ việc quan sát hình 3.3. Để các mẫu  $x(nT_s)$  chứa toàn bộ thông tin của tín hiệu  $x(t)$  (quá trình lấy mẫu không làm mất thông tin) thì lấy mẫu phải được thực hiện sao cho có thể khôi phục  $x(t)$  từ các mẫu  $x(nT_s)$  mà không có lỗi. Việc tái tạo  $x(t)$  từ  $x(nT_s)$  được thực hiện bằng cách tách thành phần  $n = 0$  từ  $X_s(f)$  bằng cách lọc thông thấp. Để khôi phục tín hiệu  $x(t)$  từ  $x(nT_s)$  mà không có lỗi, thì vùng phổ của  $X_s(f)$  được định vị xung quanh  $f = \pm f_s$  [thành phần  $n = \pm 1$  trong (3.13)] không chồng lấn lên vùng phổ của  $X_s(f)$  được định vị xung quanh  $f = 0$  [thành phần  $n=0$  trong (3.13)], nói cách khác, tất cả các thành phần phổ được dịch trong (3.13) phải không chồng lấn nhau. Vì vậy cần phải đảm bảo điều kiện  $f_s - f_h > f_h$  hay  $f_s > 2f_h$ .

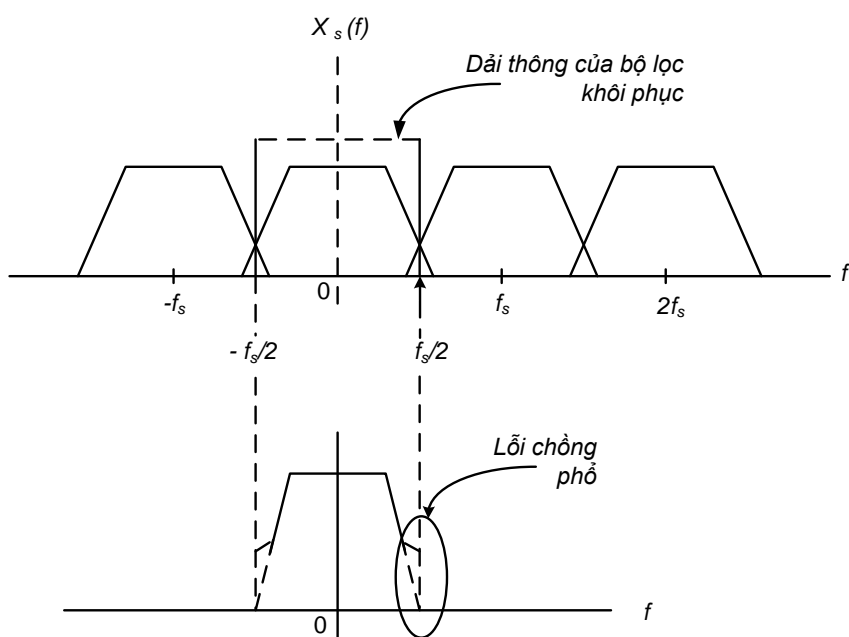
**Định lý 3.1:** *Có thể khôi phục tín hiệu băng tần hạn chế từ các mẫu của tín hiệu đó mà không có lỗi nếu tần số lấy mẫu  $f_s$  lớn hơn  $2f_h$ , trong đó  $f_h$  là tần số lớn nhất trong tín hiệu được lấy mẫu.*

Định lý 3.1 thường được gọi là định lý lấy mẫu thông thấp, nó cũng đúng đối với tín hiệu thông dải. Tuy nhiên, khi sử dụng định lý lấy mẫu thông thấp cho tín hiệu *thông dải* dẫn đến tần số lấy mẫu rất cao. Lấy mẫu tín hiệu thông dải sẽ được đề cập dưới đây.

Nếu  $f_s < 2f_h$  thì phần phổ có trung tâm tại  $f = \pm f_s$  sẽ chồng lấn lên phần phổ có trung tâm tại  $f = 0$  dẫn đến hậu quả là gây méo tín hiệu tại đầu ra  $x(t)$  của bộ lọc thông thấp như minh họa trong hình 3.4 (giả sử phổ  $x(t)$  là thực). Méo dạng này gọi là chồng phổ.



Hình 3.3: Lấy mẫu trong miền tần số



Hình 3.4: Tần số lấy mẫu thấp dẫn đến lỗi chồng phổ

### 3.2.2. Lấy mẫu tín hiệu ngẫu nhiên thông thấp

Dạng sóng  $x(t)$  ở trên được giả thiết là *tín hiệu năng lượng hữu hạn tất định*. Kết quả của các giả định này là, tồn tại biến đổi Fourier và định lý lấy mẫu dựa vào phổ (biến đổi Fourier) của tín hiệu. Trong hầu hết các ứng dụng, thường phải mô phỏng *xử lý* các hàm mẫu của một quá trình ngẫu nhiên. Vì vậy, thay vì chọn tần số lấy mẫu dựa vào phổ của tín hiệu tương tự (bằng cách biến đổi Fourier hàm tất định), ta chọn tần số lấy mẫu dựa vào mật độ phổ công suất PSD của tín hiệu ngẫu nhiên.

Trường hợp các tín hiệu *ngẫu nhiên* ta viết:

$$X_s(t) = X(t)P(t) \quad (3.15)$$

Hàm lấy mẫu  $P(t)$  khi này được viết là:

$$P(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT_s - D) \quad (3.16)$$

Trong đó  $D$  là biến ngẫu nhiên phân bố đều trong khoảng  $(0, T_s)$  và độc lập với  $X(t)$ . Lưu ý tính giống nhau giữa (3.15) và (3.1), giữa (3.16) và (3.9). Chỉ có 2 điểm khác biệt cơ bản là: (i) sử dụng các chữ in hoa trong các hàm thời gian  $X(t)$ ,  $P(t)$  và  $X_s(t)$  để ký hiệu cho các quá trình ngẫu nhiên; (ii) sử dụng biến ngẫu nhiên  $D$  trong (3.16) nhằm đảm bảo cho  $X_s(t)$  là quá trình ngẫu nhiên *dừng*. Nếu không có  $D$ , thì tín hiệu mẫu là quá trình dừng tuần hoàn (*cyclostationary*). Tác dụng của  $D$  là làm ngẫu nhiên gốc thời gian của  $P(t)$ .

Mật độ phổ công suất của  $X_s(t)$  tìm được thông qua hàm tự tương quan của:

$$X_s(t) = X(t) \sum_{k=-\infty}^{\infty} \delta(t - kT_s - D) \quad (3.17)$$

Biến đổi Fourier hàm tự tương quan ta được mật độ phổ công suất PSD của  $X_s(t)$  như sau:

$$S_{X_s}(f) = f_s^2 \sum_{n=-\infty}^{\infty} S_X(f - nf_s) \quad (3.18)$$

Trong đó  $S_X(f)$  là PSD của  $X(t)$ . Cần phải lưu ý: (i) khía cạnh giống nhau của (3.18) và (3.13); (ii) hình 3.3 và 3.4 áp dụng nếu phổ là tương ứng với PSD của  $X(t)$  và nếu các trục tọa độ được đánh nhãn phù hợp; (iii) vẫn dùng định lý lấy mẫu trước, và tín hiệu phải được lấy mẫu ở tần số lớn hơn gấp 2 lần tần số lớn nhất của tín hiệu để tránh méo do chồng phổ.

### 3.2.3. Lấy mẫu tín hiệu thông dải

Xét lấy mẫu tín hiệu thông dải. Có nhiều cách để biểu diễn tín hiệu thông dải bằng một tập các mẫu. Dưới đây xét 2 phương pháp phổ biến nhất:

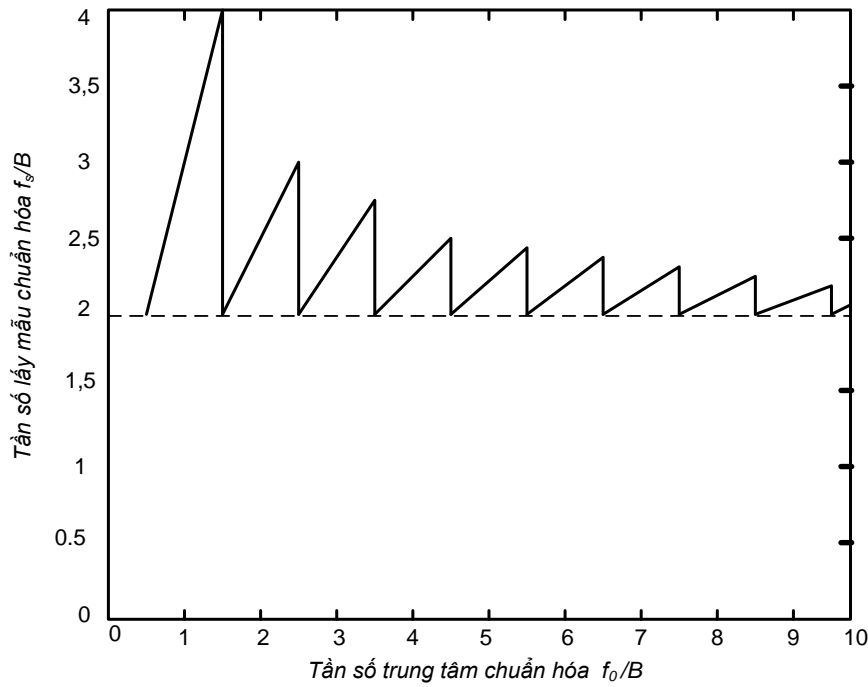
#### Định lý lấy mẫu thông dải:

Định lý lấy mẫu thông dải cho các tín hiệu thông dải giá trị thực được phát biểu như sau:

**Định lý 3.2:** Nếu một tín hiệu thông dải có độ rộng băng tần là  $B$  và có tần số cao nhất là  $f_h$ , thì tín hiệu đó có thể được lấy mẫu và được khôi phục sử dụng tần số lấy mẫu  $f_s = 2f_h/m$ ,

trong đó  $m$  là số nguyên lớn nhất nhỏ hơn  $f_h/B$ . Tất cả các tần số lấy mẫu cao hơn không nhất thiết sử dụng trừ khi chúng vượt quá  $2f_h$ , là giá trị của  $f_s$  được lấy theo định lý lấy mẫu thông thấp.

Tần số lấy mẫu chuẩn hóa  $f_s/B$  là hàm của tần số trung tâm chuẩn hóa  $f_0/B$  được minh họa ở hình 3.5, trong đó quan hệ giữa  $f_0$  và  $f_h$  là  $f_h = f_0 + B/2$ . Ta thấy rằng, tần số lấy mẫu khả dụng nằm trong khoảng  $2B \leq f_s \leq 4B$ . Tuy nhiên, trường hợp điển hình khi  $f_0 \gg B$ , thì tần số lấy mẫu được xác định theo định lý lấy mẫu thông dải là xấp xỉ bằng, nhưng được giới hạn dưới là  $2B$ .



Hình 3.5: Tần số lấy mẫu tín hiệu thông dải

### Lấy mẫu các tín hiệu đồng pha/vuông pha

Giả sử ta có một tín hiệu thông dải được biểu diễn ở dạng:

$$x(t) = A(t) \cos[2\pi f_c t + \phi(t)] \quad (3.19)$$

Các hàm  $A(t)$  và  $\phi(t)$  là đường bao và độ lệch pha của tín hiệu thông dải. Trong hầu hết các ứng dụng truyền thông cả  $A(t)$  và  $\phi(t)$  là tín hiệu *thông thấp* và có băng thông xấp xỉ bằng thông của tín hiệu mang tin. Sử dụng các tính chất lượng giác, tín hiệu thông dải được viết là:

$$x(t) = \underbrace{A(t) \cos \phi(t)}_{x_d(t)} \cos 2\pi f_c t - \underbrace{A(t) \sin \phi(t)}_{x_q(t)} \sin 2\pi f_c t \quad (3.20)$$

$$x(t) = x_d(t) \cos 2\pi f_c t - x_q(t) \sin 2\pi f_c t \quad (3.21)$$

Theo đó, các thành phần đồng pha  $x_d(t)$  và vuông pha  $x_q(t)$  được định nghĩa như sau:



$$x_d(t) = A(t) \cos \phi(t) \quad (3.22)$$

$$x_q(t) = A(t) \sin \phi(t) \quad (3.23)$$

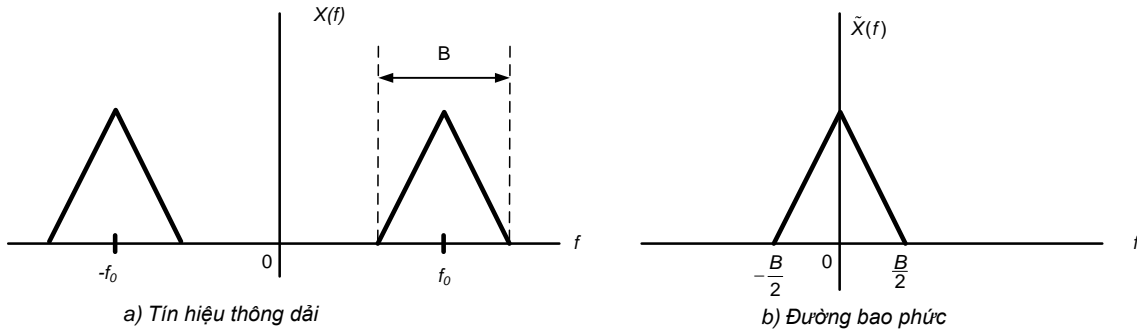
Vì  $A(t)$  và  $\phi(t)$  đều là tín hiệu thông thấp, nên  $x_d(t)$  và  $x_q(t)$  cũng là các tín hiệu thông thấp, vì vậy chúng phải được lấy mẫu theo định lý lấy mẫu thông thấp. Lưu ý rằng nếu biết trước  $x_d(t)$ ,  $x_q(t)$  và tần số sóng mang  $f_c$ , thì tín hiệu thông dải (3.21) có thể được khôi phục mà không có lỗi. Trình bày các tín hiệu *thông dải* theo các thành phần đồng pha và vuông pha sẽ được khảo sát chi tiết ở chương 4.

Biểu diễn tín hiệu thông dải trong miền tần số được cho ở hình 3.6(a), đường bao phức tương ứng được định nghĩa bởi:

$$\tilde{x}(t) = x_d(t) + jx_q(t) \quad (3.24)$$

Vì cả  $x_d(t)$  và  $x_q(t)$  đều là tín hiệu thông thấp, nên:

$$\tilde{X}(f) = X_d(f) + jX_q(f) \quad (3.25)$$



Hình 3.6: Tín hiệu thông dải và đường bao phức

Cũng là tín hiệu thông thấp như được minh họa ở hình 3.6(b). Trong hình 3.6 ta thấy rằng  $\tilde{X}(f)$ ,  $x_d(t)$  và  $x_q(t)$  đều là tín hiệu thông thấp. Vì vậy,  $x_d(t)$  và  $x_q(t)$  phải được lấy mẫu theo định lý lấy mẫu thông thấp. Vì tần số cao nhất trong  $x_d(t)$  và  $x_q(t)$  là  $B/2$ , nên tần số lấy mẫu nhỏ nhất cho mỗi thành phần này là  $B$ . Tuy nhiên, cả hai tín hiệu thông thấp  $x_d(t)$  và  $x_q(t)$  đều phải được lấy mẫu. Kết quả là, phải dùng tốc độ lấy mẫu lớn hơn  $2B$ . Vì vậy, việc lấy mẫu đường bao phức theo *định lý lấy mẫu thông thấp*, mang lại sử dụng tần số lấy mẫu đó để lấy mẫu tín hiệu thông dải giá trị thực theo định lý lấy mẫu thông dải cho trường hợp  $f_0 \gg B$ .

**Ví dụ 3.1:** Thấy rõ, có thể khôi phục tín hiệu thông dải  $x(t)$  mà không có lỗi nếu  $x_d(t)$  và  $x_q(t)$  được lấy mẫu theo định lý lấy mẫu thông thấp một cách thích hợp. Ưu điểm khi biểu diễn tín hiệu thông dải bằng thông thấp tương đương là rõ ràng. Ta biểu diễn tín hiệu FM bởi một tập các mẫu trong một giây. Giả sử, tần số sóng mang là 100 MHz (băng tần phát thanh FM)

tần số cao nhất trong tín hiệu mang tin là 15 kHz. Băng thông B của tín hiệu điều chế thường được lấy xấp xỉ theo nguyên tắc của Carson đó là:

$$B = 2(D+1)W \quad (3.26)$$

Giả sử tỉ lệ độ lệch  $D = 5$  nên ta có:

$$B = 2(5+1)15 = 180 \text{ kHz} \quad (3.27)$$

Theo đó, tần số cao nhất trong tín hiệu điều chế là  $f_c + B/2 = 100.000 + 90 = 100.090$  kHz. Vậy, theo định lý lấy mẫu thông thấp thì trong một giây tín hiệu phải có tối thiểu là  $2 * (f_c + B/2) = 200.180.000$  mẫu.

Ta giả sử, biểu diễn tín hiệu FM theo thành phần đồng pha và vuông pha. Băng thông của  $x_d(t)$  và  $x_q(t)$  đều là  $B/2 = 90$  kHz. Do vậy mỗi tín hiệu  $x_d(t)$  và  $x_q(t)$  cần ít nhất 180.000 mẫu để biểu diễn một giây tín hiệu. Điều này mang lại là 360.000 mẫu thông thấp để biểu diễn một giây dữ liệu. Nó sẽ tiết kiệm được:

$$\frac{200,180,000}{360,000} \approx 556 \quad (3.28)$$

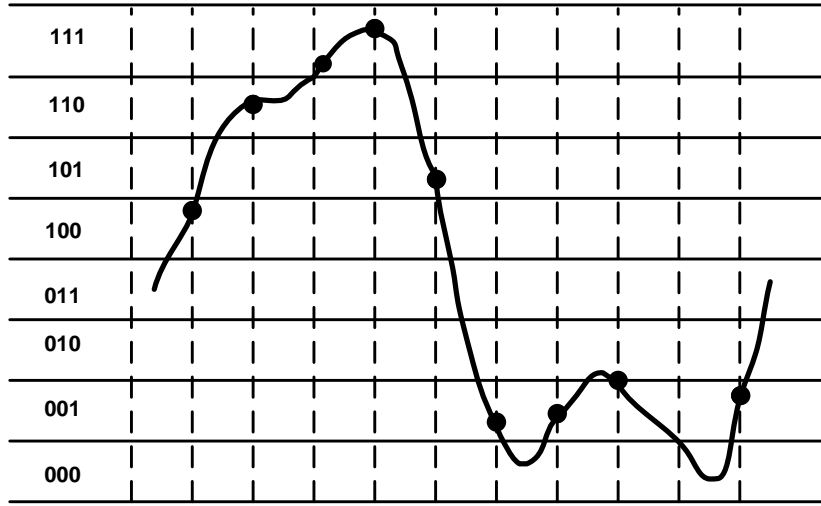
Do đó, giảm thời gian chạy của máy tính.

### 3.3. Lượng tử

Quá trình lượng tử và mã hóa được minh họa trong hình 3.7, cho thấy dạng sóng liên tục, tín hiệu mẫu được thể hiện bằng các dấu chấm đen, mỗi mẫu tín hiệu nằm trong một mức lượng tử cụ thể. Nếu lượng tử  $n$  mức và mỗi mức lượng tử được biểu diễn bởi một từ mã nhị phân  $b$  bit, thì ta có quan hệ

$$n = 2^b \quad (3.29)$$

Trong hình 3.7, mỗi một mức lượng tử được gán bằng một từ mã 3 bit ( $b = 3$  và  $n = 8$ ). Sau khi lượng tử hóa, các giá trị mẫu được biểu diễn bởi các từ mã ứng với mức lượng tử đó và số hóa dạng sóng tín hiệu bằng các từ mã này. Ví dụ, ba giá trị mẫu đầu tiên từ trái sang phải trong hình 3.7 được biểu diễn bằng chuỗi nhị phân 100110111.



Hình 3.7: Lượng tử và mã hóa

Từ lý thuyết lấy mẫu, nếu lấy mẫu các tín hiệu tương tự bằng thông hạn chế tại tần số lớn hơn tần số Nyquist thì khôi phục tín hiệu từ các mẫu mà không có lỗi. Vì vậy dưới các điều kiện này, hoạt động lấy mẫu là khả nghịch. Tuy nhiên, hoạt động lượng tử là không khả nghịch. Một khi các giá trị mẫu được lượng tử, chỉ có mức lượng tử được duy trì, và do vậy gây ra một lỗi ngẫu nhiên. Nếu ký hiệu  $x[k]$  là giá trị tín hiệu tại thời điểm lấy mẫu  $t = kT_s$ ,  $x_q[k]$  là giá trị được lượng tử hóa,  $e_q[k]$  là lỗi lượng tử, thì:

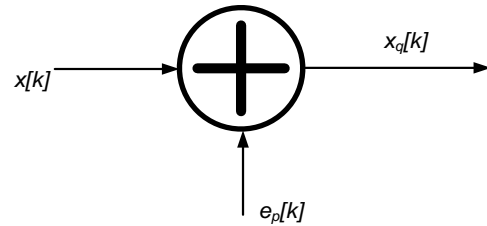
$$x_q[k] = x[k] + e_q[k] \quad (3.30)$$

Mô hình bộ lượng tử hóa cho (3.30) được minh họa ở hình 3.8. Nếu tín hiệu gốc không được hạn băng, thì tín hiệu số chứa cả lỗi lượng tử và lỗi chồng phổ.

Khi xét tỉ số tín hiệu trên tạp âm SNR, tạp âm được hiểu là do quá trình lượng tử hóa gây ra. Nếu ký hiệu  $E\{\}$  là kỳ vọng thống kê,  $N_q$  là công suất tạp âm do lượng tử hóa, thì tỉ số tín hiệu trên tạp âm do lượng tử hóa  $(SNR)_q$  được định nghĩa bởi:

$$(SNR)_q = \frac{S}{N_q} = \frac{E\{x^2[k]\}}{E\{e_q^2[k]\}} \quad (3.31)$$

Theo đó, để xác định  $(SNR)_q$ , cần phải biết hàm mật độ xác suất của lỗi  $e_q[k]$ . Hàm mật độ xác suất pdf của lỗi lượng tử là hàm của khuôn dạng được dùng để biểu diễn các con số trong máy tính. Có nhiều khuôn dạng được sử dụng. Loại phổ biến nhất là loại dấu phẩy động và dấu phẩy tĩnh.



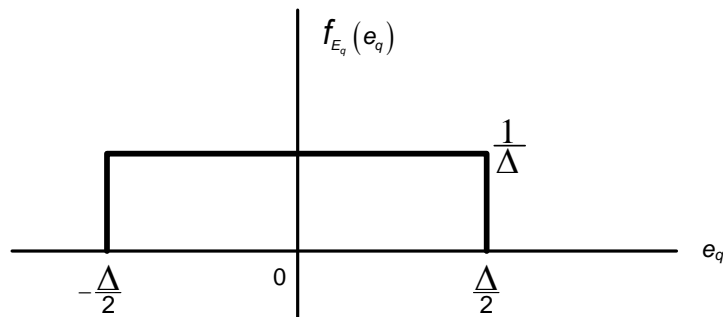
Hình 3.8: Mô hình hóa lỗi lượng tử

### Số học dấu phẩy tĩnh

Khi dùng máy tính số để mô phỏng thì các con số thường được trình bày ở dạng dấu phẩy động nhưng có nhiều lý do phải trình bày con số ở dạng dấu phẩy tĩnh (lỗi lượng tử). Một số lý giải cho việc dùng số học dấu phẩy tĩnh: (i) Minh họa các cơ chế cơ bản gây lỗi lượng tử hóa, các phép tính dấu phẩy tĩnh chạy nhanh hơn rất nhiều so với các phép tính dấu phẩy động; (ii) Tiêu thụ công suất của các vi xử lý dùng dấu phẩy tĩnh nhỏ hơn; (iii) Khi mô phỏng các thiết bị sử dụng dấu phẩy tĩnh. Chẳng hạn, các hệ thống truyền thông dựa trên phần mềm đang trở nên phổ biến (bởi lẽ tính khả lập lại cấu hình tương thích). Giá thành thiết bị được giảm thiểu khi dùng tính toán dấu phẩy tĩnh, hơn nữa các phép toán dấu phẩy tĩnh chạy nhanh hơn rất nhiều so với các phép toán dấu phẩy động. Cần lưu ý rằng, việc thiết kế các thiết bị dựa trên phần mềm thường bắt đầu bằng việc mô phỏng, nếu kết quả mô phỏng cho thấy thiết bị hoạt động đạt chỉ tiêu thiết kế, thì mã mô phỏng được nạp vào các thiết bị đó. Trong các ứng dụng như vậy, việc mô phỏng các thiết bị và thiết bị vật lý hợp nhất vào một quy mô lớn. Như đã đề cập, chỉ tiêu tốc độ, giá cả, tiêu thụ công suất dẫn đến thương mại hóa, hầu hết các thiết bị đều khai thác tính toán dấu phẩy tĩnh, mô phỏng là một công cụ quan trọng thiết kế và định lượng hiệu năng của thiết bị.

Nếu ký hiệu  $\Delta$  là độ rộng của mức lượng tử (xem hình 3.7) và giả thiết giá trị của mẫu tương ứng với một mức lượng tử nhận giá trị trung tâm của mức lượng tử đó thì lỗi lượng tử lớn nhất  $|e_{q[k]}|$  là  $\Delta/2$ . Nếu số mức lượng tử lớn thì từ mã tương ứng sẽ dài. Nếu tín hiệu thay đổi nhiều giữa các mẫu liên kề, thì khả năng một mẫu cho trước rơi vào bất kỳ một điểm nào đó trong mức lượng tử là như nhau. Khi này lỗi do lượng tử hóa được coi là phân bố đều và độc lập. Theo đó, hàm mật độ xác suất pdf của lỗi lượng tử là một hàm phân bố đều trong khoảng

$[-\Delta/2, \Delta/2]$  được minh họa trong hình 3.9. Nếu ký hiệu lỗi lượng tử của mẫu thứ  $k$  là  $e_q[k]$  thì ta có: trung bình của lỗi lượng tử; phương sai của lỗi lượng tử;  $(SNR)_q$  như sau:



Hình 3.9: Hàm mật độ xác suất giả định của lỗi lượng tử

Trung bình của lỗi lượng tử  $e_q[k]$  là:

$$E\{e_q[k]\} = \int_{-\Delta/2}^{\Delta/2} x \frac{1}{\Delta} dx = 0 \quad (3.32)$$

Lỗi lượng tử có trung bình 0.

Phương sai của lỗi lượng tử  $e_q[k]$  là:

$$E(e_q^2[k]) = \int_{-\Delta/2}^{\Delta/2} x^2 \frac{1}{\Delta} dx = \frac{\Delta^2}{12} \quad (3.33)$$

Tính tỉ số tín hiệu trên tạp âm do lượng tử hóa

Nếu giả thiết và ký hiệu: (i)  $D$  là dải động bộ lượng tử hóa (dải động là giá trị *đỉnh-đỉnh* của tín hiệu vào bộ lượng tử hóa); (ii)  $b$  là chiều dài của từ mã; (iii) nếu dùng số học nhị phân, thì có  $2^b$  mức lượng tử có thể có và dải động của nó được cho bởi:

$$D = 2^b \Delta \quad (3.34)$$

Do vậy:

$$\Delta = D 2^{-b} \quad (3.35)$$

và công suất tạp âm do lượng tử là:

$$N_q = E(e_q^2[k]) = \frac{D^2}{12} 2^{-2b} \quad (3.36)$$

Nếu ký hiệu  $S$  là công suất tín hiệu và  $(SNR)_q$  là tỉ số tín hiệu trên tạp âm do lượng tử hóa, thì ta có:

$$(SNR)_q = \frac{S}{\left(\frac{D^2}{12}\right) 2^{-2b}} = \frac{12S}{D^2} 2^{2b} \quad (3.37)$$

Giả sử tín hiệu có trung bình 0, quan hệ giữa giá trị  $S$  và  $D$  được cho bởi hệ số gợn sóng (*Crest Factor*) của tín hiệu cơ bản. Hệ số gợn sóng được định nghĩa là tỉ số của RMS (hay độ lệch chuẩn) của tín hiệu với giá trị đỉnh của tín hiệu đó. Để minh họa quan hệ này, giả sử tín hiệu cơ bản có dải rộng  $D$ , nằm trong dải  $\pm D$ . Vì công suất tín hiệu này là  $S$ , độ lệch chuẩn là  $\sqrt{S}$ , do vậy hệ số gợn sóng là:

$$F_c = \frac{\sqrt{S}}{D/2} = \frac{2\sqrt{S}}{D} \quad (3.38)$$

Thay (3.38) vào (3.37) ta có:

$$(SNR)_q = 3F_c^2 2^{2b} \quad (3.39)$$

Chuyển sang đơn vị dB là:

$$(SNR)_q = 4,7712 + 20 \log_{10} F_c + 6,0206b, \text{ dB} \quad (3.40)$$

Lưu ý rằng, các tín hiệu có hệ số gợn sóng lớn thì dễ tránh lỗi lượng tử hơn. Điều này là hợp logic bởi lẽ các tín hiệu có hệ số gợn sóng lớn có độ lệch chuẩn lớn, nghĩa là chúng được trải rộng trên các mức lượng tử hơn. Tuy vậy, chiều dài từ mã  $b$  có ảnh hưởng lên  $(SNR)_q$  nhất, tăng bit thêm vào chiều dài từ mã thì  $(SNR)_q$  tăng 6 dB.

### Số học dấu phẩy động

Khi mô phỏng trên máy tính sử dụng dấu phẩy động, khuôn dạng của các con số dấu phẩy động là  $\pm M * (\pm 10^E)$ , trong đó  $M$  là phần định trị và  $E$  là số mũ. Yêu cầu về độ chính xác, sử dụng các từ số 64-bit, 64 bit phải được phân bổ giữa phần định trị (*mantissa*) và số mũ. Việc phân bổ này có ảnh hưởng lớn đến kết quả tính toán. Tuy nhiên, việc ấn định này hầu như được tiêu chuẩn hóa, nhưng không phải tất cả. Tiêu chuẩn ANSI/IEEE đối với đại số dấu phẩy động định rõ 53 bit định trị và 11 bit số mũ. Matlab cung cấp một phương pháp đơn giản để xác định dù tiêu chuẩn IEEE có được thực hiện trên máy tính hay không bằng lệnh *isieee*.

Vì sẽ dùng Matlab để triển khai và giải thích các mô phỏng nên cần phải xét độ chính xác. Theo đó, tham số quan trọng nhất từ việc sử dụng dấu phẩy động là độ phân giải (sai khác giữa 1 và con số dấu phẩy động lớn nhất tiếp theo) là biến Matlab *eps*, số lớn nhất được trình bày (*realmax* trong Matlab) và số dương nhỏ nhất được trình bày (*realmin* trong Matlab). Chạy chương trình Matlab đơn giản được cho ở file **NVD3\_mparameters.m** trong Phụ lục 3A, nhận được các giá trị của một trong ba tham số quan trọng trên.

Chạy **NVD3\_mparameters** theo tiêu chuẩn dấu phẩy động IEEE cho kết quả sau:

```
>> NVD3_mparameters
```

Giá trị của **isieee** là: 1

Giá trị của **esp**: 2.22044604925031e-16

Giá trị của **realmax** là: 1.7976931348623e+308

Giá trị của **realmin** là: 2.22500738585072e-308

Lưu ý rằng, khi chạy chương trình mô phỏng các dấu chấm trong các kết quả cũng như biểu diễn con số là dấu phẩy của tiếng Việt và ngược lại.

Kết quả đầu tiên (*isieee* = 1) cho thấy máy tính thực hiện theo chuẩn ANSI/IEEE cho số dấu phẩy động. Kết quả tiếp theo là *eps*, là sai khác khả phân giải nhỏ nhất giữa hai số. Lưu ý, *eps* là  $2^{-52}$  (bit lớn nhất liên quan đến phần định trị giải thích cho bit dấu), nó minh họa cho quan hệ giữa *eps* và chiều dài từ. Ta thấy rằng đạt được độ chính xác lớn hơn 15 con số. Nó là giá trị của *eps* quan hệ chặt chẽ với độ rộng của mức lượng tử  $\Delta$  mà đã xét với số học dấu phẩy tĩnh. Lưu ý  $\pm realmax$  xác định dải động.

**Ví dụ 3.2:** Giả sử ta sử dụng số học dấu phẩy động theo chuẩn ANSI/IEEE, để tính giá trị của:

$$A = 1 - 0,4 - 0,3 - 0,2 - 0,1 \quad (3.41)$$

Hiển nhiên kết quả là bằng không. Tuy nhiên, thực hiện phép tính này trong Matlab như sau:

```
>> a = 1-0,4-0,3-0,2-0,1
a = -2.7756e-017
```

Ta thấy rằng lỗi do dấu phẩy động nhỏ và có thể bỏ qua trong hầu hết các ứng dụng. Tuy nhiên, do lỗi khác 0, nên cần phải nhớ rằng kết quả tính toán không hoàn toàn chính xác.

### 3.4. Tái tạo và nội suy

Tại đây, ta xét việc khôi phục tín hiệu liên tục từ một chuỗi các mẫu. Trong mô phỏng số, do chỉ xử lý các giá trị mẫu, nên không thể khôi phục hoàn toàn tín hiệu liên tục từ một tập các mẫu. Tuy nhiên, khi xét quá trình khôi phục dẫn đến nội suy, là một hoạt động quan trọng trong mô phỏng.

Ở dạng tổng quát, khôi phục là cho các mẫu qua bộ lọc thông thấp có đáp ứng xung kim  $h(t)$ . Theo đó, dạng sóng khôi phục  $x_r(t) = x_s(t) \otimes h(t)$ , trong đó  $\otimes$  ký hiệu cho tích chập. Từ (3.1) và (3.9) ta có:

$$x_s(t) = x(t) \sum_{k=-\infty}^{\infty} \delta(t - kT_s) = \sum_{k=-\infty}^{\infty} x(kT_s) \delta(t - kT_s) \quad (3.42)$$

Do vậy, dạng sóng khôi phục là:

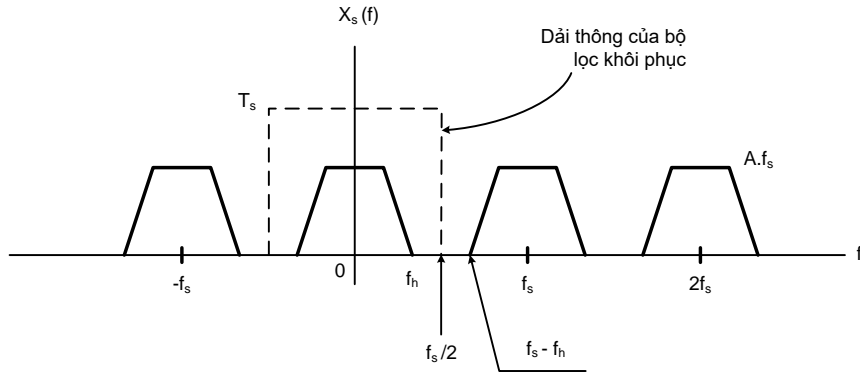
$$x_r(t) = \left[ \sum_{k=-\infty}^{\infty} x(kT_s) \delta(t - kT_s) \right] \otimes h(t) \quad (3.43)$$

$$x_r(t) = \sum_{k=-\infty}^{\infty} x(kT_s) h(t - kT_s) \quad (3.44)$$

Vấn đề là lựa chọn  $h(t)$  sao cho các kết quả phù hợp ở một mức độ phức tạp tính toán hợp lý.

#### 3.4.1. Tái tạo lý tưởng

Giả sử lấy mẫu một tín hiệu hạn chế băng tại tốc độ lớn hơn  $2f_h$ , thì có thể khôi phục tín hiệu bằng cách cho các mẫu qua bộ lọc thông thấp lý tưởng có độ rộng băng là  $f_s/2$  (xem hình 3.10). Nếu  $f_s > 2f_h$ , thì phổ được trung tâm tại  $f = \pm f_s$  không chồng lấn lên phổ được trung tâm tại  $f = 0$ . Tín hiệu ra của bộ lọc khôi phục là  $f_s X(f)$  hay trong miền thời gian là  $f_s x(t)$ , lấy tỉ lệ biên độ bởi  $1/f_s = T_s$  nhận được  $x(t)$ .



Hình 3.10: Bộ lọc khôi phục tín hiệu

Từ hình 3.10, đáp ứng xung kim của bộ lọc khôi phục là:

$$h(t) = T_s \int_{-f_s/2}^{f_s/2} e^{j2\pi ft} df \quad (3.45)$$

Trong đó tính đến hệ số tỉ lệ  $T_s$ . Vì vậy:

$$h(t) = T_s \frac{1}{j2\pi t} [e^{j\pi f_s t} - e^{-j\pi f_s t}] \quad (3.46)$$

$$h(t) = T_s \frac{1}{\pi t} \sin(\pi f_s t) = \text{sinc}(f_s t) \quad (3.47)$$

Thế vào (3.44) ta được:

$$x_r(t) = \sum_{k=-\infty}^{\infty} x(kT_s) \text{sinc}[f_s(t - kT_s)] \quad (3.48)$$

$$x_r(t) = \sum_{k=-\infty}^{\infty} x(kT_s) \text{sinc}\left(\frac{t}{T_s} - k\right) \quad (3.49)$$

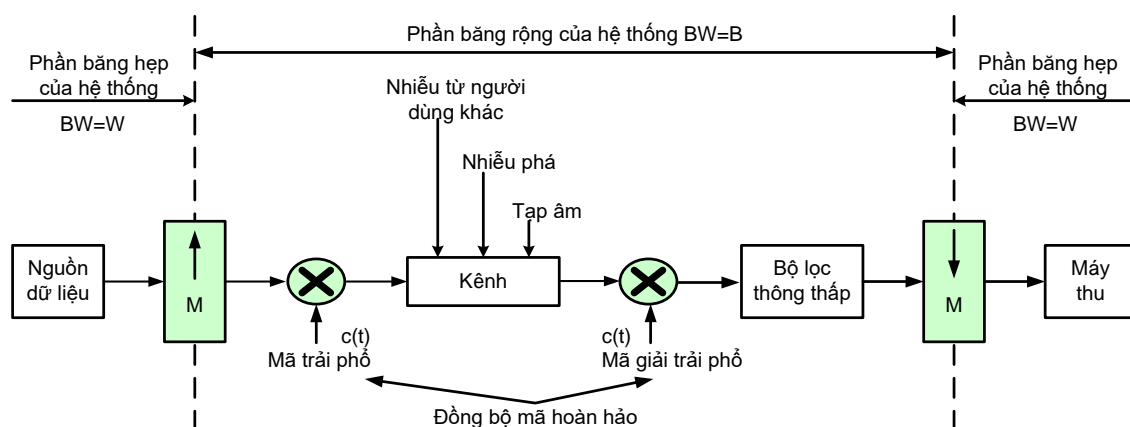
Lưu ý rằng, do giả định tín hiệu  $x(t)$  bị hạn chế băng và tần số lấy mẫu đủ lớn để tránh được lỗi chồng phổ, nên  $x_r(t) = x(t)$ . Vì vậy, ít nhất về mặt lý thuyết đạt được khôi phục hoàn hảo. Tuy nhiên, (3.49) có thể không bao giờ được dùng trong thực tế, vì hàm  $\text{sinc}(\cdot)$  là một hàm vô hạn về mặt phạm vi. Tuy vậy, phương trình (3.49) sẽ được dùng như là một cơ sở cho kỹ thuật nội suy thực tế ở phần sau.

### 3.4.2. Tăng mẫu và giảm mẫu

Tăng và giảm mẫu (upsampling và downsampling) được sử dụng trong quá trình mô phỏng nhiều hệ thống. Sự cần thiết được minh họa bằng ví dụ sau. Xét hệ thống trải phổ chuỗi trực tiếp hình 3.11. Khối nguồn dữ liệu tạo ra tín hiệu số liệu có phổ hẹp với băng tần là  $W$ . Tín hiệu số liệu được nhân với mã trải phổ băng rộng  $c(t)$ , mã này được biểu diễn bởi một chuỗi



nhị phân có tốc độ lớn hơn nhiều tốc độ số liệu. Tỉ số giữa tốc độ mã trái phổ và tốc độ số liệu được gọi là độ lợi xử lý của hệ thống. Việc nhân với mã trái phổ  $c(t)$  tạo ra một tín hiệu băng rộng có băng tần là  $B$ . Sự không hoàn hảo của kênh bao gồm nhiễu từ các người dùng khác, nhiễu phá (từ các hệ thống quân sự), tạp âm và một vài suy giảm khác như được minh họa trong hình 3.11. Tín hiệu đầu ra của kênh được nhân với mã giải trái phổ. Mã trái phổ nhận các giá trị  $\pm 1$ , nếu mã trái phổ và mã giải trái phổ là giống nhau (đồng bộ chính xác), thì khi nhân mã trái phổ và mã giải trái phổ cho ta  $c^2(t) = 1$ , do vậy các mã trái phổ và giải trái phổ sẽ không ảnh hưởng lên kênh. Lưu ý rằng, tín hiệu số liệu được nhân với  $c(t)$  hai lần và sự không hoàn hảo của kênh chỉ nhân với  $c(t)$  một lần. Vì vậy, tại đầu vào của bộ lọc thông thấp ngay sau khi nhân tín hiệu với mã giải trái phổ, tín hiệu số liệu lại là tín hiệu băng hẹp và các thành phần khác là băng rộng. Bộ lọc thông thấp sẽ tách lấy thành phần băng hẹp và đưa nó đến máy thu.



Hình 3.11: Minh họa tính hữu hiệu của việc dùng tăng và giảm mẫu

Thuộc tính quan trọng của hệ thống được cho ở hình 3.11 là xuất hiện cả hai loại tín hiệu băng hẹp và băng rộng. Khi  $B \gg W$ , lấy mẫu tín hiệu băng hẹp tại tần số lấy mẫu cho tín hiệu băng rộng sẽ không hiệu quả và kéo dài thời gian mô phỏng. Vì vậy, về lý tưởng mỗi tín hiệu được lấy mẫu tại tần số lấy mẫu phù hợp.

Do các tín hiệu có hai độ rộng băng tần khác nhau cùng tồn tại trong hệ thống, nên sử dụng hai tần số lấy mẫu. Do vậy, tần số lấy mẫu phải *tăng* tại biên giới giữa phần tín hiệu băng hẹp và băng rộng (đường đứt nét bên trái hình 3.11), và *giảm* tại biên giới giữa phần băng rộng và băng hẹp (đường đứt nét bên phải hình 3.11). Tăng tốc độ lấy mẫu được thực hiện bằng cách tăng mẫu theo sau là phép nội suy, ở đó các giá trị mẫu mới được nội suy từ các giá trị mẫu đã có. Việc giảm tốc độ lấy mẫu được thực hiện bằng cách loại bỏ mẫu có chọn lọc, ở đó các mẫu không cần thiết bị loại bỏ. Tăng mẫu được thể hiện bởi một khối có mũi tên hướng lên, và giảm mẫu được thể hiện bởi một khối có mũi tên hướng xuống. Tham số  $M$  thể hiện hệ số mà chu kỳ lấy mẫu giảm (tăng mẫu) hay tăng (giảm mẫu).

Phần sau, ta sẽ sử dụng  $T_s$  để thể hiện chu kỳ lấy mẫu trước khi quá trình tăng mẫu và giảm mẫu. Sau khi tăng mẫu hay giảm mẫu chu kỳ lấy mẫu sẽ được ký hiệu là  $T_u$  hay  $T_d$ . Tín hiệu trước khi tăng mẫu và giảm mẫu được ký hiệu là  $x(t)$  (không có chỉ số dưới cho  $x$ ), và tín hiệu sau khi tăng mẫu và giảm mẫu sẽ được ký hiệu là  $x_u(t)$  và  $x_d(t)$ .

### Tăng mẫu và nội suy

Tăng mẫu được minh họa trong hình 3.11 là quá trình thông qua nó tần số lấy mẫu được tăng lên. Vì tăng mẫu làm giảm chu kỳ lấy mẫu  $M$  lần, nên quan hệ giữa chu kỳ lấy mẫu mới  $T_u$  và chu kỳ lấy mẫu cũ  $T_s$  là  $T_u = T_s/M$ . Do vậy, với tín hiệu liên tục  $x(t)$ , thì quá trình tăng mẫu tạo ra các giá trị mẫu mới  $x(kT_u) = x(kT_s/M)$  từ các giá trị lấy mẫu cũ  $x(kT_s)$ . Ví dụ: giả sử ta tạo một tập các mẫu mới bằng cách nội suy tín hiệu khôi phục  $x_r(t)$  được cho bởi (3.49) tại các thời điểm  $t = nT_s / M$ . Thực hiện quá trình này ta có:

$$x_i(nT_u) = x(nT_s / M) = \sum_{k=-\infty}^{\infty} x(kT_s) \text{sinc}\left(\frac{n}{M} - k\right) \quad (3.50)$$

Đây là một bộ nội suy không thể thực hiện được vì hàm  $\text{sinc}(\cdot)$  mở rộng đến vô cùng. Cắt bớt hàm  $\text{sinc}(\cdot)$  ta có:

$$x_i(nT_u) = x(nT_s / M) \cong \sum_{k=-L}^L x(kT_s) \text{sinc}\left(\frac{n}{M} - k\right) \quad (3.51)$$

khả thi hơn, mặc dù không hoàn hảo. Hiển nhiên nếu  $L$  lớn sẽ giảm lỗi của nội suy. Tuy nhiên, vì mỗi tín hiệu được nội suy cần có  $2L+1$  mẫu nên việc tính toán sẽ không chấp nhận được khi  $L$  lớn. Do vậy, cần phải dung hòa giữa mức độ chính xác và khối lượng tính toán khi nghiên cứu mô phỏng. Lưu ý rằng, do hàm nhân quả phải được dùng để nội suy nên gây ra một lượng trễ là  $LT_s$ . Trễ này không gây ra lỗi trong mô phỏng nhưng cũng cần phải lưu ý đến sự tồn tại của nó.

Một bộ nội suy thực tế hơn, yêu cầu tính toán ít hơn so với bộ nội suy hàm  $\text{sinc}(\cdot)$  là bộ nội suy tuyến tính. Mặc dù, bộ nội suy tuyến tính đơn giản hơn bộ nội suy hàm  $\text{sinc}(\cdot)$  nhưng vẫn có thể sử dụng khi tín hiệu được lấy mẫu quá lớn. Đáp ứng xung kim của bộ nội suy tuyến tính được cho bởi:

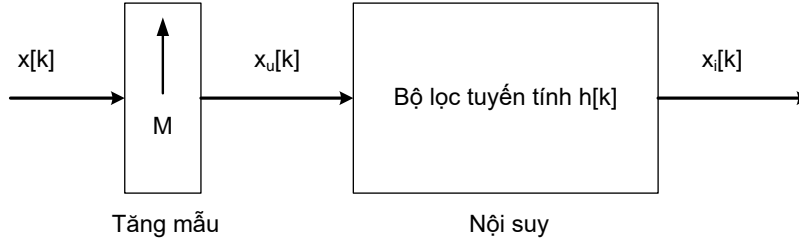
$$h[k] = \begin{cases} \frac{M-|k|}{M}, & k = 0, \pm 1, \pm 2, \dots, \pm(M-1) \\ 0, & \text{nếu khác} \end{cases} \quad (3.52)$$

Lưu ý rằng, có  $2M-1$  giá trị khác không của  $h[k]$ . Chương trình Matlab để triển khai  $h[k]$  được cho ở file **NVD3lininterp.m** dưới đây (có trong Phụ lục 3A).

Thực hiện tăng mẫu lên một tập các mẫu rời rạc là quá trình hai bước được minh họa ở hình 3.12. Trước hết ta tạo  $x_u[k]$  từ  $x[k]$  theo:

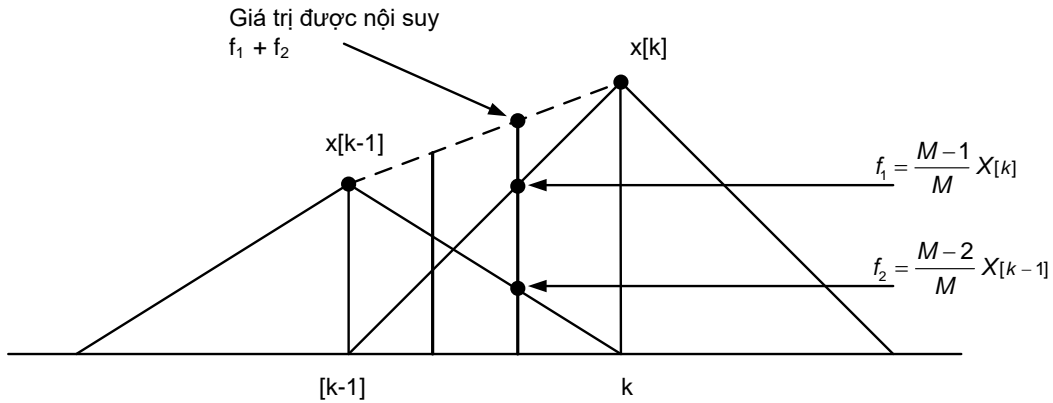
$$x_u[k] = \begin{cases} x[k/M], & k = 0, \pm M, \pm 2M, \dots, \\ 0, & \text{nếu khác} \end{cases} \quad (3.53)$$

nó được thực hiện bằng chương trình Matlab ở file **NVD3\_upsample.m** trong Phụ lục 3A.



Hình 3.12: Tăng mẫu và nội suy

Kết quả là, đặt  $M-1$  mẫu giá trị 0 giữa mỗi mẫu trong chuỗi gốc  $x[k]$ . Sau đó thực hiện nội suy bằng cách lấy tích chập  $x_u[k]$  với đáp ứng xung kim của bộ nội suy tuyến tính  $h[k]$ . Quá trình nội suy tuyến tính với  $M=3$  được minh họa ở hình 3.13. Lưu ý rằng, chỉ cần sử dụng 2 mẫu trong quá trình tăng mẫu. Trễ cần thiết là  $T_s$ . Như được minh họa ở hình 3.13, tìm được giá trị nội suy bằng cách lấy tổng các thành phần từ  $x[k]$  và  $x[k-1]$ , đó là  $\frac{M-1}{M}x[k]$  và  $\frac{M-2}{2}x[k-1]$ . Vì vậy, với  $M=3$ , giá trị nội suy là:



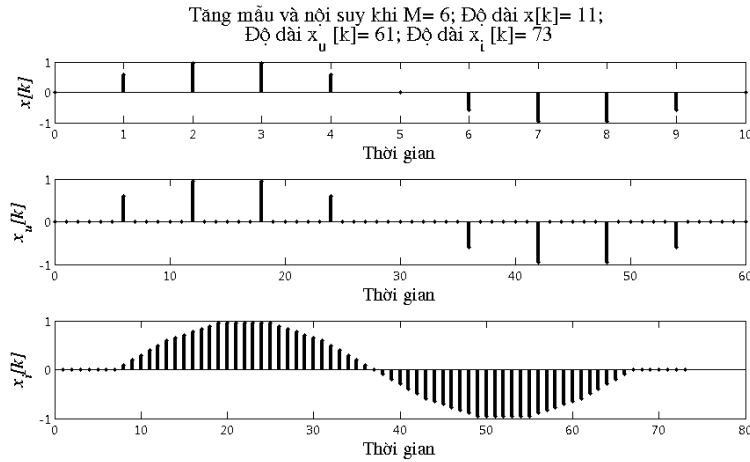
Hình 3.13: Minh họa quá trình nội suy

$$\frac{2}{3}x[k] + \frac{1}{3}x[k-1]$$

Vì chỉ có 2 mẫu được sử dụng trong quá trình nội suy, nên nội suy tuyến tính rất nhanh.

**Ví dụ 3.3:** Để minh họa quá trình tăng mẫu và nội suy, ta xét nội suy các mẫu của sóng hình sin. Các mẫu cơ bản được cho ở phần trên của hình 3.14 là  $x[k]$ . Tăng mẫu với  $M=6$

nhận được các giá trị mẫu  $x_u[k]$ . Nội suy tuyến tính với  $M = 6$  nhận được chuỗi các mẫu  $x_i[k]$ . Lưu ý trễ  $T_s$ . Chương trình Matlab cho bài toán này được cho ở file **NVD3\_upsampex.m** trong Phụ lục 3A, và kết quả mô phỏng điển hình được cho ở hình 3.14.



Hình 3.14: Nội suy và tăng mẫu được sử dụng trong ví dụ 3.3

Thấy rõ rằng, tăng mẫu và giảm mẫu bao gồm một lượng lớn mào đầu. Nếu hệ số tăng mẫu  $M$  là vừa phải, 2 hay 3, thì tốt nhất là triển khai mô phỏng bằng cách dùng một tần số lấy mẫu và do vậy xảy ra việc lấy mẫu quá nhiều không cần thiết đối với các tín hiệu băng hẹp. Tuy nhiên, nếu sai khác nhau giữa  $B$  và  $W$  lớn, thì hiệu quả hơn hết là sử dụng nhiều tần số lấy mẫu trong mô phỏng và lấy mẫu mỗi tín hiệu ở một tần số lấy mẫu phù hợp.

### Giảm mẫu

Giảm mẫu là quá trình thông qua nó tần số lấy mẫu được giảm bằng cách thay một khối  $M$  mẫu bởi một mẫu. Theo đó, thực hiện giảm mẫu đơn giản hơn so với tăng mẫu rất nhiều. Thể hiện chức năng đối với các mẫu tại đầu ra của bộ giảm mẫu bằng cách thừa nhận rằng quá trình giảm mẫu làm tăng chu kỳ lấy mẫu lên  $M$  lần. Vì vậy, các mẫu ở đầu ra của bộ giảm mẫu là  $x_d(kT_d) = x(kMT_s)$ . Các giá trị lấy mẫu là

$$x_d[k] = x[kM] \quad (3.54)$$

Tuy nhiên, cần xét cẩn thận để đảm bảo tín hiệu được lấy mẫu thấp không gây ra chồng phổ.

### 3.5. Tần số lấy mẫu trong mô phỏng

Quyết định cơ bản phải được thực hiện trong quá trình triển khai mô phỏng là lựa chọn tần số lấy mẫu. Đối với hệ thống tuyến tính không hồi tiếp, ta phải chọn tần số lấy mẫu sao cho lỗi chồng phổ ở mức cho phép, nó phụ thuộc vào mật độ phổ công suất của dạng xung tín hiệu cơ bản. Vì vậy, ta xét mô hình để biểu diễn tín hiệu băng tần cơ sở, được dùng để truyền dẫn số liệu và để triển khai các kỹ thuật tính mật độ phổ công suất. Vì các dạng xung có vai trò quan trọng để chọn tần số lấy mẫu phù hợp, nên ta xét vấn đề này chi tiết hơn.

Từ việc nghiên cứu lấy mẫu, cho thấy để loại bỏ hoàn toàn lỗi chồng phổ cần có tần số lấy mẫu vô cùng lớn dẫn đến không thực tế. Ngoài ra, khi tần số lấy mẫu tăng, phải xử lý nhiều mẫu hơn cho mỗi ký hiệu dữ liệu, làm tăng thời gian mô phỏng. Vì không thể loại bỏ được hoàn toàn lỗi chồng phổ trong thực tế, nên ta phải chọn tần số lấy mẫu sao cho dung hòa giữa lỗi chồng phổ và thời gian mô phỏng. Tất nhiên, tần số lấy mẫu phải được chọn sao cho bỏ qua được lỗi chồng phổ khi so sánh với sự suy giảm hệ thống đang được xét bằng mô phỏng.

### 3.5.1. Khai triển chung

Mô hình chung cho tín hiệu phát trong hệ thống truyền tham số là:

$$x(t) = A \sum_{k=0}^{\infty} a_k p(t - kT - \Delta) \quad (3.55)$$

Trong đó:  $\dots, a_{-2}, a_{-1}, a_0, a_1, a_2, \dots, a_k, \dots$  là chuỗi các biến ngẫu nhiên biểu diễn số liệu. Giá trị  $a_k$  thường nhận giá trị +1 hay -1 trong hệ thống số nhị phân,  $p(t)$  là hàm định dạng xung.  $T$  là chu kỳ ký hiệu,  $\Delta$  là biến ngẫu nhiên phân bố đều trong khoảng chu kỳ lấy mẫu. Tham số  $A$  là một hằng số tỉ lệ để xác lập công suất của tín hiệu phát. Bằng cách kết hợp tham số này ta có thể lấy tỉ số hàm định dạng xung sao cho giá trị đỉnh là đơn vị. Giả sử  $E\{a_k\} = 0$  và  $E\{a_k a_{k+m}\} = R_m$  là giá trị trung bình và tự tương quan của chuỗi xung số liệu.

Để thấy rằng hàm tự tương quan của tín hiệu phát được cho bởi:

$$R_{xx}(\tau) = A^2 \sum_{m=-\infty}^{\infty} R_m r(\tau - mT) \quad (3.56)$$

Trong đó:

$$r(\tau) = \frac{1}{T} \int_{-\infty}^{\infty} p(t) p(t + \tau) dt \quad (3.57)$$

Tần số lấy mẫu được xác định từ mật độ phổ công suất PSD của tín hiệu phát. Áp dụng định lý Wiener-Khintchine vào (3.56) ta có:

$$S_x(f) = A^2 \int_{-\infty}^{\infty} \left( \sum_{m=-\infty}^{\infty} R_m r(\tau - mT) \right) e^{-j2\pi f\tau} d\tau \quad (3.58)$$

$$S_x(f) = A^2 \sum_{m=-\infty}^{\infty} R_m \int_{-\infty}^{\infty} r(\tau - mT) e^{-j2\pi f\tau} d\tau \quad (3.59)$$

Ta biến đổi về dạng dễ tính toán hơn.

Trước tiên đổi biến số  $\alpha = \tau - mT$  vào (3.59). Ta có:

$$S_x(f) = A^2 \sum_{m=-\infty}^{\infty} R_m \int_{-\infty}^{\infty} r(\alpha) e^{-j2\pi f(\alpha + mT)} d\alpha \quad (3.60)$$

Ký hiệu:

$$S_r(f) = \int_{-\infty}^{\infty} r(\alpha) e^{-j2\pi f\alpha} d\alpha \quad (3.61)$$

Ta có:

$$S_x(f) = A^2 \sum_{m=-\infty}^{\infty} R_m S_r(f) \cdot e^{-j2\pi fmT} \quad (3.62)$$

Ta phải xác định  $S_r(f)$

Biến đổi Fourier lên (3.57) ta có:

$$S_r(f) = \int_{-\infty}^{\infty} \left( \frac{1}{T} \int_{-\infty}^{\infty} p(t) p(t+\alpha) dt \right) e^{-j2\pi f\alpha} d\alpha \quad (3.63)$$

Đổi biến  $\beta = t + \alpha$  thì (3.63) có thể biểu diễn ở dạng như sau:

$$S_r(f) = \frac{1}{T} \left( \int_{-\infty}^{\infty} p(t) \cdot e^{j2\pi ft} dt \right) \cdot \left( \int_{-\infty}^{\infty} p(\beta) \cdot e^{-j2\pi f\beta} d\beta \right) \quad (3.64)$$

Thành phần thứ hai trong (3.64) là biến đổi Fourier của hàm dạng xung  $p(t)$  và thành phần đầu tiên là liên hợp phức của thành phần thứ hai. Và ta có:

$$S_r(f) = \frac{|P(f)|^2}{T} = \frac{G(f)}{T} \quad (3.65)$$

Trong đó  $G(f)$  là mật độ phổ năng lượng của hàm định dạng xung  $p(t)$ . Thế (3.65) vào (3.62) cho ta kết quả tổng quát như sau:

$$S_x(f) = A^2 \frac{G(f)}{T} \sum_{m=-\infty}^{\infty} R_m e^{-j2\pi fmT} \quad (3.66)$$

Trong nhiều ứng dụng các ký hiệu số liệu có thể coi là độc lập. Giả thiết trên dẫn đến đơn giản hóa rất nhiều.

### 3.5.2. Ký hiệu dữ liệu độc lập

Nếu các ký hiệu dữ liệu  $\{a_k\}$  là độc lập, thì hàm tự tương quan trở thành:

$$\begin{aligned} R_m &= E\{a_k a_{k+m}\} = E\{a_k\} E\{a_{k+m}\} \\ &= \alpha_k^2 \delta[m] = \begin{cases} \alpha_k^2, & m = 0 \\ 0, & \text{nếu khác} \end{cases} \end{aligned} \quad (3.67)$$

Khi đó, mật độ phổ công suất PSD của  $x(t)$  được định nghĩa bởi (3.66) trở nên rất đơn giản:

$$S_x(f) = A^2 \frac{\alpha_k^2 G(f)}{T} \quad (3.68)$$

Nếu các ký hiệu số liệu  $a_k = \pm 1 \forall k \Rightarrow a_k^2 = 1$  và:

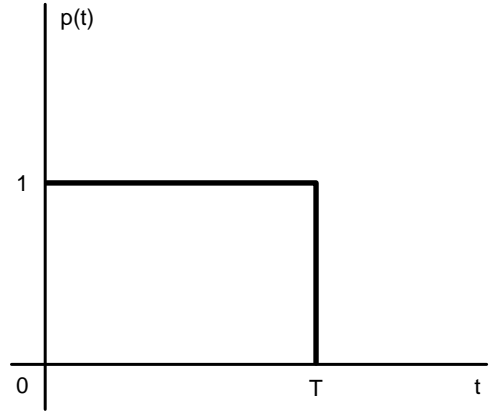
$$S_x(f) = A^2 \frac{G(f)}{T} \quad (3.69)$$

không phụ thuộc vào số liệu. Trường hợp các ký hiệu dữ liệu không độc lập, phải xác định hàm tự tương quan  $R_m$  và (3.66) phải được tính toán từng phần một.

**Ví dụ 3.4:** Xét dạng xung chữ nhật được cho ở hình 3.15. Theo hình 3.15 ta có:

$$P(f) = \int_0^T e^{-j2\pi ft} dt = \frac{1}{j2\pi f} [1 - e^{-j2\pi fT}] \quad (3.70)$$

$$\begin{aligned} P(f) &= \frac{1}{j2\pi f} [e^{j\pi fT} - e^{-j\pi fT}] \cdot e^{-j\pi fT} \\ &= \frac{\sin(\pi fT)}{\pi f} e^{-j\pi fT} \end{aligned} \quad (3.71)$$



Hình 3.15: Dạng xung chữ nhật

Hoặc ở dạng hàm  $\text{sinc}(\cdot)$ :

$$P(f) = T \text{sinc}(fT) \cdot e^{-j\pi fT} \quad (3.72)$$

Do vậy:  $G(f) = |P(f)|^2 = T^2 \text{sinc}^2(fT)$

Thế vào (3.69) ta có mật độ phổ công suất của  $x(t)$  là:

$$S_x(f) = A^2 T \text{sinc}^2(fT) \quad (3.73)$$

Công suất phát từ (3.69):

$$P = \int_{-\infty}^{\infty} S_x(f) df = A^2 \frac{1}{T} \int_{-\infty}^{\infty} G(f) df \quad (3.74)$$

Theo định lý Parseval's và hình 3.15 ta có:

$$\int_{-\infty}^{\infty} G(f) df = \int_{-\infty}^{\infty} |P(f)|^2 df = \int_{-\infty}^{\infty} p^2(t) dt = T \quad (3.75)$$

Thế vào (3.74) ta có:

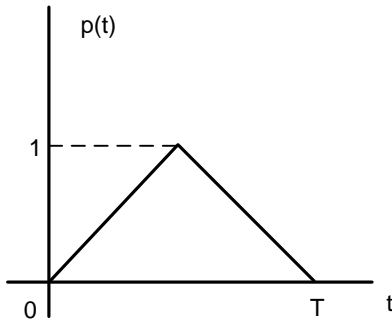
$$P = A^2 \quad (3.76)$$

Kết quả đơn giản này xuất phát từ thực tế là  $p(t)$  là một xung đơn vị. Do vậy,  $\sum a_k p(t - kT - \Delta)$  có công suất đơn vị. Phép nhân với  $A$  như trong (3.55) chỉ là lấy tỉ lệ công suất bởi  $A^2$ . Với các dạng xung khác quan hệ giữa công suất và  $A$  phải được tính toán theo kỹ

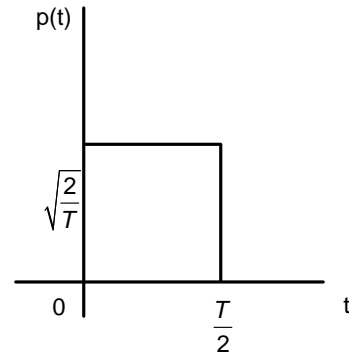
thuật đã minh họa ở trên. Lưu ý rằng chuỗi số liệu  $\{a_k\}$  là quá trình công suất (phương sai) đơn vị.

**Ví dụ 3.5:** Dạng xung được mô tả trong hình 3.16(a). Dạng xung cơ bản  $p(t)$  được biểu diễn dưới dạng  $p_I(t) \otimes p_I(t)$  trong đó  $\otimes$  là tích chập, và  $p_I(t)$  được mô tả trong hình 3.16(b). Lấy biến đổi Fourier của  $p_I(t)$  ta có:

$$P_I(f) = \sqrt{\frac{T}{2}} \operatorname{sinc}\left(\frac{T}{2}f\right) \cdot e^{-j\pi T/2} \quad (3.77)$$



a) Dạng xung tam giác



b) Dạng xung cơ sở để lấy tích chập

Hình 3.16: Dạng xung tam giác

Vì phép tích chập trong miền thời gian tương đương phép nhân trong miền tần số, nên:

$$|P(f)| = |P_I(f)|^2 = \frac{T}{2} \operatorname{sinc}^2\left(\frac{T}{2}f\right) \quad (3.78)$$

Do vậy:

$$G(f) = |P(f)|^2 = \frac{T^2}{4} \operatorname{sinc}^4\left(\frac{T}{2}f\right) \quad (3.79)$$

Thay vào (3.69) ta có:

$$S_x(f) = \frac{A^2 T}{4} \operatorname{sinc}^4\left(\frac{T}{2}f\right) \quad (3.80)$$

Kết quả này sẽ được sử dụng ở chương 4.

### 3.5.3. Tần số lấy mẫu trong mô phỏng

Trở lại vấn đề tần số lấy mẫu mô phỏng dạng xung bằng cách xét SNR của quá trình lấy mẫu trong đó công suất tạp âm do chồng phổ. Với mục đích chọn tần số lấy mẫu sao cho có thể bỏ qua lỗi chồng phổ so với suy giảm chất lượng hệ thống được mô phỏng. Có thể thấy rằng tần số lấy mẫu phụ thuộc vào dạng sóng mô hình mô phỏng.

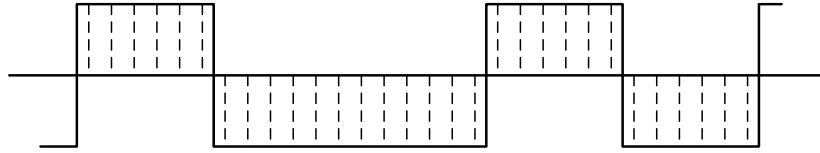


Xét dạng sóng được cho bởi (3.55) có dạng xung chữ nhật được minh họa ở hình 3.15, được lấy mẫu như trong hình 3.17.

### Xác định mật độ phổ công suất của tín hiệu

Trong hình 3.17, tần số lấy mẫu gấp 6 lần tốc độ ký hiệu. Mật độ phổ công suất của chuỗi ký hiệu được cho bởi (3.73). Kết hợp kết quả này với (3.18) ta có:

$$S_{X_s}(f) = f_s^2 \sum_{n=-\infty}^{\infty} A^2 T \sin^2 \left[ (f - nf_s)T \right] \quad (3.81)$$



Hình 3.17: Chuỗi xung chữ nhật được lấy mẫu tại 6 mẫu/ký hiệu

Lấy mẫu chuỗi số liệu tại tốc độ  $m$  mẫu trên một ký hiệu ( $f_s = m/T$ ) ta có:

$$S_{X_s}(f) = f_s^2 \sum_{n=-\infty}^{\infty} A^2 T \sin^2 \left[ \left( f - \frac{nm}{T} \right) T \right] \quad (3.82)$$

$$S_{X_s}(f) = f_s^2 \sum_{n=-\infty}^{\infty} A^2 T \sin^2 [fT - nm] \quad (3.83)$$

### Tính tỉ số tín hiệu trên tạp âm do chồng phổ

Tỉ số tín hiệu trên tạp âm do chồng phổ  $(SNR)_a = S/N_a$  trong đó:

Công suất tín hiệu là:

$$S = \int_{-f_s/2}^{f_s/2} f_s^2 A^2 T \sin^2 (fT) df = 2 f_s^2 A^2 T \int_0^{f_s/2} \sin^2 (fT) df \quad (3.84)$$

Công suất tạp âm do chồng phổ là:

$$\begin{aligned} N_a &= \int_{-f_s/2}^{f_s/2} f_s^2 \sum_{\substack{n=-\infty \\ n \neq 0}}^{\infty} A^2 T \sin^2 (fT - nm) df \\ &= 2 f_s^2 A^2 T \sum_{\substack{n=-\infty \\ n \neq 0}}^{\infty} \int_0^{f_s/2} \sin^2 (fT - nm) df \end{aligned} \quad (3.85)$$

Lưu ý rằng, PSD là hàm chẵn của tần số. Công suất tín hiệu được xác định bởi tích phân thành phần  $n = 0$  trong (3.83) trên dải tần mô phỏng  $|f| < f_s/2$ . Công suất tạp âm do chồng phổ là công suất từ tất cả các tần số ( $n \neq 0$ ) trong dải tần mô phỏng. Do vậy, ta tìm được tạp âm do chồng phổ bằng cách lấy tích phân trên tất cả các thành phần trong (3.83) trừ thành phần  $n = 0$ .

Thấy rõ trong hình 3.18, được vẽ với  $m = 6$ . Hình 3.18 minh họa phần tần số dương của thành phần  $n = 0$  trong (3.83) trên dải  $0 < f < f_s$ . Cho thấy, dịch phở đối với  $n = \pm 1$  và  $n = 2$ .

*Bước tiếp theo* trong quá trình xác định  $(SNR)_a$ , phải chỉ ra rằng cả  $S$  và  $N_a$  đều có thể xác định được bằng cách chỉ sử dụng thành phần  $n = 0$  trong (3.83). Các *búp* của hàm  $\text{sinc}(\cdot)$  đều có độ rộng  $1/T$ , được minh họa và đánh số trong hình 3.18. Lưu ý rằng với  $m = 6$  và  $n = 0$ , các búp 1, 2 và 3 rơi vào trong khoảng  $0 < f < f_s/2$  vì vậy thể hiện công suất tín hiệu. Các búp 4, 5 và 6 rơi vào trong khoảng  $0 < f < f_s/2$  với thành phần  $n = 1$  và do vậy nó thể hiện nhiễu chồng phở. Tương tự như vậy, các búp 7, 8 và 9 từ thành phần  $n = -1$  trong (3.83) và búp 10, 11 và 12 từ thành phần  $n = 2$  trong (3.83). Tiếp tục như vậy ta có:

$$\sum_{\substack{n=-\infty \\ n \neq 0}}^{\infty} \int_0^{f_s/2} \text{sinc}^2(fT - nm) df = \int_{f_s/2}^{\infty} \text{sinc}^2(fT - nm) df \quad (3.86)$$

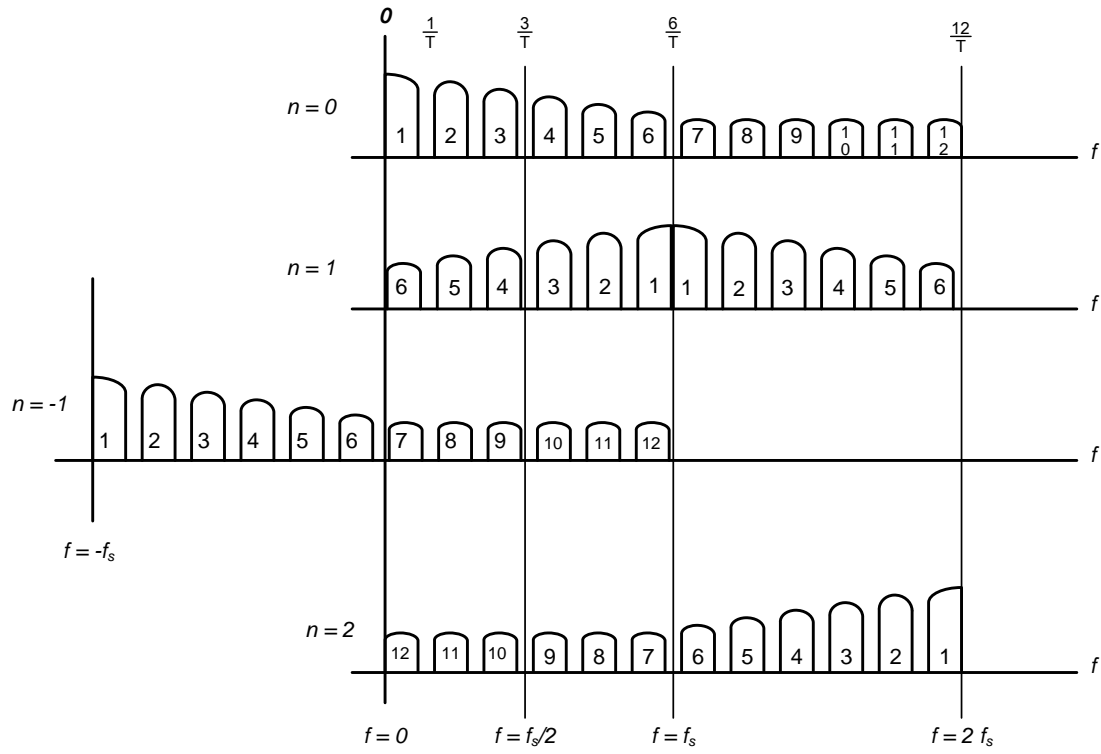
Vì vậy:

$$(SNR)_a = \frac{S}{N_a} = \frac{\int_0^{f_s/2} \text{sinc}^2(fT) df}{\int_{f_s/2}^{\infty} \text{sinc}^2(fT) df} \quad (3.87)$$

Bằng cách *so sánh* các ví dụ 3.4 và 3.5 thấy rõ dạng của hàm bị tích phân sẽ khác nhau phụ thuộc vào dạng xung.

Ta sẽ thường xuyên tìm xem có cần thiết sử dụng tích phân số để định lượng  $(SNR)_a$ . Để thực hiện điều này, quá trình lấy mẫu thứ hai được thực hiện trong đó biến tần số liên tục  $f$  được lấy mẫu ở các điểm  $f = jf_l$ . Để chính xác thì  $f_l \ll 1/T$  do vậy rất nhiều mẫu được lấy ở mỗi búp của hàm  $\text{sinc}(\cdot)$ . Theo đó, lấy mẫu tần số cho phép tích phân trong (3.87) được thay thế bởi lấy tổng. Để thỏa mãn  $f_l \ll 1/T$  đặt  $f_l = 1/(kT)$  trong đó  $k$  phải lớn để lỗi do tích phân số là nhỏ. Với  $f = jf_l$  và  $f_l = 1/(kT)$  ta có:

$$fT = \frac{j}{k} \quad (3.88)$$



Hình 3.18: Phổ và sự dịch chuyển phổ khi  $n = \pm 1$  &  $n = \pm 2$

Bước tiếp theo là tính toán tần số  $f_s/2$  theo các tham số rời rạc  $k$  và  $m$ . Từ hình 3.18 cho thấy, với  $m$  mẫu trên một ký hiệu, thì tần số  $f_s/2$  là  $m/(2T)$ . Vì lấy  $k$  mẫu cho khoảng tần số có độ rộng  $1/T$ , nên tần số tương ứng với chỉ số  $km/2$ . Sử dụng (3.88) và  $f_s/2$  tương ứng  $km/2$  trong (3.87) ta có:

$$(SNR)_a \cong \frac{\sum_{j=0}^{km/2} \sin^2(j/k)}{\sum_{j=km/2}^{\infty} \sin^2(j/k)} \quad (3.89)$$

Có được xấp xỉ trên là vì sử dụng tích phân số để lấy xấp xỉ đến giá trị thực của tích phân.

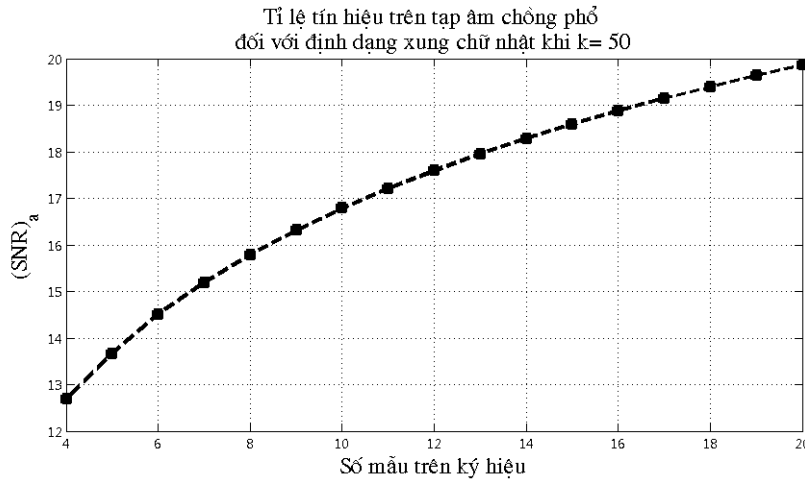
Chương trình Matlab để ước tính (3.89) được cho bởi file **NVD3\_sna.m** trong Phụ lục 3A.

Lưu ý rằng, lấy 50 mẫu tần số cho mỗi búp của hàm  $\text{sinc}(\cdot)$  và có tổng số 50000 mẫu tần được lấy. Vì vậy, tổng ở mẫu số của (3.89) có đến 1000 búp của hàm  $\text{sinc}(\cdot)$ , sau khoảng này PSD được cho là có thể bỏ qua. Có thể kiểm tra giả thiết này bằng cách làm thí nghiệm với tham số **nsamp**.

Chạy chương trình mô phỏng **NVD3\_sna.m**, kết quả được minh họa ở 3.19. Lưu ý rằng,  $(SNR)_a$  là nhỏ hơn 17 dB với  $m = 10$  mẫu trên một ký hiệu và  $(SNR)_a$  sẽ tăng khi  $m$  tăng. Tuy nhiên, ảnh hưởng lên  $(SNR)_a$  giảm khi  $m$  tăng. Cũng lưu ý rằng PSD của tín hiệu mẫu giảm  $1/f^2$  với dạng xung hình chữ nhật. Ví dụ 3.5 cho thấy, PSD của tín hiệu mẫu giảm  $1/f^4$  với dạng

xung hình tam giác. Vì vậy, với mỗi giá trị cho trước của  $m$ , thì giá trị  $(SNR)_a$  đối với dạng xung tam giác lớn hơn  $(SNR)_a$  đối với dạng xung hình chữ nhật. Dạng xung hình chữ nhật là trường hợp tồi nhất.

Trong hệ thống truyền thông thực tế, dạng xung  $p(t)$  được chọn để có được hiệu quả băng thông yêu cầu. Hiệu quả băng tần cao cho thấy phổ của  $x(t)$  được cho bởi (3.55) là tập trung xung quanh thành phần  $f = 0$ . Do đó, các tín hiệu có hiệu quả băng thông cao cần có giá trị  $m$  nhỏ đối với mỗi giá trị  $(SNR)_a$  đã cho.



Hình 3.19: Tỉ số tín hiệu trên tạp âm chồng phổ đối với dạng xung chữ nhật

### 3.6. Điều chế xung mã PCM

#### 3.6.1 Lượng tử hóa

Ở dạng tổng quát, các sơ đồ lượng tử hoá có thể được phân loại thành hai loại sau: (i) sơ đồ lượng tử hoá vô hướng, trong đó lại được phân thành lượng tử hoá đều và không đều; (ii) sơ đồ lượng tử hoá véc tơ. Trong lượng tử hoá vô hướng, mỗi đầu ra của nguồn tin được lượng tử hoá riêng rẽ, trong khi đó lượng tử hoá véc tơ, các khối đầu ra của nguồn tin được lượng tử hoá.

Trong lượng tử hoá đều, các vùng được chọn có cùng chiều dài, còn ở lượng tử hoá không đều, thì cho phép sử dụng các vùng có chiều dài có chiều dài khác nhau. Ở dạng tổng quát, các bộ lượng tử hoá không đều tốt hơn các bộ lượng tử hoá đều. Ở đây ta chỉ xét lượng tử hoá vô hướng mà không xét cho lượng tử hoá véc tơ.

Trong lượng tử hoá vô hướng, dải giá trị của biến ngẫu nhiên  $X$  được chia thành  $N$  vùng  $\mathcal{R}_i$  với  $1 \leq i \leq N$  không chồng chất nhau, được gọi là các khoảng lượng tử hoá, và trong mỗi vùng ta chọn một điểm được gọi là một mức lượng tử. Khi đó, mọi giá trị của biến ngẫu nhiên mà rơi vào vùng  $\mathcal{R}_i$  được lượng tử thành mức lượng tử thứ  $i$ , được ký hiệu là  $\hat{x}_i$ . Nghĩa là:

$$x \in \mathcal{R}_i \Leftrightarrow Q(x) = \hat{x}_i \quad (3.90)$$

Trong đó:

$$\hat{x}_i \in \mathfrak{R}_i \quad (3.91)$$

Hiển nhiên, lượng tử hoá kiểu này tạo ra một **sai số bình phương trung bình MSE** là  $E[(x - \hat{x}_i)^2]$ . Vì vậy, sai số lượng tử bình phương trung bình được cho bởi

$$D = \sum_{i=1}^N \underbrace{\int_{\mathfrak{R}_i} \underbrace{(x - \hat{x}_i)^2}_{\text{sai số}} f_X(x) dx}_{\text{lấy bình phương}} \quad (3.92)$$

lấy trung bình của biến ngẫu nhiên  $(x - \hat{x}_i)^2$   
có pdf là  $f_X(x)$  trên vùng  $\mathfrak{R}_i$

Trong đó  $f_X(x)$  ký hiệu cho hàm mật độ xác suất pdf của biến ngẫu nhiên của nguồn tin. Tỷ số tín hiệu trên tạp âm lượng tử SQNR (Signal-to-Quantization-Noise Ratio) được định nghĩa là:

$$SQNR = 10 \log_{10} \frac{E[X^2]}{D}, \text{ dB}$$

### Lượng tử hoá đều

Trong quá trình lượng tử hoá đồng đều, tất cả các vùng lượng tử hóa ngoại trừ vùng đầu tiên và vùng cuối cùng, tức là vùng  $\mathfrak{R}_1$  và  $\mathfrak{R}_N$  đều có cùng độ dài được ký hiệu là  $\Delta$ ; vì vậy:

$$\begin{aligned} \mathfrak{R}_1 &= (-\infty, a] \\ \mathfrak{R}_2 &= (a, a + \Delta] \\ \mathfrak{R}_3 &= (a + \Delta, a + 2\Delta] \\ &\dots \\ \mathfrak{R}_N &= ((N-2)\Delta, \infty] \end{aligned}$$

Mức lượng tử hoá tối ưu trong mỗi khoảng lượng tử hoá được chứng minh là trọng tâm của khoảng đó; nghĩa là

$$\hat{x}_i = E[X | X \in \mathfrak{R}_i] = \frac{\int_{\mathfrak{R}_i} x \cdot f_X(x) dx}{\int_{\mathfrak{R}_i} f_X(x) dx}, \quad 1 \leq i \leq N \quad (3.93)$$

Do đó, việc thiết kế bộ lượng tử hoá đều tương đương với việc xác định  $a$  và  $\Delta$ . Một khi  $a$  và  $\Delta$  đã được xác định, các giá trị của  $\hat{x}_i$  và lượng méo dễ dàng được xác định bằng các biểu thức (3.92) và (3.93). Trong một số trường hợp, đơn giản ta chọn các mức lượng tử là các điểm giữa của các vùng lượng tử, nghĩa là chọn tại khoảng cách  $\Delta/2$  tính từ các biên của các vùng lượng tử.

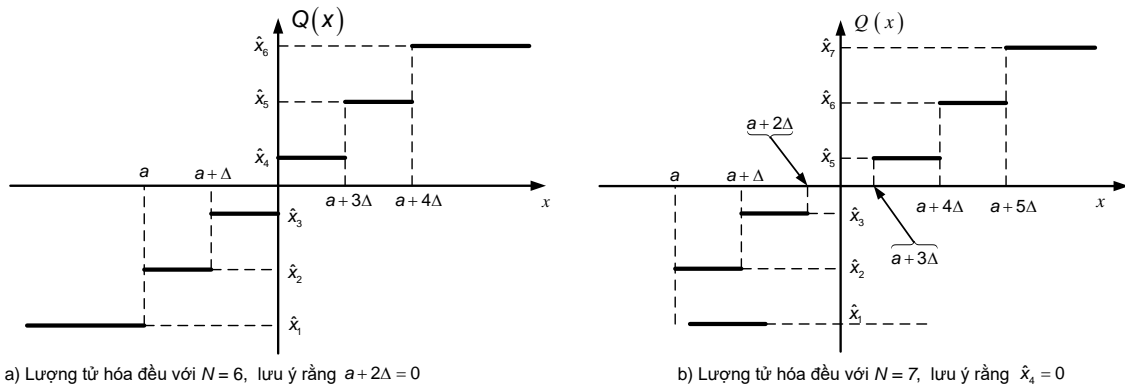
Đồ thị của hàm lượng tử hoá  $Q(x)$  đối với một hàm mật độ xác suất đối xứng của  $X$  và các giá trị chẵn và lẻ của  $N$  được thể hiện trên hình 3.20 tương ứng.

Khi hàm mật độ xác suất đối xứng, bài toán trở nên đơn giản hơn. Trong trường hợp như vậy.

$$\mathfrak{R}_i = \begin{cases} (a_{i-1}, a_i], & 1 \leq i \leq N-1 \\ (a_{i-1}, a_N), & i = N \end{cases} \quad (3.94)$$

Trong đó:

$$\begin{cases} a_0 = -\infty \\ a_i = (i - N/2)\Delta & 1 \leq i \leq N-1 \\ a_N = \infty \end{cases} \quad (3.95)$$



Hình 3.20: Minh họa lượng tử hóa đồng đều

Ta thấy rằng, trong trường hợp này, ta chỉ có một tham số  $\Delta$  phải được chọn để đạt được méo tối thiểu. Giả sử rằng, các mức lượng tử được đặt tại các trọng tâm của các vùng lượng tử hoá, ba chương trình Matlab `NVD3_centroid.m`, `NVD3_mse_dist.m`, `NVD3_uq_dist.m` trong Phụ lục 3A thực hiện tìm: (i) trọng tâm của một vùng; (ii) sai số lượng tử bình phương trung bình (MSE) khi cho trước một phân bố và các biên của vùng lượng tử hoá; (iii) lượng méo. Để sử dụng các hàm này, thì phân bố của nguồn (phụ thuộc đến 3 tham số) phải được cho trong một hàm.

Để được rõ hơn, các bài tập được viết trên Matlab để: (i) xác định trọng tâm của các vùng lượng tử hoá đối với phân bố Gausơ; (ii) xác định sai số trung bình bình phương đối với bộ lượng tử; (iii) khảo sát bộ lượng tử với các mức được đặt ở các điểm giữa của vùng. Tất cả được cho ở Phụ lục 3A và 3B.

### Lượng tử hoá không đều

Trong lượng tử hoá không đều, yêu cầu về các vùng lượng tử hoá (ngoại trừ vùng đầu và cuối) có độ dài như nhau được xem nhẹ, và mỗi vùng lượng tử hoá có thể có một độ dài bất kỳ. Do trong trường hợp này việc tối ưu hoá được thực hiện với những điều kiện giảm nhẹ hơn, kết

quả là có sự vượt trội rõ ràng hơn so với lượng tử hoá đều. Các điều kiện tối ưu khi này, được gọi là các điều kiện Lloyd-Max, và được biểu diễn là:

$$\begin{cases} \hat{x}_i = \frac{\int_{a_{i-1}}^{a_i} xf_X(x)dx}{\int_{a_{i-1}}^{a_i} f_X(x)dx} \\ a_i = \frac{(\hat{x}_{i-1} - \hat{x}_i)}{2} \end{cases} \quad (3.96)$$

Từ những phương trình này, ta kết luận rằng các mức lượng tử tối ưu là các trọng tâm của các vùng lượng tử và các biên tối ưu giữa các vùng lượng tử là các điểm giữa của các mức lượng tử. Để có được lời giải cho các phương trình Lloyd-Max, ta bắt đầu với tập các mức lượng tử  $\hat{x}_i$ . Từ tập này, ta có thể tìm được một tập các biên vùng lượng tử  $a_i$ . Từ tập các  $a_i$  này, ta nhận được tập mới các mức lượng tử. Quá trình này được tiếp tục cho đến khi việc cải thiện về méo từ bước lặp này sang bước lặp khác là nhỏ. Thuật toán này bảo đảm hội tụ tới một giá trị cực tiểu cục bộ, nhưng ở dạng tổng quát thì khó đạt được giá trị tối thiểu toàn bộ.

Thủ tục thiết kế bộ lượng tử hoá tối ưu được trình bày trong file **NVD3\_lloydmax.m** ở Phụ lục 3A. Bài tập minh họa trên Matlab cho vấn đề này được cho ở Phụ lục 3A và 3B.

### 3.6.2. Điều chế xung mã với lượng tử hóa đều và không đều

Trong điều chế xung mã, tín hiệu tương tự trước hết được lấy mẫu tại tốc độ lớn hơn tốc độ Nyquist, sau đó lượng tử hóa các mẫu này. Giả sử tín hiệu tương tự được phân bố trên khoảng  $[-x_{max}, x_{max}]$  và số mức lượng tử là lớn. Các mức lượng tử có thể cách đều hoặc không cách nhau. Dưới đây ta khảo sát cho hai trường hợp này là PCM lượng tử hoá đều và PCM lượng tử hoá không đều.

#### PCM với lượng tử hoá đều

Trong trường hợp này, khoảng  $[-x_{max}, x_{max}]$  có độ dài  $2x_{max}$  được chia thành  $N$  khoảng con bằng nhau, mỗi khoảng con có độ dài  $\Delta = 2x_{max}/N$ . Nếu  $N$  đủ lớn, thì hàm mật độ xác suất của tín hiệu lỗi vào trên mỗi khoảng con có thể được coi là phân bố đều, dẫn đến một lượng méo

$D = \Delta^2/12$ . Nếu  $N$  là một lũy thừa của 2 (tức là  $N = 2^v$ ), thì cần có  $v$  bit để biểu diễn cho mỗi mức lượng tử. Nghĩa là, nếu độ rộng băng của tín hiệu tương tự là  $W$ , và nếu thực hiện lấy mẫu tại tốc độ Nyquist, thì độ rộng băng cần thiết để truyền dẫn tín hiệu PCM tối thiểu là  $vW$  (thực tế thường lấy  $1,5vW$ ). Lượng méo được cho bởi:

$$D = \frac{\Delta}{12} = \frac{x_{max}^2}{3N^2} = \frac{x_{max}^2}{3 \times 4^v} \quad (3.97)$$

Nếu công suất của tín hiệu tương tự được ký hiệu là  $\overline{X^2}$ , thì tỉ số tín hiệu trên tạp âm lượng tử SQNR được cho bởi:

$$SQNR = 3.v^2 \frac{\overline{X^2}}{x_{max}^2} = 3 \times 4^v \frac{\overline{X^2}}{x_{max}^2} = 3 \times 4^v \overline{\tilde{X}^2} \quad (3.98)$$

Trong đó  $\tilde{X}$  là tín hiệu đầu vào chuẩn hoá được xác định là  $\tilde{X} = \frac{X}{x_{max}}$

SQNR tính theo đơn vị dB là:

$$SQNR|_{dB} \approx 4,8 + 6v + \overline{\tilde{X}^2}|_{dB} \quad (3.99)$$

Sau khi lượng tử hoá, mỗi mức lượng tử được mã hóa bởi  $v$  bit. Phương pháp mã hoá thường dùng là mã nhị phân tự nhiên (*NBC: Natural Binary Coding*), nghĩa là mức thấp nhất được ánh xạ thành một chuỗi toàn bit 0 và mức cao nhất được ánh xạ thành một chuỗi toàn bit 1. Toàn bộ các mức còn lại được ánh xạ theo thứ tự tăng dần của giá trị lượng tử.

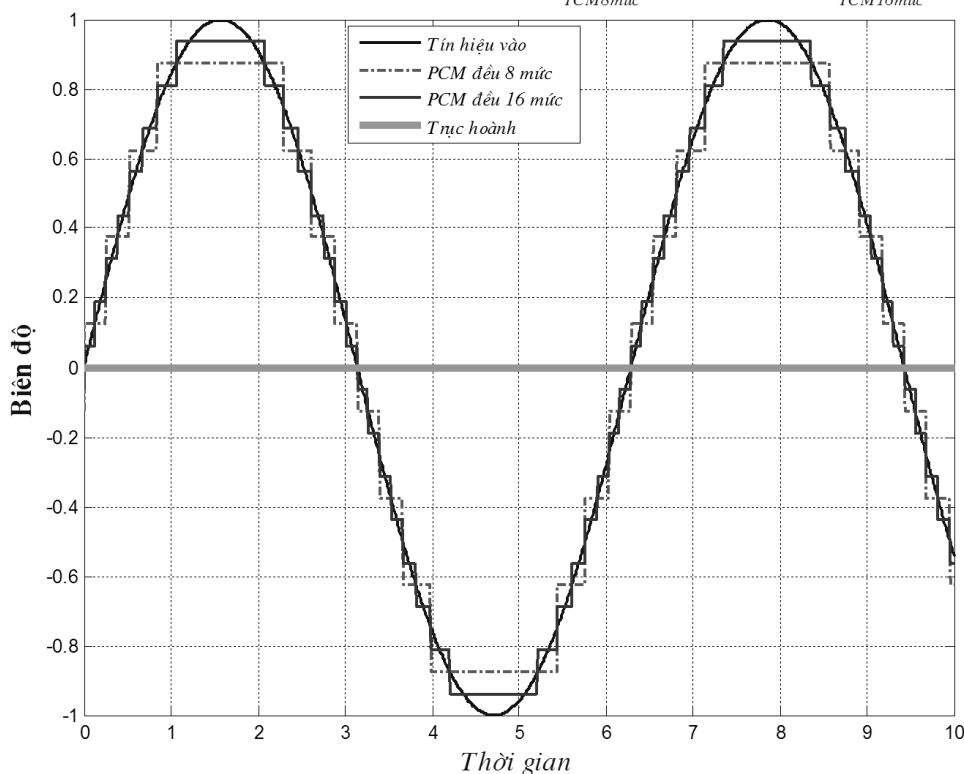
Hàm **NVD3\_u\_pem.m** trong Phụ lục 3A thực hiện bài toán này, theo đó nó nhận 2 tham số đầu vào là: chuỗi các giá trị mẫu  $a$ , số mức lượng tử hoá mong muốn; cho 3 kết quả đầu ra là: chuỗi lượng tử hóa, chuỗi bit đã mã hóa, và tỉ số tín hiệu trên tạp âm lượng tử SQNR theo đơn vị dB.

**Ví dụ 3.6:** [PCM lượng tử hoá đều]

Hãy tạo một tín hiệu hình sin có biên độ bằng 1 và tần số  $\omega = 1$ . Sử dụng sơ đồ PCM lượng tử hoá đều để: (i) lượng tử hoá 8 mức và 16 mức; (ii) vẽ đồ thị tín hiệu gốc và các tín hiệu lượng tử hoá trên cùng một hệ toạ độ; (iii) so sánh SQNR trong trường hợp 8 mức lượng tử và 16 mức lượng tử.



Mô phỏng PCM đều với trường hợp 8 mức và 16 mức:  $SQNR_{PCM8mức} = 18.9023dB$ ;  $SQNR_{PCM16mức} = 25.1272dB$



Hình 3.21: PCM lượng tử hóa đều cho tín hiệu hình sin dùng 8 mức và 16 mức lượng tử

Ta chọn một cách tùy ý thời gian tồn tại của tín hiệu là 10s (thời gian mô phỏng). Sau đó, hàm **NVD3\_u\_pcm.m** trong Phụ lục 3A để tạo các tín hiệu lượng tử cho hai trường hợp 8 và 16 mức lượng tử. Các SQNR tính được là: 18,90 dB đối với PCM 8 mức lượng tử hoá đều và 25,13 dB đối với trường hợp PCM lượng tử hóa đều 16 mức. Chương trình mô phỏng thực hiện bài toán này được cho ở file **NVD3\_Sim36.m** trong Phụ lục 3A. Kết quả chạy chương trình này được cho ở hình 3.21.

### Ví dụ 3.7: PCM lượng tử hoá đều

Hãy tạo một chuỗi 500 biến ngẫu nhiên Gauss trung bình 0 và phương sai bằng 1. Sử dụng hàm **NVD3\_u\_pcm.m** trong Phụ lục 3A để tìm SQNR khi số mức lượng tử hoá là 64. Hãy tìm 5 giá trị đầu tiên của chuỗi, tương ứng với các giá trị đã lượng tử, và các từ mã tương ứng.

Chương trình Matlab **NVD3\_sim37.m** trong Phụ lục 3A thực hiện bài toán này.

Khi chạy chương trình **NVD3\_sim37**, quan trắc được các kết quả sau:

$$SQNR = 31,66 \text{ dB}$$

Tín hiệu lỗi vào = [0,1775; -0,4540; 1,0683; - 2,2541; 0,5376]

Các giá trị sau lượng tử = [0,1569; -0,4708; 1,0985; -2,2494; 0,5754]

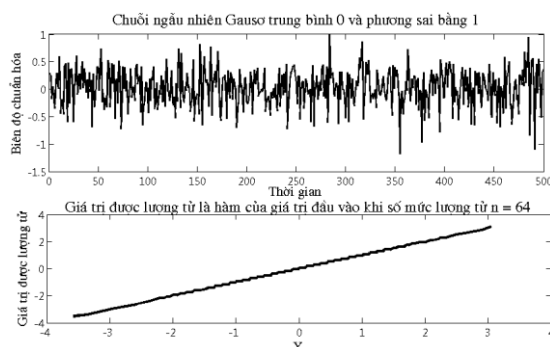
$$\text{Các từ mã} = \begin{cases} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{cases}$$

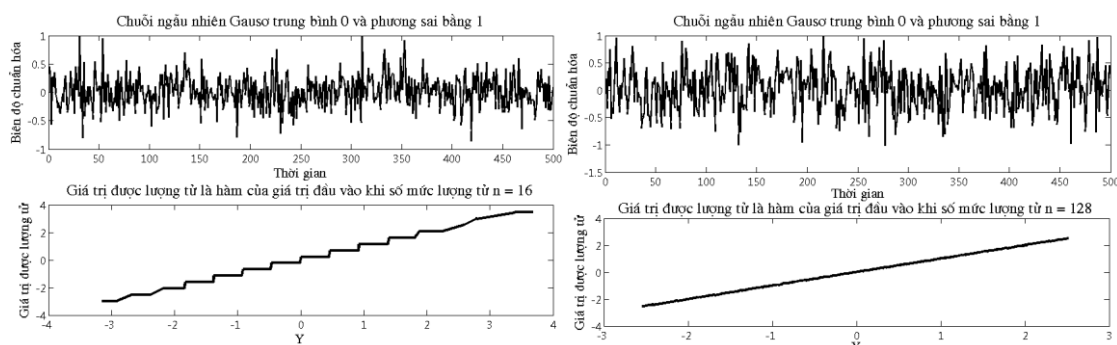
Lưu ý rằng, các lần chạy khác nhau của chương trình này cho ra các kết quả khác nhau đối với tín hiệu đầu vào, các giá trị sau lượng tử, và các từ mã. Tuy nhiên, các SQNR tính được đều rất sát nhau.

### Ví dụ 3.8: Sai số lượng tử

Từ ví dụ 3.7, hãy vẽ đồ thị sai số lượng tử, được định nghĩa là sự khác nhau giữa giá trị đầu vào và giá trị được lượng tử hóa cho các trường hợp 16, 64, và 128 mức lượng tử. Ngoài ra, vẽ đồ thị để khảo sát giá trị được lượng tử hóa theo giá trị đầu vào (tức là, giá trị được lượng tử hóa là hàm của giá trị đầu vào) ứng với các trường hợp 16, 64 và 128 mức lượng tử. Chương trình Matlab thực hiện bài toán này được cho ở file **NVD3\_sim38.m** trong Phụ lục 3A. Kết quả chạy chương trình này được cho ở hình 3.22.

Thấy rõ từ hình 3.22 khi số mức lượng tử càng lớn thì sai số lượng tử càng nhỏ, như được mong đợi. Ngoài ra cũng cần lưu ý rằng, khi số mức lượng tử càng lớn, thì quan hệ giữa tín hiệu lỗi vào và các giá trị được lượng tử hoá càng tiến tới một đường thẳng có độ dốc bằng 1 đi qua gốc toạ độ; nghĩa là, tín hiệu vào/ra của bộ lượng tử hóa trở nên gần như bằng nhau. Ngược lại, khi số mức lượng tử hoá nhỏ (chẳng hạn 16 mức) thì quan hệ này là khác nhau xa.





Hình 3.22: Sai số lượng tử trong sơ đồ PCM lượng tử hóa đều với 16; 64; 128 mức lượng tử

### PCM với lượng tử hoá không đều

Trong trường hợp này, trước hết cho tín hiệu đầu vào qua một phần tử phi tuyến để làm giảm dải động, sau đó đưa vào một hệ thống PCM lượng tử hoá đều. Tại đầu thu, tín hiệu lỗi ra được cho qua bộ nghịch đảo của phần tử phi tuyến đã được sử dụng ở phần phát. Nhờ tính đối nghịch (hàm ngược của nhau) mà ảnh hưởng tổng thể là tương đương với một hệ thống PCM có khoảng cách không đều giữa các mức lượng tử. Nói chung, để truyền tín hiệu tiếng nói, thì các đặc tính phi tuyến được dùng là các đặc tính phi tuyến hoặc luật A hoặc luật  $\mu$ :

Đặc tính phi tuyến luật  $\mu$  được định nghĩa bởi quan hệ:

$$y = g(x) = \frac{\log(1 + \mu|x|)}{\log(1 + \mu)} \text{sign}(x) \quad (4.4.4)$$

Trong đó  $x$  là tín hiệu lỗi vào chuẩn hoá ( $|x| \leq 1$ ),  $\mu$  là tham số, đặc tính phi tuyến luật  $\mu$  ở dạng tiêu chuẩn  $\mu = 255$ .

Nghịch đảo của đặc tính phi tuyến luật  $\mu$  được cho bởi

$$x = \frac{(1 + \mu)^{|y|} - 1}{\mu} \text{sign}(y) \quad (4.4.5)$$

Thực thi hóa đặc tính phi tuyến luật  $\mu$  và nghịch đảo của nó bằng Matlab được cho ở file **NVD3\_mulaw.m** và **NVD3\_invmulaw.m** trong Phụ lục 3A. Hàm **NVD3\_mula\_pcm.m** là tương đương với hàm **NVD3\_u\_pcm.m** khi dùng sơ đồ PCM luật  $\mu$ . Chương trình Matlab này được cho ở file **NVD3\_mula\_pcm.m** trong Phụ lục 3A.