

Chương 8

HẬU XỬ LÝ

8.1. Mở đầu

Chương 8 trình bày ngắn gọn chủ đề quan trọng là hậu xử lý. Vai trò của bộ hậu xử lý là thao tác dữ liệu được tạo ra bởi mô phỏng thành dạng hữu hiệu. Các bộ hậu xử lý thường có tính đồ họa mạnh vì hiển thị trực quan dễ hiểu hơn nhiều so với liệt kê dạng số, nên nó là đầu ra dữ liệu phổ biến nhất từ chương trình mô phỏng. Chẳng hạn để thể hiện kết quả mô phỏng BER với cùng lượng tin, sử dụng đồ thị BER truyền đạt thông tin nhanh hơn và dễ hiểu hơn nhiều so với sử dụng bảng số. Nói cách khác, hậu xử lý cho phép ta trực quan hóa tín hiệu và hệ thống trong các miền xử lý như: miền thời gian, miền tần số, miền z ,... cho phép ta quản lý và kiểm soát tiến trình hoạt động của hệ thống một cách tĩnh và động.

Các thường trình hậu xử lý có mức độ phức tạp tính toán đáng kể. Trong khi một số bộ hậu xử lý chỉ lấy dữ liệu được tạo ra bởi mô phỏng, sau đó định dạng dữ liệu để tạo đầu ra đồ họa phù hợp. Ví dụ, thường trình tạo đồ thị xác suất lỗi bit (hoặc xác suất lỗi ký hiệu), P_E là hàm của E_b/N_0 . Các giá trị của P_E cùng với các giá trị của E_b/N_0 được tạo ra bởi mô phỏng, được qua bộ hậu xử lý ở dạng file dữ liệu. Các bộ hậu xử lý chỉ định dạng dữ liệu và tạo đồ thị cần thiết. Các ví dụ khác về các bộ hậu xử lý là tạo đầu ra đồ họa với mức độ xử lý tối thiểu để hiển thị các dạng sóng tín hiệu, các biểu đồ mắt và các biểu đồ tán xạ.

Mặt khác, một số thường trình hậu xử lý bao gồm cả việc xử lý tín hiệu. Đa số chúng đều liên quan đến một số loại ước tính: từ mức đơn giản như tạo ra hoành đồ, là một bộ ước tính hàm mật độ xác suất, cho đến mức phức tạp hơn là các bộ ước tính trễ thời gian, tỷ số tín hiệu trên tạp âm và mật độ phổ công suất. Các ví dụ khác được xét trong chương này gồm ánh xạ tỉ số lỗi ký hiệu kênh thành tỉ số lỗi bit được giải mã đối với hệ thống sử dụng mã hóa kiểm soát lỗi. Liệt kê hết các hoạt động hậu xử lý là không thể, và chương này chỉ đề cập một vài ví dụ điển hình. Tất cả những hoạt động này đều được xem là hậu xử lý vì chúng sử dụng dữ liệu được tạo ra bởi mô phỏng, và chỉ được thực hiện sau khi việc mô phỏng đã hoàn tất.

Muốn vậy chương được tổ chức và thực hiện như sau:

Xét chủ đề hậu xử lý trong mô phỏng bộ điều chế $\pi/4$ DQPSK nhằm trình bày quá trình tạo: đồ thị dạng sóng, chòm sao tín hiệu, biểu đồ mắt, cũng như việc triển khai các bộ ước tính dựa vào dữ liệu mô phỏng. Theo đó:

Trước hết là, xét hoành đồ là bộ ước tính hàm mật độ xác suất. Ở dạng tổng quát, trong khi hoành đồ là bộ ước tính pdf chệch và không kiên định, thì ta sẽ thấy rằng nếu có đủ dữ liệu khả dụng thì cả độ chệch và phương sai đều có thể được bỏ qua.

Tiếp theo là, xét các phiên bản của periodogram như một bộ ước tính PSD của tín hiệu. Ta sẽ thấy rằng, trong khi phương sai của periodogram cơ bản thường lớn ở mức không thể chấp nhận được, thì phương sai đó có thể được giảm bằng cách lấy trung bình các periodogram

của các đoạn dữ liệu được cửa sổ hóa. Quá trình phân đoạn hóa và trung bình hóa các periodogram dẫn đến phải dung hoà giữa phương sai và độ phân giải của bộ ước tính.

Tiếp nữa là, xét bộ ước tính để cho phép ước tính: độ lợi của hệ thống, thời gian trễ, tỷ số tín hiệu trên tạp âm SNR. Các kỹ thuật được triển khai ở đây sẽ được sử dụng trong các chương sau khi xét các kỹ thuật bán giải tích.

Chủ đề cuối cùng là, ước tính xác suất lỗi giải mã trong các hệ thống có sử dụng mã hóa kiểm soát lỗi. Kỹ thuật ở đây là dùng mô phỏng để xác định tỉ lệ ký hiệu kênh và sau đó dùng các kỹ thuật giới hạn để ánh xạ các tỉ số lỗi ký hiệu SER kênh thành tỉ số lỗi bit BER được giải mã.

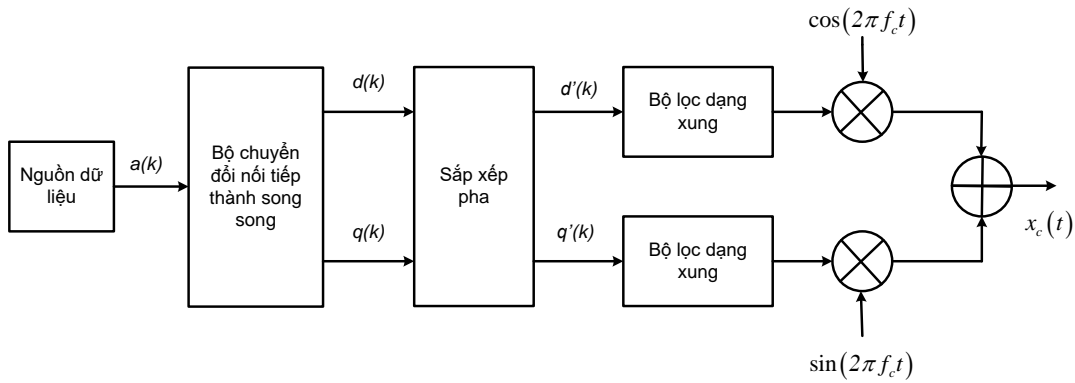
8.2. Kỹ thuật đồ họa cơ bản

Để minh họa các kỹ thuật đồ họa trong bộ hậu xử lý, các khái niệm được xét bên trong ngữ cảnh của một hệ thống minh họa. Tất nhiên, việc lựa chọn hệ thống minh họa là bất kỳ. Ở đây ta xét cho trường hợp điều chế QPSK vì sai $\pi/4$ (ký hiệu $\pi/4$ DQPSK) bởi lẽ nó có một số đặc tính được sử dụng trong nhiều hệ thống vô tuyến.

8.2.1. Hệ thống truyền dẫn $\pi/4$ DQPSK

Sơ đồ khối máy phát $\pi/4$ DQPSK được minh họa ở hình 8.1. Đầu ra của nguồn dữ liệu được giả định là một chuỗi a có dạng:

$$a(1)a(2)a(3)a(4)...a(k)...$$



Hình 8.1: Máy phát $\pi/4$ DQPSK

Bộ chuyển đổi nối tiếp thành song song chuyển các ký hiệu có chỉ số lẻ lên kênh đồng pha (trực tiếp) và các ký hiệu có chỉ số chẵn lên kênh vuông pha. Vì vậy:

$$a(1)a(2)a(3)a(4)...a(k) = d(1)q(1)d(2)q(2)...d\left(\frac{k+1}{2}\right).q\left(\frac{k}{2}+1\right)...$$

Tín hiệu phát được cho bởi:

$$x_c(t) = A \cos \left[2\pi f_c t + \theta(k) \right], \quad (k-1)T_s < t < kT_s \quad (8.1)$$

Trong đó T_s là chu kỳ ký hiệu. Độ lệch pha của tín hiệu phát được xác định bởi các giá trị của $d(k)$ và $q(k)$ cũng như độ lệch pha $\theta(k-1)$, là độ lệch pha trong chu kỳ ký hiệu trước. Tất nhiên, sự phụ thuộc vào chu kỳ ký hiệu trước, làm cho $\pi/4$ DQPSK là một kỹ thuật điều chế vi sai. Quan hệ giữa $\theta(k)$ và $\theta(k-1)$ là:

$$\theta(k) = \theta(k-1) + \phi(k) \quad (8.2)$$

Trong đó $\phi(k)$ là hàm của $d(k)$ và $q(k)$, và được định rõ trong bảng 8.1. Các pha phát được tạo ra trong bộ sắp xếp pha được cho ở hình 8.1. Bộ sắp xếp pha sử dụng $d(k)$, $q(k)$, và $\theta(k-1)$ để tạo các giá trị mới $d'(k)$ và $q'(k)$, sao cho tín hiệu phát có pha phù hợp. Sau khi định dạng xung, các tín hiệu kênh đồng pha và vuông pha được dịch lên tần số truyền dẫn f_c .

Bảng 8.1: Các dịch pha vi sai cho $\pi/4$ DQPSK

Các ký hiệu thông tin $d(k)$ và $q(k)$	Dịch pha vi sai $\phi(k)$
1 1	$\pi/4$
0 1	$3\pi/4$
0 0	$-3\pi/4$
1 0	$-\pi/4$

Như một ví dụ, giả sử đầu ra của nguồn dữ liệu là chuỗi nhị phân:

$$00 \ 10 \ 11 \ 01 \ 11... \quad (8.3)$$

Và pha ban đầu được định rõ là $\theta(0) = 0$. Vì 2 ký hiệu dữ liệu đầu tiên là 00, nên theo bảng 8.1 ta có $\phi(1) = -3\pi/4$. Từ (8.2) ta có:

$$\theta(1) = 0 - \frac{3\pi}{4} = -\frac{3\pi}{4}$$

2 ký hiệu dữ liệu tiếp theo là 10, do đó $\phi(2) = -\pi/4$. Từ (8.2) ta được:

$$\theta(2) = -\frac{3\pi}{4} - \frac{\pi}{4} = -\pi$$

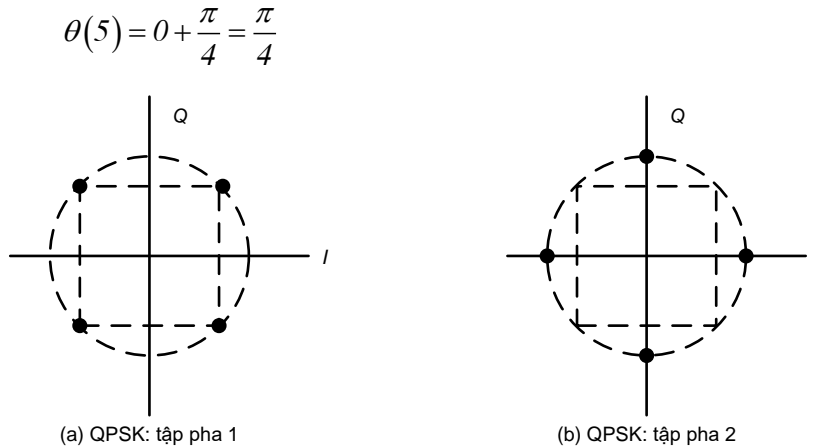
2 ký hiệu dữ liệu tiếp theo là 11. Vì vậy $\phi(3) = \pi/4$ Từ (8.2) cho ta:

$$\theta(3) = -\pi + \frac{\pi}{4} = -\frac{3\pi}{4}$$

Tương tự $\phi(4) = 3\pi/4$. Từ (8.2) ta được:

$$\theta(4) = -\frac{3\pi}{4} + \frac{3\pi}{4} = 0$$

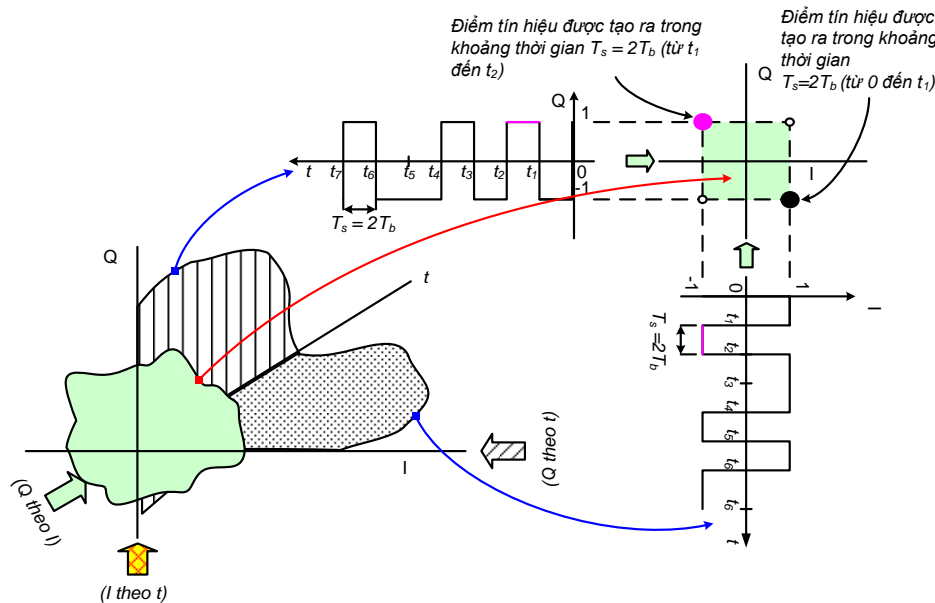
Tương tự $\phi(5) = \pi/4$ cho ta:



Hình 8.2: Các chòm sao tín hiệu đối với $\pi/4$ DQPSK

Quan sát nhanh thấy rõ, $\theta(1)$, $\theta(3)$, $\theta(5)$ là các pha từ chòm sao tín hiệu QPSK thứ nhất được minh hoạ trong hình 8.2 (a), và $\theta(2)$, $\theta(4)$ là các pha từ chòm sao tín hiệu QPSK thứ hai được minh hoạ trong hình 8.2 (b). Vì vậy, $\pi/4$ DQPSK hoạt động bằng cách phát các điểm tín hiệu phát từ các chòm sao tín hiệu QPSK một cách luân phiên nhau trong đó cứ 2 chòm sao tín hiệu QPSK được chuyển chỗ bằng cách quay pha $\pi/4$. Mặc dù điều này được minh hoạ bằng cách sử dụng một chuỗi dữ liệu cụ thể, nhưng ta thấy rằng kết quả là tổng quát, vì các pha vi sai chỉ lấy các giá trị $\pm\pi/4$ hoặc $\pm3\pi/4$.

8.2.2. Dạng sóng, biểu đồ mắt và biểu đồ tán xạ



Hình 8.3: Hệ toạ độ 3 chiều

Trước khi giải thích các chương trình cơ bản để vẽ các dạng sóng, các biểu đồ mắt, và các đồ thị tán xạ, ta minh hoạ quan hệ giữa những biểu đồ này. Giả sử ta triển khai một hệ toạ

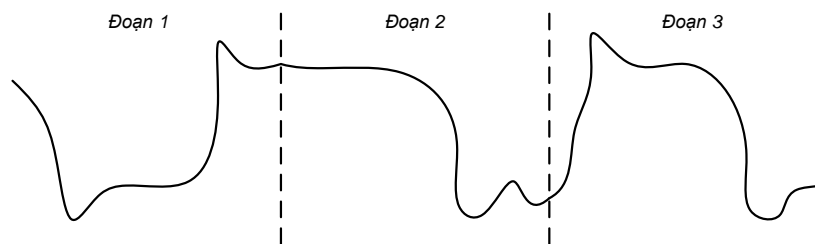
độ 3 chiều được cho ở hình 8.3 có các trục được đánh nhãn tương ứng. Lưu ý rằng, hình thành 3 mặt phẳng, mỗi mặt phẳng chứa 2 trục được tạo ra bởi các trục: (I và t), (Q và t), (Q và I). Nếu: (i) tín hiệu kênh đồng pha $x_d(t)$ được vẽ trên mặt phẳng (I, t); (ii) tín hiệu kênh vuông pha $x_q(t)$ được vẽ trên mặt phẳng (Q, t), thì tín hiệu 3 chiều được thông số hóa theo t (hàm của t) được tạo ra. Chiếu tín hiệu này lên một không gian con cho trước (I, t), (Q, t), hoặc (I, Q) tạo ra $x_d(t)$, $x_q(t)$, hay biểu đồ tán xạ (vẽ $x_q(t)$ là một hàm của $x_d(t)$). Điều này được minh họa trong hình 8.3. Quan sát từ phía phải, cho thấy tín hiệu vuông pha $x_q(t)$ là hàm của t , được vẽ trên mặt phẳng (Q, t). Tương tự, quan sát từ dưới, cho thấy tín hiệu kênh đồng pha $x_d(t)$ là hàm của t (được vẽ trên mặt phẳng (I, t)). Nhìn xuống trục thời gian sao cho trục thời gian trở thành một điểm lộ ra đồ thị tán xạ, $x_q(t)$ là hàm của $x_d(t)$, được vẽ trên mặt phẳng (Q, I).

Trong khi ảnh 3 chiều (Q, I, t) là hiếm khi được tạo ra trong thực tế, nhưng việc trực quan hóa (Q, I, t) là một công cụ học tập tốt và cho thấy rõ mối quan hệ giữa $x_d(t)$, $x_q(t)$ và biểu đồ tán xạ.

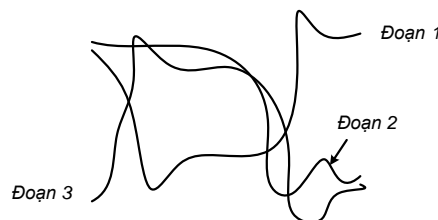
Biểu đồ mắt

Biểu đồ mắt cho phép đánh giá *định tính* hiệu năng hệ thống. Một mắt mở và sáng thể hiện hiệu năng tốt, ngược lại hiệu năng kém. Ngoài ra, kích thước của mắt liên quan đến yêu cầu độ chính xác của bộ đồng bộ ký hiệu. Trong khi biểu đồ mắt không cho đánh giá *định lượng* hiệu năng hệ thống, thì khó để quan niệm một hệ thống hiệu năng tốt có biểu đồ mắt xấu.

Quá trình hình thành biểu đồ mắt được minh họa ở hình 8.4. Ba đoạn dạng sóng trong đó mỗi đoạn tương ứng với mỗi chu kỳ ký hiệu. Dạng sóng tương ứng với 3 ký hiệu dữ liệu được minh họa trong hình 8.4(a). Giả sử rằng dạng sóng này được hiển thị trên máy hiện sóng và máy hiện sóng được đồng bộ với dạng sóng tại các điểm đường sọc thẳng. Kết quả sẽ là biểu đồ mắt 3 đoạn được minh họa trong hình 8.4 (b).



(a) 3 đoạn (60 mẫu) của dạng sóng

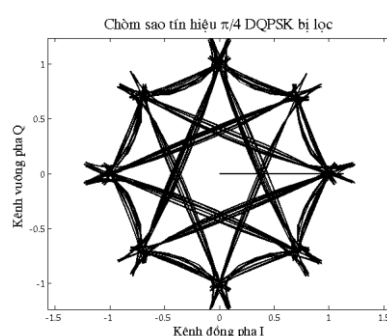
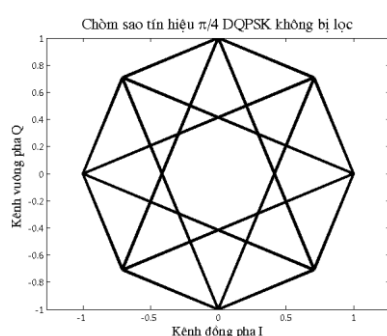
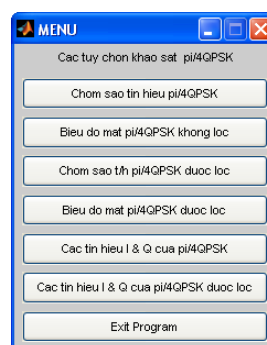


(b) Biểu đồ mắt 3 đoạn

Hình 8.4: Tạo biểu đồ mắt

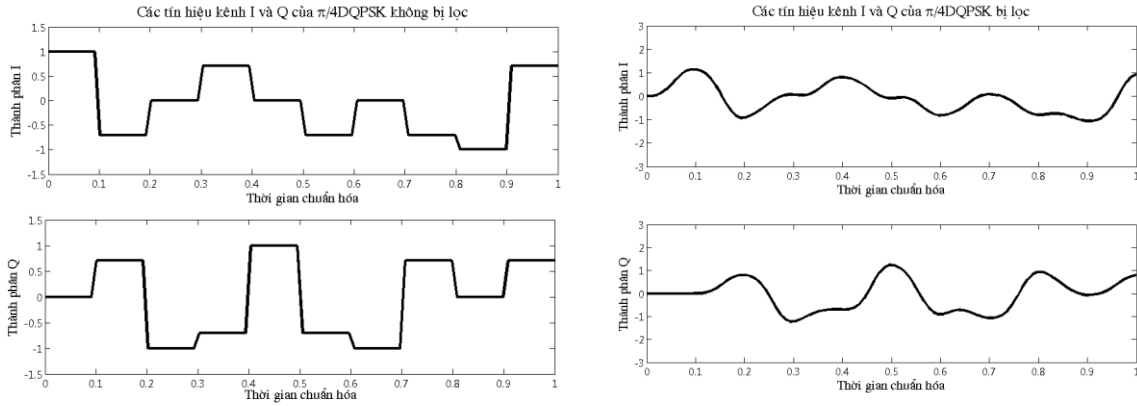
Ví dụ 8.1: Ví dụ này, sẽ tạo và hiển thị một vài tín hiệu quan trọng trong hệ thống $\pi/2$ DPSK. Chương trình Matlab mô phỏng hệ thống và tạo đồ hoạ được cho ở file **NVD8_pi4sim** ở Phụ lục 8A. Chạy chương trình **NVD_pi4sim** tại dấu nhắc cửa sổ lệnh Matlab. Từ menu này chọn một trong 7 lựa chọn sau:

1. Chòm sao tín hiệu $\pi/4$ DQPSK không bị lọc
2. Biểu đồ mắt tín hiệu $\pi/4$ DQPSK không bị lọc
3. Chòm sao tín hiệu $\pi/4$ DQPSK bị lọc
4. Biểu đồ mắt tín hiệu $\pi/4$ DQPSK bị lọc
5. Các tín hiệu vuông pha và đồng pha không bị lọc
6. Các tín hiệu vuông pha và đồng pha bị lọc
7. Thoát chương trình mô phỏng

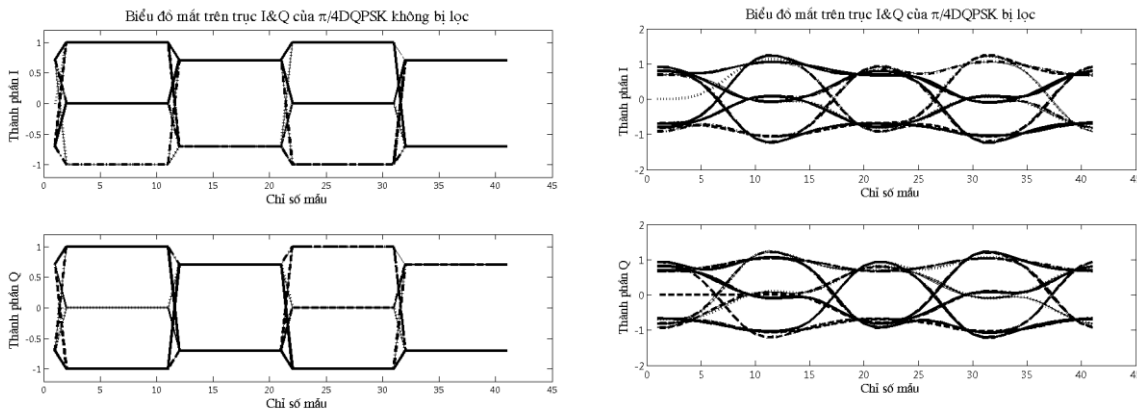


Hình 8.5: Chòm sao tín hiệu $\pi/4$ DQPSK không bị lọc và bị lọc

Để được rõ, cần phải nghiên cứu mã chương trình mô phỏng trong Phụ lục 8A, ở đó minh hoạ nhiều thủ tục hậu xử lý. Ngoài ra mã chương trình được sử dụng để tạo các đồ thị khác nhau trong bộ hậu xử lý của các chương trình mô phỏng khác. Ở đây minh hoạ 3 kết quả đáng quan tâm. Hình 8.5, 8.6 và 8.7 minh hoạ biểu đồ tán xạ (chòm sao tín hiệu), các tín hiệu kênh vuông pha và kênh đồng pha, và các biểu đồ mắt trên kênh vuông pha và đồng pha. Lưu ý rằng, bằng cách mượn tượng tín hiệu ba chiều trong không gian (I, Q, t) thấy rõ mối quan hệ giữa các hình 8.5 và 8.7. Để được rõ hơn về mức độ ảnh hưởng của tham số hệ thống lên các kết quả quan sát, ta nên thay đổi các tham số bộ lọc và các tham số khác như số ký hiệu mô phỏng, lấy mẫu trên ký hiệu.



Hình 8.6: Các thành phần I & Q của tín hiệu $\pi/4$ DQPSK bị lọc và không bị lọc



Hình 8.7: Biểu đồ mắt của tín hiệu $\pi/4$ DQPSK không bị lọc và bị lọc

8.3. Ước tính

Nhiều thường trình ước tính hiệu năng dựa trên dữ liệu được tạo ra bởi chương trình mô phỏng. Ở đây ta xét một số thường trình ước tính căn bản nhất. Chi tiết hóa cho vấn đề này được cho ở phần Phụ lục 7B2, ở đó được trình bày súc tích và rất trực quan dễ hiểu ở dạng các định nghĩa và các biểu thức kết quả.

8.3.1. Hoành đồ

Nếu có sẵn một tập các mẫu của một quá trình ngẫu nhiên (từ mô phỏng hoặc đo kiểm), thì ta có thể tạo ra hoành đồ từ tập các mẫu, là một bộ ước tính hàm mật độ xác suất pdf cơ bản. Hoành đồ được tạo ra bằng cách nhóm dữ liệu, toàn bộ N mẫu dữ liệu vào B bin (B ngăn). Giả sử mỗi bin đều có cùng độ rộng W , và tâm của mỗi bin là b_i . Vì vậy, một mẫu $x[n]$ cho trước rơi vào bin thứ i nếu:

$$b_i - \frac{W}{2} < x[n] \leq b_i + \frac{W}{2} \quad (8.4)$$

Đại lượng quan tâm là N_i , thể hiện số mẫu rơi vào bin thứ i . Rõ ràng:

$$N = \sum_{i=1}^B N_i \quad (8.5)$$

Ta ký hiệu $Count\{N:R\}$ để thể hiện số lượng mẫu trong tập N mẫu tổng rơi vào bin hoành đồ được xác định bởi R . Vì vậy:

$$N_i = Count \left\{ N : \underbrace{b_i - \frac{W}{2} < x[n] \leq b_i + \frac{W}{2}}_R \right\} \quad (8.6)$$

Đồ thị dạng thanh, trong đó chiều cao của mỗi thanh tỷ lệ với N_i , và mỗi thanh được trung tâm tại b_i . Để là một bộ ước tính pdf, thì hoành đồ được tỷ lệ hóa sao cho toàn bộ diện tích đó bằng 1. Muốn vậy, ta chia N_i cho NW , khi này chiều cao của mỗi thanh là N_i/NW . Diện tích của thanh, A_i , thể hiện bin hoành đồ thứ i , được tìm thấy bằng cách nhân chiều cao với độ rộng W . Vì vậy:

$$A_i = \left(\frac{N_i}{N.W} \right).W = \frac{N_i}{N} \quad (8.7)$$

Lưu ý rằng, A_i thể hiện tần suất tương đối của bin thứ i . Diện tích tổng là:

$$\sum_{i=1}^B \frac{N_i}{N} = 1 \quad (8.8)$$

Như yêu cầu đề ra nếu muốn sử dụng hoành đồ để biểu diễn hàm mật độ xác suất pdf

Lưu ý rằng, mỗi bin của hoành đồ biểu diễn pdf trên độ rộng W . Với một số điểm bên trong một bin cho trước, thì bộ ước tính pdf sẽ *không bị chệch*. Tuy nhiên, trên hầu hết khoảng được định nghĩa bởi một bin hoành đồ cho trước, thì bộ ước tính sẽ *bị chệch*. Dễ dàng thấy rằng, bằng cách khai triển hàm pdf $f_X(x)$ theo chuỗi Taylor xung quanh tâm của bin thứ i , thì độ chệch là:

$$E \{ f_X(b_i) - \hat{f}_X(b_i) \} \approx \frac{d^2 f_X(x)}{dx^2} \Big|_{x=b_i} \frac{W^2}{24} \quad (8.9)$$

Vì vậy, độ chệch chỉ có thể được giảm bằng cách giảm độ rộng bin W .

Cũng có thể thấy rằng phương sai của bộ ước tính là:

$$\text{var} \{ \hat{f}_X(b_i) \} \approx \frac{1}{N.W} f_X(b_i) \left[1 - \underbrace{W.f_X(b_i)}_{\substack{\text{xác suất của sự kiện} \\ \text{một mẫu cho trước rơi vào bin } b_i}} \right] \quad (8.10)$$

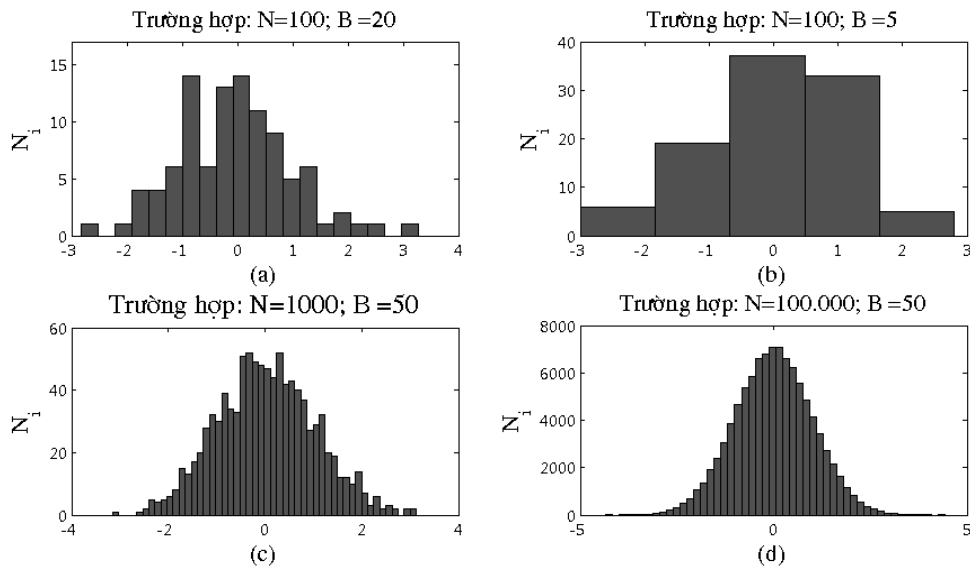
Lưu ý rằng, $W.f_X(b_i)$ là xác suất của sự kiện mà một mẫu cho trước rơi vào bin b_i . Vì nó là xác suất, nên $W.f_X(b_i) \leq 1$ là nhỏ hơn 1 và thường nhỏ hơn 1 rất nhiều. Vì vậy:

$$\text{var} \{ \hat{f}_X(b_i) \} \approx \frac{1}{N.W} f_X(b_i) \quad (8.11)$$

Ta thấy rằng, với N cố định, việc tăng W sẽ làm giảm phương sai của bộ ước tính, nhưng lại làm tăng độ chệch, được thấy từ (8.9). Vì vậy, ta mong muốn W nhỏ và NW lớn. Vì vậy, phải có quan hệ giữa N và W . Chẳng hạn, nếu $W = 1/\sqrt{N}$ và $NW = \sqrt{N}$, thì $W \rightarrow 0$ khi $N \rightarrow \infty$ và $NW \rightarrow \infty$ khi $N \rightarrow \infty$. Trong trường hợp này, với N đủ lớn thì pdf có độ chệch và phương sai là không đáng kể. Ví dụ 8.2 minh hoạ hoành đồ đối với những lựa chọn N và W khác nhau.

Ví dụ 8.2: Giả sử, ta tạo N mẫu của một biến ngẫu nhiên Gausơ phương sai đơn vị trung bình 0. Hàm pdf được ước tính bằng cách xây dựng một hoành đồ có B bin. Trong ví dụ này, ta sẽ khảo sát ảnh hưởng của việc thay đổi giá trị của N và B . Chương trình Matlab thực hiện bài toán này được cho ở file **NVD8_hist.m** (có trong Phụ lục 8A).

Kết quả chạy chương trình này được minh hoạ ở hình 8.8. Hình 8.8(a) minh hoạ các kết quả khi $N = 100$ và $B = 20$, cho thấy hoành đồ không được định rõ lắm vì $N/B = 5$ là quá nhỏ để tạo ra bộ ước tính tin cậy cho một số mẫu rơi vào bin cho trước. Hình 8.8 (b) minh hoạ kết quả khi $N = 100$ và $B = 5$. Trong khi tỷ số N/B là quá lớn thì số bin là quá nhỏ. Lưu ý rằng khi có định N thì biểu đồ thống kê có $B = 5$ sẽ biểu lộ độ chệch lớn hơn so với trường hợp $B = 20$. Hình 8.8(c) minh hoạ kết quả khi $N = 1000$ và $B = 50$. Trong khi cho ta bộ ước tính tốt hơn thì tỷ số N/B lại quá nhỏ và ta tăng N . Hình 8.8 (d) minh hoạ kết quả cho $N = 100000$ và $B = 50$, tạo ra $N/B = 2000$ và biểu đồ thống kê có hình dạng Gausơ được dự đoán trước. Nếu pdf cơ bản có hình dạng phức tạp, thì phải cần đến số lượng mẫu rất lớn nếu muốn có được hoành đồ là bộ ước tính chính xác của pdf.



Hình 8.8: Khảo sát hoành đồ của với biến ngẫu nhiên Gausơ với các giá trị của N và B

8.3.2. Ước tính mật độ phổ công suất

Một hoạt động hậu xử lý khác thường được dùng trong nghiên cứu mô phỏng là ước tính mật độ phổ công suất PSD của một tín hiệu tại một điểm trong hệ thống. Đây là một nhiệm vụ khó hơn. Thường thì, dạng sóng quan tâm là hàm mẫu của quá trình ngẫu nhiên. Dẫn đến trường hợp PSD tại một giá trị tần số cho trước f_l là một biến ngẫu nhiên. Vì vậy, nảy sinh bài

toán giảm thiểu phương sai của ước tính phổ $\hat{S}(f_i)$. Tồn tại nhiều tài liệu và bài báo được viết cho chủ đề này, và nhiều kỹ thuật đã được triển khai để ước tính PSD. Trong phần này, ta chỉ xét các kỹ thuật cơ bản nhất. Các kỹ thuật được xét ở đây dựa vào thuật toán FFT, và được dùng trong mô phỏng.

Periodogram

Thuật toán ước tính PSD đơn giản nhất, nhanh nhất, và thường được sử dụng là periodogram, được định nghĩa bởi:

$$\hat{S}(kf_\Delta) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi kf_\Delta n} \right|^2 = \frac{1}{N} |I_N(kf_\Delta)|^2, \quad k = 0, 1, \dots, N-1 \quad (8.12)$$

Trong đó: N là tổng số mẫu có trong bản ghi dữ liệu, và $I_N(kf_\Delta)$ là FFT N điểm của dữ liệu trong đó ước tính PSD tại tần số $f = kf_\Delta$ sẽ được tính toán. Hiệu quả tính toán của periodogram từ việc sử dụng FFT để tạo ước tính PSD. Kết quả ta được N ước tính trong miền tần số với độ phân giải là $f_\Delta = f_s / N$, trong đó f_s là tần số lấy mẫu liên quan đến các điểm trong miền thời gian để thực hiện ước tính phổ. Lưu ý rằng, f_s trong trường hợp này không phải lúc nào cũng là tần số lấy mẫu liên quan đến mô phỏng. Chẳng hạn, nếu trong một mô phỏng cho trước một tín hiệu được lấy mẫu quá mức, thì các mẫu đó có thể được giảm bớt trước khi thực hiện ước tính phổ (xem kỹ thuật tăng và giảm mẫu ở chương 3).

Khó khăn đối với periodogram, nó là bộ ước tính PSD *chệch* và *không kiên định* tại tần số f . Trong nhiều ứng dụng, phương sai của periodogram là không thể chấp nhận được. Thực tế, do bản ghi số liệu là hữu hạn, nên không thể tránh khỏi kết quả bị chệch. Tuy nhiên, khi N đủ lớn thì độ chệch có thể bỏ qua. Vì vậy, khó khăn chủ yếu là do phương sai lớn. Giả sử, các mẫu dữ liệu $x[n]$ là độc lập, thì phương sai của ước tính phổ tại tần số f là:

$$\text{var}(\hat{S}(f)) = \sigma_x^4 \left\{ 1 + \left(\frac{\sin[2\pi fN]}{N \cdot \sin[2\pi f]} \right)^2 \right\} \quad (8.13)$$

Trong đó σ_x^2 là phương sai của các mẫu dữ liệu $x[n]$. Ta thấy rằng, phương sai của $\hat{S}(f)$ không tiến tới 0 khi $N \rightarrow \infty$, và khi N đủ lớn thì phương sai của ước tính phổ trở nên độc lập với tần số. Tuy nhiên periodogram, nó vẫn hữu hiệu để "tìm kiếm nhanh" PSD.

Periodogram với cửa sổ dữ liệu

Nếu cửa sổ dữ liệu chưa được định rõ, thì cửa sổ chữ nhật mặc định được sử dụng. Với cửa sổ chữ nhật, mỗi giá trị mẫu $x[n]$ được nhân với $w[n] = 1$ khi $0 \leq n \leq N-1$. Ảnh hưởng của cửa sổ chữ nhật là lấy tổng chập các mẫu dữ liệu $x[n]$, với biến đổi Fourier của $w[n]$, nó có phổ biên độ là $\sin(\pi Nf) / N \sin(\pi f)$. Khi xét trong miền tần số, thì cấu trúc búp phụ của cửa sổ chữ nhật làm mất đáng kể phổ, dẫn đến làm méo và làm giảm dải động của phổ được ước tính.

Khi sử dụng của sổ dữ liệu bất kỳ, thì ước tính phổ $\hat{S}(kf_\Delta)$ có dạng:

$$\tilde{S}(kf_\Delta) = \frac{1}{U} \left| \sum_{n=0}^{N-1} x[n] \cdot w[n] \cdot e^{-j2\pi kf_\Delta n} \right|^2 \quad (8.14)$$

Trong đó U là năng lượng trong cửa sổ dữ liệu, được cho bởi:

$$U = \sum_{n=0}^{N-1} w^2[n] \quad (8.15)$$

Lưu ý rằng, với cửa sổ chữ nhật trong đó $w[n] = 1, \forall n$, thì $U = N$ và nhận được các kết quả (8.12). Việc lựa chọn cửa sổ dữ liệu thể hiện một số dung hoà. Cửa sổ chữ nhật lý tưởng phải có khoảng thời gian hữu hạn sao cho biến đổi Fourier của dữ liệu đó $I_N(kf_\Delta)$ được ước tính một cách chính xác. Ngoài ra, biến đổi Fourier của bản ghi dữ liệu được ước tính đó phải không bị ảnh hưởng xấu bởi hàm cửa sổ. Do tính đối ngẫu của biến đổi Fourier là, phép nhân trong miền thời gian tương ứng phép tích chập trong miền tần số, và vì vậy chỉ khi lấy tích chập với hàm xung kim thì mới không làm thay đổi phép biến đổi đó, nên hàm cửa sổ lý tưởng là một **xung kim** trong miền tần số (lưu ý rằng, phổ của một xung kim là vô hạn, và ngược lại biến đổi Fourier ngược của một xung kim tần số là vô hạn). Vì biến đổi Fourier của xung kim là vô hạn nên tồn tại các yêu cầu đối nghịch. Vì vậy, cần phải tìm cửa sổ dữ liệu trong miền tần số để lộ ra búp chính hẹp quanh tần số $f = 0$ và các búp phụ bị suy hao lớn.

Các periodogram phân đoạn

Kỹ thuật phổ dụng để giảm phương sai liên quan với periodogram là chia bản ghi dữ liệu N mẫu thành K đoạn với mỗi đoạn gồm M mẫu. FFT được tính toán cho mỗi đoạn, sau đó lấy trung bình các kết quả. Quá trình lấy trung bình làm giảm phương sai của ước tính phổ. Các đoạn có thể sẽ chồng lấn nhau hoặc không chồng lấn nhau. Nếu các đoạn không chồng lấn nhau thì $K = N/M$, ngược lại thì $K > N/M$.

Periodogram của đoạn dữ liệu thứ i được cho bởi:

$$I_M^{(i)}(kf_\Delta) = \frac{1}{U} \left| \sum_{n=0}^{M-1} x^{(i)}[n] \cdot w[n] \cdot e^{-j2\pi kf_\Delta n} \right|^2 \quad i = 1, 2, \dots, K \quad (8.16)$$

Trong đó $x^{(i)}[n]$ là các mẫu trong bản ghi dữ liệu thứ i và $f_\Delta = f_s / M$. Sau đó lấy trung bình K periodograms sẽ tạo bộ ước tính PSD:

$$\hat{S}(kf_\Delta) = \frac{1}{K} \sum_{i=1}^K I_M^{(i)}(kf_\Delta), \quad k = 0, 1, \dots, M-1 \quad (8.17)$$

Tất nhiên, bộ ước tính này là chệch bởi vì bản ghi dữ liệu là hữu hạn. Giả sử rằng K periodograms là độc lập:

$$\text{var}(\hat{S}(kf_\Delta)) = \frac{1}{K} S^2(kf_\Delta) \quad (8.18)$$

dần tới 0 khi $K \rightarrow \infty$.

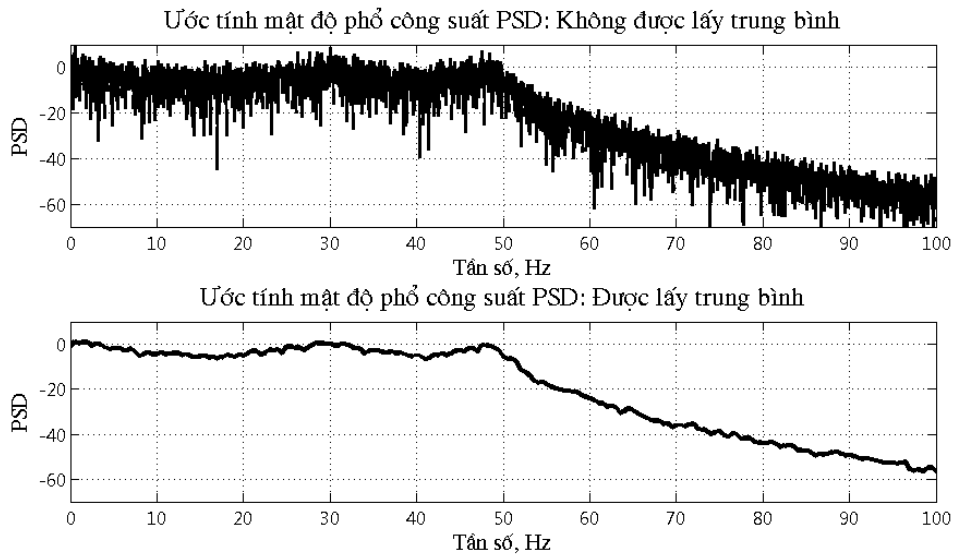
Thấy rõ từ việc so sánh (8.12) và (8.16). Trong khi periodogram được định nghĩa bởi (8.12) có độ phân giải tần số là $f_\Delta = f_s/N$, thì periodogram được định nghĩa bởi (8.16) có độ phân giải tần số là $f_\Delta = f_s/M$. Vì $M < N$ khi $K > 1$, nên khi phân đoạn bản ghi dữ liệu N mẫu ban đầu thì độ phân giải tần số bị suy giảm. Do đó, khi sử dụng kỹ thuật phân đoạn sẽ nảy sinh yêu cầu cần phải dung hoà giữa độ phân giải và phương sai. Cũng vậy, tính hợp lý của (8.18) đòi hỏi K periodogram được sử dụng trong quá trình lấy trung bình là độc lập. Vì mong muốn có giá trị lớn nhất có thể của K với N cố định, nên các đoạn thường bị chồng chập nhau. Thường dùng chồng chập 50%. Tuy nhiên, nếu các đoạn dữ liệu bị chồng chập thì K periodogram không còn độc lập nữa và việc giảm phương sai của bộ ước tính PSD là kém hơn được dự đoán bởi (8.18). Ít nhất từng phần, việc sử dụng cửa sổ dữ liệu giúp cho khôi phục tính độc lập của K đoạn.

Trong khi tồn tại nhiều loại của sổ dữ liệu được dùng trong (8.16), thì cửa sổ Hanning thường được sử dụng để ước tính PSD. Cửa sổ Hanning được định nghĩa bởi

$$w[n] = \begin{cases} \frac{1}{2} \left[1 - \cos\left(\frac{2\pi n}{M-1}\right) \right], & 0 \leq n \leq M-1 \\ 0, & \text{nếu khác} \end{cases} \quad (8.19)$$

Ví dụ 8.3: Trong ví dụ này ta cho các mẫu độc lập (tạp âm trắng) qua bộ lọc Chebyshev có độ gợn sóng băng thông 5 dB. Bài toán là ước tính PSD tại đầu ra bộ lọc. Chương trình Matlab được cho bởi file **NVD8_PSDSim.m** (có trong Phụ lục 8A) thực hiện bài toán này.

Kết quả chạy chương trình mô phỏng **NVD8_PSDSim.m** được cho ở hình 8.9. Cần lưu ý: (i) trường hợp không chồng chéo (hình 8.9 phía trên) phương sai lớn. (ii) Phương sai không phụ thuộc vào tần số. Sử dụng 25 đoạn (hình 8.9 phía dưới) làm cho phương sai nhỏ hơn nhiều với mức chi phí độ phân giải tần số được giảm. (iii) Độ gợn sóng thông dải 5 dB của bộ lọc Chebyshe khi $K = 25$ rõ ràng hơn nhiều khi $K = 1$.

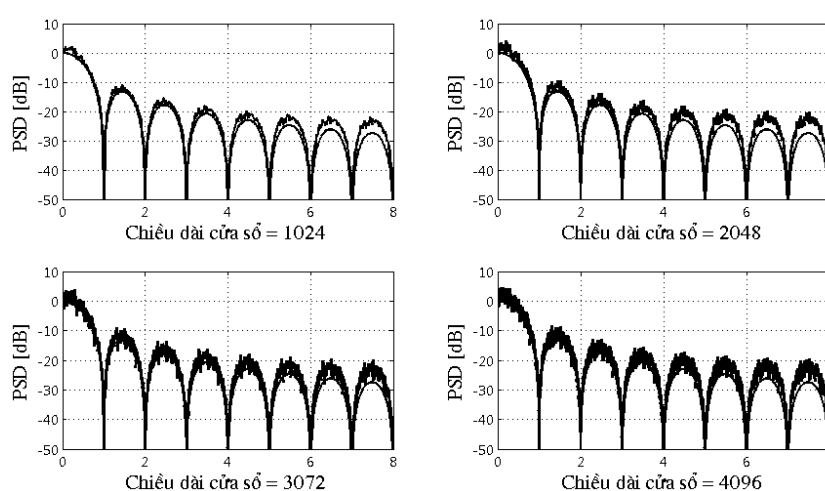


Hình 8.9: Các ước tính PSD: không được lấy trung bình và được lấy trung bình

Ta triển khai bộ ước tính PSD được minh họa trong ví dụ 8.3 bằng cách dùng các lệnh Matlab cơ bản. Hộp công cụ xử lý tín hiệu (*Signal Processing Toolbox*) trong Matlab chứa một số thường trình để ước tính PSD. Hai trong số chúng là hàm **psd** và **pwelch**. Để được rõ hơn, ta nên nghiên cứu kỹ cho một số trường hợp cụ thể. Ở đây ta minh họa periodogram Welch từ hộp công cụ xử lý tín hiệu.

Ví dụ 8.4: Trong ví dụ này, ta ước tính PSD của tín hiệu QPSK. Với giả thiết là: (i) Định dạng xung chữ nhật; (ii) lấy mẫu các thành phần đồng pha và vuông pha của tín hiệu QPSK tại tốc độ 16 mẫu/ký hiệu. Mã chương trình Matlab được cho ở file **NVD8_welchp.m** (có trong Phụ lục 8A)

Chạy chương trình mô phỏng **NVD8_welchp.m** ta được các kết quả được minh họa ở hình 8.10. Lưu ý rằng, chương trình tạo ra 16×1024 điểm và sử dụng các kích thước cửa sổ (*nwindow*) là 1024, 2048, 3072, 4096. Cũng như ví dụ trước khi *nwindow* lớn tạo ra trung bình nhỏ hơn, dẫn đến PSD được ước tính tỏ ra phương sai lớn hơn. Khi *nwindow* nhỏ sẽ làm giảm phương sai ở mức chi chỉ độ phân giải giảm. Những xu hướng này có thể được thấy từ hình 8.10.



Hình 8.10: Các ước tính PSD được tạo ra trong ví dụ 8.4

8.3.3. Độ lợi, trễ và tỷ số tín hiệu trên tạp âm

Tỷ số tín hiệu trên tạp âm SNR là hệ số phẩm chất được sử dụng phổ biến để ước lượng hiệu năng hệ thống truyền thông. Kỹ thuật ước tính SNR được trình bày ở đây được xuất phát từ phương pháp đo các lỗi méo kênh trong các hệ thống đo xa băng rộng, tạp âm trong tín hiệu tại một điểm trong hệ thống được xác định là *lỗi trung bình quân phương* MSE giữa tín hiệu thực tế và tín hiệu mong muốn tại điểm đó. SNR được ước tính bằng cách xác định tín hiệu mong muốn (biên độ được tỉ lệ hóa) và phiên bản bị trễ của tín hiệu mang thông tin tại đầu vào hệ thống. Trước đây đã tính đến việc ứng dụng kỹ thuật này để giám sát độ tin cậy truyền dẫn

dữ liệu điều chế xung mã số và ước tính tỷ số sóng mang trên méo điều chế ký sinh IM trong kênh phi tuyến.

Triển khai lý thuyết cho các tín hiệu thông thấp giá trị thực

Đối với hệ thống tuyến tính bất biến LTIV không méo, thì tín hiệu $y(t)$ tại bất kỳ điểm nào trong hệ thống đều là phiên bản biên độ bị tỷ lệ hóa và phiên bản của tín hiệu tham chiếu đầu vào $x(t)$ bị trễ. Vì vậy ta có thể viết tín hiệu không méo là:

$$z(t) = A \cdot x(t - \tau) \quad (8.20)$$

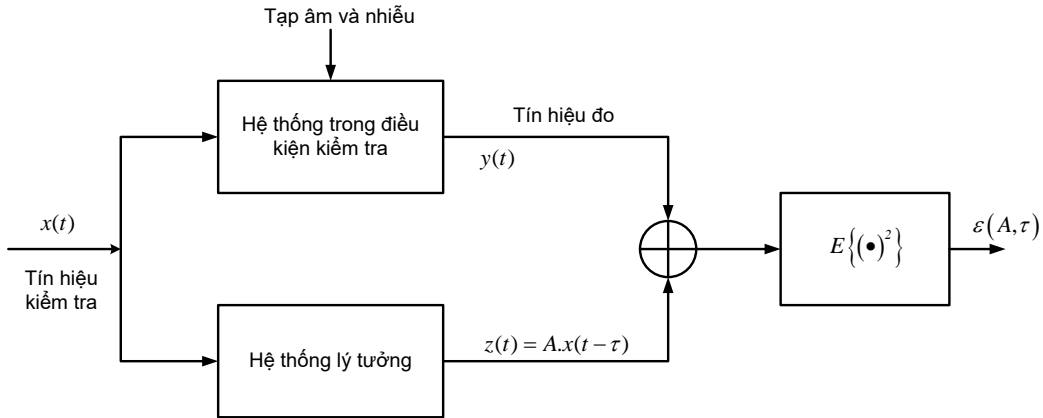
Trong đó A là độ lợi và τ là trễ nhóm đối với điểm trong hệ thống tại đó SNR sẽ được xác định. Lấy $x(t)$ là tín hiệu tham chiếu và $y(t)$ là tín hiệu đo, thì:

$$y(t) = A \cdot x(t - \tau) + n(t) + d(t) \quad (8.21)$$

Trong đó $n(t)$ là tạp âm cộng bên ngoài, $d(t)$ là méo bên trong phụ thuộc tín hiệu do hệ thống gây ra (có thể do ISI hoặc tính phi tuyến hệ thống). Sơ đồ khối mô tả quan hệ giữa các tín hiệu trong hệ thống được cho ở hình 8.11.

Công suất tạp âm được định nghĩa là MSE giữa $y(t)$ và đầu ra của hệ thống không méo $z(t) = A \cdot x(t - \tau)$. Nghĩa là:

$$\varepsilon(A, \tau) = E \left\{ [y(t) - A \cdot x(t - \tau)]^2 \right\} \quad (8.22)$$



Hình 8.11: Các thủ tục kiểm tra để ước tính độ lợi, trễ, SNR

Các ước tính mong muốn đối với A và τ là các giá trị tại đó $\varepsilon(A, \tau)$ được cực tiểu. Biểu thức trên được viết là:

$$\varepsilon(A, \tau) = E \left\{ y^2(t) + A^2 x^2(t - \tau) - 2Ax(t - \tau)y(t) \right\} \quad (8.23)$$

Đối với các tín hiệu dừng, thì các mô men là độc lập với gốc thời gian. Ngoài ra kỳ vọng toán học của một tổng là tổng các giá trị kỳ vọng. Phương trình (8.23) được viết:

$$\varepsilon(A, \tau) = \underbrace{E\{y^2(t)\}}_{P_y} + A^2 \underbrace{E\{x^2(t)\}}_{P_x} - 2A \underbrace{E\{x(t)y(t+\tau)\}}_{R_{xy}(\tau)} \quad (8.24)$$

$$\Leftrightarrow \varepsilon(A, \tau) = P_y + A^2 P_x - 2AR_{xy}(\tau) \quad (8.25)$$

Trong đó P_x và P_y thể hiện các công suất trung bình của $x(t)$ và $y(t)$ tương ứng. Thấy rõ, giá trị τ làm cực tiểu $\varepsilon(A, \tau)$ là giá trị của τ , được ký hiệu là τ_m , làm cực đại hàm tương quan chéo giữa $x(t)$ và $y(t)$, $R_{xy}(\tau)$. Ta gọi đây là trễ thời gian hệ thống. Độ lợi hệ thống A_m là giá trị A để:

$$\frac{d}{dA} \varepsilon(A, \tau_m) = \frac{d}{dA} [P_y + A^2 P_x - 2AR_{xy}(\tau_m)] = 0 \quad (8.26)$$

$$\Rightarrow A_m = \frac{R_{xy}(\tau_m)}{P_x} \quad (8.27)$$

Công suất trong tín hiệu lỗi được tìm thấy từ (8.25) với $A = A_m$ và $\tau = \tau_m$. Vì đây là thành phần của $y(t)$ trực giao với tín hiệu $x(t)$, nên ta xác định công suất trong tín hiệu lỗi là công suất tạp âm N . Vì vậy:

$$N = P_y - \frac{R_{xy}^2(\tau_m)}{P_x} \quad (8.28)$$

Công suất trong thành phần tín hiệu của $y(t)$ là:

$$S = P_z = A_m^2 P_x = \frac{R_{xy}^2(\tau_m)}{P_x} \quad (8.29)$$

Vì vậy, tỷ số tín hiệu trên tạp âm SNR là:

$$\frac{S}{N} = \frac{R_{xy}^2(\tau_m)}{P_x} \left[\frac{P_x}{P_x P_y - R_{xy}^2(\tau_m)} \right] = \frac{R_{xy}^2(\tau_m)}{P_x P_y - R_{xy}^2(\tau_m)} \quad (8.30)$$

Hệ số tương quan ρ liên quan giữa $x(t)$ và $y(t)$ được định nghĩa là:

$$\rho = \frac{R_{xy}(\tau_m)}{\sqrt{P_x P_y}} \quad (8.31)$$

Với định nghĩa này, SNR tại điểm đo $y(t)$ có dạng rất đơn giản là:

$$\frac{S}{N} = \frac{\rho^2}{1 - \rho^2} \quad (8.32)$$

Với những hệ thống có các tín hiệu thực, bài toán này được nghiên cứu bởi Turner, Tranter, Eggleston, Jeruchim và Wolfe. Việc làm cực tiểu $\varepsilon(A, \tau)$ hay đồng nghĩa với việc cực đại hóa hàm tương quan chéo $R_{xy}(\tau)$ giữa tín hiệu tham chiếu $x(t)$ và tín hiệu đo $y(t)$.

Chương trình Matlab thực hiện bộ hậu xử lý để ước tính độ lợi hệ thống, trễ và SNR được cho bởi file **NVD8_snrnmse.m** (có trong Phụ lục 8A).

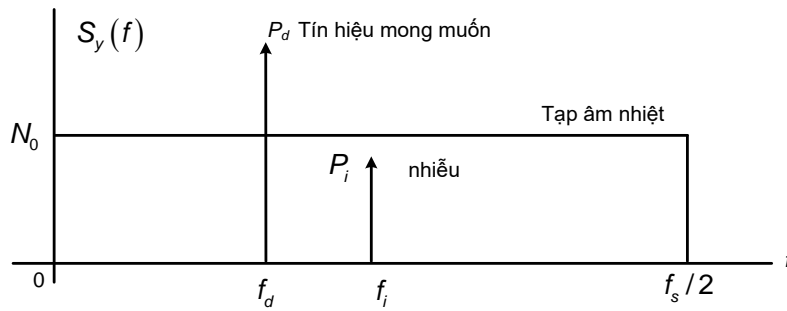
Tại đây, ta xét một ví dụ đơn giản (lưu ý: kỹ thuật được dùng ở đây để ước tính trễ sẽ được dùng trong chương 10 ở đó xét bài toán mô phỏng bán giải tích).

Ví dụ 8.5: Để minh họa các kỹ thuật trên, ta giả sử $x(t)$ là tín hiệu hình sin:

$$x(t) = A.\sin(2\pi f_d t) \quad (8.33)$$

Và tín hiệu đo là:

$$y(t) = G.A.\sin(2\pi f_d t + \phi) + B.\sin(2\pi f_i t) + \sigma_n n(t) \quad (8.34)$$



Hình 8.12: PSD một phía của tín hiệu đo $y(t)$

Trong đó G là độ lợi hệ thống, $n(t)$ là quá trình tạp âm Gauss trắng phương sai đơn vị trung bình 0 và σ_n là độ lệch chuẩn của quá trình tạp âm cộng. PSD của $y(t)$ được minh họa ở hình 8.12, trong đó P_d là công suất tín hiệu (công suất thành phần mong muốn), P_i là công suất nhiễu và N_0 là mật độ phổ công suất một phía của thành phần tạp âm. Từ chương 7 cho thấy, quan hệ giữa N_0 và σ_n là:

$$\sigma_n^2 = N_0 \frac{f_s}{2} \quad (8.35)$$

trong đó f_s là tần số lấy mẫu.

Đối với ví dụ này, tín hiệu tham chiếu được xác định bởi:

$$x(t) = 80.\sin[2\pi(2)t] \quad (8.36)$$

Tín hiệu tại đầu vào máy thu được giả định là:

$$y(t) = 20.\sin\left[2\pi(2)t - \frac{\pi}{4}\right] + 4.\sin[2\pi(8)t] + 0,8.n(t) \quad (8.37)$$

Trong đó $n(t)$ là hàm mẫu của quá trình tạp âm phương sai đơn vị trung bình 0. Mã chương trình Matlab cho kịch bản này được cho bởi file **NVD8_snrnsim.m** (có trong Phụ lục 8A).

Chạy chương trình tạo ra các kết quả sau:

Giá trị của P_x là 3200

Giá trị của P_y là 208,7872

Giá trị độ lợi là 0,25012

Giá trị trễ là 64

Giá trị SNR là 13,6728 dB

Các giá trị lý thuyết dễ dàng được tính toán. Vì tín hiệu tham chiếu là hình sin có giá trị đỉnh là 80 nên:

$$P_x = \frac{I}{2}(80)^2 = 3200 \quad (8.38)$$

Từ (8.37) ta có ba thành phần ở đầu vào máy thu là: tín hiệu sin tại 2 Hz, nhiễu hình sin tại 8 Hz, tạp âm trắng. Công suất P_y là tổng của 3 thành phần:

$$P_y = \frac{I}{2}(20)^2 + \frac{I}{2}(4)^2 + (0,8)^2 = 208,64 \quad (8.39)$$

Từ (8.36) và (8.37) ta có độ lợi là:

$$G = \frac{20}{80} = 0,25 \quad (8.40)$$

Lưu ý rằng, thành phần tín hiệu có chu kỳ 512 mẫu ($x(t)$ trải qua 2 chu kỳ là 1024 mẫu), và trễ pha $\pi/4$, độ trễ là:

$$\tau = \frac{\pi/4}{2\pi}(512) = 64 \text{ mẫu} \quad (8.41)$$

SNR là tỷ số của công suất nhiễu cộng với công suất tạp âm trên tín hiệu tại đầu vào máy thu. Đây là tỷ số của số hạng đầu trong (8.39) với tổng của 2 số hạng sau (lưu ý trong trường hợp này nhiễu chính là tạp âm vì nhiễu trực giao với thành phần tín hiệu). Ta có:

$$\frac{S}{N} = \frac{200}{8,64} = 23,1481 = 13,6452 \text{ dB} \quad (8.42)$$

Các kết quả này được tóm tắt trong bảng 8.2. Các lỗi nhỏ là do phương sai tạp âm σ_n^2 là biến ngẫu nhiên vì chiều dài bản ghi là hữu hạn. Nâng cấp chương trình sao cho 5 ước tính của SNR (đơn vị dB) là các kết quả được tạo ra ở dạng vector:

[13,6572 13,6524 13,5016 13,5245 13,5201]

Thấy rõ, SNR được ước tính là một biến ngẫu nhiên.

Bảng 8.2: Tóm tắt các kết quả cho ví dụ 8.2

Tham số	Giá trị lý thuyết	Giá trị ước tính
P_x	3.200	3.200
P_y	208,64	208,7872
G	0,25	0,25012
τ	64 mẫu	64 mẫu

S/N	13,6452 dB	13,6728 dB
-----	------------	------------

Khó khăn lớn nhất đối với phương pháp này là xác định chính xác trễ. Lưu ý rằng, nếu xảy ra lỗi nhỏ trong ước tính trễ, thì cũng gây ra lỗi nhỏ trong giá trị được ước tính của $R_{xy}^2(\tau_m)$ nhỏ. Nếu SNR là lớn, thì sẽ dẫn đến kết quả là $P_x P_y - R_{xy}^2(\tau_m) \lesssim 0$ và, ta thấy từ (8.30) chỉ một lỗi nhỏ trong ước tính của $R_{xy}^2(\tau_m)$ sẽ gây ra lỗi lớn trong SNR được ước tính. Theo đó, ta phải xác định chính xác giá trị đỉnh của $R_{xy}^2(\tau)$. Điều này cần phải lấy mẫu $R_{xy}^2(\tau)$ nhanh hơn, hay tần số lấy mẫu cao khi mô phỏng. Vì vậy, ta phải dung hoà mọi điểm trong mô hình mô phỏng giữa độ chính xác và thời gian mô phỏng.

Trong quá trình triển khai, ta đã giả định rằng các tín hiệu là các tín hiệu giá trị thực. Kỹ thuật được trình bày ở đây có thể áp dụng cho đường bao phức. Bộ ước tính cho trường hợp này được rút ra bằng cách thay $x(t)$ bằng $x_d(t) + jx_d(t)$ và $y(t)$ bằng $y_d(t) + jy_d(t)$. Biểu thức kết quả đối với SNR cũng là (8.32).

8.4. Mã hoá

Nếu mô phỏng được dùng để xác định BER trong trường hợp hệ thống có sử dụng mã hóa kiểm soát lỗi, thì thường không đếm số lỗi tại đầu ra *bộ giải mã*. Một số lý do cơ bản sau: (i) BER tại đầu ra bộ giải mã thường rất nhỏ; (ii) thời gian chạy mô phỏng rất lâu mới tập hợp số lỗi đủ lớn để các ước tính BER được chính xác; (iii) tính phức tạp của nhiều thuật toán giải mã làm tăng đáng kể thời gian chạy mô phỏng; (iv) cả bộ mã hoá và giải mã đều là thiết bị tất định. Một khi mã được xác định thì dữ liệu nguồn xác định duy nhất các từ mã đầu ra bộ mã hóa. Tương tự mẫu lỗi đầu ra máy thu xác định duy nhất BER đầu ra bộ giải mã. Điều này gọi ra giải pháp bán giải tích, trong đó tỉ số lỗi ký hiệu SER tại đầu vào máy thu được xác định bằng mô phỏng, được chuyển thành BER bằng phương pháp phân tích. Ở dạng tổng quát, việc ánh xạ này là một nhiệm vụ phức tạp nếu muốn có các kết quả chính xác. May thay, các kết quả chính xác là hiếm khi cần đến, và một số xấp xỉ và giới hạn đã được triển khai để đơn giản hóa nhiệm vụ này.

Mô phỏng mức dạng sóng thường được sử dụng để xác định tỷ số lỗi ký hiệu SER tại đầu vào máy thu. Một phương pháp thay thế là sử dụng các mô hình kênh rời rạc như các mô hình Markov HMM. HMM là kỹ thuật tính toán hiệu quả để mô phỏng các hệ thống khi cho trước tập các điều kiện kênh, vì vậy là rất hữu hiệu để nghiên cứu ảnh hưởng của các thuật toán mã hoá/giải mã khác nhau. Mô hình kênh rời rạc và HMM sẽ được nghiên cứu chi tiết ở chương 15.

Lưu ý rằng, để được rõ hơn về hoạt động của mã hóa kênh kiểm soát lỗi bạn đọc có thể tham khảo phần phụ lục 8B1. Ở đó trình bày khá chi tiết và dễ hiểu cho quá trình mã hóa và giải mã cũng như hiệu năng của mã, đặc biệt có ý nghĩa đối với sinh viên.

8.4.1. Giải pháp giải tích cho mã khối

Ta biết rằng, các mã khối được tạo ra bằng cách nhóm các ký hiệu thông tin thành các khối có độ dài k . Với mỗi khối k ký hiệu được gán vào $(n-k)$ ký hiệu kiểm tra, tạo từ mã dài n . Những từ mã này được truyền qua kênh, dưới tác động của kênh gây ra các lỗi ngẫu nhiên.

Trong hầu hết các ứng dụng thực tế, các từ mã n ký hiệu được truyền trong một khe thời gian lâu kT_b , trong đó T_b là thời gian để truyền một bit thông tin không mã hoá. Nếu công suất phát là như nhau khi có và không có mã hoá, thì năng lượng tương ứng để truyền các ký hiệu mã là $(k/n)E_b$, trong đó E_b là năng lượng trên bit và k/n là tỷ lệ mã. Vì năng lượng trên ký hiệu phát bị giảm do sử dụng mã hóa kiểm soát lỗi, nên xác suất lỗi ký hiệu kênh có mã hoá bị tăng so với xác suất lỗi ký hiệu (bit) không mã hoá. Hy vọng rằng, thông qua việc thêm các ký hiệu kiểm tra tạo khả năng sửa lỗi hiệu quả để tăng hiệu năng hệ thống. Điều này có thể đúng hoặc không đúng.

Giả sử: (i) một mã có thể sửa t lỗi trong mỗi khối n ký hiệu; (ii) các sự kiện lỗi là độc lập (có thể được đảm bảo bằng cách sử dụng kỹ thuật đan xen). Xác suất lỗi liên quan với các ký hiệu mã được phát qua kênh ký hiệu là P_{sc} , trong đó chỉ số dưới thể hiện các ký hiệu kênh. Vì bộ giải mã có thể sửa được t lỗi trong một từ mã n ký hiệu nên xác suất mà từ mã được giải mã sẽ trong trạng thái lỗi P_{cw} là:

$$P_{cw} \leq \sum_{i=t+1}^n \binom{n}{i} P_{sc}^i (1 - P_{sc})^{n-i} \quad (8.43)$$

Dấu bằng xảy ra trong (8.43) nếu tất cả các khối n ký hiệu thu chứa t lỗi hoặc ít hơn t lỗi được giải mã chính xác và không có khối n ký hiệu nào chứa $t+1$ hoặc nhiều hơn $t+1$ lỗi được giải mã chính xác, khi này được biết như là các mã hoàn hảo. Chỉ các mã nhị phân hoàn hảo là các mã lập đối với n là lẻ, các mã Hamming sửa đơn lỗi và mã Golay (23,12) sửa ba lỗi. Đối với tất cả các mã khác, biểu thức (8.43) cung cấp một giới hạn hữu hiệu.

Xác suất lỗi từ mã được giải mã P_{cw} không cho phép trực tiếp so sánh các mã khác nhau. Để so sánh các mã khác nhau ta cần ánh xạ xác suất lỗi từ mã được giải mã thành xác suất lỗi bit thông tin được giải mã, ký hiệu là P_b . Một ánh xạ chính xác là một hàm ma trận tạo mã của mã đó mà nó xác định phân bố trọng số mã. Nhiều công trình đã triển khai phép xấp xỉ chính xác cao. Phép xấp xỉ này là:

$$P_b \approx \frac{q}{2(q-1)} \left[\frac{d}{n} \sum_{i=t+1}^n \binom{n}{i} P_{sc}^i (1 - P_{sc})^{n-i} + \frac{1}{n} \sum_{i=d+1}^n \binom{n}{i} P_{sc}^i (1 - P_{sc})^{n-i} \right] \quad (8.44)$$

Trong đó q ký hiệu kênh q tia (q -ary). Nói cách khác, với các kênh nhị phân thì $q = 2$, và với các mã Reed-Solomon mã khối phi nhị phân phổ biến nhất $q = 2^k - 1$.

Ví dụ 8.6: Ta minh họa việc sử dụng (8.44). Giả sử hệ thống truyền thông BPSK ($q = 2$) hoạt động trong môi trường kênh AWGN. Khi này, xác suất lỗi bit không mã hoá là:

$$P_b = Q(\sqrt{2z}) = Q\left(\sqrt{2 \frac{E_b}{N_0}}\right) \quad (8.45)$$

Trong đó z biểu diễn cho E_b/N_0 . Với mã khối (n, k) , xác suất lỗi ký hiệu kênh là:

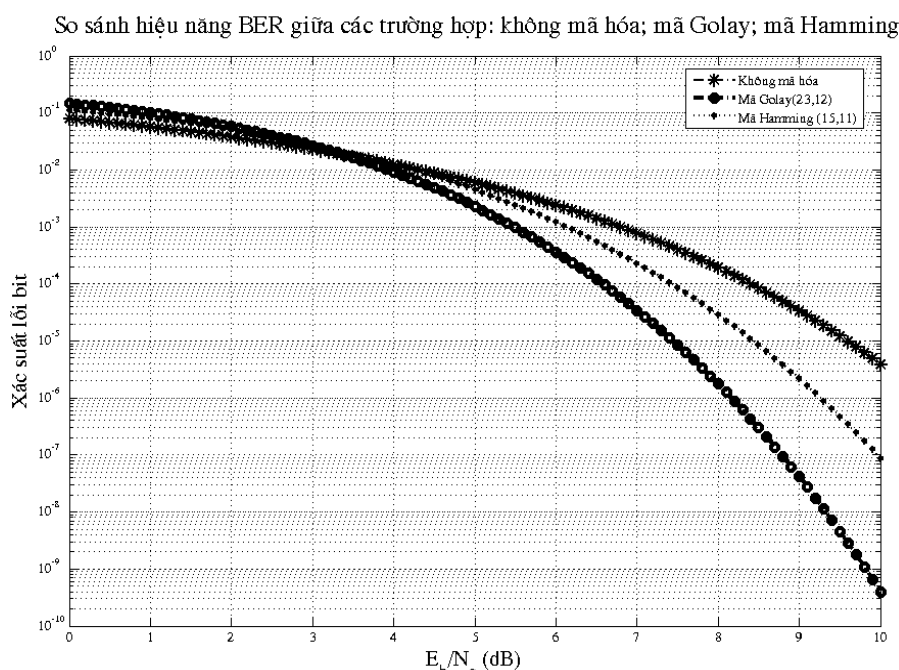
$$P_{sc} = Q\left(\sqrt{\frac{k}{n} 2z}\right) = Q\left(\sqrt{\frac{k}{n} \times 2 \frac{E_b}{N_0}}\right) \quad (8.46)$$

Xét hai mã nhị phân khác nhau là: mã Golay (23,12) trong đó $n = 23$, $t = 3$ và $d = 7$, và mã Hamming (15,11) trong đó $n = 15$, $t = 1$ và $d = 3$.

Tại điểm này tất cả các tham số và các biến số trong (8.44) đều được biết. Tuy nhiên trước khi ước lượng (8.44) ta phải ước lượng:

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \quad (8.47)$$

Chương trình Matlab thực hiện (8.47) được cho ở file **NVD8_nkchoose** (có trong Phụ lục 8A).



Hình 8.13: So sánh hiệu năng các mã khối Golay và Hamming

Chương trình Matlab để tính toán các đường cong hiệu năng cho mã Hamming (15,11) và mã Golay (23,12) sửa ba lỗi được cho ở file **NVD8_cersim** (có trong Phụ lục 8A). Cần lưu ý rằng, ta đã giả định kênh AWGN và điều chế PSK.

Cần lưu ý rằng, trong chương trình **NVD8_cersim.m** ta đã dùng chương trình con **NVD8_cer2ber** dưới đây để chuyển xác suất lỗi ký hiệu kênh thành xấp xỉ xác suất lỗi bit giải mã được cho bởi (8.44).

Kết quả của các tính toán này được minh họa ở hình 8.13

8.4.2. Giải pháp giải tích cho mã xoắn

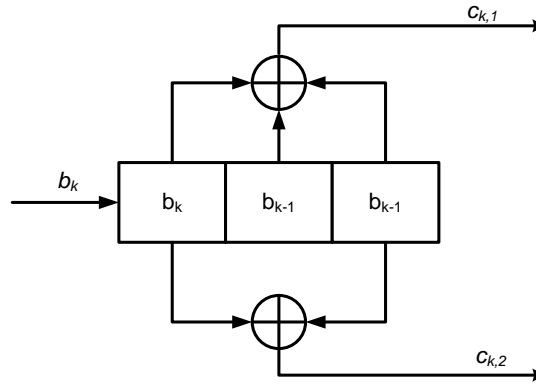
Có thể dùng một số xấp xỉ giải tích để ánh xạ xác suất lỗi ký hiệu kênh thành xác suất lỗi bit được giải mã đối với trường hợp mã xoắn. Những ánh xạ này lấy giới hạn trên của xác suất lỗi và vì vậy tương đương mã xoắn (8.44). Những giới hạn này thường dựa vào thuật toán giải mã Viterbi, gần như tiệm cận với hiệu năng bộ giải mã khả năng giống nhất ML và là tiêu chuẩn để giải mã xoắn.

Giới hạn được sử dụng thường dựa vào hàm truyền đạt của mã xoắn. Hàm truyền đạt mô tả các tính chất khoảng cách của mã xoắn và có thể được rút ra từ biểu đồ chuyển dịch trạng thái của mã. Hàm truyền đạt của bộ lập mã xoắn tỷ lệ 1/2 hình 8.14 được cho bởi:

$$T(D, L, I) = \frac{D^5 \cdot L^3 \cdot I}{1 - D \cdot L \cdot (1 + L) \cdot I} \quad (8.48)$$

Biểu diễn (8.48) ở dạng đa thức là:

$$T(D, L, I) = \sum_{k=0}^{\infty} D^{5+k} \cdot L^{3+k} \cdot (1+L)^k \cdot I^{1+k} = D^5 \cdot L^3 \cdot I + D^6 \cdot L^4 \cdot (1+L) I^2 + \dots \quad (8.49)$$



Hình 8.14: Bộ lập mã xoắn tỷ lệ 1/2 cho ví dụ 8.6

Mô tả các tính chất khoảng cách của các đường dẫn khác nhau trong biểu đồ lưới của mã đó mà bắt đầu tại trạng thái toàn 0 và hợp nhất lại trạng thái toàn 0 sau đó. Số mũ của D thể hiện khoảng cách Hamming (số các số 1 nhị phân) phân cách đường dẫn đã cho so với đường dẫn toàn 0 trong lưới giải mã. Số mũ của L cho biết chiều dài của một đường dẫn đã cho. Nói cách khác, lũy thừa của L được tăng lên mỗi khi đi qua một nhánh trong lưới. Số mũ của I được tăng lên nếu sự chuyển dịch nhánh do đầu vào bit 1 nhị phân, và không được tăng nếu sự chuyển dịch nhánh là kết quả từ đầu vào là bit 0. Ví dụ, số hạng $D^5 L^3 I$ biểu diễn cho một đường dẫn có khoảng cách Hamming là 5 so với đường dẫn toàn 0. Đường dẫn này có độ dài là 3 và kết quả từ dữ liệu đầu vào có một bit nhị phân bằng 1 và hai bit nhị phân 0 (tức là 100). Số hạng tiếp theo, $D^6 L^4 (1+L) I^2 = D^6 L^4 I^2 + D^6 L^5 I^2$, biểu diễn cho 2 đường dẫn, mỗi đường có khoảng cách Hamming là 6 so với đường dẫn toàn 0. Một đường có chiều dài 4 nhánh và đường kia có chiều dài 5 nhánh. Đường có chiều dài 4 nhánh là kết quả từ đầu vào 2 bit 1 và 2 bit 0, và đường kia là kết quả từ đầu vào có 2 bit 1 và 3 bit 0. Trọng số đầu ra nhỏ nhất của tất cả các đường dẫn bắt đầu và hợp nhất với trạng thái tất cả bit 0 thể hiện khoảng cách tự do nhỏ nhất d_f của mã, trong trường hợp này là 5.

Với mục đích xấp xỉ hóa xác suất lỗi bit giải mã. Trước hết, ta đặt $L = 1$ trong (8.48), vì ta không quan tâm đến chiều dài đường dẫn. Nên có:

$$T(D, I) = T(D, L, I)|_{L=1} = \frac{D^5 \cdot I}{1 - 2D \cdot I} \quad (8.50)$$

Đối với tín hiệu PSK đối cực trong môi trường AWGN, thì xác suất lỗi ký hiệu giải mã được cho bởi:

$$P_E < R \frac{\partial T(D, I)}{\partial I} \Big|_{I=1, D=e^{RE_b/N_0}} \quad (8.51)$$

Trong đó R là tỷ lệ mã (trường hợp này $R = 1/2$). Kết quả này được sử dụng trong ví dụ 8.6. Đối với kênh đối xứng nhị phân tổng quát, thì giới hạn Bhattachayya được dùng. Điều này dẫn đến:

$$P_E = \frac{\partial T(D, I)}{\partial I} \Big|_{I=1, D=d} \quad (8.52)$$

Trong đó:

$$d = \sqrt{4q(1-q)} \quad (8.53)$$

Và q là xác suất lỗi ký hiệu kênh được xác định bởi mô phỏng. Các giới hạn được định nghĩa bởi (8.51) và (8.52) có thể không khá chặt chẽ. Điều này đặc biệt đúng với các mã chiều có dài hạn chế ngắn.

Các phương trình (8.51) và (8.52) được giả định cho giải mã quyết định cứng. Với giải mã quyết định mềm, thì d trong (8.51) được thay bằng d_0 .

Trong đó:

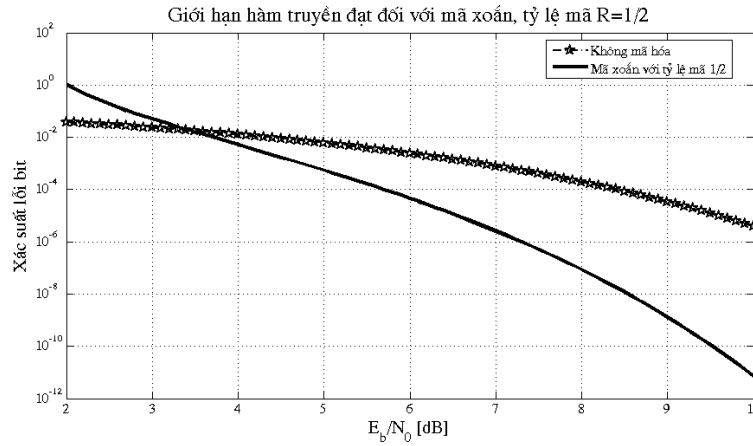
$$d_0 = \sum_{i=1}^N \sqrt{Pr(y_i / 0) Pr(y_i / 1)} \quad (8.54)$$

N là số các giá trị đầu ra bộ lượng tử hoá, y_i là giá trị đầu ra được lượng tử hoá thứ i , và các xác suất có điều kiện biểu diễn cho xác suất của các bit 0 và 1 tại đầu vào kênh sẽ xuất hiện tại đầu ra bộ lượng tử hoá là mức y_i . Các xác suất có điều kiện này được ước tính bằng kỹ thuật Monte Carlo hoặc bán phân tích trên kênh dạng sóng.

Ví dụ 8.7: Ta áp dụng (8.51) cho mã xoắn có tỷ lệ mã là 1/2 được định nghĩa bởi (8.50) và được minh hoạ trong hình 8.14. Thay (8.50) vào (8.52) tạo ra:

$$P_E < \frac{1}{2} \frac{\partial}{\partial I} \left(\frac{D^5 I}{1-2DI} \right) \Big|_{I=1, D=e^{RE_b/N_0}} = \frac{1}{2} \frac{D^5}{(1-2D)^2} \Big|_{D=e^{RE_b/N_0}} \quad (8.55)$$

Mã chương trình Matlab thực hiện bài toán này được cho ở file **NVD8_convcode.m** (có trong Phụ lục 8A). Kết quả chạy chương trình Matlab được cho ở hình 8.15 trong đó so sánh giới hạn xác suất lỗi giải mã và xác suất lỗi không được giải mã.

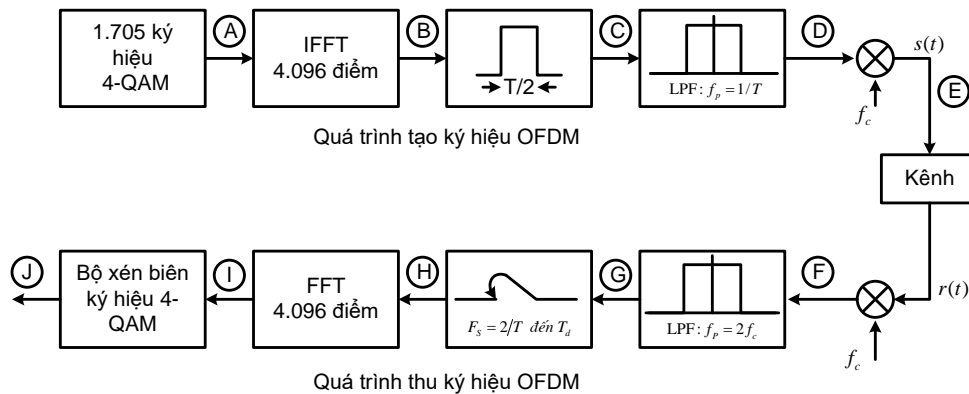


Hình 8.15: Giới hạn hàm truyền đạt đối với ví dụ mã xoắn tỷ lệ mã 1/2

8.5. Một số ứng dụng điển hình

Chương trình 1: Mô phỏng tín hiệu và hệ thống OFDM

Chương trình mô phỏng tín hiệu OFDM được cho ở file **NVD8_ofdmsimtx.m** và **NVD8_ofdmsimrx.m** trong Phụ lục 8A, mô phỏng tín hiệu OFDM trong miền thời gian và tần số. Mặc dù mục đích ở đây là trình bày cách dùng các thường trình hậu xử lý để trực quan hóa tín hiệu và hệ thống trong mô phỏng, nhưng chương trình này rất hữu hiệu để tìm hiểu cách thiết kế và thực hiện mô phỏng hệ thống OFDM đơn giản. Từ chương trình này cho phép nâng cấp để tạo chương trình mô phỏng hệ thống phức tạp hơn. Trình bày chi tiết về việc thiết kế và mô phỏng tín hiệu OFDM được cho ở Phụ lục 8B2. Để tiện theo dõi, sơ đồ mô phỏng quá trình phát thu tín hiệu OFDM được minh họa ở hình 8.16.



Hình 8.16: Sơ đồ mô phỏng quá trình tạo và thu tín hiệu OFDM

Ta mô phỏng tín hiệu OFDM có cấu trúc khung và đa khung. Theo đó, mỗi khung có độ lâu T_F gồm N_F ký hiệu OFDM, mỗi đa khung gồm M_F khung ($T_{MF} = M_F \times T_F$). Mỗi ký hiệu được tạo thành bởi một tập K (từ K_{min} đến K_{max}) sóng mang và được truyền trong khoảng thời gian T_S ; $T_S = T_U + \Delta$. Các thông số cụ thể được cho ở bảng 8.3. Cũng cần lưu ý rằng, để được tường minh ta nên đọc kỹ Phụ lục 8B2 và các chương trình mô phỏng. Ở đó trình bày khá rõ về quá

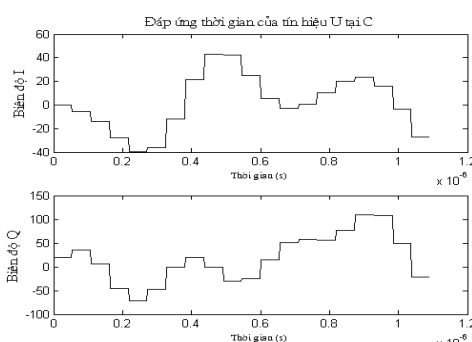
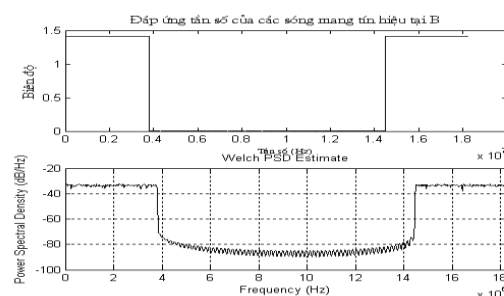
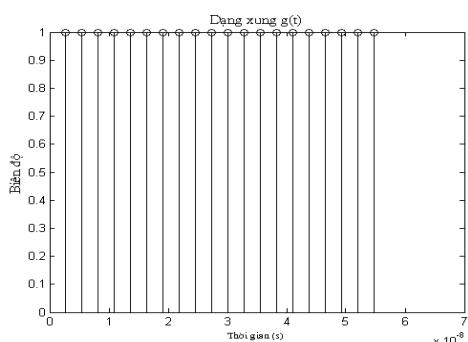
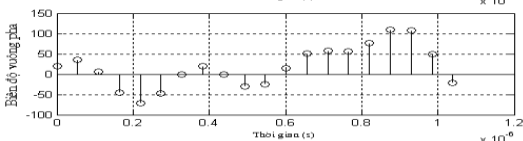
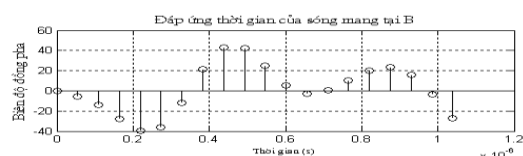
trình thiết kế và mô phỏng cho trường hợp đơn giản điển hình này. Các kết quả mô phỏng tương ứng với các điểm được đánh nhãn trên sơ đồ mô phỏng 8.16.

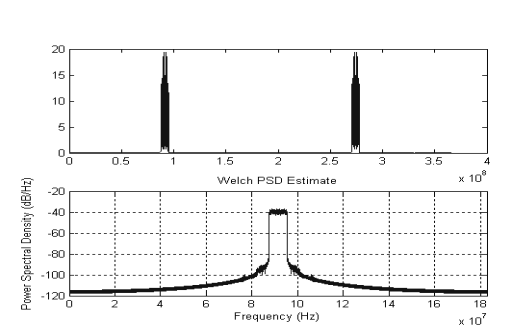
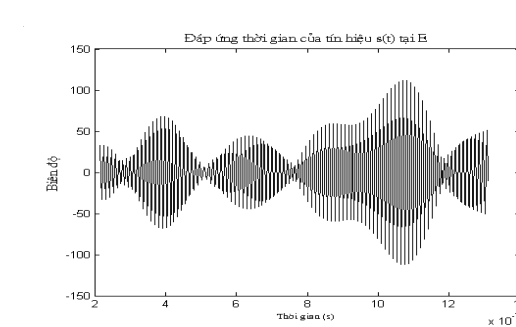
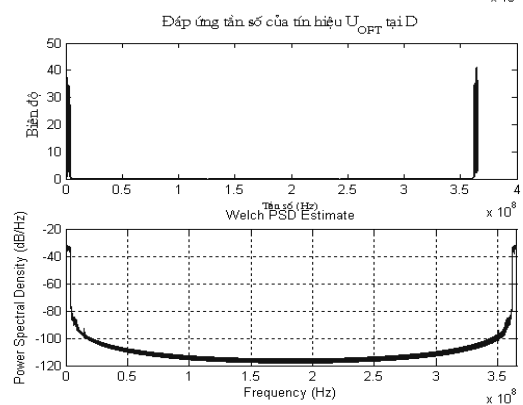
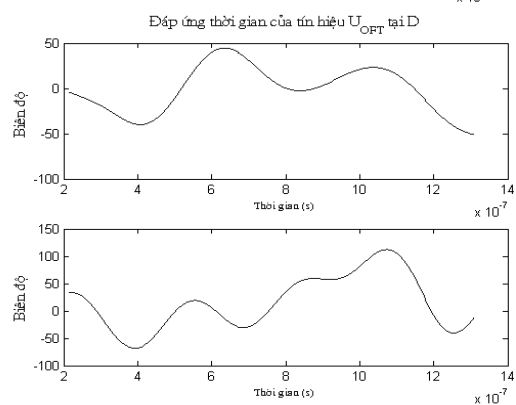
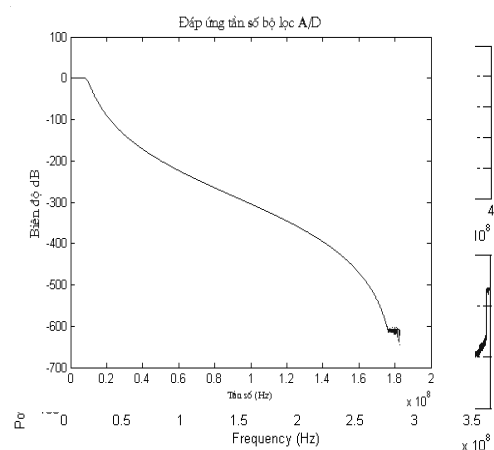
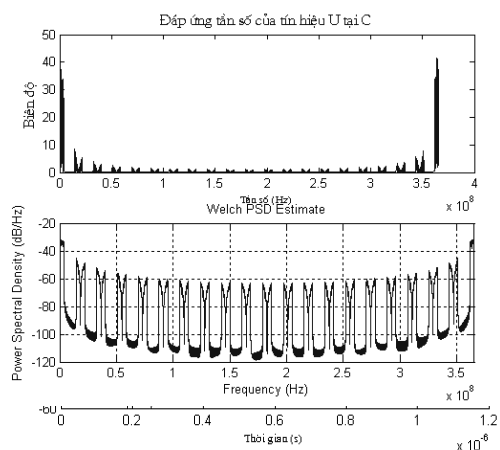
Bảng 8.3: Các giá trị đối với các thông số OFDM trong mô phỏng

Thông số	Giá trị			
Chu kỳ cơ bản T	$7/64 \mu\text{s}$			
Số sóng mang K (trong khoảng K_{\min} đến K_{\max})	1705			
Giá trị số sóng mang K_{\max}	1704			
Giá trị số sóng mang K_{\min}	0			
Khoảng thời gian ký hiệu hữu hiệu T_U	$224 \mu\text{s}$			
Khoảng cách sóng mang $1/T_U$	4464 Hz			
Khoảng cách giữa các sóng mang K_{\min} và $K_{\max}(K - 1)T_U$	7,61 MHz			
Khoảng thời gian bảo vệ được phép Δ/T_U	1/4	1/8	1/16	1/32
Khoảng thời gian phần T_U	$2048 \times T = 224 \mu\text{s}$			
Khoảng thời gian bảo vệ Δ	$512 \times T = 56 \mu\text{s}$	$256 \times T = 28 \mu\text{s}$	$128 \times T = 14 \mu\text{s}$	$64 \times T = 7 \mu\text{s}$
Khoảng thời gian ký hiệu $T_S = \Delta + T_U$	$2.560 \times T = 280 \mu\text{s}$	$2.304 \times T = 252 \mu\text{s}$	$1.176 \times T = 238 \mu\text{s}$	$2.112 \times T = 231 \mu\text{s}$

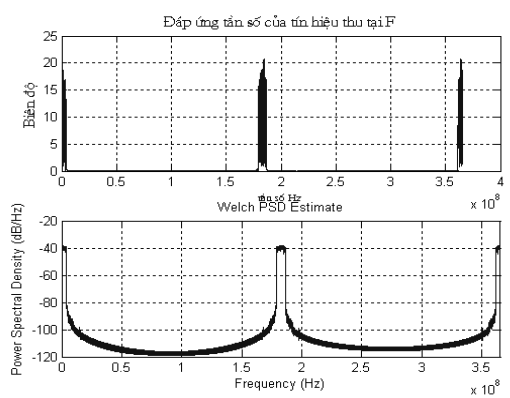
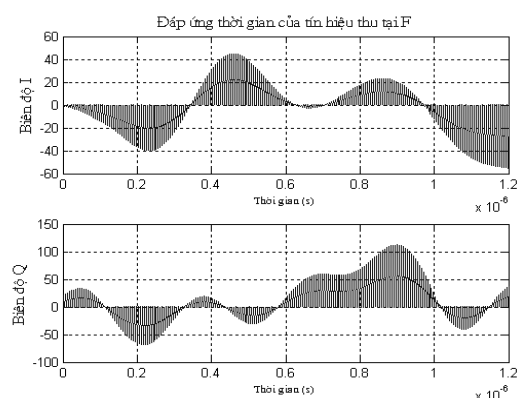
Sử dụng các chương trình thường trình hậu xử lý để trình bày các tín hiệu tại các điểm khác nhau trong sơ mô phỏng như sau:

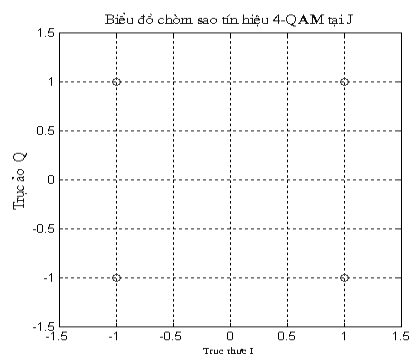
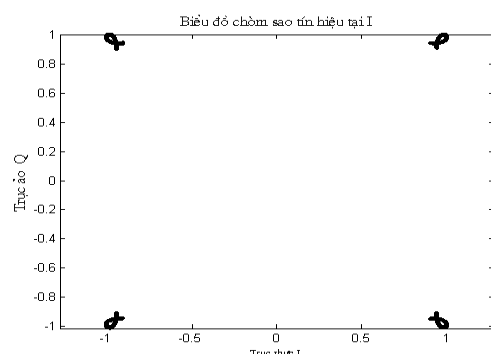
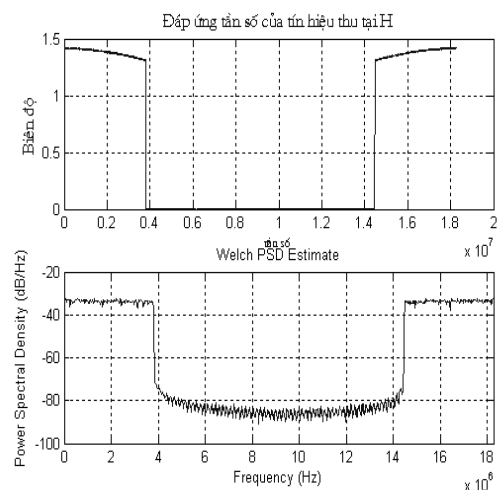
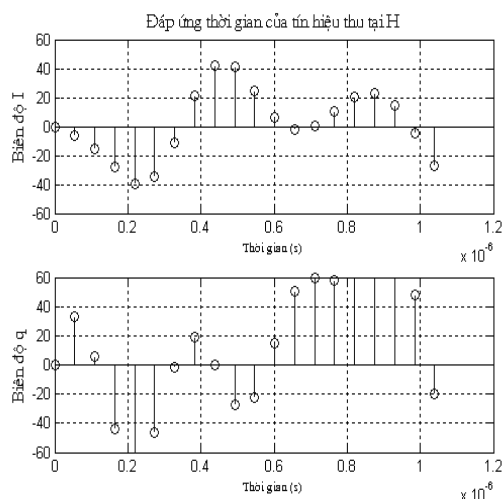
Kết quả mô phỏng phần phát





Kết quả mô phỏng phân thu



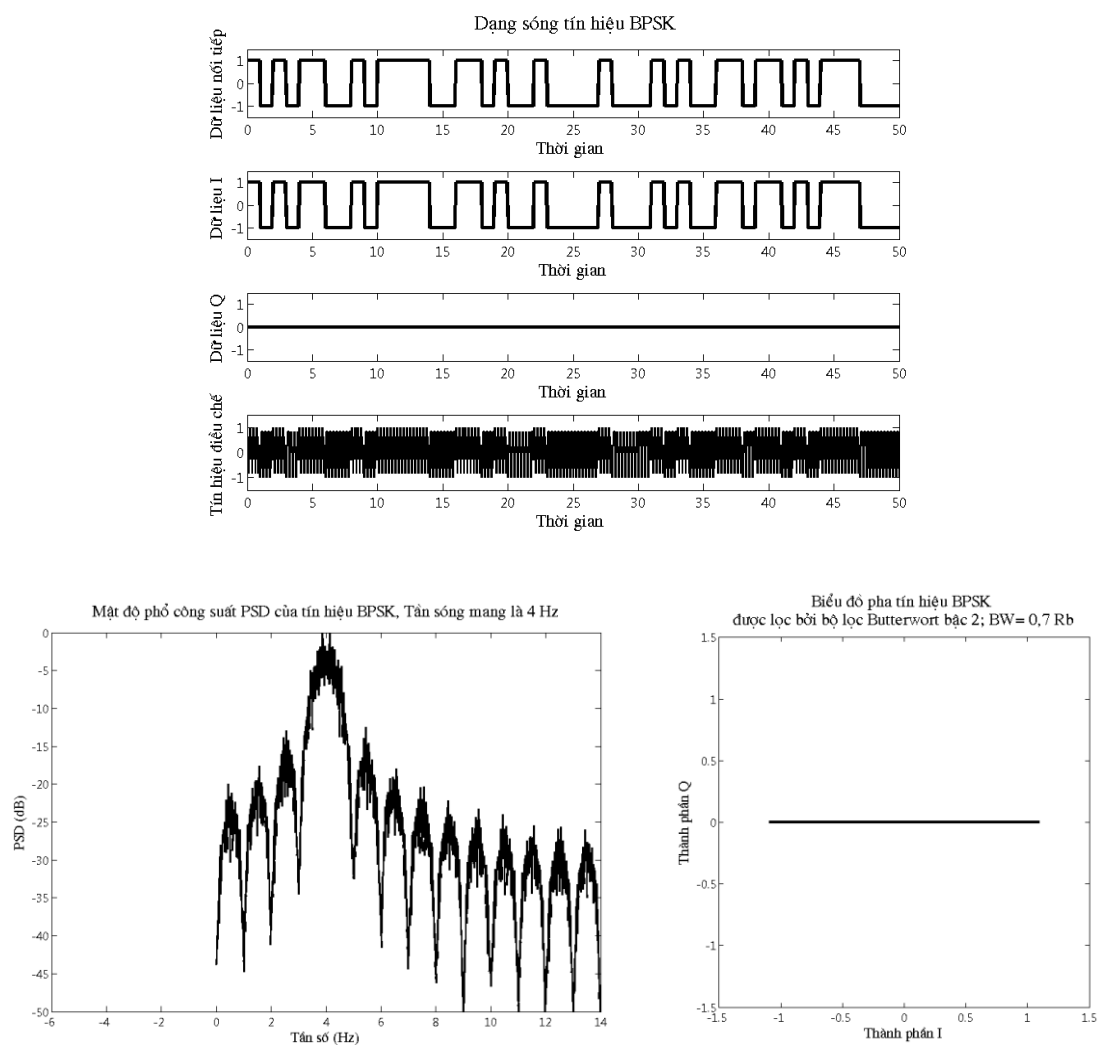


Chương trình 2: Mô phỏng tín hiệu và hệ thống BPSK, QPSK, OQPSK, MSK, 16QAM

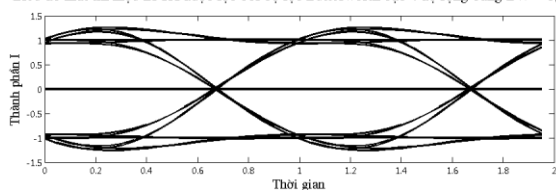
Chương trình mô phỏng tín hiệu và hệ thống BPSK, QPSK, OQPSK, MSK, 16QAM được cho ở file **NVD8_sum1.m** trong Phụ lục 8A thực hiện mô phỏng tín hiệu trong miền thời gian và tần số như: dạng sóng, biểu đồ mắt, biểu đồ pha (chòm sao), đường bao, mật độ phổ công suất PSD. Dưới đây là một số kết quả mô phỏng điển hình cho các hệ thống này. Để được tường minh, ta nên nghiên cứu kỹ mã chương trình Matlab, xác định các tham số đầu vào cho từng hệ thống, vẽ lưu đồ mô phỏng. Đây là một chương trình tổng hợp có tính liên kết cao. Khi đó, trình bày rõ cách lấy dữ liệu từ các điểm của hệ thống để xử lý hiển thị, trực quan hóa nguyên lý hoạt động của các hệ thống này. Đặc biệt có ý nghĩa đối với sinh viên, người mới lập trình Matlab và muốn tìm hiểu nguyên lý hoạt động các hệ thống truyền thông này. Dưới đây là một số kết quả điển hình khi chạy chương trình mô phỏng. Nếu mục đích là hiểu nguyên lý hoạt động các hệ thống này, thì nên dành thời gian nghiên cứu kỹ mã chương trình Matlab này, vẽ lại mô hình mô phỏng, công thức hóa cho từng đoạn chương trình và chạy chương trình

mô phỏng cho tập các tham số đặc trưng khác nhau cho từng mô hình đó. Có thể nói đây là một bài tập rất tốt.

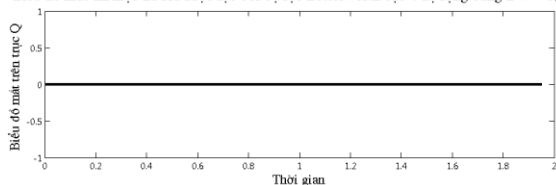
Hệ thống BPSK:



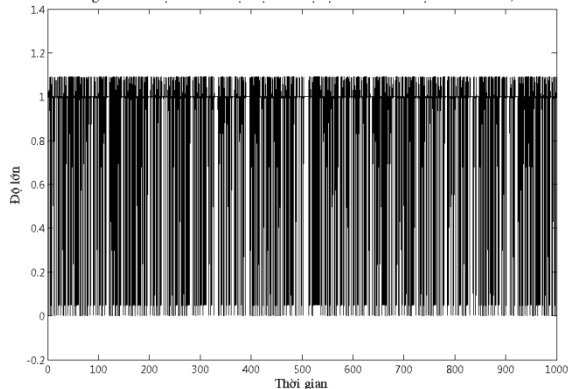
Biểu đồ mắt tín hiệu BPSK được lọc bởi bộ lọc Butterworth bậc 4 độ rộng băng $BW = 0,7 R_b$



Biểu đồ mắt tín hiệu BPSK được lọc bởi bộ lọc Butterworth bậc 4 độ rộng băng $BW = 0,7 R_b$

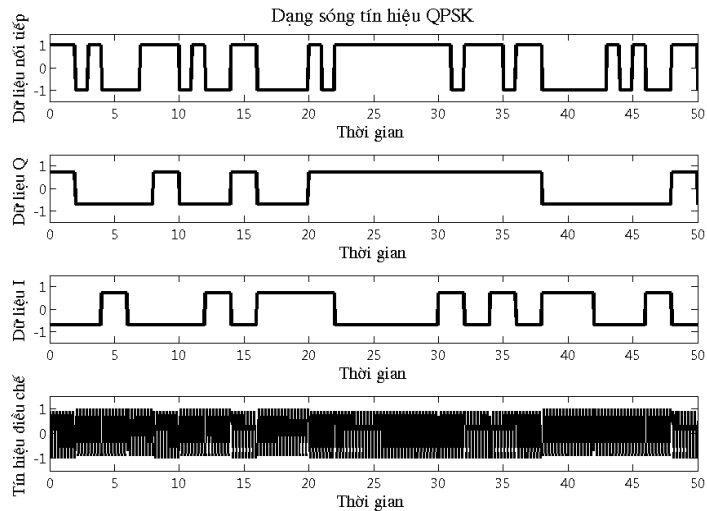


Đường bao tín hiệu BPSK được lọc bởi bộ lọc Butterworth bậc 2 có $BW = 0,7 R_b$

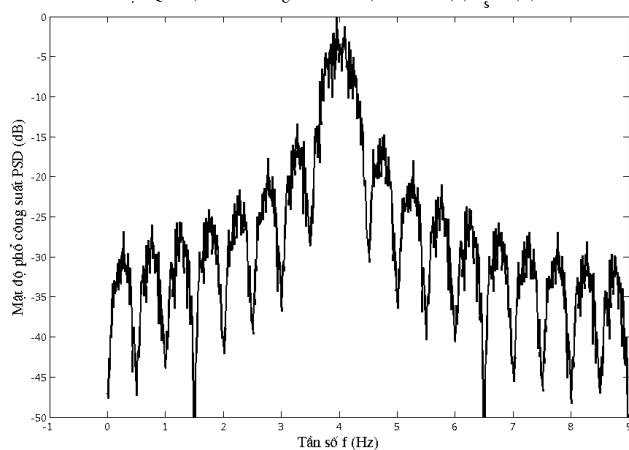


Hệ thống QPSK

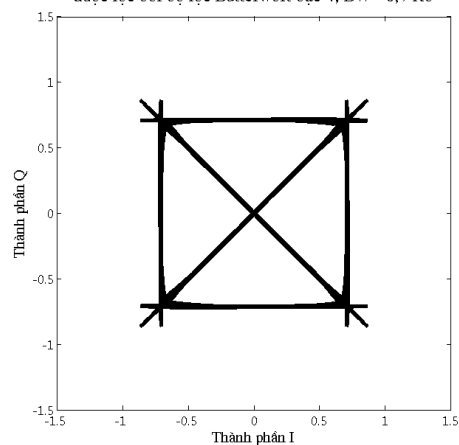
Dạng sóng tín hiệu QPSK

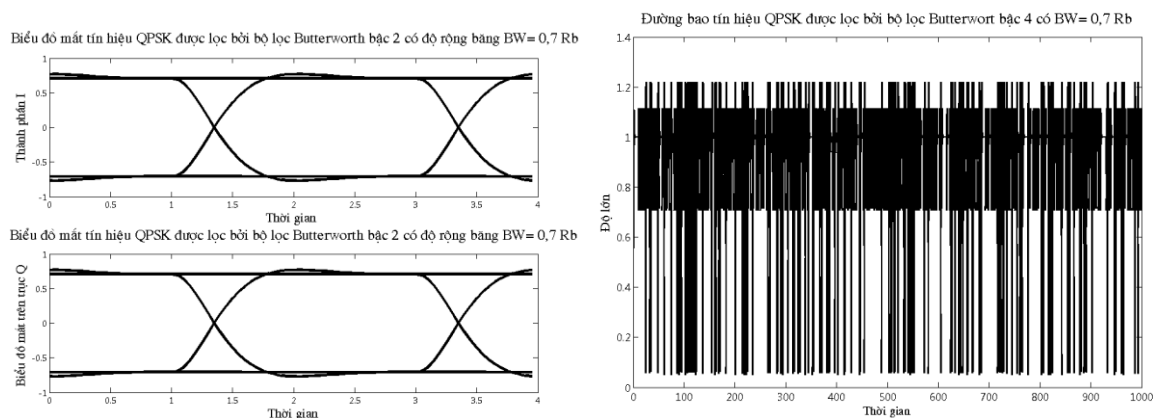


Phổ tín hiệu QPSK; Tần số trung tâm = 4Hz; $R_b = 1 \text{ bit/s}$; $R_s = 0,5$; $N_o \text{ ave} = 7$

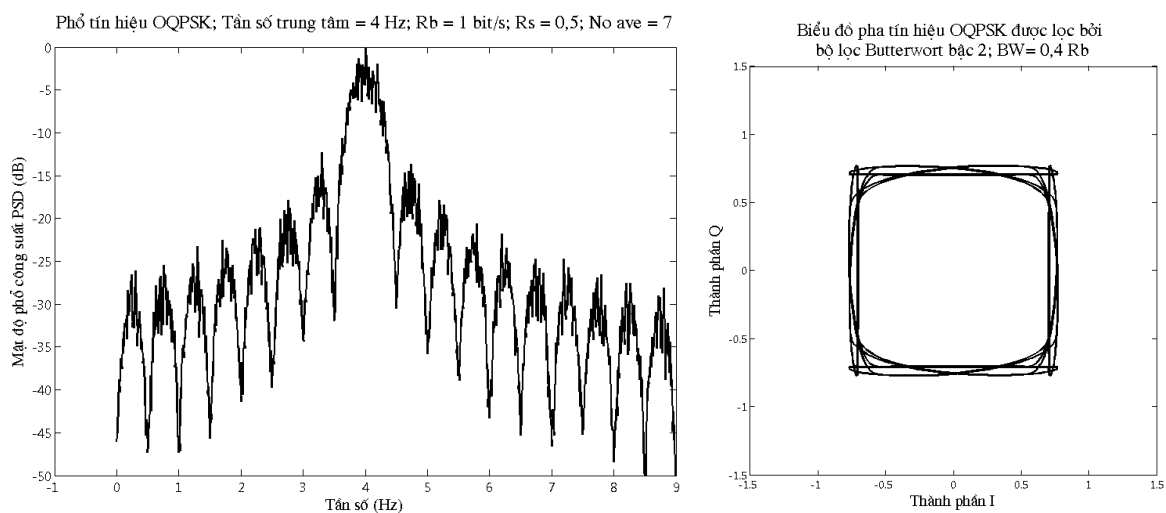
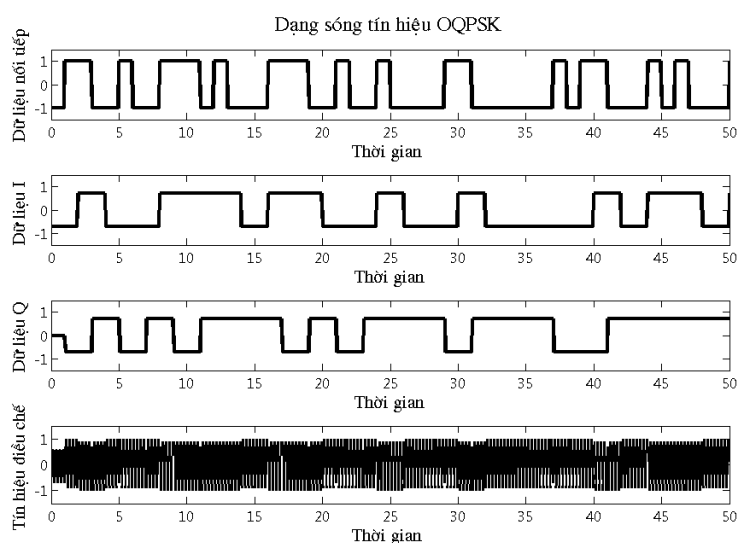


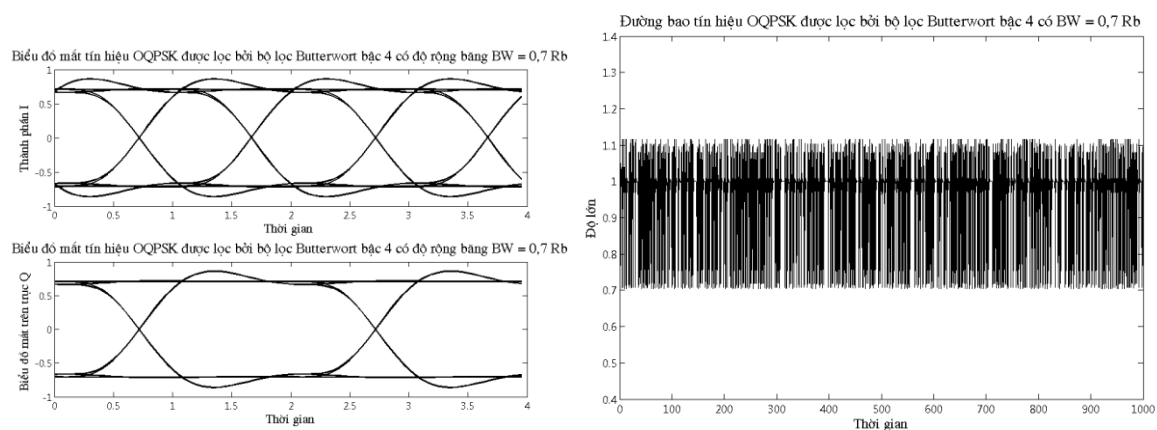
Biểu đồ pha tín hiệu QPSK được lọc bởi bộ lọc Butterworth bậc 4; $BW = 0,7 R_b$



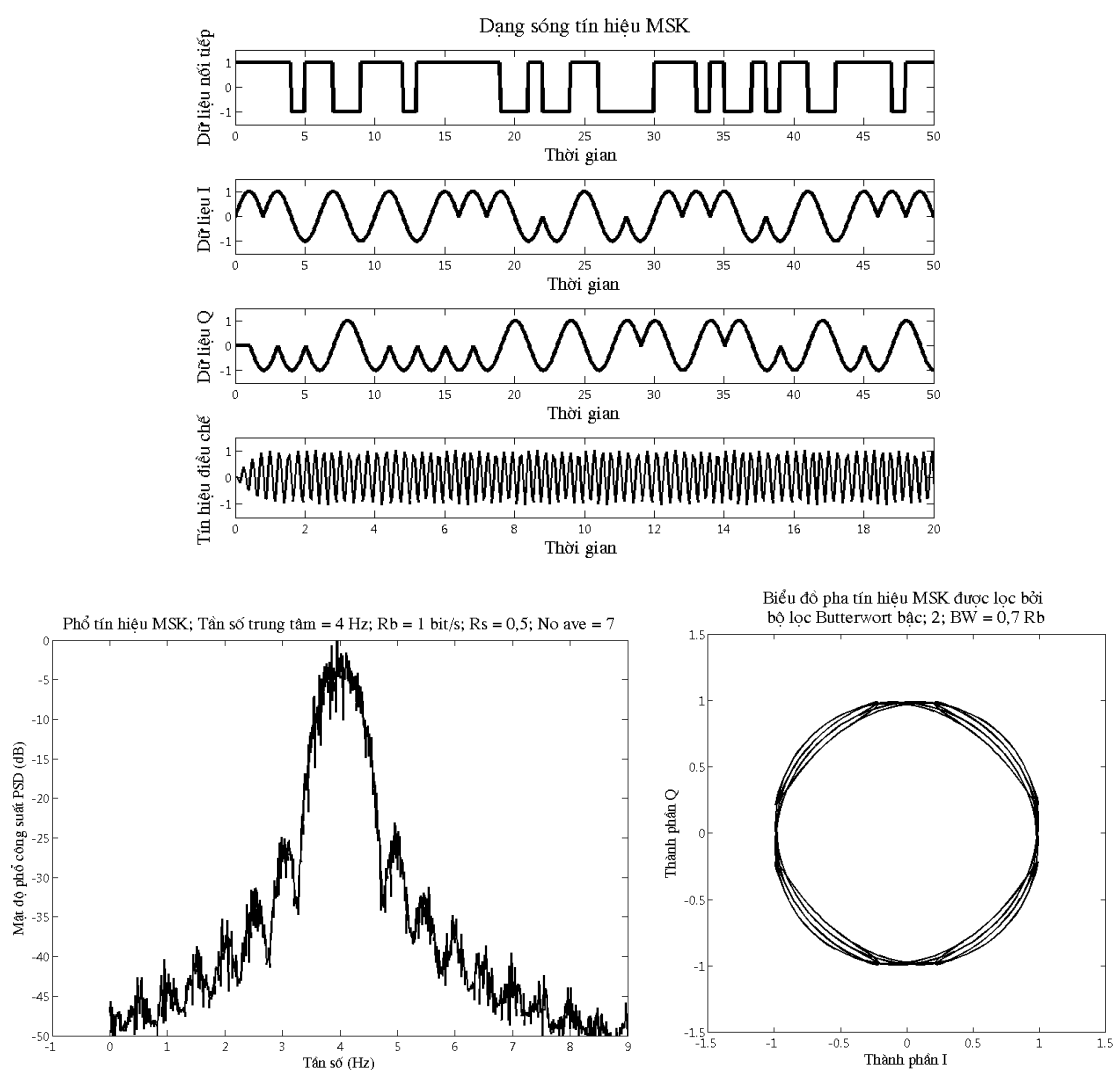


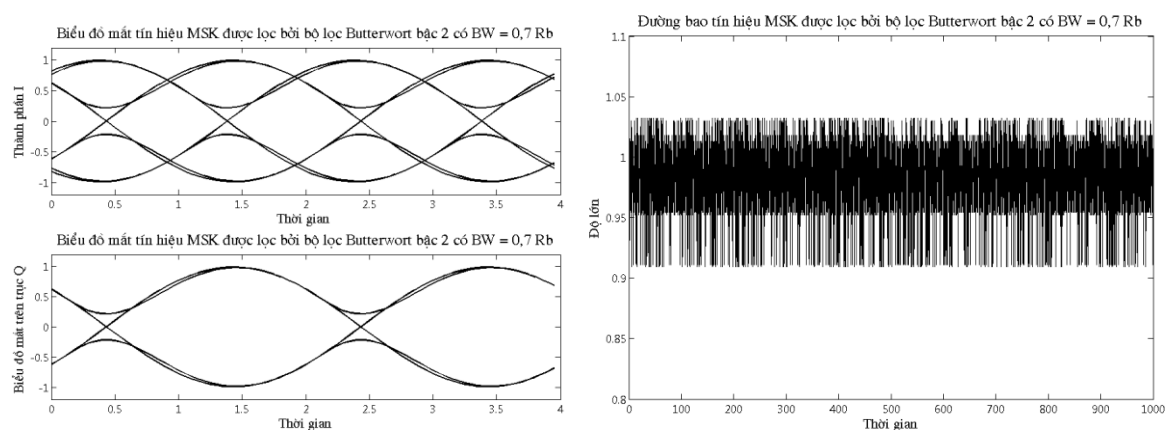
Hệ thống OQPSK





Hệ thống MSK





Hệ thống 16-QAM

