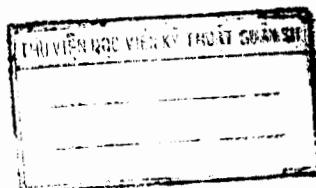


**PROAKIS J. G. và SALEHI M.**

**CÁC HỆ THỐNG THÔNG TIN HIỆN NAY  
TRÌNH BÀY THÔNG QUA SỬ DỤNG MATLAB**



**HÀ NỘI- 2003**

**NGƯỜI DỊCH:** TS. NGUYỄN QUỐC BÌNH  
KS. NGUYỄN HUY QUÂN

## MỤC LỤC

Lời người dịch	7
Lời giới thiệu	9
Lời nói đầu	11
<b>CHƯƠNG 1. CÁC TÍN HIỆU VÀ CÁC HỆ THỐNG TUYẾN TÍNH</b>	<b>16</b>
1.1 Giới thiệu chung	16
1.2 Chuỗi Fourier	16
1.2.1 Các tín hiệu tuần hoàn và các hệ thống tuyến tính không biến đổi theo thời gian	28
1.3 Các biến đổi Fourier	33
1.3.1 Định lý lấy mẫu	38
1.3.2 Phân tích trên miền tần số các hệ thống LTI	44
1.4 Công suất và năng lượng	48
1.5 Tương đương thông thấp của các tín hiệu dải thông	52
<b>CHƯƠNG 2. CÁC QUÁ TRÌNH NGẦU NHIÊN</b>	<b>63</b>
2.1 Giới thiệu chung	63
2.2 Tạo các biến ngẫu nhiên	63
2.3 Các quá trình ngẫu nhiên Gauss và Gauss-Markov	69
2.4 Phổ công suất của các quá trình ngẫu nhiên và các quá trình trắng	76
2.5 Lọc tuyến tính các quá trình ngẫu nhiên	83
2.6 Các quá trình thông dải và thông thấp	89
<b>CHƯƠNG 3. ĐIỀU CHẾ TƯƠNG TỰ</b>	<b>100</b>
3.1 Giới thiệu chung	100
3.2 Điều chế biên độ	101
3.2.1 DSB-AM	101
3.2.2 AM truyền thống	112
3.2.3 SSB-AM	119
3.3 Giải điều chế các tín hiệu điều chế biên độ	125
3.3.1 Giải điều chế DSB-AM	126
3.3.2 Giải điều chế SSB-AM	134
3.3.3 Giải điều chế AM truyền thống	139
3.4 Điều chế góc	147
<b>CHƯƠNG 4. BIẾN ĐỔI TƯƠNG TỰ – SỐ</b>	<b>164</b>
4.1 Giới thiệu chung	164
4.2 Độ đo thông tin	165
4.2.1 Mã hoá không tạp	166
4.3 Lượng tử hoá	172
4.3.1 Lượng tử hoá vô hướng	172
4.4 Điều chế mã xung (PCM)	183
4.4.1 PCM với lượng tử hoá đều	183
4.4.2 PCM với lượng tử hoá không đều	188
<b>CHƯƠNG 5. TRUYỀN DẪN TÍN HIỆU SỐ BĂNG GỐC</b>	<b>201</b>
5.1 Giới thiệu chung	201
5.2 Truyền dẫn tín hiệu nhị phân	201
5.2.1 Máy thu tối ưu đối với kênh AWGN	202
5.2.2 Mô phỏng Monte-Carlo một hệ thống truyền tín hiệu nhị phân	209
5.2.3 Các phương pháp truyền dẫn tín hiệu nhị phân khác	213
5.2.4 Biểu đồ sao của tín hiệu đối với các tín hiệu nhị phân	223
5.3 Truyền dẫn tín hiệu nhiều biên độ	225
5.3.1 Các dạng sóng với 4 mức biên độ	226
5.3.2 Máy thu tối ưu đối với kênh AWGN	228
5.3.3 Các dạng sóng tín hiệu với nhiều mức biên độ	235

5.4 Các tín hiệu nhiều chiều	239
5.4.1 Các tín hiệu trực giao nhiều chiều	239
5.4.2 Các tín hiệu trực giao đôi	249
<b>CHƯƠNG 6. TRUYỀN DẪN SỐ QUA KÊNH CÓ BĂNG THÔNG HỮU HẠN</b>	<b>261</b>
6.1 Giới thiệu chung	261
6.2 Phổ công suất của một tín hiệu số PAM	261
6.3 Đặc trưng của các kênh có độ rộng băng thông hữu hạn và méo kênh	266
6.4 Đặc trưng của xuyên nhiễu giữa các dấu	278
6.5 Thiết kế hệ thống truyền tin đối với các kênh có băng thông hữu hạn	283
6.5.1 Thiết kế tín hiệu để ISI bằng không	284
6.5.2 Thiết kế tín hiệu để có ISI kiểm soát được	289
6.5.3 Tiến mã hoá đổi với các tín hiệu đáp ứng một phần	295
6.6 Các bộ san bằng tuyến tính	299
6.6.1 Các bộ san bằng tuyến tính thích nghi	309
6.7 Các bộ san bằng phi tuyến	316
<b>CHƯƠNG 7. TRUYỀN DẪN SỐ THÔNG QUA ĐIỀU CHẾ SÓNG MANG</b>	<b>327</b>
7.1 Giới thiệu chung	327
7.2 Điều chế biên độ sóng mang	327
7.2.1 Giải điều chế các tín hiệu PAM	330
7.3 Điều chế pha sóng mang	334
7.3.1 Giải điều chế pha và tách tín hiệu	338
7.3.2 Điều chế và giải điều chế pha vi sai	345
7.4 Điều chế biên độ vuông góc	354
7.4.1 Giải điều chế và tách tín hiệu QAM	355
7.4.2 Xác suất lỗi đối với QAM trong một kênh AWGN	357
7.5 Điều chế tần số sóng mang	362
7.5.1 Khoa dịch tần số	362
7.5.2 Giải điều chế và tách các tín hiệu FSK	364
7.5.3 Xác suất lỗi đối với tách không kết hợp tín hiệu FSK	370
7.6 Đồng bộ trong các hệ thống thông tin	375
7.6.1 Đồng bộ sóng mang	376
7.6.2 Đồng bộ đồng hồ	382
<b>CHƯƠNG 8. DUNG LƯỢNG KÊNH VÀ MÃ HOÁ KÊNH</b>	<b>393</b>
8.1 Giới thiệu chung	393
8.2 Mô hình kênh và dung lượng kênh	394
8.2.1 Mô hình kênh	394
8.2.2 Dung lượng kênh	395
8.3 Mã hoá kênh	406
8.3.1 Mã khối tuyến tính	409
8.3.2 Các mã chập	423
<b>CHƯƠNG 9. CÁC HỆ THỐNG THÔNG TIN TRẢI PHỎ</b>	<b>447</b>
9.1 Giới thiệu chung	447
9.2 Các hệ thống trả phỏ chuỗi trực tiếp	448
9.2.1 Giải điều chế tín hiệu	451
9.2.2 Xác suất lỗi	453
9.2.3 Hai ứng dụng của các tín hiệu trả phỏ DS	455
9.3 Tao các chuỗi PN	462
9.4 Trả phỏ nhảy tần	468
9.4.1 Xác suất lỗi đối với các tín hiệu FH	470
9.4.2 Sử dụng phân tập tín hiệu để khắc phục nhiễu một phần băng	476
Tài liệu tham khảo	484

## LỜI NGƯỜI DỊCH

Các hệ thống thông tin liên lạc hiện đang phát triển hết sức nhanh chóng. Ngay ở Việt nam, trong hơn mươi năm qua chúng ta đã được chứng kiến một sự biến đổi ngoạn mục các hệ thống thông tin liên lạc, với việc đưa vào khai thác trên diện rộng một cơ sở hạ tầng viễn thông to lớn, phục vụ rất nhiều dịch vụ mới trên cở sở các hệ thống thông tin số. Tương ứng với điều đó, việc đào tạo một thế hệ các cán bộ kỹ thuật có trình độ cao, cập nhật với sự phát triển của các công nghệ mới đã và đang được xúc tiến một cách có hệ thống. Tuy nhiên, một trong các điểm chung tại các trường đại học đào tạo kỹ sư và cán bộ kỹ thuật chuyên ngành điện tử-viễn thông hiện nay của nước ta là sự thiếu thốn các tài liệu học tập chuyên ngành tốt cho phép người học có được cái nhìn sâu sắc vào những vấn đề kỹ thuật phức tạp của các hệ thống thông tin. Chúng tôi đã chọn dịch cuốn sách này trong số rất nhiều các tài liệu tiếng Anh về kỹ thuật thông tin khác nhằm bù đắp phần nào sự thiếu hụt nói trên.

Bạn đọc đang có trên tay bản dịch của cuốn “Các hệ thống thông tin hiện nay-Trình bày thông qua sử dụng MATLAB” (*Contemporary Communication Systems-Using MATLAB*), nhà xuất bản Brooks/Cole, Mỹ, xuất bản lần thứ nhất năm 2000. Các tác giả của cuốn sách này là các học giả lừng danh và là các giáo sư đại học đầy kinh nghiệm của Mỹ<sup>1</sup>. Cuốn sách này, theo suy nghĩ của chúng tôi - những người đang trực tiếp nghiên cứu và giảng dạy về kỹ thuật viễn thông – là một cuốn sách rất tốt về nội dung, hiện đại về phương pháp trình bày và phù hợp với nhiều

---

<sup>1</sup> Proakis J. G. là tác giả của cuốn “Thông tin số” (*Digital Communications*) nổi tiếng, đã được tái bản ở Mỹ tới lần thứ tư trong vòng 10 năm gần đây.

đối tượng, từ sinh viên đang theo học những năm cuối chuyên ngành điện tử-viễn thông đến các kỹ sư đang trực tiếp nghiên cứu và công tác trên các hệ thống viễn thông. Cuốn sách cung cấp những tóm tắt lý thuyết cơ bản nhất về các hệ thống thông tin hiện nay - cả hệ thống tương tự lẫn hệ thống số - một cách hết sức súc tích mà vẫn dễ hiểu, đồng thời cung cấp cho người học khả năng đào sâu vào các vấn đề kỹ thuật rất phức tạp vốn có vẻ như hết sức tẻ nhạt thông qua sử dụng kỹ thuật mô phỏng máy tính. Nói một cách khác, cuốn sách này lôi cuốn và gợi lên được niềm hứng khởi đối với người đọc trong quá trình học tập và tập dượt nghiên cứu.

Hiện tại, các thuật ngữ kỹ thuật trong các tài liệu ấn hành ở nước ta còn chưa thống nhất. Trong quá trình biên dịch, chúng tôi đã cố gắng tham khảo cách sử dụng các thuật ngữ kỹ thuật trong các tài liệu tiếng Việt khác và trong những trường hợp còn chưa hoàn toàn thống nhất, các thuật ngữ kỹ thuật được chuyển nghĩa sang tiếng Việt theo các tự điển phổ thông và chuyên ngành đang được lưu hành rộng rãi. Để người đọc thuận tiện trong tra cứu, trong những trường hợp này các thuật ngữ kỹ thuật được chia thành nhất bằng tiếng Anh. Một số thuật ngữ đã Việt hoá rộng rãi được dùng ở dạng phiên âm, hoặc gần như đã Việt hoá thì được để nguyên thể tiếng Anh và in nghiêng (như *symbol, analog...*).

Về phân công biên dịch, Ts. Nguyễn Quốc Bình thuộc Học viện kỹ thuật quân sự dịch các chương 1-5 và 7-9, chương 6 do Ks. Nguyễn Huy Quân thuộc Học viện Công nghệ bưu chính và viễn thông đảm nhiệm. Ts. Nguyễn Quốc Bình hiệu đính toàn bộ và chịu trách nhiệm trình bày sách. Trong quá trình biên dịch và trình bày sách, chấn chấn còn có những thiếu sót. Mọi ý kiến góp ý với những người dịch xin gửi về địa chỉ: Ts. Nguyễn Quốc Bình, Bộ môn thông tin, Học viện KTQS, 100 Đường Hoàng Quốc Việt, Hà nội.

Hà nội, tháng 10 năm 2002

Những người dịch

## LỜI GIỚI THIỆU

**“Mục đích của tính toán là để thấu hiểu vấn đề chứ không phải là các con số.”**-R. W. Hamming, *Các phương pháp tính số dành cho các kỹ sư và các nhà khoa học*, Nhà xuất bản McGraw-Hill, Inc.

Đó chính là tinh thần trong suy nghĩ của chúng tôi mà theo đó chúng tôi cho ra Bookware Companion Series™.

Các công nghệ mới nhất và các phương pháp hiện đại ngày càng được nhồi nhét nhiều hơn vào các cuả giảng vốn đã dày đặc các lý thuyết quan trọng. Hậu quả là bây giờ nhiều giảng viên đã phải đặt câu hỏi rằng “Chúng ta đơn giản đi dạy cho sinh viên về công nghệ mới nhất hay ta dạy cho họ cách suy luận?”. Chúng tôi tin tưởng rằng cả hai điều cần thiết đó không loại trừ lẫn nhau. Thật vậy, bộ sách này dựa trên niềm tin rằng các việc giải bằng máy tính và lý thuyết có thể đồng thời hỗ trợ lẫn nhau. Khi được sử dụng một cách đúng đắn, việc tính toán có thể minh họa cho lý thuyết và giúp cho sinh viên suy nghĩ, phân tích và suy luận theo các cách đầy ý nghĩa. Nó cũng giúp cho sinh viên hiểu được các mối quan hệ và các liên kết giữa thông tin mới với hiểu biết hiện có và giúp cho họ trau dồi các kỹ năng giải quyết vấn đề, khả năng trực giác cũng như tư duy phê phán. Bookware Companion Series đã được phát triển nhằm đáp ứng nhiệm vụ này.

Đặc biệt, bộ sách này được thiết kế cho những người làm công tác đào tạo muốn tích hợp các công cụ học tập dựa trên máy tính vào các cuả giảng của mình và cho các sinh viên muốn tự mình đi xa hơn nữa ngoài khuôn khổ của giáo trình. Các giảng viên sẽ tìm thấy được trong loạt sách này các phương tiện cho phép sử

dụng các công cụ phần mềm mạnh hỗ trợ các hoạt động giảng dạy của họ mà không cần phải tự mình tạo lấy các ứng dụng. Các sinh viên sẽ tìm thấy những vấn đề liên quan và có thể nhanh chóng và dễ dàng có được cách truy nhập điện tử đối với các vấn đề đó. Điều quan trọng đối với cả hai đối tượng là tiền đề xây dựng nên bộ sách này: các sinh viên học được tốt nhất khi họ được lôi cuốn vào một cách tích cực trong việc học tập của riêng mình. Bookware Companion Series sẽ thu hút họ, cho họ nếm thử các vấn đề của cuộc sống thực, bày cho họ các kỹ thuật rõ ràng để giải quyết các vấn đề thực tế và thách thức họ tự mình hiểu và áp dụng các kỹ thuật này.

Để phục vụ các nhu cầu của các bạn tốt hơn, chúng tôi luôn mong có được các cách thức cải tiến bộ sách này. Nhằm mục đích đó, các bạn hãy vui lòng liên hệ với chúng tôi trên trang Web của chúng tôi:

**<http://www.brookscole.com/engineering/ee/bookware.html>**

Các bạn có thể cho các lời khuyên về cách thức làm cho bộ sách tốt hơn nữa, chia sẻ các ý tưởng của các bạn về cách sử dụng các công nghệ trong lớp học với các đồng nghiệp của các bạn, gợi ý một vấn đề đặc biệt hoặc một thí dụ nào đó cho lần xuất bản sau hoặc cho chúng tôi biết suy nghĩ của các bạn. Chúng tôi chờ đợi nhận được phúc đáp của các bạn và cảm ơn các bạn về những trợ giúp thường xuyên.

Bill Stenquist	Người xuất bản	<i>bill.stenquist@brookscole.com</i>
Heather Woods	Phụ trách biên tập	<i>heather.woods@brookscole.com</i>
Nathan Wilbur	Phụ trách tiếp thị	<i>nathan.wilbur@brookscole.com</i>
Christina DeVeto	Trợ lý tiếp thị	<i>christina.deveto@brookscole.com</i>

## LỜI NÓI ĐẦU

Trên thị trường hiện nay rất nhiều giáo trình đề cập tới những vấn đề căn bản trong các hệ thống thông tin số và tương tự, bao gồm các thuật toán mã hoá và giải mã cũng như các kỹ thuật điều chế và giải điều chế. Vì sự cần thiết, hầu hết các giáo trình này đều chuyên chú vào các lý thuyết nhấn mạnh đến việc thiết kế và phân tích hoạt động của một số khối hình thành nên các phần tử cơ bản của hệ thống thông tin như các bộ mã hoá và giải mã, các bộ điều chế và giải điều chế. Tuy vậy, khá ít giáo trình, đặc biệt là các giáo trình được viết cho các sinh viên còn đang theo học, lại chứa đựng một loạt các ứng dụng được thiết kế nhằm thúc đẩy họ trong học tập.

### **Phạm vi của cuốn sách**

Mục tiêu của cuốn sách này là phục vụ như một sách hướng dẫn hay một phần bổ sung cho bất kỳ một giáo trình hồn hợp nào về các hệ thống thông tin. Cuốn sách này cung cấp một loạt các bài tập có thể giải được trên máy tính (đặc biệt thích hợp với các máy tính cá nhân) bằng cách sử dụng phiên bản MATLAB phổ cập cho sinh viên. Cuốn sách được chuẩn bị thoát tiền nhằm phục vụ các sinh viên đang học ở những năm trên và các sinh viên đang làm tốt nghiệp trong lĩnh vực kỹ thuật điện, kỹ thuật máy tính và khoa học máy tính. Chúng tôi xem rằng sinh viên (hay người sử dụng sách) thì đã được làm quen với các vấn đề cơ sở của MATLAB. Các chủ đề này không được đề cập đến ở đây nữa do đã có vô số sách trợ giúp và các cẩm nang về MATLAB.

Về thiết trí mà nói, việc đề cập tới một loạt các chủ đề sẽ được trình bày vắn tắt. Chúng tôi sẽ trình bày động cơ thúc đẩy vấn đề và một dẫn giải ngắn gọn về từng chủ đề một, thiết lập các ghi chú cần thiết rồi sau đó minh họa các khái niệm cơ bản

thông qua một thí dụ. Chúng tôi mong đoạn dẫn giải cơ bản này cùng với người giảng sẽ cung cấp một mức hiểu biết sâu sắc cần thiết đối với các vấn đề được đề cập tới. Thí dụ, chúng tôi đưa ra bộ thu lọc phối hợp và bộ thu tương quan và xác nhận rằng các thiết bị này dẫn đến việc giải điều chế tối ưu các tín hiệu bị nhiễu bởi tạp âm cộng trắng chuẩn (AWGN: *Additive White Gaussian Noise*), song chúng tôi sẽ không chứng minh điều xác quyết này. Sự chứng minh như thế nói chung đều đã được cung cấp trong hầu hết các giáo trình về các hệ thống truyền tin.

### Kết cấu của cuốn sách

Cuốn sách này gồm chín chương. Hai chương đầu về các tín hiệu và các hệ thống tuyến tính cũng như về các quá trình ngẫu nhiên đưa ra các kiến thức nền, nói chung cần thiết trong nghiên cứu các hệ thống truyền tin. Một chương nữa bao phủ các kỹ thuật truyền tin *analog* và sáu chương còn lại thì chuyên vào thông tin số.

### Chương 1: Các tín hiệu và các hệ thống tuyến tính

Chương này cung cấp một tổng quan về các công cụ và kỹ thuật cơ bản từ việc phân tích các hệ thống tuyến tính bao gồm cả các đặc tính miền tần số và miền thời gian. Các kỹ thuật phân tích trên miền tần số được nhấn mạnh do các kỹ thuật này thường được sử dụng nhất trong khi đề cập tới các hệ thống truyền tin.

### Chương 2: Các quá trình ngẫu nhiên

Trong chương này, chúng tôi minh họa các phương pháp tạo các biến ngẫu nhiên và các mẫu của các quá trình ngẫu nhiên. Các chủ đề bao gồm việc tạo các biến ngẫu nhiên có một hàm phân bố xác suất nhất định nào đó, việc tạo các mẫu của các quá trình ngẫu nhiên Gauss và Gauss-Markov, và đặc trưng của các quá trình ngẫu nhiên dừng trong miền thời gian cũng như trong miền tần số.

### Chương 3: Điều chế analog

Việc thực hiện các kỹ thuật điều chế và giải điều chế *analog* trong các điều kiện có và không có nhiễu cộng được đề cập trong chương này. Các hệ thống được nghiên cứu bao gồm điều chế biên độ (AM), như điều biên hai băng biên, điều chế đơn biên và điều chế biên độ truyền thống; các sơ đồ điều chế góc như

điều chế tần số (FM) và điều chế pha (PM).

#### **Chương 4: Biến đổi tương tự-số**

Trong chương này chúng tôi khảo sát một số phương pháp được sử dụng để biến đổi các tín hiệu nguồn tương tự thành các chuỗi số một cách hiệu quả. Việc biến đổi cho phép chúng ta truyền hay trữ các tín hiệu theo phương pháp số. Chúng tôi xem xét cả sơ đồ nén dữ liệu có thất thoát thông tin như điều chế mã xung (PCM) lẫn nén dữ liệu không thất thoát thông tin như mã hoá Huffman.

#### **Chương 5: Truyền dẫn số bằng gốc**

Trong chương này chúng tôi đưa ra các kỹ thuật điều chế và giải điều chế đối với truyền dẫn thông tin số thông qua một kênh AWGN. Cả kỹ thuật điều chế nhị phân lẫn nhiều mức đều được xem xét. Giải điều chế tối ưu các tín hiệu này được mô tả và chất lượng của bộ giải điều chế được đánh giá.

#### **Chương 6: Truyền dẫn tín hiệu số qua các kênh có băng tần hạn chế**

Trong chương này chúng tôi xem xét đặc tính của các kênh có băng tần hạn chế và vấn đề thiết kế các dạng sóng cho các kênh như vậy. Chúng tôi sẽ chỉ cho người sử dụng thấy rằng méo của kênh tạo nên xuyên nhiễu giữa các *symbol* (ISI: *InterSymbol Interference*) mà chúng gây nên các lỗi trong giải điều chế tín hiệu. Sau đó chúng tôi sẽ đề cập tới việc thiết kế các mạch san bằng kênh mà chúng bù khử méo kênh.

#### **Chương 7: Truyền dẫn số thông qua điều chế sóng mang**

Chúng tôi thảo luận 4 kiểu tín hiệu điều chế sóng mang thích hợp cho truyền dẫn qua các kênh có băng tần hạn chế: các tín hiệu điều chế biên độ, các tín hiệu điều chế biên độ vuông góc (QAM), điều chế pha số (PSK) và điều chế tần số số (FSK).

#### **Chương 8: Dung lượng kênh và mã hoá**

Trong chương này chúng tôi xem xét các mô hình toán học thích hợp đối với các kênh truyền tin và đưa ra một đại lượng cơ sở, gọi là dung lượng kênh, nó cho chúng ta giới hạn về lượng thông tin có thể truyền qua được trên kênh. Đặc biệt, chúng tôi xem xét hai mô hình kênh: kênh đối xứng nhị phân (BSC: *Binary*

Symmetric Channel) và kênh AWGN. Các mô hình kênh này được sử dụng trong việc đề cập tới các mã chập và mã khôi nhằm có được việc truyền tin tin cậy thông qua các kênh như thế.

### **Chương 9: Các hệ thống thông tin trai phô**

Các phân tử cơ bản của một hệ thống thông tin trai phô được đề cập tới trong chương này. Đặc biệt, các hệ thống trai phô chuỗi trực tiếp (DS: *Direct-Sequence*) và trai phô nhảy tần (FH: *Frequency-Hopped*) được xem xét liên đới với các sơ đồ điều chế pha số (PSK) và điều chế tần số số (FSK) một cách tương ứng. Việc tạo các chuỗi giả nhiễu (PN: *Pseudonoise*) để sử dụng trong các hệ thống trai phô cũng được đề cập tới.

#### **Về phần mềm**

Các *file* chương trình MATLAB cho cuốn sách này có tại Bookware Companion Resource Center trên mạng theo địa chỉ <http://www.brookscole.com/engineering/ee/bookware.html>. Các *file* này bao gồm các trình MATLAB được sử dụng trong tài liệu này. Trong hầu hết các trường hợp, chúng tôi đều đã kèm thêm vào đó những chú giải cho các *file* MATLAB để làm cho chúng dễ hiểu hơn. Tuy vậy, cũng nên lưu ý rằng trong quá trình phát triển các *file* này mục đích chủ yếu của chúng tôi là sự rõ ràng trong mã nguồn MATLAB hơn là tính hiệu quả của nó. Trong những trường hợp khi mã nguồn hiệu quả nhất lại có thể làm cho các *file* trở nên khó theo dõi thì chúng tôi đã chọn sử dụng mã nguồn kém hiệu quả hơn song lại dễ đọc hơn.

#### **Trung tâm nguồn lực Bookware Companion Series**

Cái mới đối với lần in mới này là Trung tâm nguồn lực Bookware Companion Series, một trang chủ trung tâm hỗ trợ toàn bộ bộ sách. Ở đó các bạn sẽ tìm thấy các *file* MATLAB có thể tải xuống được cho cuốn sách này. Chúng tôi dự tính giữ cho các *file* này được phổ biến, nhờ đó có được hầu hết các ưu thế của việc tán phát trên mạng. Tại Trung tâm nguồn lực, các bạn cũng sẽ tìm thấy được các nguồn lực khác như các thông tin phụ về bộ sách của chúng tôi, các kết nối tới

các trang chủ MATLAB có tác dụng trợ giúp, và các ý tưởng về công nghệ dạy học trong lớp từ các tác giả của Bookware và từ những người làm công tác đào tạo kỹ thuật khác.

Về lâu dài, chúng tôi có kế hoạch mở rộng trang chủ này thành một ngân hàng để trao đổi các ý tưởng tin cậy về giảng dạy cũng như để tường thuật trực tiếp. Bạn có một ý tưởng đối với một vấn đề hay thí dụ độc nhất nào mà bạn muốn chúng tôi xem xét đến trong lần xuất bản tới đây của cuốn sách này không? Nếu có, hãy đảo qua trang chủ của chúng tôi và bấm lên biểu tượng Mở bản thảo để tham dự vào một cuộc thảo luận trực tiếp với các tác giả, với những người như các bạn, với các sinh viên cũng như với nhà xuất bản. Trung tâm nguồn lực được bố trí tại địa chỉ <http://www.brookscole.com/engineering/ee/bookware.html>.

*John G. Proakis*

*Masoud Salehi*

## CHƯƠNG 1

# CÁC TÍN HIỆU VÀ CÁC HỆ THỐNG TUYẾN TÍNH

### 1.1 GIỚI THIỆU CHUNG

Trong chương này chúng ta sẽ xem xét những công cụ và các kỹ thuật cơ bản từ việc phân tích hệ thống tuyến tính được sử dụng trong phân tích các hệ thống truyền tin. Các hệ thống tuyến tính và các đặc tính của chúng trong các miền thời gian và tần số, cùng với lý thuyết xác suất và việc phân tích các tín hiệu ngẫu nhiên là hai chủ đề căn bản nhất thiết phải hiểu rõ trước khi nghiên cứu các hệ thống truyền tin. Hầu hết các kênh truyền tin và nhiều khối con của các máy phát và máy thu có thể mô hình hoá tốt được như là các hệ thống tuyến tính không biến đổi theo thời gian (LTI: *Linear Time-Invariant*) và do vậy các công cụ và kỹ thuật nổi tiếng xuất phát từ việc phân tích các hệ thống tuyến tính có thể áp dụng được trong việc phân tích chúng. Chúng tôi nhấn mạnh các công cụ phân tích thời gian-tần số do đây là các kỹ thuật được sử dụng thường xuyên nhất. Chúng ta khởi đầu với các chuỗi và các biến đổi Fourier; sau đó chúng ta sẽ xét tất cả các khái niệm về năng lượng và công suất, định lý lấy mẫu, việc biểu diễn thông thấp các tín hiệu dải thông.

### 1.2 CHUỖI FOURIER

Quan hệ vào-ra của một hệ thống tuyến tính không biến đổi theo thời gian được cho bởi tích chập được xác định theo

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} h(\tau) x(t - \tau) d\tau \quad (1.2.1)$$

trong đó  $h(t)$  ký hiệu phản ứng xung của hệ thống,  $x(t)$  là tín hiệu lối vào còn  $y(t)$  là tín hiệu lối ra. Nếu tín hiệu lối vào  $x(t)$  là một hàm mũ phức được cho bởi

$$x(t) = Ae^{j2\pi f_0 t} \quad (1.2.2)$$

thì tín hiệu lối ra được cho theo

$$y(t) = \int_{-\infty}^{\infty} Ae^{j2\pi f_0(t-\tau)} h(\tau) d\tau = A \left[ \int_{-\infty}^{\infty} h(\tau) e^{-j2\pi f_0 \tau} d\tau \right] e^{j2\pi f_0 t} \quad (1.2.3)$$

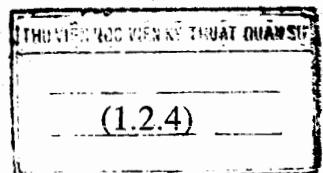
Nói một cách khác, lối ra là một tín hiệu mũ phức có cùng tần số với tín hiệu lối vào. Biên độ (phức) của tín hiệu lối ra, tuy vậy, là biên độ (phức) của tín hiệu lối vào nhân với

$$\int_{-\infty}^{\infty} h(\tau) e^{-j2\pi f_0 \tau} d\tau$$

Hãy chú ý rằng đại lượng trên đây là một hàm của phản ứng xung  $h(t)$  của hệ thống tuyến tính LTI và tần số  $f_0$  của tín hiệu lối vào. Do vậy, việc tính toán phản ứng của các hệ thống LTI với các tín hiệu mũ phức lối vào là đặc biệt dễ dàng. Hệ quả là trong phân tích các hệ thống tuyến tính LTI, một cách hoàn toàn tự nhiên người ta tìm cách khai triển các tín hiệu như một tổng các tín hiệu mũ phức. Chuỗi Fourier và các biến đổi Fourier là các kỹ thuật khai triển các tín hiệu thành các tín hiệu mũ phức.

Một chuỗi Fourier là một khai triển trực giao các tín hiệu tuần hoàn có chu kỳ  $T_0$  khi tập tín hiệu  $\{e^{j2\pi n f_0 t/T_0}\}_{n=-\infty}^{\infty}$  được sử dụng làm cơ sở của khai triển. Với cơ sở này, bất kỳ tín hiệu tuần hoàn<sup>1</sup>  $x(t)$  có chu kỳ  $T_0$  nào cũng đều có thể biểu diễn được theo

$$x(t) = \sum_{n=-\infty}^{\infty} x_n e^{j2\pi n t/T_0}$$



(1.2.4)

trong đó các hệ số  $x_n$  được gọi là các hệ số chuỗi Fourier của tín hiệu  $x(t)$  và được cho bởi

- 1) Một điều kiện đủ để tồn tại chuỗi Fourier là  $x(t)$  thoả mãn điều kiện Dirichlet.

$$x_n = \frac{1}{T_0} \int_{\alpha}^{\alpha+T_0} x(t) e^{-j2\pi nt/T_0} dt \quad (1.2.5)$$

Ở đây  $\alpha$  là một hằng số bất kỳ được chọn sao cho việc tính tích phân trở nên đơn giản. Tần số  $f_0=1/T_0$  được gọi là tần số cơ bản của tín hiệu tuần hoàn còn tần số  $f_n=n/T_0$  được gọi là hài thứ  $n$ . Trong hầu hết các trường hợp hoặc  $\alpha=0$  hoặc  $\alpha=-T_0/2$  là cách chọn thích hợp.

Dạng này của chuỗi Fourier được xem là chuỗi Fourier dạng mū và có thể áp dụng cho cả các tín hiệu  $x(t)$  thực lẫn phức chừng nào chúng còn là các tín hiệu tuần hoàn. Trong trường hợp tổng quát, các hệ số của chuỗi Fourier  $\{x_n\}$  là các số phức ngay cả khi  $x(t)$  là một tín hiệu có giá trị thực.

Khi  $x(t)$  là một tín hiệu có giá trị thực thì chúng ta có

$$x_{-n} = \frac{1}{T_0} \int_{\alpha}^{\alpha+T_0} x(t) e^{j2\pi nt/T_0} dt = \frac{1}{T_0} \left[ \int_{\alpha}^{\alpha+T_0} x(t) e^{-j2\pi nt/T_0} dt \right]^* = x_n^* \quad (1.2.6)$$

Từ đây, hiển nhiên chúng ta thấy rằng

$$\begin{cases} |x_n| = |x_{-n}| \\ \angle x_n = -\angle x_{-n} \end{cases} \quad (1.2.7)$$

Như vậy, các hệ số chuỗi Fourier của một tín hiệu thực thì có tính đối xứng Hermit; nghĩa là phần thực của chúng thì chẵn còn phần ảo của chúng thì lẻ (hoặc tương đương là biên độ của chúng chẵn còn pha của chúng lẻ).

Dạng khác của chuỗi Fourier, được biết với tên gọi chuỗi Fourier lượng giác, có thể áp dụng được chỉ đối với các tín hiệu thực tuần hoàn và nhận được bằng cách xác định

$$x_n = \frac{a_n - jb_n}{2} \quad (1.2.8)$$

$$x_{-n} = \frac{a_n + jb_n}{2} \quad (1.2.9)$$

mà theo công thức Euler

$$e^{-j2\pi nt/T_0} = \cos(2\pi t \frac{n}{T_0}) - j \sin(2\pi t \frac{n}{T_0}) \quad (1.2.10)$$

ta có

$$\begin{aligned} a_n &= \frac{2}{T_0} \int_{\alpha}^{\alpha+T_0} x(t) \cos(2\pi t \frac{n}{T_0}) dt \\ b_n &= \frac{2}{T_0} \int_{\alpha}^{\alpha+T_0} x(t) \sin(2\pi t \frac{n}{T_0}) dt \end{aligned} \quad (1.2.11)$$

và do đó

$$x(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(2\pi t \frac{n}{T_0}) + b_n \sin(2\pi t \frac{n}{T_0}) \quad (1.2.12)$$

Để ý rằng đối với  $n=0$ , chúng ta luôn có  $b_0=0$  nên  $a_0=2x_0$ .

Bằng cách đặt

$$\begin{cases} c_n = \sqrt{a_n^2 + b_n^2} \\ \theta_n = -\arctan \frac{b_n}{a_n} \end{cases} \quad (1.2.13)$$

và sử dụng quan hệ

$$a \cos \phi + b \sin \phi = \sqrt{a^2 + b^2} \cos(\phi - \arctan \frac{b}{a}) \quad (1.2.14)$$

chúng ta có thể viết phương trình (1.2.12) ở dạng

$$x(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} c_n \cos(2\pi t \frac{n}{T_0} + \theta_n) \quad (1.2.15)$$

mà đây là dạng thứ ba của khai triển chuỗi Fourier đối với các tín hiệu thực và tuần hoàn. Tổng quát, các hệ số chuỗi Fourier  $\{x_n\}$  đối với các tín hiệu thực liên hệ với  $a_n$ ,  $b_n$ ,  $c_n$  và  $\theta_n$  thông qua

$$\begin{cases} a_n = 2 \operatorname{Re}[x_n] \\ b_n = -2 \operatorname{Im}[x_n] \\ c_n = |x_n| \\ \theta_n = \angle x_n \end{cases} \quad (1.2.16)$$

Các đồ thị của  $|x_n|$  và  $\angle x_n$  theo  $n$  hay  $nf_0$  được gọi là phổ rời rạc của  $x(t)$ . Đồ thị của  $|x_n|$  thường được gọi là phổ biên độ còn của  $\angle x_n$  thì được gọi là phổ pha.

Nếu  $x(t)$  là một tín hiệu thực và chẵn, nghĩa là nếu  $x(-t)=x(t)$ , thì bằng cách lấy  $\alpha=-T_0/2$ , chúng ta có

$$b_n = \frac{2}{T_0} \int_{-T_0/2}^{T_0/2} x(t) \sin\left(2\pi t \frac{n}{T_0}\right) dt \quad (1.2.17)$$

bằng không vì hàm dưới dấu tích phân là một hàm lẻ theo  $t$ . Như thế, đối với một tín hiệu  $x(t)$  thực và chẵn, mọi hệ số  $x_n$  đều là số thực. Trong trường hợp này, chuỗi Fourier lượng giác bao gồm tất cả các hàm cosine. Tương tự, nếu  $x(t)$  thực và lẻ, nghĩa là nếu  $x(-t)=-x(t)$ , thì

$$a_n = \frac{2}{T_0} \int_{-T_0/2}^{T_0/2} x(t) \cos\left(2\pi t \frac{n}{T_0}\right) dt \quad (1.2.18)$$

cũng bằng không và mọi hệ số  $x_n$  đều là số ảo. Trong trường hợp này, chuỗi Fourier lượng giác chứa toàn các hàm sine.

### BÀI TẬP MINH HOA

**Bài tập minh họa 1.1 [chuỗi Fourier của một dãy tín hiệu hình chữ nhật]** Cho tín hiệu tuần hoàn  $x(t)$  có chu kỳ  $T_0$  được xác định theo

$$x(t) = A\Pi\left(\frac{t}{2t_0}\right) = \begin{cases} A, & |t| < t_0 \\ \frac{A}{2}, & t = \pm t_0 \\ 0, & |t| > t_0 \end{cases} \quad (1.2.19)$$

đối với  $|t| \leq T_0/2$ , trong đó  $t_0 < T_0/2$ . Tín hiệu chữ nhật  $\Pi(t)$  thường được xác định bởi

$$\Pi(t) = \begin{cases} 1, & |t| < 1/2 \\ 1/2, & t = \pm 1/2 \\ 0, & |t| > 1/2 \end{cases} \quad (1.2.20)$$

Đồ thị của  $x(t)$  được vẽ trên hình 1.1. Giả sử rằng  $A=1$ ,  $T_0=4$  còn  $t_0=1$ :

1. Hãy xác định các hệ số chuỗi Fourier của  $x(t)$  ở dạng mũ và lượng giác.
2. Hãy vẽ phổ rời rạc của  $x(t)$ .

## Lời giải

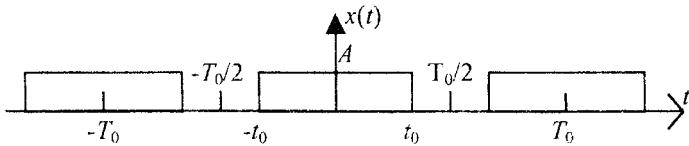
1. Để tính ra các hệ số chuỗi Fourier trong khai triển của  $x(t)$ , chúng ta có

$$x_n = \frac{1}{4} \int_{-1}^1 e^{-j2\pi t/4} dt = \frac{1}{-2j\pi n} [e^{-j2\pi n/4} - e^{j2\pi n/4}] \quad (1.2.21)$$

$$= \frac{1}{2} \sin c(n/2) \quad (1.2.22)$$

trong đó  $\sin c(x)$  được định nghĩa là

$$\sin c(x) = \frac{\sin(\pi x)}{\pi x} \quad (1.2.23)$$



Hình 1.1 Tín hiệu  $x(t)$  của bài tập minh họa 1.1

Đồ thị hàm sinc được cho trên hình 1.2. Hiển nhiên là mọi hệ số  $x_n$  đều thực (do  $x(t)$  thực và chẵn)

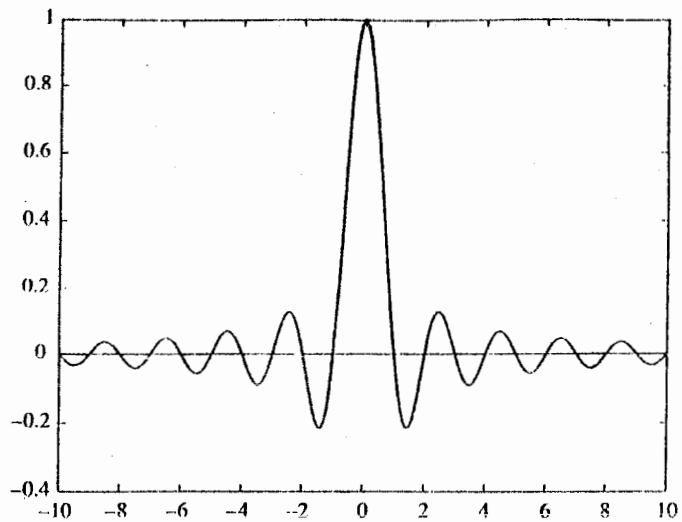
$$\begin{cases} a_n = \sin c\left(\frac{n}{2}\right) \\ b_n = 0 \\ c_n = \left|\sin c\left(\frac{n}{2}\right)\right| \\ \theta_n = 0, \pi \end{cases} \quad (1.2.24)$$

Chú ý rằng với các  $n$  chẵn,  $x_n=0$  (ngoại trừ  $n=0$ , khi đó  $a_0=c_0=1$  và  $x_0=1/2$ ).

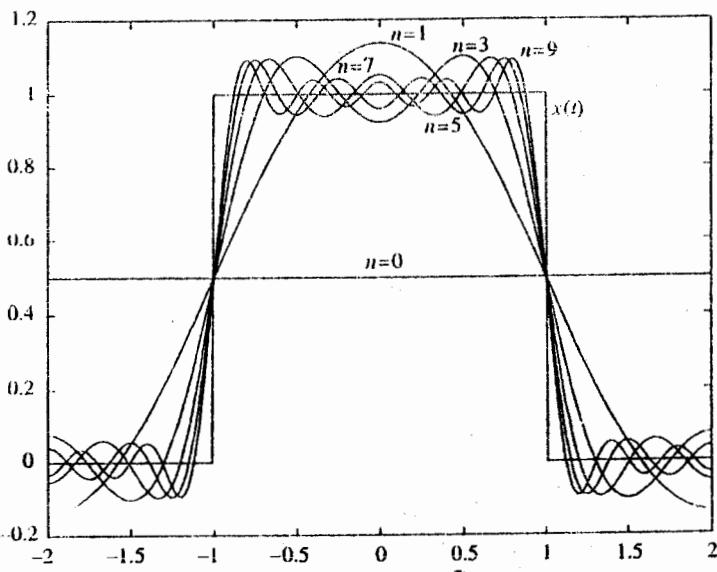
Bằng cách sử dụng các hệ số này, chúng ta có

$$x(t) = \sum_{n=-\infty}^{\infty} \frac{1}{2} \sin c\left(\frac{n}{2}\right) e^{j2\pi nt/4} = \frac{1}{2} + \sum_{n=1}^{\infty} \sin c\left(\frac{n}{2}\right) \cos\left(2\pi t \frac{n}{4}\right) \quad (1.2.25)$$

Một đồ thị của các xấp xỉ chuỗi Fourier đối với tín hiệu này vẽ trên một chu kỳ với  $n=0, 1, 3, 5, 7, 9$  được trình bày trên hình 1.3. Hãy chú ý rằng khi  $n$  tăng thì xấp xỉ trở nên gần hơn với tín hiệu gốc  $x(t)$ .



Hình 1.2 Tín hiệu sinc.



Hình 1.3 Một số xấp xỉ chuỗi Fourier đối với tín hiệu chữ nhật của bài tập minh họa 1.1.

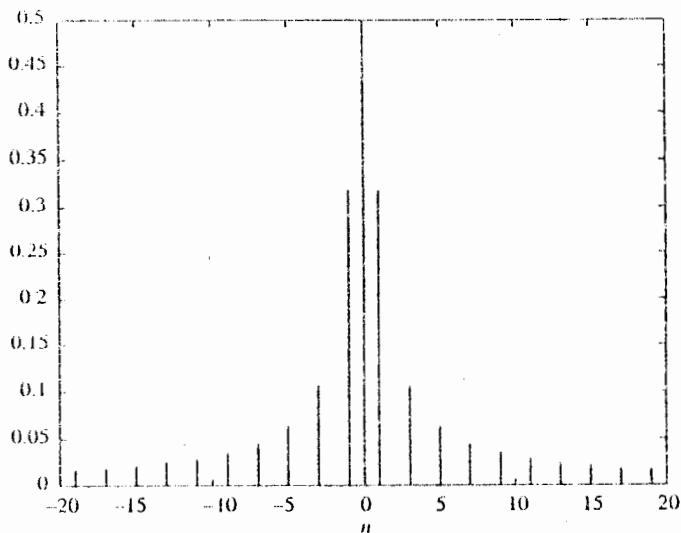
2. Hãy chú ý rằng  $x_n$  luôn luôn thực. Do vậy, tuỳ theo dấu của nó pha sẽ hoặc là không hoặc  $\pi$ . Độ lớn của các  $x_n$  là  $\frac{1}{2} \left| \sin c\left(\frac{n}{2}\right) \right|$ . Phổ rời rạc được trình bày trên hình 1.4.

MATLAB script để vẽ phổ rời rạc của tín hiệu này được cho dưới đây.

### m-file

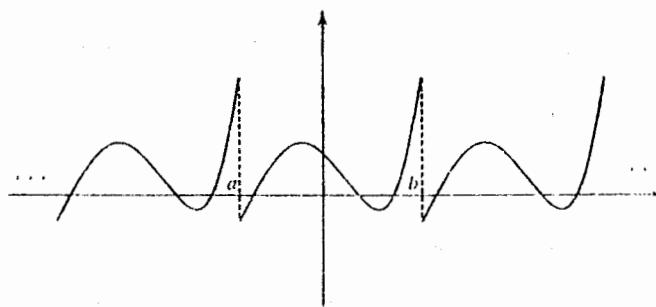
```
% MATLAB script for Illustrative Problem 1, Chapter 1.
```

```
n=[-20:1:20];
x=abs(sinc(n/2));
stem(n,x);
```



Hình 1.4 Phổ rời rạc của tín hiệu trong bài tập minh họa 1.1.

Khi tín hiệu  $x(t)$  được mô tả trên một chu kỳ giữa  $a$  và  $b$ , như được trình bày trên hình 1.5, và tín hiệu trong khoảng  $[a, b]$  được cho trong một *m-file*, các hệ số chuỗi Fourier có thể nhận được bằng cách sử dụng *m-file fseries.m* được cho dưới đây.



Hình 1.5 Một tín hiệu tuần hoàn.

**m-file**

```

function xx=fseries(funfcn,a,b,n,tol,p1,p2,p3)
%FSERIES Returns the Fourier series coefficients.
%
%          XX=FSERIES(FUNFCN,A,B,N,TOL,P1,P2,P3)
%
%          funfcn=The given function, in an m-file.
%
%          It can depend on up to three parameters
%
%          p1,p2, and p3. The function is given
%
%          over one period extending from 'a' to 'b'
%
%          xx=vector of length n+1 of Fourier Series
%
%          Coefficients, xx0,xx1,...,xxn.
%
%          p1,p2,p3=parameters of funfcn.
%
%          tol=the error level.

j=sqrt(-1);
args0=[];
for nn=1:nargin-5
    args0=[args0,',p',int2str(nn)];
end
args=[args0,')'];
t=b-a;
xx(1)=eval(['1/(',num2str(t),').*quad(funfcn,a,b,tol,[],args)]);

for i=1:n
    new_fun = 'exp_fnct';
    args=['', num2str(i), ',', num2str(t), args0, ')'];
    xx(i+1)=eval(['1/(',num2str(t),').*quad(new_fun,a,b,tol,[],funfcn,args)]);
end

```

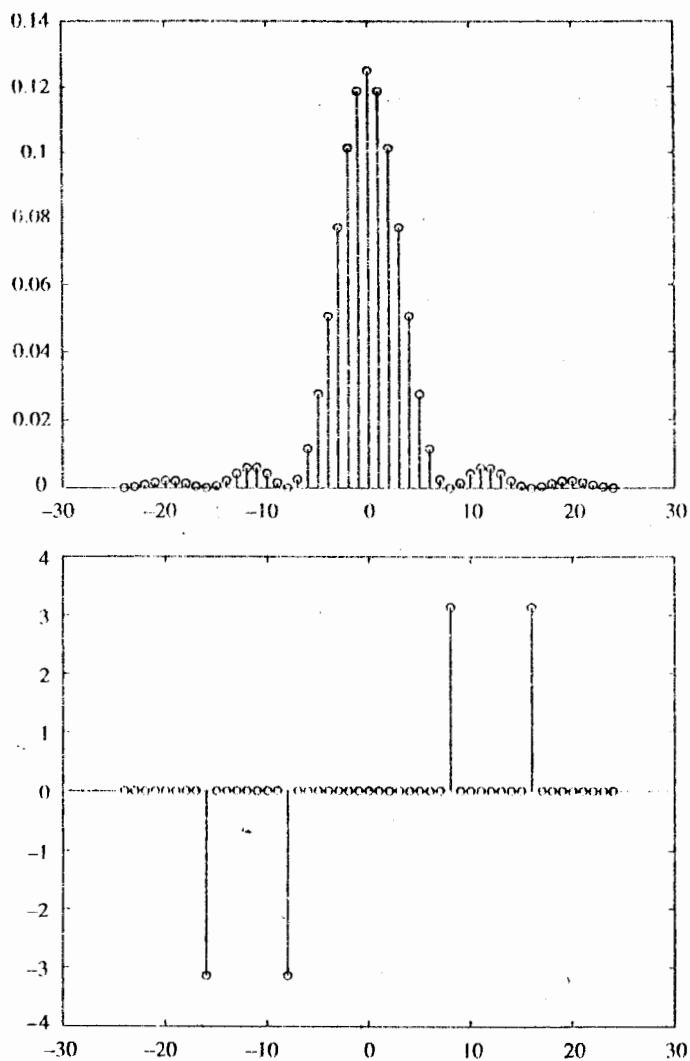
**BÀI TẬP MINH HOẠ**

**Bài tập minh họa 1.2 [Phổ biên độ và phổ pha]** Hãy xác định và vẽ các phổ biên độ và phổ pha của tín hiệu tuần hoàn  $x(t)$  với chu kỳ bằng 8 và được xác định theo  $x(t)=\Lambda(t)$  đối với  $|t| \leq 4$ .

**Lời giải**

Do tín hiệu được cho bằng một *m-file* **lambda.m**, chúng ta có thể chọn khoảng  $[a, b]=[-4, 4]$  và xác định được các hệ số. Hãy chú ý rằng *m-file* **fseries.m** xác định các hệ số của chuỗi Fourier đối với các giá trị không âm của  $n$ , song do ở đây  $x(t)$  là thực nên

chúng ta có  $x_n = x_{-n}$ . Trên hình 1.6 các phổ biên độ và phổ pha của tín hiệu này được vẽ đối với  $n$  được chọn bằng 24.



Hình 1.6 Phổ biên độ và phổ pha trong bài tập minh họa 1.2.

MATLAB script để xác định và vẽ phổ biên độ và phổ pha được cho dưới đây.

### m-file

```
% MATLAB script for Illustrative Problem 2, Chapter 1.
```

```
echo on
```

```
fnct='lambda';
```

```
a=-4;
```

```
b=4;
```

```

n=24;
tol=0.1;
xx=fseries(fnct,a,b,n,tol);
xx1=xx(n+1:-1:2);
xx1=[conj(xx1),xx];
absxx1=abs(xx1);
pause % Press any key to see a plot of the magnitude spectrum
n1=[-n:n];
stem(n1,absxx1)
title('The Discrete Magnitude Spectrum')
phasexx1=angle(xx1);
pause % Press any key to see a plot of the phase
stem(n1,phasexx1)
title('The Discrete Phase Spectrum')

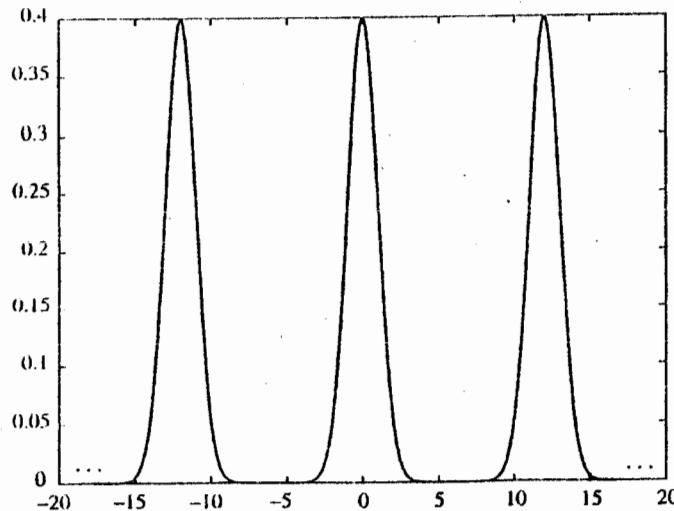
```

### BÀI TẬP MINH HOA

**Bài tập minh họa 1.3 [Phổ biên độ và phổ pha]** Hãy xác định và vẽ phổ biên độ và phổ pha của một tín hiệu tuần hoàn với một chu kỳ bằng 12 mà nó được cho theo

$$x(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2}$$

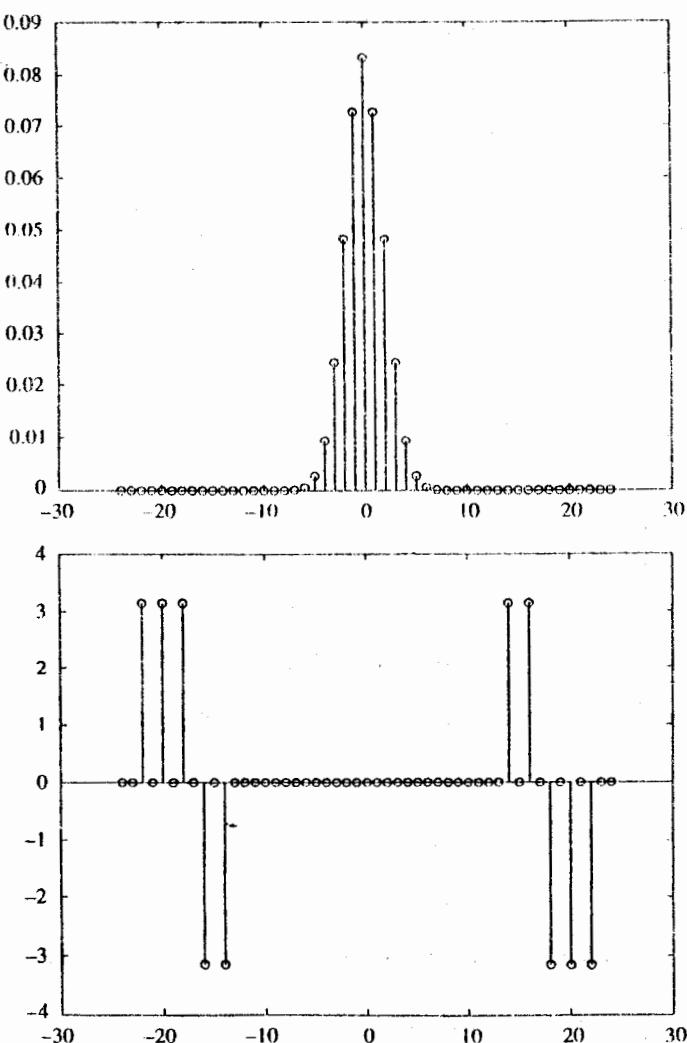
trong khoảng  $[-6, 6]$ . Đồ thị tín hiệu này được cho trên hình 1.7.



Hình 1.7 Tín hiệu tuần hoàn trong bài tập minh họa 1.3.

### Lời giải

Tín hiệu này chính trùng với hàm mật độ xác suất của một biến ngẫu nhiên Gao-xơ (*normal-chuẩn*) có kỳ vọng không và phương sai bằng 1 được cho trong *m-file normal.m*. File này cần hai tham số là  $m$  và  $s$  là kỳ vọng (trị trung bình) và độ lệch chuẩn phương của biến ngẫu nhiên đó mà trong bài tập này chúng lần lượt là 0 và 1. Do vậy, chúng ta có thể sử dụng MATLAB *script* sau đây để có được các đồ thị phổ biên độ và phổ pha như được thể hiện trên hình 1.8.



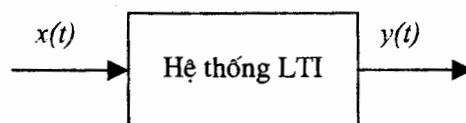
Hình 1.8 Phổ biên độ và phổ pha trong bài tập minh họa 1.3.

**m-file**

```
% MATLAB script for Illustrative Problem 3, Chapter 1.
echo on
fnct='normal';
a=-6;
b=6;
n=24;
tol=0.1;
xx=fseries(fnct,a,b,n,tol,0,1);
xx1=xx(n+1:-1:2);
xx1=[conj(xx1),xx];
absxx1=abs(xx1);
pause % Press any key to see a plot of the magnitude
n1=[-n:n];
stem(n1,absxx1)
title('The Discrete Magnitude Spectrum')
phasexx1=angle(xx1);
pause % Press any key to see a plot of the phase
stem(n1,phasexx1)
title('The Discrete Phase Spectrum')
```

### 1.2.1. CÁC TÍN HIỆU TUẦN HOÀN VÀ CÁC HỆ THỐNG TUYẾN TÍNH KHÔNG BIẾN ĐỔI THEO THỜI GIAN (LTI)

Khi một tín hiệu tuần hoàn  $x(t)$  được cho qua một hệ thống tuyến tính không biến đổi theo thời gian (LTI) như được thể hiện trên hình 1.9 thì tín hiệu lối ra  $y(t)$  cũng là một tín hiệu tuần hoàn, thường thường có cùng một chu kỳ như của tín hiệu lối vào<sup>2</sup> (tại sao?), và do đó nó có một khai triển chuỗi Fourier.



Hình 1.9 Các tín hiệu tuần hoàn qua các hệ thống LTI.

Nếu  $x(t)$  và  $y(t)$  được khai triển theo

2) Chúng ta nói *thường thường* có cùng chu kỳ với tín hiệu lối vào. Bạn có thể đưa ra một thí dụ nào mà chu kỳ tín hiệu lối ra khác với chu kỳ tín hiệu lối vào không?

$$x(t) = \sum_{n=-\infty}^{\infty} x_n e^{j2\pi n t / T_0} \quad (1.2.26)$$

$$y(t) = \sum_{n=-\infty}^{\infty} y_n e^{j2\pi n t / T_0} \quad (1.2.27)$$

thì quan hệ giữa các hệ số chuỗi Fourier của  $x(t)$  và  $y(t)$  có thể nhận được bằng cách sử dụng tích chập

$$\begin{aligned} y(t) &= \int_{-\infty}^{\infty} x(t-\tau) h(\tau) d\tau = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x_n e^{j2\pi n(t-\tau)/T_0} h(\tau) d\tau \\ &\sum_{n=-\infty}^{\infty} x_n \left( \int_{-\infty}^{\infty} h(\tau) e^{-j2\pi n\tau/T_0} d\tau \right) e^{j2\pi n t / T_0} = \sum_{n=-\infty}^{\infty} y_n e^{j2\pi n t / T_0} \end{aligned} \quad (1.2.28)$$

Từ quan hệ trên đây, ta có

$$y_n = x_n H\left(\frac{n}{T_0}\right) \quad (1.2.29)$$

trong đó  $H(f)$  là ký hiệu của hàm truyền<sup>3</sup> của hệ thống LTI, là biến đổi Fourier của hàm phản ứng (hay đáp ứng) xung  $h(t)$  của nó:

$$H(f) = \int_{-\infty}^{\infty} h(t) e^{-j2\pi f t} dt \quad (1.2.30)$$

## BÀI TẬP MINH HOA

**Bài tập minh họa 1.4 [Việc lọc các tín hiệu tuần hoàn]** Một chuỗi xung tam giác  $x(t)$  có chu kỳ  $T_0=2$  được xác định trên một chu kỳ theo

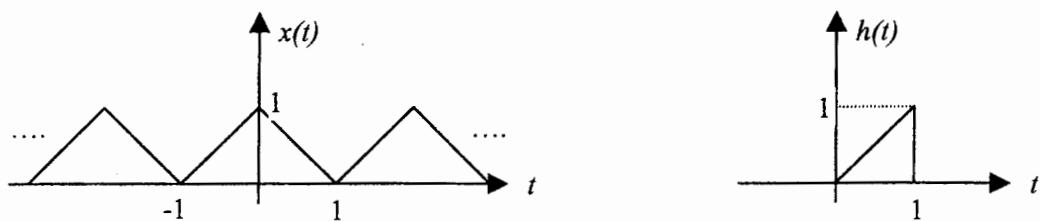
$$\Lambda(t) = \begin{cases} t+1, & -1 \leq t \leq 0 \\ -t+1, & 0 \leq t \leq 1 \\ 0, & \text{Với các giá trị } t \text{ khác} \end{cases} \quad (1.2.31)$$

1. Hãy xác định các hệ số chuỗi Fourier của  $x(t)$ .
2. Hãy vẽ đồ thị phổ rời rạc của  $x(t)$ .
3. Giả sử tín hiệu này đi qua một hệ thống LTI có đáp ứng xung được cho theo

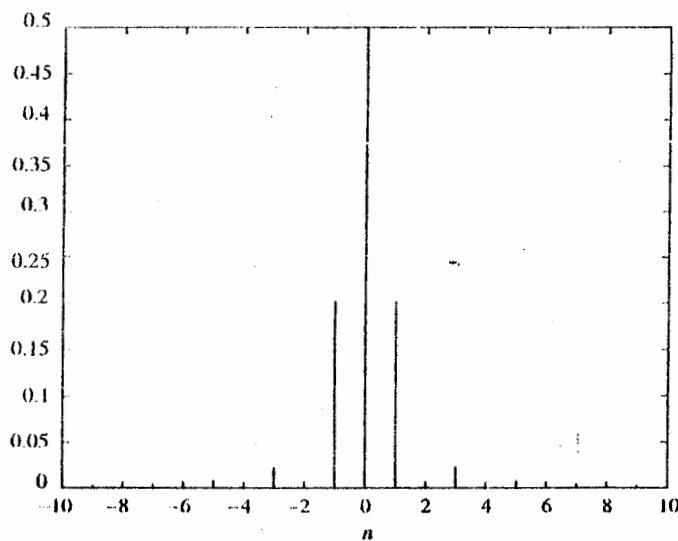
$$h(t) = \begin{cases} t, & 0 \leq t < 1 \\ 0, & \text{Với các giá trị } t \text{ khác} \end{cases} \quad (1.2.32)$$

3) Cũng còn được gọi là đáp ứng tần số của hệ thống.

hãy vẽ đồ thị phổ rời rạc và tín hiệu lối ra  $y(t)$ . Các đồ thị của  $x(t)$  và  $h(t)$  được cho trên hình 1.10.



Hình 1.10 Tín hiệu lối vào và phản ứng xung của hệ thống.



Hình 1.11 Phổ rời rạc của tín hiệu.

### Lời giải

- Chúng ta có 
$$x_n = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} x(t) e^{-j2\pi nt/T_0} dt \quad (1.2.33)$$

$$= \frac{1}{2} \int_{-1}^1 \Lambda(t) e^{-j\pi nt} dt \quad (1.2.34)$$

$$= \frac{1}{2} \int_{-\infty}^{\infty} \Lambda(t) e^{-j\pi nt} dt \quad (1.2.35)$$

$$= \frac{1}{2} F[\Lambda(t)]_{f=n/2} \quad (1.2.36)$$

$$= \frac{1}{2} \sin c^2\left(\frac{n}{2}\right) \quad (1.2.37)$$

ở đây chúng ta đã sử dụng các sự kiện là  $\Lambda(t)$  đồng nhất không ngoài khoảng  $[-1, 1]$  và biến đổi Fourier của  $\Lambda(t)$  là  $\text{sinc}^2(f)$ . Kết quả này cũng có thể nhận được bằng cách sử dụng biểu thức đối với  $\Lambda(t)$  và lấy tích phân từng phần (phân đoạn). Hiển nhiên, chúng ta có  $x_n=0$  với mọi  $n$  chẵn ngoại trừ  $n=0$ .

2. Một đồ thị phổ rời rạc của  $x(t)$  được thể hiện trên hình 1.11.

3. Trước hết chúng ta phải tìm  $H(f)$ , hàm truyền của hệ thống. Mặc dù điều này có thể làm được một cách giải tích, chúng ta vẫn cứ chấp nhận giải pháp tính số. Biên độ nhận được của hàm truyền và cả độ lớn của  $H(n/T_0)=H(n/2)$  được thể hiện trên hình 1.12. Để xác định phổ rời rạc của tín hiệu lõi ra, chúng ta hãy áp dụng quan hệ

$$y_n = x_n H\left(\frac{n}{T_0}\right) \quad (1.2.38)$$

$$= \frac{1}{2} \sin c^2\left(\frac{n}{2}\right) H\left(\frac{n}{2}\right) \quad (1.2.39)$$

Phổ rời rạc nhận được của tín hiệu lõi ra được trình bày trên hình 1.13.

MATLAB *script* cho bài tập này như sau.

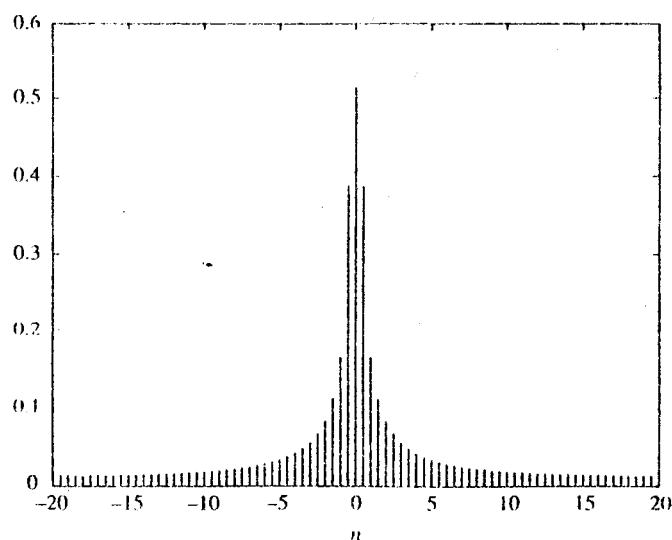
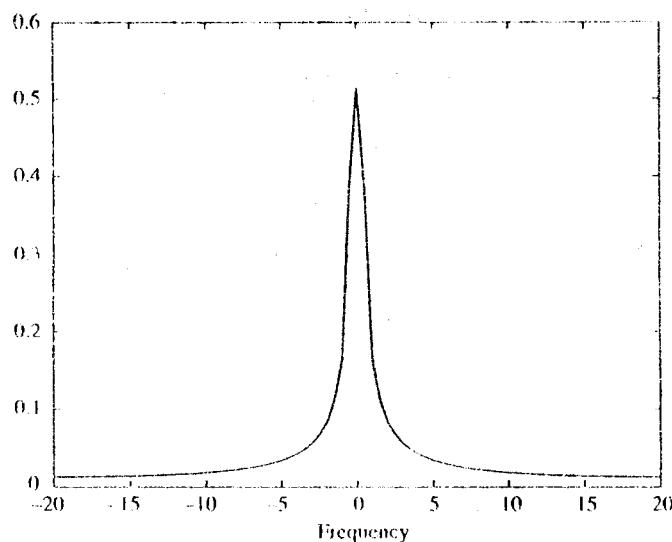
### m-file

```
% MATLAB script for Illustrative Problem 4, Chapter 1.
echo on
n=[-20:1:20];
% Fourier series coefficients of x(t) vector
x=.5*(sinc(n/2)).^2;
% sampling interval
ts=1/40;
% Time vector
t=[-.5:ts:1.5];
% impulse response
fs=1/ts;
```

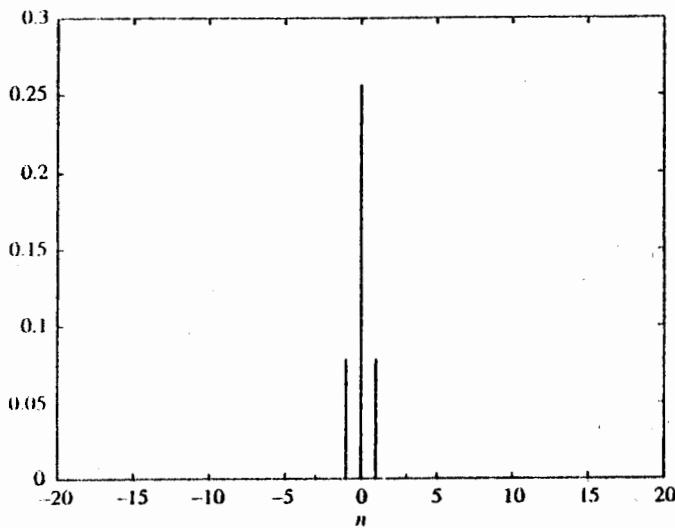
```

h=zeros(1,20),t(21:61),zeros(1,20)];
% transfer function
H=fft(h)/fs;
% frequency resolution
df=fs/80;
f=[0:df:fs]-fs/2;
% rearrange H
H1=fftshift(H);
y=x.*H1(21:61);
% Plotting commands follow

```



Hình 1.12 Hàm truyền của hệ thống LTI trong bài tập minh họa 1.4 và độ lớn của  $H(n/2)$ .



Hình 1.13 Phổ rời rạc của tín hiệu lối ra.

### 1.3 CÁC BIẾN ĐỔI FOURIER

Biến đổi Fourier là sự mở rộng của chuỗi Fourier cho các tín hiệu không tuần hoàn. Biến đổi Fourier của một tín hiệu  $x(t)$  ( $x(t)$  thoả mãn một số điều kiện nhất định gọi là các điều kiện Dirichlet [1]) được ký hiệu là  $X(f)$  hay được ký hiệu một cách tương đương là  $F[x(t)]$  và được xác định bởi

$$F[x(t)] = X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \quad (1.3.1)$$

Biến đổi ngược Fourier của  $X(f)$  là  $x(t)$  được cho theo

$$F^{-1}[X(f)] = x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df \quad (1.3.2)$$

Nếu  $x(t)$  là một tín hiệu thực thì  $X(f)$  thoả mãn tính chất đối xứng Hermit, nghĩa là

$$X(-f) = X^*(f) \quad (1.3.3)$$

Có các tính chất nhất định mà biến đổi Fourier thoả mãn. Các tính chất quan trọng nhất của biến đổi Fourier được tóm tắt như sau.

1. *Tính chất tuyến tính:* Biến đổi Fourier của một tổ hợp tuyến tính của hai hay nhiều tín hiệu là tổ hợp tuyến tính của các biến đổi Fourier tương ứng:

$$F[\alpha x_1(t) + \beta x_2(t)] = \alpha F[x_1(t)] + \beta F[x_2(t)] \quad (1.3.4)$$

2. *Tính đổi ngẫu*: Nếu  $X(f) = F[x(t)]$ , thì

$$F[X(t)] = x(-f) \quad (1.3.5)$$

3. *Dịch thời gian*: Sự dịch chuyển trên miền thời gian dẫn đến dịch pha trên miền tần số. Nếu  $X(f) = F[x(t)]$ , thì

$$F[x(t-t_0)] = e^{-j2\pi f_0 t_0} X(f) \quad (1.3.6)$$

4. *Đổi tỷ lệ xích*: Sự mở rộng trên miền thời gian dẫn đến sự thu hẹp trên miền tần số và ngược lại. Nếu  $X(f) = F[x(t)]$ , thì

$$F[x(at)] = \frac{1}{|a|} X\left(\frac{f}{a}\right), \quad a \neq 0 \quad (1.3.7)$$

5. *Sự điều chế*: Việc nhân trên miền thời gian một tín hiệu với một hàm mũ tương ứng với việc dịch chuyển tần số trên miền tần số. Nếu  $X(f) = F[x(t)]$ , thì

$$\begin{cases} F[e^{j2\pi f_0 t} x(t)] = X(f - f_0) \\ F[x(t) \cos 2\pi f_0 t] = \frac{1}{2} [X(f - f_0) + X(f + f_0)] \end{cases} \quad (1.3.8)$$

6. *Vi phân*: Vi phân trên miền thời gian dẫn đến việc nhân với  $j2\pi f$  trên miền tần số. Nếu  $X(f) = F[x(t)]$ , thì

$$F[x'(t)] = j2\pi f X(f) \quad (1.3.9)$$

$$F\left[\frac{d^n}{dt^n} x(t)\right] = (j2\pi f)^n X(f) \quad (1.3.10)$$

7. *Tích chập*: Tích chập trên miền thời gian thì tương đương với tích thường trên miền tần số và ngược lại. Nếu  $X(f) = F[x(t)]$  và  $Y(f) = F[y(t)]$ , thì

$$F[x(t)*y(t)] = X(f).Y(f) \quad (1.3.11)$$

$$F[x(t).y(t)] = X(f) * Y(f) \quad (1.3.12)$$

8. *Quan hệ Parseval*: Nếu  $X(f) = F[x(t)]$  và  $Y(f) = F[y(t)]$ , thì

$$\int_{-\infty}^{\infty} x(t)y^*(t) dt = \int_{-\infty}^{\infty} X(f)Y^*(f) df \quad (1.3.13)$$

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} |X(f)|^2 df \quad (1.3.14)$$

Quan hệ thứ hai trên đây cũng còn được gọi là quan hệ Rayleigh.

Bảng 1.1 cho các cặp biến đổi Fourier hữu ích nhất. Trong bảng này,  $u_1(t)$  là ký hiệu hàm bậc thang đơn vị;  $\delta(t)$  là tín hiệu xung;  $\text{sgn}(t)$  là hàm *signum* (hàm dấu), được định nghĩa theo

$$\text{sgn}(t) = \begin{cases} 1, & t > 0 \\ 0, & t = 0 \\ -1 & t < 0 \end{cases} \quad (1.3.15)$$

còn  $\delta^{(n)}(t)$  ký hiệu cho đạo hàm bậc  $n$  của tín hiệu xung.

Đối với một tín hiệu  $x(t)$  tuân hoà với chu kỳ  $T_0$  mà các hệ số chuỗi Fourier của nó được cho là  $x_n$ , nghĩa là

$$x(t) = \sum_{n=-\infty}^{\infty} x_n e^{j2\pi nt/T_0}$$

thì biến đổi Fourier nhận được theo

$$X(f) = F[x(t)] = F\left[\sum_{n=-\infty}^{\infty} x_n e^{j2\pi nt/T_0}\right] = \sum_{n=-\infty}^{\infty} x_n F[e^{j2\pi nt/T_0}] = \sum_{n=-\infty}^{\infty} x_n \delta(f - \frac{n}{T_0}) \quad (1.3.16)$$

Nói một cách khác, biến đổi Fourier của một tín hiệu tuân hoà bao gồm các xung tại các tần số là bội của tần số cơ sở (các hài) của tín hiệu gốc.

Cũng có thể biểu diễn các hệ số chuỗi Fourier thông qua biến đổi Fourier của *tín hiệu được cắt ngắn* (*truncated signal*) theo

$$x_n = \frac{1}{T_0} X_{T_0}(\frac{n}{T_0}) \quad (1.3.17)$$

trong đó, theo định nghĩa,  $X_{T_0}(f)$  là biến đổi Fourier của  $x_{T_0}(t)$  là tín hiệu được cắt ngắn xác định theo

$$x_{T_0}(t) = \begin{cases} x(t), & -\frac{T_0}{2} < t \leq \frac{T_0}{2} \\ 0, & \text{Với các giá trị } t \text{ khác} \end{cases} \quad (1.3.18)$$

Biến đổi Fourier của một tín hiệu được gọi là *phổ* của tín hiệu. Phổ của một tín

hiệu nói chung là một hàm phức  $X(f)$ , vì vậy để vẽ đồ thị phô, thường thường phải vẽ hai đồ thị: phô biên độ  $|X(f)|$  và phô pha  $\angle X(f)$ .

Bảng 1.1 Bảng các cặp biến đổi Fourier

$x(t)$	$X(f)$
$\delta(t)$	1
1	$\delta(f)$
$\delta(t-t_0)$	$e^{-j2\pi f t_0}$
$e^{j2\pi f t_0}$	$\delta(f - f_0)$
$\cos 2\pi f_0 t$	$\frac{1}{2}[\delta(f - f_0) + \delta(f + f_0)]$
$\sin 2\pi f_0 t$	$\frac{1}{2j}[\delta(f - f_0) - \delta(f + f_0)]$
$\Pi(t)$	$\text{sinc}(f)$
$\text{sinc}(t)$	$\Pi(f)$
$\Lambda(t)$	$\text{sinc}^2(f)$
$\text{sinc}^2(t)$	$\Lambda(f)$
$e^{-\alpha t} u_{-1}(t), \quad \alpha > 0$	$\frac{1}{\alpha + j2\pi f}$
$te^{-\alpha t} u_{-1}(t), \quad \alpha > 0$	$\frac{1}{(\alpha + j2\pi f)^2}$
$e^{-\alpha t }, \quad \alpha > 0$	$\frac{2\alpha}{\alpha^2 + (2\pi f)^2}$
$e^{-\pi t^2}$	$e^{-\pi f^2}$
$\text{sgn}(t)$	$\frac{1}{j\pi f}$
$u_{-1}(t)$	$\frac{1}{2}\delta(f) + \frac{1}{j2\pi f}$
$\delta'(t)$	$j2\pi f$
$\delta^{(n)}(t)$	$(j2\pi f)^n$
$\sum_{n=-\infty}^{\infty} \delta(t - nT_0)$	$\frac{1}{T_0} \sum_{n=-\infty}^{\infty} \delta(f - \frac{n}{T_0})$

### BÀI TẬP MINH HOA

**Bài tập minh họa 1.5 [các biến đổi Fourier]** Hãy vẽ các phô biên độ và pha của các tín hiệu  $x_1(t)$  và  $x_2(t)$  được cho như trên hình 1.14.

Hình 1.14 Các tín hiệu  $x_1(t)$  và  $x_2(t)$ .

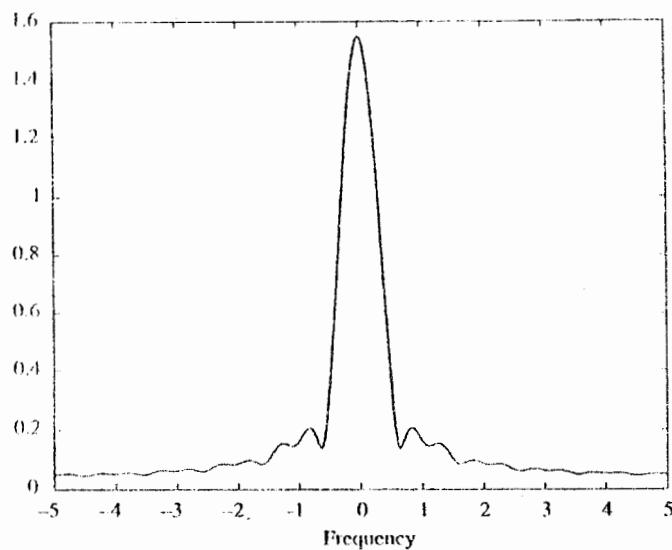
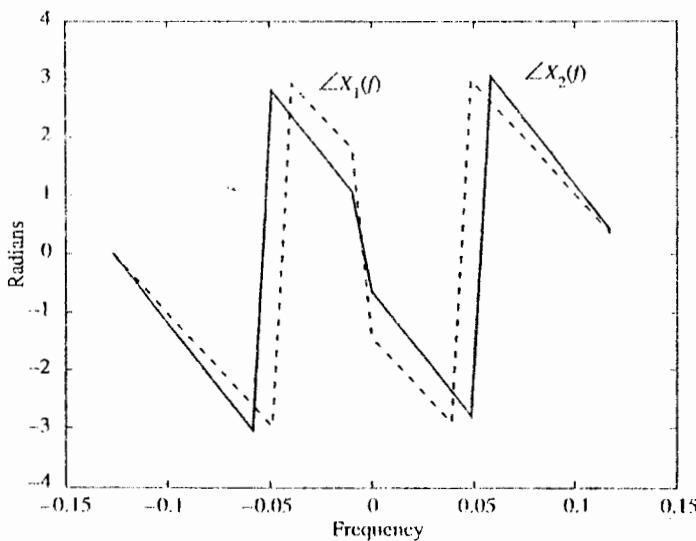
### Lời giải

Do các tín hiệu này tương tự nhau chỉ khác là dịch về thời gian, chúng ta có thể đoán được chúng có cùng phổ biên độ. Phổ biên độ chung và hai phổ pha được vẽ chung trên cùng một hệ trục được trình bày một cách tương ứng trên các hình vẽ 1.15 và 1.16.

MATLAB *script* cho bài tập này cho ở dưới đây. Trong mục 1.3.1, chúng tôi sẽ chỉ cho thấy làm thế nào để có được biến đổi Fourier của một tín hiệu nhờ sử dụng MATLAB.

#### m-file

```
% MATLAB script for Illustrative Problem 5, Chapter 1.
df=0.01;
fs=10;
ts=1/fs;
t=[-5:ts:5];
x1=zeros(size(t));
x1(41:51)=t(41:51)+1;
x1(52:61)=ones(size(x1(52:61)));
x2=zeros(size(t));
x2(51:71)=x1(41:61);
[X1,x11,df1]=fftseq(x1,ts,df);
[X2,x21,df2]=fftseq(x2,ts,df);
X11=X1/fs;
X21=X2/fs;
f=[0:df1:df1*(length(x11)-1)]-fs/2;
plot(f,fftshift(abs(X11)))
figure
plot(f(500:525),fftshift(angle(X11(500:525))),f(500:525),fftshift(angle(X21(500:525))), '--')
```

Hình 1.15 Phổ biên độ chung của các tín hiệu  $x_1(t)$  và  $x_2(t)$ .Hình 1.16 Các phổ pha của các tín hiệu  $x_1(t)$  và  $x_2(t)$ .

### 1.3.1 ĐỊNH LÝ LẤY MẪU

Định lý lấy mẫu là một trong các kết quả quan trọng nhất trong phân tích các hệ thống và tín hiệu; nó tạo ra cơ sở cho mối quan hệ giữa các tín hiệu có thời gian liên tục và các tín hiệu có thời gian rời rạc. Định lý lấy mẫu nói rằng một tín hiệu có băng tần hạn chế – nghĩa là một tín hiệu mà biến đổi Fourier của nó

bằng không với  $|f| > W$  với một giá trị  $W$  nào đó – có thể mô tả được một cách đầy đủ thông qua các giá trị mẫu của nó lấy qua các khoảng  $T_s$  miên là  $T_s \leq 1/2W$ . Nếu việc lấy mẫu được thực hiện qua các khoảng  $T_s = 1/2W$ , được gọi là khoảng Nyquist (hay lấy mẫu với tốc độ Nyquist), tín hiệu  $x(t)$  có thể khôi phục lại được từ các giá trị mẫu của mình  $\{x[n] = x(nT_s)\}_{n=-\infty}^{\infty}$  theo

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT_s) \sin c[2W(t - nT_s)] \quad (1.3.19)$$

Kết quả này dựa trên sự kiện rằng dạng sóng sau lấy mẫu  $x_\delta(t)$  xác định bởi

$$x_\delta(t) = \sum_{n=-\infty}^{\infty} x(nT_s) \delta(t - nT_s) \quad (1.3.20)$$

thì có biến đổi Fourier được cho theo

$$\begin{aligned} X_\delta(f) &= \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(f - \frac{n}{T_s}) \quad \text{với mọi tần số } f \\ &= \frac{1}{T_s} X(f) \quad \text{với } |f| < W \end{aligned} \quad (1.3.21)$$

nên khi cho nó qua một mạch lọc thông thấp có độ rộng băng là  $W$  và có tăng ích là  $T_s$  trong băng thông sẽ cho ra lại tín hiệu ban đầu.

Hình 1.17 là một biểu diễn của (1.3.19) đối với  $T_s = 1$  và  $\{x[n]\}_{n=-3}^3 = \{1, 1, -1, 2, -2, 1, 2\}$ . Nói cách khác,

$$x(t) = \text{sinc}(t+3) + \text{sinc}(t+2) - \text{sinc}(t+1) + 2\text{sinc}(t) - 2\text{sinc}(t-1) + \text{sinc}(t-2) + 2\text{sinc}(t-3)$$

Biến đổi Fourier rời rạc (DFT: *Discrete Fourier Transform*) của chuỗi rời rạc theo thời gian  $x[n]$  được biểu diễn theo

$$X_d(f) = \sum_{n=-\infty}^{\infty} x[n] e^{-j2\pi fnT_s} \quad (1.3.22)$$

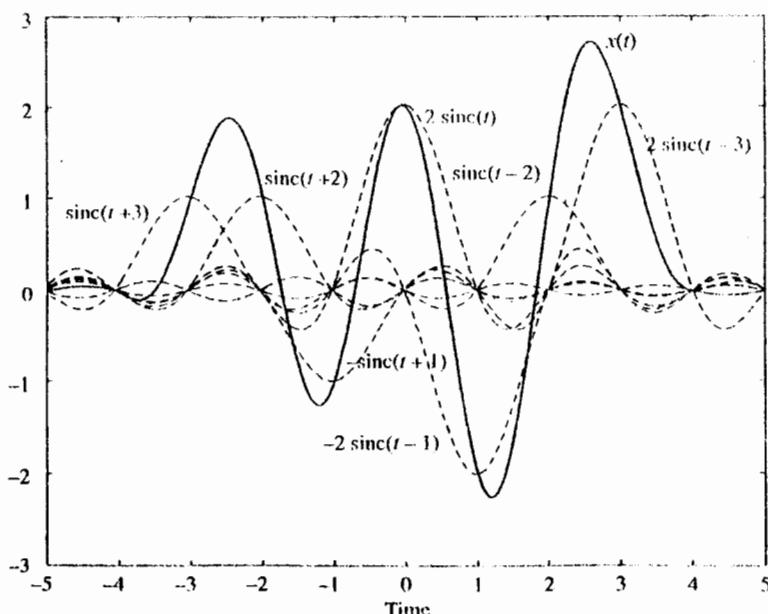
So sánh các phương trình (1.3.22) và (1.3.21), chúng ta có kết luận rằng

$$X(f) = T_s X_d(f) \quad \text{với } |f| < W \quad (1.3.23)$$

mà nó cho chúng ta mối quan hệ giữa biến đổi Fourier của một tín hiệu tương tự

và biến đổi Fourier rời rạc của tín hiệu được lấy mẫu tương ứng của nó.

Việc tính toán số biến đổi Fourier rời rạc được thực hiện thông qua thuật toán biến đổi Fourier nhanh (FFT: *Fast Fourier Transform*) nổi tiếng. Trong thuật toán này, một dãy có độ dài  $N$  các mẫu của tín hiệu  $x(t)$  lấy tại các khoảng  $T_s$  được sử dụng như biểu diễn của tín hiệu. Kết quả của biến đổi là một dãy có độ dài  $N$  mẫu của  $X_k(f)$  trong khoảng tần số  $[0, f_s]$ , trong đó  $f_s = 1/T_s = 2W$  là tần số Nyquist. Khi các mẫu nằm cách nhau  $\Delta f = f_s/N$ , giá trị của  $\Delta f$  cho chúng ta độ phân giải tần số của biến đổi Fourier nhận được. Thuật toán FFT hiệu quả về mặt tính toán nếu độ dài của dãy lối vào,  $N$ , là một luỹ thừa của 2. Trong nhiều trường hợp, nếu độ dài này không phải là một luỹ thừa của 2 thì nó sẽ được làm thành một luỹ thừa của 2 bằng các kỹ thuật như là kỹ thuật “nhồi thêm không” (*zero-padding*). Hãy chú ý rằng do thuật toán FFT về cơ bản cho chúng ta biến đổi Fourier rời rạc (DFT) của tín hiệu đã được lấy mẫu nên để nhận được biến đổi Fourier của tín hiệu tương tự chúng ta phải áp dụng biểu thức (1.3.23). Điều đó có nghĩa là sau khi tính FFT, chúng ta phải nhân kết quả với  $T_s$  hay một cách tương đương là chia cho  $f_s$  nhằm nhận được biến đổi Fourier của tín hiệu *analog* gốc.



Hình 1.17 Biểu diễn định lý lấy mẫu.

Hàm MATLAB **fftseq.m** được cho dưới đây nhận đầu vào của nó là một dãy theo thời gian  $m$ , khoảng thời gian lấy mẫu  $t_s$ , và độ phân giải tần số yêu cầu  $df$  rồi chuyển thành một dãy có độ dài là một luỹ thừa của 2, biến đổi FFT của dãy  $M$  này và độ phân giải tần số yêu cầu.

### m-file

```
function [M,m,df]=fftseq(m,ts,df)
%      [M,m,df]=fftseq(m,ts,df)
%      [M,m,df]=fftseq(m,ts)
%FFTSEQ Generates M, the FFT of the sequence m.
%      The sequence is zero padded to meet the required frequency resolution df.
%      ts is the sampling interval. The output df is the final frequency resolution.
%      Output m is the zero padded version of input m. M is the FFT.
fs=1/ts;
if nargin == 2
    n1=0;
else
    n1=fs/df;
end
n2=length(m);
n=2^(max(nextpow2(n1),nextpow2(n2)));
M=fft(m,n);
m=[m,zeros(1,n-n2)];
df=fs/n;
```

## BÀI TẬP MINH HOẠ

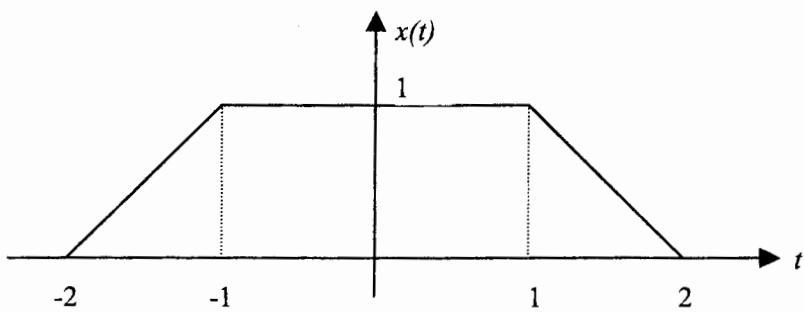
**Bài tập minh họa 1.6 [Tính toán số và giải tích biến đổi Fourier]** Tín hiệu  $x(t)$  được xác định bởi

$$x(t) = \begin{cases} t + 2, & -2 \leq t \leq -1 \\ 1, & -1 < t \leq 1 \\ -t + 2 & 1 < t \leq 2 \\ 0, & \text{Với các giá trị } t \text{ khác} \end{cases} \quad (1.3.24)$$

và được thể hiện như trên hình vẽ 1.18.

1. Hãy xác định biến đổi Fourier của  $x(t)$  một cách giải tích và vẽ đồ thị phổ của  $x(t)$ .
2. Bằng cách sử dụng MATLAB, hãy xác định theo phương pháp tính số biến đổi

Fourier và vẽ ra kết quả.

Hình 1.18 Tín hiệu  $x(t)$ .

### Lời giải

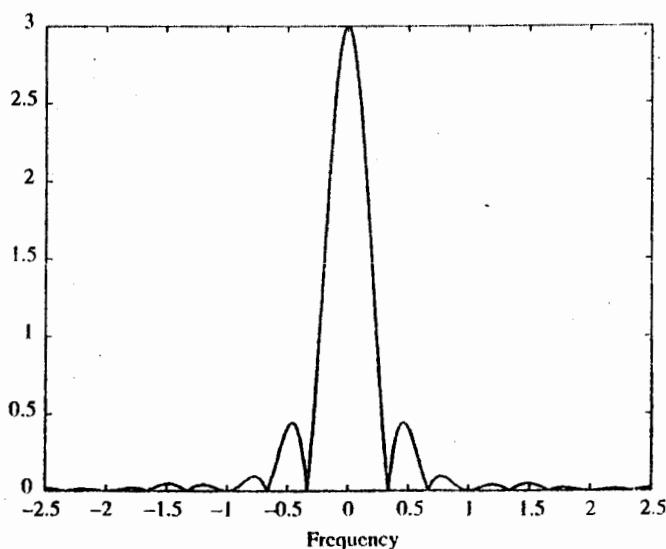
1. Tín hiệu  $x(t)$  có thể viết được thành

$$x(t) = 2\Lambda\left(\frac{t}{2}\right) - \Lambda(t) \quad (1.3.25)$$

và do đó

$$X(f) = 4\text{sinc}^2(2f) - \text{sinc}^2(f) \quad (1.3.26)$$

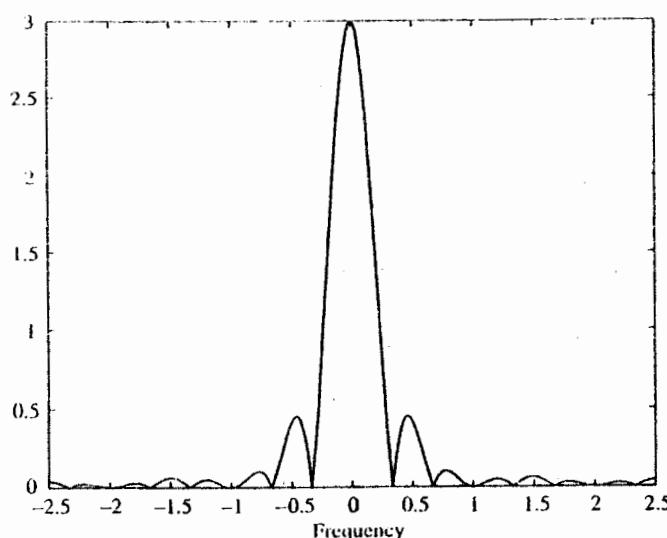
trong đó chúng ta đã sử dụng tính chất tuyến tính, tính chất đổi tỷ lệ xích và sự kiện biến đổi Fourier của  $\Lambda(t)$  là  $\text{sinc}^2(f)$ . Hiển nhiên, biến đổi Fourier là thực. Phổ biên độ được thể hiện trên hình 1.19.

Hình 1.19 Phổ biên độ của  $x(t)$ , có được một cách giải tích.

2. Để xác định biến đổi Fourier bằng cách sử dụng MATLAB, trước tiên chúng ta hãy cho một ước lượng về độ rộng băng tần của tín hiệu. Do tín hiệu khá tròn chu nên độ rộng băng của nó tỷ lệ với nghịch đảo thời gian tồn tại tín hiệu. Thời gian tồn tại của tín hiệu này là 4. Để cho an toàn chúng ta sẽ lấy độ rộng băng là 10 lần nghịch đảo của thời gian tồn tại tín hiệu, hay

$$BW = 10 \times \frac{1}{4} = 2.5 \quad (1.3.27)$$

và do vậy tần số Nyquist là hai lần của độ rộng băng tín hiệu và bằng 5. Như thế, khoảng cách lấy mẫu là  $T_s = 1/f_s = 0.2$ . Chúng ta xét tín hiệu trên khoảng  $[-4, 4]$  và lấy mẫu nó theo các khoảng  $T_s$ . Với sự lựa chọn như thế, bằng cách sử dụng một MATLAB *script* đơn giản có áp dụng hàm **fftseq.m** chúng ta có thể tính FFT được bằng số. Chúng tôi đã chọn độ phân giải tần số yêu cầu là 0.01 Hz nên độ phân giải tần số kết quả được chuyển đổi bởi **fftseq.m** là 0.0098 Hz thoả mãn với yêu cầu của bài toán. Véc-tơ tín hiệu  $x$  có độ dài 41 được nhồi thêm không tới độ dài 256 để thoả mãn yêu cầu độ phân giải tần số và cũng là để làm cho độ dài thành một luỹ thừa của 2 nhằm đạt hiệu quả tính toán cao. Một đồ thị phổ biên độ của biến đổi Fourier được cho trên hình vẽ 1.20.



Hình 1.20 Phổ biên độ của  $x(t)$  tính bằng số.

MATLAB script cho bài tập này được cho dưới đây.

### m-file

```
% MATLAB script for Illustrative Problem 6, Chapter 1
echo on
ts=0.2; % set parameters
fs=1/ts;
df=0.01;
x=[zeros(1,10),[0:0.2:1],ones(1,9),[1:-0.2:0],zeros(1,10)];
[X,x,df1]=fftseq(x,ts,df); % derive the FFT
X1=X/fs; % scaling
f=[0:df1:df1*(length(x)-1)]-fs/2; % frequency vector for FFT
f1=[-2.5:0.001:2.5]; % frequency vector for analytic approach
y=4*(sinc(2*f1)).^2-(sinc(f1)).^2; % Exact Fourier Transform
pause % Press akey to see the plot of the Fourier Transform derived analytically
clf
subplot(2,1,1)
plot(f1,abs(y));
xlabel('Frequency')
title('Magnitude spectrum of x(t) derived analytically')
pause % Press akey to see the plot of the Fourier Transform derived numerically
subplot(2,1,2)
plot(f,fftshift(abs(X1)));
xlabel('Frequency')
title('Magnitude Spectrum of x(t) derived numerically')
```

### 1.3.2 PHÂN TÍCH TRÊN MIỀN TẦN SỐ CÁC HỆ THỐNG LTI

Tín hiệu lối ra của một hệ thống LTI có phản ứng xung  $h(t)$  khi tín hiệu lối vào là  $x(t)$  được cho bởi tích chập

$$y(t)=x(t)*h(t) \quad (1.3.28)$$

Bằng cách áp dụng định lý chập, chúng ta nhận được

$$Y(f)=X(f)H(f) \quad (1.3.29)$$

trong đó

$$H(f) = F[h(t)] = \int_{-\infty}^{\infty} h(t) e^{-j2\pi ft} dt \quad (1.3.30)$$

là hàm truyền của hệ thống. Phương trình (1.3.29) có thể viết được dưới dạng

$$\begin{cases} |Y(f)| = |X(f)|H(f) \\ \angle Y(f) = \angle X(f) + \angle H(f) \end{cases} \quad (1.3.31)$$

cho chúng ta quan hệ giữa các phô biên độ và phô pha của các tín hiệu lõi ra và lõi vào.

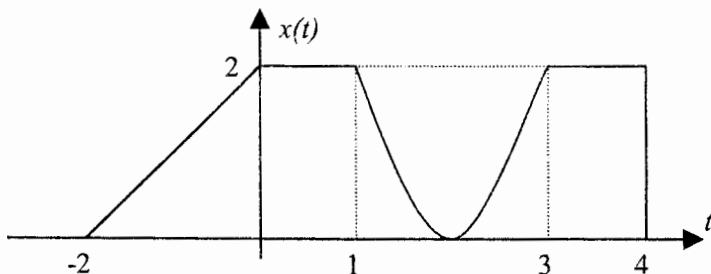
### BÀI TẬP MINH HOẠ

**Bài tập minh họa 1.7 [Phân tích hệ thống LTI trên miền tần số]** Tín hiệu  $x(t)$  có đồ thị như trên hình vẽ 1.21 gồm một số đoạn thẳng và một đoạn hình *sine*.

1. Hãy xác định FFT của tín hiệu này và vẽ đồ thị của nó.
2. Nếu tín hiệu đi qua một bộ lọc thông thấp lý tưởng có độ rộng băng là 1.5 Hz, hãy tìm tín hiệu lõi ra của mạch lọc và vẽ đồ thị của nó.
3. Nếu tín hiệu được cho qua một bộ lọc có phản ứng xung được cho bởi

$$h(t) = \begin{cases} t, & 0 \leq t < 1 \\ 1, & 1 \leq t \leq 2 \\ 0, & \text{Với các giá trị } t \text{ khác} \end{cases} \quad (1.3.32)$$

hãy vẽ đồ thị tín hiệu lõi ra.



Hình 1.21 Tín hiệu  $x(t)$ .

### Lời giải

Trước hết chúng ta tìm biểu thức của phần hình *sine* của tín hiệu. Đây là một hình *sine* với nửa chu kỳ bằng 2; do đó nó có tần số  $f_0=1/4=0.25$  Hz. Tín hiệu có một biên độ bằng 2 và được nâng lên cao độ 2, do đó biểu thức tổng quát đối với nó là  $2\cos(2\pi \times 0.25t + \theta) + 2 = 2\cos(0.5\pi t + \theta) + 2$ . Giá trị của góc pha  $\theta$  tìm được bằng cách sử dụng các điều kiện biên

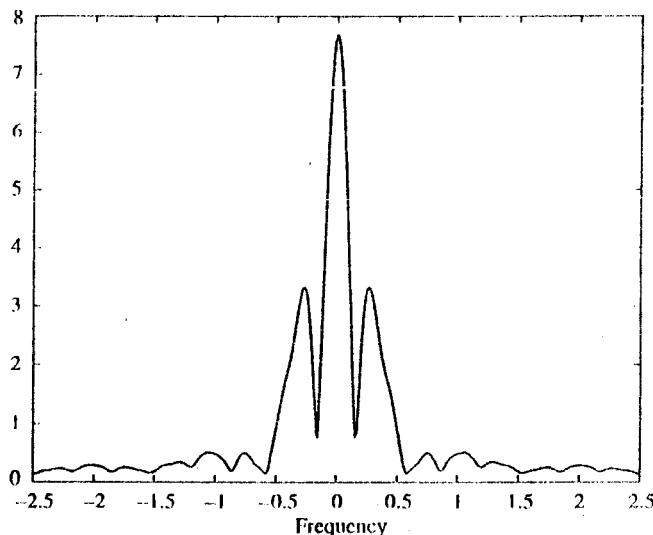
$$2+2\cos(0.5\pi t+\theta) \Big|_{t=2}=0 \quad (1.3.33)$$

hay  $\theta=0$ . Do đó tín hiệu tổng cộng có thể viết được theo

$$x(t) = \begin{cases} t + 2, & -2 \leq t \leq 0 \\ 1, & 0 < t \leq 1 \\ 2 + 2 \cos(0.5\pi t), & 1 < t \leq 3 \\ 1, & 3 < t \leq 4 \\ 0, & \text{Với các giá trị } t \text{ khác} \end{cases} \quad (1.3.34)$$

Khi đã có một mô tả đầy đủ về tín hiệu, chúng ta có thể tiếp tục lời giải.

1. Độ rộng băng của tín hiệu đã được chọn bằng 5 Hz. Độ phân giải tần số yêu cầu là 0.01 Hz. Đồ thị của phổ biên độ của tín hiệu được cho trên hình 1.22.



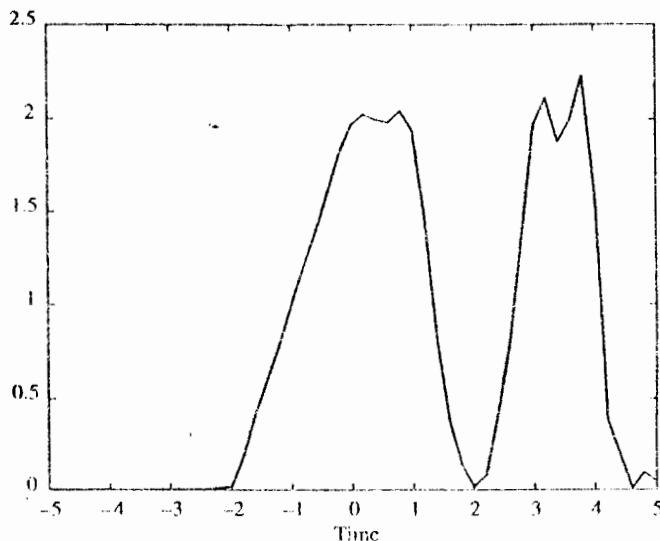
Hình 1.22 Phổ biên độ của tín hiệu.

2. Ở đây  $f_s=5$  Hz. Do độ rộng băng của bộ lọc thông thấp là 1.5 Hz nên hàm truyền của nó được cho bởi

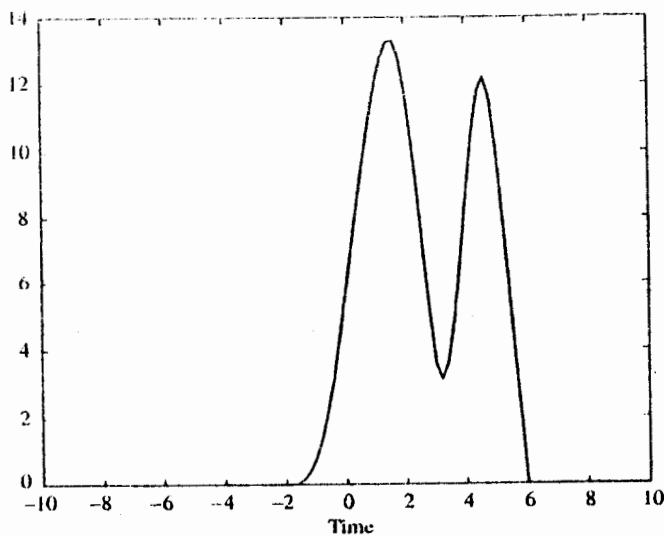
$$H(f) = \begin{cases} 1, & 0 \leq f \leq 1.5 \\ 0, & 1.5 < f \leq 3.5 \\ 1, & 3.5 < f \leq 5 \end{cases} \quad (1.3.35)$$

mà nó được nhân với  $X(f)$  để tạo ra  $Y(f)$  là biến đổi Fourier của tín hiệu lõi ra. Việc sử dụng hàm truyền này cho tín hiệu lõi ra được vẽ trên hình 1.23.

3. Ở đây chúng ta nhận được tín hiệu lỗi ra của bộ lọc bằng một tích chập đơn giản. Kết quả được vẽ trên hình 1.24.



Hình 1.23 Tín hiệu lỗi ra của bộ lọc thông thấp.



Hình 1.24 Tín hiệu lỗi ra trong phần 3 của bài tập minh họa 1.7.

MATLAB *script* cho bài tập này được cho dưới đây.

#### m-file

```
% MATLAB script for Illustrative Problem 7, Chapter 1.  
echo on
```

```

df=0.01; % Freq. resolution
fs=5; % Sampling frequency
ts=1/fs; % Sampling interval
t=[-5:ts:5]; % Time vector
x=zeros(1,length(t)); % Input signal initiation
x(16:26)=t(16:26)+2;
x(27:31)=2*ones(1,5);
x(32:41)=2+2*cos(0.5*pi*t(32:41));
x(42:46)=2*ones(1,5);
% Part 1
[X,x1,df1]=fftseq(x,ts,df); % Spectrum of the input
f=[0:df1:df1*(length(x1)-1)]-fs/2; % frequency vector
X1=X/fs; % Scaling
% Part 2
% Filter transfer function
H=[ones(1,ceil(1.5/df1)),zeros(1,length(X)-2*ceil(1.5/df1)),ones(1,ceil(1.5/df1))];
Y=X.*H; % Output spectrum
y1=ifft(Y); % Output of the filter
% Part 3
% LTI system impulse response
h=[zeros(1,ceil(5/ts)),t(ceil(5/ts)+1:ceil(6/ts)),ones(1,ceil(7/ts)-ceil(6/ts)),zeros(1,51-ceil(7/ts))];
y2=conv(h,x); % Output of the LTI system
pause % Press a key to see spectrum of the input
plot(f,fftshift(abs(X1))) % Press a key to see the output of the lowpass filter
plot(t,abs(y1(1:length(t)))) % Press a key to see the output of the LTI system
plot([-10:ts:10],y2);

```

## 1.4. CÔNG SUẤT VÀ NĂNG LƯỢNG

Công suất và năng lượng của một tín hiệu thực  $x(t)$ , lần lượt được ký hiệu là  $E_x$  và  $P_x$ , được xác định theo

$$\begin{cases} E_x = \int_{-\infty}^{\infty} x^2(t) dt \\ P_x = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} x^2(t) dt \end{cases} \quad (1.4.1)$$

Một tín hiệu với năng lượng hữu hạn được gọi là một tín hiệu *kiểu năng lượng* (*energy-type*), còn một tín hiệu có công suất hữu hạn dương là một tín hiệu *kiểu công suất* (*power-type*)<sup>4</sup>. Thí dụ,  $x(t)=\Pi(t)$  là một thí dụ về một tín hiệu kiểu năng lượng, trong khi đó  $x(t)=\cos(t)$  là một thí dụ về một tín hiệu kiểu công suất. Mọi tín hiệu tuần hoàn<sup>5</sup> đều là các tín hiệu kiểu công suất. *Mật độ phổ năng lượng* của một tín hiệu kiểu năng lượng cho chúng ta phân bố năng lượng tại các tần số khác nhau của tín hiệu và được cho bởi

$$G_X(f) = |X(f)|^2 \quad (1.4.2)$$

Do đó,

$$E_X = \int_{-\infty}^{\infty} G_X(f) df \quad (1.4.3)$$

Bằng cách sử dụng định lý tích chập, chúng ta có

$$G_X(f) = F[R_X(\tau)] \quad (1.4.4)$$

trong đó  $R_X(\tau)$  là hàm tự tương quan của  $x(t)$  được xác định theo

$$R_X(\tau) = \int_{-\infty}^{\infty} x(t)x(t+\tau) dt = x(\tau) * x(-\tau) \quad (1.4.5)$$

đối với các tín hiệu giá trị thực. Đối với các tín hiệu kiểu công suất, chúng ta định nghĩa *hàm tự tương quan trung bình thời gian* (*time-average autocorrelation function*) theo

$$R_X(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t)x(t+\tau) dt \quad (1.4.6)$$

và mật độ phổ công suất nói chung được cho bởi

$$S_X(f) = F[R_X(\tau)] \quad (1.4.7)$$

Công suất tổng cộng là tích phân của mật độ phổ công suất được cho theo

4) Có các tín hiệu chẳng phải là kiểu năng lượng lẫn kiểu công suất. Một thí dụ về các tín hiệu như vậy là  $x(t) = e^t u_1(t)$ .

5) Trường hợp ngoại lệ duy nhất là các tín hiệu bằng không khắp mọi nơi.

$$P_X = \int_{-\infty}^{\infty} S_X(f) df \quad (1.4.8)$$

Trường hợp đặc biệt, đối với một tín hiệu tuần hoàn  $x(t)$  có chu kỳ  $T_0$  và các hệ số chuỗi Fourier là  $x_n$ , mật độ phổ công suất được cho bởi

$$S_X(f) = \sum_{n=-\infty}^{\infty} |x_n|^2 \delta(f - \frac{n}{T_0}) \quad (1.4.9)$$

mà điều này có nghĩa là mọi công suất đều tập trung tại các hài của tần số cơ bản và công suất tại hài thứ  $n$  ( $n/T_0$ ) là  $|x_n|^2$ , nghĩa là bằng bình phương của độ lớn hệ số tương ứng của chuỗi Fourier.

Khi tín hiệu  $x(t)$  đi qua một bộ lọc có hàm truyền  $H(f)$ , mật độ phổ năng lượng hay mật độ phổ công suất của tín hiệu lõi ra nhận được thông qua

$$\begin{cases} G_Y(f) = |H(f)|^2 G_X(f) \\ S_Y(f) = |H(f)|^2 S_X(f) \end{cases} \quad (1.4.10)$$

Nếu chúng ta sử dụng tín hiệu thời gian rời rạc (được lấy mẫu) thì các quan hệ năng lượng và công suất tương đương với phương trình (1.4.1) ở dạng tín hiệu rời rạc về thời gian sẽ trở thành

$$\begin{cases} E_X = T_S \sum_{n=-\infty}^{\infty} x^2[n] \\ P_X = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N x^2[n] \end{cases} \quad (1.4.11)$$

và nếu FFT được áp dụng, nghĩa là nếu độ dài của dãy là hữu hạn còn dãy được lặp đi lặp lại, thì

$$\begin{cases} E_X = T_S \sum_{n=0}^{N-1} x^2[n] \\ P_X = \frac{1}{N} \sum_{n=0}^{N-1} x^2[n] \end{cases} \quad (1.4.12)$$

Hàm MATLAB **power.m** được cho dưới đây cho chúng ta công suất chứa trong

một véc-tơ tín hiệu.

### m-file

```
function p=spower(x)
%
%          p=spower(x)
%SPOWER      Returns the power in signal x
p=(norm(x)^2)/length(x);
```

Nếu  $X_d(f)$  là DFT của dãy  $x[n]$  thì mật độ phổ năng lượng của  $x(t)$ —là tín hiệu *analog* tương đương—sẽ nhận được bằng cách sử dụng phương trình (1.3.23) và được cho bởi

$$G_x(f) = T_s^2 |X_d(f)|^2 \quad (1.4.13)$$

trong đó  $T_s$  là khoảng thời gian lấy mẫu. Mật độ phổ công suất của một dãy  $x[n]$  tìm được dễ dàng nhất bằng cách sử dụng hàm MATLAB **spectrum.m**.

### BÀI TẬP MINH HOA

**Bài tập minh họa 1.8 [Công suất và phổ công suất]** Tín hiệu  $x(t)$  có thời gian tồn tại là 10 và là tổng của hai tín hiệu hình *sine* cùng có biên độ bằng 1, một có tần số 47 Hz, một có tần số 219 Hz:

$$x(t) = \begin{cases} \cos(2\pi \times 47t) + \cos(2\pi \times 219t), & 0 \leq t \leq 10 \\ 0, & \text{Với các giá trị } t \text{ khác} \end{cases}$$

Tín hiệu này được lấy mẫu với tốc độ lấy mẫu là 1000 mẫu trên giây. Hãy sử dụng MATLAB để tìm công suất và mật độ phổ công suất của tín hiệu này.

### Lời giải

Với hàm MATLAB **spower.m**, công suất của tín hiệu tìm được là 1.0003 W. Bằng cách sử dụng **spectrum.m** và **specplot.m**, chúng ta có thể vẽ mật độ phổ công suất của tín hiệu này như được thể hiện trên hình 1.25. Các đỉnh đôi trong phổ công suất tương ứng với hai tần số có mặt trong tín hiệu.

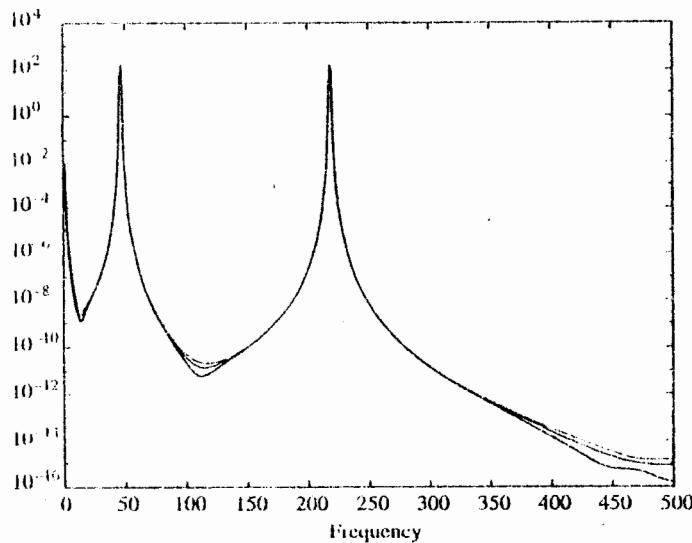
MATLAB script cho bài tập này như sau.

### m-file

```
% MATLAB script for Illustrartive Problem 8, Chapter 1.
ts=0.001;
```

```

fs=1/ts;
t=[0:ts:10];
x=cos(2*pi*47*t)+cos(2*pi*219*t);
p=spower(x);
psd=spectrum(x,1024);
pause % Press a key to see the power in the signal
p
pause % Press a key to see the power spectrum
specplot(psd,fs)
    
```



Hình 1.25 Mật độ phổ công suất của tín hiệu gồm hai tín hiệu hình *sine* tại các tần số  $f_1=47$  Hz và  $f_2=219$  Hz.

## 1.5 TƯƠNG ĐƯƠNG THÔNG THẤP CỦA CÁC TÍN HIỆU DẢI THÔNG

Một tín hiệu dải thông là một tín hiệu mà đối với nó mọi thành phần tần số đều nằm trong một vùng quanh một tần số trung tâm  $f_0$  (và, tất nhiên là quanh cả  $-f_0$ ). Nói khác đi, đối với một tín hiệu dải thông  $X(f) \neq 0$  đối với  $|f \pm f_0| > W$ , trong đó  $W \ll f_0$ . Một tín hiệu thông thấp là một tín hiệu mà với nó các thành phần tần số đều nằm quanh tần số 0, nghĩa là đối với  $|f| > W$  chúng ta có  $X(f) \equiv 0$ .

Tương ứng với một tín hiệu dải thông  $x(t)$  chúng ta có thể xác định *tín hiệu giải tích*  $z(t)$  có biến đổi Fourier được cho theo

$$Z(f) = 2u_{-1}(f)X(f) \quad (1.5.1)$$

trong đó  $u_{-1}(f)$  là hàm bậc thang đơn vị. Trong miền thời gian, quan hệ này được viết theo

$$z(t) = x(t) + j\hat{x}(t) \quad (1.5.2)$$

trong đó  $\hat{x}(t)$  là biến đổi Hilbert của  $x(t)$  được xác định theo  $\hat{x}(t) = x(t) * (1/\pi t)$ ; trong miền tần số, điều này được cho bởi

$$\hat{X}(f) = -j \operatorname{sgn}(f) X(f) \quad (1.5.3)$$

Ta hãy chú ý rằng hàm biến đổi Hilbert trong MATLAB, được ký hiệu là **hiltbert.m**, tạo ra dãy phức  $z(t)$ . Phần thực của  $z(t)$  là dãy tín hiệu gốc (tức là  $x(t)$ ), còn phần ảo của nó chính là biến đổi Hilbert của dãy gốc.

Tương đương thông thấp của tín hiệu  $x(t)$ , được ký hiệu là  $x_l(t)$ , biểu diễn được thông qua  $z(t)$  theo

$$x_l(t) = z(t) e^{-j2\pi f_0 t} \quad (1.5.4)$$

Từ quan hệ này, chúng ta có

$$\begin{cases} x_l(t) = \operatorname{Re}[x_l(t) e^{j2\pi f_0 t}] \\ \hat{x}_l(t) = \operatorname{Im}[x_l(t) e^{j2\pi f_0 t}] \end{cases} \quad (1.5.5)$$

Trong miền tần số, chúng ta có

$$X_l(f) = Z(f + f_0) = 2u_{-1}(f + f_0)X(f + f_0) \quad (1.5.6)$$

$$\text{và } X_l(f) = X(f - f_0) + X^*(-f - f_0) \quad (1.5.7)$$

Nói chung, tương đương thông thấp của một tín hiệu dải thông thực là một tín hiệu phức. Phần thực của nó, ký hiệu là  $x_c(t)$ , được gọi là thành phần cùng pha của  $x(t)$  còn phần ảo của nó được gọi là thành phần vuông góc của  $x(t)$  và được ký hiệu là  $x_s(t)$ , nghĩa là

$$x_l(t) = x_c(t) + j x_s(t) \quad (1.5.8)$$

Biểu diễn thông qua các thành phần cùng pha và vuông góc chúng ta có

$$\begin{cases} x(t) = x_c(t) \cos(2\pi f_0 t) - x_s(t) \sin(2\pi f_0 t) \\ \hat{x}(t) = x_s(t) \cos(2\pi f_0 t) + x_c(t) \sin(2\pi f_0 t) \end{cases} \quad (1.5.9)$$

Nếu chúng ta biểu diễn  $x_l(t)$  trong hệ toạ độ cực thì chúng ta có

$$\dot{x}_l(t) = V(t)e^{j\Theta(t)} \quad (1.5.10)$$

trong đó  $V(t)$  và  $\Theta(t)$  được gọi là đường bao và pha của tín hiệu  $x(t)$ . Biểu diễn thông qua hai hàm này, chúng ta có

$$x(t) = V(t)\cos[2\pi f_0 t + \Theta(t)] \quad (1.5.11)$$

Đường bao và pha có thể biểu diễn được theo

$$\begin{cases} V(t) = \sqrt{x_c^2(t) + x_s^2(t)} \\ \Theta(t) = \arctan \frac{x_s(t)}{x_c(t)} \end{cases} \quad (1.5.12)$$

hay tương đương với

$$\begin{cases} V(t) = \sqrt{x^2(t) + \hat{x}^2(t)} \\ \Theta(t) = \arctan \frac{\hat{x}(t)}{x(t)} - 2\pi f_0 t \end{cases} \quad (1.5.13)$$

Từ các quan hệ trên đây, hiển nhiên là đường bao thì độc lập với việc lựa chọn  $f_0$  trong khi đó pha thì phụ thuộc vào việc lựa chọn này.

Chúng tôi đã viết một vài file MATLAB đơn giản để tạo ra tín hiệu giải tích, biểu diễn thông thấp của một tín hiệu, các thành phần cùng pha và vuông góc, đường bao và pha. Các hàm MATLAB tương ứng là **analytic.m**, **loweq.m**, **quadcomp.m** và **env\_phas.m**. Các hàm này được liệt ra dưới đây.

### m-file

```
function z=analytic(x)
%
%          z=analytic(x)
%
%ANALYTIC      Returns the analytic signal corresponding to signal x.
z=hilbert(x);
```

### m-file

```
function xl=loweq(x,ts,f0)
%
%      xl=loweq(x,ts,f0)
%
%LOWEQ        Returns the lowpass equivalent of the signal x.
%
%      f0 is the center frequency. ts is the sampling interval.
t=[0:ts:ts*(length(x)-1)];
z=hilbert(x);
xl=z.*exp(-j*2*pi*f0*t);
```

### m-file

```
function [xc,xs]=quadcomp(x,ts,f0)
% [xc,xs]=quadcomp(x,ts,f0)
%QUADCOMP      Returns the in-phase and quadrature components of the
%               signal x. f0 is the center frequency. ts is the sampling interval.
z=loweq(x,ts,f0);
xc=real(z);
xs=imag(z);
```

### m-file

```
function [v,phi]=env_phas(x,ts,f0)
% [v,phi]=env_phas(x,ts,f0)
% v=env_phas(x,ts,f0)
%ENV_PHAS      Returns the envelope and the phase of the bandpass signal x.
%               f0 is the center frequency. ts is the sampling interval.
if nargout == 2
    z=loweq(x,ts,f0);
    phi=angle(z);
end
v=abs(hilbert(x));
```

## BÀI TẬP MINH HỌA

**Bài tập minh họa 1.9 [Biến đổi từ dải thông thành thông thấp]** Tín hiệu  $x(t)$  được cho theo

$$x(t)=\text{sinc}(100t)\cos(2\pi \times 200t) \quad (1.5.14)$$

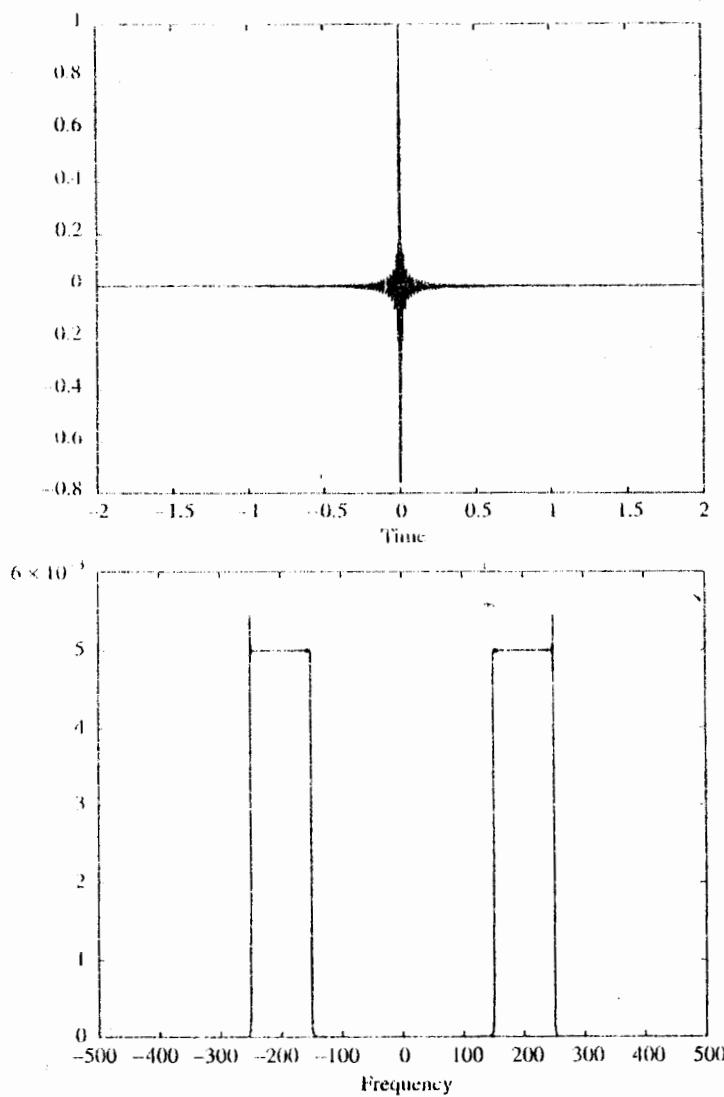
1. Hãy vẽ đồ thị tín hiệu này và đồ thị phổ biên độ của nó.
2. Với  $f_0=200$  Hz, hãy tìm tần số trung trung thông thấp và vẽ đồ thị phổ biên độ của nó. Hãy vẽ đồ thị các thành phần cùng pha và vuông góc, đường bao của tín hiệu này.
3. Lặp lại phần 2 với giả thiết  $f_0=100$  Hz.

### Lời giải

Bằng cách chọn khoảng lấy mẫu là  $t_s=0.001$  s, chúng ta có một tần số lấy mẫu là  $f_s=1/t_s=1000$  Hz. Bằng cách chọn một độ phân giải tần số mong muốn là  $df=0.5$  Hz, chúng ta có như sau.

1. Các đồ thị của tín hiệu và phổ biên độ của nó được cho trên hình 1.26. Các

đồ thị được tạo ra bằng MATLAB.

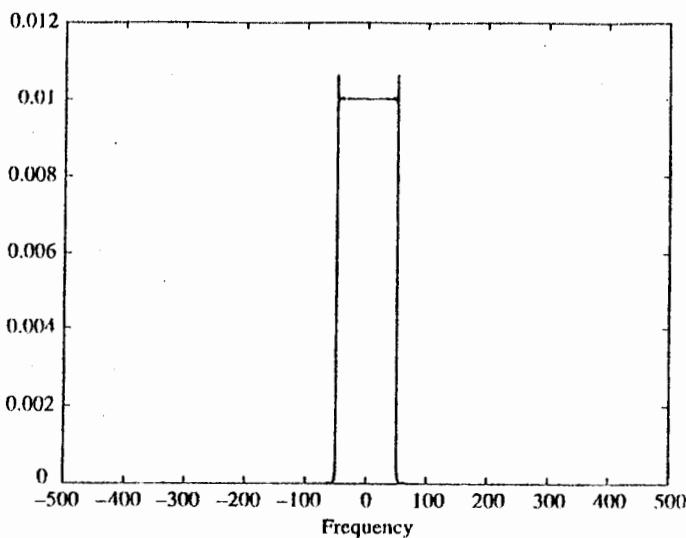


Hình 1.26 Tín hiệu  $x(t)$  và phổ biên độ của nó

2. Bằng cách chọn  $f_0=200$  Hz, chúng ta tìm tương đương thông thấp với  $x(t)$  bằng cách sử dụng hàm **loweq.m**. Sau đó, bằng cách sử dụng **fftseq.m**, chúng ta nhận được phổ của nó; đồ thị phổ biên độ của nó trên hình 1.27. Ta có thể thấy được rằng phổ biên độ là một hàm chẵn trong trường hợp này bởi vì chúng ta có thể viết

$$x(t)=\operatorname{Re}[\operatorname{sinc}(100t)e^{j2\pi \times 200t}] \quad (1.5.15)$$

$$\text{So sánh với } x(t) = \operatorname{Re}[x_i(t)e^{j2\pi \times f_0 t}] \quad (1.5.16)$$



Hình 1.27 Phổ biên độ của tín hiệu tương đương thông thấp đối với  $x_l(t)$  trong bài tập minh họa 1.9 khi  $f_0=200$  Hz.

chúng ta kết luận được rằng

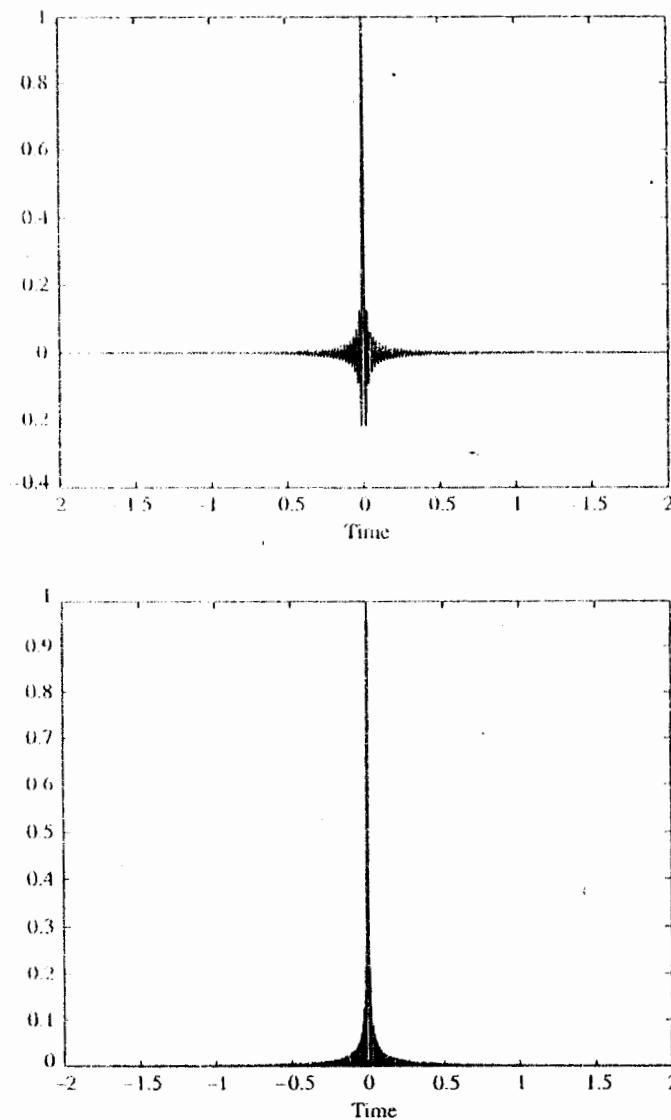
$$x_l(t)=\text{sinc}(100t) \quad (1.5.17)$$

mà điều đó có nghĩa là trong trường hợp này tín hiệu tương đương thông thấp là một tín hiệu thực. Điều này, đến lượt mình lại có nghĩa là  $x_c(t)=x_l(t)$  còn  $x_s(t)=0$ . Chúng ta cũng kết luận được rằng

$$\begin{cases} V(t) = |x_c(t)| \\ \Theta(t) = \begin{cases} 0, & x_c(t) \geq 0 \\ \pi, & x_c(t) < 0 \end{cases} \end{cases} \quad (1.5.18)$$

Các đồ thị của  $x_c(t)$  và  $V(t)$  được cho trên hình 1.28. Hãy để ý rằng việc chọn  $f_0$  làm tần số mà  $X(f)$  đối xứng qua đó dẫn đến các hình vẽ này.

3. Nếu  $f_0=100$  Hz thì các kết quả trên nói chung sẽ không còn đúng nữa và  $x_l(t)$  sẽ là một số phức. Phổ biên độ của tín hiệu tương đương thông thấp được vẽ trên hình 1.29. Như có thể thấy ở đây, phổ biên độ thiếu tính đối xứng hiện hữu trong biến đổi Fourier của các tín hiệu thực. Các đồ thị của thành phần cùng pha của  $x(t)$  và đường bao của nó được cho trên hình 1.30.



Hình 1.28 Thành phần cùng pha và đường bao của  $x(t)$ .

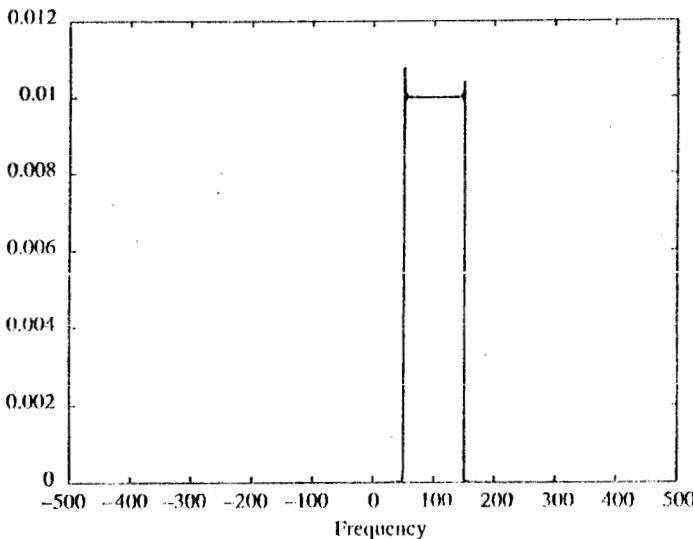
## CÁC BÀI TẬP

**1.1** Hãy xét tín hiệu tuần hoàn của bài tập minh họa 1.1 được thể hiện trên hình 1.1. Giả sử  $A=1$ ,  $T_0=10$  và  $t_0=1$ , hãy xác định và vẽ phổ rời rạc của tín hiệu này. Hãy so sánh các kết quả của bạn với các kết quả đã nhận được trong bài tập minh họa 1.1 và hãy chứng minh rằng các sai khác đó là đúng.

**1.2** Trong bài tập minh họa 1.1, giả sử rằng  $A=1$ ,  $T_0=4$  và  $t_0=2$ , hãy xác định và vẽ

phổ rời rạc của tín hiệu. Hãy so sánh các kết quả của bạn với các kết quả đã nhận được trong bài tập minh họa 1.1 và hãy chứng minh rằng các sai khác đó là đúng.

**1.3** Sử dụng *m-file fseries.m*, hãy xác định các hệ số chuỗi Fourier của tín hiệu đã cho trên hình 1.1 với  $A=1$ ,  $T_0=4$  và  $t_0=1/2$  với  $-24 \leq n \leq 24$ . Hãy vẽ đồ thị phổ biên độ của tín hiệu. Bây giờ, bằng cách sử dụng phương trình (1.2.5), hãy xác định các hệ số chuỗi Fourier và vẽ phổ biên độ. Tại sao các kết quả không tuyệt đối giống nhau.



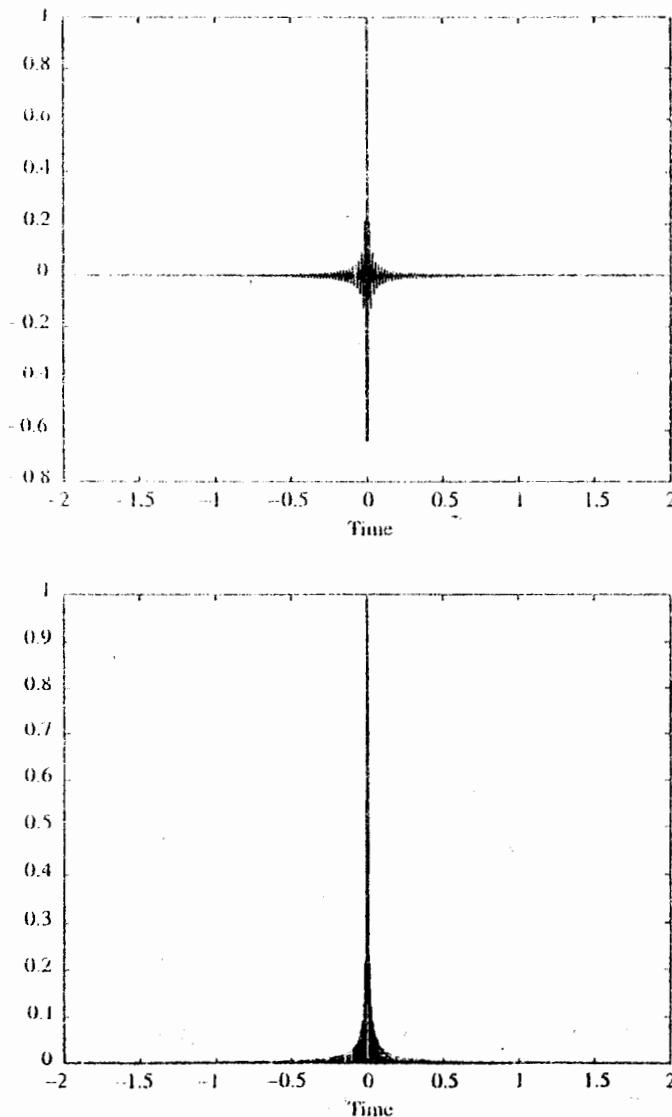
Hình 1.29 Phổ biên độ của tín hiệu tương đương thông thấp với  $x(t)$  trong bài tập minh họa 1.9 khi  $f_0=100$  Hz.

**1.4** Lặp lại bài tập 1.3 với  $T_0=4.6$  và so sánh các kết quả với các kết quả nhận được bằng cách sử dụng phương trình (1.2.5). Bạn có quan sát thấy ở đây có cùng sự không nhất quán giữa hai kết quả không? Tại sao?

**1.5** Sử dụng MATLAB *script dis\_spect.m*, hãy xác định và vẽ các phổ biên độ và phổ pha của tín hiệu tuần hoàn  $x(t)$  có chu kỳ  $T_0=4.6$  và được mô tả trong khoảng  $[-2.3, 2.3]$  bằng quan hệ  $x(t)=\Lambda(t)$ . Hãy vẽ các phổ đó với  $-24 \leq n \leq 24$ . Bây giờ, hãy xác định một cách giải tích các hệ số của chuỗi Fourier của tín hiệu và hãy chứng minh rằng mọi hệ số đó đều là các số thực không âm. Phổ pha mà bạn đã vẽ trước đó có thống nhất với kết quả này hay không? Nếu không thì hãy giải thích tại sao.

**1.6** Trong bài tập 1.5, hãy định nghĩa  $x(t)=\Lambda(t)$  trong khoảng  $[-1.3, 3.3]$ ; chu kỳ bấy

giờ vẫn là  $T_0=4.6$ . Hãy xác định và vẽ các phổ biên độ và pha bằng cách sử dụng MATLAB script **dis\_spect.m**. Hãy chú ý rằng tín hiệu này thì giống như tín hiệu trong bài tập 1.5. Hãy so sánh phổ biên độ và phổ pha với các phổ biên độ và pha nhận được trong bài tập 1.5. Phổ biên độ hay phổ pha cho thấy một sai biệt đáng kể hơn? Tại sao?



Hình 1.30 Thành phần cùng pha và đường bao của tín hiệu  $x(t)$  khi  $f_0=100$  Hz.

**1.7** Lặp lại bài tập minh họa 1.2 với  $[a, b]=[-4, 4]$  và  $x(t)=\cos(\pi t/8)$  đối với  $|t| \leq 4$ .

**1.8** Lặp lại bài tập minh họa 1.2 với  $[a, b]=[-4, 4]$  và  $x(t)=\sin(\pi t/8)$  đối với  $|t| \leq 4$  rồi

so sánh các kết quả của bạn với kết quả nhận được trong bài tập 1.7.

**1.9** Hãy xác định bằng số và vẽ các phổ biên độ và pha của một tín hiệu  $x(t)$  có chu kỳ là  $10^{-6}$  giây và được xác định theo

$$x(t) = \begin{cases} -10^6 t + 0.5, & 0 \leq t \leq 5 \times 10^{-7} \\ 0, & \text{Với các giá trị } t \text{ khác} \end{cases}$$

trong khoảng  $|t| \leq 5 \times 10^{-7}$ .

**1.10** Một tín hiệu  $x(t)$  có chu kỳ  $T_0=6$  được xác định bởi  $x(t)=\Pi(t/3)$  đối với  $|t| \leq 3$ .

Tín hiệu này đi qua một hệ thống LTI có phản ứng xung được cho bởi

$$h(t) = \begin{cases} e^{-t/2}, & 0 \leq t \leq 4 \\ 0, & \text{Với các giá trị } t \text{ khác} \end{cases}$$

hãy tính bằng số và vẽ phổ rời rạc của tín hiệu lối ra.

**1.11** Lặp lại bài tập 1.10 với  $x(t)=e^{-2t}$  đối với  $|t| \leq 3$  và

$$h(t) = \begin{cases} 1, & 0 \leq t \leq 4 \\ 0, & \text{Với các giá trị } t \text{ khác} \end{cases}$$

**1.12** Hãy kiểm chứng định lý tích chập của biến đổi Fourier bằng tính toán số đối với các tín hiệu  $x(t)=\Pi(t)$  và  $y(t)=\Lambda(t)$ , một lần làm bằng cách xác định trực tiếp tích chập, một lần bằng cách sử dụng các biến đổi Fourier của hai tín hiệu đó.

**1.13** Hãy vẽ các đồ thị phổ biên độ và phổ pha của một tín hiệu cho bởi

$$x(t) = \begin{cases} 1, & -2 \leq t \leq -1 \\ |t|, & |t| < 1 \\ 1, & 1 \leq t \leq 2 \\ 0, & \text{Với các giá trị } t \text{ khác} \end{cases}$$

**1.14** Hãy xác định và vẽ phổ của một tín hiệu chẵn  $x(t)$  mà đối với các giá trị dương của  $t$  nó được cho bởi

$$x(t) = \begin{cases} t+1, & 0 \leq t \leq 1 \\ 2, & 1 \leq t \leq 2 \\ -t+4, & 2 \leq t \leq 4 \\ 0, & \text{Với các giá trị } t \text{ khác} \end{cases}$$

**1.15** Tín hiệu đã được mô tả trong bài tập 1.14 được cho qua một hệ thống LTI có hàm phản ứng xung được cho theo

$$h(t) = \begin{cases} 1, & 0 \leq t \leq 2 \\ 2, & 2 < t \leq 3 \\ 0, & \text{Với các giá trị } t \text{ khác} \end{cases}$$

Hãy xác định các phổ biên độ và pha của tín hiệu lõi ra.

**1.16** Tín hiệu

$$x(t) = \begin{cases} \cos(2\pi \times 47t) + \cos(2\pi \times 219t), & 0 \leq t \leq 10 \\ 0, & \text{Với các giá trị } t \text{ khác} \end{cases}$$

được xem xét. Như trong bài tập minh họa 1.8, giả sử tín hiệu này được lấy mẫu với tốc độ 1000 mẫu/giây. Bằng cách sử dụng MATLAB *m-file butter.m*, hãy thiết kế một mạch lọc Butterworth thông thấp bậc 4 có tần số cắt 100 Hz rồi cho  $x(t)$  qua bộ lọc này. Hãy xác định và vẽ phổ công suất của tín hiệu lõi ra và so sánh nó với hình 1.25. Bây giờ hãy thiết kế một bộ lọc Butterworth bậc 8 với cùng một tần số cắt, hãy xác định tín hiệu lõi ra của mạch lọc đó và vẽ phổ công suất của nó. Hãy so sánh các kết quả của bạn trong hai trường hợp này.

**1.17** Lặp lại bài tập 1.16 song lần này hãy thiết kế các bộ lọc Butterworth thông cao với cùng các tần số cắt và cùng bậc như trong bài tập 1.16. hãy vẽ các kết quả của bạn và tiến hành so sánh chúng.

**1.18** Hãy xét tín hiệu

$$x(t) = \begin{cases} \cos(2\pi \times 47t) + \cos(2\pi \times 219t), & 0 \leq t \leq 10 \\ 0, & \text{Với các giá trị } t \text{ khác} \end{cases}$$

- a. Hãy xác định tín hiệu giải tích tương ứng với tín hiệu này.
- b. Hãy xác định và vẽ biến đổi Hilbert của tín hiệu này.
- c. Hãy xác định và vẽ đường bao của tín hiệu này.
- d. Một lần giả sử  $f_0 = 47$  và một lần giả sử  $f_0 = 219$ , hãy xác định tín hiệu thông thấp tương đương và các thành phần cùng pha và vuông góc của tín hiệu này.

## CHƯƠNG 2

# CÁC QUÁ TRÌNH NGẪU NHIÊN

### 2.1 GIỚI THIỆU CHUNG

Trong chương này chúng tôi minh họa các phương pháp tạo các biến ngẫu nhiên và các mẫu của các quá trình ngẫu nhiên. Chúng ta bắt đầu với việc mô tả một phương pháp tạo các biến ngẫu nhiên có một hàm phân bố xác suất xác định. Sau đó chúng ta sẽ xem xét các quá trình ngẫu nhiên Gauss (chuẩn) và Gauss-Markov và minh họa một phương pháp tạo các mẫu của các quá trình như thế. Chủ đề thứ ba mà chúng ta sẽ xem xét là việc đặc tả một quá trình ngẫu nhiên dùng bằng hàm tự tương quan của nó trên miền thời gian và bằng phổ công suất của nó trên miền tần số. Do các mạch lọc tuyến tính đóng một vai trò rất quan trọng trong các hệ thống truyền tin, chúng ta cũng sẽ xem xét hàm tự tương quan và phổ công suất của một quá trình ngẫu nhiên được lọc tuyến tính. Phần cuối cùng của chương này giải quyết vấn đề các đặc tính của các quá trình ngẫu nhiên thông thấp và thông dải.

### 2.2 TẠO CÁC BIẾN NGẪU NHIÊN

Các bộ tạo số ngẫu nhiên thường được sử dụng trong thực tế để mô phỏng tác động của các tín hiệu kiểu nhiễu và các hiện tượng ngẫu nhiên khác thường gặp phải trong thế giới vật lý. Tạp nhiễu như thế tồn tại trong các thiết bị và hệ thống điện tử và thường hạn chế khả năng truyền tin trên các cự ly lớn và tách các tín hiệu tương đối yếu của chúng ta. Bằng cách tạo ra tạp nhiễu như thế trên một máy tính, chúng ta có thể cứu xét các tác động của nó thông qua việc mô phỏng các hệ thống truyền tin và đánh giá

chất lượng hoạt động của những hệ thống như vậy trong điều kiện có tạp nhiễu.

Hầu hết các thư viện phần mềm máy tính đều chứa một bộ tạo số ngẫu nhiên phân bố đều. Một bộ tạo số ngẫu nhiên như thế tạo ra một con số nằm giữa 0 và 1 với xác suất như nhau. Chúng ta gọi tín hiệu lõi ra của một bộ tạo số ngẫu nhiên là một biến ngẫu nhiên. Nếu  $A$  ký hiệu cho một biến ngẫu nhiên như vậy thì giải giá trị của nó là khoảng  $0 \leq A \leq 1$ .

Chúng ta đều biết rằng sản phẩm số của một máy tính số có tính chính xác hạn chế, và hệ quả là không thể biểu diễn continuum<sup>\*)</sup> của các số trong khoảng  $0 \leq A \leq 1$ . Tuy nhiên, chúng ta có thể giả sử rằng máy tính của chúng ta biểu diễn mỗi một sản phẩm bằng một số lớn các bit ở dạng dấu phẩy tĩnh hoặc động. Hệ quả là đối với mọi mục đích thực tế, số các sản phẩm trong khoảng  $0 \leq A \leq 1$  đủ lớn đến độ chúng ta có thể yên lòng khi giả sử rằng bất kỳ giá trị nào trong khoảng này cũng là một sản phẩm có thể có từ bộ tạo số ngẫu nhiên.

Hàm mật độ xác suất đều đối với biến ngẫu nhiên  $A$ , được ký hiệu là  $f(A)$ , được minh họa trên hình 2.1(a). Chúng ta chú ý rằng giá trị trung bình hay giá trị kỳ vọng của  $A$ , ký hiệu là  $m_A$  là  $m_A = 1/2$ . Tích phân của hàm mật độ xác suất, mà nó biểu thị diện tích miền nằm dưới  $f(A)$ , được gọi là hàm phân bố xác suất của biến ngẫu nhiên  $A$  và được xác định theo

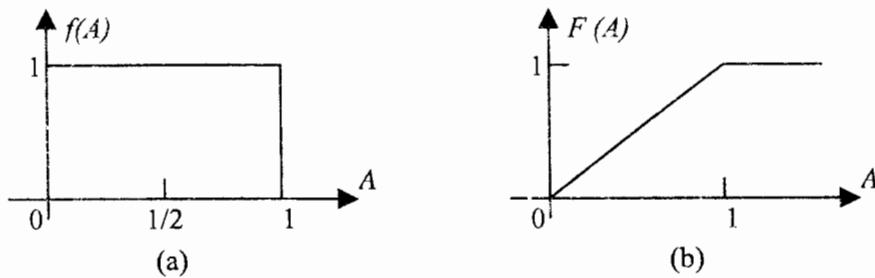
$$F(A) = \int_{-\infty}^A f(x)dx \quad (2.2.1)$$

Đối với biến ngẫu nhiên bất kỳ, diện tích miền nằm dưới hàm mật độ xác suất luôn là 1, mà nó chính là giá trị cực đại có thể đạt được bởi một hàm phân bố. Do vậy, đối với biến ngẫu nhiên phân bố đều  $A$  chúng ta có

$$F(1) = \int_{-\infty}^1 f(x)dx = 1 \quad (2.2.2)$$

và giải giá trị của  $F(A)$  là  $0 \leq F(A) \leq 1$  với  $0 \leq A \leq 1$ . Hàm phân bố xác suất này được thể hiện trên hình 2.1(b).

<sup>\*) Continuum:</sup> Tên gọi lực lượng của tập hợp  $L$  các điểm trong đoạn thẳng  $0 \leq x \leq 1$ . (Từ điển toán học. Nxb. Khoa học và kỹ thuật, Hà nội, 1977. ND).



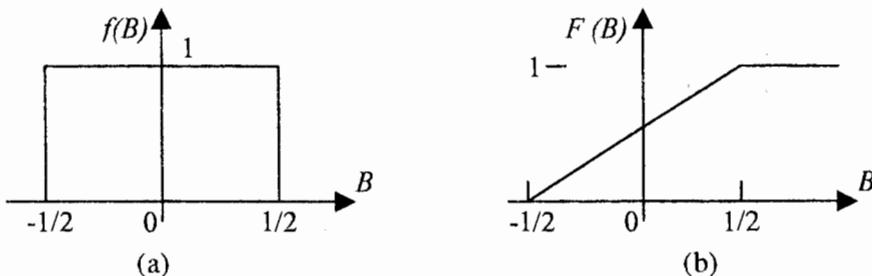
Hình 2.1 Hàm mật độ xác suất  $f(A)$  và hàm phân bố xác suất  $F(A)$  của biến ngẫu nhiên  $A$  phân bố đều.

Nếu chúng ta muốn tạo một tệp âm trong một khoảng  $(b, b+1)$ , điều đó có thể thực hiện được một cách đơn giản nhờ sử dụng số lõi ra  $A$  của một bộ tạo số ngẫu nhiên và dịch nó đi một lượng bằng  $b$ . Như thế, một biến ngẫu nhiên mới  $B$  có thể xác định được theo

$$B = A + b \quad (2.2.3)$$

mà nó bây giờ có giá trị kỳ vọng là  $m_B = b + \frac{1}{2}$ . Thí dụ, nếu  $b = -1/2$ , biến ngẫu

nhiên  $B$  sẽ là biến phân bố đều trên khoảng  $(-1/2, 1/2)$  như được thể hiện trên hình 2.2(a). Hàm phân bố xác suất  $F(B)$  của nó được thể hiện trên hình 2.2(b).



Hình 2.2 Hàm mật độ xác suất và hàm phân bố xác suất của một biến ngẫu nhiên phân bố đều có kỳ vọng bằng không.

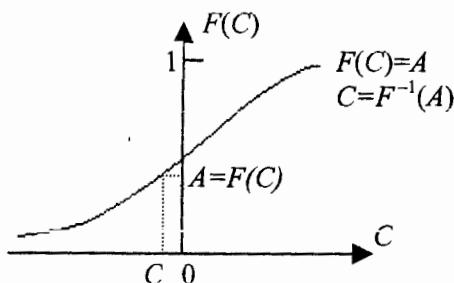
Một biến ngẫu nhiên phân bố đều trên giải  $(0, 1)$  có thể được sử dụng để tạo ra các biến ngẫu nhiên với các hàm phân bố xác suất khác. Thí dụ, giả sử rằng chúng ta muốn tạo một biến ngẫu nhiên  $C$  với hàm phân bố xác suất  $F(C)$  như được minh họa trên hình 2.3. Do giải của  $F(C)$  là khoảng  $(0, 1)$ , chúng ta bắt đầu bằng việc tạo ra một biến ngẫu nhiên  $A$  phân bố đều trong khoảng  $(0, 1)$ . Nếu chúng ta đặt

$$F(C)=A \quad (2.2.4)$$

thì

$$C=F^{-1}(A) \quad (2.2.5)$$

Như vậy, chúng ta giải (2.2.4) để có được  $C$  và lời giải của (2.2.5) sẽ cho chúng ta giá trị của  $C$  mà với nó  $F(C)=A$ . Bằng cách này, chúng ta nhận được một biến ngẫu nhiên mới  $C$  với hàm phân bố xác suất  $F(C)$ . Ánh xạ ngược từ  $A$  thành  $C$  này được minh họa trên hình 2.3.

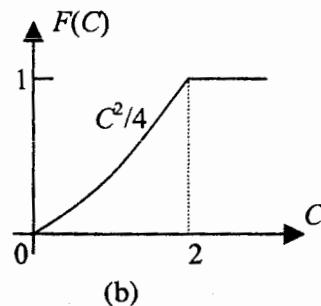
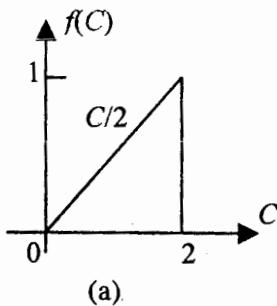


Hình 2.3 ánh xạ ngược từ một biến ngẫu nhiên phân bố đều thành một biến ngẫu nhiên mới  $C$ .

### BÀI TẬP MINH HỌA

**Bài tập minh họa 2.1** hãy tạo một biến ngẫu nhiên  $C$  có hàm mật độ xác suất tuyến tính như được thể hiện trên hình vẽ 2.4(a); nghĩa là

$$f(C) = \begin{cases} \frac{1}{2}C, & 0 \leq C \leq 2 \\ 0, & \text{Với mọi giá trị } C \text{ khác} \end{cases}$$



Hình 2.4 Hàm mật độ xác suất tuyến tính và hàm phân bố xác suất tương ứng.

**Lời giải**

Biến ngẫu nhiên này có hàm phân bố xác suất

$$F(C) = \begin{cases} 0, & C < 0 \\ \frac{1}{4}C^2, & 0 \leq C \leq 2 \\ 1, & C > 2 \end{cases}$$

nó như được minh họa trên hình 2.4(b). Chúng ta tạo một biến ngẫu nhiên phân bố đều  $A$  và đặt  $F(C)=A$ . Do đó,

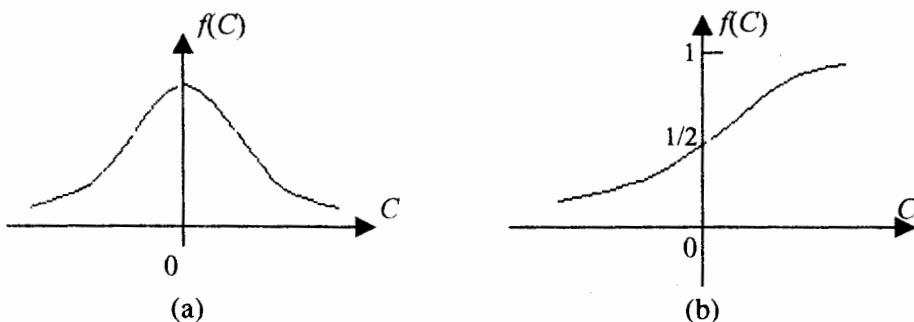
$$F(C) = \frac{1}{4}C^2 = A \quad (2.2.6)$$

Giải tìm  $C$ , chúng ta được

$$C = 2\sqrt{A} \quad (2.2.7)$$

Như thế, chúng ta tạo ra một biến ngẫu nhiên  $C$  có hàm phân bố xác suất  $F(C)$  như được minh họa trên hình 2.4(b).

Trong bài tập minh họa 2.1, ánh xạ ngược  $C=F^{-1}(A)$  là đơn giản. Trong một số trường hợp thì ánh xạ ngược này lại không đơn giản. Vấn đề này nảy sinh khi cố gắng tạo các biến ngẫu nhiên có hàm phân bố chuẩn (Gauss).



Hình 2.5 Hàm mật độ xác suất Gauss và hàm phân bố xác suất tương ứng.

Tập âm gấp phải trong các hệ thống vật lý thường được đặc tả bằng phân bố xác suất chuẩn (Gauss) được minh họa trên hình 2.5. Hàm mật độ xác suất được cho bởi

$$f(C) = \frac{1}{\sqrt{2\pi}\sigma} e^{-C^2/2\sigma^2}, \quad -\infty < C < \infty \quad (2.2.8)$$

trong đó  $\sigma^2$  là phương sai của  $C$ , là độ đo sự doãng rộng của hàm mật độ xác suất  $f(C)$ . Hàm phân bố xác suất  $F(C)$  là diện tích miền nằm dưới  $f(C)$  trên giải  $(-\infty, C)$ . Như vậy,

$$F(C) = \int_{-\infty}^C f(x)dx \quad (2.2.9)$$

Chẳng may là tích phân trong (2.2.9) không thể biểu diễn được bằng các hàm đơn giản. Hệ quả là ánh xạ ngược rất khó mà đạt được. Một cách thức đã được tìm ra nhằm phá vỡ khó khăn này. Từ lý thuyết xác suất người ta đã biết được rằng một biến ngẫu nhiên phân bố Rayleigh  $R$  có hàm phân bố xác suất

$$F(R) = \begin{cases} 0, & R < 0 \\ 1 - e^{-R^2/2\sigma^2}, & R \geq 0 \end{cases} \quad (2.2.10)$$

liên hệ được với một cặp biến ngẫu nhiên Gauss  $C$  và  $D$  thông qua biến đổi

$$C=R\cos\Theta \quad (2.2.11)$$

$$D=R\sin\Theta \quad (2.2.12)$$

trong đó  $\Theta$  là một biến ngẫu nhiên phân bố đều trong khoảng  $(0, 2\pi)$ . Tham số  $\sigma^2$  là phương sai của  $C$  và  $D$ . Do (2.2.10) lấy hàm ngược được dễ dàng, chúng ta có

$$F(R) = 1 - e^{-R^2/2\sigma^2} = A \quad (2.2.13)$$

và do vậy

$$R = \sqrt{2\sigma^2 \ln\left(\frac{1}{1-A}\right)} \quad (2.2.14)$$

trong đó  $A$  là một biến ngẫu nhiên phân bố đều trong khoảng  $(0, 1)$ . Nay giờ, nếu chúng ta tạo ra một biến ngẫu nhiên phân bố đều trong khoảng  $(0, 1)$  thứ hai là  $B$  và xác định

$$\Theta = 2\pi B \quad (2.2.15)$$

thì từ (2.2.11) và (2.2.12) chúng ta nhận được hai biến ngẫu nhiên phân bố chuẩn độc lập thống kê  $C$  và  $D$ .

Phương pháp đã mô tả trên đây thường được sử dụng trong thực tế để tạo các

biến ngẫu nhiên phân bố chuẩn. Như thấy được trên hình 2.5, các biến ngẫu nhiên này có kỳ vọng bằng 0 và phương sai bằng  $\sigma^2$ . Nếu muốn có một biến ngẫu nhiên chuẩn kỳ vọng khác không, khi đó  $C$  và  $D$  có thể tính tiến được bằng cách cộng thêm giá trị kỳ vọng.

MATLAB script thực hiện phương pháp nói trên nhằm tạo các biến ngẫu nhiên phân bố chuẩn được cho dưới đây.

### m-file

```
function [gsrv1,gsrv2]=gngauss(m,sgma)
% /gsrv1,gsrv2=gngauss(m,sgma)
% /gsrv1,gsrv2=gngauss(sgma)
% /gsrv1,gsrv2=gngauss
% GNGAUSS generates two independent Gaussian random variables with mean
% m and standard deviation sgma. If one of the input arguments is missing, it
% takes the mean as 0.
% If neither mean nor the variance is given, it generates two standard
% Gaussian random variables.
if nargin == 0,
    m=0; sgma=1;
elseif nargin == 1,
    sgma=m; m=0;
end;
u=rand; % a uniform random variable in (0,1)
z=sgma*(sqrt(2*log(1/(1-u)))); % a Rayleigh distributed random variable
u=rand; % another uniform random variable in (0,1)
gsrv1=m+z*cos(2*pi*u);
gsrv2=m+z*sin(2*pi*u);
```

## 2.3 CÁC QUÁ TRÌNH NGẪU NHIÊN GAUSS VÀ GAUSS-MARKOV

Các quá trình ngẫu nhiên Gauss đóng một vai trò quan trọng trong các hệ thống truyền tin. Lý do căn bản đối với tầm quan trọng này là ở chõ tệp âm nhiệt trong các thiết bị điện tử, mà nó được sinh ra bởi các chuyển động ngẫu nhiên của các điện tử dưới tác động nhiệt, có thể mô hình hoá rất sát bằng một quá trình ngẫu nhiên Gauss. Lý do đối với đặc tính Gauss của tệp âm nhiệt là ở chõ dòng điện tạo ra từ chuyển động của các điện tử trong một mạch điện tử có thể xem

như là tổng của các dòng nhỏ của một số rất lớn các nguồn-là từng điện tử riêng lẻ. Có thể giả sử được rằng chí ít là đa số các nguồn này chạy một cách độc lập và do vậy dòng điện tổng cộng là tổng của một số lớn các biến ngẫu nhiên độc lập có phân bố như nhau (i.i.d.: *independent and identically distributed*). Theo định lý giới hạn trung tâm, dòng tổng cộng này có một phân bố Gauss.

Ngoài tạp âm nhiệt, các quá trình ngẫu nhiên Gauss còn cho những mô hình khá tốt đối với cả một số nguồn thông tin. Một số đặc tính lý thú của các quá trình ngẫu nhiên Gauss được trình bày dưới đây làm cho các quá trình này trở nên dễ xử lý một cách toán học và dễ dàng có thể giả quyết được. Chúng ta sẽ mở đầu bằng một định nghĩa có tính hình thức về một quá trình Gauss.

**Định nghĩa:** Một quá trình ngẫu nhiên  $X(t)$  là một quá trình Gauss nếu với mọi  $n$  và mọi  $(t_1, t_2, \dots, t_n)$ , các biến ngẫu nhiên  $\{X(t_i)\}_{i=1}^n$  có một hàm mật độ Gauss cùng nhau, có thể biểu diễn được theo

$$f(x) = \frac{1}{(2\pi)^{n/2} [\det(C)]^{1/2}} \exp\left[-\frac{1}{2}(x - m)' C^{-1}(x - m)\right] \quad (2.3.1)$$

trong đó véc-tơ  $x = (x_1, x_2, \dots, x_n)'$  ký hiệu cho  $n$  biến ngẫu nhiên  $x_i \equiv X(t_i)$ ,  $m$  là véc-tơ giá trị kỳ vọng; nghĩa là  $m = E(X)$ , còn  $C$  là ma trận hiệp phương sai (*covariance matrix*)  $n \times n$  của các biến ngẫu nhiên  $(x_1, x_2, \dots, x_n)$  với các phần tử

$$c_{ij} = E[(x_i - m_i)(x_j - m_j)] \quad (2.3.2)$$

Chỉ số trên  $t$  ký hiệu cho chuyển vị của một véc-tơ hay ma trận, còn  $C^{-1}$  là nghịch đảo của ma trận hiệp phương sai  $C$ .

Từ định nghĩa này có thể thấy riêng được rằng tại bất kỳ một thời điểm  $t_0$  nào biến ngẫu nhiên  $X(t_0)$  cũng là một biến Gauss và tại hai điểm  $t_1, t_2$  bất kỳ nào thì các biến ngẫu nhiên  $(X(t_1), X(t_2))$  cũng được phân bố theo một biến ngẫu nhiên Gauss hai chiều. Hơn thế, do một mô tả thống kê đầy đủ của  $\{X(t_i)\}_{i=1}^n$  chỉ phụ thuộc vào véc-tơ kỳ vọng và ma trận hiệp phương sai  $C$ , chúng ta có được các tính chất sau.

**Tính chất 1:** Đối với các quá trình ngẫu nhiên Gauss, việc biết được véc-tơ kỳ vọng  $m$

và ma trận hiệp phương sai  $C$  cho chúng ta một mô tả thống kê đầy đủ về quá trình này.

Một tính chất rất quan trọng nữa của một quá trình Gauss liên quan tới các đặc trưng của nó khi đi qua một hệ thống LTI. Tính chất này có thể phát biểu được như sau.

**Tính chất 2:** Nếu quá trình Gauss  $X(t)$  được cho qua một hệ thống LTI thì tín hiệu lõi ra của hệ thống này cũng là một quá trình Gauss. Tác động của hệ thống tới  $X(t)$  chỉ được đáp ứng một cách đơn giản bằng một sự thay đổi về giá trị kỳ vọng và hiệp phương sai của  $X(t)$ .

### BÀI TẬP MINH HỌA

**Bài tập minh họa 2.2 [Tạo các mẫu của một quá trình Gauss nhiều biến]** Hãy tạo các mẫu của một quá trình ngẫu nhiên Gauss nhiều biến  $X(t)$  có một giá trị kỳ vọng xác định  $\mathbf{m}_x$  và một hiệp phương sai xác định  $\mathbf{C}_x$ .

#### Lời giải

Trước hết chúng ta hãy tạo một dãy gồm  $n$  biến ngẫu nhiên Gauss độc lập thống kê có kỳ vọng bằng không và phương sai bằng 1 bằng cách sử dụng phương pháp đã được mô tả trong mục 2.2. Cúng ta hãy ký hiệu dãy  $n$  biến này bằng véc-tơ  $\mathbf{Y}=(y_1, y_2, \dots, y_n)'$ . Thứ đến, chúng ta hãy phân tích thành thừa số ma trận hiệp phương sai  $n \times n$  mong muốn  $\mathbf{C}_x$  thành

$$\mathbf{C}_x = \mathbf{C}_x^{1/2} (\mathbf{C}_x^{1/2})' \quad (2.3.3)$$

Vậy, chúng ta xác định véc-tơ ( $n \times 1$ )  $X$  được biến đổi tuyến tính theo

$$\mathbf{X} = \mathbf{C}_x^{1/2} \mathbf{Y} + \mathbf{m}_x \quad (2.3.4)$$

Như thế, hiệp phương sai của  $X$  là

$$\begin{aligned} \mathbf{C}_x &= E[(\mathbf{X} - \mathbf{m}_x)(\mathbf{X} - \mathbf{m}_x)'] = E[\mathbf{Y} \mathbf{Y}' (\mathbf{C}_x^{1/2})'] \\ &= \mathbf{C}_x^{1/2} E(\mathbf{Y} \mathbf{Y}') (\mathbf{C}_x^{1/2})' = \mathbf{C}_x^{1/2} (\mathbf{C}_x^{1/2})' \end{aligned} \quad (2.3.5)$$

Bước khó khăn nhất trong quá trình này là việc phân tích ma trận hiệp phương sai  $\mathbf{C}_x$  thành thừa số. Chúng ta hãy diễn tả thủ tục này bằng một thí dụ áp dụng phân bố Gauss hai biến. Giả sử chúng ta bắt đầu với một cặp biến ngẫu

nhiên Gauss độc lập  $y_1$  và  $y_2$  có kỳ vọng bằng không và phương sai bằng 1. Ta muốn biến đổi chúng thành một cặp biến ngẫu nhiên Gauss  $x_1$  và  $x_2$  có kỳ vọng  $m=0$  và ma trận hiệp phương sai

$$C = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 1 \end{bmatrix} \quad (2.3.6)$$

trong đó  $\sigma_1^2$  và  $\sigma_2^2$  tương ứng là các phương sai của  $x_1$  và  $x_2$  còn  $\rho$  là hiệp phương sai chuẩn hoá được xác định theo

$$\rho = \frac{E[(X_1 - m_1)(X_2 - m_2)]}{\sigma_1\sigma_2} = \frac{c_{12}}{\sigma_1\sigma_2} \quad (2.3.7)$$

Ma trận hiệp phương sai  $C$  có thể phân tích được thành

$$C = C^{1/2}(C^{1/2})'$$

trong đó

$$C^{1/2} = \frac{1}{2\sqrt{2}} \begin{bmatrix} \sqrt{3} + 1 & \sqrt{3} - 1 \\ \sqrt{3} - 1 & \sqrt{3} + 1 \end{bmatrix} \quad (2.3.8)$$

Do vậy,

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = C^{1/2} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \frac{1}{2\sqrt{2}} \begin{bmatrix} \sqrt{3} + 1 & \sqrt{3} - 1 \\ \sqrt{3} - 1 & \sqrt{3} + 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \frac{1}{2\sqrt{2}} \begin{bmatrix} (\sqrt{3} + 1)y_1 + (\sqrt{3} - 1)y_2 \\ (\sqrt{3} - 1)y_1 + (\sqrt{3} + 1)y_2 \end{bmatrix} \quad (2.3.9)$$

Các MATLAB script đối với tính toán này được cho dưới đây  
**m-file**

```
% MATLAB script for Illustrative Problem 2, Chapter 2.
echo on
mx=[0 0]';
Cx=[1 1/2;1/2 1];
x=multi_gp(mx,Cx);
% Computation of the pdf of (x1,x2) follows
delta=0.3;
x1=-3:delta:3;
x2=-3:delta:3;
```

```

for i=1:length(x1),
    for j=1:length(x2),
        f(i,j)=(1/((2*pi)*det(Cx)^1/2))*exp((-1/2)*(([x1(i) x2(j)]-
mx')*inv(Cx)*([x1(i);x2(j)]-mx)));
    echo off ;
end;
echo on ;
% plotting command for pdf follows
mesh(x1,x2,f);

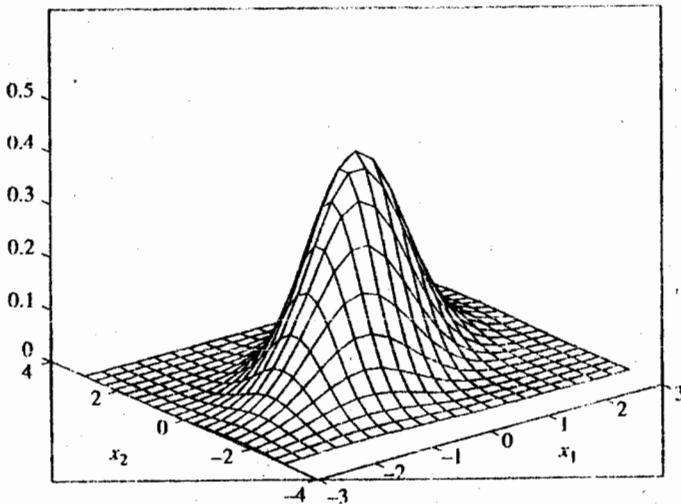
```

### m-file

```

function [x] = multi_gp(m,C)
% [x]=multi_gp(m,C)
%      MULTI_GP generates a multivariate Gaussian random
%      process with mean vector m (column vector), and covariance matrix C.
N=length(m);
for i=1:N,
    y(i)=gngauss;
end;
y=y.}';
x=sqrtn(C)*y+m;

```



Hình 2.6 Hàm mật độ xác suất cùng nhau của  $x_1$  và  $x_2$ .

Hình 2.6 minh họa hàm mật độ xác suất cùng nhau (*joint pdf*) đối với ma trận hiệp phương sai  $C$  được cho trong (2.3.6).

Như đã nêu, bước khó nhất trong tính toán là xác định  $\mathbf{C}^{1/2}$ . Với ma trận hiệp phương sai mong muốn đã cho, chúng ta có thể xác định các giá trị riêng  $\{\lambda_k, 1 \leq k \leq n\}$  và các véc-tơ riêng tương ứng  $\{v_k, 1 \leq k \leq n\}$ . Vậy thì ma trận hiệp phương sai  $\mathbf{C}$  có thể biểu diễn được theo

$$\mathbf{C} = \sum_{k=1}^n \lambda_k v_k v_k' \quad (2.3.10)$$

và do  $\mathbf{C} = \mathbf{C}^{1/2}(\mathbf{C}^{1/2})'$ , chúng ta có

$$\mathbf{C}^{1/2} = \sum_{k=1}^n \lambda_k^{1/2} v_k v_k' \quad (2.3.11)$$

**Định nghĩa:** Một quá trình Markov  $X(t)$  là một quá trình ngẫu nhiên mà quá khứ của nó không ảnh hưởng gì tới tương lai nếu hiện tại của nó được xác định; nghĩa là, nếu  $t_n > t_{n-1}$  thì

$$P[X(t_n) \leq x_n | X(t), t \leq t_{n-1}] = P[X(t_n) \leq x_n | X(t_{n-1})] \quad (2.3.12)$$

Định nghĩa này dẫn đến nếu  $t_1 < t_2 < \dots < t_n$  thì

$$P[X(t_n) \leq x_n | X(t_{n-1}), X(t_{n-2}), \dots, X(t_1)] = P[X(t_n) \leq x_n | X(t_{n-1})] \quad (2.3.13)$$

**Định nghĩa:** Một quá trình Gauss-Markov  $X(t)$  là một quá trình Markov mà hàm mật độ xác suất của nó là Gauss.

Phương pháp đơn giản nhất để tạo một quá trình Markov là phương pháp sử dụng công thức lặp đơn giản

$$X_n = \rho X_{n-1} + w_n \quad (2.3.14)$$

khi  $w_n$  là một dãy các biến ngẫu nhiên (trắng) độc lập cùng phân bố (i.i.d) có kỳ vọng bằng không còn  $\rho$  là một tham số xác định bậc của tương quan giữa  $X_n$  và  $X_{n-1}$ ; nghĩa là

$$E(X_n X_{n-1}) = \rho E(X_{n-1}^2) = \rho \sigma_{n-1}^2 \quad (2.3.15)$$

Nếu dãy  $\{w_n\}$  có phân bố Gauss thì quá trình  $X(t)$  có được sẽ là một quá trình Gauss-Markov.

## BÀI TẬP MINH HỌA

**Bài tập minh họa 2.3 [Quá trình Gauss-Markov]** Hãy tạo một dãy (phân cách

đều nhau) gồm 1000 mẫu của một quá trình Gauss-Markov từ công thức lặp

$$X_n = 0.95 X_{n-1} + w_n, \quad n=1, 2, \dots, 1000 \quad (2.3.16)$$

trong đó  $X_0=0$  còn  $\{w_n\}$  là một dãy các biến ngẫu nhiên Gauss *i.i.d* có kỳ vọng không và phương sai bằng 1. Hãy vẽ đồ thị dãy  $\{X_n, 1 \leq n \leq 1000\}$  như là một hàm của chỉ số thời gian  $n$  và tự tương quan

$$R_x(m) = \frac{1}{N-m} \sum_{n=1}^{N-m} X_n X_{n+m}, \quad m = 0, 1, \dots, 50 \quad (2.3.17)$$

trong đó  $N=1000$ .

### Lời giải

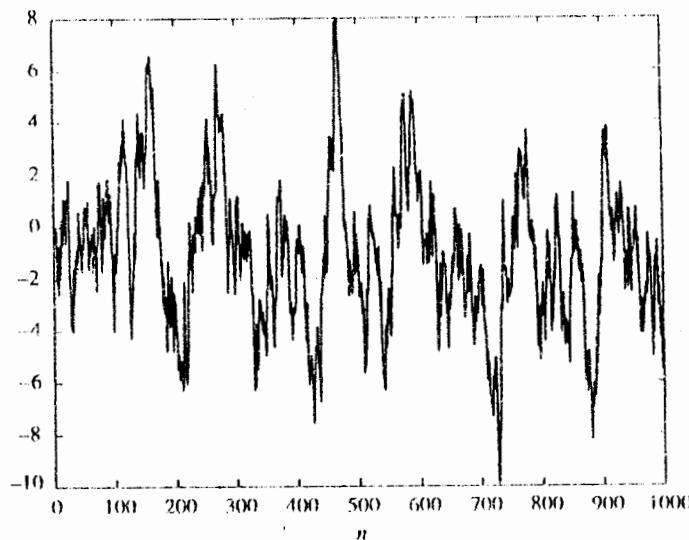
Các MATLAB *script* đối với việc tính toán này được cho dưới đây. Các hình 2.7 và 2.8 lần lượt minh họa dãy  $(X_n)$  và hàm tự tương quan  $\hat{R}_x(m)$ .

### m-file

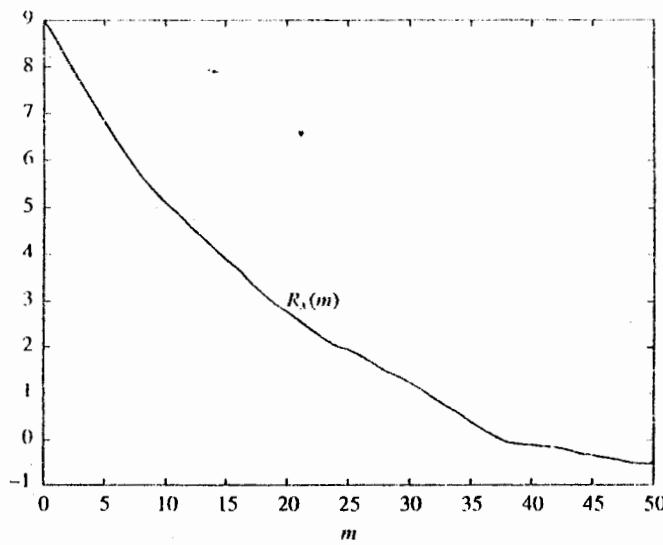
```
% MATLAB script for Illustrative Problem 3, Chapter 2.
echo on
rho=0.95;
X0=0;
N=1000;
X=gaus_mar(X0,rho,N);
M=50;
Rx=Rx_est(X,M);
% plotting commands follow
```

### m-file

```
function [X]=gaus_mar(X0,rho,N)
% [X]=gaus_mar(X0,rho,N)
%
% GAUS_MAR generates a Gauss-Markov process of length N.
% The noise process is taken to be white Gaussian
% noise with zero mean and unit variance.
for i=1:2:N,
    [Ws(i) Ws(i+1)]=gngauss; % Generate the noise process
end;
X(1)=rho*X0+Ws(1); % First element in the Gauss-Markov process
for i=2:N,
    X(i)=rho*X(i-1)+Ws(i); % the remaining elements
end;
```



Hình 2.7 Dãy Gauss-Markov.



Hình 2.8 Hàm tự tương quan của quá trình Gauss-Markov.

## 2.4 PHỔ CÔNG SUẤT CỦA CÁC QUÁ TRÌNH NGẪU NHIÊN VÀ CÁC QUÁ TRÌNH TRẮNG

Một quá trình ngẫu nhiên dừng  $X(t)$  được đặc tả trên miền tần số bằng phổ công suất của nó  $S_x(f)$  là biến đổi Fourier của hàm tự tương quan  $R_x(\tau)$  của quá trình đó;

nghĩa là

$$S_x(f) = \int_{-\infty}^{\infty} R_x(\tau) e^{-j2\pi f\tau} d\tau \quad (2.4.1)$$

Ngược lại, hàm tự tương quan  $R_x(\tau)$  của một quá trình ngẫu nhiên dừng  $X(t)$  nhận được từ phổ công suất  $S_x(f)$  bằng biến đổi Fourier ngược; nghĩa là

$$R_x(\tau) = \int_{-\infty}^{\infty} S_x(f) e^{j2\pi f\tau} df \quad (2.4.2)$$

Khi mô hình hoá tạp âm nhiệt phát sinh ra trong các thiết bị điện tử được sử dụng trong việc thực hiện các hệ thống truyền tin, chúng ta thường giả sử rằng tạp âm như thế là một quá trình ngẫu nhiên trắng. Một quá trình ngẫu nhiên như thế được định nghĩa như sau.

**Định nghĩa:** Một quá trình ngẫu nhiên  $X(t)$  được gọi là một quá trình trắng nếu nó có một phổ công suất bằng phẳng - nghĩa là nếu  $S_x(f)$  là một hằng số với mọi  $f$ .

Như đã nêu, tầm quan trọng của các quá trình trắng xuất phát từ sự thực là tạp âm nhiệt có thể mô hình hoá được một cách sát thực bằng một hằng số về phổ trên một dải tần số rộng. Một vài quá trình - được sử dụng để mô tả một số các nguồn thông tin - cũng mô hình hoá được như sản phẩm đầu ra của các hệ thống LTI có đầu vào là một quá trình trắng.

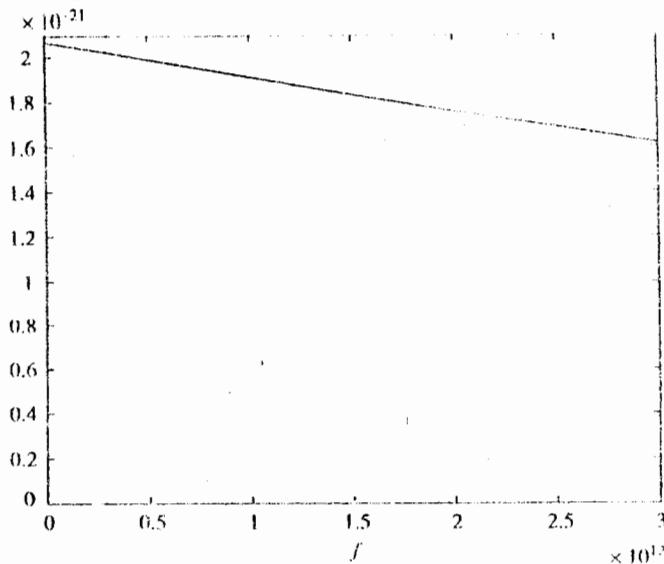
Tuy nhiên, chúng ta thấy được rằng nếu  $S_x(f)=C$  đối với mọi  $f$  thì

$$\int_{-\infty}^{\infty} S_x(f) df = \int_{-\infty}^{\infty} C df = \infty \quad (2.4.3)$$

cho nên công suất tổng cộng bằng vô cùng. Hiển nhiên là chẳng có một quá trình vật lý thực nào lại có thể có công suất vô hạn và do đó một quá trình trắng không thể là một quá trình vật lý có ý nghĩa nào. Dầu sao, việc phân tích cơ chế lượng tử của một tạp âm nhiệt cho thấy rằng nó có một mật độ phổ công suất cho được theo

$$S_n(f) = \frac{hf}{2(e^{hf/kT} - 1)} \quad (2.4.4)$$

trong đó  $h$  ký hiệu cho hằng số Planck (bằng  $6.6 \times 10^{-34}$  Js) còn  $k$  là hằng số Boltzmann (bằng  $1.38 \times 10^{-23}$  J/K).  $T$  ký hiệu cho nhiệt độ Kelvin. Phổ công suất này được thể hiện trên hình 2.9.



Hình 2.9 Đồ thị của  $S_n(f)$  trong công thức (2.4.4).

Phổ này đạt giá trị lớn nhất của nó tại  $f=0$ , và giá trị cực đại này là  $kT/2$ . Phổ này tiến đến không khi  $f$  tiến đến vô cùng song tốc độ hội tụ tới không rất chậm. Thí dụ, tại nhiệt độ phòng ( $T=300$  K),  $S_n(f)$  hạ thấp xuống còn 90% giá trị cực đại của nó tại quãng  $f \approx 2.10^{12}$  Hz mà tần số này nằm trên các tần số được sử dụng trong các hệ thống truyền tin truyền thống. Từ điều này chúng ta có thể kết luận được rằng tạp âm nhiệt, mặc dù không tráng một cách chính xác, vẫn có thể mô hình hóa được đối với mọi mục đích thực tế như là một quá trình tráng với phổ công suất bằng  $kT/2$ . Giá trị  $kT/2$  thường được ký hiệu là  $N_0$ ; do vậy mật độ phổ công suất của tạp âm nhiệt thường được cho là  $S_n(f)=N_0/2$  và đôi khi nó cũng được gọi với tên là *mật độ phổ công suất hai phía* nhằm nhấn mạnh rằng phổ này trải trên cả các tần số dương lẫn âm. Chúng ta từ đây trở đi sẽ tránh dùng thuật ngữ đó mà chỉ sử dụng một cách đơn giản là *phổ công suất* hay *mật độ phổ công suất*.

Đối với một quá trình ngẫu nhiên tráng  $X(t)$  có phổ công suất  $S_x(f)=N_0/2$ , hàm tự

tương quan  $R_x(\tau)$  là

$$R_x(\tau) = \int_{-\infty}^{\infty} S_x(f) e^{j2\pi f\tau} df = \frac{N_0}{2} \int_{-\infty}^{\infty} e^{j2\pi f\tau} df = \frac{N_0}{2} \delta(\tau) \quad (2.4.5)$$

trong đó  $\delta(\tau)$  là hàm xung đơn vị. Hệ quả là đối với mọi  $\tau \neq 0$ , chúng ta có  $R_x(\tau)=0$ ; nghĩa là nếu chúng ta lấy mẫu một quá trình ngẫu nhiên trắng tại hai điểm  $t_1$  và  $t_2$  ( $t_1 \neq t_2$ ) thì các biến ngẫu nhiên được tạo ra không tương quan với nhau. Ngoài việc là một quá trình trắng, nếu quá trình ngẫu nhiên còn là quá trình Gauss thì các biến ngẫu nhiên có được do lấy mẫu sẽ là các biến ngẫu nhiên Gauss độc lập thống kê.

### BÀI TẬP MINH HỌA

**Bài tập minh họa 2.4 [Hàm tự tương quan và phổ công suất]** Hãy tạo một dãy rời rạc theo thời gian gồm  $N=1000$  số ngẫu nhiên phân bố đều *i.i.d.* trong khoảng  $(-1/2, 1/2)$  và tính giá trị tự tương quan của dãy  $\{X_n\}$ , được định nghĩa theo

$$\begin{aligned} R_x(m) &= \frac{1}{N-m} \sum_{n=1}^{N-m} X_n X_{n+m}, & m &= 0, 1, \dots, M \\ &= \frac{1}{N-|m|} \sum_{n=|m|}^N X_n X_{n+m}, & m &= -1, -2, \dots, -M \end{aligned} \quad (2.4.6)$$

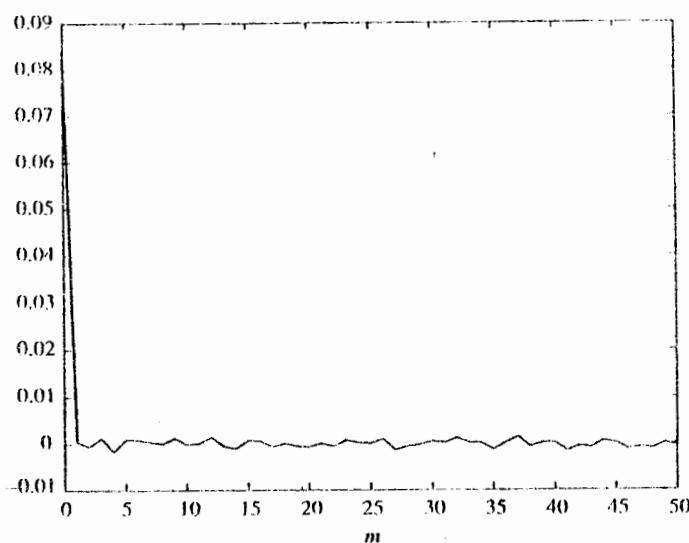
Cũng hãy xác định phổ công suất của dãy  $\{X_n\}$  bằng cách tính biến đổi Fourier rời rạc (DFT: *Discrete Fourier Transform*) của DFT này, được tính toán một cách hiệu quả nhờ sử dụng thuật toán FFT, được xác định theo

$$S_x(f) = \sum_{m=-M}^M R_x(m) e^{-j2\pi fm/(2M+1)} \quad (2.4.7)$$

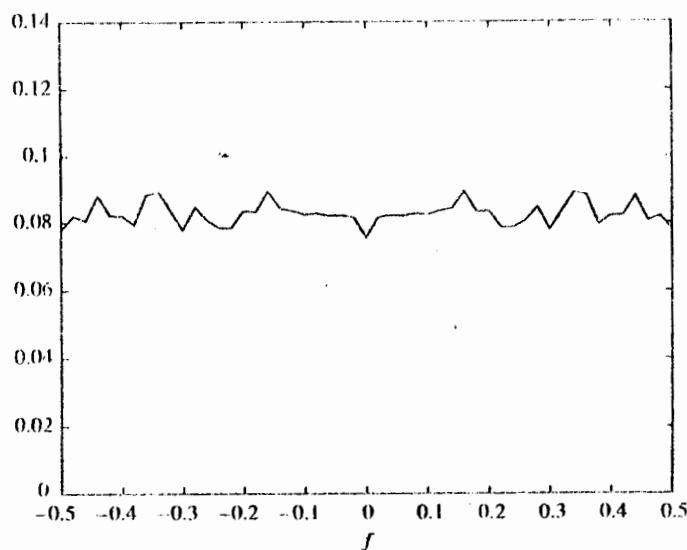
### Lời giải

MATLAB *script* thực hiện việc tạo dãy  $\{X_n\}$ , tính hàm tự tương quan và tính phổ công suất  $S_x(f)$  được cho dưới đây. Chúng ta nên để ý rằng hàm tự tương quan và phổ công suất thể hiện một sự biến động đáng kể. Vì vậy, cần phải lấy trung bình giá trị tự tương quan mẫu trên một số lần thực hiện. Các hình 2.10 và 2.11 minh họa  $R_x(m)$  và  $S_x(f)$  nhận được bằng cách chạy chương trình này và áp dụng lấy

giá trị tự tương quan trung bình qua 10 lần thực hiện quá trình ngẫu nhiên.



Hình 2.10 Hàm tự tương quan trong bài tập minh họa 2.4.



Hình 2.11 Phổ công suất trong bài tập minh họa 2.4.

### m-file

```
% MATLAB script for Illustrative Problem 4, Chapter 2.  
echo on  
N=1000;  
M=50;  
Rx_av=zeros(1,M+1);
```

```

Sx_av=zeros(1,M+1);
for j=1:10, % take the ensemble average over 10 realizations
    X=rand(1,N)-1/2; % N i.i.d. uniformly distributed random variables
    % between -1/2 and 1/2
    Rx=Rx_est(X,M); % Autocorrelation of the realization
    Sx=fftshift(abs(fft(Rx))); % Power spectrum of the realization
    Rx_av=Rx_av+Rx; % sum of the autocorrelations
    Sx_av=Sx_av+Sx; % sum of the spectrums
    echo off ;
end;
echo on ;
Rx_av=Rx_av/10; % ensemble average autocorrelation
Sx_av=Sx_av/10; % ensemble average spectrum
% Plotting comments follow

```

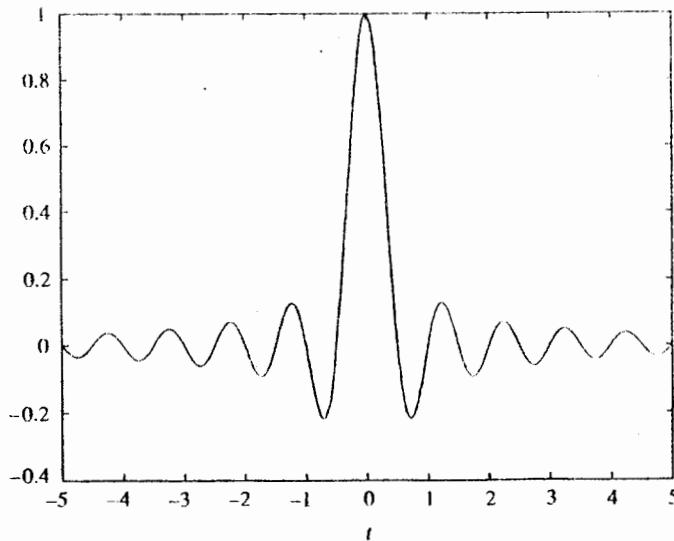
Một quá trình ngẫu nhiên với độ rộng băng hạn chế  $X(t)$  có phổ công suất

$$S_x(f) = \begin{cases} N_0 / 2 & |f| \leq B \\ 0, & |f| > B \end{cases} \quad (2.4.8)$$

Ta hãy xác định hàm tự tương quan của nó. Từ (2.4.1) chúng ta có

$$R_x(\tau) = \int_{-B}^B \frac{N_0}{2} e^{j2\pi f\tau} df = N_0 B \frac{\sin 2\pi B\tau}{2\pi B\tau} \quad (2.4.9)$$

Hình 2.12 minh họa  $R_x(\tau)$ .



Hình 2.12 Đồ thị hàm tự tương quan  $R_x(\tau)$  được cho bởi (2.4.9) với  $B=N_0=1$ .

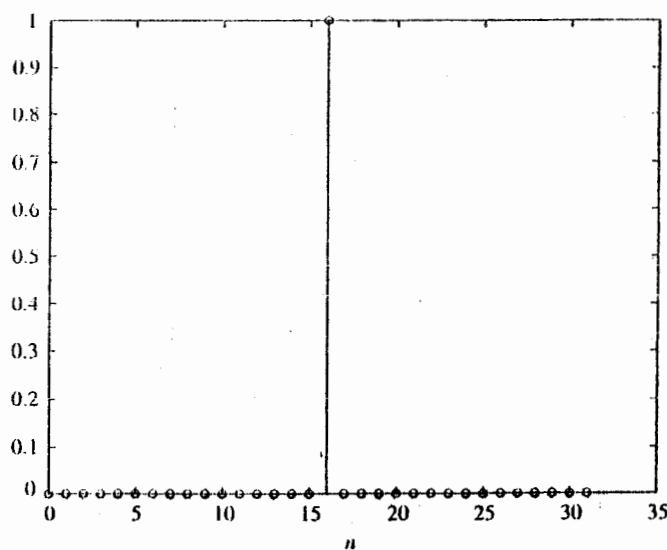
MATLAB có thể sử dụng được để tính  $R_x(\tau)$  từ  $S_x(f)$  và ngược lại. Thuật toán FFT có thể sử dụng được cho việc tính toán này.

### BÀI TẬP MINH HỌA

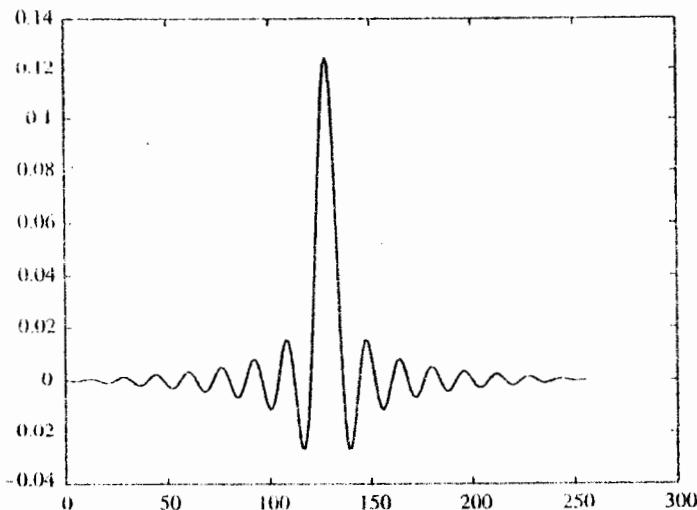
**Bài tập minh họa 2.5 [Hàm tự tương quan và phổ công suất]** Hãy tính hàm tự tương quan  $R_x(\tau)$  đối với quá trình ngẫu nhiên mà phổ công suất của nó được cho bởi (2.4.8).

#### Lời giải

Để thực hiện tính toán này, chúng ta biểu diễn  $S_x(f)$  bằng  $N$  mẫu trên dải tần số  $|f| \leq B$ , với mỗi một mẫu được chuẩn hoá về 1. Kết quả của việc tính toán FFT ngược với  $N=32$  được minh họa trên hình vẽ 2.13. Hãy chú ý rằng chúng ta chỉ nhận được một biểu diễn thô của hàm tự tương quan  $R_x(\tau)$  bởi vì chúng ta đã chỉ lấy mẫu  $S_x(f)$  trong dải tần số  $|f| \leq B$ . Phân cách tần số trong thí dụ này là  $\Delta f = 2B/N$ . Nếu chúng ta giữ nguyên  $\Delta f$  và tăng số mẫu bằng cách lấy cả các mẫu với  $|f| > B$ , chúng ta sẽ nhận được các giá trị trung gian của  $R_x(\tau)$ . Hình 2.14 minh họa kết quả tính toán FFT ngược với  $N_1=256$  mẫu trong đó  $N=32$  mẫu bằng 1.



Hình 2.13 FFT ngược của phổ công suất của quá trình ngẫu nhiên với băng hạn chế trong bài tập minh họa 2.5 với 32 mẫu.



Hình 2.14 FFT ngược của phô công suất của quá trình ngẫu nhiên với băng hạn chế trong bài tập minh họa 2.5 với 256 mẫu.

MATLAB *script* thực hiện các biến đổi Fourier ngược kể trên được cho dưới đây.

### m-file

```
% MATLAB script for Illustrative Problem 5, Chapter 2.
```

```
echo on
```

```
% first part
```

```
Sx1=[ones(1,32)];
```

```
Rx1=ifft(Sx1,32);
```

```
% second part
```

```
Sx2=[ones(1,16),zeros(1,224),ones(1,16)];
```

```
Rx2=ifft(Sx2,256);
```

```
% plotting commands follow
```

## 2.5 LỌC TUYẾN TÍNH CÁC QUÁ TRÌNH NGẪU NHIÊN

Giả sử rằng một quá trình ngẫu nhiên  $X(t)$  được cho qua một mạch lọc tuyến tính không biến đổi theo thời gian được đặc tả trên miền thời gian bằng hàm đáp ứng xung  $h(t)$  của nó và trên miền tần số bởi đáp ứng tần số của nó

$$H(f) = \int_{-\infty}^{\infty} h(t) e^{-j2\pi ft} dt \quad (2.5.1)$$

Do đó sản phẩm lối ra của bộ lọc tuyến tính là quá trình ngẫu nhiên

$$Y(t) = \int_{-\infty}^{\infty} X(\tau)h(t-\tau)d\tau \quad (2.5.2)$$

Giá trị trung bình của  $Y(t)$  là

$$m_y \equiv E[Y(t)] = \int_{-\infty}^{\infty} E[X(\tau)]h(t-\tau)d\tau = m_x \int_{-\infty}^{\infty} h(t-\tau)d\tau = m_x \int_{-\infty}^{\infty} h(\tau)d\tau = m_x H(0) \quad (2.5.3)$$

trong đó  $H(0)$  là đáp ứng tần số  $H(f)$  của bộ lọc được đánh giá tại  $f=0$ .

Hàm tự tương quan của  $Y(t)$  là

$$\begin{aligned} R_y(\tau) &= E[Y(\tau)Y(t+\tau)] \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E[X(\tau)X(\alpha)]h(t-\tau)h(t+\tau-\alpha)d\tau d\alpha \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R_x(\tau-\alpha)h(t-\tau)h(t+\tau-\alpha)d\tau d\alpha \end{aligned} \quad (2.5.4)$$

Trên miền tần số, phổ công suất của quá trình ngẫu nhiên lối ra  $Y(t)$  được liên hệ với phổ công suất của quá trình ngẫu nhiên lối vào  $X(t)$  và đáp ứng tần số của bộ lọc tuyến tính theo biểu thức

$$S_y(f) = S_x(f)|H(f)|^2 \quad (2.5.5)$$

Điều này dễ dàng thấy được nhờ lấy biến đổi Fourier của (2.5.4).

### BÀI TẬP MINH HỌA

**Bài tập minh họa 2.6 [Tập nhiễu được lọc]** Giả sử rằng một quá trình ngẫu nhiên trắng  $X(t)$  có phổ công suất  $S_x(f)=1$  với mọi  $f$  kích thích ở lối vào một bộ lọc tuyến tính có đáp ứng xung

$$h(t) = \begin{cases} e^{-t}, & t \geq 0 \\ 0, & t < 0 \end{cases} \quad (2.5.6)$$

Hãy xác định phổ công suất  $S_y(f)$  của sản phẩm lối ra của bộ lọc.

#### Lời giải

Đáp ứng tần số của bộ lọc này dễ dàng thấy được là

$$H(f) = \frac{1}{1 + j2\pi f} \quad (2.5.7)$$

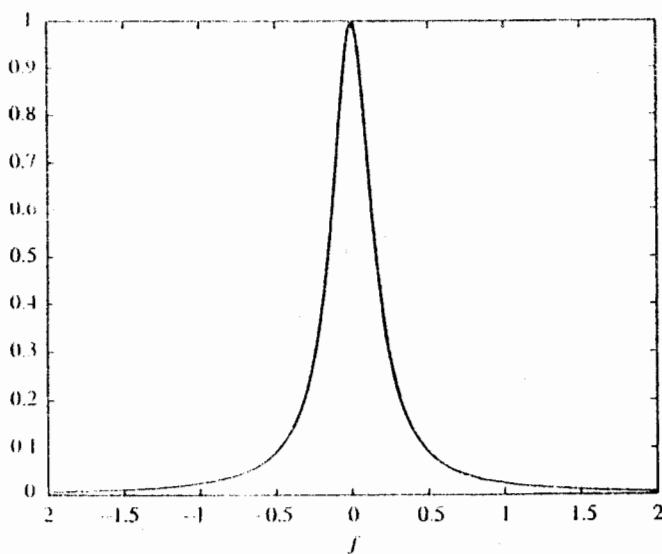
Do đó,  $S_y(f) = |H(f)|^2 = \frac{1}{1 + (2\pi f)^2}$  (2.5.8)

Đồ thị của  $S_y(f)$  được minh họa trên hình 2.15. MATLAB script để tính  $S_y(f)$  đối với  $S_x(f)$  và  $H(f)$  đã xác định được cho dưới đây.

### m-file

% MATLAB script for Illustrative Problem 6, Chapter 2.

```
echo on
delta=0.01;
F_min=-2;
F_max=2;
f=F_min:delta:F_max;
Sx=ones(1,length(f));
H=1./(1+(2*pi*f).^2);
Sy=Sx.*H.^2;
```



Hình 2.15 Đồ thị của  $S_y(f)$  được cho bởi (2.5.8)

### BÀI TẬP MINH HỌA

**Bài tập minh họa 2.7 [Hàm tự tương quan và phổ công suất]** Hãy tính hàm tự tương quan  $R_y(\tau)$  ứng với  $S_y(f)$  trong bài tập minh họa 2.6 đối với phổ công

suất đã định  $S_x(f)=1$ .

### Lời giải

Trong trường hợp này, chúng ta có thể sử dụng thuật toán FFT ngược trên các mẫu của  $S_y(f)$  đã cho bởi (2.5.8). Hình 2.16 minh họa tính toán này với  $N=256$  mẫu tần số và khoảng phân cách tần số  $\Delta f=0.1$ . MATLAB script cho tính toán này được cho dưới đây.

### m-file

```
% MATLAB script for Illustrative Problem 7, Chapter 2.
```

```
echo on
```

```
N=256;
```

*% number of samples*

```
deltaf=0.1;
```

*% frequency separation*

```
f=[0:deltaf:(N/2)*deltaf, -(N/2-1)*deltaf:deltaf:-deltaf];
```

*% swap the first half*

```
Sy=1./(1+(2*pi*f).^2);
```

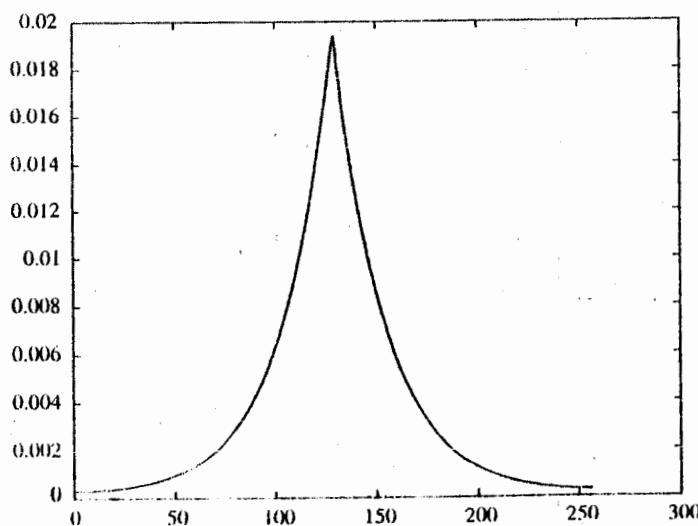
*% sampled spectrum*

```
Ry=ifft(Sy);
```

*% Autocorrelation of Y*

*% plotting command follows*

```
plot(fftshift(real(Ry)));
```



Hình 2.16 Đồ thị của  $R_y(f)$  trong bài tập minh họa 2.7.

Bây giờ chúng ta hãy xét vấn đề thời gian rời rạc tương đương. Giả sử rằng một quá trình ngẫu nhiên dừng  $X(t)$  được lấy mẫu và các mẫu đã lấy được cho qua một bộ lọc tuyến tính rời rạc về thời gian với hàm đáp ứng xung  $h(n)$ . Sản phẩm

lối ra của bộ lọc tuyến tính này được cho bởi công thức tổng chập

$$Y(n) = \sum_{k=0}^{\infty} h(k) X(n-k) \quad (2.5.9)$$

trong đó  $X(n) \equiv X(t_n)$  là các giá trị thời gian rời rạc của quá trình ngẫu nhiên lối vào còn  $Y(n)$  là sản phẩm lối ra của bộ lọc rời rạc về thời gian. Giá trị trung bình của quá trình lối ra là

$$m_y = E[Y(n)] = \sum_{k=0}^{\infty} h(k) E[X(n-k)] = m_x \sum_{k=0}^{\infty} h(k) = m_x H(0) \quad (2.5.10)$$

trong đó  $H(0)$  là đáp ứng tần số  $H(f)$  của bộ lọc được tính tại  $f=0$  và

$$H(f) = \sum_{n=0}^{\infty} h(n) e^{-j2\pi fn} \quad (2.5.11)$$

Hàm tự tương quan của quá trình lối ra là

$$\begin{aligned} R_y(m) &= E[Y(n)Y(n+m)] = \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} h(k)h(l) E[X(n-k)X(n+m-l)] \\ &= \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} h(k)h(l) R_x(m-l+k) \end{aligned} \quad (2.5.12)$$

Biểu thức tương ứng trên miền tần số là

$$S_y(f) = S_x(f) |H(f)|^2 \quad (2.5.13)$$

ở đây các phổ công suất được xác định theo

$$S_x(f) = \sum_{m=-\infty}^{\infty} R_x(m) e^{-j2\pi fm} \quad (2.5.14)$$

và

$$S_y(f) = \sum_{m=-\infty}^{\infty} R_y(m) e^{-j2\pi fm} \quad (2.5.15)$$

## BÀI TẬP MINH HỌA

**Bài tập minh họa 2.8 [Tạp âm trắng được lọc]** Giả sử rằng một quá trình ngẫu nhiên trắng với các mẫu  $\{X(n)\}$  được cho qua một mạch lọc tuyến tính có đáp ứng xung

$$h(n) = \begin{cases} (0.95)^n, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

Hãy xác định phổ công suất của quá trình lối ra  $\{Y(n)\}$

### Lời giải

Dễ dàng thấy được rằng

$$H(f) = \sum_{n=0}^{\infty} h(n)e^{-j2\pi fn} = \sum_{n=0}^{\infty} (0.95e^{-j2\pi f})^n = \frac{1}{1 - 0.95e^{-j2\pi f}} \quad (2.5.16)$$

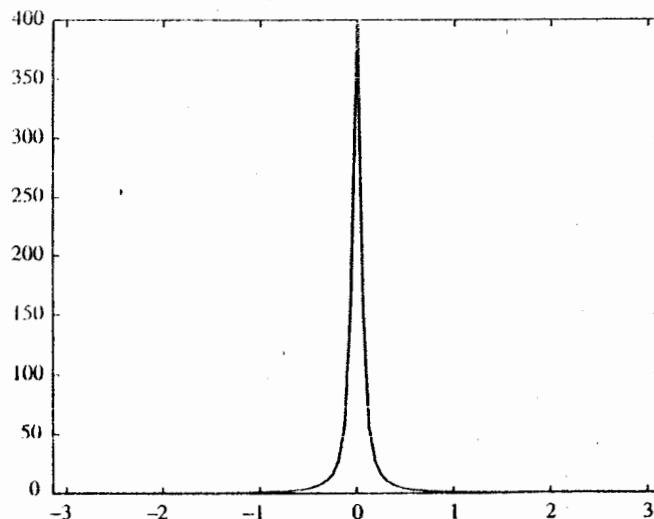
$$\text{và } |H(f)|^2 = \frac{1}{|1 - 0.95e^{-j2\pi f}|^2} = \frac{1}{1.9025 - 1.9 \cos(2\pi f)} \quad (2.5.17)$$

Do đó, phổ công suất của quá trình lối ra là

$$S_y(f) = |H(f)|^2 S_x(f) \quad (2.5.18)$$

$$= \frac{1}{1.9025 - 1.9 \cos(2\pi f)} \quad (2.5.19)$$

trong đó chúng ta đã giả thiết  $S_x(f)$  được chuẩn hoá thành 1. Hình 2.17 minh họa  $S_y(f)$ . Hãy để ý rằng  $S_y(f)$  tuân hoà với chu kỳ  $2\pi$ . MATLAB script cho tính toán này được cho dưới đây.



Hình 2.17 Đồ thị của  $S_y(f)$  trong bài tập minh họa 2.8.

### m-file

```
% MATLAB script for Illustrative Problem 8, Chapter 2.
delta_w=2*pi/100;
w=-pi:delta_w:pi; % one period of Sy
```

```
Sy=1./(1.9025-1.9*cos(w));
% plotting command follows
plot(w,Sy);
```

Hàm tự tương quan của quá trình lối ra  $\{Y(n)\}$  có thể xác định được bằng cách lấy FFT ngược của  $S_y(f)$ . Sinh viên sẽ thấy lý thú khi so sánh hàm tự tương quan ấy với hàm tự tương quan ~~này~~ được trong bài tập minh họa 2.3.

## 2.6 CÁC QUÁ TRÌNH THÔNG DÀI VÀ THÔNG THẤP

Cũng giống như trong trường hợp các tín hiệu tất định, các tín hiệu ngẫu nhiên cũng có thể được đặc tả theo các quá trình ngẫu nhiên thông dài và thông thấp.

**Định nghĩa:** Một quá trình ngẫu nhiên được gọi là thông thấp nếu phổ công suất của nó lớn quanh tần số  $f=0$  và nhỏ (tiến đến 0) tại các tần số cao. Nói một cách khác, một quá trình ngẫu nhiên thông thấp có hầu hết công suất của nó tập trung tại các tần số thấp.

**Định nghĩa:** Một quá trình ngẫu nhiên thông thấp  $X(t)$  là có băng hạn chế nếu phổ công suất  $S_x(f)=0$  với  $|f| > B$ . Tham số  $B$  được gọi là độ rộng băng của quá trình ngẫu nhiên đó.

### BÀI TẬP MINH HỌA

**Bài tập minh họa 2.9 [Các quá trình thông thấp]** Hãy xét vấn đề tạo các mẫu của một quá trình ngẫu nhiên thông thấp bằng cách cho một dãy tạp âm trắng  $\{X_n\}$  qua một bộ lọc thông thấp. Dãy lối vào là một dãy các biến ngẫu nhiên *i.i.d.* phân bố đều trong khoảng  $(-1/2, 1/2)$ . Bộ lọc thông thấp có đáp ứng xung

$$h(n) = \begin{cases} (0.9)^n, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

và được đặc tả bằng một phương trình (sai phân) lặp vào-ra

$$y_n = 0.9 y_{n-1} + x_n, \quad n \geq 1, \quad y_{-1} = 0$$

Hãy tính dãy lối ra  $\{y_n\}$  và hãy xác định các hàm tự tương quan  $R_x(m)$  và  $R_y(m)$  như đã được chỉ ra trong (2.4.6). Hãy xác định các phổ công suất  $S_x(f)$  và  $S_y(f)$

bằng cách tính DFT của  $R_x(m)$  và  $R_y(m)$ .

### Lời giải

Các MATLAB *script* cho các tính toán này được cho dưới đây. Các hình 2.18 và 2.19 minh họa các hàm tự tương quan và các phổ công suất. Chúng tôi lưu ý rằng các đồ thị của hàm tự tương quan và các phổ công suất là các trung bình qua mươi lần thực hiện quá trình ngẫu nhiên.

### m-file

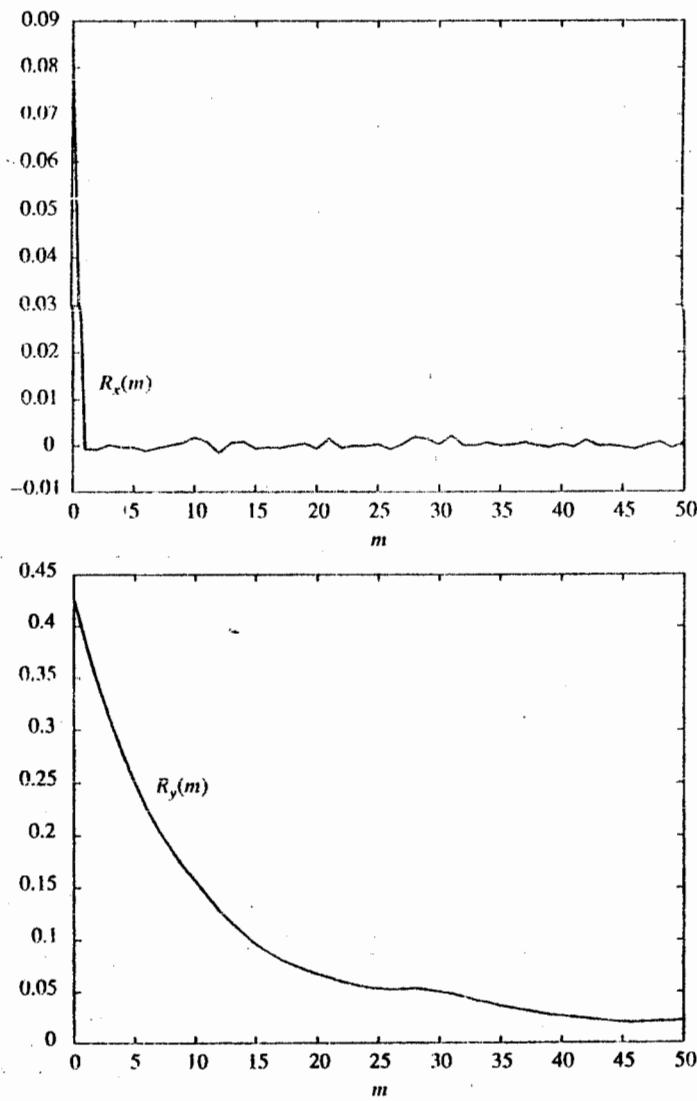
```
% MATLAB script for Illustrative Problem 9, Chapter 2.
N=1000; % The maximum value of n
M=50;
Rxav=zeros(1,M+1);
Ryav=zeros(1,M+1);
Sxav=zeros(1,M+1);
Syav=zeros(1,M+1);
for i=1:10, % take the ensemble average over 10 realizations
    X=rand(1,N)-(1/2); % Generate a uniform number sequence on (-1/2,1/2)
    Y(1)=0;
    for n=2:N, Y(n)=0.9*Y(n-1)+X(n); end; % note that Y(n) means Y(n-1)
    Rx=Rx_est(X,M); % Autocorrelation of {Xn}
    Ry=Rx_est(Y,M); % Autocorrelation of {Yn}
    Sx=fftshift(abs(fft(Rx))); % Power spectrum of {Xn}
    Sy=fftshift(abs(fft(Ry))); % Power spectrum of {Yn}
    Rxav=Rxav+Rx;
    Ryav=Ryav+Ry;
    Sxav=Sxav+Sx;
    Syav=Syav+Sy;
    echo off ;
end;
echo on ;
Rxav=Rxav/10;
Ryav=Ryav/10;
Sxav=Sxav/10;
Syav=Syav/10;
% Plotting commands follow
```

### m-file

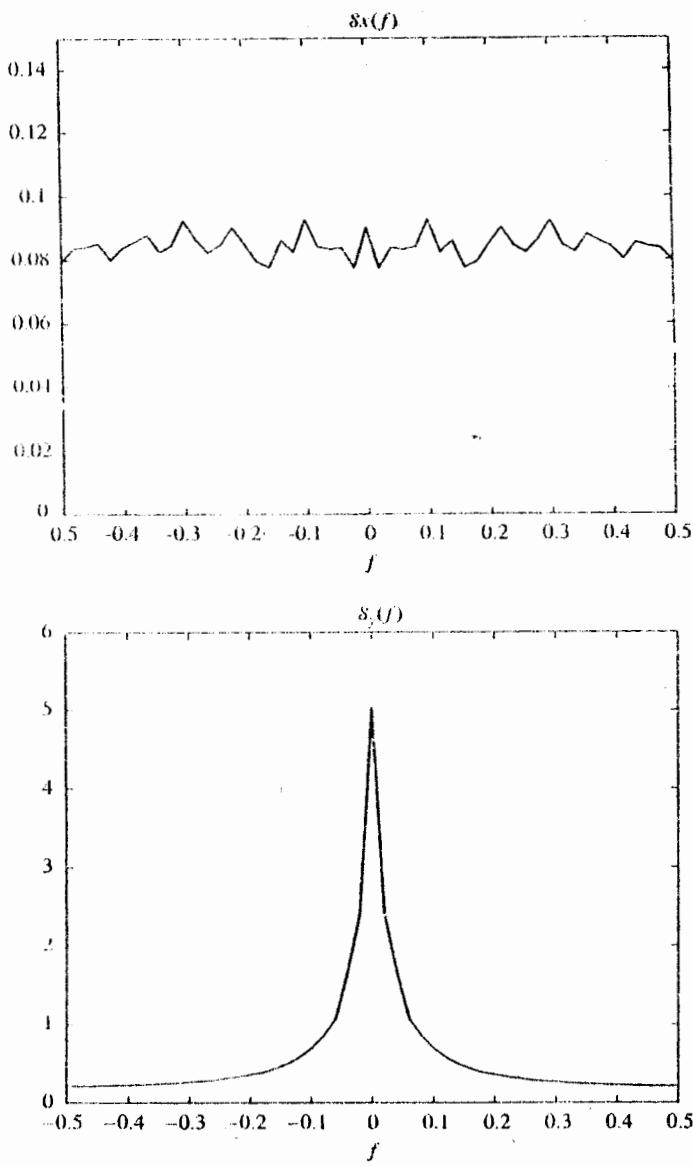
```
function [Rx]=Rx_est(X,M)
% [Rx]=Rx_est(X,M)
```

% *RX\_EST Estimates the autocorrelation of the sequence of random variables given in X. Only Rx(0), Rx(1), ..., Rx(M) are computed.*  
% *Note that Rx(m) actually means Rx(m-1).*

```
N=length(X);
Rx=zeros(1,M+1);
for m=1:M+1,
    for n=1:N-m+1,
        Rx(m)=Rx(m)+X(n)*X(n+m-1);
    end;
    Rx(m)=Rx(m)/(N-m+1);
end;
```



Hình 2.18 Các hàm tự tương quan  $R_x(m)$  và  $R_y(m)$  trong bài tập minh họa 2.9.



Hình 2.19 Các phổ công suất  $S_x(f)$  và  $S_y(f)$  trong bài tập minh họa 2.9.

**Định nghĩa:** Một quá trình ngẫu nhiên được gọi là thông dải nếu phổ công suất của nó lớn trong một băng tần số tập trung quanh một tần số trung tâm  $\pm f_0$  và khá bé bên ngoài băng tần số này. Một quá trình ngẫu nhiên được gọi là băng hẹp nếu độ rộng băng của nó  $B < < f_0$ .

Các quá trình thông dải thích hợp cho việc biểu diễn các tín hiệu đã được điều

chế. Trong một hệ thống truyền tin, tín hiệu mang thông tin thường là một quá trình ngẫu nhiên thông thấp mà tín hiệu này điều chế một sóng mang để truyền dẫn đi trên một kênh truyền tin thông dải (băng hẹp). Do đó, tín hiệu đã qua điều chế là một quá trình ngẫu nhiên thông dải.

Cũng như trong trường hợp các tín hiệu tất định, một quá trình ngẫu nhiên  $X(t)$  có thể biểu diễn được theo

$$X(t) = X_c(t)\cos 2\pi f_0 t - X_s(t)\sin 2\pi f_0 t \quad (2.6.1)$$

trong đó  $X_c(t)$  và  $X_s(t)$  được gọi là các thành phần cùng pha và vuông góc của  $X(t)$ . Các quá trình ngẫu nhiên  $X_c(t)$  và  $X_s(t)$  là các quá trình thông thấp. Định lý sau đây, được tuyên bố mà không chứng minh, cho chúng ta một mối quan hệ quan trọng giữa  $X(t)$ ,  $X_c(t)$  và  $X_s(t)$ .

**Định lý:** Nếu  $X(t)$  là một quá trình ngẫu nhiên dừng, kỳ vọng bằng không, thì các quá trình  $X_c(t)$  và  $X_s(t)$  cũng là các quá trình có kỳ vọng bằng không và dừng cùng nhau.

Thật vậy, có thể dễ dàng chứng minh được (xem [1]) rằng các hàm tự tương quan của  $X_c(t)$  và  $X_s(t)$  là như nhau và có thể biểu diễn được theo

$$R_c(\tau) = R_s(\tau) = R_x(\tau) \cos 2\pi f_0 \tau + \hat{R}_x(\tau) \sin 2\pi f_0 \tau \quad (2.6.2)$$

trong đó  $R_x(\tau)$  là hàm tự tương quan của quá trình thông dải  $X(t)$  còn  $\hat{R}_x(\tau)$  là biến đổi Hilbert của  $R_x(\tau)$  mà nó được xác định theo

$$\hat{R}_x(\tau) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{R(t)}{\tau - t} dt \quad (2.6.3)$$

Cũng vậy, hàm tương quan chéo của  $X_c(t)$  và  $X_s(t)$  được biểu diễn theo

$$R_{cs}(\tau) = R_x(\tau) \sin 2\pi f_0 \tau - \hat{R}_x(\tau) \cos 2\pi f_0 \tau \quad (2.6.4)$$

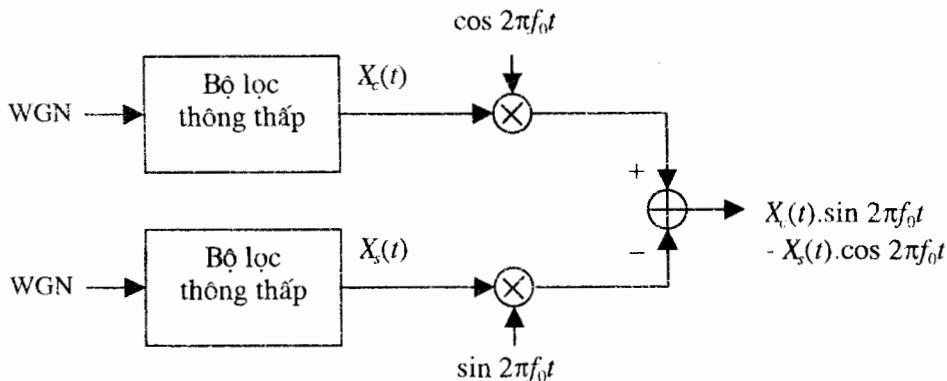
Cuối cùng, hàm tự tương quan của quá trình thông dải  $X(t)$  biểu diễn được thông qua hàm tự tương quan  $R_c(\tau)$  và hàm tương quan chéo  $R_{cs}(\tau)$  theo

$$R_x(\tau) = R_c(\tau) \cos 2\pi f_0 \tau - R_{cs}(\tau) \sin 2\pi f_0 \tau \quad (2.6.5)$$

## BÀI TẬP MINH HỌA

### Bài tập minh họa 2.10 [Tạo các mẫu của một quá trình ngẫu nhiên thông

**dải]** Hãy tạo các mẫu của một quá trình ngẫu nhiên thông dải bằng cách trước hết tạo các mẫu của hai quá trình ngẫu nhiên độc lập thống kê  $X_c(t)$  và  $X_s(t)$  rồi sau đó sử dụng các quá trình đó điều chế các sóng mang vuông pha nhau  $\cos 2\pi f_0 t$  và  $\sin 2\pi f_0 t$  như được thể hiện trên hình 2.20.



Hình 2.20 Việc tạo ra một quá trình ngẫu nhiên thông dải.

### Lời giải

Trên một máy tính số, các mẫu của các quá trình ngẫu nhiên thông thấp  $X_c(t)$  và  $X_s(t)$  được tạo ra nhờ lọc hai quá trình tạp nhiễu trắng độc lập bằng bộ lọc thông thấp như nhau. Như thế, chúng ta nhận được các mẫu  $X_c(n)$  và  $X_s(n)$  tương ứng với các giá trị lấy mẫu của và  $X_s(t)$ . Sau đó  $X_c(n)$  điều chế sóng mang được lấy mẫu  $\cos 2\pi f_0 nT$  còn  $X_s(n)$  thì điều chế sóng mang vuông góc  $\sin 2\pi f_0 nT$ , trong đó  $T$  là khoảng lấy mẫu thích hợp.

MATLAB *script* cho các tính toán này được cho dưới đây. Nhằm mục đích minh họa, chúng tôi đã chọn bộ lọc thông thấp có một hàm truyền như sau

$$H(z) = \frac{1}{1 - 0.9z^{-1}}$$

Cũng nhằm mục đích đó, chúng tôi đã chọn  $T=1$  và  $f_0=1000/\pi$ . Phổ công suất tính được của quá trình thông dải được thể hiện trên hình 2.21.

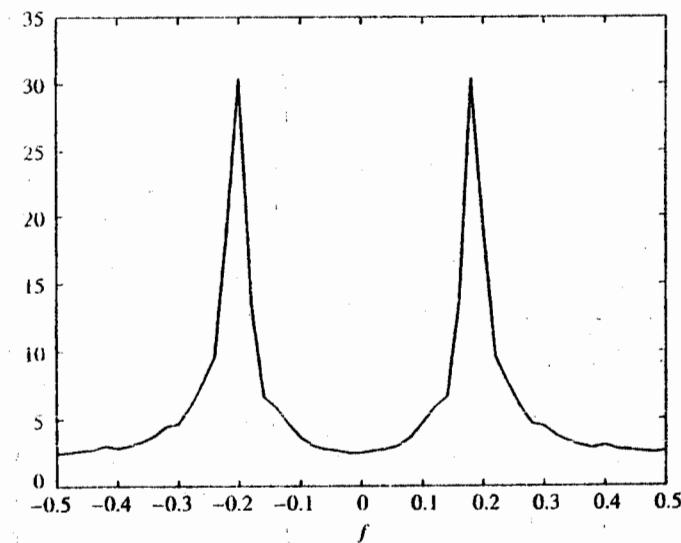
### m-file

```
% MATLAB script for Illustrative Problem 10, Chapter 2.
N=1000; % number of samples
for i=1:2:N,
```

```

[X1(i) X1(i+1)]=gngauss;
[X2(i) X2(i+1)]=gngauss;
echo off ;
end;                                % standard Gaussian input noise processes
echo on ;
A=[1 -0.9];                           % lowpass filter parameters
B=1;
Xc=filter(B,A,X1);
Xs=filter(B,A,X2);
fc=1000/pi;                            % carrier frequency
for i=1:N,
    band_pass_process(i)=Xc(i)*cos(2*pi*fc*i)-Xs(i)*sin(2*pi*fc*i);
    echo off ;
end;                                  % T=1 is assumed
echo on;
% Determine the autocorrelation and the spectrum of the band-pass process
M=50;
bpp_autocorr=Rx_est(band_pass_process,M);
bpp_spectrum=fftshift(abs(fft(bpp_autocorr)));
% plotting commands follow

```



Hình 2.21 Phổ công suất của quá trình thông dải trong bài tập minh họa 2.10.

## CÁC BÀI TẬP

**2.1** Hãy tạo một tập gồm 1000 số ngẫu nhiên phân bố đều trong khoảng [0, 1] bằng

cách sử dụng hàm MATLAB **rand(1, N)**. Hãy vẽ biểu đồ (cột) tần suất (*histogram*) và hàm phân bố xác suất cho dãy này. Biểu đồ tần suất có thể xác định được bằng cách chia khoảng nói trên thành mười khoảng con có độ rộng như nhau, lấp đầy dải  $[0, 1]$  và đếm các số rơi vào mỗi khoảng.

**2.2** Hãy tạo một tập gồm 1000 số ngẫu nhiên phân bố đều trong khoảng  $[-\frac{1}{2}, \frac{1}{2}]$  bằng cách sử dụng hàm MATLAB **rand(1, N)**. Hãy vẽ biểu đồ tần suất và hàm phân bố xác suất cho dãy này.

**2.3** Hãy tạo một tập gồm 1000 số ngẫu nhiên phân bố đều trong khoảng  $[-2, 2]$  bằng cách sử dụng hàm MATLAB **rand(1, N)**. Hãy vẽ biểu đồ tần suất và hàm phân bố xác suất cho dãy này.

**2.4** Hãy tạo một tập gồm 1000 số ngẫu nhiên có hàm mật độ xác suất tuyến tính

$$f(x) = \begin{cases} \frac{x}{2}, & 0 \leq x \leq 2 \\ 0, & x \notin [0, 2] \end{cases}$$

Hãy vẽ đồ thị biểu đồ tần suất và hàm phân bố xác suất.

**2.5** Hãy tạo một tập gồm 1000 số ngẫu nhiên Gauss có kỳ vọng bằng không và phương sai bằng 1 theo phương pháp đã được mô tả trong mục 2.2. Hãy vẽ biểu đồ tần suất và hàm phân bố xác suất cho dãy này. Trong việc xác định biểu đồ tần suất, dải của các số ngẫu nhiên có thể được chia thành các khoảng con có độ rộng  $\sigma^2/5$ , bắt đầu với khoảng thứ nhất phủ cả dải  $-\sigma^2/10 < x < \sigma^2/10$ , trong đó  $\sigma^2$  là phương sai.

**2.6** Hãy tạo một tập gồm 1000 số ngẫu nhiên Gauss có kỳ vọng bằng không và phương sai bằng 1 bằng cách sử dụng hàm MATLAB **randn(1, N)**. Hãy vẽ biểu đồ tần suất và hàm phân bố xác suất cho dãy này. Hãy so sánh các kết quả này với các kết quả nhận được trong bài tập 2.5.

**2.7** Hãy tạo 1000 cặp số ngẫu nhiên Gauss  $(x_1, x_2)$  mà chúng có véc-tơ kỳ vọng

$$\mathbf{m} = E[x_1 \ x_2] = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

và ma trận hiệp phương sai

$$C = \begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 1 \end{bmatrix}$$

a. Hãy xác định các trị trung bình của các mẫu ( $x_{1i}, x_{2i}$ ),  $i=1, 2, \dots, 1000$ , được định nghĩa theo

$$\hat{m}_1 = \frac{1}{1000} \sum_{i=1}^{1000} x_{1i}, \quad \hat{m}_2 = \frac{1}{1000} \sum_{i=1}^{1000} x_{2i}$$

Cũng hãy xác định các phương sai của chúng

$$\hat{\sigma}_1^2 = \frac{1}{1000} \sum_{i=1}^{1000} (x_{1i} - \hat{m}_1)^2, \quad \hat{\sigma}_2^2 = \frac{1}{1000} \sum_{i=1}^{1000} (x_{2i} - \hat{m}_2)^2$$

và hiệp phương sai của chúng

$$\hat{c}_{ij} = \frac{1}{1000} \sum_{i=1}^{1000} (x_{1i} - \hat{m}_1)(x_{2i} - \hat{m}_2)$$

b. Hãy so sánh các giá trị tính được từ các mẫu với các giá trị lý thuyết.

**2.8** Hãy tạo một dãy gồm 1000 mẫu của một quá trình Gauss-Markov được mô tả bởi quan hệ lặp

$$X_n = \rho X_{n-1} + W_n, \quad n=1, 2, \dots, 1000$$

trong đó  $X_0=0$ ,  $\rho=0.9$  còn  $\{W_n\}$  là một dãy các biến ngẫu nhiên Gauss *i.i.d.* có kỳ vọng bằng không và phương sai bằng 1.

**2.9** Lặp lại bài tập minh họa 2.4 với một dãy *i.i.d.* các biến ngẫu nhiên Gauss có kỳ vọng bằng không và phương sai bằng 1.

**2.10** Lặp lại bài tập minh họa 2.5 khi phổ công suất của một quá trình ngẫu nhiên có băng hạn chế là

$$S_x(f) = \begin{cases} 1 - \frac{|f|}{B}, & |f| \leq B \\ 0, & \text{với mọi giá trị } f \text{ khác} \end{cases}$$

**2.11** Lặp lại bài tập minh họa 2.6 với một bộ lọc tuyến tính có đáp ứng xung

$$h(t) = \begin{cases} e^{-3t}, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

**2.12** Hãy xác định bằng phương pháp tính số hàm tự tương quan của quá trình ngẫu nhiên tại lối ra của bộ lọc tuyến tính trong bài tập 2.11.

**2.13** Lặp lại bài tập minh họa 2.8 khi

$$h(n) = \begin{cases} (0.8)^n, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

**2.14** Hãy tạo một dãy *i.i.d.*  $\{x_n\}$  gồm  $N=1000$  số ngẫu nhiên phân bố đều trong khoảng  $[-\frac{1}{2}, \frac{1}{2}]$ . Dãy này được cho qua một bộ lọc tuyến tính có đáp ứng xung

$$h(n) = \begin{cases} (0.95)^n, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

Phương trình lặp mô tả lối ra của bộ lọc này như một hàm của lối vào là

$$y_n = 0.95y_{n-1} + x_n, \quad n \geq 0, \quad y_{-1} = 0.$$

Hãy tính các hàm tự tương quan  $R_x(m)$  và  $R_y(m)$  của các dãy  $\{x_n\}$  và  $\{y_n\}$  và các phổ công suất tương ứng  $S_x(f)$  và  $S_y(f)$  bằng cách sử dụng các quan hệ đã cho trong (2.4.6) và (2.4.7). Hãy so sánh kết quả đối với  $S_y(f)$  này với kết quả tương ứng nhận được trong bài tập minh họa 2.8.

**2.15** Hãy tạo hai dãy *i.i.d.*  $\{w_{cn}\}$  và  $\{w_{sn}\}$  gồm 1000 số ngẫu nhiên phân bố đều trong khoảng  $[-\frac{1}{2}, \frac{1}{2}]$ . Mỗi một dãy trong chúng đều được cho qua một mạch lọc tuyến tính có đáp ứng xung

$$h(n) = \begin{cases} (0.5)^n, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

mà đặc tính vào-ra của nó được cho bởi quan hệ lặp

$$x_n = \frac{1}{2}x_{n-1} + w_n, \quad n \geq 1, \quad x_0 = 0.$$

Như thế, chúng ta nhận được hai dãy,  $\{x_{cn}\}$  và  $\{x_{sn}\}$ . Dãy lối ra  $\{x_{cn}\}$  điều chế sóng mang  $\cos(\pi/2)n$ , còn dãy lối ra  $\{x_{sn}\}$  điều chế sóng mang vuông góc

$\sin(\pi/2)n$ . Tín hiệu thông dải được hình thành bằng cách kết hợp các thành phần đã được điều chế như trong (2.6.1).

Hãy tính và vẽ đồ thị các hàm tự tương quan thành phần  $R_c(m)$  và  $R_s(m)$  đối với  $|m| \leq 10$  tương ứng cho các dãy  $\{x_{cn}\}$  và  $\{x_{sn}\}$ . Hãy tính hàm tự tương quan  $R_x(m)$  của tín hiệu thông dải đối với  $|m| \leq 10$ . Hãy sử dụng thuật toán DFT (hoặc thuật toán FFT) để tính các phổ công suất  $S_c(f)$ ,  $S_s(f)$  và  $S_x(f)$ . Hãy vẽ các phổ công suất này và bình luận về các kết quả.

## CHƯƠNG 3

# ĐIỀU CHẾ TƯƠNG TỰ

### 3.1 GIỚI THIỆU CHUNG

Trong chương này, chúng ta nghiên cứu đặc tính của một loạt sơ đồ điều chế và giải điều chế tương tự, cả trong trường hợp có lẫn không có tạp nhiễu cộng. Các hệ thống được nghiên cứu trong chương này bao gồm các sơ đồ điều chế biên độ (AM: *Amplitude Modulation*) như thể điều chế biên độ hai biên (DSB-AM: *Double-SideBand-*), điều chế đơn biên (SSB-AM: *Single-SideBand-Amplitude Modulation*) và điều chế biên độ truyền thống (AM), và các sơ đồ điều chế góc như điều chế tần số và điều chế pha. Mỗi một thành viên của lớp các sơ đồ điều chế tương tự đều được đặc trưng bởi năm tính chất cơ bản:

1. Biểu diễn trên miền thời gian của tín hiệu đã điều chế
2. Biểu diễn trên miền tần số của tín hiệu đã điều chế
3. Độ rộng băng của tín hiệu đã điều chế
4. Công suất của tín hiệu đã điều chế
5. Tỷ số tín hiệu trên tạp âm (SNR: *Signal-to-Noise Ratio*) sau giải điều chế.

Các tính chất này hiển nhiên là không độc lập với nhau. Tồn tại một quan hệ chặt chẽ giữa các biểu diễn miền tần số và miền thời gian của các tín hiệu thể hiện thông qua quan hệ biến đổi Fourier. Cũng như vậy, độ rộng băng của một tín hiệu được xác định thông qua các đặc trưng tần số của nó.

Do sự khác biệt căn bản giữa các sơ đồ điều chế góc và điều chế biên độ nên các sơ đồ này được đề cập đến một cách riêng rẽ và trong các mục khác nhau.

Chúng ta bắt đầu chương này bằng việc nghiên cứu sơ đồ điều chế đơn giản nhất là điều chế biên độ.

### 3.2 ĐIỀU CHẾ BIÊN ĐỘ (AM)

Điều chế biên độ (AM), thường được xem là điều chế tuyến tính, là họ các sơ đồ điều chế trong đó biên độ của một sóng mang hình sin bị biến đổi như một hàm của tín hiệu điều chế. Lớp các sơ đồ điều chế này gồm điều chế hai biên (DSB-AM), điều chế biên độ truyền thống, điều chế đơn biên (SSB-AM) và điều chế vết đơn biên (VSB-AM: *Vestigial-SideBand – Amplitude Modulation*). Sự phụ thuộc giữa tín hiệu điều chế và biên độ của sóng mang đã được điều chế có thể là rất đơn giản, thí dụ như trong trường hợp DSB-AM, hoặc phức tạp hơn nhiều, như trong SSB-AM hay VSB-AM. Các hệ thống điều chế biên độ thường được đặc trưng bởi một yêu cầu khá thấp về độ rộng băng và tính không hiệu quả về công suất so với các sơ đồ điều chế góc. Yêu cầu độ rộng băng đối với các hệ thống AM xê xích giữa  $W$  và  $2W$ , trong đó  $W$  ký hiệu cho độ rộng băng của tín hiệu bản tin. Đối với SSB-AM độ rộng băng là  $W$ , đối với DSB-AM và AM truyền thống độ rộng băng là  $2W$  còn đối với VSB-AM thì độ rộng băng nằm giữa  $W$  và  $2W$ . Các hệ thống này được sử dụng rộng rãi trong phát quảng bá (phát thanh AM và phát thị tần trong truyền hình), trong liên lạc điểm tới điểm (SSB) và trong các ứng dụng ghép kênh (thí dụ như trong truyền dẫn nhiều kênh điện thoại trên các kênh vi ba).

#### 3.2.1 DSB-AM

Trong DSB-AM, biên độ của tín hiệu đã điều chế tỷ lệ thuận với tín hiệu bản tin. Điều đó có nghĩa là biểu diễn trên miền thời gian của tín hiệu sau điều chế được cho bởi

$$u(t) = A_c m(t) \cos(2\pi f_c t) \quad (3.2.1)$$

trong đó

$$c(t) = A_c \cos(2\pi f_c t) \quad (3.2.2)$$

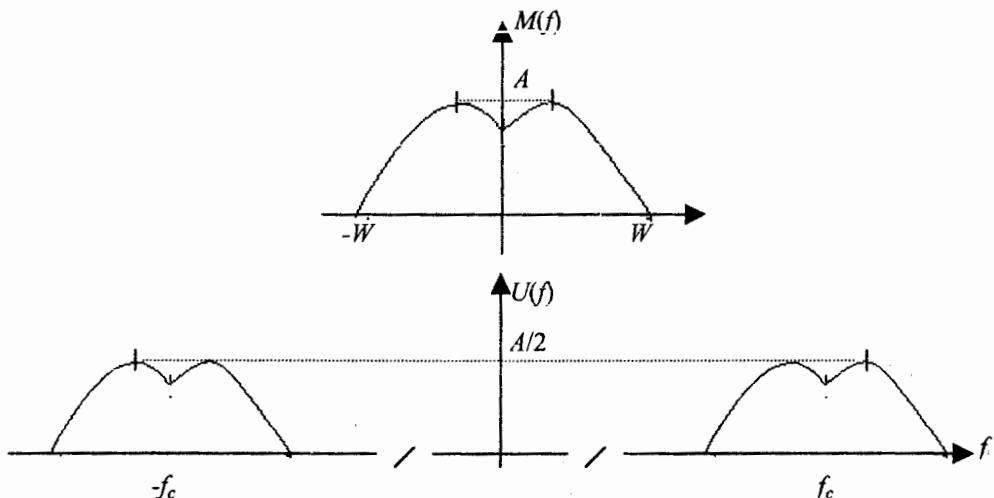
là sóng mang, còn  $m(t)$  là tín hiệu bản tin. Biểu diễn trên miền tần số của tín hiệu DSB-AM nhận được bằng cách lấy biến đổi Fourier của  $u(t)$  và dẫn đến

$$U(f) = \frac{A_c}{2} M(f - f_c) + \frac{A_c}{2} M(f + f_c) \quad (3.2.3)$$

trong đó  $M(f)$  là biến đổi Fourier của  $m(t)$ . Hiển nhiên kiểu điều chế này dẫn đến một sự dịch chuyển  $\pm f_c$  và một sự đổi tỷ lệ  $A_c/2$  đối với phổ của tín hiệu bản tin. Độ rộng băng truyền dẫn được ký hiệu bởi  $B_T$  bằng hai lần độ rộng băng của tín hiệu bản tin

$$B_T = 2W \quad (3.2.4)$$

Một phổ tín hiệu bản tin tiêu biểu và phổ của tín hiệu điều chế DSB-AM tương ứng được thể hiện trên hình 3.1.



Hình 3.1 Phổ của tín hiệu bản tin và của tín hiệu điều chế DSB-AM.

Công suất của tín hiệu đã điều chế được cho bởi

$$\begin{aligned} P_u &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} u^2(t) dt = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} A_c^2 m^2(t) \cos^2(2\pi f_c t) dt = \\ &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} A_c^2 m^2(t) \frac{1 + \cos(4\pi f_c t)}{2} dt = \\ &= A_c^2 \left\{ \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} \frac{m^2(t)}{2} dt + \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} m^2(t) \frac{\cos(4\pi f_c t)}{2} dt \right\} = \end{aligned}$$

$$= A_c^2 \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} \frac{m^2(t)}{2} dt = \quad (3.2.5)$$

$$= \frac{A_c^2}{2} P_m \quad (3.2.6)$$

trong đó  $P_m$  là công suất của tín hiệu bản tin. Phương trình (3.2.5) có được do  $m(t)$  là một tín hiệu thông thấp với tần số nhỏ hơn nhiều so với  $2f_c$  là tần số của  $\cos(2\pi f_c t)$ . Do vậy tích phân

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} m^2(t) \frac{\cos(4\pi f_c t)}{2} dt \quad (3.2.7)$$

sẽ tiến tới không khi  $T \rightarrow \infty$ . Cuối cùng, tỷ số tín/tạp (SNR) đối với một hệ thống DSB-AM thì bằng SNR băng gốc; nghĩa là nó được cho bởi

$$\left(\frac{S}{N}\right)_0 = \frac{P_R}{N_0 W} \quad (3.2.8)$$

trong đó  $P_R$  là công suất tín hiệu nhận được (là công suất trong tín hiệu điều chế tính tại máy thu),  $N_0/2$  là mật độ phổ công suất tạp âm (khi giả sử tạp âm trắng), còn  $W$  là độ rộng băng bản tin.

## BÀI TẬP MINH HỌA

**Bài tập minh họa 3.1 [Điều chế DSB-AM]** Tín hiệu bản tin  $m(t)$  được xác định theo

$$m(t) = \begin{cases} 1, & 0 \leq t \leq t_0 / 3 \\ -2, & t_0 / 3 < t \leq 2t_0 / 3 \\ 0, & \text{với các giá trị khác của } t \end{cases}$$

Bản tin này thực hiện điều chế DSB-AM sóng mang  $c(t) = \cos 2\pi f_c t$  và cho ra tín hiệu đã điều chế được ký hiệu là  $u(t)$ . Giả sử là  $t_0 = 0.15$  s còn  $f_c = 250$  Hz.

1. Hãy có được biểu thức cho  $u(t)$ .
2. Hãy tìm phổ của  $m(t)$  và  $u(t)$ .
3. Giả sử rằng tín hiệu bản tin tuần hoàn với chu kỳ  $T_0 = t_0$ , hãy xác định công suất của tín hiệu đã điều chế.

4. Nếu tạp âm được cộng với tín hiệu đã được điều chế như trong phần 3 sao cho SNR kết quả là 10 dB, hãy tìm công suất tạp âm.

### Lời giải

1.  $m(t)$  có thể viết được theo

$$m(t) = \Pi\left(\frac{t - t_0 / 6}{t_0 / 3}\right) - 2\Pi\left(\frac{t - t_0 / 2}{t_0 / 3}\right)$$

do đó  $u(t) = \left[ \Pi\left(\frac{t - 0.025}{0.05}\right) - 2\Pi\left(\frac{t - 0.075}{0.05}\right) \right] \cos(500\pi t)$  (3.2.9)

2. Bằng cách sử dụng quan hệ biến đổi Fourier tiêu chuẩn  $F[\Pi(t)] = \text{sinc}(f)$  cùng với các định lý về dịch trực và thay đổi tỷ lệ xích của biến đổi Fourier, chúng ta nhận được

$$\begin{aligned} F[m(t)] &= \frac{t_0}{3} e^{-j\pi f t_0 / 3} \sin c\left(\frac{t_0 f}{3}\right) - 2 \frac{t_0}{3} e^{-j\pi f t_0} \sin c\left(\frac{t_0 f}{3}\right) = \\ &= \frac{t_0}{3} e^{-j\pi f t_0 / 3} \sin c\left(\frac{t_0 f}{3}\right) (1 - 2e^{-j2\pi f t_0 / 3}) \end{aligned} \quad (3.2.10)$$

Việc thay  $t_0 = 0.15$  s cho chúng ta

$$F[m(t)] = 0.05e^{-0.05j\pi f} \text{sinc}(0.05f)(1 - 2e^{-0.1j\pi f}) \quad (3.2.11)$$

Đối với tín hiệu đã điều chế  $u(t)$ , chúng ta có

$$\begin{aligned} U(f) &= 0.025e^{-0.05j\pi(f - f_c)} \sin c[(0.05(f - f_c))(1 - 2e^{-0.1j\pi(f - f_c)})] + \\ &\quad + 0.025e^{-0.05j\pi(f + f_c)} \sin c[(0.05(f + f_c))(1 - 2e^{-0.1j\pi(f + f_c)})] \end{aligned}$$

Các đồ thị của các phổ biên độ của các tín hiệu bản tin và tín hiệu sau điều chế được thể hiện trên hình 3.2.

3. Công suất trong tín hiệu sau điều chế được cho bởi

$$P_u = \frac{A_c^2}{2} P_m = \frac{1}{2} P_m$$

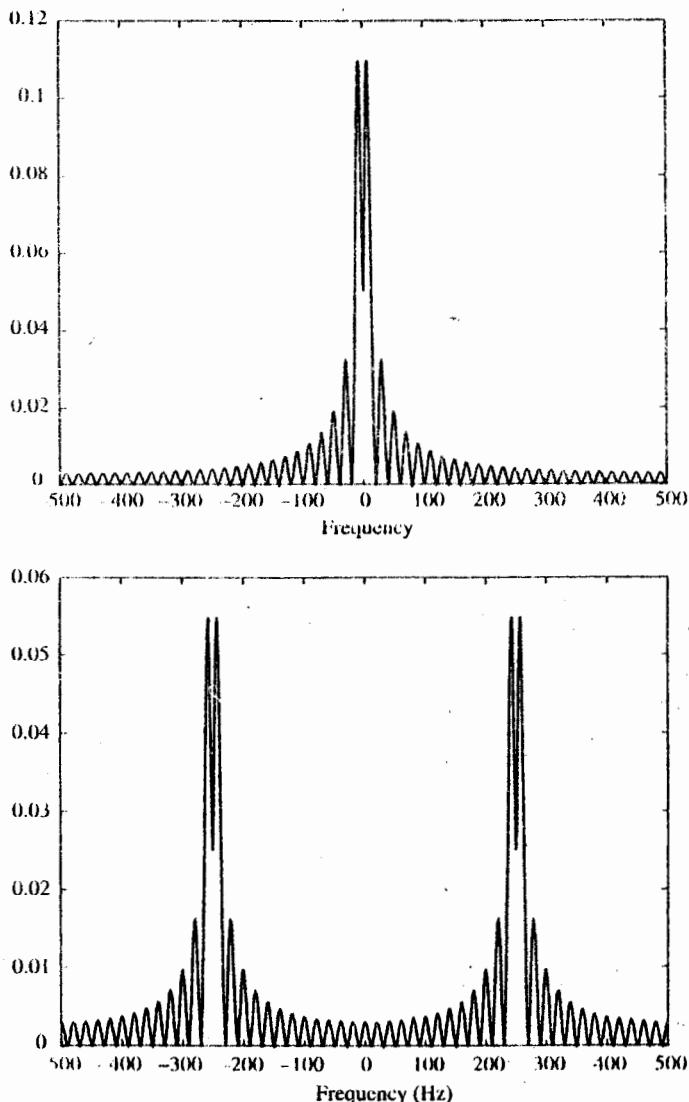
trong đó  $P_m$  là công suất trong tín hiệu bản tin;

$$P_m = \frac{1}{t_0} \int_0^{2t_0/3} m^2(t) dt = \frac{1}{t_0} \left( \frac{t_0}{3} + \frac{4t_0}{3} \right) = \frac{5}{3} = 1.666$$

và  $P_u = \frac{1.666}{2} = 0.833$

4. Ở đây  $10 \log_{10} \left( \frac{P_R}{P_n} \right) = 10$

hay  $P_R = P_u = 10P_n$ , dẫn đến  $P_n = P_u/10 = 0.0833$ .



Hình 3.2 Các phổ biến độ của tín hiệu bản tín và tín hiệu sau điều chế trong bài tập minh họa 3.1.

MATLAB script cho bài tập trên được cho dưới đây.

### m-file

```
% dsb1.m
% Matlab demonstration script for DSB-AM modulation. The message signal
% is +1 for 0 < t < t0/3, -2 for t0/3 < t < 2t0/3 and zero otherwise.
echo on
t0=.15;                                % signal duration
ts=0.001;                               % sampling interval
fc=250;                                 % carrier frequency
snr=20;                                  % SNR in dB (logarithmic)
fs=1/ts;                                 % sampling frequency
df=0.3;                                  % desired freq. resolution
t=[0:ts:t0];                            % time vector
snr_lin=10^(snr/10);                     % linear SNR
% message signal
m=[ones(1,t0/(3*ts)),-2*ones(1,t0/(3*ts)),zeros(1,t0/(3*ts)+1)];
c=cos(2*pi*fc.*t);                      % carrier signal
u=m.*c;                                  % modulated signal
[M,m,df1]=fftseq(m,ts,df);              % Fourier transform
M=M/fs;                                 % scaling
[U,u,df1]=fftseq(u,ts,df);              % Fourier transform
U=U/fs;                                 % scaling
[C,c,df1]=fftseq(c,ts,df);              % Fourier transform
f=[0:df1:(length(m)-1)]-fs/2;          % freq. vector
signal_power=spower(u(1:length(t)));    % power in modulated signal
noise_power=signal_power/snr_lin;        % compute noise power
noise_std=sqrt(noise_power);            % compute noise standard deviation
noise=noise_std*randn(1,length(u));     % generate noise
r=u+noise;                             % add noise to the modulated signal
[R,r,df1]=fftseq(r,ts,df);              % spectrum of the signal+noise
R=R/fs;                                 % scaling
pause % Press a key to show the modulated signal power
signal_power
pause % Press any key to see a plot of the message
clf
subplot(2,2,1)
plot(t,m(1:length(t)))
xlabel('Time')
title('The message signal')
pause % Press any key to see a plot of the carrier
subplot(2,2,2)
plot(t,c(1:length(t)))
```

```
xlabel('Time')
title('The carrier')
pause % Press any key to see a plot of the modulated signal
subplot(2,2,3)
plot(t,u(1:length(t)))
xlabel('Time')
title('The modulated signal')
pause % Press any key to see a plots of the magnitude of the message and the
% modulated signal in the frequency domain.
subplot(2,1,1)
plot(f,abs(fftshift(M)))
xlabel('Frequency')
title('Spectrum of the message signal')
subplot(2,1,2)
plot(f,abs(fftshift(U)))
title('Spectrum of the modulated signal')
xlabel('Frequency')
pause % Press a key to see a noise sample
subplot(2,1,1)
plot(t,noise(1:length(t)))
title('noise sample')
xlabel('Time')
pause % Press a key to see the modulated signal and noise
subplot(2,1,2)
plot(t,r(1:length(t)))
title('Signal and noise')
xlabel('Time')
pause % Press a key to see the modulated signal and noise in freq. domain
subplot(2,1,1)
plot(f,abs(fftshift(U)))
title('Signal spectrum')
xlabel('Frequency')
subplot(2,1,2)
plot(f,abs(fftshift(R)))
title('Signal and noise spectrum')
xlabel('Frequency')
```

## BÀI TẬP MINH HỌA

**Bài tập minh họa 3.2 [Điều chế DSB đối với một tín hiệu có băng gầm như hạn chế]** Tín hiệu bản tin  $m(t)$  được cho bởi

$$m(t) = \begin{cases} \sin c(100t), & |t| \leq t_0 \\ 0, & \text{với các giá trị khác của } t \end{cases}$$

trong đó  $t_0=0.1$ . Bản tin này điều chế sóng mang  $c(t)=\cos(2\pi f_c t)$ , trong đó  $f_c=250$  Hz.

1. Hãy xác định tín hiệu sau điều chế  $u(t)$ .

2. Hãy xác định các phổ của  $m(t)$  và  $u(t)$ .

3. Nếu tín hiệu bản tin tuân hoàn với chu kỳ  $T_0=0.2$  s, hãy xác định công suất trong tín hiệu sau điều chế.

4. Nếu một tạp nhiễu Gauss được cộng vào tín hiệu sau điều chế sao cho có tỷ số SNR là 10 dB, hãy tìm công suất tạp nhiễu.

### Lời giải

1. Chúng ta có

$$u(t) = m(t)c(t) = \begin{cases} \sin c(100t)\cos(500t), & |t| \leq 0.1 \\ 0, & \text{với các giá trị khác của } t \end{cases} \quad (3.2.12)$$

$$= \text{sinc}(100t)\Pi(5t).\cos(500t) \quad (3.2.13)$$

2. Một đồ thị các phổ của  $m(t)$  và  $u(t)$  được thể hiện trên hình 3.3. Như hình vẽ cho thấy, tín hiệu bản tin là một tín hiệu có băng gần như hạn chế với một độ rộng băng là 50 Hz.

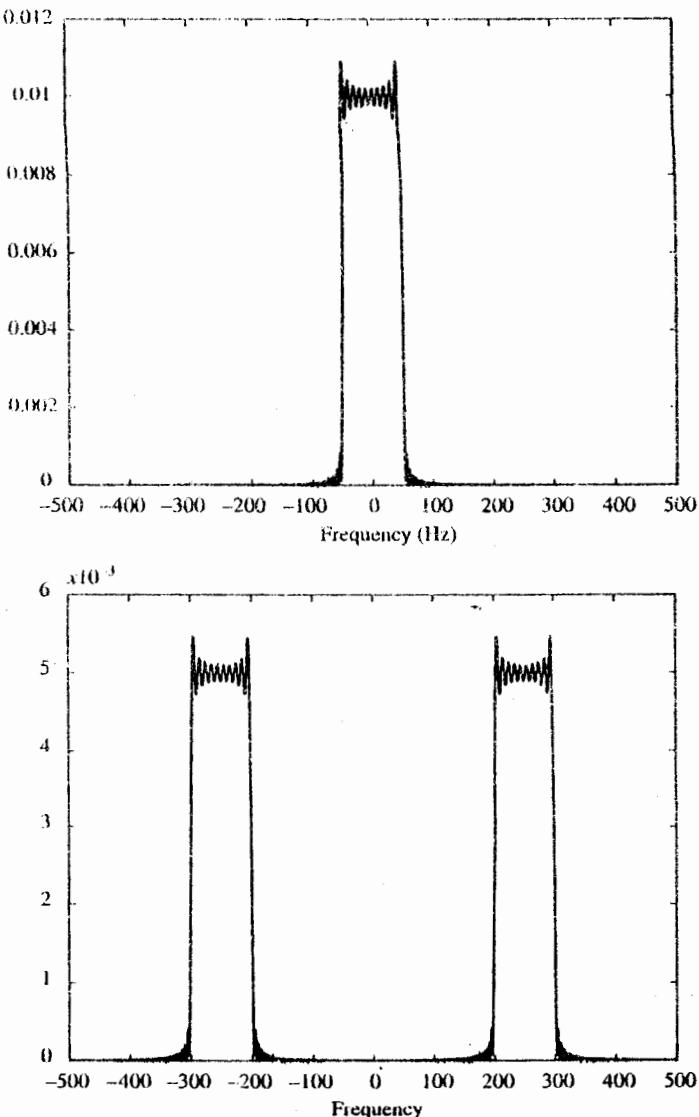
3. Công suất trong tín hiệu sau điều chế là một nửa công suất trong tín hiệu bản tin. Công suất trong tín hiệu bản tin được cho bởi

$$P_m = \frac{1}{0.2} \int_{-0.1}^{0.1} \sin^2(100t) dt$$

tích phân này có thể tính được bằng cách sử dụng *m-file* của MATLAB là **quad8.m** mà nó dẫn đến kết quả  $P_m=0.0495$  và do đó  $P_u=0.0247$ .

### 4. Ở đây

$$10 \log_{10} \left( \frac{P_R}{P_n} \right) = 10 \Rightarrow P_n = 0.1 P_R = 0.1 P_u = 0.00247$$



Hình 3.3 Các phổ của các tín hiệu bản tin và sau điều chế trong bài tập minh họa 3.2.

MATLAB *script* đối với bài tập này như sau

### m-file

```
% dsb2.m
% Matlab demonstration script for DSB-AM modulation. The message signal
% is m(t)=sinc(100t).
echo on
t0=.2; % signal duration
ts=0.001; % sampling interval
fc=250; % carrier frequency
```

```

snr=20; % SNR in dB (logarithmic)
fs=1/ts; % sampling frequency
df=0.3; % required freq. resolution
t=[-t0/2:ts:t0/2]; % time vector
snr_lin=10^(snr/10); % linear SNR
m=sinc(100*t); % the message signal
c=cos(2*pi*fc.*t); % the carrier signal
u=m.*c; % the DSB-AM modulated signal
[M,m,df1]=fftseq(m,ts,df); % Fourier transform
M=M/fs; % scaling
[U,u,df1]=fftseq(u,ts,df); % Fourier transform
U=U/fs; % scaling
f=[0:df1:df1*(length(m)-1)]-fs/2; % frequency vector
signal_power=spower(u(1:length(t))); % compute modulated signal power
noise_power=signal_power/snr_lin; % compute noise power
noise_std=sqrt(noise_power); % compute noise standard deviation
noise=noise_std*randn(1,length(u)); % generate noise sequence
r=u+noise; % add noise to the modulated signal
[R,r,df1]=fftseq(r,ts,df); % Fourier transform
R=R/fs; % scaling
pause % Press a key to show the modulated signal power
signal_power
pause % Press any key to see a plot of the message
clf
subplot(2,2,1)
plot(t,m(1:length(t)))
xlabel('Time')
title('The message signal')
pause % Press any key to see a plot of the carrier
subplot(2,2,2)
plot(t,c(1:length(t)))
xlabel('Time')
title('The carrier')
pause % Press any key to see a plot of the modulated signal
subplot(2,2,3)
plot(t,u(1:length(t)))
xlabel('Time')
title('The modulated signal')
pause % Press any key to see a plot of the magnitude of the message and the
% modulated signal in the frequency domain.
subplot(2,1,1)
plot(f,abs(fftshift(M)))

```

```
xlabel('Frequency')
title('Spectrum of the message signal')
subplot(2,1,2)
plot(f,abs(fftshift(U)))
title('Spectrum of the modulated signal')
xlabel('Frequency')
pause % Press a key to see a noise sample
subplot(2,1,1)
plot(t,noise(1:length(t)))
title('noise sample')
xlabel('Time')
pause % Press a key to see the modulated signal and noise
subplot(2,1,2)
plot(t,r(1:length(t)))
title('Signal and noise')
xlabel('Time')
pause % Press a key to see the modulated signal and noise in freq. domain
subplot(2,1,1)
plot(f,abs(fftshift(U)))
title('Signal spectrum')
xlabel('Frequency')
subplot(2,1,2)
plot(f,abs(fftshift(R)))
title('Signal and noise spectrum')
xlabel('Frequency')
```

### Cái gì sẽ nêu...

Điều gì sẽ xảy ra nếu như thời gian tồn tại của tín hiệu bản tin  $t_0$  thay đổi; nói riêng, việc có  $t_0$  lớn hay nhỏ thì có ảnh hưởng gì? Ảnh hưởng tới công suất của tín hiệu bản tin và tín hiệu sau điều chế là gì?

**M-file dsb\_mod.m** được cho dưới đây là một bộ điều chế DSB tổng quát với tín hiệu bản tin được cho trong véc-tơ  $m$  trên một tần số  $f_c$ .

### m-file

```
function u=dsb_mod(m,ts,fc)
%           u=dsb_mod(m,ts,fc)
%DSB_MOD   takes signal m sampled at ts and carrier freq. fc as input and
%           returns the DSB modulated signal. ts << 1/2fc. The modulated
%           signal is normalized to have half the message power.
%           The message signal starts at 0.
```

```
t=[0:length(m)-1]*ts;
u=m.*cos(2*pi*t);
```

### 3.2.2 AM TRUYỀN THỐNG

Về nhiều khía cạnh, AM truyền thống hoàn toàn tương tự với DSB-AM; sự khác biệt duy nhất là ở chỗ trong AM truyền thống thì  $m(t)$  được thay thế bằng  $[1+am_n(t)]$ , trong đó  $m_n(t)$  là tín hiệu bản tin chuẩn hoá (tức là  $|m_n(t)| \leq 1$ ) còn  $a$  là chỉ số điều chế mà nó là một hằng số dương nằm giữa 0 và 1. Do vậy, chúng ta có

$$u(t) = A_c[1 + am_n(t)]\cos(2\pi f_c t) \quad (3.2.14)$$

và

$$U(f) = \frac{A_c}{2}[\delta(f - f_c) + aM_n(f - f_c) + \delta(f + f_c) + aM_n(f + f_c)] \quad (3.2.15)$$

Tác dụng thực sự của việc thay đổi thang tỷ lệ tín hiệu bản tin và cộng thêm vào nó một hằng số nằm ở chỗ số hạng  $[1+am_n(t)]$  thì luôn luôn dương. Điều này làm cho việc giải điều chế các tín hiệu này trở nên dễ dàng hơn nhiều nhờ việc áp dụng các bộ tách sóng đường bao. Hãy chú ý đến sự tồn tại của thành phần hình sin tại tần số  $f_c$  trong  $U(f)$ . Điều đó có nghĩa là một phần (thường là đáng kể) của công suất được phát đi thì nằm trong sóng mang tín hiệu mà nó thì không thực sự phục vụ việc truyền thông tin. Việc này cho thấy rằng so với DSB-AM thì AM truyền thống là một sơ đồ điều chế kém kinh tế hơn về phương diện tận dụng công suất. Độ rộng băng, tất nhiên, bằng với độ rộng băng của DSB-AM và được cho bởi

$$B_T=2W \quad (3.2.16)$$

Các đồ thị miền tần số tiêu biểu của tín hiệu bản tin và của tín hiệu AM truyền thống tương ứng được thể hiện trên hình 3.4.

Công suất của tín hiệu sau điều chế, khi giả thiết rằng tín hiệu bản tin là một tín hiệu có kỳ vọng bằng không, được cho bởi

$$P_u = \frac{A_c^2}{2}[1 + a^2 P_{m_n}] \quad (3.2.17)$$

mà nó gồm hai phần:  $A_c^2 / 2$  biểu thị công suất sóng mang và  $A_c^2 / 2 a^2 P_{m_n}$  là

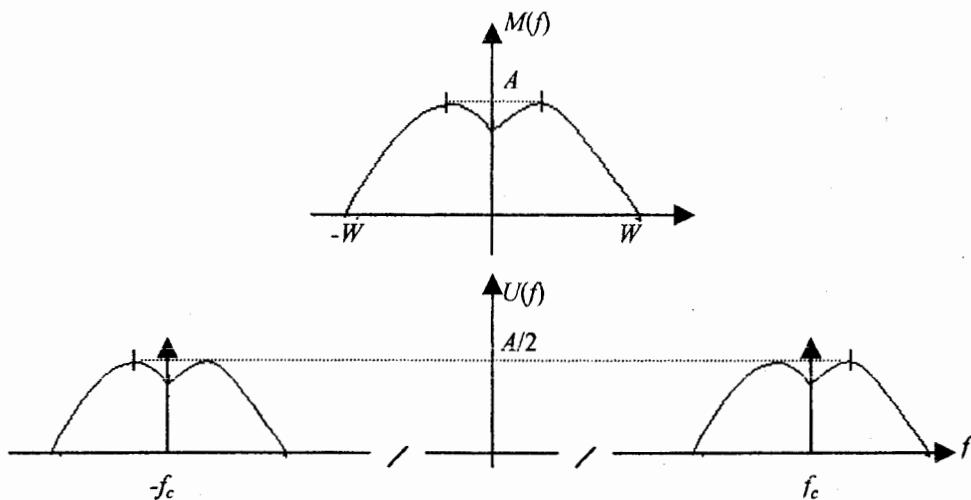
công suất trong phần mang bản tin của tín hiệu sau điều chế. Đây là công suất mà nó thực sự được dùng để truyền bản tin đi. Tỷ lệ giữa công suất được sử dụng cho truyền bản tin và công suất tổng cộng trong tín hiệu sau điều chế được gọi là hiệu suất điều chế và được xác định bởi

$$\eta = \frac{a^2 P_{m_n}}{1 + a^2 P_{m_n}} \quad (3.2.18)$$

Do  $|m_n(t)| \leq 1$  và  $a \leq 1$ , chúng ta luôn có  $\eta \leq 0.5$ . Trong thực tế, giá trị của  $\eta$  tuy vậy thường nằm quanh 0.1. Tỷ số tín trên tạp được cho bởi

$$\left( \frac{S}{N} \right)_0 = \eta \frac{P_r}{N_0 W} \quad (3.2.19)$$

trong đó  $\eta$  là hiệu suất điều chế. Chúng ta thấy rằng so với DSB-AM, SNR bị giảm đi theo một hệ số bằng  $\eta$ . Sự sút giảm về chất lượng hoạt động này là một hệ quả trực tiếp của việc một phần đáng kể trong công suất tổng cộng lại nằm ở sóng mang (các hàm *delta* trong biểu thức phổ), mà nó không mang thông tin gì và bị lọc bỏ đi tại máy thu.



Hình 3.4 Các phổ của tín hiệu bản tin và tín hiệu AM truyền thống.

## BÀI TẬP MINH HỌA

### Bài tập minh họa 3.3 [AM truyền thống] Tín hiệu bản tin

$$m(t) = \begin{cases} 1, & 0 \leq t < t_0 / 3 \\ -2, & t_0 / 3 \leq t < 2t_0 / 3 \\ 0, & \text{với các giá trị khác của } t \end{cases}$$

điều chế sóng mang  $c(t)=\cos 2\pi f_c t$  bằng cách sử dụng sơ đồ AM truyền thống. Giả sử rằng  $f_c=250$  Hz còn  $t_0=0.15$  s; chỉ số điều chế  $a=0.85$ .

1. Hãy tìm biểu thức của tín hiệu sau điều chế.
2. Hãy xác định các phổ của tín hiệu bản tin và tín hiệu sau điều chế.
3. Nếu tín hiệu bản tin tuần hoàn với một chu kỳ bằng  $t_0$ , hãy xác định công suất của tín hiệu sau điều chế và hiệu suất điều chế.
4. Nếu một tín hiệu tạp nhiễu được cộng vào tín hiệu bản tin sao cho SNR tại lối ra của bộ giải điều chế là 10 dB, hãy tìm công suất của tín hiệu tạp nhiễu.

### Lời giải

1. Trước hết hãy để ý rằng  $\max|m(t)| = 2$ ; cho nên  $m_n(t)=m(t)/2$ . Từ đó, chúng ta có

$$\begin{aligned} u(t) &= [1 + 0.85 \frac{m(t)}{2}] \cos(2\pi f_c t) = \\ &= [1 + 0.425 \Pi(\frac{t - 0.025}{0.05}) - 0.85 \Pi(\frac{t - 0.075}{0.05})] \cos(500\pi t) \end{aligned}$$

Một đồ thị của tín hiệu bản tin và sau điều chế được cho trên hình 3.5.

2. Đối với tín hiệu bản tin, chúng ta có

$$F[m(t)] = 0.05e^{-0.05j\pi f} \sin c[0.05(f - 250)] (1 - e^{-0.1j\pi f}) \quad (3.2.20)$$

còn đối với tín hiệu sau điều chế

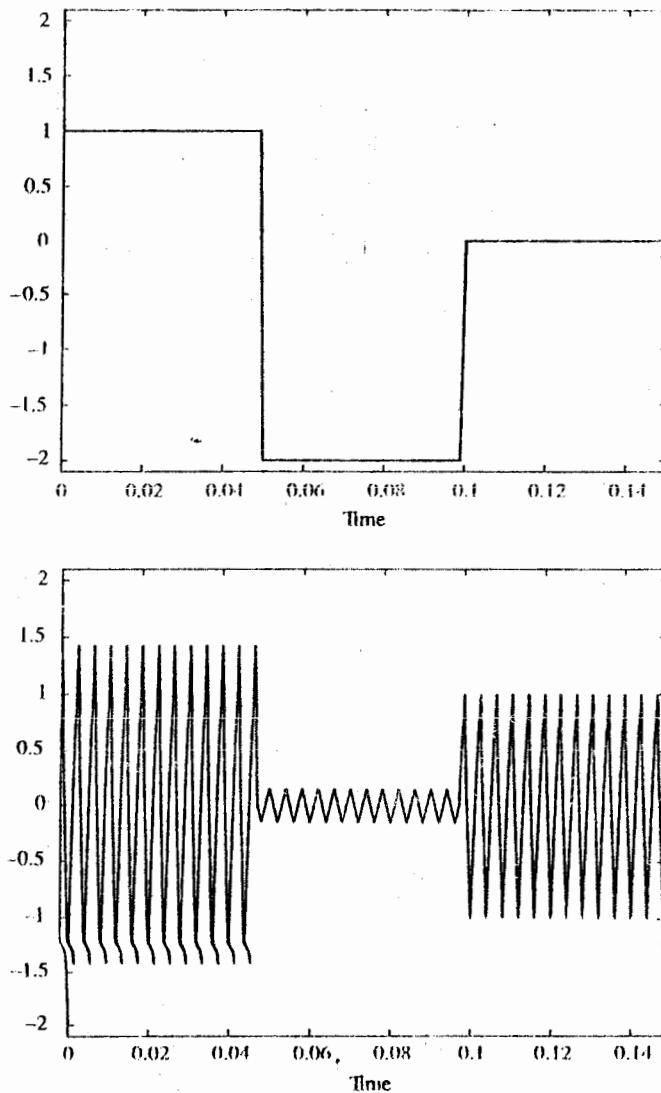
$$\begin{aligned} U(f) &= 0.010625e^{-0.05j\pi(f-250)} \sin c[0.05(f - 250)] (1 - 2e^{-0.1j\pi(f-250)}) \\ &\quad + 0.010625e^{-0.05j\pi(f+250)} \sin c[0.05(f + 250)] (1 - 2e^{-0.1j\pi(f+250)}) \end{aligned}$$

Các đồ thị được cho trên hình 3.6.

Hãy chú ý rằng các thang trên hai đồ thị phổ đó khác nhau. Sự có mặt của hai hàm *delta* trong phổ của tín hiệu sau điều chế thấy được rõ rệt.

3. Công suất của tín hiệu bản tin có thể nhận được theo

$$P_m = \frac{1}{0.15} \left[ \int_0^{0.05} dt + 4 \int_0^{0.1} dt \right] = 1.667$$



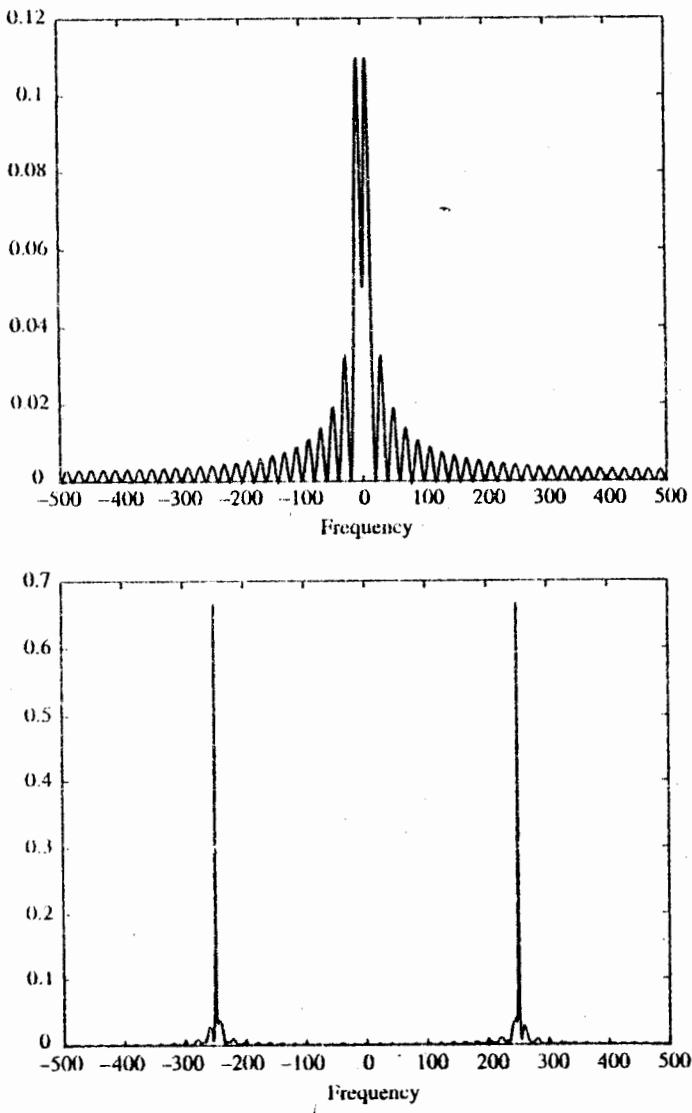
Hình 3.5 Các tín hiệu bản tin và sau điều chế trong bài tập minh họa 3.3.

Công suất trong tín hiệu bản tin chuẩn hoá  $P_{m_n}$  được cho bởi

$$P_{m_n} = \frac{1}{4} P_m = \frac{1.667}{4} = 0.4167$$

và hiệu suất điều chế là

$$\eta = \frac{a^2 P_{m_n}}{1 + a^2 P_{m_n}} = \frac{0.85^2 \times 0.4167}{1 + 0.85^2 \times 0.4167} = 0.2314$$



Hình 3.6 Các phổ của tín hiệu bản tin và tín hiệu sau điều chế trong bài tập minh họa 3.3.

Công suất tín hiệu sau điều chế được cho bởi (E ký hiệu cho kỳ vọng)

$$P_u = \frac{A_c^2}{2} E[1 + \alpha m_n(t)]^2 = \frac{1}{2} (1 + 0.3010 - 1.7 \times \frac{0.025}{0.15}) = 0.5088$$

4. Trong trường hợp này

$$10 \log_{10} \left[ \eta \frac{P_R}{N_0 W} \right] = 10$$

hay  $\eta \frac{P_R}{N_0 W} = 10$

Thay  $\eta=0.2314$  và  $P_R=P_u=0.5088$  vào ta được

$$P_n = \frac{\eta P_u}{10} = 0.0118$$

### Bình luận

Trong khi tìm công suất tín hiệu sau điều chế trong bài tập này, chúng ta đã không thể sử dụng quan hệ

$$P_u = \frac{A_c^2}{2} [1 + a^2 P_{m_n}]$$

bởi vì trong bài tập này  $m(t)$  không phải là một tín hiệu có giá trị trung bình bằng không.

MATLAB script cho bài tập này được cho dưới đây.

### m-file

```
% am.m
% Matlab demonstration script for DSB-AM modulation. The message signal
% is +1 for 0 < t < t0/3, -2 for t0/3 < t < 2t0/3 and zero otherwise.
echo on
t0=.15; % signal duration
ts=0.001; % sampling interval
fc=250; % carrier frequency
snr=10; % SNR in dB (logarithmic)
a=0.85; % Modulation index
fs=1/ts; % sampling frequency
t=[0:ts:t0]; % time vector
df=0.2; % required frequency resolution
snr_lin=10^(snr/10); % SNR
% message signal
m=[ones(1,t0/(3*ts)),-2*ones(1,t0/(3*ts)),zeros(1,t0/(3*ts)+1)];
c=cos(2*pi*fc.*t); % carrier signal
m_n=m/max(abs(m)); % normalized message signal
[M,m,df1]=fftseq(m,ts,df); % Fourier transform
M=M/fs; % scaling
f=[0:df1:df1*(length(m)-1)]-fs/2; % frequency vector
u=(1+a*m_n).*c; % modulated signal
```

```

[U,u,df1]=fftseq(u,ts,df); % Fourier transform
U=U/fs; % scaling
signal_power=spower(u(1:length(t))); % power in modulated signal
% power in normalized message
pmn=spower(m(1:length(t)))/(max(abs(m)))^2;
eta=(a^2*pmn)/(1+a^2*pmn); % modulation efficiency
noise_power=eta*signal_power/snr_lin; % noise power
noise_std=sqrt(noise_power); % noise standard deviation
noise=noise_std*randn(1,length(u)); % generate noise
r=u+noise; % add noise to the modulated signal
[R,r,df1]=fftseq(r,ts,df); % Fourier transform
R=R/fs; % scaling
pause % Press a key to show the modulated signal power
signal_power
pause % Press a key to show the modulation efficiency
eta
pause % Press any key to see a plot of the message
subplot(2,2,1)
plot(t,m(1:length(t)))
axis([0 0.15 -2.1 2.1])
xlabel('Time')
title('The message signal')
pause
pause % Press any key to see a plot of the carrier
subplot(2,2,2)
plot(t,c(1:length(t)))
axis([0 0.15 -2.1 2.1])
xlabel('Time')
title('The carrier')
pause % Press any key to see a plot of the modulated signal
subplot(2,2,3)
plot(t,u(1:length(t)))
axis([0 0.15 -2.1 2.1])
xlabel('Time')
title('The modulated signal')
pause % Press any key to see a plots of the magnitude of the message and the
% modulated signal in the frequency domain.
subplot(2,1,1)
plot(f,abs(fftshift(M)))
xlabel('Frequency')
title('Spectrum of the message signal')
subplot(2,1,2)

```

```

plot(f,abs(fftshift(U)))
title('Spectrum of the modulated signal')
xlabel('Frequency')
pause % Press a key to see a noise sample
subplot(2,1,1)
plot(t,noise(1:length(t)))
title('noise sample')
xlabel('Time')
pause % Press a key to see the modulated signal and noise
subplot(2,1,2)
plot(t,r(1:length(t)))
title('Signal and noise')
xlabel('Time')
pause % Press a key to see the modulated signal and noise in freq. domain
subplot(2,1,1)
plot(f,abs(fftshift(U)))
title('Signal spectrum')
xlabel('Frequency')
subplot(2,1,2)
plot(f,abs(fftshift(R)))
title('Signal and noise spectrum')
xlabel('Frequency')

```

MATLAB *m-file am\_mode.m* được cho dưới đây là một bộ điều chế AM truyền thống tổng quát.

### **m-file**

```

function u=am_mod(a,m,ts,fc)
%           u=am_mod(a,m,ts,fc)
%AM_MOD takes signal m sampled at ts and carrier freq. fc as input and returns
%           the AM modulated signal. "a" is the modulation index and ts << 1/2fc.

t=[0:length(m)-1]*ts;
c=cos(2*pi*fc.*t);
m_n=m/max(abs(m));
u=(1+a*m_n).*c;

```

### **3.2.3 SSB-AM**

SSB-AM bắt nguồn từ DSB-AM bằng cách loại bỏ một trong các băng biên. Do đó, nó chiếm một nửa độ rộng băng như của DSB-AM. Tuỳ theo băng biên nào

còn lại, hoặc băng biên trên hoặc băng biên dưới, mà có hai kiểu SSB-AM: USSB-AM và LSSB-AM. Biểu diễn trên miền thời gian của các tín hiệu này được cho bởi

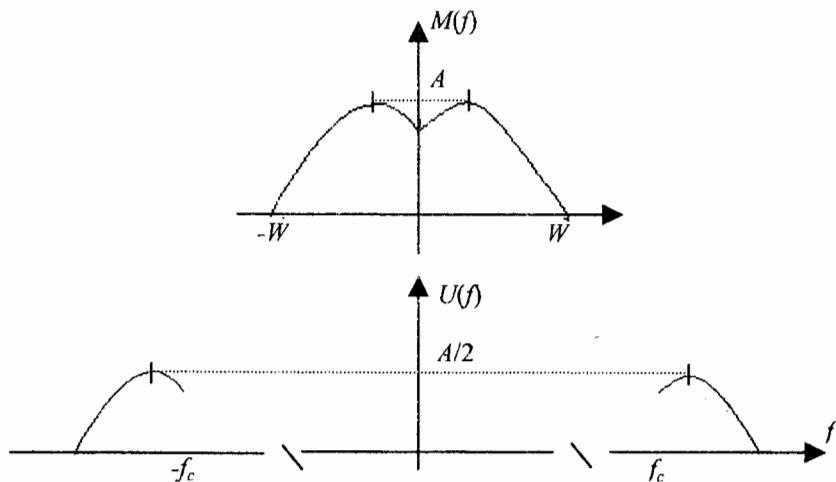
$$u(t) = \frac{A_c}{2} m(t) \cos(2\pi f_c t) \mp \frac{A_c}{2} \hat{m}(t) \sin(2\pi f_c t) \quad (3.2.21)$$

trong đó dấu trừ ứng với USSB-AM còn dấu cộng ứng với LSSB-AM. Tín hiệu được ký hiệu bằng  $\hat{m}(t)$  là biến đổi Hilbert của  $m(t)$ , được xác định bởi  $\hat{m}(t) = m(t) * 1/\pi t$  hay trong miền tần số  $\hat{M}(f) = -j \operatorname{sgn}(f) M(f)$ . Nói một cách khác đi, biến đổi Hilbert của một tín hiệu biểu thị một sự dịch pha  $\pi/2$  đối với mọi thành phần của tín hiệu đó. Trên miền tần số, chúng ta có

$$U_{USSB}(f) = \begin{cases} M(f - f_c) + M(f + f_c), & f_c \leq |f| \\ 0, & \text{với mọi giá trị khác của } f \end{cases} \quad (3.2.22)$$

$$\text{và } U_{LSSB}(f) = \begin{cases} M(f - f_c) + M(f + f_c), & |f| \leq f_c \\ 0, & \text{với mọi giá trị khác của } f \end{cases} \quad (3.2.23)$$

Các đồ thị phổ tiêu biểu của một tín hiệu bản tin và tín hiệu USSB-AM tương ứng được thể hiện trên hình 3.7.



Hình 3.7 Các phổ của tín hiệu bản tin và tín hiệu USSB-AM.

Độ rộng băng của tín hiệu SSB là một nửa độ rộng băng của DSB và AM truyền thống và do vậy bằng với độ rộng băng của tín hiệu bản tin; nghĩa là

$$B_T = W \quad (3.2.24)$$

Công suất của tín hiệu SSB được cho bởi

$$P_u = \frac{A_c^2}{4} P_m \quad (3.2.25)$$

Hãy chú ý rằng công suất này bằng một nửa của công suất của tín hiệu DSB-AM tương ứng do một trong hai băng biên đã bị loại bỏ. Mặt khác, do tín hiệu sau điều chế có độ rộng băng bằng một nửa so với tín hiệu DSB-AM tương ứng nên công suất tạp nhiễu tại đầu cuối máy thu cũng là một nửa so với trường hợp DSB-AM, và vì thế SNR trong cả hai hệ thống như nhau; nghĩa là

$$\left( \frac{S}{N} \right)_0 = \frac{P_R}{N_0 W} \quad (3.2.26)$$

### BÀI TẬP MINH HỌA

#### Bài tập minh họa 3.4 [Thí dụ đơn giản] Tín hiệu bản tin

$$m(t) = \begin{cases} 1, & 0 \leq t < t_0 / 3 \\ -2, & t_0 / 3 \leq t < 2t_0 / 3 \\ 0, & \text{với các giá trị khác của } t \end{cases}$$

điều chế sóng mang  $c(t)=\cos(2\pi f_c t)$  bằng sử dụng sơ đồ LSSB-AM. Giả sử rằng  $t_0=0.15$  s còn  $f_c=250$  Hz.

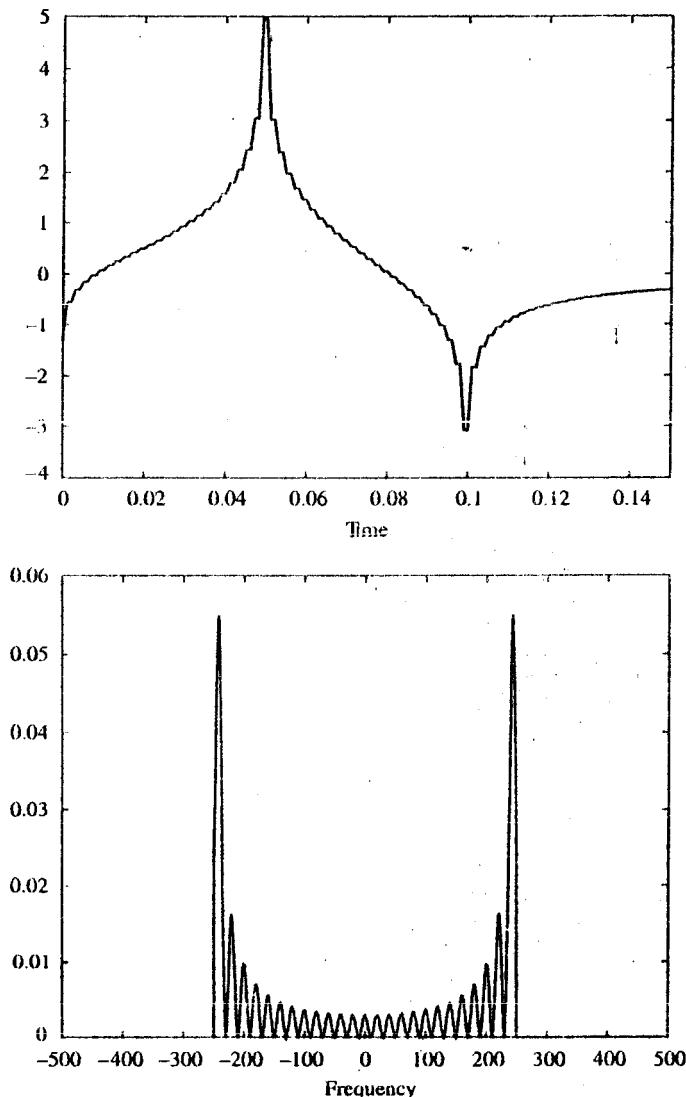
1. Hãy vẽ đồ thị của biến đổi Hilbert của tín hiệu bản tin và tín hiệu sau điều chế  $u(t)$ . Hãy vẽ cả đồ thị phổ của tín hiệu sau điều chế.
2. Giả sử rằng tín hiệu bản tin tuần hoàn với chu kỳ  $t_0$ , hãy xác định công suất tín hiệu sau điều chế.
3. Nếu một tạp nhiễu được cộng vào tín hiệu sau điều chế sao cho SNR sau giải điều chế là 10 dB, hãy xác định công suất tạp nhiễu.

#### Lời giải

1. Biến đổi Hilbert của tín hiệu bản tin có thể tính được bằng cách sử dụng *m-file* biến đổi Hilbert của MATLAB là **hilbert.m**. Tuy nhiên nên chú ý rằng hàm này cho chúng ta một dãy phức mà phần thực của nó là tín hiệu gốc còn phần ảo của nó mới là biến đổi Hilbert mong muốn. Do đó, biến đổi Hilbert của dãy m nhận được nhờ lệnh **imag(hilbert(m))**. Nay giờ, sử dụng quan hệ

$$u(t) = m(t) \cos(2\pi f_c t) + \hat{m}(t) \sin(2\pi f_c t) \quad (3.2.27)$$

chúng ta có thể tìm được tín hiệu sau điều chế. Các đồ thị của  $\hat{m}(t)$  và phổ của tín hiệu  $u(t)$  sau điều chế LSSB-AM được cho trên hình 3.8.



Hình 3.8 Biến đổi Hilbert và phổ của tín hiệu điều chế LSSB-AM.

2. Công suất của tín hiệu bản tin là

$$P_m = \frac{1}{0.15} \int_0^{0.15} m^2(t) dt = 1.667$$

và do đó

$$P_u = \frac{A_c^2}{4} P_m = 0.416$$

3. Tỷ số SNR sau giải điều chế được cho bởi

$$10 \log_{10} \left( \frac{P_R}{P_n} \right)_0 = 10$$

Do vậy,  $P_n=0.1P_R=0.1P_u=0.0416$ .

MATLAB script cho bài tập này như sau đây.

### m-file

```
% lssb.m
% Matlab demonstration script for LSSB-AM modulation. The message signal
% is +1 for 0 < t < t0/3, -2 for t0/3 < t < 2t0/3 and zero otherwise.
echo on
t0=.15; % signal duration
ts=0.001; % sampling interval
fc=250; % carrier frequency
snr=10; % SNR in dB (logarithmic)
fs=1/ts; % sampling frequency
df=0.25; % desired freq. resolution
t=[0:ts:t0]; % time vector
snr_lin=10^(snr/10); % SNR
% the message vector
m=[ones(1,t0/(3*ts)),-2*ones(1,t0/(3*ts)),zeros(1,t0/(3*ts)+1)];
c=cos(2*pi*fc.*t); % carrier vector
udsb=m.*c; % DSB modulated signal
[UDSB,udssb,df1]=fftseq(udsb,ts,df); % Fourier transform
UDSB=UDSB/fs; % scaling
f=[0:df1:df1*(length(udssb)-1)]-fs/2; % frequency vector
n2=ceil(fc/df1); % location of carrier in freq. vector
% remove the upper sideband from DSB
UDSB(n2:length(UDSB)-n2)=zeros(size(UDSB(n2:length(UDSB)-n2)));
ULSSB=UDSB; % generate LSSB-AM spectrum
[M,m,df1]=fftseq(m,ts,df); % Fourier transform
M=M/fs; % scaling
u=real(ifft(ULSSB))*fs; % generate LSSB signal from spectrum
signal_power=spower(udsb(1:length(t)))/2;
% % compute signal power
noise_power=signal_power/snr_lin; % compute noise power
noise_std=sqrt(noise_power); % compute noise standard deviation
```

```
noise=noise_std*randn(1,length(u)); % generate noise vector
r=u+noise; % add the signal to noise
[R,r,df1]=fftseq(r,ts,df); % Fourier transform
R=R/fs; % scaling
pause % Press a key to show the modulated signal power
signal_power
pause % Press any key to see a plot of the message signal
clf
subplot(2,1,1)
plot(t,m(1:length(t)))
axis([0,0.15,-2.1,2.1])
xlabel('Time')
title('The message signal')
pause % Press any key to see a plot of the carrier
subplot(2,1,2)
plot(t,c(1:length(t)))
xlabel('Time')
title('The carrier')
pause % Press any key to see a plot of the modulated signal and its spectrum
clf
subplot(2,1,1)
plot([0:ts:ts*(length(u)-1)/8],u(1:length(u)/8))
xlabel('Time')
title('The LSSB-AM modulated signal')
subplot(2,1,2)
plot(f,abs(fftshift(ULSSB)))
xlabel('Frequency')
title('Spectrum of the LSSB-AM modulated signal')
pause % Press any key to see the spectra of the message and the modulated signals
clf
subplot(2,1,1)
plot(f,abs(fftshift(M)))
xlabel('Frequency')
title('Spectrum of the message signal')
subplot(2,1,2)
plot(f,abs(fftshift(ULSSB)))
xlabel('Frequency')
title('Spectrum of the LSSB-AM modulated signal')

pause % Press any key to see a noise sample
subplot(2,1,1)
plot(t,noise(1:length(t)))
title('noise sample')
```

```

xlabel('Time')
pause % Press a key to see the modulated signal and noise
subplot(2,1,2)
plot(t,r(1:length(t)))
title('Modulated signal and noise')
xlabel('Time')
subplot(2,1,1)
pause % Press any key to see the spectrum of the modulated signal
plot(f,abs(fftshift(ULSSB)))
title('Modulated signal spectrum')
xlabel('Frequency')
subplot(2,1,2)

pause % Press a key to see the modulated signal noise in freq. domain
plot(f,abs(fftshift(R)))
title('Modulated signal noise spectrum')
xlabel('Frequency')

```

Các *m-file* **ussb\_mod.m** và **lssb\_mod.m** được cho dưới đây sử dụng các sơ đồ điều chế USSB và LSSB thực hiện điều chế tín hiệu bản tin được cho trong véc-tơ *m*.

#### **m-file**

```

function u=ussb_mod(m,ts,fc)
%           u=ussb_mod(m,ts,fc)
%USSB_MOD takes signal m sampled at ts and carrier freq. fc as input and
% returns the USSB modulated signal. ts << 1/2fc.
t=[0:length(m)-1]*ts;
u=m.*cos(2*pi*t)-imag(hilbert(m)).*sin(2*pi*t);

```

#### **m-file**

```

function u=lssb_mod(m,ts,fc)
%           u=lssb_mod(m,ts,fc)
%LSSB_MOD takes signal m sampled at ts and carrier freq. fc as input and
% returns the LSSB modulated signal. ts << 1/2fc.
t=[0:length(m)-1]*ts;
u=m.*cos(2*pi*t)+imag(hilbert(m)).*sin(2*pi*t);

```

### **3.3 GIẢI ĐIỀU CHẾ CÁC TÍN HIỆU ĐIỀU CHẾ BIÊN ĐỘ**

Giải điều chế là quá trình tách tín hiệu bản tin từ tín hiệu đã được điều chế. Quá trình giải điều chế tùy thuộc vào kiểu điều chế đã được áp dụng. Đối với DSB- AM và

SSB-AM, phương pháp giải điều chế là giải điều chế kết hợp (*coherent*) mà nó yêu cầu phải có một tín hiệu có cùng tần số và pha của sóng mang tại máy thu. Đối với AM truyền thống, các bộ tách sóng đường bao được sử dụng cho giải điều chế. Trong trường hợp này việc biết chính xác tần số và pha của sóng mang không phải là điều sống còn, cho nên quá trình giải điều chế dễ dàng hơn nhiều. Việc giải điều chế kết hợp đối với DSB-AM và SSB-AM bao gồm việc nhân (trộn) tín hiệu đã điều chế với một tín hiệu hình sin có cùng tần số và pha với sóng mang, sau đó cho sản phẩm của việc nhân qua một bộ lọc thông thấp. Bộ dao động tạo ra tại máy thu tín hiệu hình sin có yêu cầu như trên được gọi là bộ dao động nội.

### 3.3.1 Giải điều chế DSB-AM

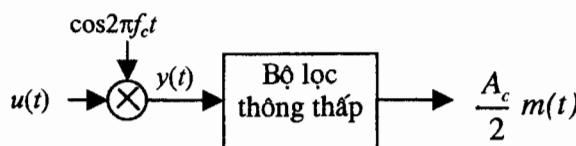
Trong trường hợp DSB, tín hiệu sau điều chế được cho bởi  $A_c m(t) \cos 2\pi f_c t$  mà khi nó được nhân với  $\cos 2\pi f_c t$  hay được trộn với  $\cos 2\pi f_c t$  sẽ sinh ra

$$y(t) = A_c m(t) \cos 2\pi f_c t \cos 2\pi f_c t = \frac{A_c}{2} m(t) + \frac{A_c}{2} m(t) \cos 4\pi f_c t \quad (3.3.1)$$

ở đây  $y(t)$  ký hiệu cho sản phẩm lối ra mạch trộn và biến đổi Fourier của nó được cho bởi

$$Y(f) = \frac{A_c}{2} M(f) + \frac{A_c}{4} M(f - 2f_c) + \frac{A_c}{4} M(f + 2f_c) \quad (3.3.2)$$

Như thấy được, lối ra mạch trộn có một thành phần thông thấp ( $A_c/2)M(f)$  và các thành phần cao tần nằm quanh  $\pm 2f_c$ . Khi  $y(t)$  qua một bộ lọc thông thấp với độ rộng bằng  $W$ , các thành phần cao tần sẽ bị lọc bỏ còn thành phần thông thấp ( $A_c/2)m(t)$ , tỷ lệ với tín hiệu bản tin, sẽ được giải điều chế. Quá trình này được trình bày trên hình 3.9.



Hình 3.9 Giải điều chế các tín hiệu DSB-AM.

## BÀI TẬP MINH HỌA

**Bài tập minh họa 3.5 [Giải điều chế DSB-AM]** Tín hiệu bản tin  $m(t)$  được xác định theo

$$m(t) = \begin{cases} 1, & 0 \leq t < t_0 / 3 \\ -2, & t_0 / 3 \leq t < 2t_0 / 3 \\ 0, & \text{với các giá trị khác của } t \end{cases}$$

Bản tin này điều chế DSB-AM một sóng mang  $c(t) = \cos 2\pi f_c t$  và sinh ra tín hiệu sau điều chế  $u(t)$ . Giả sử rằng  $t_0 = 0.15$  s và  $f_c = 250$  Hz.

1. Hãy viết biểu thức đối với  $u(t)$ .
2. Hãy tìm các phổ của  $m(t)$  và  $u(t)$ .
3. Hãy giải điều chế tín hiệu sau điều chế  $u(t)$  và khôi phục  $m(t)$ . Hãy vẽ các kết quả trên các miền thời gian và tần số.

### Lời giải

1, 2. Hai phần đầu của bài tập này thì giống như hai phần đầu của bài tập minh họa 3.1 và ta chỉ lặp lại ở đây các kết quả:

$$u(t) = \left[ \Pi\left(\frac{t - 0.025}{0.05}\right) - 2\Pi\left(\frac{t - 0.075}{0.05}\right) \right] \cos(500\pi t)$$

và

$$\begin{aligned} F[m(t)] &= \frac{t_0}{3} e^{-j\pi f t_0 / 3} \sin c\left(\frac{t_0 f}{3}\right) - 2 \frac{t_0}{3} e^{-j\pi f t_0} \sin c\left(\frac{t_0 f}{3}\right) = \\ &= \frac{t_0}{3} e^{-j\pi f t_0 / 3} \sin c\left(\frac{t_0 f}{3}\right) (1 - 2e^{-j2\pi f t_0 / 3}) \\ &= 0.05 e^{-0.05j\pi f} \text{sinc}(0.05f)(1 - 2e^{-0.1j\pi f}) \end{aligned}$$

Do đó,

$$\begin{aligned} U(f) &= 0.025 e^{-0.05j\pi(f-250)} \sin c[(0.05(f-250))(1 - 2e^{-0.1j\pi(f-250)})] + \\ &\quad + 0.025 e^{-0.05j\pi(f+250)} \sin c[(0.05(f+250))(1 - 2e^{-0.1j\pi(f+250)})] \end{aligned}$$

3. Để giải điều chế, chúng ta nhân  $u(t)$  với  $\cos 2\pi f_c t$  để nhận được tín hiệu lối ra mạch trộn  $y(t)$ :

$$y(t) = u(t) \cos(2\pi f_c t) = \left[ \Pi\left(\frac{t - 0.025}{0.05}\right) - 2\Pi\left(\frac{t - 0.075}{0.05}\right) \right] \cos^2(500\pi t)$$

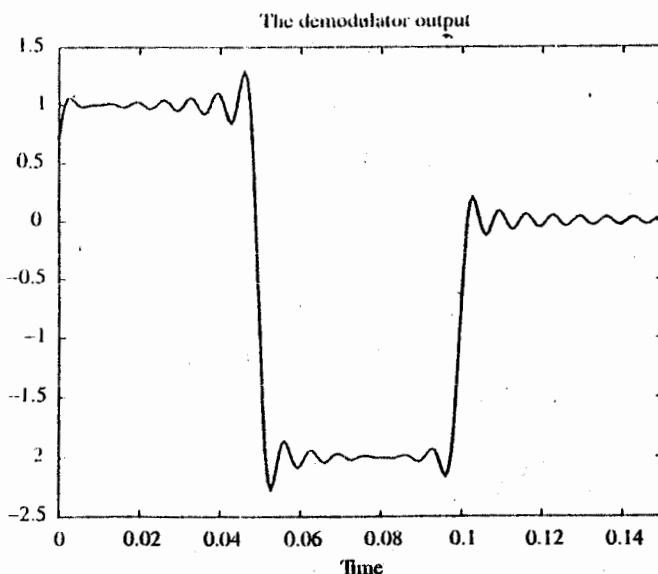
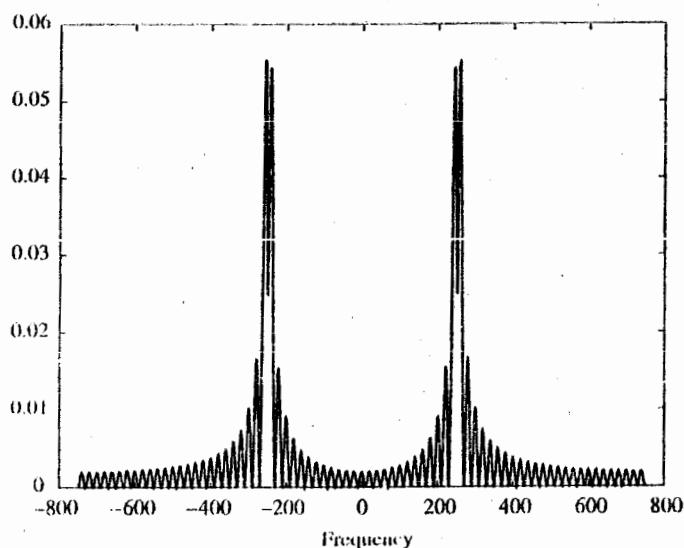
$$= \frac{1}{2} \left[ \Pi\left(\frac{t - 0.025}{0.05}\right) - 2\Pi\left(\frac{t - 0.075}{0.05}\right) \right] + \frac{1}{2} \left[ \Pi\left(\frac{t - 0.025}{0.05}\right) - 2\Pi\left(\frac{t - 0.075}{0.05}\right) \right] \cos(1000\pi t)$$

mà biến đổi Fourier của nó được cho bởi

$$Y(f) = 0.025e^{-0.05j\pi f} \sin c(0.05f)(1 - 2e^{-0.1j\pi f}) +$$

$$+ 0.0125e^{-0.05j\pi(f-500)} \sin c[0.05(f-500)](1 - 2e^{-0.1j\pi(f-500)}) +$$

$$+ 0.0125e^{-0.05j\pi(f+500)} \sin c[0.05(f+500)](1 - 2e^{-0.1j\pi(f+500)})$$



Hình 3.10 Các phổ của tín hiệu sau điều chế và tín hiệu lối ra mạch trộn trong bài tập minh họa 3.5.

trong đó số hạng thứ nhất tương ứng với tín hiệu bản tin còn hai số hạng sau tương ứng với các thành phần cao tần tại tần số gấp hai lần tần số sóng mang. Chúng ta thấy rằng việc lọc lấy thành phần thứ nhất sẽ thu được tín hiệu bản tin gốc (được nâng lên với một hằng số tỷ lệ). Một đồ thị về các biên độ của  $U(f)$  và  $Y(f)$  được trình bày trên hình 3.10.

Như đã thấy, phổ của tín hiệu lõi ra mạch trộn có một thành phần thông thấp mà nó hoàn toàn tương tự như phổ của tín hiệu bản tin, ngoại trừ mỗi một điều là có hệ số  $1/2$ , và một thành phần thông dải nằm tại  $\pm 2f_c$  (trong trường hợp này là 500 Hz). Sử dụng một bộ lọc thông thấp chúng ta có thể tách một cách đơn giản thành phần thông thấp khỏi thành phần thông dải. Để khôi phục lại tín hiệu bản tin  $m(t)$ , chúng ta cho  $y(t)$  qua một bộ lọc thông thấp có độ rộng băng 150 Hz. Việc chọn độ rộng băng của mạch lọc ở đây thì tương đối tuỳ tiện một chút do tín hiệu bản tin không phải là tín hiệu có băng hạn chế một cách chặt chẽ. Đối với một tín hiệu bản tin có băng hạn chế một cách chặt chẽ thì việc chọn lựa thích hợp đối với độ rộng băng của bộ lọc thông thấp sẽ nên là  $W$  là độ rộng băng của tín hiệu bản tin. Do vậy, bộ lọc thông thấp lý tưởng được áp dụng ở đây có một đặc tính

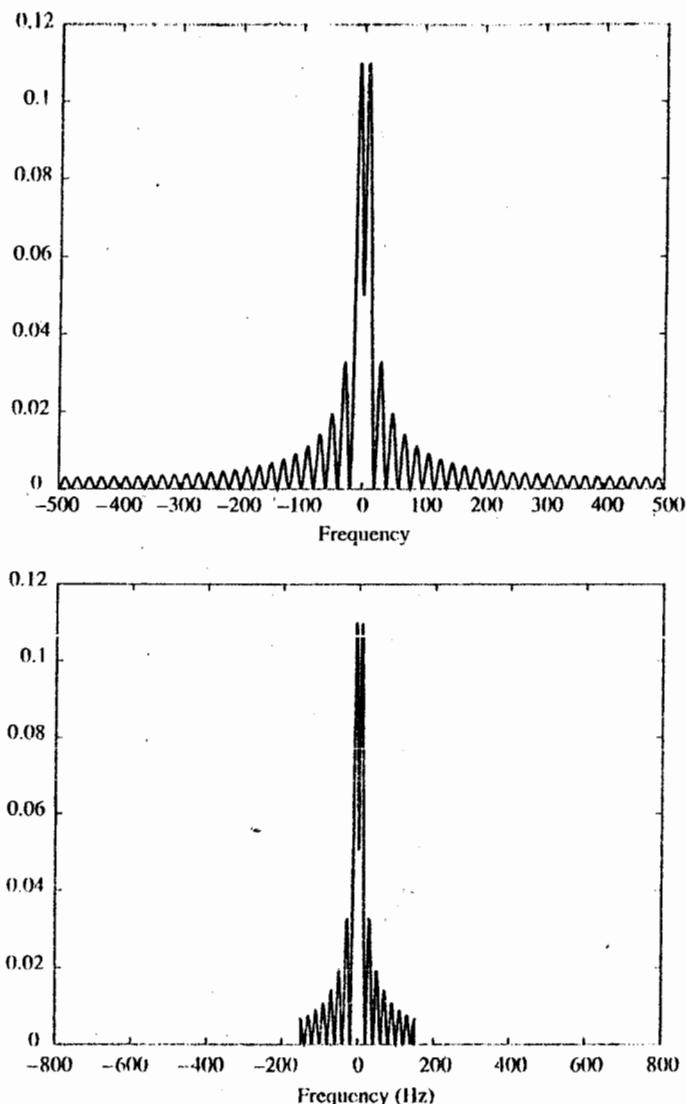
$$H(f) = \begin{cases} 1, & |f| \leq 150 \\ 0, & \text{với mọi tần số } f \text{ khác} \end{cases}$$

Một sự so sánh các phổ của  $m(t)$  và tín hiệu lõi ra bộ giải điều chế được thể hiện trên hình 3.11 còn việc so sánh trên miền thời gian được thể hiện trên hình 3.12.

MATLAB script cho bài tập này được cho dưới đây.

### m-file

```
% dsb_dem.m
% Matlab demonstration script for DSB-AM demodulation. The message signal
% is +1 for 0 < t < t0/3, -2 for t0/3 < t < 2t0/3 and zero otherwise.
echo on
t0=.15; % signal duration
ts=1/1500; % sampling interval
fc=250; % carrier frequency
fs=1/ts; % sampling frequency
t=[0:ts:t0]; % time vector
```

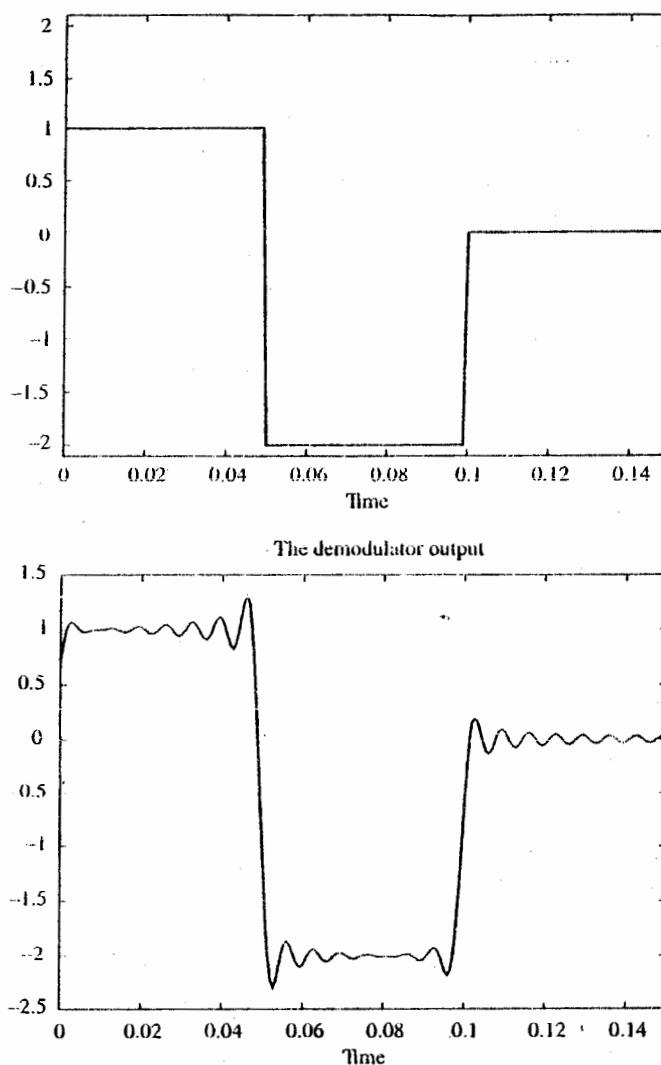


Hình 3.11 Các phổ của các tín hiệu bản tin và sau giải điều chế trong bài tập minh họa 3.5.

```

df=0.3; % desired frequency resolution
% message signal
m=[ones(1,t0/(3*ts)),-2*ones(1,t0/(3*ts)),zeros(1,t0/(3*ts)+1)];
c=cos(2*pi*fc.*t); % carrier signal
u=m.*c; % modulated signal
y=u.*c; % mixing
[M,m,df1]=fftseq(m,ts,df); % Fourier transform
M=M/fs; % scaling
[U,u,df1]=fftseq(u,ts,df); % Fourier transform

```



Hình 3.12 Tín hiệu bản tin và tín hiệu lối ra bộ giải điều chế trong bài tập minh họa 3.5.

```

U=U/fs; % scaling
[Y,y,df1]=fftseq(y,ts,df); % Fourier transform
Y=Y/fs; % scaling
f_cutoff=150; % cutoff freq. of the filter
n_cutoff=floor(150/df1); % design the filter
f=[0:df1:df1*(length(y)-1)]-fs/2;
H=zeros(size(f));
H(1:n_cutoff)=2*ones(1,n_cutoff);
H(length(f)-n_cutoff+1:length(f))=2*ones(1,n_cutoff);

```

```
DEM=H.*Y; % spectrum of the filter output
dem=real(ifft(DEM)); % filter output
pause % Press a key to see the effect of mixing
clf
subplot(3,1,1)
plot(f,fftshift(abs(M)))
title('Spectrum of the the Message Signal')
xlabel('Frequency')
subplot(3,1,2)
plot(f,fftshift(abs(U)))
title('Spectrum of the Modulated Signal')
xlabel('Frequency')
subplot(3,1,3)
plot(f,fftshift(abs(Y)))
title('Spectrum of the Mixer Output')
xlabel('Frequency')
pause % Press a key to see the effect of filtering on the mixer output
clf
subplot(3,1,1)
plot(f,fftshift(abs(Y)))
title('Spectrum of the Mixer Output')
xlabel('Frequency')
subplot(3,1,2)
plot(f,fftshift(abs(H)))
title('Lowpass Filter Characteristics')
xlabel('Frequency')
subplot(3,1,3)
plot(f,fftshift(abs(DEM)))
title('Spectrum of the Demodulator output')
xlabel('Frequency')
pause % Press a key to compare the spectra of the message an the received signal
clf
subplot(2,1,1)
plot(f,fftshift(abs(M)))
title('Spectrum of the Message Signal')
xlabel('Frequency')
subplot(2,1,2)
plot(f,fftshift(abs(DEM)))
title('Spectrum of the Demodulator Output')
xlabel('Frequency')
pause % Press a key to see the message and the demodulator output signals
subplot(2,1,1)
plot(t,m(1:length(t)))
```

```

title('The Message Signal')
xlabel('Time')
subplot(2,1,2)
plot(t,dem(1:length(t)))
title('The Demodulator Output')
xlabel('Time')

```

## BÀI TẬP MINH HỌA

**Bài tập minh họa 3.6 [Tác động của sai pha trong giải điều chế DSB-AM]** Trong giải điều chế các tín hiệu DSB-AM chúng ta đã giả sử rằng pha của bộ dao động nội thì bằng với pha của sóng mang. Nếu không đúng như thế, nghĩa là tồn tại một lượng dịch pha  $\phi$  giữa bộ dao động nội và sóng mang, thì liệu quá trình giải điều chế sẽ thay đổi như thế nào?

### Lời giải

Trong trường hợp này chúng ta có  $u(t) = A_c m(t) \cos(2\pi f_c t)$ , còn bộ dao động nội tạo ra một tín hiệu hình sin được cho bởi  $\cos(2\pi f_c t + \phi)$ . Việc trộn hai tín hiệu này cho chúng ta

$$y(t) = A_c m(t) \cos(2\pi f_c t) \times \cos(2\pi f_c t + \phi) \quad (3.3.3)$$

$$= \frac{A_c}{2} m(t) \cos(\phi) + \frac{A_c}{2} m(t) \cos(4\pi f_c t + \phi) \quad (3.3.4)$$

Như trước đây, có hai số hạng có mặt trong tín hiệu lối ra mạch trộn. Số hạng thông dải có thể lọc bỏ được bằng một bộ lọc thông thấp. Số hạng thông thấp,  $(A_c/2)m(t)\cos(\phi)$ , tuy vậy lại phụ thuộc vào  $\phi$ . Công suất trong số hạng thông thấp được cho bởi

$$P_{dem} = \frac{A_c^2}{4} P_m \cos^2 \phi \quad (3.3.5)$$

trong đó  $P_m$  ký hiệu cho công suất tín hiệu bản tin. Chúng ta do vậy có thể thấy rằng trong trường hợp này chúng ta có thể khôi phục lại tín hiệu bản tin về căn bản là không có méo song chúng ta sẽ phải chịu một lượng tổn hao công suất là  $\cos^2(\phi)$ . Đối với  $\phi = \pi/4$  thì lượng tổn hao này là 3 dB còn với  $\phi = \pi/2$  thì chẳng có gì khôi phục lại được trong quá trình giải điều chế.

### 3.3.2 GIẢI ĐIỀU CHẾ SSB-AM

Quá trình giải điều chế các tín hiệu SSB-AM giống về cơ bản như quá trình giải điều chế đối với các tín hiệu DSB-AM, nghĩa là trước hết là trộn, sau đó là lọc thông thấp. Trong trường hợp này

$$u(t) = \frac{A_c}{2} m(t) \cos(2\pi f_c t) \mp \frac{A_c}{2} \hat{m}(t) \sin(2\pi f_c t) \quad (3.3.6)$$

trong đó dấu trừ ứng với USSB còn dấu cộng ứng với LSSB. Trộn  $u(t)$  với tín hiệu ra của bộ dao động nội, chúng ta nhận được

$$\begin{aligned} y(t) &= \frac{A_c}{2} m(t) \cos^2(2\pi f_c t) \mp \frac{A_c}{2} \hat{m}(t) \cos(2\pi f_c t) \sin(2\pi f_c t) \\ &= \frac{A_c}{4} m(t) + \frac{A_c}{4} m(t) \cos(4\pi f_c t) \mp \frac{A_c}{4} \hat{m}(t) \sin(4\pi f_c t) \end{aligned} \quad (3.3.7)$$

mà nó chứa các thành phần thông dài tại  $\pm 2f_c$  và một thành phần thông thấp tỷ lệ với tín hiệu bản tin. Thành phần thông thấp có thể lọc lấy ra được nhờ một bộ lọc thông thấp nhằm khôi phục lại tín hiệu bản tin. Quá trình này đối với trường hợp USSB-AM được vẽ trên hình 3.13.

### BÀI TẬP MINH HỌA

**Bài tập minh họa 3.7 [Thí dụ LSSB-AM]** Trong một hệ thống điều chế USSB-AM, nếu tín hiệu bản tin là

$$m(t) = \begin{cases} 1, & 0 \leq t < t_0 / 3 \\ -2, & t_0 / 3 \leq t < 2t_0 / 3 \\ 0, & \text{với các giá trị khác của } t \end{cases}$$

với  $t_0 = 0.15$  s, còn sóng mang có tần số 250 Hz, hãy tìm  $U(f)$  và  $Y(f)$  và so sánh tín hiệu sau giải điều chế với tín hiệu bản tin.

#### Lời giải

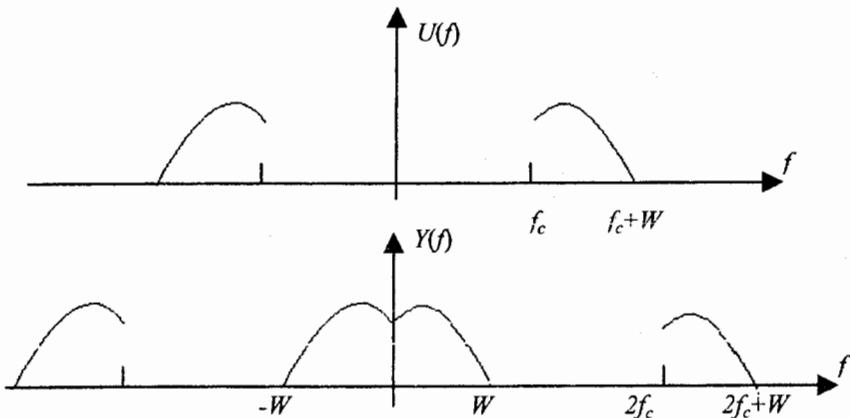
Tín hiệu sau điều chế và phổ của nó được cho trong bài tập minh họa 3.4. Biểu thức đối với  $U(f)$  được cho bởi

$$U(f) = \begin{cases} 0.025e^{-0.05j\pi(f-250)} \sin c(f-250)(1-2e^{-0.1j\pi(f-250)}) + \\ + 0.025e^{-0.05j\pi(f+250)} \sin c(f+250)(1-2e^{-0.1j\pi(f+250)}), & |f| \leq f_c \\ 0, & \text{với các giá trị khác của } f \end{cases}$$

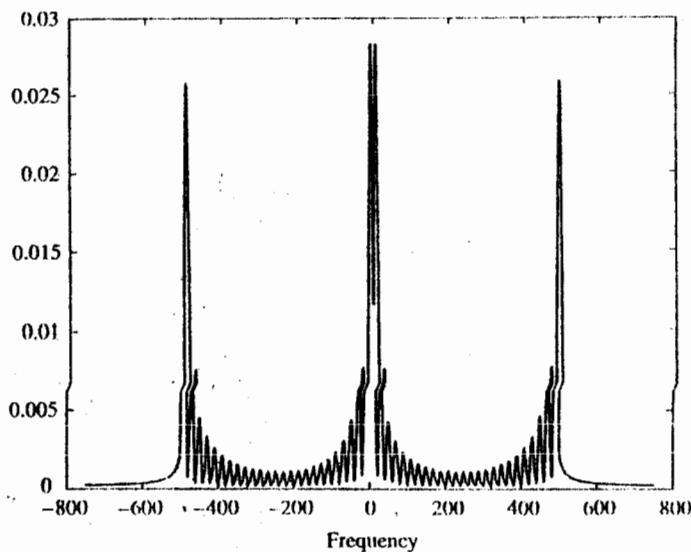
và

$$Y(f) = \frac{1}{2}U(f - f_c) + \frac{1}{2}U(f + f_c)$$

$$\approx \begin{cases} 0.0125e^{-0.05j\pi f} \sin c(0.05f)(1 - 2e^{-0.1j\pi f}), & |f| \leq f_c \\ 0.0125e^{-0.05j\pi(f-500)} \sin c[0.05(f-500)](1 - 2e^{-0.1j\pi(f-500)}), & f_c \leq f \leq 2f_c \\ 0.0125e^{-0.05j\pi(f+500)} \sin c[0.05(f+500)](1 - 2e^{-0.1j\pi(f+500)}), & -2f_c \leq f \leq -f_c \\ 0, & \text{với các giá trị khác của } f \end{cases}$$

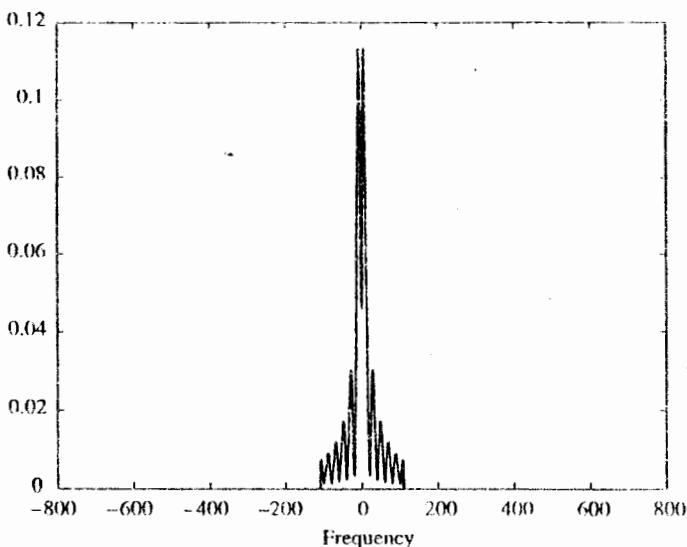


Hình 3.13 Giải điều chế các tín hiệu USSB-AM.



Hình 3.14 Phổ biên độ của tín hiệu lõi ra bộ trộn trong bài tập minh họa 3.7.

Một đồ thị của  $Y(f)$  được cho trên hình 3.14. Tín hiệu  $y(t)$  được lọc bằng một bộ lọc thông thấp với tần số cắt 150 Hz; phổ của tín hiệu lõi ra được thể hiện trên hình 3.15. Trên hình 3.16, tín hiệu bản tin gốc được so sánh với tín hiệu sau giải điều chế.



Hình 3.15 Phổ của tín hiệu lõi ra bộ giải điều chế trong bài tập minh họa 3.7.

Dưới đây là MATLAB script cho bài tập này.

#### m-file

```
% lssb_dem.m
% Matlab demonstration script for LSSB-AM demodulation. The message signal
% is +1 for  $0 < t < t_0/3$ , -2 for  $t_0/3 < t < 2t_0/3$  and zero otherwise.
echo on
t0=15;                                % signal duration
ts=1/1500;                             % sampling interval
fc=250;                                 % carrier frequency
fs=1/ts;                                % sampling frequency
df=0.25;                                % desired freq.resolution
t=[0:ts:t0];                            % time vector
% the message vector
m=[ones(1,t0/(3*ts)),-2*ones(1,t0/(3*ts)),zeros(1,t0/(3*ts)+1)];
c=cos(2*pi*fc.*t);                      % carrier vector
udsb=m.*c;                              % DSB modulated signal
[UDSB,udsb,df1]=fftseq(udsb,ts,df);    % Fourier transform
UDSB=UDSB/fs;                           % scaling
n2=ceil(fc/df1);                        % location of carrier in freq. vector
% remove the upper sideband from DSB
UDSB(n2:length(UDSB)-n2)=zeros(size(UDSB(n2:length(UDSB)-n2)));
ULSSB=UDSB;                            % generate LSSB-AM spectrum
[M,m,df1]=fftseq(m,ts,df);              % spectrum of the message signal
M=M/fs;                                 % scaling
```

```

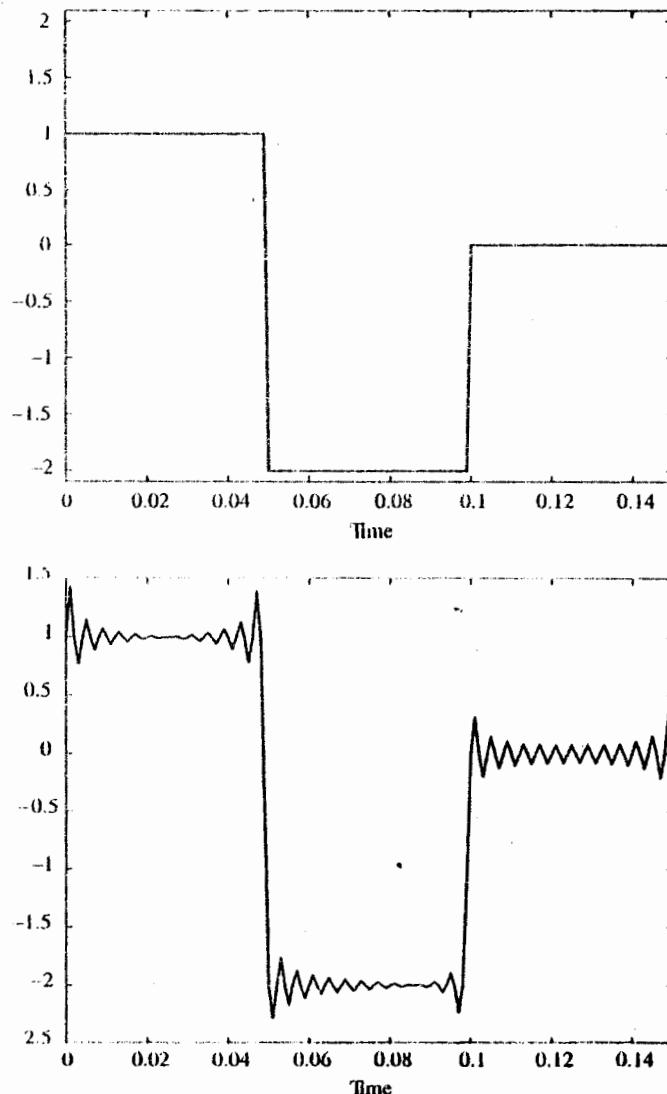
f=[0:df1:df1*(length(M)-1)]-fs/2;           % frequency vector
u=real(ifft(ULSSB))*fs;                      % generate LSSB signal from spectrum
% mixing
y=u.*cos(2*pi*fc*[0:ts:ts*(length(u)-1)]);
[Y,y,df1]=fftseq(y,ts,df);                   % spectrum of the output of the mixer
Y=Y/fs;                                         % scaling
f_cutoff=150;                                    % choose the cutoff freq. of the filter
n_cutoff=floor(150/df);                         % design the filter
H=zeros(size(f));
H(1:n_cutoff)=4*ones(1,n_cutoff);
% spectrum of the filter output
H(length(f)-n_cutoff+1:length(f))=4*ones(1,n_cutoff);
DEM=H.*Y;                                         % spectrum of the filter output
dem=real(ifft(DEM))*fs;                         % filter output
pause % Press a key to see the effect of mixing
clf
subplot(3,1,1)
plot(f,fftshift(abs(M)))
title('Spectrum of the Message Signal')
xlabel('Frequency')
subplot(3,1,2)
plot(f,fftshift(abs(ULSSB)))
title('Spectrum of the Modulated Signal')
xlabel('Frequency')
subplot(3,1,3)
plot(f,fftshift(abs(Y)))
title('Spectrum of the Mixer Output')
xlabel('Frequency')
pause % Press a key to see the effect of filtering on the mixer output
clf
subplot(3,1,1)
plot(f,fftshift(abs(Y)))
title('Spectrum of the Mixer Output')
xlabel('Frequency')
subplot(3,1,2)
plot(f,fftshift(abs(H)))
title('Lowpass Filter Characteristics')
xlabel('Frequency')
subplot(3,1,3)
plot(f,fftshift(abs(DEM)))
title('Spectrum of the Demodulator output')
xlabel('Frequency')
pause % Press a key to see the message and the demodulator output signals

```

```

subplot(2,1,1)
plot(t,m(1:length(t)))
title('The Message Signal')
xlabel('Time')
subplot(2,1,2)
plot(t,dem(1:length(t)))
title('The Demodulator Output')
xlabel('Time')

```



Hình 3.16 Tín hiệu bản tin và tín hiệu lối ra bộ giải điều chế trong bài tập minh họa 3.7.

### BÀI TẬP MINH HỌA

#### Bài tập minh họa 3.8 [Tác động của sai pha tới SSB-AM] Tác động của sai

pha tới SSB-AM là gì?

### Lời giải

Giả sử rằng bộ dao động nội tạo ra một tín hiệu hình sin có pha dịch so với sóng mang một lượng là  $\phi$ , chúng ta có

$$\begin{aligned} y(t) &= u(t) \cos(2\pi f_c t + \phi) = \left[ \frac{A_c}{2} m(t) \cos(2\pi f_c t) \mp \frac{A_c}{2} \hat{m}(t) \sin(2\pi f_c t) \right] \cos(2\pi f_c t + \phi) \\ &= \frac{A_c}{4} m(t) \cos \phi \pm \frac{A_c}{4} \hat{m}(t) \sin \phi + \text{các số hạng tần số cao} \end{aligned} \quad (3.3.8)$$

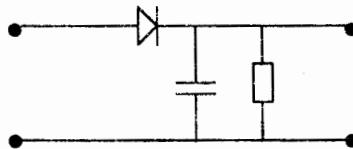
Như đã thấy, không như trường hợp DSB-AM, tác động của sai lệch pha ở đây không đơn thuần làm tổn hao tín hiệu sau giải điều chế. Ở đây tín hiệu đã giải điều chế bị tổn hao đi một lượng  $\cos \phi$  cũng như bị méo do cộng thêm số hạng  $\pm (A_c / 4) \hat{m}(t) \sin \phi$ . Trong trường hợp đặc biệt khi  $\phi = \pi/2$ , chính biến đổi Hilbert của tín hiệu bản tin sẽ được giải điều chế chứ không phải là tín hiệu mong muốn.

### 3.3.3 GIẢI ĐIỀU CHẾ AM TRUYỀN THỐNG

Chúng ta đã thấy rằng xét về công suất và SNR thì AM truyền thống kém DSB-AM và SSB-AM. Lý do là ở chỗ một phần thường là lớn công suất của tín hiệu sau điều chế nằm trong thành phần sóng mang không mang thông tin. Vai trò của thành phần sóng mang là làm cho việc giải điều chế AM truyền thống trở nên dễ dàng hơn thông qua tách sóng đường bao, chứ không phải là giải điều chế kết hợp vốn đỗi hỏi đối với DSB-AM và SSB-AM. Do đó, việc giải điều chế các tín hiệu AM truyền thống thì ít phức tạp hơn một cách đáng kể so với giải điều chế các tín hiệu DSB-AM và SSB-AM. Chính vì thế sơ đồ điều chế này được sử dụng rộng rãi trong phát quảng bá, trong đó chỉ có một máy phát và vô số máy thu với giá thành cần phải giữ ở mức thấp. Trong tách sóng đường bao, đường bao của tín hiệu sau điều chế được tách bằng một mạch đơn giản bao gồm một đi-ốt, một điện trở và một tụ điện như được cho thấy trên hình vẽ 3.17.

Về mặt toán học mà nói, bộ tách sóng đường bao tạo ra đường bao của tín hiệu AM truyền thống, mà đường bao đó là

$$V(t) = |1 + am_n(t)| \quad (3.3.9)$$



Hình 3.17 Một bộ tách đường bao đơn giản.

Do  $1 + m_n(t) \geq 0$ , chúng ta kết luận được rằng

$$V(t) = 1 + am_n(t) \quad (3.3.10)$$

trong đó  $m_n(t)$  tỷ lệ với tín hiệu bản tin  $m(t)$  còn 1 thì ứng với thành phần sóng mang mà nó có thể phân tách ra được bằng một mạch ngăn tín hiệu một chiều. Như đã thấy trong thủ tục nói trên, không cần phải có một hiểu biết gì về  $\phi$  là pha của tín hiệu sóng mang. Đó là lý do tại sao sơ đồ giải điều chế như thế lại được gọi là giải điều chế không kết hợp (*noncoherent demodulation*) hay giải điều chế không đồng bộ (*asynchronous demodulation*). Ta hãy nhớ lại từ chương 1 rằng đường bao của một tín hiệu thông dài có thể biểu diễn được như biên độ của tín hiệu thông thấp tương đương của nó. Như vậy, nếu  $u(t)$  là tín hiệu thông dài với tần số trung tâm là  $f_c$  còn tương đương thông thấp của  $u(t)$  ký hiệu là  $u_r(t)$  thì đường bao của  $u(t)$ , được ký hiệu là  $V(t)$ , có thể biểu diễn được theo

$$V(t) = \sqrt{u_r^2(t) + u_s^2} = \sqrt{u_c^2(t) + u_s^2(t)} \quad (3.3.11)$$

trong đó  $u_r(t)$  và  $u_s(t)$  biểu thị các thành phần cùng pha và vuông góc của tín hiệu thông dài  $u(t)$ . Do vậy, để nhận được đường bao này thì việc nhận được tương đương thông thấp của tín hiệu thông dài là đủ. Đường bao đơn giản là biên độ của tín hiệu tương đương thông thấp của tín hiệu thông dài.

## BÀI TẬP MINH HỌA

### Bài tập minh họa 3.9 [Tách sóng đường bao] Tín hiệu bản tin

$$m(t) = \begin{cases} 1, & 0 \leq t < t_0 / 3 \\ -2, & t_0 / 3 \leq t < 2t_0 / 3 \\ 0, & \text{với các giá trị khác của } t \end{cases}$$

điều chế sóng mang  $c(t) = \cos(2\pi f_c t)$  bằng một sơ đồ điều chế AM truyền thống. Giả sử rằng  $f_c = 250$  Hz còn  $t_0 = 0.15$  s và chỉ số điều chế  $a = 0.85$ .

1. Sử dụng tách sóng đường bao, hãy giải điều chế lấy tín hiệu bản tin.

2. Nếu tín hiệu bản tin là tuần hoàn với chu kỳ bằng  $t_0$  và nếu một quá trình AWGN (*Additive White Gaussian Noise: Tạp âm trắng cộng Gauss, hay tạp âm trắng chuẩn cộng tính*) được cộng vào tín hiệu sau điều chế sao cho công suất của quá trình tạp âm bằng một phần trăm công suất tín hiệu đã điều chế, hãy sử dụng một bộ tách sóng đường bao để giải điều chế tín hiệu thu được. Hãy so sánh trường hợp này với trường hợp khi không có tạp âm.

### Lời giải

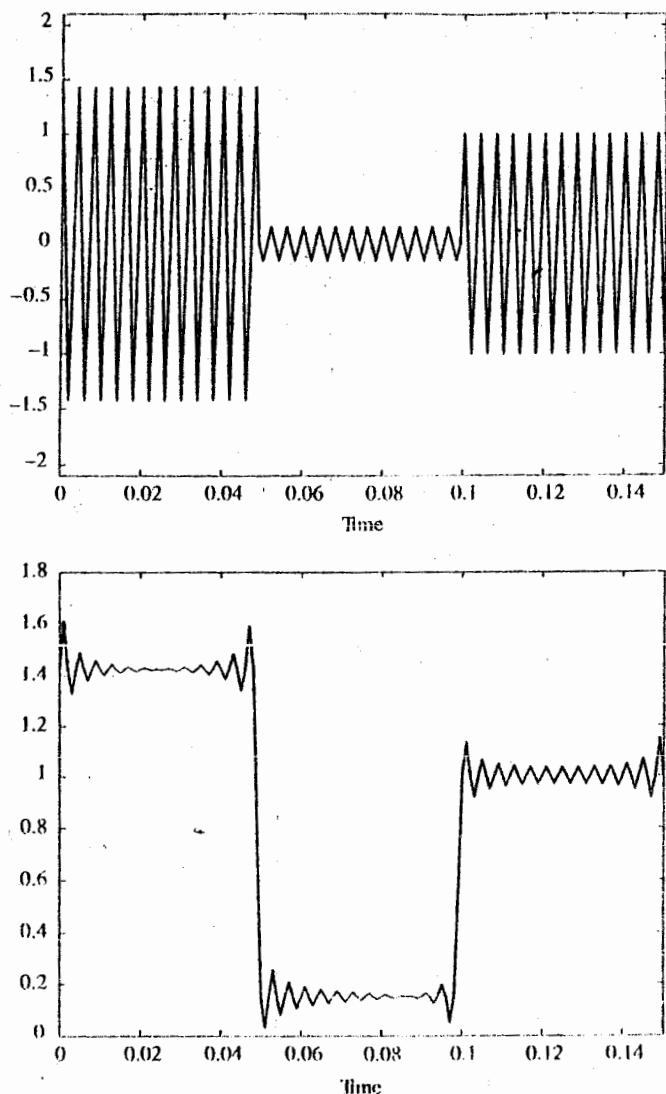
1. Như trong bài tập minh họa 3.3, chúng ta có

$$u(t) = [1 + 0.85 \frac{m(t)}{2}] \cos(2\pi f_c t) = [1 + 0.425 \Pi(\frac{1 - 0.025}{0.05}) - 0.85 \Pi(\frac{1 - 0.075}{0.05})] \cos(500\pi t)$$

Nếu một bộ tách sóng đường bao được sử dụng để giải điều chế tín hiệu và sóng mang được loại bỏ bằng một mạch ngăn thành phần một chiều, thì bản tin gốc  $m(t)$  được khôi phục lại. Hãy chú ý rằng một điểm cốt yếu trong việc khôi phục  $m(t)$  là đối với mọi giá trị của  $t$  thì biểu thức  $1+am_n(t)$  đều dương; do vậy, đường bao của tín hiệu  $[1+am_n(t)]\cos(2\pi f_c t)$ , mà nó là  $V(t) = |1+am_n(t)|$ , bằng với  $1+am_n(t)$ , từ đó  $m(t)$  có thể dễ dàng khôi phục lại được. Các đồ thị của tín hiệu đã điều chế AM truyền thống và đường bao của nó khi đã được tách bằng bộ tách sóng đường bao được trình bày trên hình 3.18.

Sau khi bộ tách sóng đường bao tách lấy đường bao của tín hiệu đã được điều chế, thành phần một chiều của tín hiệu được loại bỏ và tín hiệu được nhân tỷ lệ để tạo nên tín hiệu đầu ra của bộ giải điều chế. Các đồ thị của tín hiệu bản tin gốc và tín hiệu lõi ra bộ giải điều chế được thể hiện trên hình 3.19.

2. Khi có tạp âm, sẽ có một lượng méo nào đó do tạp âm. Tín hiệu thu được và đường bao của nó được cho thấy trên hình vẽ 3.20. Trên hình 3.21, tín hiệu bản tin và tín hiệu sau giải điều chế được so sánh trong trường hợp này.

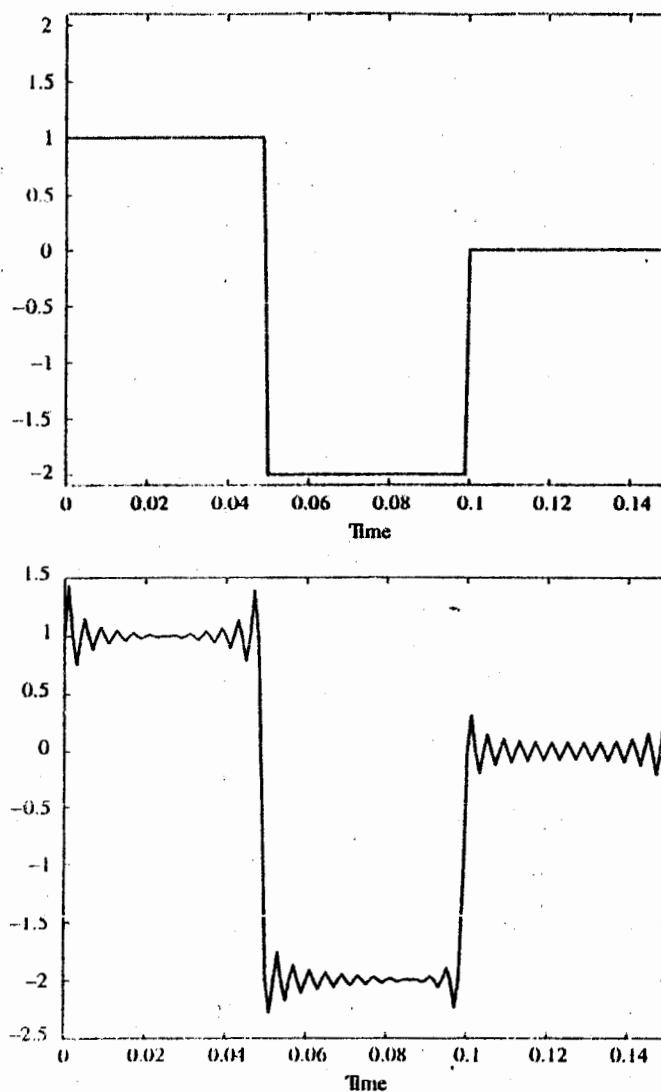


Hình 3.18 Tín hiệu sau điều chế AM truyền thống và đường bao của nó.

MATLAB script cho bài tập này như sau.

#### m-file

```
% am-dem.m
% Matlab demonstration script for envelope detection. The message signal
% is +1 for 0 < t < t0/3, -2 for t0/3 < t < 2t0/3 and zero otherwise.
echo on
t0=.15;                      %,signal duration
ts=0.001;                     % sampling interval
```

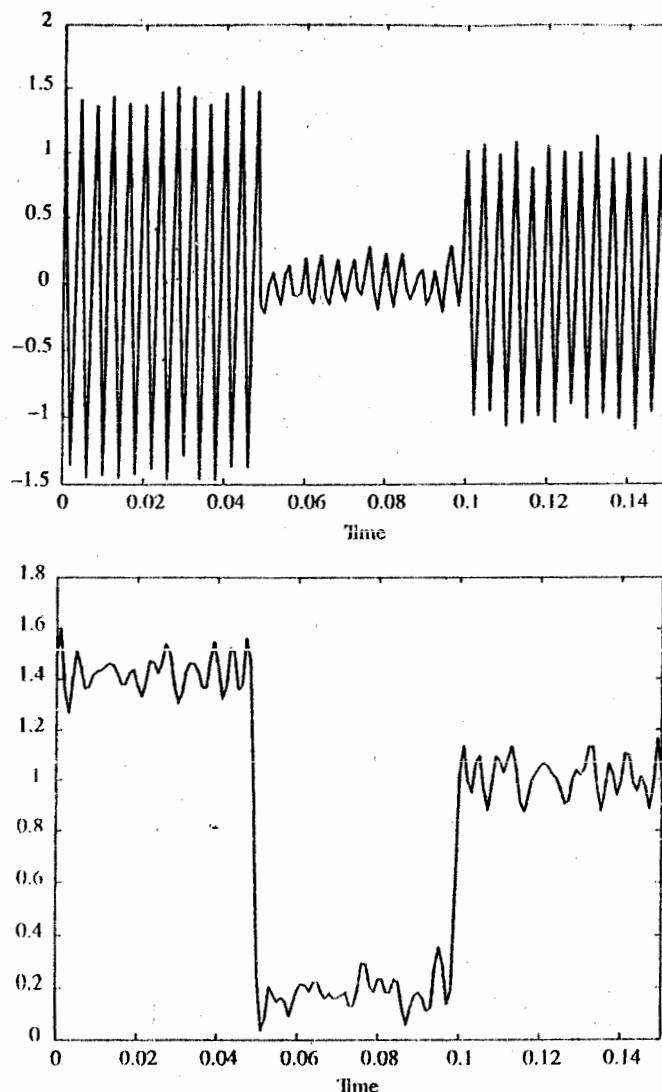


Hình 3.19 Tín hiệu bản tin và tín hiệu sau giải điều chế khi không có tạp nhiễu.

```

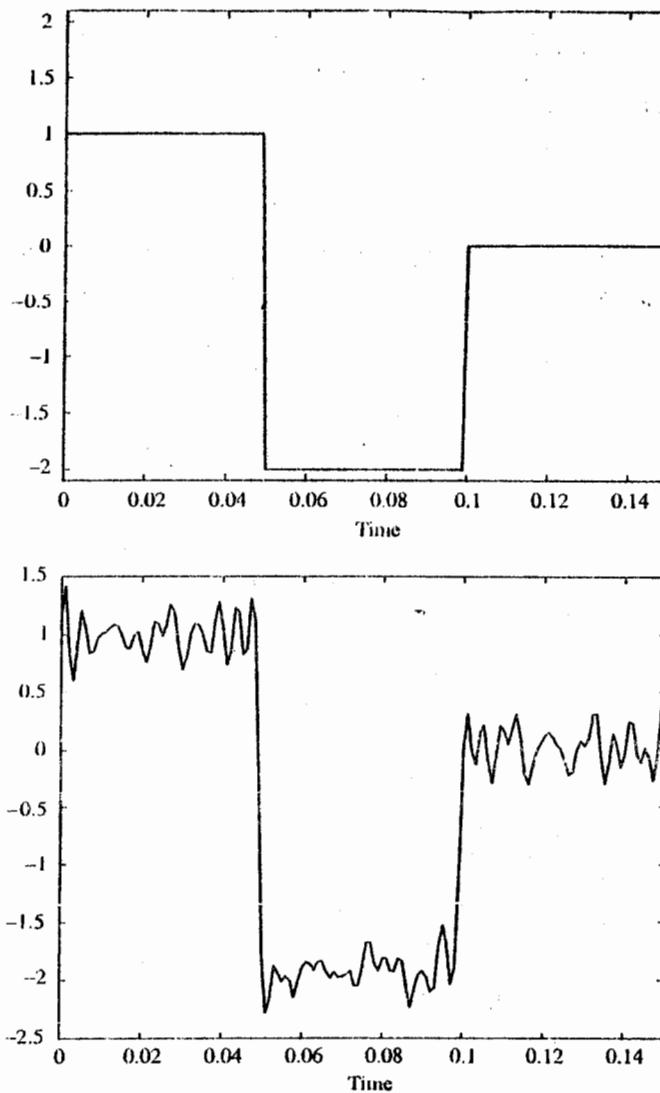
fc=250; % carrier frequency
a=0.85; % Modulation index
fs=1/ts; % sampling frequency
t=[0:ts:t0]; % time vector
df=0.25; % required frequency resolution
% message signal
m=[ones(1,t0/(3*ts)),-2*ones(1,t0/(3*ts)),zeros(1,t0/(3*ts)+1)];
c=cos(2*pi*fc.*t); % carrier signal
m_n=m/max(abs(m)); % normalized message signal

```



Hình 3.20 Tín hiệu thu được và đường bao của nó khi có tạp nhiễu.

```
[M,m,df1]=fftseq(m,ts,df);          % Fourier transform
f=[0:df1:df1*(length(m)-1)]-fs/2;    % frequency vector
u=(1+a*m_n).*c;                      % modulated signal
[U,u,df1]=fftseq(u,ts,df);           % Fourier transform
env=env_phas(u);                     % find the envelope
dem1=2*(env-1)/a;                   % remove dc and rescale
signal_power=spower(u(1:length(t))); % power in modulated signal
noise_power=signal_power/100;         % noise power
noise_std=sqrt(noise_power);          % noise standard deviation
```



Hình 3.21 Tín hiệu bản tin và tín hiệu sau giải điều chế khi có tạp nhiễu.

```

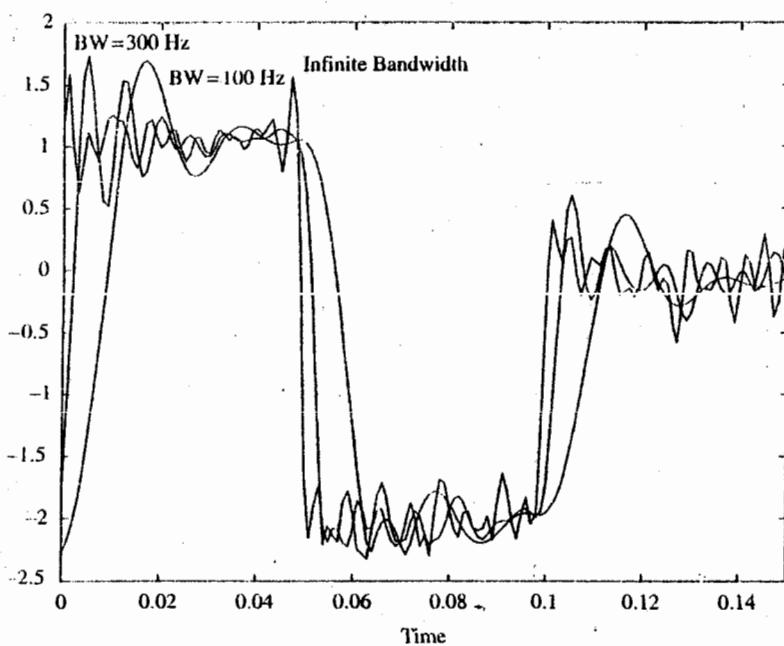
noise=noise_std*randn(1,length(u)); % generate noise
r=u+noise; % add noise to the modulated signal
[R,r,df1]=fftseq(r,ts,df); % Fourier transform
env_r=env_phas(r); % envelope, when noise is present
dem2=2*(env_r-1)/a; % demodulate in the presence of noise
pause % Press any key to see a plot of the message
subplot(2,1,1)
plot(t,m(1:length(t)))
axis([0 0.15 -2.1 2.1])

```

```
xlabel('Time')
title('The message signal')
pause % Press any key to see a plot of the modulated signal
subplot(2,1,2)
plot(t,u(1:length(t)))
axis([0 0.15 -2.1 2.1])
xlabel('Time')
title('The modulated signal')
pause % Press a key to see the envelope of the modulated signal
clf
subplot(2,1,1)
plot(t,u(1:length(t)))
axis([0 0.15 -2.1 2.1])
xlabel('Time')
title('The modulated signal')
subplot(2,1,2)
plot(t,env(1:length(t)))
xlabel('Time')
title('Envelope of the modulated signal')
pause % Press a key to compare the message and the demodulated signal
clf
subplot(2,1,1)
plot(t,m(1:length(t)))
axis([0 0.15 -2.1 2.1])
xlabel('Time')
title('The message signal')
subplot(2,1,2)
plot(t,dem1(1:length(t)))
xlabel('Time')
title('The demodulated signal')
pause % Press a key to compare in the presence of noise
clf
subplot(2,1,1)
plot(t,m(1:length(t)))
axis([0 0.15 -2.1 2.1])
xlabel('Time')
title('The message signal')
subplot(2,1,2)
plot(t,dem2(1:length(t)))
xlabel('Time')
title('The demodulated signal in the presence of noise')
```

### Bình luận

Trong quá trình giải điều chế kể trên, chúng ta đã bỏ qua tác động của bộ lọc hạn chế nhiễu mà nó là một bộ lọc thông dải nằm ngay ở tầng đầu tiên của bất kỳ một máy thu nào. Trong thực tế, tín hiệu thu được  $r(t)$  được cho qua một mạch lọc hạn chế nhiễu rồi mới được cấp đến bộ tách sóng đường bao. Trong thí dụ trước, do độ rộng băng của tín hiệu bản tin không phải là hữu hạn, việc cho tín hiệu  $r(t)$  qua bất kỳ bộ lọc thông dải nào cũng sẽ gây ra méo trên bản tin sau điều chế, song nó cũng đồng thời làm giảm lượng tạp nhiễu trong tín hiệu lõi ra bộ giải điều chế. Trên hình vẽ 3.22, chúng tôi đã vẽ các tín hiệu lõi ra của bộ giải điều chế khi các bộ lọc hạn chế nhiễu với các độ rộng băng khác nhau được sử dụng. Trường hợp có độ rộng băng vô hạn thì tương đương với kết quả đã trình bày trên hình 3.21.



Hình 3.22 Tác động của độ rộng băng của mạch lọc hạn chế nhiễu tới tín hiệu đầu ra của bộ tách sóng đường bao.

### 3.4 ĐIỀU CHẾ GÓC

Các sơ đồ điều chế góc, bao gồm điều chế tần số (FM) và điều chế pha (PM), thuộc về lớp các sơ đồ điều chế phi tuyến. Họ các sơ đồ điều chế này được đặc trưng

bởi các yêu cầu độ rộng băng cao của chúng và hoạt động tốt trong điều kiện có tạp nhiễu. Các sơ đồ này có thể hình dung được như các kỹ thuật điều chế chấp nhận thiệt về độ rộng băng để được lợi về mặt công suất và do đó được sử dụng trong các tinh huống trong đó độ rộng băng không phải là mối quan tâm chính song lại yêu cầu có một SNR cao. Điều chế tần số được sử dụng rộng rãi trong các hệ thống phát thanh FM có độ trung thực cao, phát âm thanh trong truyền hình, điều chế sóng mang vi ba cũng như trong các hệ thống liên lạc điểm-tới-diểm.

Trong khi đề cập tới các sơ đồ điều chế góc, chúng ta lại sẽ tập trung vào năm đặc tính cơ bản của chúng là biểu diễn trên miền thời gian, biểu diễn trên miền tần số, độ rộng băng, công suất và cuối cùng là SNR. Do tồn tại một mối quan hệ chặt chẽ giữa PM và FM, chúng ta sẽ đề cập tới chúng một cách đồng thời, có nhấn mạnh hơn tới FM.

Biểu diễn trên miền thời gian của các tín hiệu được điều chế góc, khi sóng mang là  $c(t) = A_c \cos(2\pi f_c t)$  còn tín hiệu bản tin là  $m(t)$ , được cho bởi

$$u(t) = \begin{cases} A_c \cos[2\pi f_c t + k_p m(t)], & \text{PM} \\ A_c \cos[2\pi f_c t + 2\pi k_f \int_{-\infty}^t m(\tau) d\tau], & \text{FM} \end{cases} \quad (3.4.1)$$

trong đó  $k_f$  và  $k_p$  biểu thị các *hằng số di tần hoặc pha (deviation constant)* của FM và PM một cách tương ứng. Nói chung, biểu diễn trên miền tần số của tín hiệu được điều chế góc thì rất phức tạp do tính phi tuyến của các sơ đồ điều chế này. Chúng ta sẽ chỉ đề cập tới trường hợp khi tín hiệu bản tin  $m(t)$  là một tín hiệu hình sin. Ta giả sử rằng  $m(t) = a \cos(2\pi f_m t)$  với PM và  $m(t) = -a \sin(2\pi f_m t)$  đối với FM. Khi ấy tín hiệu sau điều chế có dạng

$$u(t) = \begin{cases} A_c \cos[2\pi f_c t + \beta_p \cos(2\pi f_m t)], & \text{PM} \\ A_c \cos[2\pi f_c t + \beta_f \cos(2\pi f_m t)], & \text{FM} \end{cases} \quad (3.4.2)$$

trong đó

$$\beta_p = k_p a, \quad \beta_f = \frac{k_f a}{f_m} \quad (3.4.3)$$

còn  $\beta_p$  và  $\beta_f$  tương ứng là các chỉ số điều chế của PM và FM. Nói chung, đối với một tín hiệu  $m(t)$  không phải hình sin thì các chỉ số điều chế được định nghĩa theo

$$\beta_p = k_p \max|m(t)|, \quad \beta_f = \frac{k_f \max|m(t)|}{W} \quad (3.4.4)$$

trong đó  $W$  là độ rộng băng của tín hiệu bản tin  $m(t)$ . Trong trường hợp của một tín hiệu bản tin hình sin, tín hiệu sau điều chế có thể biểu diễn được bởi

$$u(t) = \sum_{n=-\infty}^{\infty} A_c J_n(\beta) \cos[2\pi(f_c + nf_m)t] \quad (3.4.5)$$

ở đây  $J_n(\beta)$  là hàm Bessel loại 1 bậc  $n$ , còn  $\beta$  là  $\beta_p$  hay  $\beta_f$  tùy theo chúng ta đang đề cập đến PM hay FM. Trên miền tần số chúng ta có

$$U(f) = \sum_{n=-\infty}^{\infty} \left[ \frac{A_c J_n(\beta)}{2} \delta[f - (f_c + nf_m)] + \frac{A_c J_n(\beta)}{2} \delta[f + (f_c + nf_m)] \right] \quad (3.4.6)$$

Hiển nhiên, độ rộng băng của tín hiệu sau điều chế không hữu hạn. Tuy nhiên, ta có thể định nghĩa *độ rộng băng hiệu dụng (effective bandwidth)* của tín hiệu này như là độ rộng băng chứa từ 98% đến 99% công suất của tín hiệu sau điều chế. Độ rộng băng này được cho bởi qui tắc Carson như sau

$$B_T = 2(\beta+1)W \quad (3.4.7)$$

trong đó  $\beta$  là chỉ số điều chế,  $W$  là độ rộng băng của bản tin còn  $B_T$  là độ rộng băng của tín hiệu sau điều chế.

Biểu thức của công suất chứa trong các tín hiệu đã điều chế góc thì rất đơn giản. Do tín hiệu sau điều chế là một tín hiệu hình sin với tần số tức thời thì biến thiên còn biên độ thì bằng hằng số nên công suất của nó là hằng số và không phụ thuộc vào tín hiệu bản tin. Công suất đối với cả tín hiệu PM lẫn FM được cho bởi

$$P_u = \frac{A_c^2}{2} \quad (3.4.8)$$

Tỷ số SNR đối với các tín hiệu điều chế góc, khi lọc nhấn trước (*pre-emphasis filtering*) và lọc giải nhấn (*de-emphasis filtering*) không được áp dụng, được cho bởi

$$\left(\frac{S}{N}\right)_0 = \begin{cases} \frac{P_M \beta_p^2}{(\max|m(t)|)^2} \frac{P_R}{N_0 W}, & \text{PM} \\ 3 \frac{P_M \beta_f^2}{(\max|m(t)|)^2} \frac{P_R}{N_0 W}, & \text{FM} \end{cases} \quad (3.4.9)$$

Do  $\max |m(t)|$  ký hiệu cho biên độ cực đại của tín hiệu bản tin nên chúng ta có thể hiểu  $P_M/(\max |m(t)|)^2$  như là công suất của tín hiệu bản tin chuẩn hoá và ký hiệu nó là  $P_{mn}$ . Khi các bộ lọc nhấn trước và giải nhấn với tần số cắt-3dB bằng  $f_0$  được áp dụng thì SNR đối với FM được cho bởi

$$\left(\frac{S}{N}\right)_{\text{OPD}} = \frac{(W/f_0)^3}{3[W/f_0 - \arctan(W/f_0)]} \left(\frac{S}{N}\right)_0 \quad (3.4.10)$$

trong đó  $\left(\frac{S}{N}\right)_0$  là SNR khi không áp dụng lọc nhấn trước và lọc giải nhấn,

được cho bởi biểu thức (3.4.9).

### BÀI TẬP MINH HỌA

#### Bài tập minh họa 3.10 [Điều chế tần số] Tín hiệu bản tin

$$m(t) = \begin{cases} 1, & 0 \leq t < t_0 / 3 \\ -2, & t_0 / 3 \leq t < 2t_0 / 3 \\ 0, & \text{với các giá trị khác của } t \end{cases}$$

điều chế sóng mang  $c(t) = \cos(2\pi f_c t)$  bằng sơ đồ điều chế tần số. Giả sử rằng  $f_c = 200$  Hz và  $t_0 = 0.15$  s; hằng số di tần là  $k_f = 50$ .

1. Hãy vẽ tín hiệu sau điều chế.
2. Hãy xác định phổ của các tín hiệu bản tin và sau điều chế.

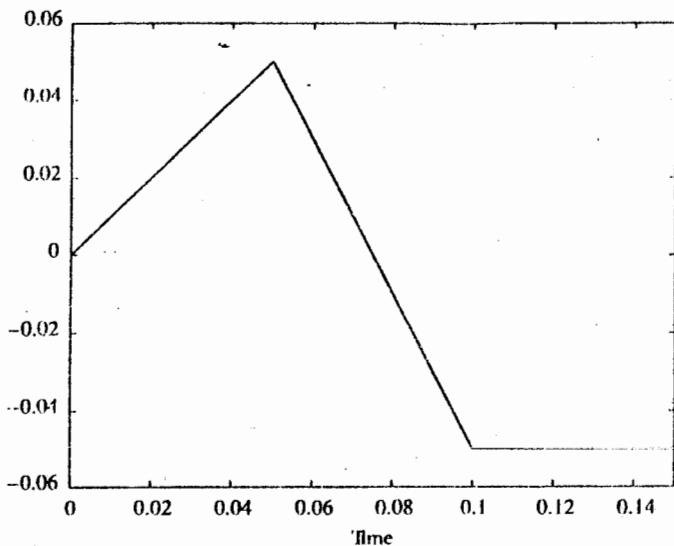
#### Lời giải

1. Chúng ta có

$$u(t) = A_c \cos[2\pi f_c t + 2\pi k_f \int_{-\infty}^t m(\tau) d\tau]$$

Chúng ta phải tìm  $\int_{-\infty}^t m(\tau) d\tau$ . Điều này có thể làm được một cách giải tích hoặc thông qua tính bằng số và các kết quả được trình bày trên hình 3.23. Sử dụng quan

hệ đối với  $u(t)$  và giá trị của tích phân của  $m(t)$  như đã nêu trên, chúng ta có được biểu diễn đối với  $u(t)$ . Một đồ thị của  $m(t)$  và  $u(t)$  được thể hiện trên hình vẽ 3.24.



Hình 3.23 Tích phân của tín hiệu bản tin.

2. Sử dụng các thủ tục biến đổi Fourier của MATLAB, chúng ta có được biểu diễn đối với phổ của  $u(t)$  như được thể hiện trên hình 3.25. Thấy ngay được rằng, khác với AM, trong trường hợp FM không có sự tương tự rõ rệt giữa phổ của tín hiệu bản tin và phổ của tín hiệu sau điều chế. Trong thí dụ cụ thể này, độ rộng băng của tín hiệu bản tin thì không hữu hạn và do đó để xác định chỉ số điều chế thì một độ rộng băng xấp xỉ đối với bản tin cần phải được sử dụng trong biểu thức

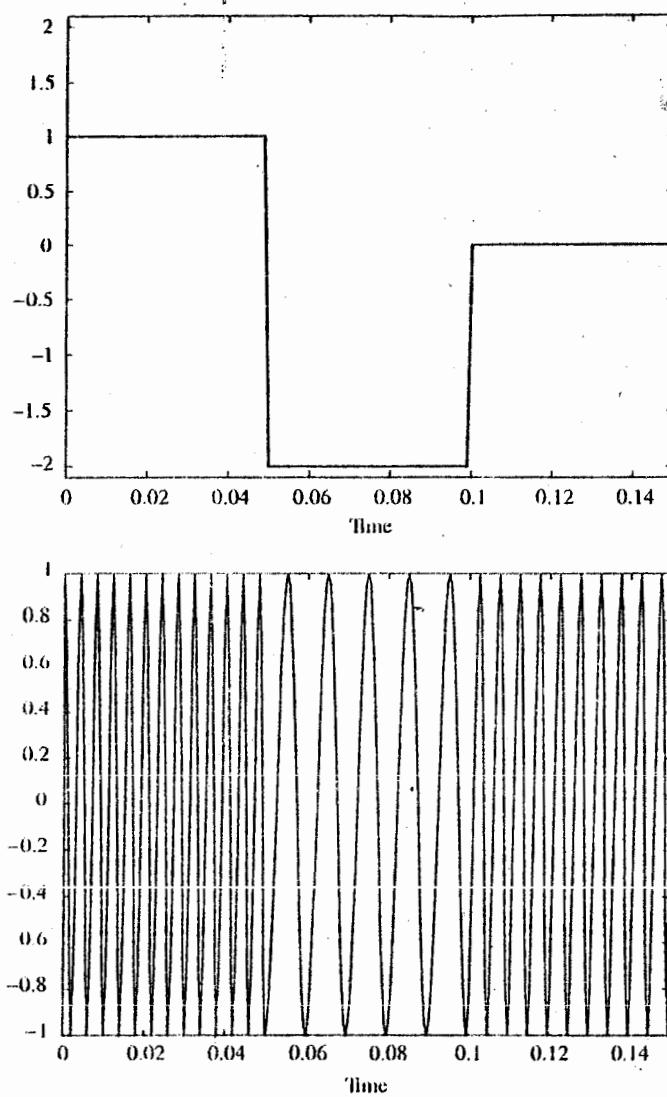
$$\beta = \frac{k_f \max|m(t)|}{W}$$

Thí dụ, chúng ta có thể định nghĩa độ rộng băng đó là độ rộng băng thùy chính của phổ của  $m(t)$  mà nó dẫn đến

$$W=20 \text{ Hz}$$

và vì thế

$$\beta = \frac{50 \times 2}{20} = 5$$



Hình 3.24 Tín hiệu bản tin và tín hiệu sau điều chế.

MATLAB *script* cho bài tập này như sau.

#### m-file

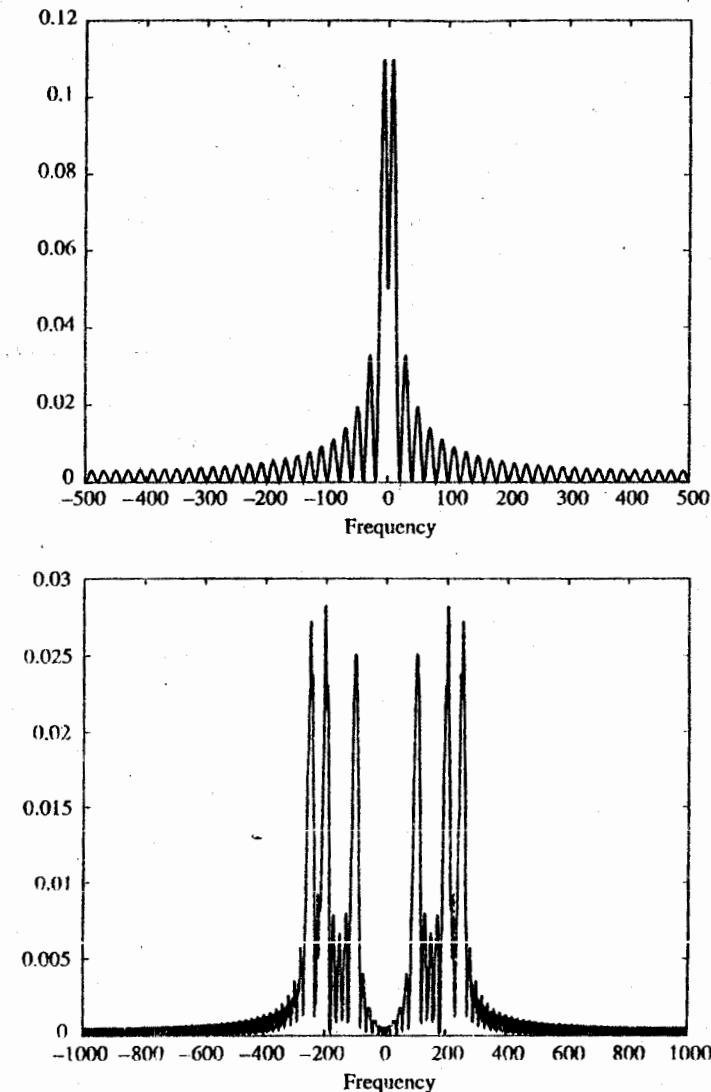
```
% fm1.m
% Matlab demonstration script for frequency modulation. The message signal
% is +1 for 0 < t < t0/3, -2 for t0/3 < t < 2t0/3 and zero otherwise.
echo on
```

```

t0=.15; % signal duration
ts=0.0005; % sampling interval
fc=200; % carrier frequency
kf=50; % Modulation index
fs=1/ts; % sampling frequency
t=[0:ts:t0]; % time vector
df=0.25; % required frequency resolution

% message signal
m=[ones(1,t0/(3*ts)),-2*ones(1,t0/(3*ts)),zeros(1,t0/(3*ts)+1)];
int_m(1)=0;
for i=1:length(t)-1 % Integral of m
    int_m(i+1)=int_m(i)+m(i)*ts;
    echo off ;
end
echo on ;
[M,m,df1]=fftseq(m,ts,df); % Fourier transform
M=M/fs; % scaling
f=[0:df1:df1*(length(m)-1)]-fs/2; % frequency vector
u=cos(2*pi*fc*t+2*pi*kf*int_m); % modulated signal
[U,u,df1]=fftseq(u,ts,df); % Fourier transform
U=U/fs; % scaling
pause % Press any key to see a plot of the message and the modulated signal
subplot(2,1,1)
plot(t,m(1:length(t)))
axis([0 0.15 -2.1 2.1])
xlabel('Time')
title('The message signal')
subplot(2,1,2)
plot(t,u(1:length(t)))
axis([0 0.15 -2.1 2.1])
xlabel('Time')
title('The modulated signal')
pause % Press any key to see a plots of the magnitude of the message and the
% modulated signal in the frequency domain.
subplot(2,1,1)
plot(f,abs(fftshift(M)))
xlabel('Frequency')
title('Magnitude-spectrum of the message signal')
subplot(2,1,2)
plot(f,abs(fftshift(U)))
title('Magnitude-spectrum of the modulated signal')
xlabel('Frequency')

```



Hình 3.25 Các phổ biến độ của các tín hiệu bản tin và sau điều chế.

### BÀI TẬP MINH HỌA

**Bài tập minh họa 3.11 [Điều chế tần số]** Ta hãy cho tín hiệu bản tin là

$$m(t) = \begin{cases} \sin c(100t), & |t| \leq t_0 \\ 0, & \text{với các giá trị khác của } t \end{cases}$$

trong đó  $t_0=0.1$ . Bản tin này điều chế sóng mang  $c(t)=\cos(2\pi f_c t)$ , trong đó  $f_c=250$  Hz. Hàng số di tần là  $k_f=100$ .

1. Hãy vẽ tín hiệu sau điều chế trên miền thời gian và tần số.

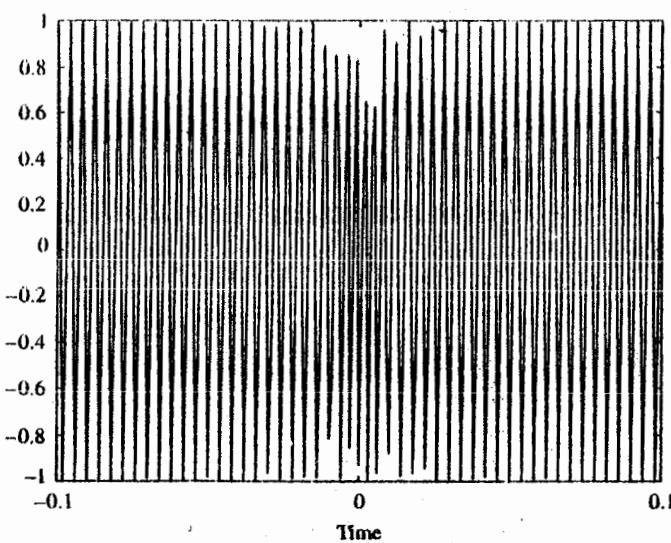
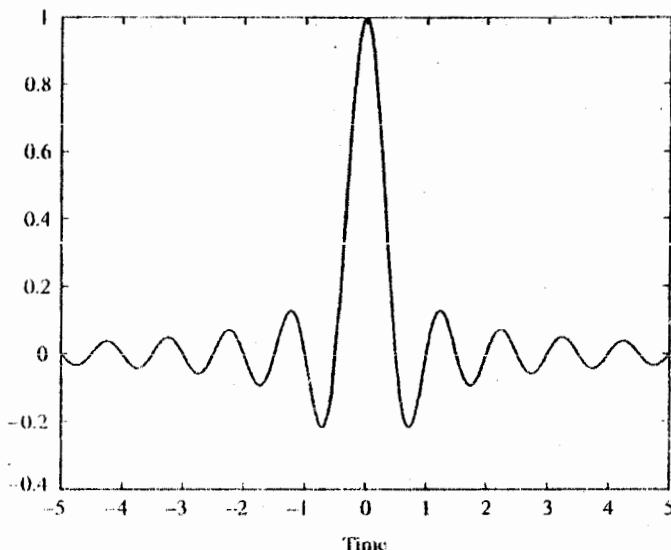
2. Hãy so sánh tín hiệu lõi ra bộ giải điều chế và tín hiệu bản tin gốc.

### Lời giải

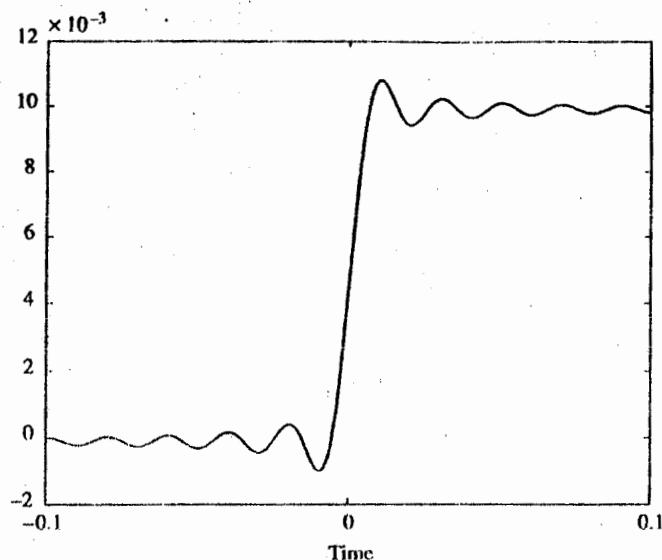
1. Trước hết chúng ta lấy tích phân tín hiệu bản tin rồi sử dụng quan hệ

$$u(t) = A_c \cos[2\pi f_c t + 2\pi k_f \int_{-\infty}^t m(\tau) d\tau]$$

để tìm  $u(t)$ . Một đồ thị của  $u(t)$  cùng với tín hiệu bản tin được cho thấy trên hình vẽ 3.26. Tích phân của tín hiệu bản tin được trình bày trên hình 3.27. Một đồ thị của tín hiệu sau điều chế trên miền tần số được trình bày trên hình vẽ 3.28.



Hình 3.26 Các tín hiệu bản tin và sau điều chế.



Hình 3.27 Tích phân của tín hiệu bản tin.

2. Để giải điều chế tín hiệu FM, trước hết chúng ta tìm góc pha của tín hiệu sau điều chế  $u(t)$ . Pha này là  $2\pi k_f \int_{-\infty}^t m(\tau) d\tau$ , có thể vi phân được và chia cho  $2\pi k_f$  để có được  $m(t)$ . Hãy chú ý rằng để khôi phục được góc pha và loại bỏ tác động của hiện tượng bội pha  $2\pi$ , chúng tôi đã áp dụng hàm **unwrap.m** của MATLAB. Các đồ thị của tín hiệu bản tin và tín hiệu sau giải điều chế được thể hiện trên hình 3.29. Như các bạn có thể thấy được, tín hiệu sau giải điều chế hoàn toàn tương tự với tín hiệu bản tin.

MATLAB *script* cho bài tập này như sau.

#### **m-file**

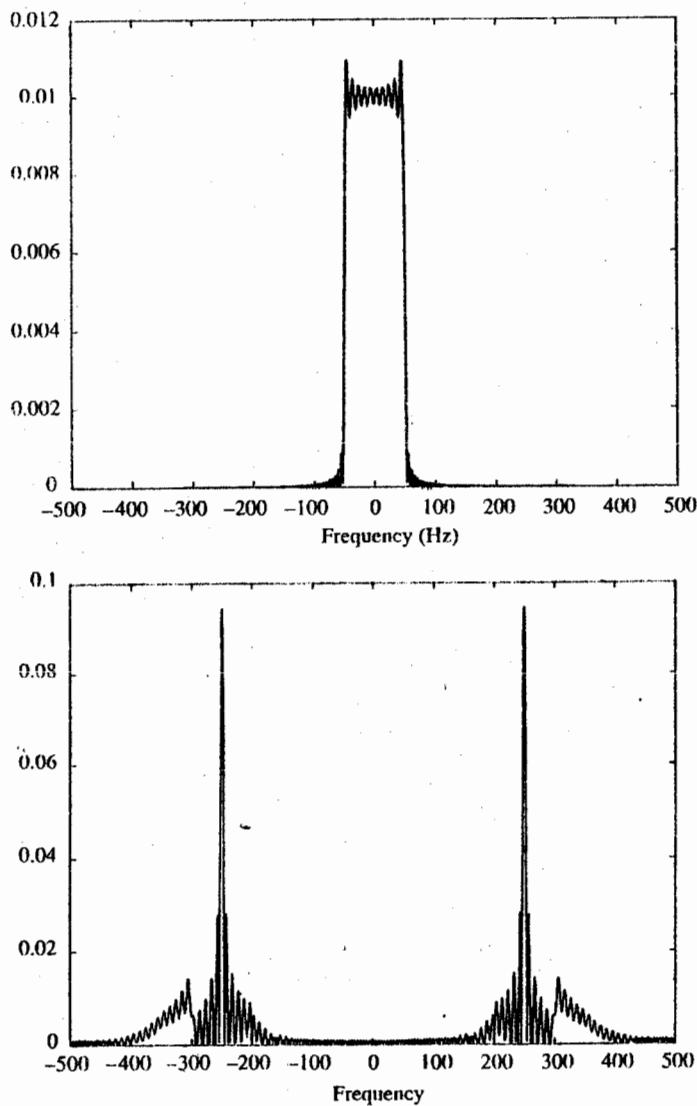
```
% fm2.m
% Matlab demonstration script for frequency modulation. The message signal
% is m(t)=sinc(100t).
echo on
t0=.2; % signal duration
ts=0.001; % sampling interval
fc=250; % carrier frequency
snr=20; % SNR in dB (logarithmic)
```

```

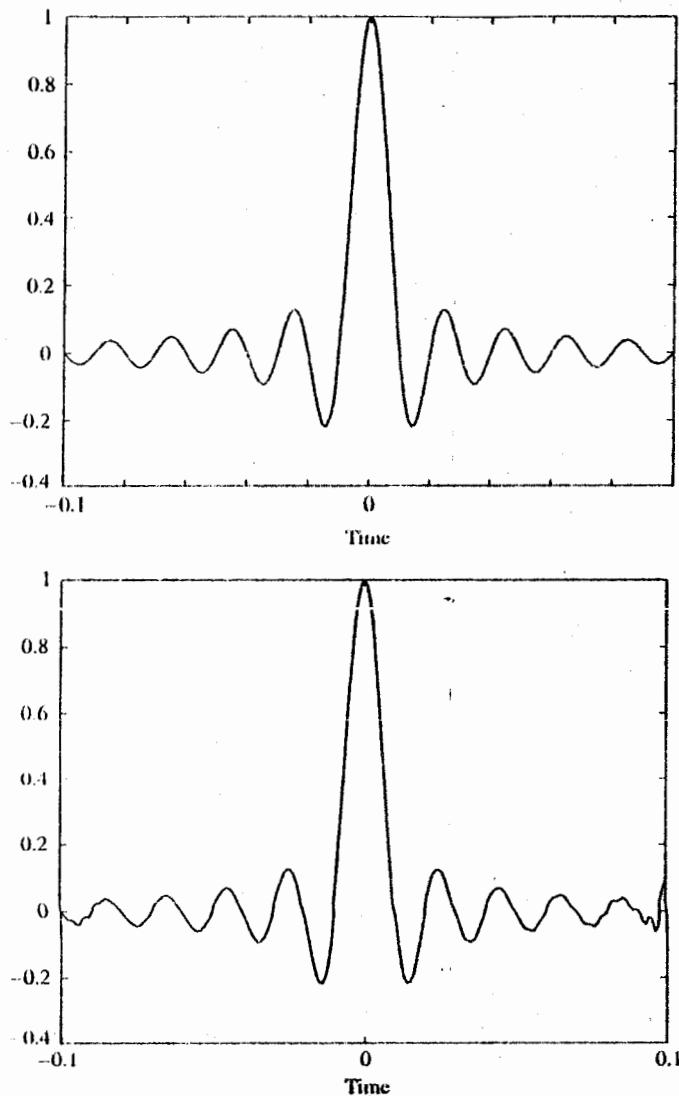
fs=1/ts; % sampling frequency
df=0.3; % required freq. resolution
t=[-t0/2:ts:t0/2]; % time vector
kf=100; % deviation constant
df=0.25; % required frequency resolution
m=sinc(100*t); % the message signal
int_m(1)=0;
for i=1:length(t)-1 % Integral of m
    int_m(i+1)=int_m(i)+m(i)*ts;
    echo off ;
end
echo on ;
[M,m,df1]=fftseq(m,ts,df); % Fourier transform
M=M/fs; % scaling
f=[0:df1:df1*(length(m)-1)]-fs/2; % frequency vector
u=cos(2*pi*fc*t+2*pi*kf*int_m); % modulated signal
[U,u,df1]=fftseq(u,ts,df); % Fourier transform
U=U/fs; % scaling
[v,phase]=env_phas(u,ts,250); % demodulation, find phase of u
phi=unwrap(phase); % restore original phase
dem=(1/(2*pi*kf))*(diff(phi)/ts); % demodulator output, differentiate and scale phase
pause % Press any key to see a plot of the message and the modulated signal
subplot(2,1,1)
plot(t,m(1:length(t)))
xlabel('Time')
title('The message signal')
subplot(2,1,2)
plot(t,u(1:length(t)))
xlabel('Time')
title('The modulated signal')
pause % Press any key to see a plots of the magnitude of the message and the
% modulated signal in the frequency domain.
subplot(2,1,1)
plot(f,abs(fftshift(M)))
xlabel('Frequency')
title('Magnitude-spectrum of the message signal')
subplot(2,1,2)
plot(f,abs(fftshift(U)))
title('Magnitude-spectrum of the modulated signal')
xlabel('Frequency')
pause % Pres any key to see plots of the message and the demodulator output
% with no noise

```

```
subplot(2,1,1)
plot(t,m(1:length(t)))
xlabel('Time')
title('The message signal')
subplot(2,1,2)
plot(t,dem(1:length(t)))
xlabel('Time')
title('The demodulated signal')
```



Hình 3.28 Các phổ biến độ của bản tin và tín hiệu sau điều chế.



Hình 3.29 Tín hiệu bản tin và tín hiệu sau giải điều chế.

### Câu hỏi

Một tín hiệu điều chế tần số có biên độ không đổi. Tuy vậy, trên hình vẽ 3.26 biên độ của tín hiệu  $u(t)$  lại rõ ràng không phải là hằng số. Bạn có thể giải thích được tại sao lại như vậy không?

## CÁC BÀI TẬP

### 3.1 Tín hiệu $m(t)$ được cho bởi

$$m(t) = \begin{cases} t, & 0.1 \leq t < 1 \\ -t + 2, & 1 \leq t < 1.9 \\ 0.1, & \text{với các giá trị khác của } t \end{cases}$$

trong khoảng  $[0, 2]$ . Tín hiệu này được sử dụng để điều chế DSB một sóng mang có tần số  $25 \text{ Hz}$  và biên độ bằng  $1$  để tạo nên tín hiệu sau điều chế  $u(t)$ . Hãy viết một MATLAB *m-file* và sử dụng nó để làm những việc sau đây.

- a. Vẽ đồ thị tín hiệu sau điều chế.
- b. Xác định công suất của tín hiệu sau điều chế.
- c. Xác định phổ của tín hiệu sau điều chế.
- d. Xác định mật độ phổ công suất của tín hiệu sau điều chế và so sánh nó với mật độ phổ công suất của tín hiệu bản tin.

### 3.2 Lặp lại bài tập 3.1 với

$$m(t) = \begin{cases} t, & 0 \leq t < 1 \\ -t + 2, & 1 \leq t \leq 2 \end{cases}$$

trong khoảng  $[0, 2]$ . Bạn nghĩ gì về sự khác biệt giữa bài tập này và bài tập 3.1?

### 3.3 Lặp lại bài tập 3.1 với

$$m(t) = \begin{cases} \sin c^2(10t), & |t| \leq 2 \\ 0, & \text{với các giá trị khác của } t \end{cases}$$

và với một sóng mang có tần số  $100 \text{ Hz}$ .

### 3.4 Trong bài tập 3.1, giả sử rằng thay cho DSB, một sơ đồ AM truyền thống được sử dụng với chỉ số điều chế $\alpha=0.2$ .

- a. Xác định và vẽ phổ của tín hiệu sau điều chế.
- b. Hãy thay đổi chỉ số điều chế từ  $0.1$  đến  $0.9$  và giải thích xem nó tác động thế nào tới phổ đã tìm được trong phần a.
- c. Hãy vẽ tỷ số công suất của các băng biên với công suất sóng mang như một hàm của chỉ số điều chế.

**3.5** Trong bài tập 3.1, hãy lấy sơ đồ điều chế là USSB thay cho DSB. Trước hết, hãy sử dụng một sơ đồ điều chế DSB rồi thì loại bỏ đi băng biên dưới.

a. Hãy xác định và vẽ tín hiệu sau điều chế.

b. Hãy xác định và vẽ phổ của tín hiệu sau điều chế.

c. Hãy so sánh phổ của tín hiệu sau điều chế với phổ của tín hiệu không được điều chế.

**3.6** Lặp lại bài tập 3.5, song thay cho việc sử dụng một DSB có lọc để tạo ra tín hiệu USSB, hãy sử dụng quan hệ

$$u(t) = \frac{A_c}{2} m(t) \cos(2\pi f_c t) - \frac{A_c}{2} \hat{m}(t) \sin(2\pi f_c t)$$

Bạn có nhận thấy sự khác biệt nào so với các kết quả đã nhận được trong bài tập 3.5 không?

**3.7** Lặp lại bài tập 3.5 và 3.6 song thay tín hiệu bản tin bằng

$$m(t) = \begin{cases} \sin c^2(10t), & |t| \leq 2 \\ 0, & \text{với các giá trị khác của } t \end{cases}$$

và sử dụng một sóng mang có tần số 100 Hz để tạo ra một tín hiệu LSSB.

**3.8** Tín hiệu

$$m(t) = \begin{cases} t, & 0.1 \leq t < 1 \\ -t + 2, & 1 \leq t < 1.9 \\ 0.1, & \text{với các giá trị khác của } t \end{cases}$$

được sử dụng để điều chế một sóng mang có tần số 25 Hz bằng sơ đồ DSB.

A. Hãy xác định và vẽ đồ thị tín hiệu sau điều chế.

B. Giả sử rằng bộ dao động nội tại bộ giải điều chế có pha chậm so với pha của sóng mang một góc  $\theta$ , trong đó  $0 \leq \theta \leq \pi/2$ . Hãy sử dụng MATLAB vẽ đồ thị công suất của tín hiệu sau giải điều chế theo  $\theta$  (giả sử việc truyền không có tạp nhiễu).

**3.9** Trong bài tập 3.8, giả sử rằng sơ đồ điều chế là USSB. Hãy vẽ tín hiệu sau giải điều chế đối với  $\theta = 0^\circ, 30^\circ, 45^\circ, 60^\circ$  và  $90^\circ$ .

**3.10** Lặp lại bài tập 3.8 với giả sử rằng việc điều chế là AM truyền thống với chỉ số

điều chế bằng 0.2. Hãy sử dụng một bộ tách sóng đường bao để giải điều chế tín hiệu, trong trường hợp không có tạp âm, và vẽ tín hiệu sau giải điều chế và phổ của nó.

**3.11** Một tín hiệu bản tin tuần hoàn với chu kỳ 2 s và được xác định trong khoảng [0, 2] theo

$$m(t) = \begin{cases} t, & 0.1 \leq t < 1 \\ -t + 2, & 1 \leq t < 1.9 \\ 0, & \text{với các giá trị khác của } t \end{cases}$$

Tín hiệu bản tin này thực hiện điều chế DSB một sóng mang có tần số 50 Hz. Hãy vẽ tín hiệu lõi ra của bộ giải điều chế DSB và so sánh nó với bản tin trong trường hợp tạp nhiễu trắng Gauss được cộng vào tín hiệu sau điều chế với một công suất bằng 0.001, 0.01, 0.05, 0.1 và 0.3 công suất của tín hiệu sau điều chế.

**3.12** Lặp lại bài tập 3.11 với một sơ đồ điều chế LSSB. Hãy so sánh các kết quả với các kết quả đã nhận được trong bài tập 3.11.

**3.13** Lặp lại bài tập 3.11 với một sơ đồ điều chế AM truyền thống và giải điều chế đường bao.

#### 3.14 Tín hiệu

$$m(t) = \begin{cases} t, & 0 \leq t < 1 \\ -t + 2, & 1 < t \leq 2 \\ 0.1, & \text{với các giá trị khác của } t \end{cases}$$

điều chế tần số một sóng mang có tần số 1000 Hz. Hằng số di tần là  $k_f=25$ .

- a. Hãy xác định dải tần số tức thời của tín hiệu sau điều chế.
- b. Hãy xác định độ rộng băng của tín hiệu sau điều chế.
- c. Hãy vẽ các phổ của bản tin và của tín hiệu sau điều chế.
- d. Hãy xác định chỉ số điều chế.

**3.15** Hãy giải điều chế tín hiệu đã được điều chế trong bài tập 3.14 nhờ sử dụng một file MATLAB giải điều chế tần số và so sánh tín hiệu sau giải điều chế với tín hiệu bản tin.

**3.16** Cho tín hiệu bản tin là một tín hiệu tuần hoàn với chu kỳ bằng 2, mô tả được trong khoảng  $[0, 2]$  theo

$$m(t) = \begin{cases} t, & 0 \leq t < 1 \\ -t + 2, & 1 < t \leq 2 \\ 0.1, & \text{với các giá trị khác của } t \end{cases}$$

và cho sơ đồ điều chế là sơ đồ đã được mô tả trong bài tập 3.14. Trước khi giải điều chế, một nhiễu trắng Gauss được cộng với tín hiệu sau điều chế. Hãy giải điều chế và vẽ đồ thị tín hiệu sau giải điều chế khi tỷ số công suất tạp nhiễu trên công suất tín hiệu sau điều chế là 0.001, 0.01, 0.05, 0.1 và 0.3.

**3.17** Trong bài tập 3.16, giả sử  $m(t)=0$ . Hãy cộng các quá trình tạp nhiễu với tín hiệu sau điều chế như đã mô tả rồi giải điều chế tín hiệu vừa có được. Hãy vẽ đồ thị mật độ phổ công suất của tín hiệu lối ra bộ giải điều chế trong từng trường hợp.

## CHƯƠNG 4

# BIẾN ĐỔI TƯƠNG TỰ-SỐ

### 4.1 GIỚI THIỆU CHUNG

Về bản chất, đa số các nguồn thông tin là các nguồn tương tự. Các nguồn tin tương tự bao gồm tiếng nói, hình ảnh và các nguồn điều khiển xa. Trong chương này chúng ta xem xét một số phương pháp và kỹ thuật được sử dụng để biến đổi các bản tin từ các nguồn tương tự thành các chuỗi số một cách hữu hiệu. Điều này là cần thiết do, như chúng ta sẽ thấy trong các chương tiếp theo, thông tin dưới dạng số thì dễ xử lý, dễ truyền và dễ lưu trữ hơn. Chủ đề tổng quát về nén dữ liệu, mà biến đổi tương tự-số là một trường hợp riêng, có thể chia được thành hai nhánh chính:

1. **Lượng tử hoá** (hay **nén dữ liệu có thất thoát**), trong đó nguồn dữ liệu được lượng tử hoá thành một số hữu hạn các mức. Trong quá trình này một số méo sẽ xảy ra không thể tránh khỏi được, do đó một số thông tin sẽ bị mất. Lượng tin thất thoát này không thể khôi phục lại được. Các kỹ thuật biến đổi tương tự-số tổng quát như điều chế mã xung (PCM: *Pulse-Code Modulation*), điều chế mã xung vi sai (DPCM: *Differential Pulse-Code Modulation*), điều chế delta (DM: *Delta Modulation*), lượng tử hoá đều, lượng tử hoá không đều và lượng tử hoá véc-tơ đều thuộc về lớp này. Giới hạn cản bản về chất lượng hoạt động của lớp các sơ đồ nén dữ liệu này được cho bởi *giới hạn chặn méo tỷ lệ* (*rate-distortion bound*).

2. **Mã hoá không tạp** (hay **nén dữ liệu không có thất thoát**), trong đó dữ liệu số (thường là kết quả của việc lượng tử hoá như đã nói trên đây) được nén với mục đích biểu diễn chúng với càng ít bít càng tốt sao cho chuỗi dữ liệu gốc có thể khôi phục lại được hoàn toàn từ chuỗi đã được nén. Các kỹ thuật mã hoá nguồn

như mã Huffman, mã Lempel-Ziv và mã số học thuộc vào lớp các sơ đồ nén dữ liệu này. Trong lớp các sơ đồ mã hoá này việc thất thoát thông tin không xảy ra. Giới hạn căn bản về nén đạt được nhờ lớp này được cho bởi *entropy* của nguồn.

## 4.2 ĐỘ ĐO THÔNG TIN

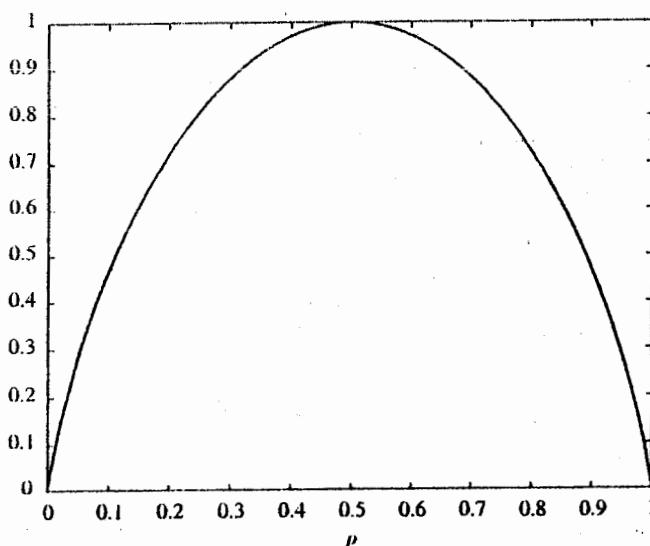
Tín hiệu lối ra của một nguồn tin (dữ liệu, tiếng nói, hình ảnh...) có thể mô hình hoá được như một quá trình ngẫu nhiên. Đối với một quá trình ngẫu nhiên rời rạc không nhớ và dừng, mà nó có thể được xét đến như những bức tranh độc lập của một biến ngẫu nhiên  $X$ , thì nội dung thông tin, hay *entropy*, được định nghĩa theo

$$H(X) = - \sum_{x \in \chi} p(x) \log p(x) \quad (4.2.1)$$

trong đó  $\chi$  ký hiệu cho bảng chữ cái của nguồn còn  $p(x)$  là xác suất của chữ cái  $x$ . Cơ số của *logarit* thường được chọn bằng 2 dẫn đến *entropy* được tính bằng *bit*. Đối với bảng chữ cái nhị phân với các xác suất  $p$  và  $p=1-p$ , *entropy* được ký hiệu bằng  $H_b(p)$  và được cho bởi

$$H_b(p) = -p \log p - (1-p) \log(1-p) \quad (4.2.2)$$

Một đồ thị của hàm *entropy* nhị phân được cho trên hình 4.1.



Hình 4.1 Đồ thị của hàm *entropy* nhị phân.

Entropy của một nguồn cung cấp cho chúng ta một giới hạn chặn cơ bản về số bít cần thiết để biểu diễn một nguồn để khôi phục được đầy đủ. Nói một cách khác, số bít trung bình trên một sản phẩm lối ra của nguồn để mã hoá một nguồn nhằm khôi phục lại được không có lỗi có thể làm cho gần tới  $H(X)$  tuy ý ta muốn song không thể ít hơn  $H(X)$ .

#### 4.2.1 MÃ HOÁ KHÔNG TẠP

*Mã hoá không tạp* là thuật ngữ chung cho mọi sơ đồ làm giảm số bít cần thiết để biểu diễn một sản phẩm của nguồn nhằm khôi phục lại được một cách hoàn hảo. Định lý mã hoá không tạp do Shannon phát biểu tuyên bố rằng để khôi phục lại hoàn hảo một nguồn có thể sử dụng một mã với một tốc độ gần với *entropy* của nguồn một cách tùy ý song không thể sử dụng một mã có tốc độ bé hơn *entropy* của nguồn. Nói một cách khác đi, đối với bất kỳ một số  $\varepsilon > 0$ , chúng ta có thể có một mã với tốc độ nhỏ hơn  $H(X) + \varepsilon$ , song ta không thể có được một mã với tốc độ  $H(X)$ , bất luận độ phức tạp của bộ mã hoá và bộ giải mã thế nào. Có một loạt các thuật toán đối với mã hoá nguồn không tạp: mã hoá Huffman và mã hoá Lempel-Ziv là hai thí dụ. Chúng ta thảo luận ở đây thuật toán Huffman.

##### *Mã hoá Huffman*

Trong mã hoá Huffman, chúng ta gán các từ mã dài hơn cho các lối ra của nguồn có khả năng xuất hiện ít hơn và các từ mã ngắn hơn cho các lối ra của nguồn có khả năng xuất hiện lớn hơn. Để là điều đó, chúng ta bắt đầu bằng việc hợp nhất hai lối ra của nguồn có khả năng ít nhất để tạo nên một lối ra đã hợp nhất mới có xác suất là tổng của các xác suất tương ứng. Quá trình này được lặp lại cho tới khi chỉ còn lại một lối ra hợp nhất. Theo cách này, chúng ta tạo ra một cây mã. Bằng cách khởi đầu từ gốc của cây và gán các *digit* 0 và 1 cho hai nhánh bất kỳ hợp nhất từ cùng một nút, chúng ta tạo ra được mã. Có thể chỉ ra được rằng bằng cách này chúng ta tạo ra một mã với độ dài trung bình nhỏ nhất trong số các

mã tự do prefix<sup>1</sup> (*prefix-free codes*). Thí dụ sau đây sẽ cho thấy làm thế nào để thiết kế một mã Huffman.

### BÀI TẬP MINH HỌA

**Bài tập minh họa 4.1 [Mã hoá Huffman]** Hãy thiết kế một mã Huffman cho một nguồn với bảng chữ cái  $\chi = \{x_1, x_2, \dots, x_9\}$  và véc-tơ xác suất tương ứng

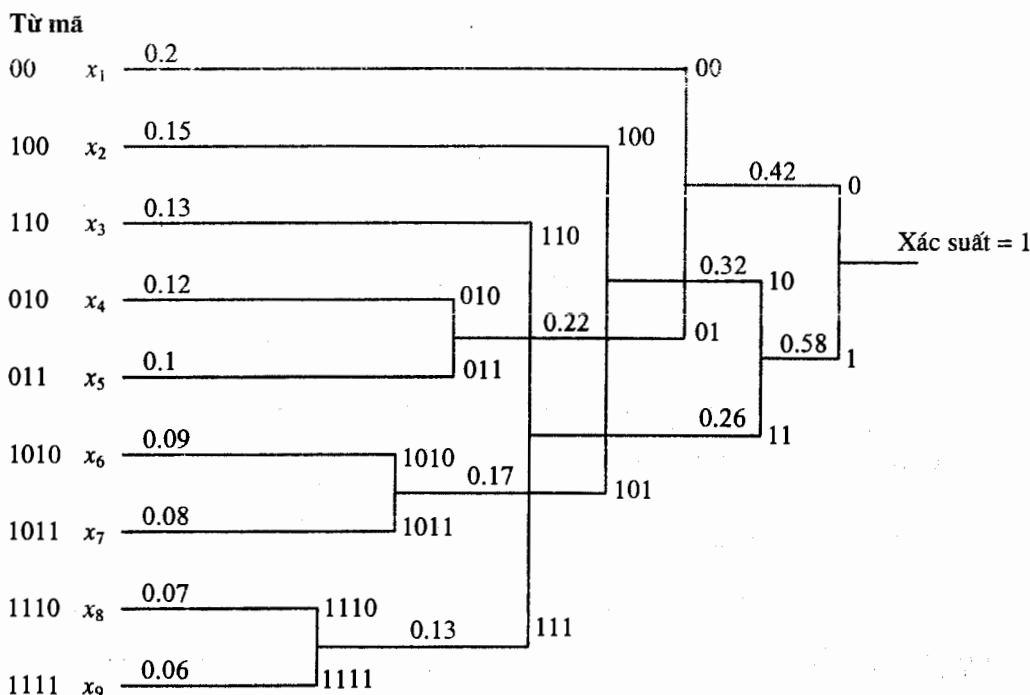
$$p = (0.2, 0.15, 0.13, 0.12, 0.1, 0.09, 0.08, 0.07, 0.06)$$

Hãy tìm độ dài từ mã trung bình của mã được tạo ra và so sánh nó với *entropy* của nguồn.

#### Lời giải

Chúng ta theo thuật toán đã được nêu ra trên đây để có được cây mã như trên hình vẽ 4.2. Độ dài từ mã trung bình đối với mã này là

$\bar{L} = 2 \times 0.2 + 3 \times (0.15 + 0.13 + 0.12 + 0.1) + 4 \times (0.09 + 0.08 + 0.07 + 0.06) = 3.1$  bít  
trên một lối ra của nguồn.



Hình 4.2 Cây mã Huffman.

1) Các mã tự do *prefix* là các mã trong đó không một từ mã nào là tiếp đầu của một từ mã khác

*Entropy* của nguồn được cho theo

$$H(X) = - \sum_{i=1}^9 p_i \log p_i = 3.0371 \text{ bít/sản phẩm lối ra của nguồn.}$$

Chúng ta thấy, như có thể đoán được, rằng  $\bar{L} > H(X)$ .

Hàm MATLAB **entropy.m** cho dưới đây thực hiện tính *entropy* của một véc-tơ xác suất  $p$ .

### m-file

```
function h=entropy(p)
%      H=ENTROPY(P) returns the entropy function of the probability vector P.
if length(find(p<0))~=0,
    error('Not a prob. vector, negative component(s)')
end
if abs(sum(p)-1)>10e-10,
    error('Not a prob. vector, components do not add up to 1')
end
h=sum(-p.*log2(p));
```

Đại lượng

$$\eta = \frac{H(X)}{\bar{L}} \quad (4.2.3)$$

được gọi là hiệu suất của mã Huffman. Hiển nhiên, chúng ta luôn luôn có  $\eta \leq 1$ . Nói chung, có thể chỉ ra được rằng độ dài từ mã trung bình đối với một mã Huffman bất kỳ đều thoả mãn các bất đẳng thức

$$H(X) \leq \bar{L} < H(X)+1 \quad (4.2.4)$$

Nếu chúng ta thiết kế một mã Huffman đối với các khối có độ dài  $K$  chứ không phải cho từng chữ cái đơn, chúng ta sẽ có

$$H(X) \leq \bar{L} < H(X) + \frac{1}{K} \quad (4.2.5)$$

và do đó, bằng cách tăng  $K$ , chúng ta có thể tiến gần đến  $H(X)$  bao nhiêu cũng được tùy thích. Khỏi phải nói, việc tăng  $K$  sẽ làm tăng mức độ phức tạp một cách đáng kể. Cũng cần phải lưu ý rằng thuật toán mã Huffman không dẫn đến một mã duy nhất do cách gán tùy ý các bít 0 và 1 cho các nhánh cây khác nhau. Đó là tại sao mà chúng ta nói đến **một** mã Huffman chứ không nói mã Huffman.

Hàm MATLAB **huffman.m**, thực hiện việc thiết kế một mã Huffman cho một nguồn không nhớ rời rạc có véc-tơ xác suất  $p$  và cho ra cả các từ mã lẫn độ dài trung bình của từ mã, được cho dưới đây.

### m-file

```

function [h,l]=huffman(p);
%HUFFMAN      Huffman code generator.
%      [h,l]=huffman(p), Huffman code generator returns h the Huffman code matrix,
%      and l the average codeword length for a source with probability vector p.

if length(find(p<0))~=0,
    error('Not a prob. vector, negative component(s)')
end
if abs(sum(p)-1)>10e-10,
    error('Not a prob. vector, components do not add up to 1')
end
n=length(p);
q=p;
m=zeros(n-1,n);
for i=1:n-1
    [q,l]=sort(q);
    m(i,:)=[l(1:n-i+1),zeros(1,i-1)];
    q=[q(1)+q(2),q(3:n),1];
end
for i=1:n-1
    c(i,:)=blanks(n*n);
end
c(n-1,n)='0';
c(n-1,2*n)='1';
for i=2:n-1
    c(n-i,1:n-1)=c(n-i+1,n)*(find(m(n-i+1,:)==1)...%
        -(n-2):n*(find(m(n-i+1,:)==1)));
    c(n-i,n)='0';
    c(n-i,n+1:2*n-1)=c(n-i,1:n-1);
    c(n-i,2*n)='1';
    for j=1:i-1
        c(n-i,(j+1)*n+1:(j+2)*n)=c(n-i+1,...%
            n*(find(m(n-i+1,:)==j+1)-1)+1:n*find(m(n-i+1,:)==j+1));
    end
end
for i=1:n
    h(i,1:n)=c(1,n*(find(m(1,:)==i)-1)+1:find(m(1,:)==i)*n);

```

```
l1(i)=length(find(abs(h(i,:))~=32));  
end  
l=sum(p.*l1);
```

## BÀI TẬP MINH HỌA

**Bài tập minh họa 4.2 [Mã hoá Huffman]** Một nguồn tin rời rạc không nhớ có bảng chữ cái  $\chi = \{x_1, x_2, \dots, x_6\}$  và các xác suất tương ứng

$$p = \{0.1, 0.3, 0.05, 0.09, 0.21, 0.25\}$$

cần mã bằng cách sử dụng mã hoá Huffman.

1. Hãy xác định *entropy* của nguồn này.
2. Hãy tìm một mã Huffman cho nguồn này và xác định hiệu suất của mã Huffman đó.
3. Bây giờ, hãy thiết kế một mã Huffman cho các dãy của nguồn có độ dài bằng 2 và hãy so sánh hiệu suất của mã này với hiệu suất mã của mã đã tìm được trong phần 2.

### Lời giải

1. *Entropy* của nguồn xác định được bằng hàm **entropy.m** và tính được bằng 2.3549 bít/*symbol* của nguồn.

2. Sử dụng hàm **huffman.m**, chúng ta có thể thiết kế một mã Huffman cho nguồn này. Các từ mã tìm được là 010, 11, 0110, 0111, 00 và 10. Độ dài từ mã trung bình đối với mã này tìm được là 2.38 *symbol* nhị phân/một sản phẩm của nguồn. Do đó, hiệu suất của mã này là

$$\eta_1 = \frac{2.3549}{2.38} = 0.9895$$

3. Một nguồn mới, mà các sản phẩm của nó là các cặp chữ cái của nguồn ban đầu, có 36 chữ cái lối ra ở dạng  $\{(x_i, x_j)\}_{i,j=1}^6$ . Do nguồn này là không nhớ nên xác suất của mỗi một cặp là tích của các xác suất của từng chữ cái riêng lẻ. Như thế, để nhận được véc-tơ xác suất đối với nguồn đã mở rộng này, chúng ta phải tạo ra một véc-tơ với 36 thành phần, mỗi một thành phần là tích của hai xác suất trong véc-tơ

xác suất gốc  $p$ . Điều này có thể làm được bằng cách áp dụng hàm MATLAB **kron.m** dưới dạng  $\text{kron}(p, p)$ . Các từ mã Huffman được cho như sau

1110000, 01110, 10110111, 1011001, 111001, 00101, 01111, 000, 011010, 00111, 1001, 1100, 11101110, 011011, 111011110, 111011111, 1110001, 001000, 1011010, 01100, 10110110, 1011000, 101110, 111110, 111010, 1010, 1110110, 101111, 11110, 0100, 00110, 1101, 001001, 111111, 0101, 1000

Độ dài từ mã trung bình đối với nguồn mở rộng này là 4.7420. *Entropy* của nguồn mở rộng tìm được bằng 4.7097, do vậy hiệu suất của mã Huffman này là

$$\eta_2 = \frac{4.7097}{4.7420} = 0.9932$$

mà nó cho thấy có một sự cải thiện so với hiệu suất của mã Huffman đã được thiết kế trong phần 2.

## BÀI TẬP MINH HỌA

**Bài tập minh họa 4.3 [Một mã Huffman với hiệu suất cực đại]** Hãy thiết kế một mã Huffman cho một nguồn có véc-tơ xác suất

$$p = \left\{ \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}, \frac{1}{256}, \frac{1}{256} \right\}$$

### Lời giải

Chúng ta sử dụng hàm **huffman.m** để xác định một mã Huffman và độ dài từ mã trung bình tương ứng. Các từ mã kết quả của tính toán là 1, 01, 001, 0001, 00001, 000001, 00000000 và 00000001. Độ dài từ mã trung bình tính được là 1.9922 symbol nhị phân/sản phẩm lối ra của nguồn. Nếu chúng ta tìm *entropy* của nguồn này bằng cách sử dụng hàm **entropy.m**, chúng ta sẽ thấy rằng *entropy* của nguồn này cũng là 1.9922 bit/sản phẩm lối ra của nguồn; do đó hiệu suất của mã này bằng 1.

### Câu hỏi

Bạn có thể nói được trong các điều kiện nào thì hiệu suất của một mã Huffman bằng 1 không?

## 4.3 LUỢNG TỬ HOÁ

Trong mục trước chúng ta đã nghiên cứu hai phương pháp nhằm mã hoá không tạp-đó là nén sản phẩm đầu ra của nguồn sao cho việc khôi phục đầy đủ từ các dữ liệu đã nén là có thể được. Trong các phương pháp này, dữ liệu đã nén là một hàm xác định của sản phẩm lối ra của nguồn và sản phẩm lối ra của nguồn là một hàm xác định của dữ liệu đã nén. Sự tương ứng một-một này giữa dữ liệu đã nén và sản phẩm lối ra của nguồn nghĩa là các *entropy* của chúng bằng nhau và không có thông tin nào bị mất trong quá trình mã hoá-giải mã.

Trong nhiều ứng dụng, như trong xử lý số các tín hiệu tương tự, trong đó bảng chữ cái của nguồn không phải là rời rạc, thì số các bít cần thiết để biểu diễn mỗi một sản phẩm lối ra của nguồn là không hữu hạn. Để xử lý sản phẩm lối ra của nguồn theo phương pháp số, nguồn phải được lượng tử hoá thành một số các mức hữu hạn. Quá trình này giảm số bít xuống thành một số hữu hạn song đồng thời lại mang vào một lượng méo nào đó. Lượng thông tin mất trong quá trình lượng tử hoá có thể không bao giờ khôi phục lại được.

Nói chung, các sơ đồ lượng tử hoá có thể phân loại thành các sơ đồ lượng tử hoá vô hướng và lượng tử hoá véc-tơ. Trong lượng tử hoá vô hướng, từng sản phẩm lối ra của nguồn được lượng tử hoá riêng rẽ, trong khi đó với lượng tử hoá véc-tơ thì việc lượng tử hoá được thực hiện trên các khối sản phẩm lối ra của nguồn.

Các bộ lượng tử hoá vô hướng còn có thể phân loại hơn nữa thành các bộ lượng tử hoá đều và các bộ lượng tử hoá không đều. Trong lượng tử hoá đều, các miền lượng tử được chọn có độ dài như nhau; trong lượng tử hoá không đều, các miền với nhiều độ dài khác nhau là được phép. Rõ ràng rằng, nói chung, các bộ lượng tử hoá không đều thì tốt hơn các bộ lượng tử hoá đều.

### 4.3.1 LUỢNG TỬ HOÁ VÔ HƯỚNG

Trong lượng tử hoá vô hướng, dải giá trị của biến ngẫu nhiên  $X$  được chia thành  $N$  miền  $R_i$  không chồng lấn lên nhau với  $1 \leq i \leq N$ , gọi là các khoảng lượng

tử hoá, và một điểm đơn gọi là một mức lượng tử được chọn trong mỗi một miền. Khi đó, mọi giá trị của biến ngẫu nhiên nói trên rơi vào trong miền  $R_i$  đều được lượng tử thành mức lượng tử thứ  $i$ , ký hiệu là  $\hat{x}_i$ . Điều này có nghĩa là

$$x \in R_i \Leftrightarrow Q(x) = \hat{x}_i \quad (4.3.1)$$

trong đó

$$\hat{x}_i \in R_i \quad (4.3.2)$$

Hiển nhiên, một việc lượng tử hoá kiểu này tạo ra một sai số trung bình bình phương là  $(x - \hat{x}_i)^2$ . Sai số lượng tử trung bình bình phương do đó được cho bởi

$$D = \sum_{i=1}^N \int_{R_i} (x - \hat{x}_i)^2 f_X(x) dx \quad (4.3.3)$$

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

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

### Lượng tử hoá đều

Trong lượng tử hoá đều, mọi miền lượng tử–ngoại trừ miền đầu tiên và miền cuối cùng–tức là miền  $R_1$  và  $R_N$ , đều có độ dài bằng nhau ký hiệu là  $\Delta$ ; do đó

$$R_1 = (-\infty, a]$$

$$R_2 = (a, a+\Delta]$$

$$R_3 = (a+\Delta, a+2\Delta]$$

...

$$R_N = (a+(N-2)\Delta, \infty)$$

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

$$\hat{x}_i = E[X | X \in R_i] = \frac{\int_{R_i} xf_X(x) dx}{\int_{R_i} f_X(x) dx}, \quad 1 \leq i \leq N \quad (4.3.4)$$

Do đó, việc thiết kế bộ lượng tử hoá đều tương đương với việc xác định  $a$  và  $\Delta$ . Sau khi  $a$  và  $\Delta$  đã được xác định, các giá trị của các  $\hat{x}_i$  và lượng méo dẫn xuất có thể xác định được dễ dàng bằng các biểu thức (4.3.3) và (4.3.4). Trong một số trường hợp, sẽ là thuận tiện hơn cả nếu chọn một cách đơn giản các điểm giữa của các miền lượng tử hoá làm các mức lượng tử hoá, nghĩa là tại khoảng cách  $\Delta/2$  tính từ các biên của các miền lượng tử.

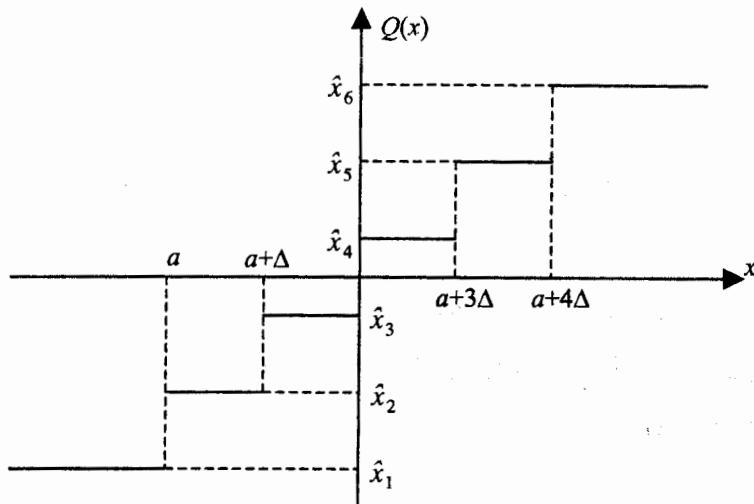
Các đồ thị của hàm lượng tử hoá  $Q(x)$  đối với một hàm mật độ xác suất đối xứng của  $X$  và các giá trị chẵn và lẻ của  $N$  được thể hiện trên các hình vẽ 4.3 và 4.4 một cách tương ứng.

Đối với các hàm mật độ xác suất đối xứng, bài toán thậm chí trở nên đơn giản hơn. Trong trường hợp như vậy,

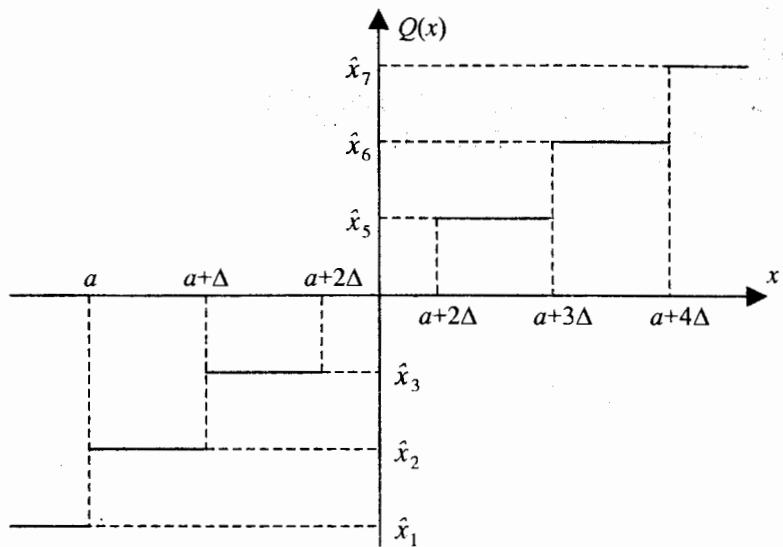
$$R_i = \begin{cases} (a_{i-1}, a_i], & 1 \leq i \leq N-1 \\ (a_{i-1}, a_N), & i = N \end{cases} \quad (4.3.5)$$

trong đó

$$\begin{cases} a_0 = -\infty \\ a_i = (i - N/2)\Delta & 1 \leq i \leq N-1 \\ a_N = \infty \end{cases} \quad (4.3.6)$$



Hình 4.3 Hàm lượng tử hoá đều đối với  $N=6$ . (Hãy chú ý rằng ở đây  $a+2\Delta = 0$ .)



Hình 4.4 Hàm lượng tử hoá đối với  $N=7$ . (Hãy chú ý rằng ở đây  $\hat{x}_4 = 0$ .)

Chúng ta thấy rằng trong trường hợp này chúng ta chỉ có một tham số  $\Delta$  phải lựa chọn để đạt được méo tối thiểu. Ba *m-file* – **centroid.m**, **mse\_dist.m** và **uq\_dist.m** – tìm trọng tâm của một miền, sai số lượng tử trung bình bình phương đối với một phân bố và các biên miền lượng tử hoá cho trước, và cuối cùng là lượng méo khi một bộ lượng tử đều được sử dụng để lượng tử hoá một nguồn cho trước. (Giả sử rằng các mức lượng tử được lấy là các trọng tâm của các miền lượng tử hoá). Để sử dụng mỗi một *m-file* này, phân bố của nguồn (mà nó có thể phụ thuộc đến 3 tham số) phải được cho trong một *m-file*. Các *m-file* này được cho dưới đây.

### **m-file**

```

function y=centroid(funfcn,a,b,tol,p1,p2,p3)
% CENTROID      Finds the centroid of a function over a region.
% Y=CENTROID('F',A,B,TOL,P1,P2,P3), finds the centroid of the
% function F defined in an m-file on the [A,B] region. The
% function can contain up to three parameters, P1, P2, P3.
% tol=the relative error.

args=[];
for n=1:nargin-4
    args=[args,',p',int2str(n)];
end

```

```
args=[args,')'];
funfcn1='x_fnct';
y1=eval(['quad(funfcn1,a,b,tol,[],funfcn,args]');
y2=eval(['quad(funfcn,a,b,tol,[],args]');
y=y1/y2;
```

### m-file

```
function [y,dist]=mse_dist(funfcn,a,tol,p1,p2,p3)
%MSE_DIST Returns the mean-squared quantization error.
% [Y,DIST]=MSE_DIST(FUNFCN,A,TOL,P1,P2,P3)
% funfcn=The distribution function given in an m-file. It can depend
% on up to three parameters, p1,p2,p3.
% a=the vector defining the boundaries of the quantization regions,
% (note: [a(1),a(length(a))] is the support of funfcn).
% p1,p2,p3=parameters of funfcn.
% tol=the relative error.
```

```
args=[];
for n=1:nargin-3
    args=[args,'p',int2str(n)];
end
args=[args,')'];
for i=1:length(a)-1
    y(i)=eval(['centroid(funfcn,a(i),a(i+1),tol,args]');
end
dist=0;
for i=1:length(a)-1
    newfun = 'x_a2_fnct' ;
    dist=dist+eval(['quad(newfun,a(i),a(i+1),tol,[],funfcn,', num2str(y(i)), ',args]');
end
```

### m-file

```
function [y,dist]=uq_dist(funfcn,b,c,n,delta,s,tol,p1,p2,p3)
%UQ_DIST Returns the distortion of a uniform quantizer
% with quantization points set to the centroids.
% [Y,DIST]=UQ_DIST(FUNFCN,B,C,N,DELTA,S,TOL,P1,P2,P3)
% funfcn=source density function given in an m-file
% with at most three parameters, p1,p2,p3.
% [b,c]=The support of the source density function.
% n=number of levels.
```

```

% delta=level size.
% s=the leftmost quantization region boundary.
% p1,p2,p3=parameters of the input function.
% y=quantization levels.
% dist=distortion.
% tol=the relative error.

if (c-b<delta*(n-2))
    error('Too many levels for this range.');?>
end
if (s<b)
    error('The leftmost boundary too small.');?>
end
if (s+(n-2)*delta>c)
    error('The leftmost boundary too large.');?>
end
args=[];
for j=1:nargin-7
    args=[args,' ',p,int2str(j)];
end
args=[args,')'];
a(1)=b;
for i=2:n
    a(i)=s+(i-2)*delta;
end
a(n+1)=c;
[y,dist]=eval(['mse_dist(funfcn,a,tol',args)]);

```

## BÀI TẬP MINH HỌA

**Bài tập minh họa 4.4 [Xác định các trọng tâm]** Hãy xác định các trọng tâm của các miền lượng tử hoá đối với một phân bố Gauss có kỳ vọng bằng 0 và phương sai bằng 1, trong đó các biên của các miền lượng tử hoá được cho bởi  $(-5, -4, -2, 0, 1, 3, 5)$ .

### Lời giải

Phân bố Gauss được cho trong *m-file* **normal.m**. Phân bố này là một hàm có hai tham số, là kỳ vọng và phương sai, được ký hiệu là  $m$  và  $s$  (hay  $\sigma$ ) một cách tương ứng. Miền xác định của phân bố Gauss là  $(-\infty, \infty)$  song để áp dụng các thủ

tục tính toán số thì sử dụng một dải rộng bằng nhiều lần độ lệch chuẩn phương của phân bố là đủ. Thí dụ dải  $(m-10\sqrt{s}, m+10\sqrt{s})$  có thể sử dụng được. *M-file* sau đây xác định các trọng tâm (các mức lượng tử tối ưu).

### m-file

% MATLAB script for Illustrative Problem 4, Chapter 4.

```
echo on;  
a=[-10,-5,-4,-2,0,1,3,5,10];  
for i=1:length(a)-1  
    y(i)=centroid('normal',a(i),a(i+1),0.001,0,1);  
    echo off;  
end
```

Việc tính toán này dẫn đến các mức lượng tử hoá sau:  $(-5.1865, -4.2168, -2.3706, -0.4599, 0.7228, 1.5101, 3.2827, 5.1865)$ .

### BÀI TẬP MINH HỌA

**Bài tập minh họa 4.5 [Sai số trung bình bình phương]** Trong bài tập 4.4, hãy xác định sai số trung bình bình phương.

#### Lời giải

Cho  $a=(-10, -5, -4, -2, 0, 1, 3, 5, 10)$  và sử dụng **mse\_dist.m**, chúng ta nhận được sai số trung bình bình phương là 0.177.

### BÀI TẬP MINH HỌA

**Bài tập minh họa 4.6 [Méo của bộ lượng tử hoá đều]** Hãy xác định sai số bình phương trung bình đối với một bộ lượng tử hoá đều với 12 mức lượng tử có độ dài bằng 1 được thiết kế cho một nguồn Gauss có kỳ vọng bằng 0, phương sai bằng 4. Giả sử rằng các miền lượng tử hoá đối xứng qua kỳ vọng của phân bố.

#### Lời giải

Do giả thiết về tính đối xứng, các biên của các miền lượng tử hoá là  $0, \pm 1, \pm 2, \pm 3, \pm 4$  và  $\pm 5$  còn các miền lượng tử hoá là  $(-\infty, -5], (-5, -4], (-4, -3], (-3, -2], (-2, -1], (-1, 0], (0, 1], (1, 2], (2, 3], (3, 4], (4, 5] \text{ và } (5, +\infty)$ . Điều này có

nghĩa là trong hàm **uq\_dist.m**, chúng ta có thể thay  $b=-20$ ,  $c=20$ ,  $\Delta=1$ ,  $n=12$ ,  $s=-5$ ,  $tol=0.001$ ,  $p_1=0$  và  $p_2=2$ . Thay các giá trị này vào **uq\_dist.m**, chúng ta nhận được một méo sai số bình phương bằng 0.0851 và các giá trị lượng tử bằng  $\pm 0.4897$ ,  $\pm 1.4691$ ,  $\pm 2.4487$ ,  $\pm 3.4286$ ,  $\pm 4.4089$  và  $\pm 5.6455$ .

*m-file uq\_mdpnt.m* xác định méo sai số bình phương đối với một hàm mật độ đối xứng khi các mức lượng tử được chọn là điểm giữa của các khoảng lượng tử. Trong trường hợp này, các mức lượng tử tương ứng với các miền lượng tử hoá đầu tiên và cuối cùng được chọn nằm cách hai biên lượng tử ngoài cùng một khoảng  $\Delta/2$ . Điều đó nghĩa là nếu số các mức lượng tử mà chẵn thì các biên lượng tử hoá là 0,  $\pm\Delta$ ,  $\pm 2\Delta, \dots, \pm(N/2-1)\Delta$  còn các mức lượng tử được cho bởi  $\pm\Delta/2$ ,  $\pm 3\Delta/2, \dots, \pm(N-1)\Delta/2$ . Nếu số các mức lượng tử mà lẻ thì các biên lượng tử hoá là  $\pm\Delta/2$ ,  $\pm 3\Delta/2, \dots, \pm(N/2-1)\Delta$  còn các mức lượng tử được cho bởi 0,  $\pm\Delta$ ,  $\pm 2\Delta, \dots, \pm(N-1)\Delta/2$ . *M-file uq\_mdpnt.m* được cho dưới đây.

### m-file

```
function dist=uq_mdpnt(funfcn,b,n,delta,tol,p1,p2,p3)
%UQ_MDPNT      Returns the distortion of a uniform quantizer
%           with quantization points set to the midpoints.
%           DIST=UQ_MDPNT(FUNFCN,B,N,DELTA,TOL,P1,P2,P3)
%           funfcn=source density function given in an m-file with at most three parameters,
%           p1,p2,p3. The density function is assumed to be an even function.
%           [-b,b]=The support of the source density function.
%           n=number of levels.
%           delta=level size.
%           p1,p2,p3=parameters of the input function.
%           dist=distortion.
%           tol=the relative error.

if (2*b<delta*(n-1))
    error('Too many levels for this range.');?>
end
args=[];
for j=1:nargin-5
    args=[args,'p',int2str(j)];
end
```

```
args=[args,')'];
a(1)=-b;
a(n+1)=b;
a(2)=-(n/2-1)*delta;
y(1)=a(2)-delta/2;
for i=3:n
    a(i)=a(i-1)+delta;
    y(i-1)=a(i)-delta/2;
end
y(n)=a(n)+delta;
dist=0;
for i=1:n
    newfun = 'x_a2_fnct' ;
    dist=dist+eval(['quad(newfun,a(i),a(i+1),tol,[],funfcn,', num2str(y(i)), args]);
end
```

### BÀI TẬP MINH HỌA

**Bài tập minh họa 4.7 [Bộ lượng tử hoá với các mức được đặt tại các điểm giữa]** Hãy tìm lượng méo khi sử dụng một bộ lượng tử hoá đều để lượng tử hoá một biến ngẫu nhiên Gauss có kỳ vọng bằng không và phương sai bằng một. Số các mức lượng tử hoá là 11 và độ dài của mỗi miền lượng tử hoá là 1.

#### Lời giải

Trong **uq\_mdpnt.m**, chúng ta thay ‘normal’<sup>2</sup> cho tên hàm mật độ,  $p_1=0$  và  $p_2=1$  đối với các tham số của hàm mật độ,  $n=11$  cho số các mức lượng tử và  $\Delta=1$  cho độ dài của các mức lượng tử. Đối với tham số  $b$ , mà nó xác định dải tính của hàm mật độ, chúng ta sử dụng giá trị  $b=10p_2=10$  và chọn độ phân giải (‘tol’: *tolerance*) là 0.001. Lượng méo tính được là 0.0833.

#### Lượng tử hoá không đều

Trong lượng tử hoá không đều, yêu cầu về các miền lượng tử hoá đều có độ dài như nhau, ngoại trừ các miền đầu và cuối, sẽ không đòi hỏi nữa và mỗi một miền có thể có một độ dài bất kỳ nào đó. Do trong trường hợp này việc tối ưu hoá được thực hiện với những điều kiện giảm nhẹ hơn, kết quả là sự vượt trội hiển nhiên

2) Tên hàm cần phải được thay bằng ‘normal’ với cả các dấu nháy.

so với lượng tử hoá đều.

Các điều kiện tối ưu đối với trường hợp này, gọi là *các điều kiện Lloyd-Max*, có thể biểu diễn được theo

$$\begin{cases} \hat{x}_i = \frac{\int_{a_{i-1}}^{a_i} xf_X(x)dx}{\int_{a_{i-1}}^{a_i} f_X(x)dx} \\ a_i = \frac{(\hat{x}_{i-1} + \hat{x}_i)}{2} \end{cases} \quad (4.3.7)$$

Từ những phương trình này, chúng ta kết luận rằng các mức lượng tử hoá tối ưu là trọng tâm của các miền lượng tử hoá và các biên tối ưu giữa các miền lượng tử hoá là các điểm giữa giữa các mức lượng tử hoá. Để có được lời giải đối với các phương trình Lloyd-Max, chúng ta bắt đầu với tập các mức lượng tử hoá  $\hat{x}_i$ . Từ tập này, chúng ta có thể tìm được một cách đơn giản tập các biên miền lượng tử hoá  $a_i$ . Từ tập các  $a_i$  này, có thể nhận được một tập mới các mức lượng tử hoá. Quá trình này được tiếp tục cho đến khi việc cải thiện về méo từ bước lặp này sang bước lặp khác không thể nhận ra được. Thuật toán này bảo đảm được là sẽ hội tụ tới một trị tối thiểu cục bộ song nói chung thì không bảo đảm có được trị tối thiểu toàn bộ.

Thủ tục thiết kế một bộ lượng tử hoá tối ưu được trình bày trong *m-file* **lloydmax.m** được cho dưới đây.

### **m-file**

```
function [a,y,dist]=lloydmax(funfcn,b,n,tol,p1,p2,p3)
%LLOYDMAX      Returns the the Lloyd-Max quantizer and the mean-squared
%               quantization error for a symmetric distribution.
% [A,Y,DIST]=LLOYDMAX(FUNFCN,B,N,TOL,P1,P2,P3)
% funfcn=The density function given in an m-file. It can depent on up to
% three parameters, p1,p2,p3.
% a=The vector giving the boundaries of the quantization regions.
% [-b,b] approximates support of the density function.
% n=The number of quantization regions.
% y=The quantization levels.
% p1,p2,p3=Parameters of funfcn.
```

% tol=the relative error.

```
args=[];
for j=1:nargin-4
    args=[args,'p',int2str(j)];
end
args=[args,')'];
v=eval(['variance(funfcn,-b,b,tol',args]);
a(1)=-b;
d=2*b/n;
for i=2:n
    a(i)=a(i-1)+d;
end
a(n+1)=b;
dist=v;
[y,newdist]=eval(['mse_dist(funfcn,a,tol',args]);
while(newdist<0.99*dist),
    for i=2:n
        a(i)=(y(i-1)+y(i))/2;
    end
    dist=newdist;
    [y,newdist]=eval(['mse_dist(funfcn,a,tol',args]);
end
```

### BÀI TẬP MINH HỌA

**Bài tập minh họa 4.8 [Thiết kế bộ lượng tử hoá Lloyd-Max]** Hãy thiết kế một bộ lượng tử hoá Lloyd-Max 10 mức cho một nguồn Gauss có kỳ vọng bằng không và phương sai bằng 1.

#### Lời giải

Sử dụng  $b = 10$ ,  $n = 10$ ,  $\text{tol} = 0.01$ ,  $p_1 = 0$ ,  $p_2 = 1$  trong **lloydmax.m**, chúng ta nhận được véc-tơ các biên lượng tử hoá và mức lượng tử hoá  $a$  và  $y$  là

$$a = \pm 10, \pm 2.16, \pm 1.51, \pm 0.98, \pm 0.48, 0$$

$$y = \pm 2.52, \pm 1.78, \pm 1.22, \pm 0.72, \pm 0.24$$

và lượng méo tính được là 0.02. Các giá trị này là các giá trị xấp xỉ rất gần với các giá trị đã được Max cho trong bảng [2].

## 4.4 ĐIỀU CHẾ MÃ XUNG (PCM)

Trong điều chế mã xung, một tín hiệu tương tự trước hết được lấy mẫu với một tốc độ lớn hơn tốc độ Nyquist rồi sau đó các mẫu được lượng tử hoá. Giả sử rằng tín hiệu tương tự được phân bố trên một khoảng ký hiệu là  $[-x_{\max}, x_{\max}]$  và số các mức lượng tử lớn. Các mức lượng tử có thể cách đều nhau hoặc cách nhau không đều. Trong trường hợp thứ nhất chúng ta làm việc với một PCM lượng tử hoá đều còn trong trường hợp sau thì với một PCM lượng tử hoá không đều.

### 4.4.1 PCM VỚI LƯỢNG TỬ HOÁ ĐỀU

Trong PCM lượng tử hoá đều, khoảng  $[-x_{\max}, x_{\max}]$  có độ dài  $2x_{\max}$  được chia thành  $N$  khoảng con bằng nhau, mỗi khoảng có độ dài  $\Delta = 2x_{\max}/N$ . Nếu  $N$  đủ lớn thì hàm mật độ của tín hiệu lối vào trên mỗi một khoảng con có thể giả sử được là hàm mật độ xác suất phân bố đều, dẫn đến một lượng méo  $D = \Delta^2/12$ . Nếu  $N$  là một luỹ thừa của 2, hay  $N = 2^v$  thì cần đến  $v$  bít để biểu diễn (mã) mỗi một mức. Điều này có nghĩa là độ rộng băng để truyền dẫn tín hiệu PCM tối thiểu là  $vW$ . (Trong thực tế,  $1.5 vW$  là sát với khả năng hiện thực). Lượng méo được cho bởi

$$D = \frac{\Delta^2}{12} = \frac{x_{\max}^2}{3N^2} = \frac{x_{\max}^2}{3 \times 4^v} \quad (4.4.1)$$

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

$$SQNR = 3N^2 \frac{\overline{X^2}}{x_{\max}^2} = 3 \times 4^v \frac{\overline{X^2}}{x_{\max}^2} = 3 \times 4^v \overline{\check{X}^2} \quad (4.4.2)$$

trong đó  $\check{X}$  ký hiệu cho tín hiệu lối vào chuẩn hoá được xác định theo

$$\check{X} = \frac{X}{x_{\max}}$$

SQNR tính theo dB được cho bởi

$$SQNR|_{\text{dB}} \approx 4.8 + 6v + \overline{\check{X}^2}|_{\text{dB}} \quad (4.4.3)$$

Sau lượng tử hoá, các mức lượng tử được mã hoá bằng  $v$  bit đối với mỗi một mức đã lượng tử hoá. Số đố mã hoá thường được áp dụng là mã hoá nhị phân tự nhiên (NBC: *Natural Binary Coding*), có nghĩa là mức thấp nhất được ánh xạ thành một dãy gồm toàn bít 0 và mức cao nhất được ánh xạ thành một dãy gồm toàn bít 1. Mọi mức khác được ánh xạ theo thứ tự tăng dần của giá trị đã lượng tử hoá.

*M-file u\_pcm.m* được cho dưới đây nhận làm số liệu đầu vào của mình một chuỗi các giá trị đã lấy mẫu và số mức lượng tử hoá mong muốn rồi tìm dãy đã được lượng tử hoá, chuỗi bít đã mã và SQNR hậu quả (tính theo dB).

### m-file

```

function [sqnr,a_quan,code]=u_pcm(a,n)
%U_PCM Uniform PCM encoding of a sequence.
% [SQNR,A_QUAN,CODE]=U_PCM(A,N)
% a=input sequence.
% n=number of quantization levels (even).
% sqnr=output SQNR (in dB).
% a_quan=quantized output before encoding.
% code=the encoded output.

amax=max(abs(a));
a_quan=a/amax;
b_quan=a_quan;
d=2/n;
q=d.*[0:n-1];
q=q-((n-1)/2)*d;
for i=1:n
    a_quan(find((q(i)-d/2 <= a_quan) & (a_quan <= q(i)+d/2)))=...
    q(i).*ones(1,length(find((q(i)-d/2 <= a_quan) & (a_quan <= q(i)+d/2))));
    b_quan(find( a_quan==q(i) ))=(i-1).*ones(1,length(find( a_quan==q(i) )));
end
a_quan=a_quan*amax;
nu=ceil(log2(n));
code=zeros(length(a),nu);
for i=1:length(a)
    for j=nu:-1:0
        if ( fix(b_quan(i)/(2^j)) == 1 )
            code(i,(nu-j)) = 1;
            b_quan(i) = b_quan(i) - 2^j;
        end
    end
end

```

```

end
end
sqnr=20*log10(norm(a)/norm(a-a_quan));

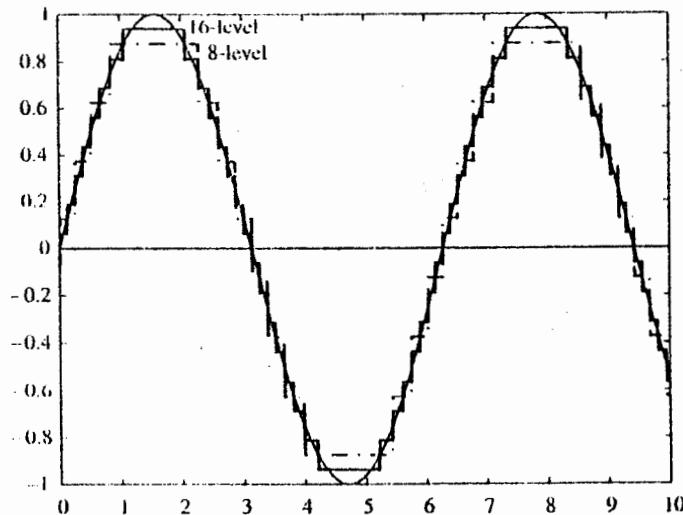
```

### BÀI TẬP MINH HỌA

**Bài tập minh họa 4.9 [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, hãy lượng tử hoá nó một lần thành 8 mức và một lần thành 16 mức. Hãy vẽ đồ thị tín hiệu gốc và các tín hiệu đã lượng tử hoá trên cùng một hệ toạ độ. Hãy so sánh SQNR hậu quả trong hai trường hợp.

#### Lời giải

Chúng ta chọn một cách tuỳ tiện thời gian tồn tại của tín hiệu là 10 s. Sau đó, sử dụng *m-file u\_pcm.m* tạo ra các tín hiệu đã lượng tử hoá cho hai trường hợp 8 và 16 mức lượng tử. Các SQNR tính được là 18.90 dB đối với PCM lượng tử hoá đều 8 mức và 25.13 dB đối với PCM lượng tử hoá đều 16 mức. Các đồ thị được trình bày trên hình 4.5.



Hình 4.5 PCM lượng tử hoá đều đối với một tín hiệu hình sin dùng 8 và 16 mức.

Một MATLAB *script* cho bài tập này được cho dưới đây.

### m-file

% MATLAB script for Illustrative Problem 9, Chapter 4

```
echo on
t=[0:0.01:10];
a=sin(t);
[sqnr8,aquan8,code8]=u_pcm(a,8);
[sqnr16,aquan16,code16]=u_pcm(a,16);
pause % Press a key to see the SQNR for N = 8.
sqnr8
pause % Press a key to see the SQNR for N = 16.
sqnr16
pause % Press a key to see the plot of the signal and its quantized versions.
plot(t,a,'-',t,aquan8,'-',t,aquan16,'-',t,zeros(1,length(t)))
```

## BÀI TẬP MINH HỌA

**Bài tập minh họa 4.10 [PCM lượng tử hoá đều]** Hãy tạo một dãy có độ dài 500, gồm các biến ngẫu nhiên Gauss có kỳ vọng bằng 0 và phương sai bằng 1. Sử dụng **u\_pcm.m**, hãy tìm SQNR hậu quả khi số mức lượng tử hoá là 64. Hãy tìm 5 giá trị đầu tiên của dãy, các giá trị đã lượng tử tương ứng và các từ mã tương ứng.

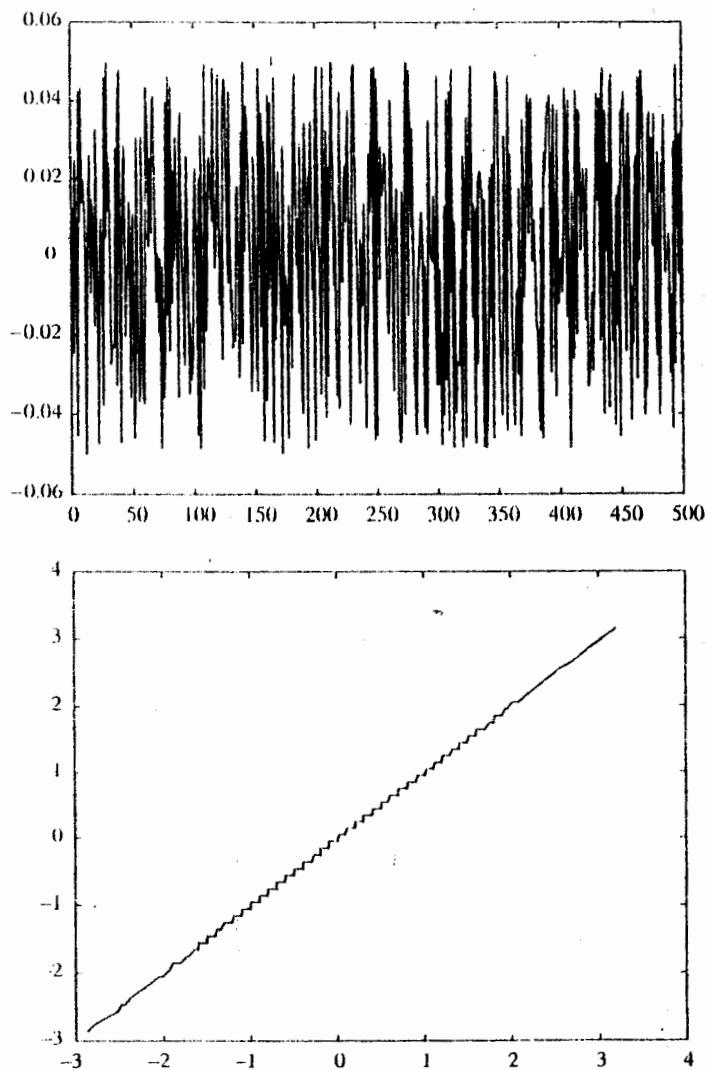
### Lời giải

M-file sau đây cho chúng ta đáp án.

### m-file

% MATLAB script for Illustrative Problem 10, Chapter 4

```
echo on
a=randn(1,500);
n=64;
[sqnr,a_quan,code]=u_pcm(a,64);
pause % Press a key to see the SQNR
sqnr
pause % Press a key to see the first five input values.
a(1:5)
pause % Press a key to see the first five quantized values.
a_quan(1:5)
pause % Press a key to see the first five codewords.
code(1:5,:)
```



Hình 4.6 Sai số lượng tử trong PCM lượng tử hoá đều với 64 mức lượng tử.

Trong một lần chạy điển hình của file này các giá trị sau đây đã quan sát được:

$$\text{SQNR} = 31.66 \text{ dB}$$

Tín hiệu lõi 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]

$$\text{Các từ mã} = \begin{cases} 100001 \\ 011011 \\ 101010 \\ 001010 \\ 100101 \end{cases}$$

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

### BÀI TẬP MINH HỌA

**Bài tập minh họa 4.11 [Sai số lượng tử]** Trong bài tập minh họa 4.10, hãy vẽ đồ thị sai số lượng tử được xác định như sai lệch giữa giá trị tín hiệu đầu vào và giá trị sau lượng tử hoá. Hãy vẽ cả đồ thị của giá trị  $\Delta$  lượng tử hoá như một hàm của giá trị lối vào.

#### Lời giải

Hai đồ thị cần vẽ được trình bày trên hình 4.6.

### BÀI TẬP MINH HỌA

**Bài tập minh họa 4.12 [Sai số lượng tử]** Lặp lại bài tập minh họa 4.11 với số các mức lượng tử một lần được đặt bằng 16 và một lần được đặt bằng 128. Hãy so sánh các kết quả.

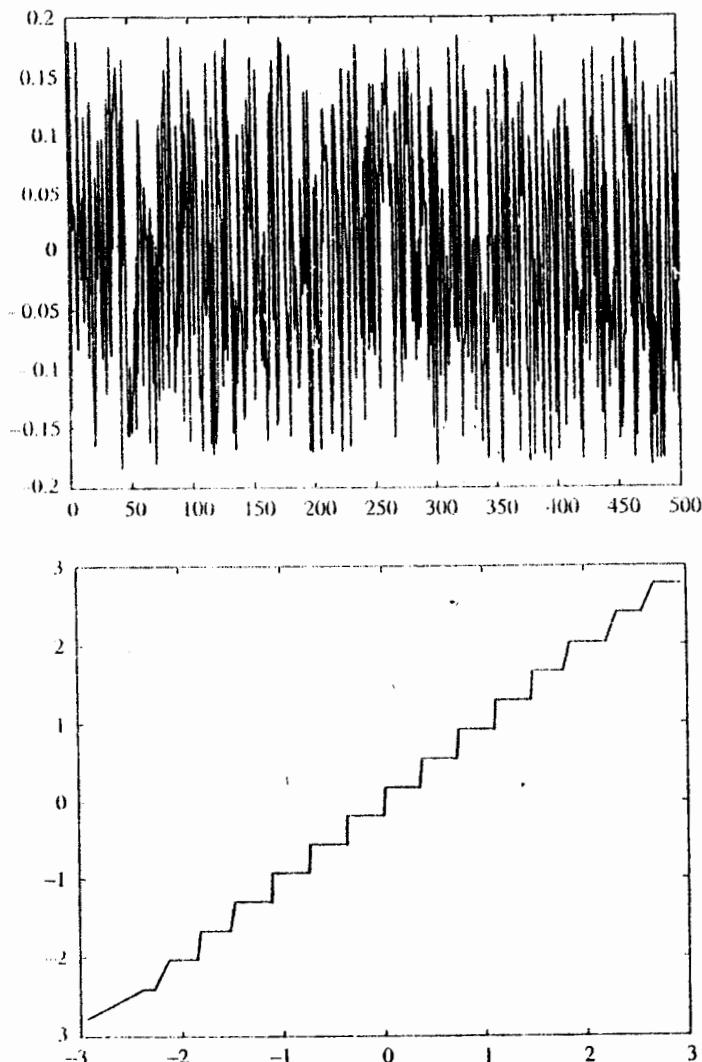
#### Lời giải

Kết quả đối với 16 mức lượng tử được thể hiện trên hình 4.7 còn kết quả đối với 128 mức lượng tử thấy được trên hình 4.8. Từ các hình vẽ 4.6, 4.7 và 4.8 rõ ràng là như ta đã nghĩ, số mức lượng tử càng lớn thì sai số lượng tử càng nhỏ. Cũng chú ý rằng đối với số các mức lượng tử hoá lớn, quan hệ giữa tín hiệu lối vào và các giá trị  $\Delta$  lượng tử hoá tiến tới một đường thẳng có độ dốc bằng 1 chạy qua gốc toạ độ; nghĩa là tín hiệu lối vào và các giá trị  $\Delta$  lượng tử hoá trở nên hầu như bằng nhau. Đối với số các mức lượng tử hoá bé (thí dụ là 16) thì quan hệ này là khác nhau xa như thấy được trên hình 4.7.

### 4.4.2 PCM VỚI LUỢNG TỬ HOÁ KHÔNG ĐỀU

Trong PCM lượng tử hoá không đều, tín hiệu lối vào trước hết được cho qua một phần tử phi tuyến nhằm giảm bớt dải động của nó rồi tín hiệu lối ra được đưa vào một hệ thống PCM với lượng tử hoá đều. Tại đầu thu, tín hiệu lối ra được cho qua bộ nghịch đảo của phần tử phi tuyến đã được sử dụng ở phần phát. Tác động

tổng cộng thì tương đương với một hệ thống PCM có các mức lượng tử được đặt cách nhau không đều. Nói chung, để truyền tín hiệu tiếng nói, các đặc tính phi tuyến được áp dụng là các đặc tính phi tuyến hoặc luật A hoặc luật  $\mu$ .



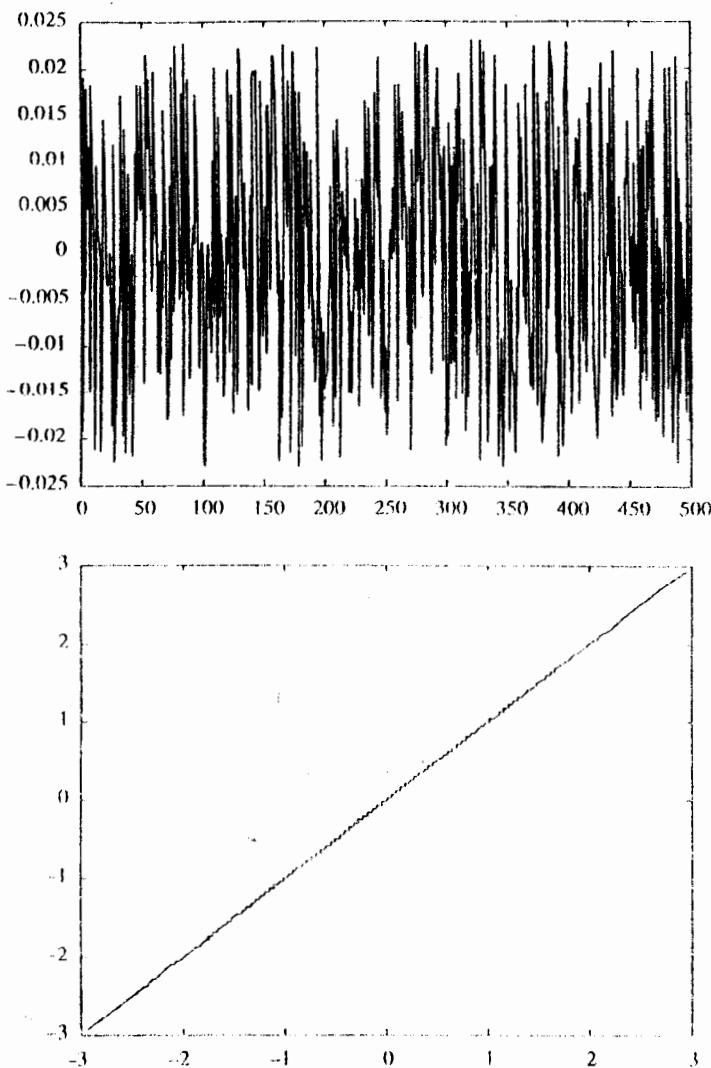
Hình 4.7 Sai số lượng tử đối với 16 mức lượng tử hoá.

Một đặc tính phi tuyến luật  $\mu$  được xác định bởi quan hệ

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

trong đó  $x$  là tín hiệu lối vào chuẩn hoá ( $|x| \leq 1$ ) còn  $\mu$  là một tham số mà đối

với đặc tính phi tuyến luật  $\mu$  tiêu chuẩn thì bằng 255. Một đồ thị của đặc tính phi tuyến này với các giá trị khác nhau của  $\mu$  thấy được trên hình 4.9.

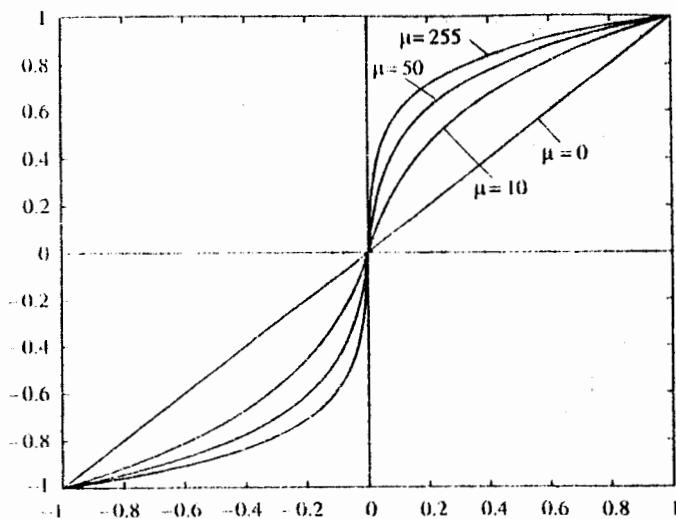


Hình 4.8 Sai số lượng tử đối với 128 mức lượng tử hoá.

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

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

Hai *m-file* **mulaw.m** và **invmulaw.m** được cho dưới đây thi hành đặc tính phi tuyến luật  $\mu$  và nghịch đảo của nó.



Hình 4.9 Bộ nén luật  $\mu$ .

### m-file

```
function [y,a]=mulaw(x,mu)
%MULAW           mu-law nonlinearity for non-uniform PCM.
%          Y=MULAW(X,MU)
%          X=input vector.

a=max(abs(x));
y=(log(1+mu*abs(x/a))./log(1+mu)).*signum(x);
```

### m-file

```
function x=invmulaw(y,mu)
%INVMULAW        The inverse of mu-law nonlinearity
%X=INVMULAW(Y,MU)    Y=Normalized output of the mu-law nonlinearity

x=((((1+mu).^(abs(y))-1)./mu).*signum(y));
```

*M-file mula\_pcm.m* thì tương đương với *m-file u\_pcm.m* khi sử dụng một sơ đồ PCM luật  $\mu$ . File này được cho dưới đây.

### m-file

```
function [sqnr,a_quan,code]=mula_pcm(a,n,mu)
```

```
%MULA_PCM      mu-law PCM encoding of a sequence.
% [SQNR,A_QUAN,CODE]=MULA_PCM(A,N,MU)
% a=input sequence.
% n=number of quantization levels (even).
% sqnr=output SQNR (in dB).
% a_quan=quantized output before encoding.
% code=the encoded output.

[y,maximum]=mulaw(a,mu);
[sqnr,y_q,code]=u_pcm(y,n);
a_quan=invmulaw(y_q,mu);
a_quan=maximum*a_quan;
sqnr=20*log10(norm(a)/norm(a-a_quan));
```

### BÀI TẬP MINH HỌA

**Bài tập minh họa 4.13 [PCM lượng tử hoá không đều]** Hãy tạo một dãy 500 biến ngẫu nhiên theo một phân bố  $N(0, 1)$ . Khi sử dụng 16, 64 và 128 mức lượng tử và một đặc tính phi tuyến luật  $\mu$  với  $\mu=255$ , hãy vẽ đồ thị sai số và quan hệ vào-ra đối với bộ lượng tử hoá trong từng trường hợp. Hãy xác định cả SQNR trong từng trường hợp.

#### Lời giải

Đặt véc-tơ  $a$  là véc-tơ có độ dài 500 tạo được theo  $N(0, 1)$ ; nghĩa là ta hãy lấy

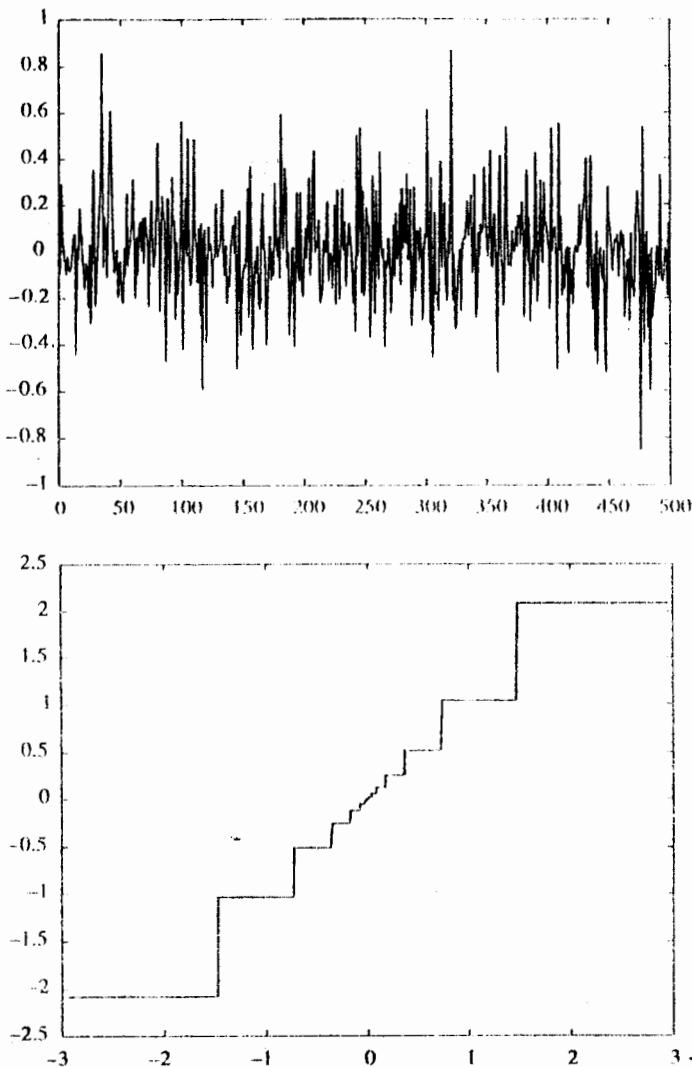
$$a = randn(1, 500)$$

Sau đó, bằng cách sử dụng

$$[dist,a_quan,code] = mula_pcm(a, 16, 255)$$

chúng ta có thể nhận được dãy đã lượng tử hoá và SQNR đối với lượng tử hoá 16 mức. SQNR sẽ là 13.76 dB. Đối với trường hợp 64 mức chúng ta nhận được SQNR=25.89 dB còn đối với 128 mức ta có SQNR=31.76 dB. So sánh các kết quả này với PCM lượng tử hoá đều chúng ta nhận thấy rằng trong mọi trường hợp chất lượng đều kém hơn PCM lượng tử hoá đều. Các đồ thị của quan hệ vào-ra đối với bộ lượng tử hoá và sai số lượng tử được cho trên các hình vẽ 4.10, 4.11 và 4.12.

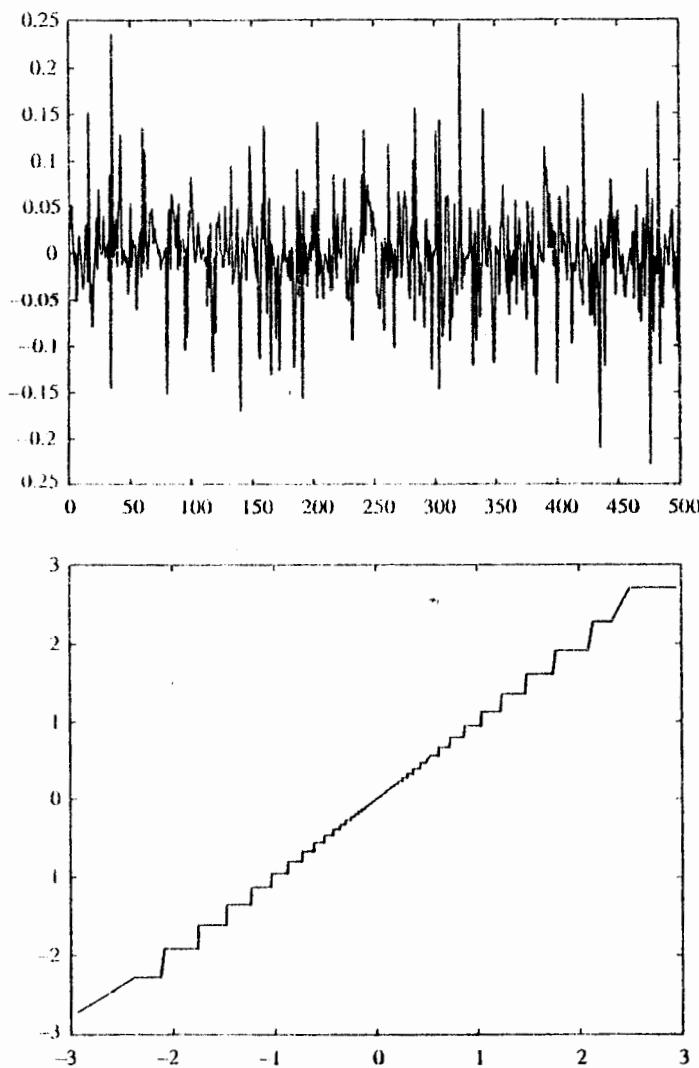
Việc so sánh quan hệ vào-ra đối với PCM lượng tử hoá đều và lượng tử hoá không đều, thấy được trên các hình vẽ 4.7 và 4.10, cho thấy rõ tại sao loại thứ nhất lại được gọi là PCM lượng tử hoá đều còn loại sau thì được gọi là PCM lượng tử hoá không đều.



Hình 4.10 Sai số lượng tử và quan hệ vào-ra bộ lượng tử hoá đối với một PCM luật  $\mu$  16 mức

Từ ví dụ trên đây, chúng ta thấy rằng chất lượng hoạt động của PCM lượng tử hoá không đều, trong trường hợp này, lại không tốt bằng PCM lượng tử hoá đều. Lý do là ở chỗ trong thí dụ đó dải động của tín hiệu lối vào không rất lớn.

Thí dụ tiếp theo sẽ khảo sát trường hợp trong đó chất lượng của PCM lượng tử hoá không đều thì vượt trội so với chất lượng của PCM lượng tử hoá đều.

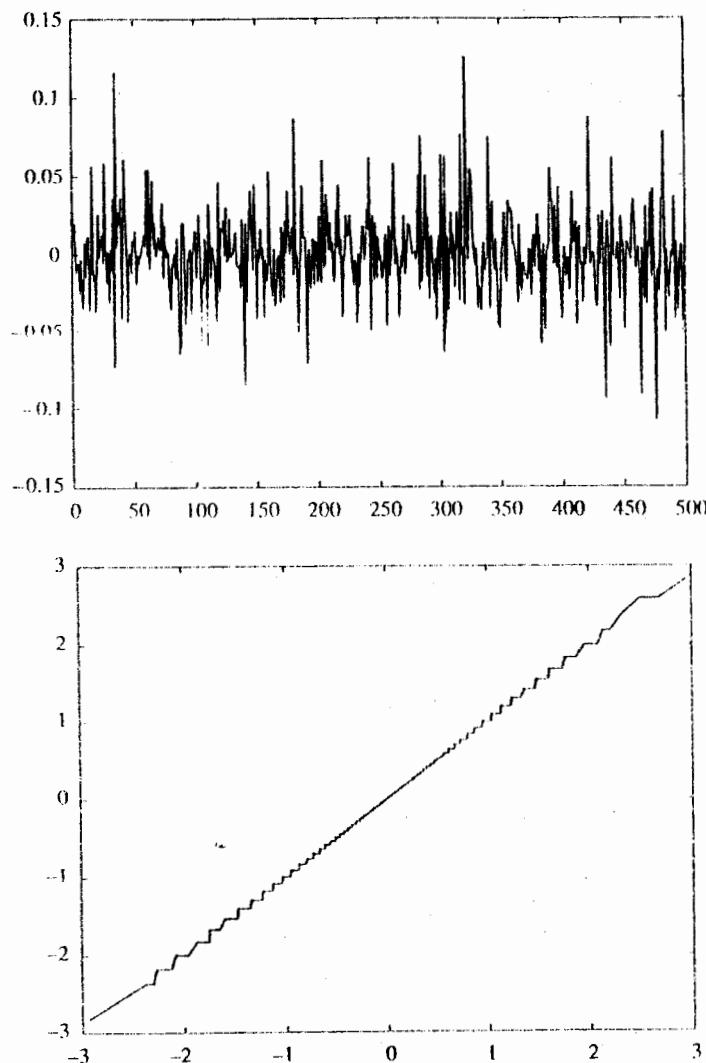


Hình 4.11 Sai số lượng tử và quan hệ vào-ra bộ lượng tử hoá đối với một PCM luật  $\mu$  64 mức

### BÀI TẬP MINH HỌA

**Bài tập minh họa 4.14 [PCM lượng tử hoá không đều]** Dãy không dừng  $a$  độ dài 500 bao gồm hai phần. Hai mươi mẫu đầu tiên được tạo ra theo một biến ngẫu nhiên Gauss với kỳ vọng bằng 0 và phương sai bằng 400 ( $\sigma=20$ ), 480 mẫu

tiếp sau được đổ ra theo một biến ngẫu nhiên Gauss với kỳ vọng 0 và phương sai bằng 1. Dãy này được lượng tử hoá lần đầu bằng một sơ đồ PCM lượng tử hoá đều và lần sau sử dụng một sơ đồ PCM lượng tử hoá không đều. Hãy so sánh kết quả SQNR trong hai trường hợp đó.



Hình 4.12 Sai số lượng tử và quan hệ vào-ra bộ lượng tử hoá đối với một PCM luật  $\mu$  128 mức

### Lời giải

Dãy tín hiệu được tạo ra bằng lệnh MATLAB

`a=[20*randn(1, 20)randn(1, 480)]`

Bây giờ chúng ta có thể áp dụng các file **u\_pcm.m** và **mula\_pcm.m** để xác định SQNR. Các SQNR là 20.49 dB và 24.95 dB một cách tương ứng. Trong trường hợp này, chất lượng của PCM lượng tử hoá không đều rõ ràng vượt hơn chất lượng của PCM lượng tử hoá đều.

## CÁC BÀI TẬP

**4.1** Hãy thiết kế một mã Huffman cho một nguồn thông tin với các xác suất

$$p = \{0.1, 0.05, 0.21, 0.07, 0.02, 0.2, 0.15\}$$

Hãy xác định hiệu suất của mã này bằng cách tính độ dài từ mã trung bình và *entropy* của nguồn.

**4.2** Một nguồn thông tin không nhớ rời rạc được mô tả bằng véc-tơ xác suất  $p = \{0.2, 0.3, 0.1, 0.4\}$ .

a. Hãy viết một file MATLAB để tính các xác suất của mở rộng thứ  $K$  của nguồn này đối với một  $K$  đã cho.

b. Hãy thiết kế các mã Huffman cho nguồn này và cho các mở rộng thứ  $K$  của nó, với  $K=1, 2, 3, 4, 5$ .

c. Hãy vẽ đồ thị độ dài từ mã trung bình (tính trên một sản phẩm lối ra của nguồn) như là một hàm của  $K$ .

**4.3** Các xác suất của các con chữ của bảng chữ cái xuất hiện trong tiếng Anh ẩn loát được cho trong bảng 4.1.

a. Hãy xác định *entropy* của tiếng Anh ẩn loát.

b. Hãy thiết kế một mã Huffman cho tiếng Anh ẩn loát.

c. Hãy xác định độ dài từ mã trung bình và hiệu quả của mã Huffman đó.

**4.4** Lặp lại bài tập 4.2 cho một nguồn không nhớ rời rạc với một véc-tơ xác suất  $p = \{0.5, 0.25, 0.125, 0.125\}$ . Hãy giải thích tại sao kết quả trong trường hợp này lại khác kết quả nhận được trong bài tập 4.2.

**4.5** Một nguồn thông tin liên tục có phân bố Gauss với kỳ vọng bằng không và phương sai bằng 1. Nguồn này được lượng tử hoá bằng cách sử dụng một bộ

lượng tử hoá đều đối xứng, trong đó độ dài của các miền lượng tử hoá bằng 1. Số các mức lượng tử hoá là  $N$ . Đối với  $N=2, 3, 4, 5, 6, 7, 8$ , hãy xác định *entropy* của lối ra nguồn đã được lượng tử hoá và vẽ nó như một hàm của  $N$ . Trên một đồ thị khác nào đó, hãy vẽ  $\log_2 N$  theo  $N$  và hãy giải thích tại sao hai đường cong đồ thị trên lại khác nhau.

Bảng 4.1 Các xác suất của các chữ cái trong tiếng Anh ẩn loát.

Chữ cái	Xác suất
A	0.0642
B	0.0127
C	0.0218
D	0.0317
E	0.1031
F	0.0208
G	0.0152
H	0.0467
I	0.0575
J	0.0008
K	0.0049
L	0.0321
M	0.0198
N	0.0574
O	0.0632
P	0.0152
Q	0.0008
R	0.0484
S	0.0514
T	0.0796
U	0.0228
V	0.0083
W	0.0175
X	0.0013
Y	0.0164
Z	0.0005
Khoảng trống giữa các từ	0.1859

**4.6** Một nguồn Gauss có kỳ vọng 0 và phương sai 1 được lượng tử hoá bằng một bộ lượng tử hoá đều. Bộ lượng tử hoá lượng tử đều khoảng  $[-10, 10]$ . Giả sử các mức lượng tử hoá được đặt tại các điểm giữa của các miền lượng tử hoá, hãy xác

định và vẽ đồ thị méo trung bình bình phương đối với  $N=3, 4, 5, 6, 7, 8, 9, 10$  như một hàm của  $N$  là số mức lượng tử.

**4.7** Trên cùng một hình vẽ mà bạn đã vẽ trong bài tập 4.6, hãy vẽ đồ thị méo trung bình bình phương khi các mức lượng tử được lấy là các trọng tâm của các miền lượng tử hoá. Với những giá trị như thế nào của  $N$  thì hai đồ thị này gần nhau hơn và tại sao?

**4.8** Đối với một nguồn Gauss có kỳ vọng bằng không, phương sai bằng 1, hãy thiết kế các bộ lượng tử hóa không đều tối ưu với số các mức  $N=2, 3, 4, 5, 6, 7, 8$ . Đối với mỗi trường hợp, hãy xác định  $H(\hat{X})$  là *entropy* của nguồn đã được lượng tử hóa, và hãy xác định  $R$  là độ dài từ mã trung bình của một mã Huffman được thiết kế cho nguồn đó. Hãy vẽ đồ thị  $H(\hat{X}), R$  và  $\log_2 N$  như là hàm của  $N$  trên cùng một hình vẽ.

**4.9** Một biến ngẫu nhiên Laplace được xác định bằng hàm mật độ xác suất

$$f(x) = \frac{\lambda}{2} e^{-\lambda|x|}$$

trong đó  $\lambda > 0$  là một hằng số cho trước.

- a. Hãy chứng tỏ rằng phương sai của một biến ngẫu nhiên Laplace thì bằng  $2/\lambda^2$ .
- b. Giả sử  $\lambda=1$ , hãy thiết kế các bộ lượng tử hóa đều với  $N=2, 3, 4, 5, 6, 7, 8$  mức cho nguồn này. Như thông lệ, hãy lấy khoảng quan tâm là  $[-10\sigma, 10\sigma]$ , trong đó  $\sigma$  là độ lệch chuẩn phương của nguồn đó.
- c. Hãy vẽ đồ thị *entropy* của nguồn đã lượng tử hóa và  $\log_2 N$  như các hàm của  $N$  trên cùng một hình vẽ.

**4.10** Lặp lại bài tập 4.9 song thay bộ lượng tử hóa đều bằng bộ lượng tử hóa không đều tối ưu.

**4.11** Hãy thiết kế một bộ lượng tử hóa tối ưu 8 mức cho một nguồn Laplace và vẽ đồ thị lượng méo trung bình bình phương tính được như một hàm của  $\lambda$  khi  $\lambda$  thay đổi trong khoảng  $[0.1, 5]$ .

**4.12** Hãy thiết kế các bộ lượng tử hoá không đều tối ưu với  $N=2, 3, 4, 5, 6, 7, 8$  cho nguồn Laplace đã cho trong bài tập 4.9 với  $\lambda=\sqrt{2}$ . (Hãy chú ý rằng việc chọn như thế sẽ dẫn đến một nguồn Laplace có kỳ vọng bằng không và phương sai bằng 1). Hãy vẽ đồ thị sai số trung bình bình phương như một hàm của  $N$  đối với nguồn này. Hãy so sánh các kết quả này với các kết quả nhận được từ việc lượng tử hoá một nguồn Gauss cũng có kỳ vọng bằng không và phương sai bằng 1.

**4.13** Tín hiệu tuần hoàn  $x(t)$  có chu kỳ bằng 2 và trên khoảng  $[0, 2]$  được xác định theo

$$x(t) = \begin{cases} t, & 0 \leq t < 1 \\ -t + 2 & 1 \leq t < 2 \end{cases}$$

a. Hãy thiết kế một bộ lượng tử hoá PCM đều 8 mức cho tín hiệu này và hãy vẽ đồ thị tín hiệu lỗi ra để lượng tử hoá của hệ thống này.

b. Hãy vẽ đồ thị sai số lượng tử cho hệ thống này.

c. Bằng cách tính công suất của tín hiệu sai số, hãy xác định SQNR tính theo dB cho hệ thống này.

d. Lặp lại các phần a, b và c với việc sử dụng một hệ thống PCM lượng tử hoá đều 16 mức.

**4.14** Hãy tạo một dãy Gauss với kỳ vọng bằng không và phương sai bằng một có 1000 phần tử. Hãy thiết kế các sơ đồ PCM lượng tử hoá đều 4 mức, 8 mức, 16 mức, 32 mức và 64 mức cho dãy này rồi vẽ SQNR hậu quả tính theo dB như một hàm của số bít được ấn định cho mỗi sản phẩm lỗi ra của nguồn.

**4.15** Hãy tạo một dãy Gauss kỳ vọng 0, phương sai 1 với độ dài 1000 và hãy lượng tử hoá nó bằng cách sử dụng một sơ đồ PCM lượng tử hoá đều với 6 bit/symbol. 6000 bít nhận được được truyền tới máy thu qua một kênh có tạp nhiễu. Xác suất lỗi của của kênh được ký hiệu là  $p$ . Hãy vẽ đồ thị SQNR tổng cộng tính theo dB như một hàm của  $p$  đối với các giá trị  $p=10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}, 0.1, 0.2$ . Để mô phỏng tác động của tạp nhiễu, bạn có thể tạo các dãy ngẫu nhiên nhị phân với các xác suất này và cộng chúng (theo modulo 2) vào dãy đã được mã hoá.

- 4.16** Lặp lại bài tập 4.13 bằng cách sử dụng một PCM lượng tử hoá không đều luật  $\mu$  với  $\mu=255$ .
- 4.17** Lặp lại bài tập 4.14 bằng cách sử dụng một PCM lượng tử hoá không đều luật  $\mu$  với  $\mu=255$ .
- 4.18** Lặp lại bài tập 4.15 bằng cách sử dụng một PCM lượng tử hoá không đều luật  $\mu$  với  $\mu=255$ .

## CHƯƠNG 5

# TRUYỀN DẪN TÍN HIỆU SỐ BẰNG GỐC

### 5.1 GIỚI THIỆU CHUNG

Trong chương này chúng ta sẽ xem xét một số kỹ thuật điều chế và giải điều chế số bằng gốc để truyền thông tin số qua một kênh có tạp âm cộng trăng Gauss (AWGN). Chúng ta bắt đầu với điều chế xung nhị phân, sau đó chúng tôi sẽ giới thiệu một số phương pháp điều chế không nhị phân. Chúng ta sẽ mô tả các máy thu tối ưu đối với các tín hiệu khác nhau này và xem xét việc đánh giá chất lượng hoạt động của chúng theo xác suất lỗi trung bình.

### 5.2 TRUYỀN DẪN TÍN HIỆU NHỊ PHÂN

Trong một hệ thống truyền tin nhị phân, dữ liệu nhị phân gồm một dãy các số 0 và 1 được truyền đi bằng hai dạng sóng tín hiệu  $s_0(t)$  và  $s_1(t)$ . Giả sử rằng tốc độ dữ liệu được ấn định là  $R$  bit/s. Khi đó mỗi một bít được ánh xạ thành một dạng sóng tín hiệu tương ứng theo qui luật

$$0 \rightarrow s_0(t), \quad 0 \leq t \leq T_b$$

$$1 \rightarrow s_1(t), \quad 0 \leq t \leq T_b$$

trong đó  $T_b=1/R$  được định nghĩa là khoảng thời gian của một bít. Chúng ta giả sử rằng các bít dữ liệu 0 và 1 là đồng khả năng, nghĩa là mỗi một trong chúng xuất hiện với xác suất  $1/2$ , và độc lập thống kê với nhau.

Kênh mà tín hiệu được truyền dẫn qua được giả sử là gây nhiễu tín hiệu bằng việc cộng thêm tạp âm, được ký hiệu là  $n(t)$ , là một hàm mẫu của một quá

trình ngẫu nhiên Gauss trắng có phổ công suất là  $N_0/2$  W/Hz. Một kênh truyền như thế được gọi là một kênh tạp âm cộng trắng Gauss (AWGN: *Additive White Gaussian Noise*). Hệ quả là dạng sóng tín hiệu nhận được biểu diễn được theo

$$r(t) = s_i(t) + n(t), \quad i=0, 1, \quad 0 \leq t \leq T_b \quad (5.2.1)$$

Nhiệm vụ của máy thu là xác định xem một con 0 hay một con 1 đã được truyền đi sau khi quan sát tín hiệu thu được  $r(t)$  trong khoảng  $0 \leq t \leq T_b$ . Máy thu được thiết kế để cực tiểu hóa xác suất thu lỗi. Máy thu như thế được gọi là máy thu tối ưu.

### 5.2.1 MÁY THU TỐI UỐNG VỚI KÊNH AWGN

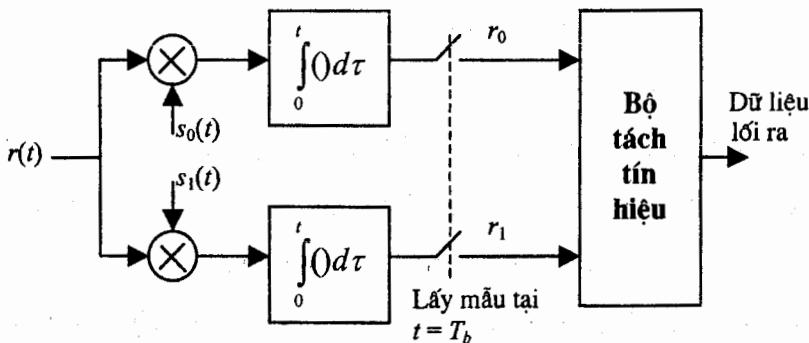
Trong hầu như tất cả các tài liệu về thông tin số cơ bản, người ta đều trình bày rằng máy thu tối ưu đối với kênh AWGN thì bao gồm hai khối cấu thành. Một khối thì hoặc là một bộ tương quan hoặc là một mạch lọc phổi hợp. Một khối là một bộ tách tín hiệu.

#### *Bộ tương quan tín hiệu*

Bộ tương quan tín hiệu tính tương quan chéo tín hiệu nhận được  $r(t)$  với hai tín hiệu đã được truyền đi có thể có  $s_0(t)$  và  $s_1(t)$ , như được minh họa trên hình 5.1. Nghĩa là bộ tương quan tín hiệu tính hai tín hiệu lối ra

$$r_0(t) = \int_0^t r(\tau)s_0(\tau)d\tau \quad \text{và} \quad r_1(t) = \int_0^t r(\tau)s_1(\tau)d\tau \quad (5.2.2)$$

trong khoảng  $0 \leq t \leq T_b$ , lấy mẫu hai tín hiệu lối ra này tại  $t = T_b$  rồi đưa các tín hiệu lối ra đã được lấy mẫu đó tới bộ tách tín hiệu.



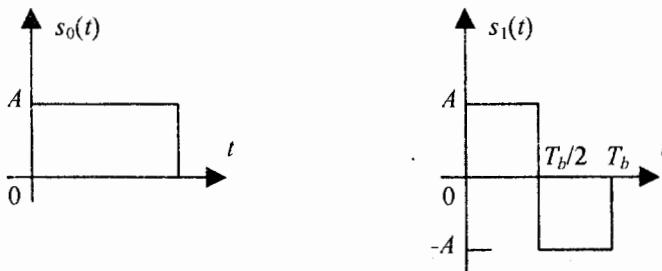
Hình 5.1 Tính tương quan chéo của tín hiệu thu được  $r(t)$  với hai tín hiệu đã được truyền.

## BÀI TẬP MINH HỌA

**Bài tập minh họa 5.1 [Bộ tương quan tín hiệu]** Giả sử các dạng sóng tín hiệu  $s_0(t)$  và  $s_1(t)$  là các dạng sóng được cho trên hình 5.2 và cho rằng  $s_0(t)$  là tín hiệu đã được truyền đi. Khi đó tín hiệu nhận được là

$$r(t) = s_0(t) + n(t), \quad 0 \leq t \leq T_b \quad (5.2.3)$$

Hãy xác định các tín hiệu lỗi ra của bộ tương quan tại các thời điểm lấy mẫu.



Hình 5.2 Các dạng sóng tín hiệu  $s_0(t)$  và  $s_1(t)$  đối với một hệ thống thông tin nhị phân.

## Lời giải

Khi tín hiệu  $r(t)$  được xử lý bởi hai bộ tương quan tín hiệu như trên hình 5.1, các tín hiệu lõi ra  $r_0$  và  $r_1$  tại thời điểm lấy mẫu  $t=T_b$  là

$$r_0 = \int_0^b r(t) s_0(t) dt = \int_0^b s_0^2(t) dt + \int_0^b n(t) s_0(t) dt = E + n_0 \quad (5.2.4)$$

$$\text{và} \quad r_i = \int_0^{T_b} r(t) s_i(t) dt = \int_0^{T_b} s_0(t) s_i(t) dt + \int_0^{T_b} n(t) s_i(t) dt = n_i \quad (5.2.5)$$

ở đây  $n_0$  và  $n_1$  là các thành phần tạp âm tại lối ra của các bộ tương quan tín hiệu; nghĩa là

$$n_0 = \int_0^{T_b} n(t) s_0(t) dt; \quad n_1 = \int_0^{T_b} n(t) s_1(t) dt \quad (5.2.6)$$

còn  $E=A^2T_b$  là năng lượng của các tín hiệu  $s_0(t)$  và  $s_1(t)$ . Chúng ta cũng chú ý rằng hai dạng sóng tín hiệu này thì trực giao với nhau, nghĩa là

$$\int_0^b s_0(t) s_1(t) dt = 0 \quad (5.2.7)$$

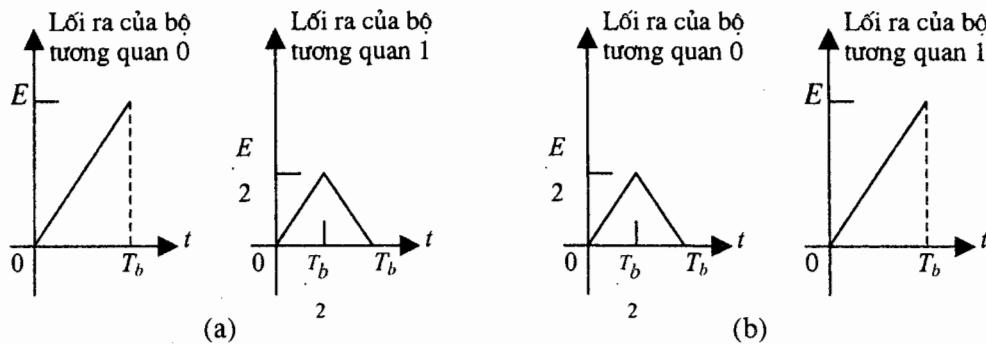
Mặt khác, khi  $s_1(t)$  được phát đi thì tín hiệu thu được là

$$r(t) = s_1(t) + n(t), \quad 0 \leq t \leq T_b$$

Trong trường hợp này, dễ dàng chỉ ra rằng các tín hiệu lối ra của các bộ tương quan tín hiệu sẽ là

$$r_0 = n_0, \quad r_1 = E + n_1 \quad (5.2.8)$$

Hình vẽ 5.3 minh họa các tín hiệu lối ra không có tạp âm của các bộ tương quan tín hiệu trong khoảng  $0 \leq t \leq T_b$  đối với từng trường hợp một trong hai trường hợp kể trên, tức là khi  $s_0(t)$  được truyền đi và khi  $s_1(t)$  được truyền đi.



Hình 5.3 Các lối ra không có tạp âm của các bộ tương quan.

(a)  $s_0(t)$  đã được phát đi; (b)  $s_1(t)$  đã được phát đi.

Do  $n(t)$  là một hàm mẫu của một quá trình Gauss trắng với phổ công suất bằng  $N_0/2$  nên các thành phần  $n_0$  và  $n_1$  có phân bố Gauss với kỳ vọng bằng không, nghĩa là

$$\begin{aligned} E(n_0) &= \int_0^{T_b} s_0(t) E[n(t)] dt = 0 \\ E(n_1) &= \int_0^{T_b} s_1(t) E[n(t)] dt = 0 \end{aligned} \quad (5.2.9)$$

và các phương sai  $\sigma_i^2$ , với  $i=0, 1$ , trong đó

$$\begin{aligned} \sigma_i^2 &= E(n_i^2) = \int_0^{T_b} \int_0^{T_b} s_i(t) s_i(\tau) E[n(t)n(\tau)] dt d\tau \\ &= \frac{N_0}{2} \int_0^{T_b} s_i(t) s_i(t) \delta(t-t) dt = \frac{N_0}{2} \int_0^{T_b} s_i^2(t) dt \end{aligned} \quad (5.2.10)$$

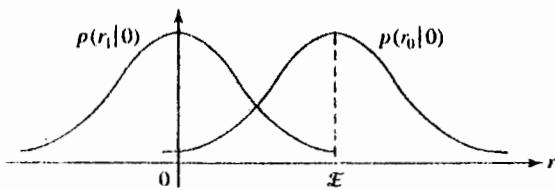
$$= \frac{EN_0}{2} = \sigma^2, \quad i = 0, 1 \quad (5.2.11)$$

Do đó, khi  $s_0(t)$  được truyền đi, hàm mật độ xác suất của  $r_0$  và  $r_1$  là

$$p(r_0 | s_0(t) \text{ đã được truyền đi}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(r_0 - E)^2 / 2\sigma^2}$$

$$p(r_1 | s_0(t) \text{ đã được truyền đi}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-r_1^2 / 2\sigma^2} \quad (5.2.12)$$

Hai hàm mật độ xác suất nào, được ký hiệu là  $p(r_0 | 0)$  và  $p(r_1 | 0)$ , được minh họa trên hình 5.4. Tương tự như vậy, khi  $s_1(t)$  được truyền đi,  $r_0$  là biến ngẫu nhiên Gauss với kỳ vọng bằng không và phương sai bằng  $\sigma^2$  còn  $r_1$  là một biến ngẫu nhiên Gauss với kỳ vọng bằng  $E$  và phương sai bằng  $\sigma^2$ .



Hình 5.4 Các hàm mật độ xác suất  $p(r_0 | 0)$  và  $p(r_1 | 0)$  khi  $s_0(t)$  đã được truyền đi.

### Bộ lọc phối hợp

Bộ lọc phối hợp cung cấp một cách thức khác với bộ tương quan tín hiệu nhằm giải điều chế tín hiệu nhận được  $r(t)$ . Một bộ lọc được phối hợp với dạng sóng tín hiệu  $s(t)$ , trong đó  $0 \leq t \leq T_b$ , có một phản ứng xung

$$h(t) = s(T_b - \tau), \quad 0 \leq t \leq T_b \quad (5.2.13)$$

Hệ quả là dạng sóng tín hiệu - gọi là  $y(t)$  - tại lối ra của mạch lọc phối hợp khi tín hiệu lối vào là  $s(t)$  sẽ được cho bởi tích chập

$$y(t) = \int s(\tau) h(t - \tau) d\tau \quad (5.2.14)$$

Nếu ta thay  $h(t - \tau)$  từ (5.2.13) vào (5.2.14) ta sẽ nhận được

$$y(t) = \int s(\tau) s(T_b - t + \tau) d\tau \quad (5.2.15)$$

và nếu chúng ta lấy mẫu  $y(t)$  tại  $t = T_b$  thì chúng ta sẽ có được

$$y(T_b) = \int_0^{T_b} s^2(t) dt = E \quad (5.2.16)$$

trong đó  $E$  là năng lượng của tín hiệu  $s(t)$ . Vì vậy, tín hiệu lối ra mạch lọc phối hợp tại thời điểm lấy mẫu  $t=T_b$  thì đồng nhất với tín hiệu lối ra của bộ tương quan tín hiệu.

### BÀI TẬP MINH HỌA

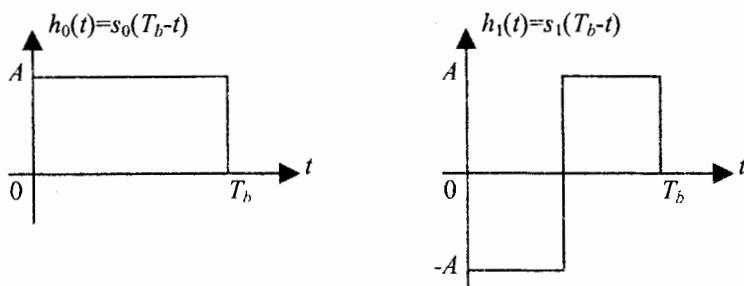
**Bài tập minh họa 5.2 [Bộ lọc phối hợp]** Hãy xem xét việc sử dụng các bộ lọc phối hợp để giải điều chế hai dạng sóng tín hiệu được cho trên hình 5.2 và xác định các tín hiệu lối ra.

#### Lời giải

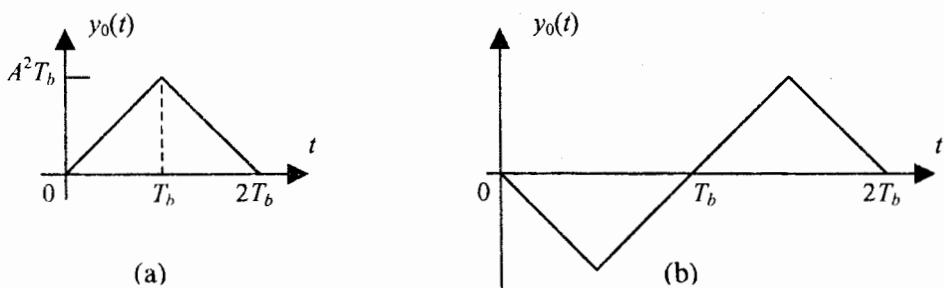
Các phản ứng xung của hai bộ lọc phối hợp là

$$h_0(t)=s_0(T_b-t), \quad h_1(t)=s_1(T_b-t) \quad (5.2.17)$$

như được minh họa trên hình 5.5. Hãy để ý rằng mỗi một phản ứng xung trên đây đều nhận được bằng cách thoát tiên gập tín hiệu  $s(t)$  để có được  $s(-t)$  rồi sau đó giữ chậm tín hiệu đã gập  $s(-t)$  một lượng  $T_b$  để có được  $s(T_b-t)$ .



Hình 5.5 Các phản ứng xung của các bộ lọc phối hợp đối với các tín hiệu  $s_0(t)$  và  $s_1(t)$ .



Hình 5.6 Các tín hiệu lối ra của các mạch lọc phối hợp khi  $s_0(t)$  được truyền đi.

Bây giờ giả sử rằng dạng sóng tín hiệu  $s_0(t)$  được truyền đi. Khi đó tín hiệu nhận được  $r(t) = s_0(t) + n(t)$  được cho qua hai mạch lọc phối hợp đó. Đáp ứng lối ra

của mạch lọc có phản ứng xung  $h_0(t)$  đối với thành phần tín hiệu  $s_0(t)$  được minh họa trên hình 5.6(a). Cũng như vậy, đáp ứng lối ra của mạch lọc phối hợp có phản ứng xung  $h_1(t)$  đối với thành phần tín hiệu  $s_1(t)$  được minh họa trên hình vẽ 5.6(b). Do đó, tại thời điểm lấy mẫu  $t = T_b$ , tín hiệu lối ra của hai mạch lọc phối hợp với các phản ứng xung  $h_0(t)$  và  $h_1(t)$  lần lượt là

$$r_0 = E + n_0, \quad r_1 = n_1 \quad (5.2.18)$$

Hãy để ý rằng các tín hiệu đầu ra này hoàn toàn giống như các tín hiệu lối ra nhận được từ việc lấy mẫu các tín hiệu lối ra bộ tương quan tín hiệu tại  $t=T_b$ .

### Bộ tách tín hiệu

Bộ tách tín hiệu quan sát các tín hiệu lối ra của bộ tính tương quan hay mạch lọc phối hợp  $r_0$  và  $r_1$  rồi thực hiện quyết định xem tín hiệu đã truyền đi là  $s_0(t)$  hay  $s_1(t)$  mà chúng tương ứng với một con 0 hay một con 1. Bộ tách tín hiệu tối ưu được định nghĩa là bộ tách tín hiệu làm cực tiểu xác suất lỗi.

## BÀI TẬP MINH HỌA

**Bài tập minh họa 5.3 [Tách tín hiệu nhị phân]** Ta hãy xét bộ tách tín hiệu đối với các tín hiệu được cho trên hình 5.2, mà chúng đồng xác suất và cùng năng lượng. Bộ tách tín hiệu tối ưu đối với các tín hiệu này thực hiện so sánh  $r_0$  và  $r_1$  rồi quyết định một con 0 đã được truyền đi nếu  $r_0 > r_1$  và một con 1 đã được truyền đi nếu  $r_1 > r_0$ . Hãy xác định xác suất lỗi.

### Lời giải

Khi  $s_0(t)$  là dạng sóng đã được truyền, xác suất lỗi sẽ là

$$P_e = P(r_1 > r_0) = P(n_1 > E + n_0) = P(n_1 - n_0 > E) \quad (5.2.19)$$

Do  $n_1$  và  $n_0$  là các biến ngẫu nhiên Gauss có kỳ vọng bằng không nên hiệu của chúng  $x = n_1 - n_0$  cũng là một biến ngẫu nhiên Gauss có kỳ vọng bằng không. Phương sai của biến ngẫu nhiên  $x$  là

$$E(x^2) = E[(n_1 - n_0)^2] = E(n_1^2) + E(n_0^2) - 2E(n_1 n_0) \quad (5.2.20)$$

Song  $E(n_1 n_0) = 0$  do các dạng sóng trực giao nhau, tức là

$$\begin{aligned}
 E(n_1 n_0) &= E\left[\int_0^{T_b} \int_0^{T_b} s_0(t) s_1(\tau) n(t) n(\tau) dt d\tau\right] \\
 &= \frac{N_0}{2} \int_0^{T_b} \int_0^{T_b} s_0(t) s_1(\tau) \delta(t - \tau) dt d\tau \\
 &= \frac{N_0}{2} \int_0^{T_b} s_0(t) s_1(t) dt = 0
 \end{aligned} \tag{5.2.21}$$

Do đó,

$$E(x^2) = 2\left(\frac{EN_0}{2}\right) = EN_0 = \sigma_x^2 \tag{5.2.22}$$

Do vậy, xác suất lỗi là

$$P_e = \frac{1}{\sqrt{2\pi}\sigma_x} \int_E^\infty e^{-x^2/2\sigma_x^2} dx = \frac{1}{\sqrt{2\pi}} \int_{\sqrt{E/N_0}}^\infty e^{-x^2/2} dx = Q\left(\sqrt{\frac{E}{N_0}}\right) \tag{5.2.23}$$

Tỷ số  $E/N_0$  được gọi là tỷ số tín/tạp (SNR).

Việc dẫn đến chất lượng hoạt động của bộ tách tín hiệu đã cho trong thí dụ này là dựa trên việc truyền dẫn dạng sóng tín hiệu  $s_0(t)$ . Người đọc có thể chứng minh lấy được rằng xác suất lỗi nhận được khi  $s_1(t)$  được truyền đi thì giống y như xác suất lỗi đã nhận được khi  $s_0(t)$  được truyền đi. Do các bit 0 và 1 trong dãy dữ liệu là đồng xác suất nên xác suất lỗi trung bình đúng bằng xác suất lỗi đã được cho bởi (5.2.23). Biểu thức này đối với xác suất lỗi được đánh giá bằng MATLAB script được cho dưới đây và được vẽ trên hình 5.7 như một hàm của SNR, trong đó SNR được hiển thị trên thang logarít ( $10\log_{10}(E/N_0)$ ). Như đoán trước, xác suất lỗi giảm theo hàm mũ khi SNR tăng.

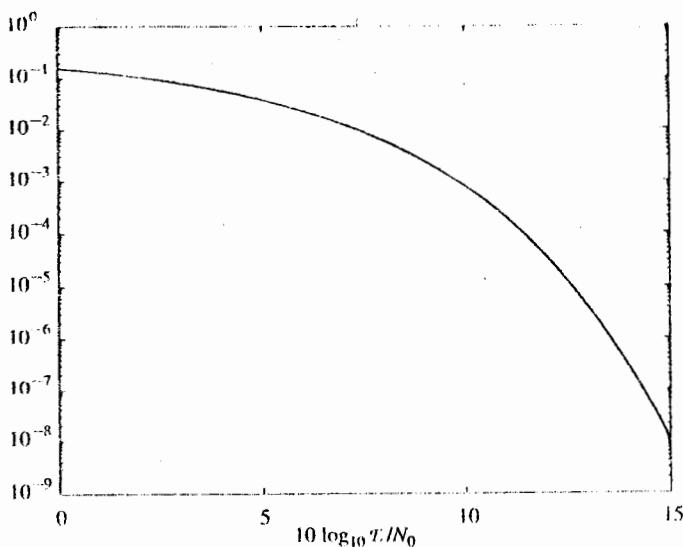
### m-file

```
% The MATLAB script that generates the probability of error versus the signal to noise ratio
initial_snr=0;
final_snr=15;
snr_step=0.25;
snr_in_dB=initial_snr:snr_step:final_snr;
for i=1:length(snr_in_dB),
    snr=10^(snr_in_dB(i)/10);
```

```

Pe(i)=Qfunct(sqrt(snr));
echo off;
end;
echo on;
semilogy(snr_in_dB,Pe);

```



Hình 5.7 Xác suất lỗi đối với các tín hiệu trực giao.

### 5.2.2 MÔ PHỎNG MONTE-CARLO MỘT HỆ THỐNG TRUYỀN TIN NHỊ PHÂN

Các mô phỏng Monte-Carlo bằng máy tính thường được tiến hành trong thực tế nhằm ước lượng xác suất lỗi bít của một hệ thống truyền tin số, đặc biệt là các trường hợp trong đó khó thực hiện việc phân tích chất lượng bộ tách tín hiệu. Chúng tôi sẽ trình diễn phương pháp ước lượng xác suất lỗi đối với hệ thống truyền tin nhị phân đã được mô tả trong phần trước.

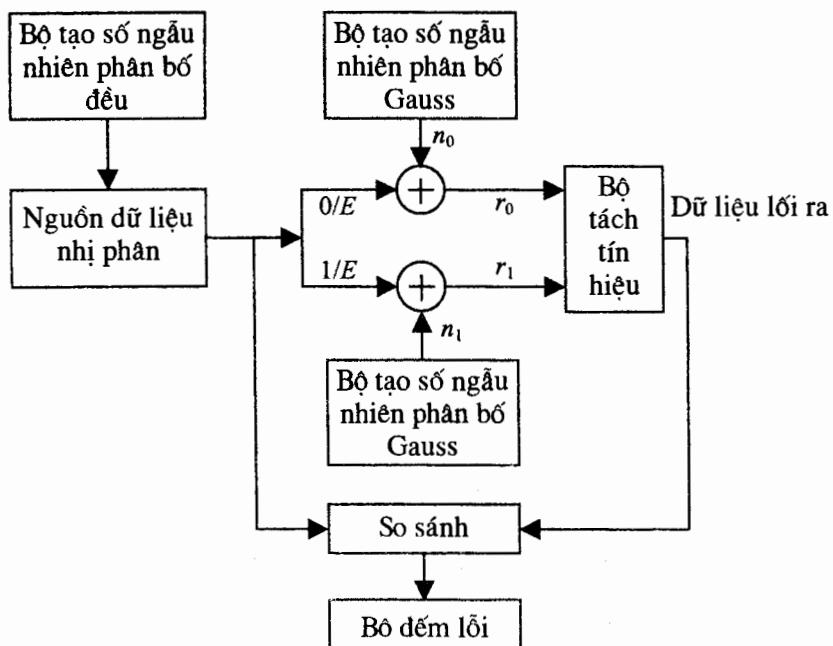
#### BÀI TẬP MINH HỌA

**Bài tập minh họa 5.4 [Mô phỏng Monte-Carlo]** Hãy sử dụng mô phỏng Monte-Carlo để ước lượng và vẽ đồ thị  $P_e$  theo SNR đối với một hệ thống truyền tin nhị phân có áp dụng các bộ tương quan tín hiệu hay các mạch lọc phổi hợp. Mô hình của hệ thống được minh họa trên hình 5.8.

#### Lời giải

Chúng ta mô phỏng việc tạo ra các biến ngẫu nhiên  $r_0$  và  $r_1$  tạo thành tín hiệu

lỗi vào của bộ tách tín hiệu. Chúng ta bắt đầu bằng việc tạo một dãy nhị phân gồm các bít 0 và 1 xuất hiện với xác suất như nhau và độc lập thống kê với nhau. Để thực hiện nhiệm vụ này, chúng ta sử dụng một bộ tạo số ngẫu nhiên tạo ra một số ngẫu nhiên phân bố đều trong khoảng  $(0, 1)$ . Nếu con số được tạo ra nằm trong khoảng  $(0, 0.5)$  thì lỗi ra của nguồn nhị phân là một bít 0. Còn không thì lỗi ra của nguồn là bít 1. Nếu một bít 0 được tạo ra, thì  $r_0 = E + n_0$  và  $r_1 = n_1$ . Còn nếu một bít 1 được tạo ra thì  $r_0 = n_0$  và  $r_1 = E + n_1$ .

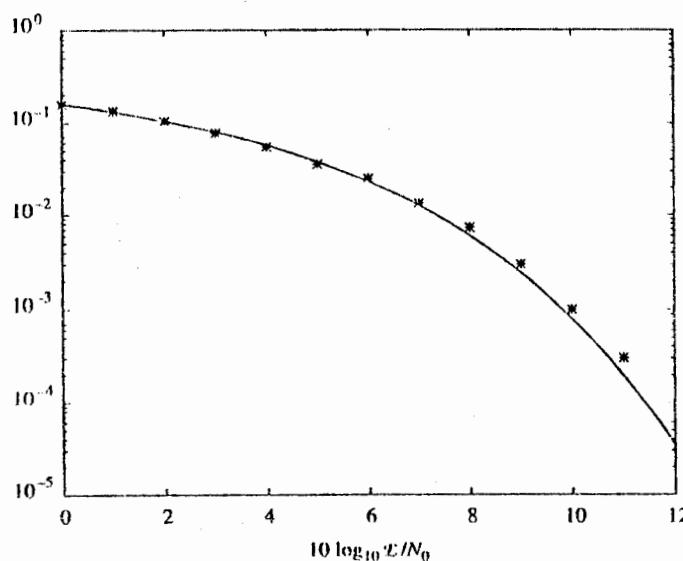


Hình 5.8 Mô hình mô phỏng cho bài tập minh họa 5.4.

Các thành phần tạp âm cộng  $n_0$  và  $n_1$  được tạo ra bằng hai bộ tạo tạp âm Gauss. Các kỳ vọng của chúng bằng không còn các phương sai của chúng là  $\sigma^2 = EN_0/2$ . Để thuận tiện, chúng ta có thể chuẩn hoá năng lượng tín hiệu  $E$  thành đơn vị ( $E=1$ ) và cho  $\sigma^2$  biến thiên. Hãy để ý rằng SNR, được định nghĩa là  $E/N_0$ , khi này bằng  $1/\sigma^2$ . Tín hiệu lỗi ra bộ tách tín hiệu được so sánh với chuỗi nhị phân đã được truyền đi và một bộ đếm lỗi được sử dụng để đếm số các lỗi bít.

Hình 5.9 minh họa các kết quả của việc mô phỏng này đối với việc truyền đưa của  $N=10\,000$  bít tại các mức khác nhau của SNR. Hãy chú ý đến sự khớp

nhau giữa các kết quả mô phỏng và giá trị lý thuyết của  $P_e$  cho bởi (5.2.23). Chúng ta cũng cần phải lưu ý rằng một mô phỏng với  $N=10\,000$  bít dữ liệu cho phép chúng ta ước lượng một cách tin cậy xác suất lỗi xuống tới  $P_e$  vào cỡ  $10^{-3}$ . Nói một cách khác đi thì với  $N=10\,000$  bít dữ liệu, chúng ta có ít nhất 10 lỗi đối với một ước lượng tin cậy của  $P_e$ . Các MATLAB *script* cho bài tập này được cho dưới đây.



Hình 5.9 Xác suất lỗi từ mô phỏng Monte-Carlo so sánh với xác suất lỗi lý thuyết đối với việc truyền tin bằng các tín hiệu trực giao.

### m-file

```
% MATLAB script for Illustrative Problem 4, Chapter 5.
echo on
SNRindB1=0:1:12;
SNRindB2=0:0.1:12;
for i=1:length(SNRindB1),
    % simulated error rate
    smld_err_prb(i)=smldPe54(SNRindB1(i));
    echo off ;
end;
echo on ;
for i=1:length(SNRindB2),
    SNR=exp(SNRindB2(i)*log(10)/10);
```

```
% theoretical error rate
theo_err_prb(i)=Qfunct(sqrt(SNR));
echo off;
end;
echo on;
% Plotting commands follow
semilogy(SNRindB1,smld_err_prb,'*');
hold
semilogy(SNRindB2,theo_err_prb);
```

**m-file**

```
function [p]=smldPe54(snr_in_dB)
% p=smldPe54(snr_in_dB)
% SMLDPE54 finds the probability of error for the given snr_in_dB, signal to
% noise ratio in dB.
E=1;
SNR=exp(snr_in_dB*log(10)/10); % signal to noise ratio
sgma=E/sqrt(2*SNR); % sigma, standard deviation of noise
N=10000;
% generation of the binary data source
for i=1:N,
    temp=rand; % a uniform random variable over (0,1)
    if (temp<0.5),
        dsource(i)=0; % with probability 1/2, source output is 0
    else
        dsource(i)=1; % with probability 1/2, source output is 1
    end
end;
% detection, and probability of error calculation
numoferr=0;
for i=1:N,
    % matched filter outputs
    if (dsource(i)==0),
        r0=E+gngauss(sgma);
        r1=gngauss(sgma); % if the source output is "0"
    else
        r0=gngauss(sgma);
        r1=E+gngauss(sgma); % if the source output is "1"
    end;
    % detector follows
    if (r0>r1),
        decis=0; % decision is "0"
    else
```

```

else
    decis=1;                                % decision is "1"
end;
if (decis~=dsource(i)),                  % if it is an error, increase the error counter
    numoferr=numoferr+1;
end;
end;
p=numoferr/N;                            % probability of error estimate

```

### Câu hỏi

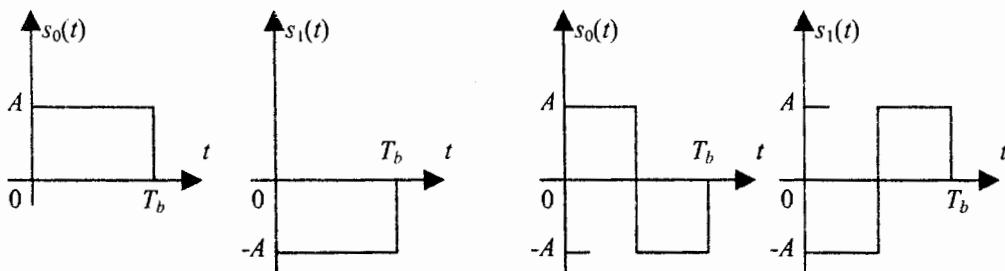
Trên hình 5.9, các kết quả mô phỏng và lý thuyết hoàn toàn khớp nhau tại các tỷ số tín/tạp khá nhỏ, trong khi tại các SNR cao thì chúng kém khớp nhau hơn. Bạn có thể giải thích tại sao lại như thế được không? Chúng ta sẽ cần phải thay đổi thế nào quá trình mô phỏng để có được kết quả khớp với giá trị lý thuyết tại các tỷ số tín/tạp cao?

### 5.2.3 CÁC PHƯƠNG PHÁP TRUYỀN DẪN TÍN HIỆU NHỊ PHÂN KHÁC

Phương pháp truyền dẫn tín hiệu nhị phân đã được mô tả trên đây dựa trên việc sử dụng các tín hiệu trực giao. Tiếp theo đây chúng tôi sẽ mô tả hai phương pháp khác dùng để truyền thông tin nhị phân qua một kênh truyền tin. Một phương pháp thì áp dụng các tín hiệu đối cực còn phương pháp kia áp dụng tín hiệu kiểu đóng-ngắt (*on-off-type signal*).

#### *Các tín hiệu đối cực dùng cho truyền dẫn tín hiệu nhị phân*

Hai dạng sóng tín hiệu được nói là đối cực nếu dạng sóng tín hiệu này bằng âm của dạng sóng tín hiệu kia. Thí dụ, một cặp tín hiệu đối cực được minh họa trên hình 5.10(a) và một cặp thứ hai được minh họa trên hình 5.10(b).



Hình 5.10 Các thí dụ về các cặp tín hiệu đối cực.

Giả sử chúng ta sử dụng các dạng sóng tín hiệu đối cực  $s_0(t)=s(t)$  và  $s_1(t)=-s(t)$  để truyền dẫn thông tin nhị phân, trong đó  $s(t)$  là một dạng sóng tùy ý nào đó có năng lượng là  $E$ . Dạng sóng tín hiệu thu được từ một kênh AWGN có thể biểu diễn được theo

$$r(t) = \pm s(t) + n(t), \quad 0 \leq t \leq T_b \quad (5.2.24)$$

Máy thu tối ưu để khôi phục thông tin nhị phân sử dụng chỉ một bộ tính tương quan hoặc chỉ một bộ lọc phôi hợp, phôi hợp với  $s(t)$ , mắng nối tiếp với một bộ tách tín hiệu như được minh họa trên hình 5.11. Ta hãy giả sử rằng  $s(t)$  đã được truyền đi, vì vậy tín hiệu thu được là

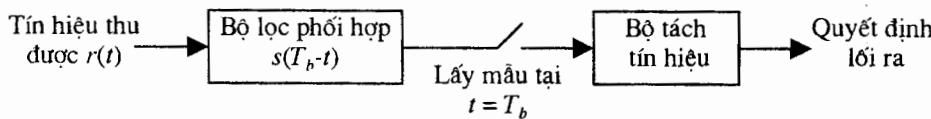
$$r(t) = s(t) + n(t) \quad (5.2.25)$$

Lối ra của bộ tương quan hay bộ lọc phôi hợp ở thời điểm lấy mẫu  $t=T_b$  là

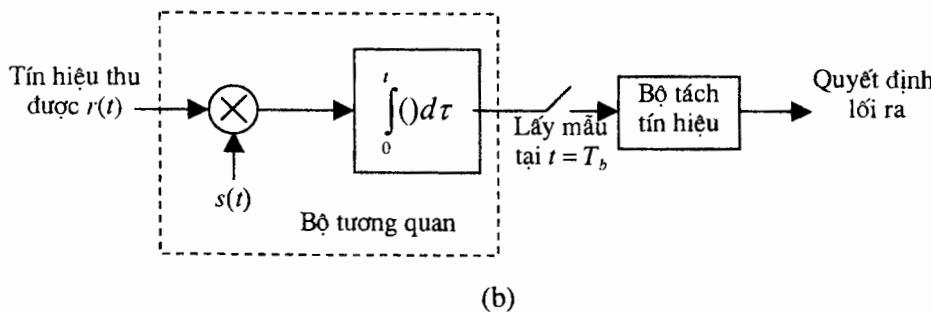
$$r = E + n \quad (5.2.26)$$

trong đó  $E$  là năng lượng tín hiệu còn  $n$  là thành phần tạp âm cộng biểu diễn được theo

$$n = \int_0^{T_b} n(t)s(t)dt \quad (5.2.27)$$



(a)



(b)

Hình 5.11 Máy thu tối ưu đối với các tín hiệu đối cực. (a) Bộ giải điều chế dùng mạch lọc phôi hợp; (b) Bộ giải điều chế dùng bộ tính tương quan.

Do quá trình tạp âm cộng có kỳ vọng bằng không nên  $E(n) = 0$ . Phương sai của thành phần tạp âm  $n$  là

$$\begin{aligned}\sigma^2 &= E(n^2) = \int_0^{T_b} \int_0^{T_b} E[n(t)n(\tau)] s(t)s(\tau) dt d\tau \\ &= \frac{N_0}{2} \int_0^{T_b} \delta(t-\tau) s(t)s(\tau) dt d\tau = \frac{N_0}{2} \int_0^{T_b} s^2(t) dt = \frac{N_0 E}{2}\end{aligned}\quad (5.2.28)$$

Do đó hàm mật độ xác suất của  $r$  khi  $s(t)$  đã được truyền đi là

$$p(r | s(t) \text{ đã được truyền đi}) \equiv p(r | 0) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(r-E)^2/2\sigma^2} \quad (5.2.29)$$

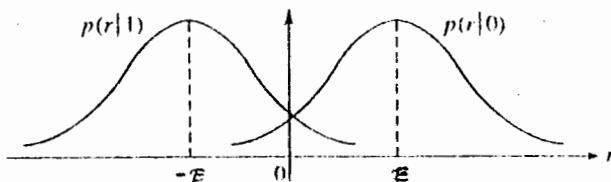
Tương tự như vậy, khi dạng sóng tín hiệu  $-s(t)$  được truyền đi, tín hiệu lõi vào bộ tách tín hiệu sẽ là

$$r = -E + n \quad (5.2.30)$$

và hàm mật độ xác suất của  $r$  là

$$p(r | s(t) \text{ đã được truyền đi}) \equiv p(r | 1) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(r+E)^2/2\sigma^2} \quad (5.2.31)$$

Hai hàm mật độ xác suất này được minh họa trên hình 5.12.



Hình 5.12 Hàm mật độ xác suất đối với tín hiệu lõi vào bộ tách tín hiệu.

Đối với các dạng sóng tín hiệu đồng xác suất, bộ tách tín hiệu tối ưu thực hiện so sánh  $r$  với ngưỡng bằng 0. Nếu  $r > 0$ , nó sẽ quyết định rằng  $s(t)$  đã được truyền đi. Nếu  $r < 0$ , bộ tách tín hiệu sẽ quyết định rằng  $-s(t)$  đã được truyền đi.

Tạp âm gây rối tín hiệu thì gây nên các lõi tại bộ tách tín hiệu. Xác suất của một lõi tách tín hiệu tính được dễ dàng. Giả sử rằng  $s(t)$  đã được truyền đi. Khi đó xác suất lõi bằng với xác suất để  $r < 0$ ; nghĩa là

$$\begin{aligned}
 P_e = P(r < 0) &= \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^0 e^{-(r-E)^2/2\sigma^2} dr \\
 &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-E/\sigma} e^{-r^2/2} dr = Q\left(\frac{E}{\sigma}\right) = Q\left(\sqrt{\frac{2E}{N_0}}\right)
 \end{aligned} \tag{5.2.32}$$

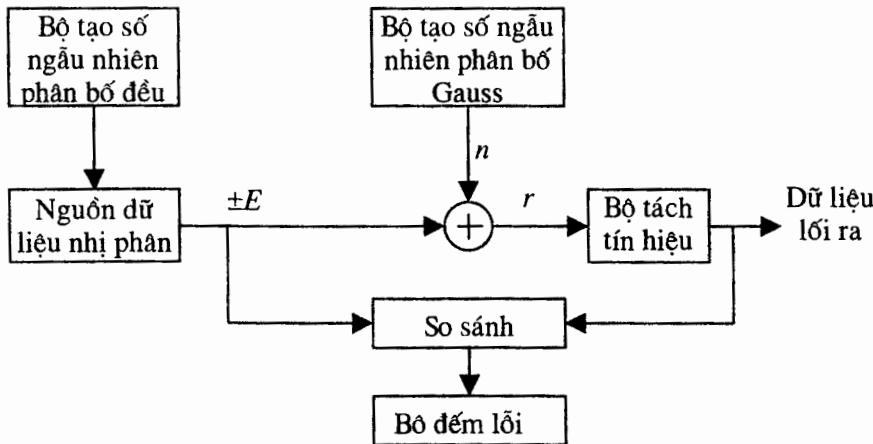
Một kết quả tương tự cũng sẽ nhận được khi  $-s(t)$  được truyền đi. Hệ quả là khi hai dạng sóng tín hiệu đồng xác suất thì xác suất lỗi trung bình được cho bởi (5.2.32).

Khi so sánh xác suất lỗi đối với các tín hiệu đối cực với xác suất lỗi đối với các tín hiệu trực giao được cho bởi (5.2.23) chúng ta thấy rằng đối với cùng một năng lượng tín hiệu được truyền đi  $E$  thì các tín hiệu đối cực cho một chất lượng hoạt động tốt hơn. Hoặc theo một cách khác, chúng ta có thể nói rằng các tín hiệu đối cực sẽ có cùng một chất lượng hoạt động (có cùng xác suất lỗi) như các tín hiệu trực giao bằng cách sử dụng một năng lượng tín hiệu truyền đi bằng một nửa năng lượng tín hiệu được truyền đi đối với các tín hiệu trực giao. Do vậy, các tín hiệu đối cực hiệu quả hơn các tín hiệu trực giao 3 dB.

### BÀI TẬP MINH HỌA

#### Bài tập minh họa 5.5 [Mô phỏng hệ thống tín hiệu đối cực nhị phân]

Hãy sử dụng mô phỏng Monte-Carlo để ước lượng và vẽ đồ thị chất lượng xác suất lỗi của một hệ thống truyền tin sử dụng tín hiệu đối cực nhị phân. Mô hình của hệ thống này được minh họa trên hình 5.13.



Hình 5.13 Mô hình của một hệ thống truyền tin nhị phân áp dụng các tín hiệu đối cực.

## Lời giải

Như đã được cho thấy, chúng ta sẽ mô phỏng việc tạo biến ngẫu nhiên  $r$  là tín hiệu đầu vào của bộ tách tín hiệu. Một bộ tạo số ngẫu nhiên phân bố đều được sử dụng để tạo ra chuỗi thông tin nhị phân từ nguồn dữ liệu nhị phân. Chuỗi các bit 0 và 1 này được ánh xạ thành một chuỗi gồm  $\pm E$ , trong đó  $E$  biểu thị năng lượng của tín hiệu.  $E$  có thể được chuẩn hóa thành đơn vị. Một bộ tạo tạp âm Gauss được sử dụng để tạo ra chuỗi các số ngẫu nhiên Gauss có kỳ vọng bằng 0 và phương sai bằng  $\sigma^2$ . Bộ tách tín hiệu so sánh biến ngẫu nhiên  $r$  với ngưỡng 0. Nếu  $r > 0$ , quyết định được đưa ra là bit đã được truyền đi là bit 0. Nếu  $r < 0$ , quyết định được đưa ra là bit đã được truyền đi là bit 1. Lỗi ra của bộ tách tín hiệu được so sánh với chuỗi bit thông tin đã được truyền đi và lỗi sẽ được đếm. Hình 5.14 minh họa chuỗi các kết quả của mô phỏng này đối với việc truyền dẫn 10 000 bit tại một số giá trị khác nhau của SNR. Giá trị lý thuyết của  $P_e$  được cho bởi (5.2.32) cũng được vẽ trên hình 5.14 để so sánh.

Các MATLAB script cho bài tập này được cho tiếp sau đây.

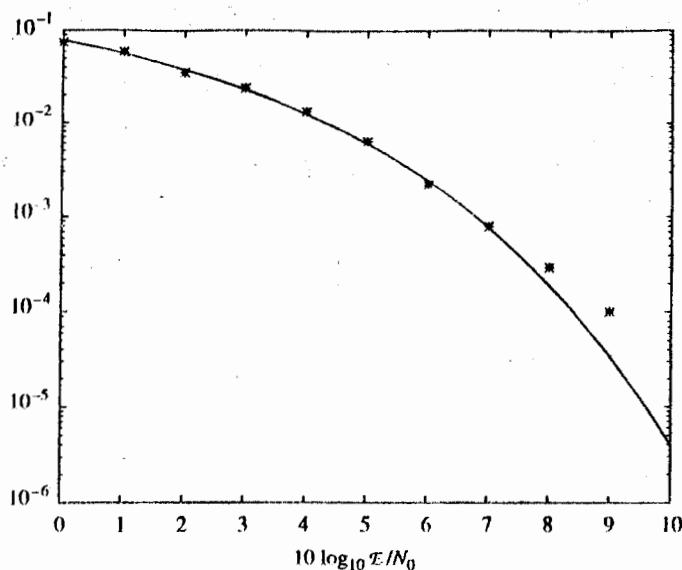
### m-file

```
% MATLAB script for Illustrated Problem 5, Chapter 5.
echo on
SNRindB1=0:1:10;
SNRindB2=0:0.1:10;
for i=1:length(SNRindB1),
    % simulated error rate
    smld_err_prb(i)=smldPe55(SNRindB1(i));
    echo off;
end;
echo on;
for i=1:length(SNRindB2),
    SNR=exp(SNRindB2(i)*log(10)/10);
    % theoretical error rate
    theo_err_prb(i)=Qfunct(sqrt(2*SNR));
    echo off;
end;
echo on;
% Plotting commands follow
```

```
semilogy(SNRindB1,smld_err_prb,'*');
hold
semilogy(SNRindB2,theo_err_prb);
```

### m-file

```
function [p]=smldPe55(snr_in_dB)
% [p]=smldPe55(snr_in_dB)
%      SMLDPE55 simulates the probability of error for the particular
%      value of snr_in_dB, signal to noise ratio in dB.
E=1;
SNR=exp(snr_in_dB*log(10)/10);           % signal to noise ratio
sgma=E/sqrt(2*SNR);                      % sigma, standard deviation of noise
N=10000;
% generation of the binary data source follows
for i=1:N,
    temp=rand;                            % a uniform random variable over (0,1)
    if (temp<0.5),
        dsource(i)=0;                   % with probability 1/2, source output is 0
    else
        dsource(i)=1;                   % with probability 1/2, source output is 1
    end
end;
% the detection, and probability of error calculation follows
numoferr=0;
for i=1:N,
    % The matched filter outputs
    if (dsource(i)==0),
        r=-E+gngauss(sgma);          % if the source output is "0"
    else
        r=E+gngauss(sgma);          % if the source output is "1"
    end;
    % detector follows
    if (r<0),
        decis=0;                     % decision is "0"
    else
        decis=1;                     % decision is "1"
    end;
    if (decis~=dsource(i)),           % if it is an error, increase the error counter
        numoferr=numoferr+1;
    end;
end;
p=numoferr/N;                           % probability of error estimate
```



Hình 5.14 Xác suất lỗi nhận được từ mô phỏng Monte-Carlo so với xác suất lỗi lý thuyết đối với các tín hiệu đối cực.

### Các tín hiệu đóng-ngắt dùng cho truyền dẫn tín hiệu nhị phân

Một chuỗi thông tin nhị phân cũng có thể truyền dẫn được nhờ sử dụng các tín hiệu đóng-ngắt. Để truyền một bít 0, không tín hiệu nào được truyền đi trong khoảng thời gian tồn tại tín hiệu  $T_b$ . Để truyền một bít 1 thì một dạng sóng tín hiệu  $s(t)$  được truyền đi. Hệ quả là, dạng sóng tín hiệu nhận được có thể biểu diễn được theo

$$r(t) = \begin{cases} n(t), & \text{nếu một 0 được truyền đi} \\ s(t) + n(t), & \text{nếu một 1 được truyền đi} \end{cases} \quad (5.2.33)$$

trong đó  $n(t)$  biểu thị tạp âm Gauss cộng trắng.

Giống như trong trường hợp các tín hiệu đối cực, máy thu tối ưu cũng gồm một bộ tương quan tín hiệu hay một mạch lọc phối hợp, phối hợp với  $s(t)$ , có đầu ra được lấy mẫu tại  $t=T_b$  rồi mắc nối tiếp tới một bộ tách tín hiệu mà nó so sánh tín hiệu đầu ra đã được lấy mẫu đó với giá trị ngưỡng ký hiệu là  $\alpha$ . Nếu  $r > \alpha$ , một bít 1 được tuyên bố là đã được truyền đi, trái lại thì một 0 được tuyên bố là đã được truyền đi.

Tín hiệu lỗi vào bộ tách tín hiệu có thể biểu diễn được theo

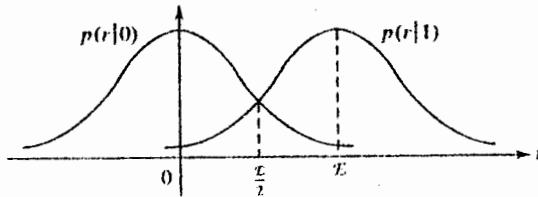
$$r = \begin{cases} n, & \text{nếu một 0 được truyền đi} \\ E + n, & \text{nếu một 1 được truyền đi} \end{cases} \quad (5.2.34)$$

trong đó  $n$  là một biến ngẫu nhiên Gauss kỳ vọng bằng 0 với phương sai  $\sigma^2=EN_0/2$ . Do đó các hàm mật độ xác suất điều kiện của biến ngẫu nhiên  $r$  là

$$p(r|0) = \frac{1}{\sqrt{2\pi}\sigma} e^{-r^2/2\sigma^2}, \quad \text{nếu một 0 được truyền đi}$$

$$p(r|1) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(r-E)^2/2\sigma^2}, \quad \text{nếu một 1 được truyền đi}$$

Các hàm mật độ xác suất này được minh họa trên hình 5.15.



Hình 5.15 Các hàm mật độ xác suất đối với tín hiệu nhận được tại lối ra của bộ tương quan đối với các tín hiệu đóng-ngắt.

Khi một 0 được truyền đi, xác suất lỗi là

$$P_{e0}(\alpha) = P(r > \alpha) = \frac{1}{\sqrt{2\pi}\sigma} \int_{\alpha}^{\infty} e^{-r^2/2\sigma^2} dr \quad (5.2.35)$$

trong đó  $\alpha$  là ngưỡng. Mặt khác, khi một 1 được truyền đi, xác suất lỗi là

$$P_{e1}(\alpha) = P(r < \alpha) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\alpha} e^{-(r-E)^2/2\sigma^2} dr \quad (5.2.36)$$

Khi giả sử rằng các bít thông tin nhị phân là đồng xác suất, đối với xác suất lỗi trung bình chúng ta có

$$P_e(\alpha) = P_{e0}(\alpha) + \frac{1}{2} P_{e1}(\alpha) \quad (5.2.37)$$

Giá trị của ngưỡng  $\alpha$  làm cực tiểu xác suất lỗi trung bình tìm được bằng cách đạo hàm  $P_e(\alpha)$  rồi giải tìm ngưỡng tối ưu. Dễ dàng chỉ ra được rằng ngưỡng tối ưu là

$$\alpha_{opt} = \frac{1}{2} E \quad (5.2.38)$$

Việc thay giá trị tối ưu này vào (5.2.35), (5.2.36) và (5.2.37) thu được xác suất lỗi

$$P_e(\alpha_{\text{opt}}) = Q\left(\sqrt{\frac{E}{2N_0}}\right) \quad (5.2.39)$$

Chúng ta quan sát thấy được rằng chất lượng tỷ lệ lỗi với các tín hiệu đóng-ngắt thì không tốt bằng các tín hiệu đối cực. Về tỷ số tín/tạp, nó hoá ra là tồi hơn các tín hiệu đối cực 6 dB và tồi hơn các tín hiệu trực giao 3 dB để có cùng một xác suất lỗi. Tuy nhiên, năng lượng trung bình được truyền đi đối với các tín hiệu đóng-ngắt thì thấp hơn 3 dB so với các tín hiệu đối cực hay trực giao. Hệ quả là sự khác biệt này cần phải được tính đến khi thực hiện so sánh chất lượng hoạt động với các kiểu tín hiệu khác.

## BÀI TẬP MINH HỌA

### Bài tập minh họa 5.6 [Mô phỏng truyền dẫn bằng tín hiệu đóng-ngắt]

Hãy sử dụng mô phỏng Monte-Carlo để ước lượng và vẽ đồ thị chất lượng về lỗi của một hệ thống truyền tin nhị phân áp dụng tín hiệu đóng ngắt.

#### Lời giải

Mô hình đối với hệ thống đưa vào mô phỏng thì cũng tương tự như mô hình được thể hiện trên hình 5.13, ngoại trừ một điều là một trong các dạng sóng tín hiệu thì bằng 0. Do đó, chúng ta tạo ra một dãy các biến ngẫu nhiên  $\{r_i\}$  như đã cho bằng (5.3.24). Bộ tách tín hiệu so sánh các biến ngẫu nhiên  $\{r_i\}$  với ngưỡng tối ưu  $E/2$  và ra các quyết định thích hợp. Hình 5.16 minh họa xác suất lỗi ước lượng được dựa trên 10 000 digit nhị phân. Tỷ lệ lỗi lý thuyết được cho bởi (5.2.39) cũng được minh họa trên hình này.

Các MATLAB script cho bài tập này được cho dưới đây.

#### m-file

```
% MATLAB script for Illustrative Problem 6, Chapter 5.
echo on
SNRindB1=0:1:15;
SNRindB2=0:0.1:15;
for i=1:length(SNRindB1),
    smld_err_prb(i)=smldPe56(SNRindB1(i)); % simulated error rate
    echo off;
```

```

end;
echo on;
for i=1:length(SNRindB2),
    SNR=exp(SNRindB2(i)*log(10)/10);           % signal to noise ratio
    theo_err_prb(i)=Qfunct(sqrt(SNR/2));        % theoretical error rate
    echo off;
end;
echo on;
% Plotting commands follow
semilogy(SNRindB1,smld_err_prb,'*');
hold
semilogy(SNRindB2,theo_err_prb);

```

### m-file

```

function [p]=smldPe56(snr_in_dB)
% [p]=smldPe56(snr_in_dB)
% SMLDPE56 simulates the probability of error for a given
% snr_in_dB, signal to noise ratio in dB.

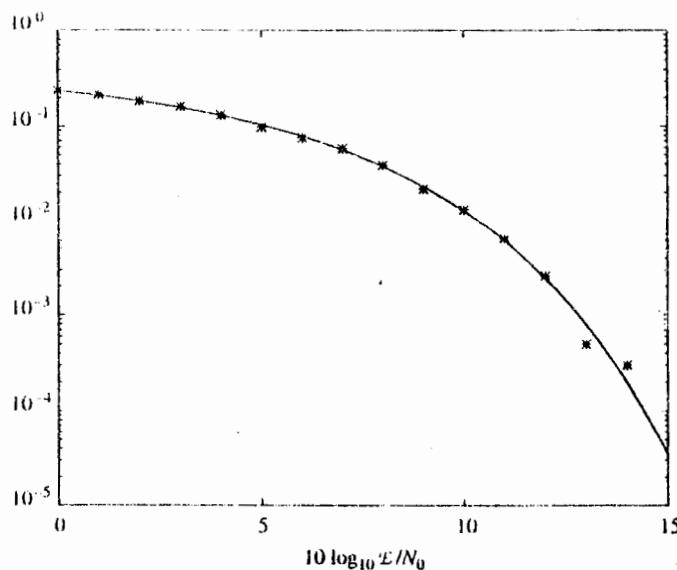
E=1;
alpha_opt=1/2;
SNR=exp(snr_in_dB*log(10)/10)           % signal to noise ratio
sgma=E/sqrt(2*SNR);                      % sigma, standard deviation of noise
N=10000;
% generation of the binary data source follows
for i=1:N,
    temp=rand;                           % a uniform random variable over (0,1)
    if (temp<0.5),
        dsource(i)=0;                   % with probability 1/2, source output is 0
    else
        dsource(i)=1;                   % with probability 1/2, source output is 1
    end
end;
% detection, and probability of error calculation
numoferr=0;
for i=1:N,
    % The matched filter outputs
    if (dsource(i)==0),
        r=gngauss(sgma);            % if the source output is "0"
    else
        r=E+gngauss(sgma);         % if the source output is "1"
    end
end;

```

```

end;
% detector follows
if (r<alpha_opt),
  decis=0;
  % decision is "0"
else
  decis=1;
  % decision is "1"
end;
if (decis~=dsource(i)),           % if it is an error, increase the error counter
  numoferr=numoferr+1;
end;
end;
p=numoferr/N;
% probability of error estimate

```

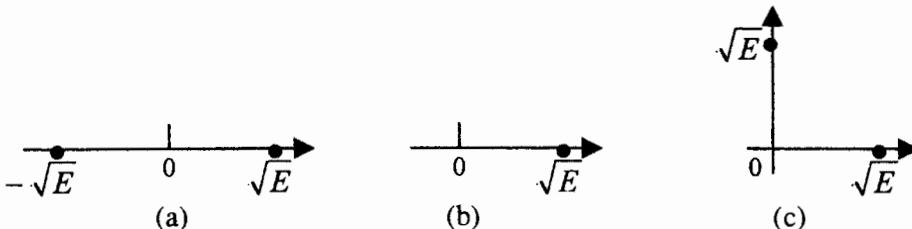


Hình 5.16 Xác suất lỗi thu được từ mô phỏng Monte-Carlo so với xác suất lỗi lý thuyết đối với các tín hiệu đóng-ngắt.

#### 5.2.4 BIỂU ĐỒ SAO CỦA TÍN HIỆU ĐỐI VỚI CÁC TÍN HIỆU NHỊ PHÂN

Ba kiểu tín hiệu của các nhị phân trên đây, cụ thể là đối cực, đóng-ngắt và trực giao, có thể đặc trưng được một cách hình học như các điểm trong “không gian tín hiệu”. Trong trường hợp các tín hiệu đối cực, trong đó các tín hiệu là  $s(t)$  và  $-s(t)$  mà từng tín hiệu này đều có năng lượng là  $E$ , thì các điểm này nằm tại

$\pm\sqrt{E}$  trên đường thẳng trực thực như được cho thấy trên hình vẽ 5.17(a). Biểu diễn hình học một chiều của các tín hiệu đối cực xuất phát từ thực tế là chỉ một dạng sóng tín hiệu hay hàm cơ sở, cụ thể là  $s(t)$ , là đủ để biểu diễn các tín hiệu đối cực trong không gian tín hiệu.



Hình 5.17 Các biểu đồ sao của tín hiệu. (a) Các tín hiệu đối cực;  
(b) Các tín hiệu đóng-ngắt; (c) Các tín hiệu trực giao.

Các tín hiệu đóng-ngắt cũng là tín hiệu một chiều (*one-dimensional*). Do đó, hai điểm tín hiệu cũng nằm trên trực thực, tại các điểm 0 và  $\sqrt{E}$  như được thể hiện trên hình vẽ 5.17(b).

Mặt khác, các tín hiệu trực giao nhị phân đòi hỏi một sự biểu diễn hình học hai chiều do có hai hàm độc lập tuyến tính  $s_0(t)$  và  $s_1(t)$  tham gia tạo nên hai dạng sóng tín hiệu. Hệ quả là các điểm tín hiệu tương ứng với các điểm này là  $(\sqrt{E}, 0)$  và  $(0, \sqrt{E})$  như thấy được trên hình vẽ 5.17(c).

Các biểu diễn hình học của các tín hiệu nhị phân được thể hiện trên hình vẽ 5.17 được gọi là biểu đồ sao của các tín hiệu (*signal constellation*).

### BÀI TẬP MINH HOA

#### Bài tập minh họa 5.7 [Tác động của tạp âm và biểu đồ sao của tín hiệu]

Tác động của tạp âm tới hoạt động của một hệ thống truyền tin nhị phân có thể quan sát thấy được qua tín hiệu thu được cộng tạp âm tại lối vào mạch tách tín hiệu. Ta hãy xét làm thí dụ các tín hiệu nhị phân trực giao mà đối với chúng tín hiệu lối vào bộ tách tín hiệu gồm cặp các biến ngẫu nhiên  $(r_0, r_1)$ , trong đó hoặc là

$$(r_0, r_1) = (\sqrt{E} + n_0, n_1)$$

hoặc là

$$(r_0, r_1) = (n_0, \sqrt{E} + n_1)$$

Các biến ngẫu nhiên tập nhiễu  $n_0$  và  $n_1$  là các biến ngẫu nhiên Gauss độc lập, có kỳ vọng bằng 0 và phương sai bằng  $\sigma^2$ . Giống như trong bài tập minh họa 5.4, hãy sử dụng mô phỏng Monte-Carlo để tạo ra 100 mẫu của  $(r_0, r_1)$  đối với từng giá trị của  $\sigma=0.1$ ,  $\sigma=0.3$  và  $\sigma=0.5$  và hãy vẽ cả 100 mẫu này trên các đồ thị hai chiều khác nhau cho từng  $\sigma$  khác nhau. Năng lượng  $E$  của tín hiệu có thể chuẩn hoá được thành 1.

### Lời giải

Các kết quả mô phỏng Monte-Carlo được cho thấy trên hình 5.18. Hãy chú ý rằng với mức công suất tạp âm thấp ( $\sigma$  nhỏ), tác động của tạp âm tới hoạt động của hệ thống truyền tin thì nhỏ. Khi mức công suất tạp âm tăng, các thành phần tạp âm tăng về kích thước và gây nên các lỗi.

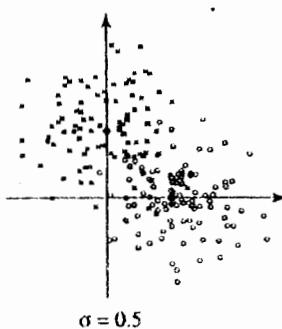
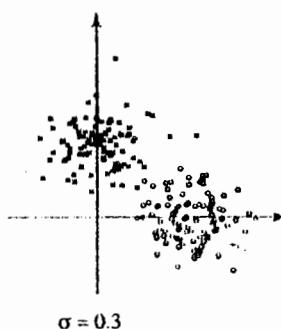
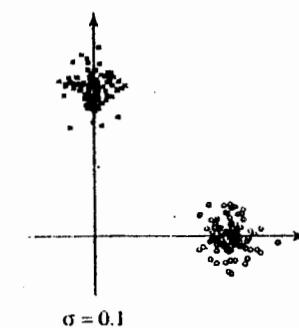
MATLAB script cho bài tập này đối với  $\sigma=0.5$  được cho dưới đây.

### m-file

```
% MATLAB script for Illustrated Problem 7, Chapter 5.  
echo on  
n0=.5*randn(100,1);  
n1=.5*randn(100,1);  
n2=.5*randn(100,1);  
n3=.5*randn(100,1);  
x1=1.+n0;  
y1=n1;  
x2=n2;  
y2=1.+n3;  
plot(x1,y1,'o',x2,y2,'*')  
axis('square')
```

## 5.3 TRUYỀN DẪN TÍN HIỆU NHIỀU BIÊN ĐỘ

Trong mục trước, chúng ta đã đề cập tới việc truyền dẫn thông tin số bằng cách sử dụng các dạng sóng tín hiệu nhị phân. Như thế thì mỗi một dạng sóng tải một bít thông tin. Trong mục này chúng ta sẽ sử dụng các dạng sóng mà chúng nhận nhiều mức biên độ. Bằng cách đó, chúng ta có thể truyền dẫn nhiều bít trên một dạng sóng tín hiệu.



Hình 5.18 Các điểm tín hiệu thu được tại lối vào mạch quyết định đối với các tín hiệu trực giao. (Mô phỏng Monte-Carlo).

### 5.3.1 CÁC DẠNG SÓNG VỚI 4 MỨC BIÊN ĐỘ

Ta hãy xét một tập các dạng sóng tín hiệu có dạng

$$s_m(t) = A_m g(t), \quad 0 \leq t \leq T \quad (5.3.1)$$

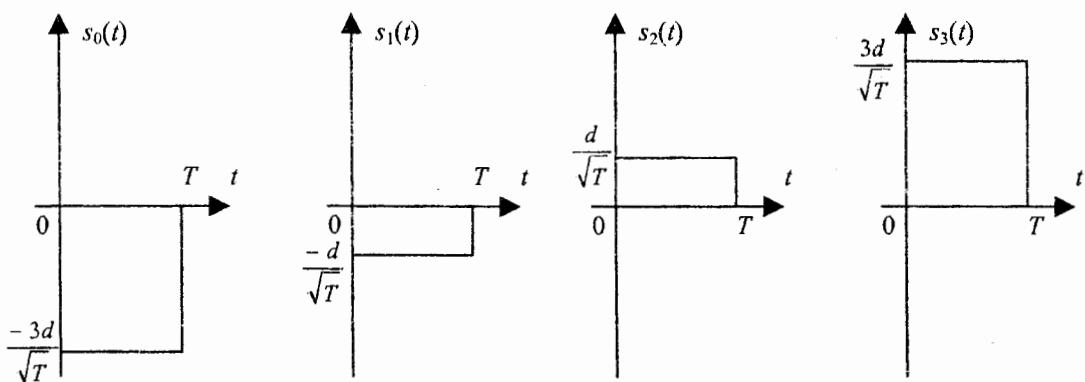
trong đó  $A_m$  là biên độ của dạng sóng thứ  $m$  và  $g(t)$  là một xung chũ nhặt được xác định theo

$$g(t) = \begin{cases} \sqrt{1/T}, & 0 \leq t \leq T \\ 0, & \text{với các giá trị khác của } t \end{cases} \quad (5.3.2)$$

trong đó năng lượng chứa trong xung  $g(t)$  được chuẩn hoá bằng 1. Đặc biệt, chúng ta xét trường hợp trong đó biên độ tín hiệu nhận một trong 4 giá trị phân cách đều nhau, cụ thể là  $\{A_m\} = \{-3d, -d, d, 3d\}$  hay tương đương là

$$A_m = (2m-3)d, \quad m=0, 1, 2, 3 \quad (5.3.3)$$

trong đó  $2d$  là khoảng cách Euclide giữa hai mức tín hiệu nằm cạnh nhau. Bốn dạng sóng tín hiệu này được minh họa trên hình 5.19. Chúng ta gọi tập hợp này của các dạng sóng là các tín hiệu được điều chế biên xung (PAM: Pulse-Amplitude Modulated).



Hình 5.19 Các dạng sóng tín hiệu nhiều biên độ.

Bốn dạng sóng tín hiệu PAM như trên hình 5.19 có thể được sử dụng để truyền dẫn hai bít thông tin trên một dạng sóng. Như vậy, chúng ta gán các cặp bít thông tin sau đây thành bốn dạng sóng tín hiệu:

$$00 \rightarrow s_0(t)$$

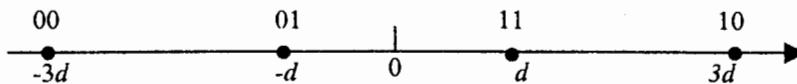
$$01 \rightarrow s_1(t)$$

$$10 \rightarrow s_2(t)$$

$$11 \rightarrow s_3(t)$$

Mỗi một cặp bít thông tin  $\{00, 01, 10, 11\}$  được gọi là một ký hiệu hay một dấu (*symbol*) và thời gian tồn tại  $T$  được gọi là khoảng thời gian của *symbol*. Hãy để ý rằng nếu tốc độ bít là  $R=1/T_b$ , thời gian của một *symbol*  $T=2T_b$ . Do tất cả các dạng sóng tín hiệu đều là các phiên bản tỷ lệ của hàm cơ sở của tín hiệu  $g(t)$  nên

các dạng sóng tín hiệu này có thể biểu diễn được một cách hình học như các điểm trên đường thẳng trực thực. Như vậy, biểu diễn hình học của 4 tín hiệu PAM là *biểu đồ sao của tín hiệu (signal constellation diagram)* được thể hiện trên hình 5.20.



Hình 5.20 Biểu đồ sao của tín hiệu đối với 4 dạng sóng tín hiệu PAM.

Cũng như trong trường hợp các tín hiệu nhị phân, chúng ta cũng giả sử rằng các dạng sóng tín hiệu PAM được truyền qua một kênh AWGN. Hệ quả là, tín hiệu thu được biểu diễn được theo

$$r(t) = s_i(t) + n(t), \quad i = 0, 1, 2, 3, \quad 0 \leq t \leq T, \quad (5.3.4)$$

trong đó  $n(t)$  là một hàm mẫu của một quá trình tạp âm Gauss trắng với phổ công suất  $N_0/2$  W/Hz. Nhiệm vụ của máy thu là xác định xem dạng sóng nào trong bốn dạng sóng tín hiệu đã được truyền đi sau khi quan sát tín hiệu thu được  $r(t)$  trong khoảng thời gian  $0 \leq t \leq T$ . Máy thu tối ưu được thiết kế để tối thiểu hóa xác suất của một lỗi *symbol*.

### 5.3.2 MÁY THU TỐI ƯU ĐỐI VỚI KÊNH AWGN

Máy thu làm tối thiểu xác suất lỗi được thực hiện bằng cách cho tín hiệu qua một bộ tương quan tín hiệu hay một bộ lọc phồi hợp rồi được đưa tiếp tới một bộ tách biên độ. Do cả bộ tương quan tín hiệu lẫn mạch lọc tín hiệu đều thu được cùng tín hiệu lỗi ra như nhau tại thời điểm lấy mẫu nên trong việc đề cập của ta chúng ta sẽ chỉ xét bộ tương quan tín hiệu.

#### Bộ tương quan tín hiệu

Bộ tương quan tín hiệu tính tương quan chéo tín hiệu thu được  $r(t)$  với xung tín hiệu  $g(t)$  và lỗi ra của nó được lấy mẫu tại  $t = T$ . Như thế, tín hiệu lỗi ra bộ tương quan tín hiệu là

$$r = \int_0^T r(t)g(t)dt = \int_0^T A_i g^2(t)dt + \int_0^T g(t)n(t)dt = A_i + n \quad (5.3.5)$$

trong đó  $n$  biểu thị thành phần tạp âm, được xác định theo

$$n = \int_0^T g(t)n(t)dt \quad (5.3.6)$$

Chú ý rằng  $n$  là một biến ngẫu nhiên Gauss với kỳ vọng

$$E(n) = \int_0^T g(t)E[n(t)]dt = 0 \quad (5.3.7)$$

và phương sai

$$\begin{aligned} \sigma^2 &= E(n^2) = \int_0^T \int_0^T g(t)g(\tau)E[n(t)n(\tau)]dtd\tau \\ &= \frac{N_0}{2} \int_0^T \int_0^T g(t)g(\tau)\delta(t-\tau)dtd\tau = \frac{N_0}{2} \int_0^T g^2(t)dt \\ &= \frac{N_0}{2} \end{aligned} \quad (5.3.8)$$

Do đó, hàm mật độ xác suất của tín hiệu lối ra  $r$  của bộ tách quan tín hiệu là

$$p(r|s_i(t) \text{ đã được truyền đi}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(r-A_i)^2/2\sigma^2} \quad (5.3.9)$$

trong đó  $A_i$  là một trong bốn giá trị biên độ có thể có.

### Bộ tách tín hiệu (tách biên độ)

Bộ tách biên độ quan sát lối ra  $r$  của bộ tách quan và quyết định tín hiệu nào trong bốn tín hiệu PAM đã được truyền đi trong khoảng thời gian của tín hiệu. Trong tính toán sau đây về chất lượng của máy thu tối ưu, chúng ta giả sử rằng cả bốn mức biên độ có thể có đều có cùng khả năng.

Do biên độ tín hiệu nhận được  $A_i$  có thể nhận các giá trị  $\pm d, \pm 3d$ , như đã được minh họa trên biểu đồ sao của tín hiệu hình 5.20, nên bộ tách biên độ tối ưu so sánh lối ra  $r$  của bộ tách quan với bốn mức biên độ đã truyền đi có thể có và chọn ra mức biên độ gần nhất với  $r$  theo khoảng cách Euclidean. Như thế, bộ tách biên độ tối ưu tính các khoảng cách

$$D_i = |r - A_i|, \quad i = 0, 1, 2, 3 \quad (5.3.10)$$

và chọn biên độ tương ứng với khoảng cách nhỏ nhất.

Chúng ta hãy để ý rằng một lỗi quyết định xảy ra khi biến ngẫu nhiên tạp âm  $n$  về biên độ vượt quá một nửa khoảng cách giữa các mức biên độ, nghĩa là khi  $|n| > d$ . Tuy nhiên, khi mức biên độ  $+3d$  và  $-3d$  thì lỗi chỉ có thể xảy ra về một phía. Do bốn mức biên độ đồng khả năng, xác suất lỗi một *symbol* trung bình là

$$\begin{aligned} P_4 &= \frac{3}{4} P(|r - A_m| > d) = \frac{3}{2} \int_d^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} dx \\ &= \frac{3}{2} \int_{d/\sigma}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2} dx = \frac{3}{2} Q\left(\sqrt{\frac{d^2}{\sigma^2}}\right) = \frac{3}{2} Q\left(\sqrt{\frac{2d^2}{N_0}}\right) \end{aligned} \quad (5.3.11)$$

Chúng ta thấy rằng bình phương khoảng cách giữa các mức biên độ kề nhau là  $(2d)^2 \equiv \delta^2$ . Do đó, xác suất lỗi trung bình có thể biểu diễn được theo

$$P_4 = \frac{3}{2} Q\left(\sqrt{\frac{\delta^2}{2N_0}}\right) \quad (5.3.12)$$

Một cách khác, xác suất lỗi trung bình có thể biểu diễn được theo năng lượng tín hiệu. Do tất cả các mức biên độ là đồng khả năng nên năng lượng tín hiệu trung bình được truyền đi trên một *symbol* là

$$E_{av} = \frac{1}{4} \sum_{k=1}^4 \int_0^T s_k^2(t) dt = 5d^2 \quad (5.3.13)$$

Hệ quả là  $d^2 = E_{av}/5$  và do vậy

$$P_4 = \frac{3}{2} Q\left(\sqrt{\frac{2E_{av}}{5N_0}}\right) \quad (5.3.14)$$

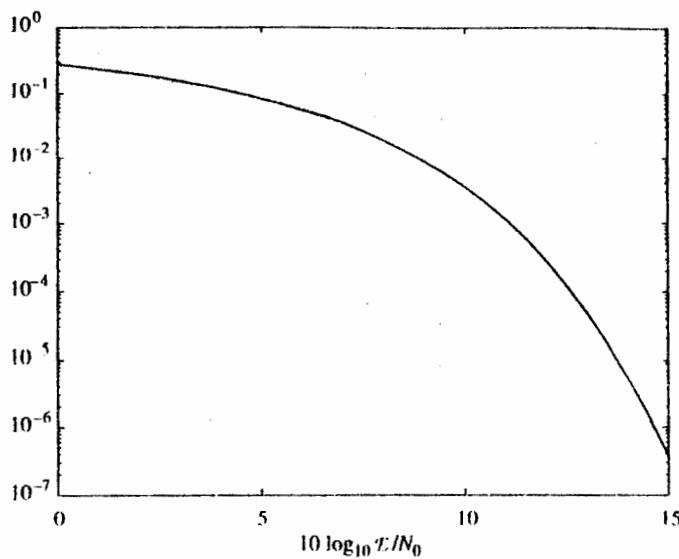
Do mỗi một *symbol* được truyền đi gồm hai bít thông tin, năng lượng phát trung bình trên một bít là  $E_{av}/2 \equiv E_{avb}$ .

Xác suất lỗi trung bình  $P_4$  được vẽ trên hình 5.21 như một hàm của SNR được định nghĩa là  $10\log_{10}(E_{avb}/N_0)$ .

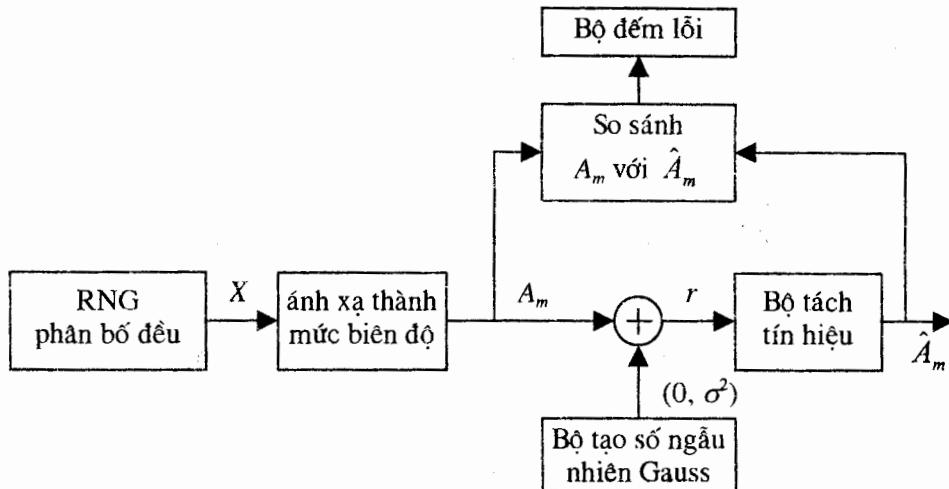
### BÀI TẬP MINH HỌA

**Bài tập minh họa 5.8 [Mô phỏng tín hiệu nhiều biên độ]** Hãy thực hiện một mô phỏng Monte-Carlo của hệ thống truyền tin PAM bốn mức có sử dụng một bộ tương quan tín hiệu, như đã được mô tả ở trên, măc nối tiếp với một bộ

tách biên độ. Mô hình đối với hệ thống được đưa vào mô phỏng này được thể hiện trên hình 5.22.



Hình 5.21 Xác suất lỗi *symbol* đối với PAM bốn mức.



Hình 5.22 Sơ đồ khối của PAM bốn mức để mô phỏng Monte-Carlo.

### Lời giải

Như đã hướng dẫn, chúng ta mô phỏng việc tạo biến ngẫu nhiên  $r$  là lỗi ra của bộ tương quan tín hiệu và là lỗi vào của bộ tách biên độ. Chúng ta bắt đầu bằng việc tạo ra một dãy các *symbol* bốn mức mà chúng rồi sẽ được ánh xạ thành

các biên độ tương ứng  $\{A_m\}$ . Để hoàn thành nhiệm vụ này, chúng ta sử dụng một bộ tạo số ngẫu nhiên tạo ra một số ngẫu nhiên phân bố đều trên dải  $(0, 1)$ . Dải này được chia nhỏ thành 4 khoảng bằng nhau  $(0, 0.25)$ ,  $(0.25, 0.5)$ ,  $(0.5, 0.75)$  và  $(0.75, 1)$ , trong đó các khoảng con lần lượt ứng với các *symbol* (các cặp bít mang tin) 00, 01, 10 và 11. Như thế, sản phẩm đầu ra của bộ tạo số ngẫu nhiên phân bố đều được ánh xạ thành các mức biên độ tín hiệu tương ứng  $(-3d, -d, d, 3d)$ .

Thành phần tạp âm cộng có kỳ vọng bằng 0 và phương sai  $\sigma^2$  được tạo ra bằng một bộ tạo số ngẫu nhiên Gauss. Để thuận tiện, chúng ta có thể chuẩn hoá tham số khoảng cách  $d = 1$  và cho  $\sigma^2$  biến thiên. Bộ tách biên độ quan sát  $r = A_m + n$  và tính khoảng cách giữa  $r$  và bốn mức biên độ được truyền đi có thể có. Lối ra  $\hat{A}_m$  của nó là biên độ ứng với khoảng cách nhỏ nhất.  $\hat{A}_m$  được so sánh với biên độ tín hiệu thực sự đã truyền đi và một bộ đếm lỗi được sử dụng để đếm các lỗi sinh ra do bộ tách biên độ quyết định nhầm.

Hình 5.23 minh họa các kết quả của mô phỏng với việc truyền  $N = 10\,000$  *symbol* tại các giá trị khác nhau của SNR bít trung bình được xác định theo

$$\frac{E_{avb}}{N_0} = \frac{5}{4} \left( \frac{d^2}{\sigma^2} \right) \quad (5.3.15)$$

Hãy để ý mức độ sát hợp giữa các kết quả mô phỏng và các giá trị lý thuyết của  $P_4$  được tính từ (5.3.14).

Các MATLAB *script* cho bài tập này được cho dưới đây.

### m-file

*% MATLAB script for Illustrated Problem 8, Chapter 5.*

```

echo on
SNRindB1=0:1:12;
SNRindB2=0:0.1:12;
for i=1:length(SNRindB1),
    % simulated error rate
    smld_err_prb(i)=smldPe58(SNRindB1(i));
    echo off;
end;
```

```

echo on;
for i=1:length(SNRindB2),
    % signal to noise ratio
    SNR_per_bit=exp(SNRindB2(i)*log(10)/10);
    % theoretical error rate
    theo_err_prb(i)=(3/2)*Qfunct(sqrt((4/5)*SNR_per_bit));
    echo off;
end;
echo on;
% Plotting commands follow
semilogy(SNRindB1,smld_err_prb,'*');
hold
semilogy(SNRindB2,theo_err_prb);

```

**m-file**

```

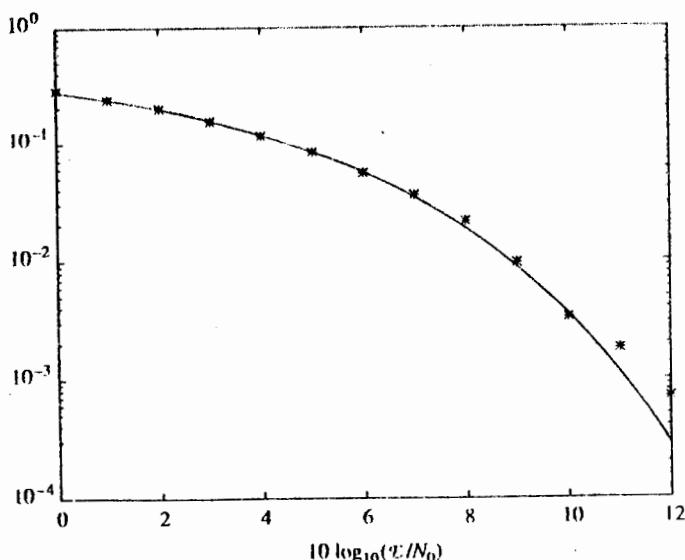
function [p]=smldPe58(snr_in_dB)
% [p]=smldPe58(snr_in_dB)
%      SMLDPE58 simulates the probability of error for the given
%      snr_in_dB, signal to noise ratio in dB.
d=1;
SNR=exp(snr_in_dB*log(10)/10);           % signal to noise ratio per bit
sgma=sqrt((5*d^2)/(4*SNR));             % sigma, standard deviation of noise
N=10000;                                  % number of symbols being simulated
% generation of the quarternary data source follows
for i=1:N,
    temp=rand;                            % a uniform random variable over (0,1)
    if (temp<0.25),                       % with probability 1/4, source output is "00"
        dsource(i)=0;
    elseif (temp<0.5),
        dsource(i)=1;                      % with probability 1/4, source output is "01"
    elseif (temp<0.75),
        dsource(i)=2;                      % with probability 1/4, source output is "10"
    else
        dsource(i)=3;                      % with probability 1/4, source output is "11"
    end
end;
% detection, and probability of error calculation
numoferr=0;
for i=1:N,
    % The matched filter outputs
    if (dsource(i)==0),

```

```

r=-3*d+gngauss(sgma);           % if the source output is "00"
elseif (dsource(i)==1),
    r=-d+gngauss(sgma);         % if the source output is "01"
elseif (dsource(i)==2)
    r=d+gngauss(sgma);         % if the source output is "10"
else
    r=3*d+gngauss(sgma);       % if the source output is "11"
end;
% detector follows
if (r<-2*d),
    decis=0;                   % decision is "00"
elseif (r<0),
    decis=1;                   % decision is "01"
elseif (r<2*d),
    decis=2;                   % decision is "10"
else
    decis=3;                   % decision is "11"
end;
if (decis~=dsource(i)),
    numoferr=numoferr+1;
end;
end;
p=numoferr/N;                   % probability of error estimate

```



Hình 5.23 Xác suất lỗi đối với mô phỏng Monte-Carlo  
so sánh với xác suất lỗi lý thuyết đối với 4-PAM.

### 5.3.3 CÁC DẠNG SÓNG TÍN HIỆU VỚI NHIỀU MỨC BIÊN ĐỘ

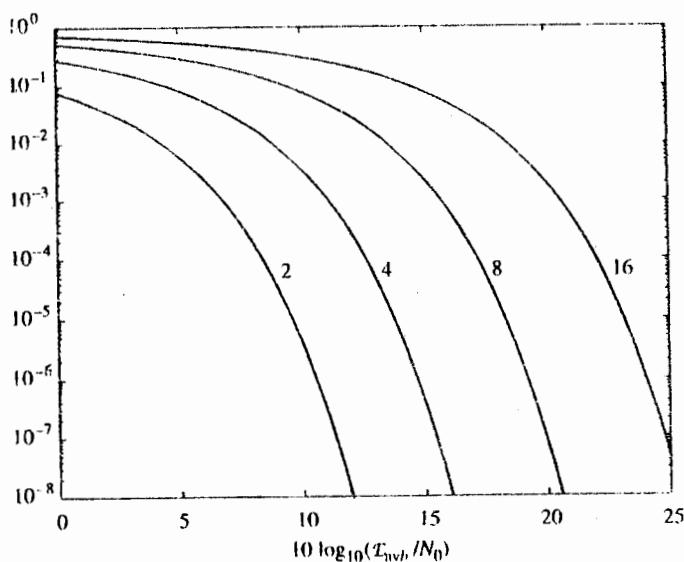
Việc xây dựng các tín hiệu nhiều hơn bốn mức biên độ thì tương đối đơn giản. Nói chung, một tập gồm  $M = 2^k$  dạng sóng tín hiệu nhiều mức biên độ biểu diễn được theo

$$s_m(t) = A_m g(t), \quad 0 \leq t \leq T, \quad m = 0, 1, 2, \dots, M-1$$

trong đó  $M$  giá trị biên độ được phân cách đều và được cho theo

$$A_m = (2m - M + 1)d, \quad m = 0, 1, 2, \dots, M-1 \quad (5.3.16)$$

còn  $g(t)$  là một xung hình chữ nhật đã được định nghĩa trong (5.3.2). Mỗi một dạng sóng tín hiệu tải  $k = \log_2 M$  bít thông tin. Khi tốc độ bít là  $R = 1/T_b$ , thì tốc độ symbol tương ứng là  $1/T = 1/kT_b$ . Giống như trong trường hợp PAM bốn mức, máy thu cũng gồm một bộ tách tín hiệu (hay một bộ lọc phôi hợp) mắc nối tiếp với một bộ tách biên độ thực hiện tính các khoảng cách Euclide được cho bởi (5.3.10) đối với  $m = 0, 1, 2, \dots, M-1$ . Đối với các mức biên độ đồng khả năng, quyết định được biểu quyết cho mức biên độ nào ứng với khoảng cách nhỏ nhất.



Hình 5.24 Xác suất lỗi symbol (SER) đối với M-PAM, với  $M=2, 4, 8, 16$ .

Xác suất lỗi đối với bộ tách tín hiệu tối ưu trong hệ thống PAM  $M$  mức (M-PAM) dễ dàng chỉ ra được là

$$P_M = \frac{2(M-1)}{M} Q\left(\sqrt{\frac{6E_{avb} \log_2 M}{(M^2-1)N_0}}\right) \quad (5.3.17)$$

trong đó  $E_{avb}$  là năng lượng trung bình của tín hiệu tính trên một bít mang tin. Hình 5.24 minh họa xác suất lỗi *symbol* đối với  $M=2, 4, 8, 16$ .

### BÀI TẬP MINH HOA

**Bài tập minh họa 5.9 [Mô phỏng PAM]** Hãy thực hiện một mô phỏng Monte-Carlo của một hệ thống truyền tin số PAM 16 mức và hãy đánh giá chất lượng tỷ lệ lỗi của nó.

#### Lời giải

Sơ đồ khối cơ bản được trình bày trên hình 5.22 được áp dụng một cách tổng quát. Một bộ tạo số ngẫu nhiên phân bố đều được sử dụng để tạo chuỗi các *symbol* mang thông tin mà chúng được xem như các khối 4 bít thông tin. Các *symbol* 16 phân có thể tạo được một cách trực tiếp bằng cách chia nhỏ khoảng (0, 1) thành 16 khoảng con rộng như nhau rồi sau đó ánh xạ các *symbol* 16 phân thành các biên độ tín hiệu 16 mức. Một chuỗi tạp âm Gauss trắng được cộng vào chuỗi *symbol* mang tin 16 mức và tín hiệu cộng tạp âm thu được được cấp vào bộ tách biên độ. Bộ tách biên độ tính các *metric* khoảng cách được cho theo (5.3.10) và chọn ra biên độ ứng với *metric* nhỏ nhất. Lỗi ra bộ tách biên độ được so sánh với chuỗi *symbol* thông tin đã được truyền và các lỗi sẽ được đếm. Hình 5.25 minh họa tỷ lệ lỗi đánh giá được đổi với 10 000 *symbol* được truyền và tỷ lệ lỗi *symbol* lý thuyết tính theo (5.3.17) với  $M=16$ .

Các MATLAB *script* cho bài tập này được cho dưới đây.

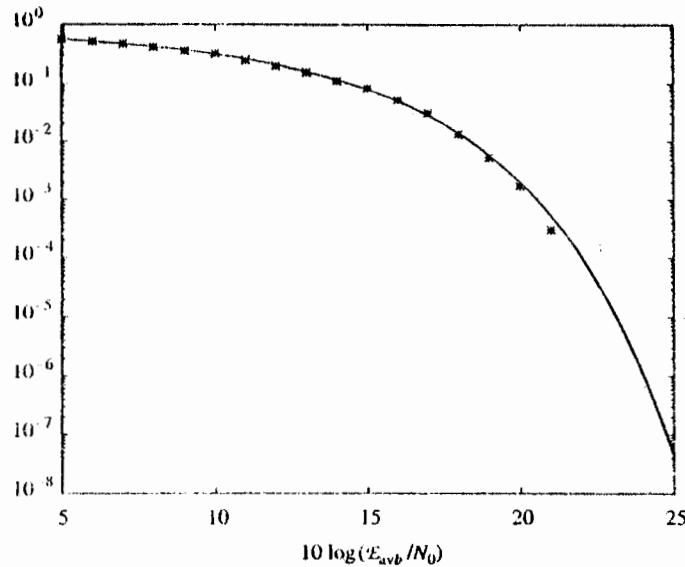
#### m-file

```
% MATLAB script for Illustrative Problem 9, Chapter 5.
echo on
SNRindB1=5:1:25;
SNRindB2=5:0.1:25;
M=16;
for i=1:length(SNRindB1),
    % simulated error rate
```

```

smld_err_prb(i)=smldPe59(SNRindB1(i));
echo off;
end;
echo on ;
for i=1:length(SNRindB2),
SNR_per_bit=exp(SNRindB2(i)*log(10)/10);
% theoretical error rate
theo_err_prb(i)=(2*(M-1)/M)*Qfunct(sqrt((6*log2(M)/(M^2-1))*SNR_per_bit));
echo off;
end;
echo on;
% Plotting commands follow
semilogy(SNRindB1,smld_err_prb,'*');
hold
semilogy(SNRindB2,theo_err_prb);

```



Hình 5.25 Tỷ lệ lỗi từ mô phỏng Monte-Carlo so sánh với xác suất lỗi lý thuyết đối với 16-PAM.

### m-file

```

function [p]=smldPe59(snr_in_dB)
% [p]=smldPe59(snr_in_dB)
% SMLDPE59 simulates the error probability for the given
% snr_in_dB, signal to noise ratio in dB.
M=16; % 16-ary PAM
d=1;

```

```

SNR=exp(snr_in_dB*log(10)/10); % signal to noise ratio per bit
sgma=sqrt((85*d^2)/(8*SNR)); % sigma, standard deviation of noise
N=10000; % number of symbols being simulated
% generation of the quarternary data source
for i=1:N,
    temp=rand; % a uniform random variable over (0,1)
    index=floor(M*temp); % the index is an integer from 0 to M-1, where
    % all the possible values are equally likely
    dsource(i)=index;
end;
% detection, and probability of error calculation
numoferr=0;
for i=1:N,
    % matched filter outputs
    %  $(2*dsource(i)-M+1)*d$  is the mapping to the 16-ary constellation
    r=(2*dsource(i)-M+1)*d+gngauss(sgma);
    % the detector
    if (r>(M-2)*d),
        decis=15;
    elseif (r>(M-4)*d),
        decis=14;
    elseif (r>(M-6)*d),
        decis=13;
    elseif (r>(M-8)*d),
        decis=12;
    elseif (r>(M-10)*d),
        decis=11;
    elseif (r>(M-12)*d),
        decis=10;
    elseif (r>(M-14)*d),
        decis=9;
    elseif (r>(M-16)*d),
        decis=8;
    elseif (r>(M-18)*d),
        decis=7;
    elseif (r>(M-20)*d),
        decis=6;
    elseif (r>(M-22)*d),
        decis=5;
    elseif (r>(M-24)*d),
        decis=4;
    elseif (r>(M-26)*d),
        decis=3;

```

```

elseif (r>(M-28)*d),
decis=2;
elseif (r>(M-30)*d),
decis=1;
else
decis=0;
end;
if (decis~=dsource(i)),    % if it is an error, increase the error counter
numoferr=numoferr+1;
end;
end;
p=numoferr/N;             % probability of error estimate

```

## 5.4 CÁC TÍN HIỆU NHIỀU CHIỀU

Trong mục trước chúng ta đã xây dựng các dạng sóng tín hiệu nhiều biến độ cho phép chúng ta truyền nhiều bít trên một dạng sóng tín hiệu. Như thế, với các dạng sóng tín hiệu có  $M = 2^k$  mức biến độ chúng ta có thể truyền được  $k = \log_2 M$  bít thông tin trên một dạng sóng tín hiệu. Chúng ta cũng đã thấy được rằng các tín hiệu nhiều mức biến độ có thể biểu diễn một cách hình học được như  $M$  điểm trên đường thẳng trực thực (xem hình 5.20). Các dạng sóng tín hiệu như thế được gọi là các tín hiệu một chiều (*one-dimensional*).

Trong mục này, chúng ta sẽ xét việc xây dựng một lớp gồm  $M = 2^k$  dạng sóng tín hiệu mà chúng có biểu diễn hình học nhiều chiều. Nghĩa là, tập các dạng sóng tín hiệu có thể biểu diễn hình học được bằng các điểm trong một không gian  $N$  chiều. Chúng ta cũng đã thấy được rằng các tín hiệu trực giao nhị phân biểu diễn được một cách hình học như các điểm trong không gian hai chiều.

### 5.4.1 CÁC TÍN HIỆU TRỰC GIAO NHIỀU CHIỀU

Có nhiều cách để xây dựng các dạng sóng tín hiệu nhiều chiều với nhiều tính chất khác nhau. Trong mục này chúng ta sẽ xem xét việc xây dựng một tập gồm  $M = 2^k$  dạng sóng  $s_i(t)$ , đối với  $i = 0, 1, 2, \dots, M-1$ , mà chúng có các tính chất (a) tính trực giao lẫn nhau và (b) có cùng năng lượng. Hai tính chất này có thể biểu diễn

được một cách cô đọng như sau

$$\int_0^T s_i(t) s_k(t) dt = E \delta_{ik}, \quad i, k = 0, 1, 2, \dots, M-1 \quad (5.4.1)$$

trong đó  $E$  là năng lượng của mỗi một dạng sóng tín hiệu còn  $\delta_{ik}$  được gọi là *delta Kronecker*, được định nghĩa theo

$$\delta_i^k = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases} \quad (5.4.2)$$

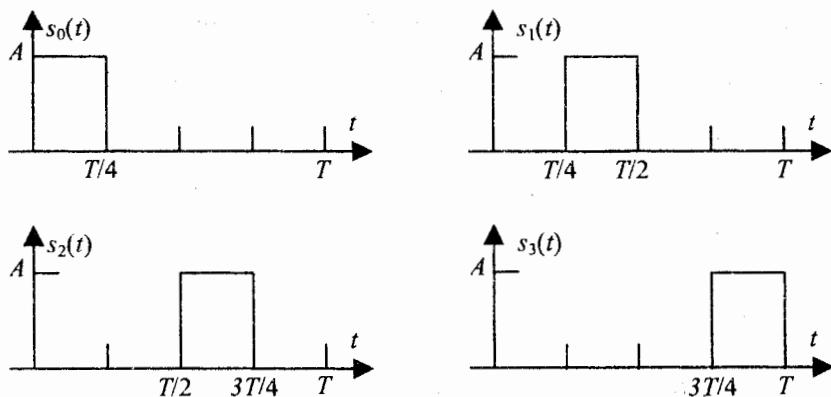
Giống như trong thảo luận trước đây của ta, chúng ta giả sử rằng một nguồn thông tin cung cấp một chuỗi các bít thông tin mà chúng thì được truyền qua một kênh truyền tin. Các bít thông tin xuất hiện với một tốc độ đều  $R$  bít/giây. Nghịch đảo của  $R$  là khoảng thời gian của một bít  $T_b$ . Bộ điều chế nhận  $k$  bít một lần và ánh xạ chúng thành một trong  $M = 2^k$  dạng sóng tín hiệu. Mỗi một khối  $k$  bít đó được gọi là một *symbol*. Khoảng thời gian có được để truyền mỗi một *symbol* là  $T = kT_b$ . Do vậy  $T$  là thời gian của một *symbol*.

Cách đơn giản nhất để xây dựng một tập gồm  $M = 2^k$  dạng sóng trực giao cùng năng lượng trong khoảng  $(0, T)$  là chia nhỏ khoảng thời gian này thành  $M$  khoảng con đều nhau có thời gian tồn tại là  $T/M$  và gán một dạng sóng tín hiệu cho mỗi một khoảng con đó. Hình 5.26 minh họa một cách xây dựng như vậy đối với  $M = 4$  tín hiệu. Tất cả các dạng sóng tín hiệu được xây dựng theo cách này có cùng năng lượng được cho theo

$$\begin{aligned} E &= \int_0^T s_i^2(t) dt, \quad i = 0, 1, 2, \dots, M-1 \\ &= \frac{A^2 T}{M} \end{aligned} \quad (5.4.3)$$

Tập các dạng sóng trực giao như thế có thể biểu diễn được như một tập gồm  $M$  véc-tơ trực giao, tức là

$$\begin{aligned} s_0 &= (\sqrt{E}, 0, 0, \dots, 0) \\ s_1 &= (0, \sqrt{E}, 0, \dots, 0) \\ &\dots \\ s_{M-1} &= (0, 0, 0, \dots, \sqrt{E}) \end{aligned} \quad (5.4.4)$$



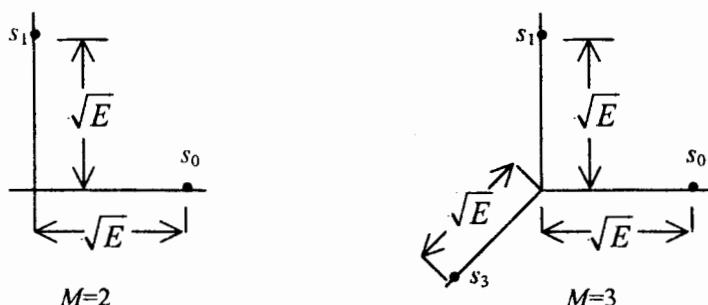
Hình 5.26 Một thí dụ về 4 dạng sóng tín hiệu trực giao cùng năng lượng.

Hình 5.27 minh họa các điểm tín hiệu (các biểu đồ sao của tín hiệu) tương ứng với  $M=2$  và  $M=3$  tín hiệu trực giao.

Ta hãy giả sử rằng các dạng sóng tín hiệu trực giao này được sử dụng để truyền thông tin qua một kênh AWGN. Hệ quả là, nếu dạng sóng đã truyền là  $s_i(t)$  thì dạng sóng thu được là

$$r(t) = s_i(t) + n(t), \quad 0 \leq t \leq T, \quad i=0, 1, 2, \dots, M-1 \quad (5.4.5)$$

trong đó  $n(t)$  là một hàm mẫu của một quá trình tạp âm Gauss trắng với phổ công suất  $N_0/2$  [W/Hz]. Máy thu quan sát tín hiệu  $r(t)$  và quyết định xem dạng sóng nào trong  $M$  dạng sóng đã được truyền đi.

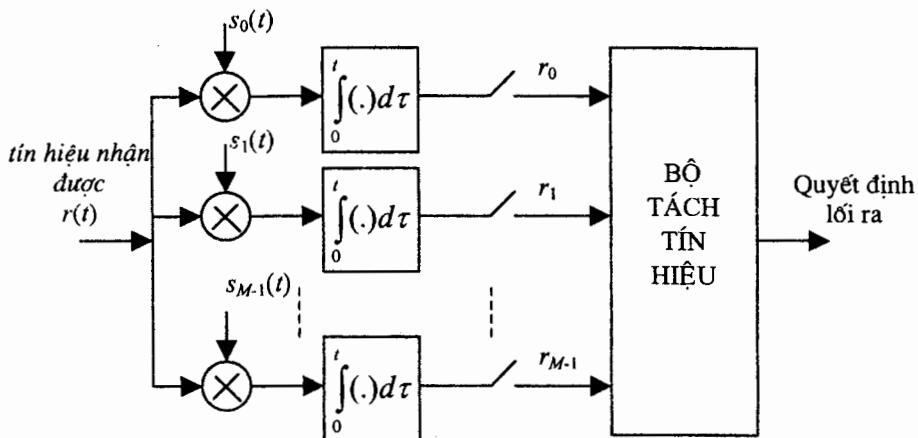


Hình 5.27 Các biểu đồ sao tín hiệu đối với các tín hiệu trực giao với  $M=2$  và  $M=3$ .

### **Máy thu tối ưu đối với kênh AWGN**

Máy thu tối thiểu hóa xác suất lỗi trước hết cho tín hiệu  $r(t)$  qua một dãy song song gồm  $M$  bộ tương quan. Do các bộ tương quan tín hiệu và các bộ lọc phối hợp

có được cùng các tín hiệu lối ra tại thời điểm lấy mẫu nên ta hãy xét trường hợp các bộ tương quan tín hiệu được sử dụng, như được thể hiện trên hình 5.28.



Hình 5.28 Máy thu tối ưu đối với các tín hiệu trực giao nhiều chiều.

### Các bộ tương quan tín hiệu

Tín hiệu thu được  $r(t)$  được tính tương quan chéo với từng dạng sóng trong  $M$  dạng sóng tín hiệu và các lối ra của các bộ tương quan được lấy mẫu tại  $t=T$ . Như vậy,  $M$  lối ra của các bộ tương quan là

$$r_i = \int_0^T r(t)s_i(t)dt, \quad i = 0, 1, 2, \dots, M-1 \quad (5.4.6)$$

mà chúng có thể biểu diễn được dưới dạng véc-tơ:  $\mathbf{r} = [r_0, r_1, \dots, r_{M-1}]'$ . Giả sử rằng dạng sóng tín hiệu  $s_0(t)$  đã được truyền đi. Khi đó

$$r_0 = \int_0^T s_0^2(t)dt + \int_0^T n(t)s_0(t)dt = E + n_0 \quad (5.4.7)$$

còn

$$\begin{aligned} r_i &= \int_0^T s_0(t)s_i(t)dt + \int_0^T n(t)s_i(t)dt \\ &= \int_0^T n(t)s_i(t)dt = n_i, \quad i = 1, 2, 3, \dots, M-1 \end{aligned} \quad (5.4.8)$$

trong đó

$$n_i = \int_0^T n(t)s_i(t)dt \quad (5.4.9)$$

Như vậy, lối ra  $r_0$  gồm một thành phần tín hiệu  $E$  và một thành phần tạp âm  $n_0$ .  $M-1$  lối ra khác chỉ có thành phần tạp âm. Mỗi một trong các thành phần tạp âm đều là biến ngẫu nhiên Gauss với kỳ vọng bằng không và phương sai

$$\begin{aligned}\sigma^2 &= E(n_i^2) = \int_0^T \int_0^T s_i(t)s_i(\tau)E[n(t)n(\tau)]dt d\tau \\ &= \frac{N_0}{2} \int_0^T \int_0^T s_i(t)s_i(\tau)\delta(t-\tau)dt d\tau = \frac{N_0}{2} \int_0^T s_i^2(t)dt = \frac{N_0 E}{2}\end{aligned}\quad (5.4.10)$$

Bạn đọc hãy mạnh dạn chứng minh rằng  $E(n_i n_j) = 0$ , trong đó  $i \neq j$ . Điều đó dẫn đến các hàm mật độ xác suất đối với các lối ra của các bộ tương quan là

$$\begin{aligned}p(r_0 | s_0(t) \text{ đã được truyền đi}) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-(r_0 - E)^2 / 2\sigma^2} \\ p(r_i | s_0(t) \text{ đã được truyền đi}) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-r_i^2 / 2\sigma^2}, \quad i = 1, 2, \dots, M-1\end{aligned}$$

### Bộ tách tín hiệu

Bộ tách tín hiệu tối ưu quan sát  $M$  lối ra  $r_i$  của các bộ tương quan, trong đó  $i = 0, 1, 2, \dots, M-1$ , và chọn ra tín hiệu tương ứng với lối ra bộ tương quan có trị lớn nhất. Trong trường hợp  $s_0(t)$  được truyền đi, xác suất quyết định đúng đơn giản là xác suất để  $r_0 > r_i$  đối với  $i = 1, 2, \dots, M-1$ , hay

$$P_C = P(r_0 > r_1, r_0 > r_2, \dots, r_0 > r_{M-1}) \quad (5.4.11)$$

và xác suất của một lối symbol đơn giản là

$$P_M = 1 - P_C = 1 - P(r_0 > r_1, r_0 > r_2, \dots, r_0 > r_{M-1}) \quad (5.4.12)$$

Có thể chứng minh được rằng  $P_M$  có thể biểu diễn được dưới dạng tích phân theo

$$P_M = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \left\{ 1 - [1 - Q(y)]^{M-1} \right\} e^{-(y - \sqrt{2E/N_0})^2 / 2} dy \quad (5.4.13)$$

Đối với trường hợp đặc biệt  $M=2$ , biểu thức (5.4.13) rút lại còn

$$P_2 = Q\left(\sqrt{\frac{E_b}{N_0}}\right)$$

mà nó chính là kết quả chúng ta đã thu được trong mục 5.2 đối với tín hiệu trực giao nhị phân.

Cùng một biểu thức về xác suất lỗi nhận được khi bất kỳ một trong  $M-1$  tín hiệu khác được truyền đi. Do tất cả  $M$  tín hiệu là đồng xác suất, biểu thức đổi với  $P_M$  được cho bởi (5.4.13) chính là xác suất lỗi *symbol* trung bình. Xác suất này có thể đánh giá được bằng phương pháp tính số.

Đôi khi chúng ta muốn biến đổi xác suất của một lỗi *symbol* thành một xác suất tương đương của một lỗi *digit* nhị phân (bít). Đối với các tín hiệu trực giao đồng xác suất, mọi lỗi *symbol* là đồng khả năng và xảy ra với xác suất

$$\frac{P_M}{M-1} = \frac{P_M}{2^k - 1} \quad (5.4.14)$$

Hơn thế nữa, có cả thảy  $\binom{k}{n}^*$  cách trong đó  $n$  bít trong  $k$  bít của một *symbol* có thể bị lỗi. Do vậy, số lượng trung bình các lỗi bít trên một *symbol*  $k$  bít bị lỗi là

$$\sum_{n=1}^k n \binom{k}{n} \frac{P_M}{2^k - 1} = k \frac{2^{k-1}}{2^k - 1} P_M \quad (5.4.15)$$

và xác suất lỗi bít trung bình thì chính là kết quả trong (5.4.15) được chia cho  $k$  là số bít của một *symbol*. Như vậy,

$$P_b = \frac{2^{k-1}}{2^k - 1} P_M \quad (5.4.16)$$

Các đồ thị của xác suất của một lỗi *digit* nhị phân như một hàm của SNR tính theo bít,  $E_b/N_0$ , được thể hiện trên hình 5.29 đối với  $M=2, 4, 8, 16, 32, 64$ , trong đó  $E_b=E/k$  là năng lượng tính trên một bít. Hình vẽ này cho thấy rõ rằng bằng cách tăng số  $M$  các dạng sóng ta có thể giảm SNR theo bít cần thiết để đạt được một xác suất lỗi bít đã cho.

MATLAB script để tính xác suất lỗi trong (5.4.13) được cho dưới đây.

### m-file

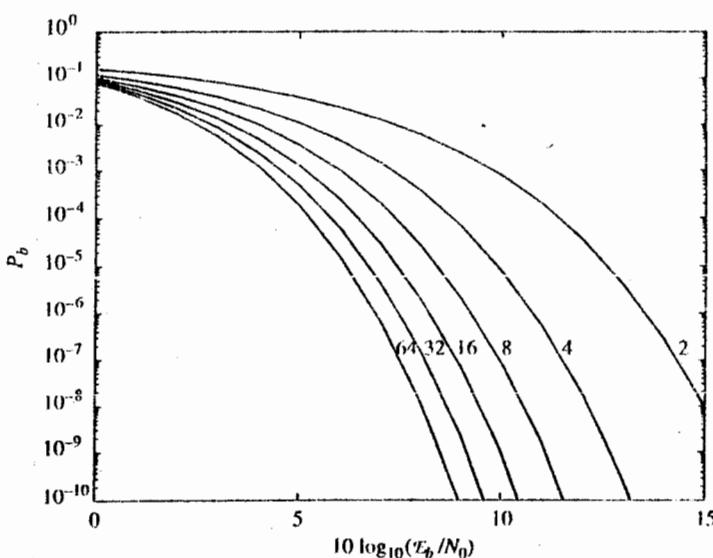
```
% MATLAB script that generates the probability of error versus the signal to noise ratio
initial_snr=0;
final_snr=15;
```

\*  $\binom{k}{n}$  là ký hiệu tổ hợp chập  $n$  của  $k$  phần tử, được xác định theo  $\binom{k}{n} = \frac{k!}{n!(k-n)!}$ . (ND.)

```

snr_step=1;
tolerance=1e-7; % Tolerance used for the integration
minus_inf=-20; % This is practically -infinity
plus_inf=20; % This is practically infinity
snr_in_dB=initial_snr:snr_step:final_snr;
for i=1:length(snr_in_dB),
    snr=10^(snr_in_dB(i)/10);
    Pe_2(i)=Qfunct(sqrt(snr));
    Pe_4(i)=(2/3)*quad8('bdt_int',minus_inf,plus_inf,tolerance,[],snr,4);
    Pe_8(i)=(4/7)*quad8('bdt_int',minus_inf,plus_inf,tolerance,[],snr,8);
    Pe_16(i)=(8/15)*quad8('bdt_int',minus_inf,plus_inf,tolerance,[],snr,16);
    Pe_32(i)=(16/31)*quad8('bdt_int',minus_inf,plus_inf,tolerance,[],snr,32);
    Pe_64(i)=(32/63)*quad8('bdt_int',minus_inf,plus_inf,tolerance,[],snr,64);
    echo off;
end;
% Plotting commands follow

```

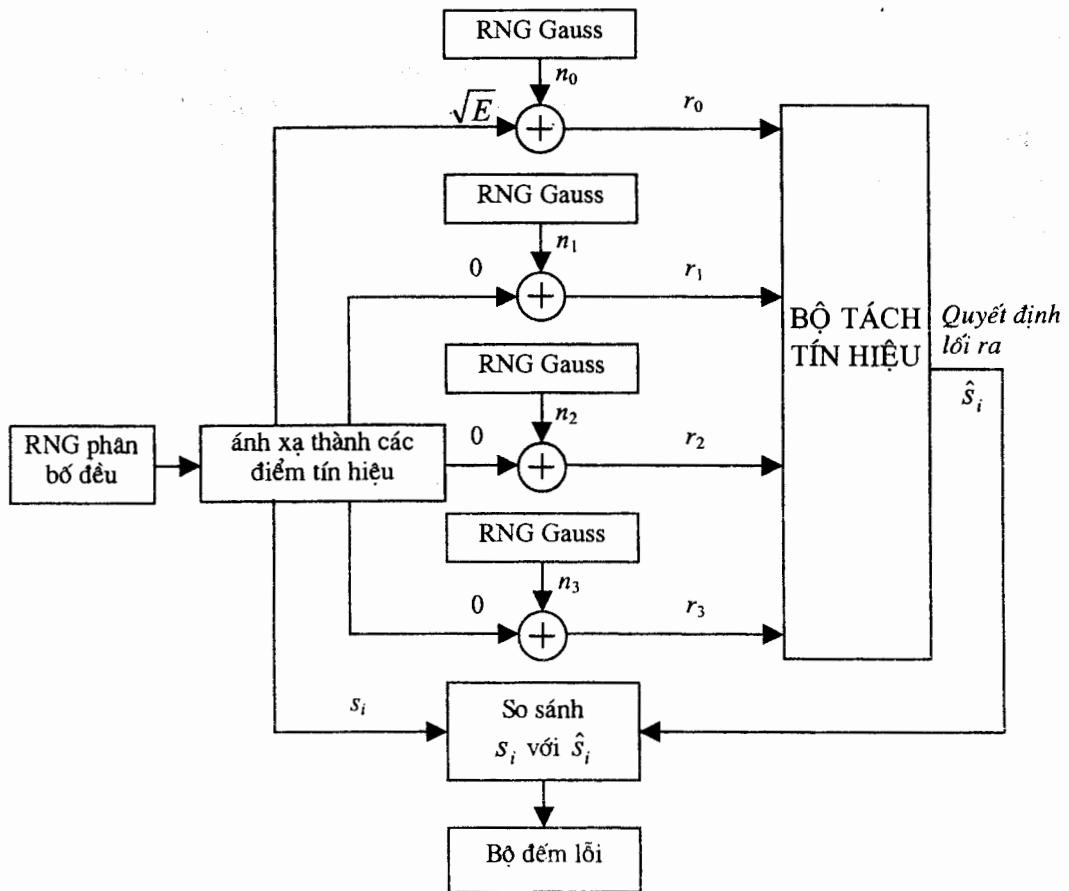


Hình 5.29 Xác suất lỗi bít đối với các tín hiệu trực giao.

## BÀI TẬP MINH HỌA

### Bài tập minh họa 5.10 [Mô phỏng truyền dẫn bằng tín hiệu trực giao]

Hãy thực hiện một mô phỏng Monte-Carlo về một hệ thống truyền tin số áp dụng  $M=4$  tín hiệu trực giao. Mô hình của hệ thống đưa vào mô phỏng được minh họa trên hình 5.30.



Hình 5.30 Sơ đồ khối của hệ thống với  $M=4$  tín hiệu trực giao dùng cho mô phỏng Monte-Carlo.

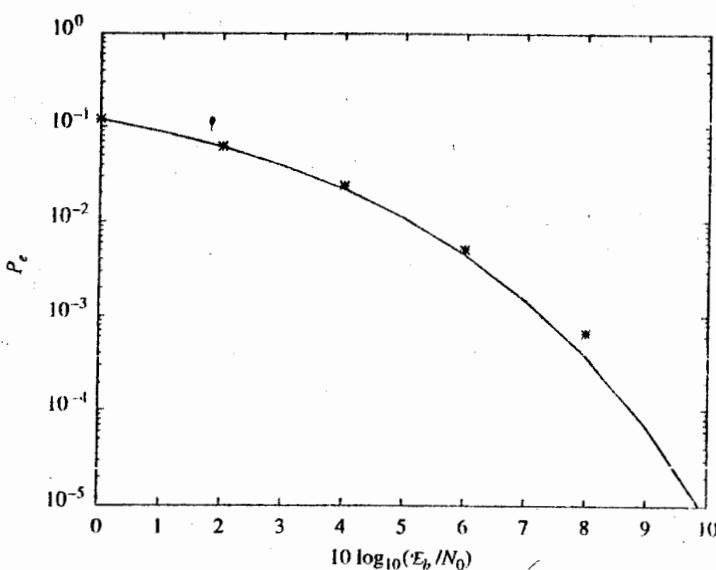
### Lời giải

Như đã hướng dẫn, chúng ta mô phỏng việc tạo các biến ngẫu nhiên  $r_0, r_1, r_2, r_3$  là tín hiệu lỗi vào bộ tách tín hiệu. Chúng ta trước hết tạo ra một chuỗi nhị phân các con số 0 và 1 mà chúng xuất hiện với xác suất như nhau và độc lập thống kê với nhau như trong bài tập minh họa 5.4. Chuỗi nhị phân này được nhóm thành các cặp bít mà chúng sau đó được ánh xạ thành các thành phần tín hiệu tương ứng. Một cách khác đối với việc tạo các bít riêng rẽ là tạo nên các cặp bít như trong bài tập minh họa 5.8. Trong bất kỳ trường hợp nào, chúng ta cũng có được việc ánh xạ của 4 *symbol* thành các điểm tín hiệu

$$\begin{aligned}
 00 &\rightarrow s_0 = (\sqrt{E}, 0, 0, 0); & 01 &\rightarrow s_1 = (0, \sqrt{E}, 0, 0) \\
 10 &\rightarrow s_2 = (0, 0, \sqrt{E}, 0); & 11 &\rightarrow s_3 = (0, 0, 0, \sqrt{E})
 \end{aligned} \tag{5.4.17}$$

Các thành phần tạp âm cộng  $n_0, n_1, n_2, n_3$  được tạo ra bằng bốn bộ tạo tạp âm Gauss, mỗi một trong chúng đều có kỳ vọng bằng 0 và phương sai bằng  $\sigma^2=N_0E/2$ . Để thuận tiện, chúng ta có thể chuẩn hoá năng lượng của một *symbol* thành  $E=1$  và cho  $\sigma^2$  biến thiên. Do  $E=2E_b$  nên  $E_b=1/2$ . Lối ra bộ tách tín hiệu được so sánh với chuỗi bít đã được truyền đi và một bộ đếm lỗi được sử dụng để đếm số các lỗi bít.

Hình 5.31 minh họa các kết quả mô phỏng đối với việc truyền dẫn 20 000 bít tại các giá trị khác nhau của SNR  $E_b/N_0$ . Hãy để ý tới sự thống nhất giữa các kết quả mô phỏng và giá trị lý thuyết của  $P_b$  được cho trong (5.4.16).



Hình 5.31 Xác suất lỗi bít đối với  $M=4$  tín hiệu trực giao nhận được từ mô phỏng Monte-Carlo, so sánh với xác suất lỗi lý thuyết.

Các MATLAB *script* cho bài tập này được cho dưới đây.

### m-file

```
% MATLAB script for Illustrative Problem 10, Chapter 5.
echo on
SNRindB=0:2:10;
for i=1:length(SNRindB),
    % simulated error rate
    smld_err_prb(i)=smldP510(SNRindB(i));
    echo off;
end;
```

```
echo on;  
% Plotting commands follow  
semilogy(SNRindB,smld_err_prb,'*');
```

### m-file

```
function [p]=smldP510(snr_in_dB)  
% [p]=smldP510(snr_in_dB)  
% SMLDP510 simulates the probability of error for the given  
% snr_in_dB, signal to noise ratio in dB.  
M=4; % quarternary orthogonal signalling  
E=1;  
SNR=exp(snr_in_dB*log(10)/10); % signal to noise ratio per bit  
sgma=sqrt(E^2/(4*SNR)); % sigma, standard deviation of noise  
N=10000; % number of symbols being simulated  
% generation of the quarternary data source  
for i=1:N,  
    temp=rand; % a uniform random variable over (0,1)  
    if (temp<0.25),  
        dsource1(i)=0;  
        dsource2(i)=0;  
    elseif (temp<0.5),  
        dsource1(i)=0;  
        dsource2(i)=1;  
    elseif (temp<0.75),  
        dsource1(i)=1;  
        dsource2(i)=0;  
    else  
        dsource1(i)=1;  
        dsource2(i)=1;  
    end  
end;  
% detection, and probability of error calculation  
numoferr=0;  
for i=1:N,  
    % matched filter outputs  
    if ((dsource1(i)==0) & (dsource2(i)==0)),  
        r0=gngauss(sgma);  
        r1=gngauss(sgma);  
        r2=gngauss(sgma);  
        r3=gngauss(sgma);  
    elseif ((dsource1(i)==0) & (dsource2(i)==1)),  
        r0=gngauss(sgma);
```

```

r1=sqrt(E)+gngauss(sgma);
r2=gngauss(sgma);
r3=gngauss(sgma);
elseif ((dsource1(i)==1) & (dsource2(i)==0)),
r0=gngauss(sgma);
r1=gngauss(sgma);
r2=sqrt(E)+gngauss(sgma);
r3=gngauss(sgma);
else
r0=gngauss(sgma);
r1=gngauss(sgma);
r2=gngauss(sgma);
r3=sqrt(E)+gngauss(sgma);
end;
% the detector
max_r=max([r0 r1 r2 r3]);
if (r0==max_r),
decis1=0;
decis2=0;
elseif (r1==max_r),
decis1=0;
decis2=1;
elseif (r2==max_r),
decis1=1;
decis2=0;
else
decis1=1;
decis2=1;
end;
% count the number of bit errors made in this decision
if (decis1~=dsource1(i)),           % if it is an error, increase the error counter
  numoferr=numoferr+1;
end;
if (decis2~=dsource2(i)),           % if it is an error, increase the error counter
  numoferr=numoferr+1;
end;
end;
p=numoferr/(2*N);                  % bit error probability estimate

```

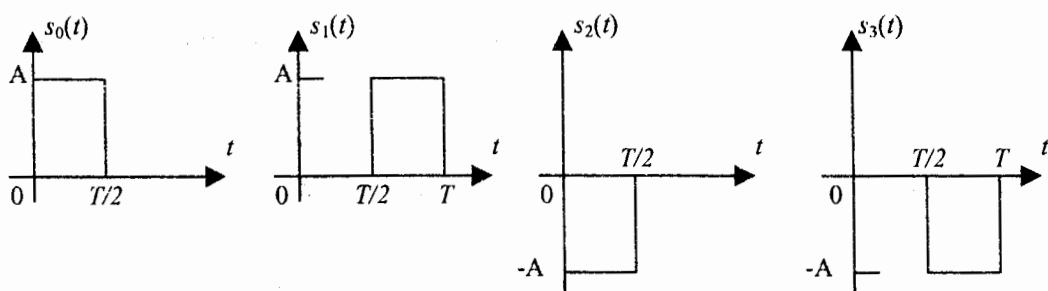
#### 5.4.2 CÁC TÍN HIỆU TRỰC GIAO ĐÔI (BIORTHOGONAL)

Như chúng ta đã xem xét trong mục trước, một tập gồm  $M = 2^k$  dạng sóng trực

giao cùng năng lượng có thể xây dựng được bằng cách chia nhỏ khoảng thời gian symbol  $T$  thành  $M$  khoảng con bằng nhau có độ dài  $T/M$  và gán một xung chữ nhật cho mỗi một khoảng con đó. Một biện pháp tương tự có thể áp dụng được để xây dựng một tập khác gồm  $M = 2^k$  tín hiệu nhiều chiều có tính chất trực giao đôi. Trong một tập tín hiệu như thế, một nửa số các dạng sóng là trực giao còn nửa kia là các dạng sóng đó song với dấu ngược lại; nghĩa là,  $s_0(t), s_1(t), \dots, s_{M/2-1}(t)$  là các dạng sóng trực giao.  $M/2$  dạng sóng còn lại đơn giản là  $s_{i+M/2}(t) = -s_i(t)$ , đối với  $i=0, 1, 2, \dots, (M/2)-1$ . Như thế, chúng ta có được  $M$  tín hiệu mà mỗi một trong chúng có  $M/2$  chiều.

$M/2$  dạng sóng trực giao có thể xây dựng được một cách dễ dàng bằng cách chia nhỏ khoảng thời gian symbol  $T = kT_b$  thành  $M/2$  khoảng con không chồng lấn lên nhau, mỗi khoảng con có độ dài  $2T/M$ , và gán cho mỗi một khoảng con một xung hình chữ nhật. Hình 5.32 minh họa một tập gồm  $M = 4$  dạng sóng trực giao đôi được xây dựng theo cách này. Biểu diễn hình học của một tập  $M$  tín hiệu được xây dựng theo cách này được cho bởi các điểm tín hiệu có  $M/2$  chiều như sau:

$$\begin{aligned} s_0 &= (\sqrt{E}, 0, 0, \dots, 0) \\ s_1 &= (0, \sqrt{E}, 0, \dots, 0) \\ &\dots \\ s_{M/2-1} &= (0, 0, \dots, 0, \sqrt{E}) \\ s_{M/2} &= (-\sqrt{E}, 0, 0, \dots, 0) \\ &\dots \\ s_{M-1} &= (0, 0, \dots, 0, -\sqrt{E}) \end{aligned} \quad (5.4.18)$$



Hình 5.32 Một tập gồm 4 dạng sóng tín hiệu trực giao đôi.

Giống như trong trường hợp các tín hiệu trực giao, ta hãy giả sử rằng các dạng sóng tín hiệu trực giao đôi được sử dụng để truyền thông tin qua một kênh AWGN. Khi đó dạng sóng thu được có thể biểu diễn được theo

$$r(t) = s_i(t) + n(t), \quad 0 \leq t \leq T \quad (5.4.19)$$

trong đó  $s_i(t)$  là dạng sóng được truyền đi còn  $n(t)$  là một hàm mẫu của một quá trình tạp âm trắng có phô công suất  $N_0/2$  [W/Hz].

### **Máy thu tối ưu**

Máy thu tối ưu có thể thực hiện được bằng cách tính tương quan chéo tín hiệu thu được  $r(t)$  với mỗi một trong  $M/2$  dạng sóng trực giao, lấy mẫu các lối ra của các bộ tương quan tín hiệu tại  $t = T$ , rồi cho  $M/2$  lối ra đã lấy mẫu của các bộ tương quan đi vào bộ tách tín hiệu. Như thế, chúng ta có

$$r_i = \int_0^T r(t)s_i(t)dt, \quad i = 0, 1, \dots, \frac{M}{2} - 1 \quad (5.4.20)$$

Giả sử rằng dạng sóng đã được truyền đi là  $s_0(t)$ . Khi đó

$$\begin{aligned} r_i &= \int_0^T r(t)s_i(t)dt, \quad i = 0, 1, \dots, \frac{M}{2} - 1 \\ &= \begin{cases} E + n_0, & i = 0 \\ n_i, & i \neq 0 \end{cases} \end{aligned} \quad (5.4.21)$$

trong đó

$$n_i = \int_0^T n(t)s_i(t)dt, \quad i = 0, 1, \dots, \frac{M}{2} - 1 \quad (5.4.22)$$

còn  $E$  là năng lượng symbol đối với từng dạng sóng tín hiệu. Các thành phần tạp âm là Gauss kỳ vọng 0 và có một phương sai  $\sigma^2 = EN_0/2$ .

### **Bộ tách tín hiệu**

Bộ tách tín hiệu quan sát  $M/2$  lối ra bộ tương quan  $\{r_i, 0 \leq i \leq (M/2)-1\}$  và chọn ra lối ra bộ tương quan nào có  $|r_i|$  lớn nhất. Giả sử rằng

$$|r_j| = \max \{|r_i|\} \quad (5.4.23)$$

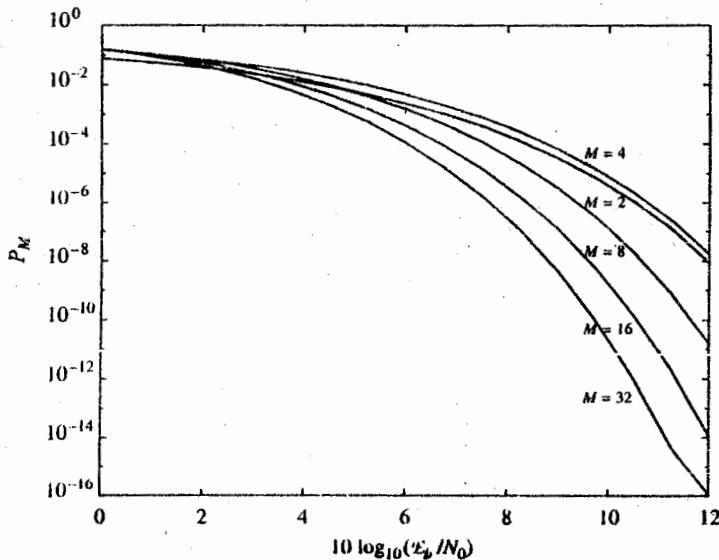
Khi đó bộ tách tín hiệu chọn ra tín hiệu  $s_i(t)$  nếu  $r_i > 0$  và chọn ra  $-s_i(t)$  nếu  $r_i < 0$ .

Để xác định xác suất lỗi, giả sử rằng  $s_0(t)$  đã được truyền đi. Khi đó xác suất của một quyết định đúng bằng với xác suất để  $r_0 = E + n_0 > 0$  và  $|r_0| > |r_i|$  đối với  $i = 1, 2, \dots, (M/2)-1$ . Như vậy,

$$P_C = \int_0^{\infty} \left[ \frac{1}{\sqrt{2\pi}} \int_{-r_0/\sqrt{EN_0/2}}^{r_0/\sqrt{EN_0/2}} e^{-x^2/2} dx \right]^{M-1} p(r_0) dr_0 \quad (5.4.24)$$

trong đó

$$p(r_0) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(r_0-E)^2/2\sigma^2} \quad (5.4.25)$$



Hình 5.33 Xác suất lỗi symbol đối với các tín hiệu trực giao đối.

Cuối cùng, xác suất của một lỗi symbol thì đơn giản là

$$P_M = 1 - P_C \quad (5.4.26)$$

$P_C$  và  $P_M$  có thể đánh giá được bằng số đối với các giá trị  $M$  khác nhau từ (5.4.24) và (5.4.25). Đồ thị trên hình 5.33 minh họa  $P_M$  như một hàm của tỷ số tín trên tạp  $E_b/N_0$ , trong đó  $E=kE_b$ , đối với  $M=2, 4, 8, 16$ , và  $32$ . Chúng ta thấy rằng đồ thị này tương tự với đồ thị tương ứng đối với các tín hiệu trực giao. Tuy nhiên,

đối với các tín hiệu trực giao đôi, ta hãy chú ý rằng  $P_4 > P_2$ . Điều này là do chúng ta đã vẽ trên đồ thị hình 5.33 xác suất lỗi symbol  $P_M$ . Nếu chúng ta vẽ đồ thị xác suất lỗi bít tương đương, ta sẽ thấy rằng các đồ thị đối với  $M = 2$  và  $M = 4$  thì trùng nhau.

MATLAB script để tính xác suất lỗi trong (5.4.24) và (5.4.26) được cho dưới đây.

### m-file

```
% MATLAB script that generates the probability of error versus the signal to noise ratio.  
initial_snr=0;  
final_snr=12;  
snr_step=0.75;  
tolerance=eps; % Tolerance used for the integration  
plus_inf=20; % This is practically infinity  
snr_in_dB=initial_snr:snr_step:final_snr;  
for i=1:length(snr_in_dB),  
    snr=10^(snr_in_dB(i)/10);  
    Pe_2(i)=1-quad8('bdt_int2',0,plus_inf,tolerance,[],snr,2);  
    Pe_4(i)=1-quad8('bdt_int2',0,plus_inf,tolerance,[],snr,4);  
    Pe_8(i)=1-quad8('bdt_int2',0,plus_inf,tolerance,[],snr,8);  
    Pe_16(i)=1-quad8('bdt_int2',0,plus_inf,tolerance,[],snr,16);  
    Pe_32(i)=1-quad8('bdt_int2',0,plus_inf,tolerance,[],snr,32);  
    echo off;  
end;  
% Plotting commands follow
```

## BÀI TẬP MINH HỌA

**Bài tập minh họa 5.11 [Mô phỏng truyền tin bằng tín hiệu trực giao đối]** Hãy thực hiện một mô phỏng Monte-Carlo đối với một hệ thống truyền tin số áp dụng  $M=4$  tín hiệu trực giao đối. Mô hình của hệ thống để mô phỏng được minh họa trên hình 5.34.

### Lời giải

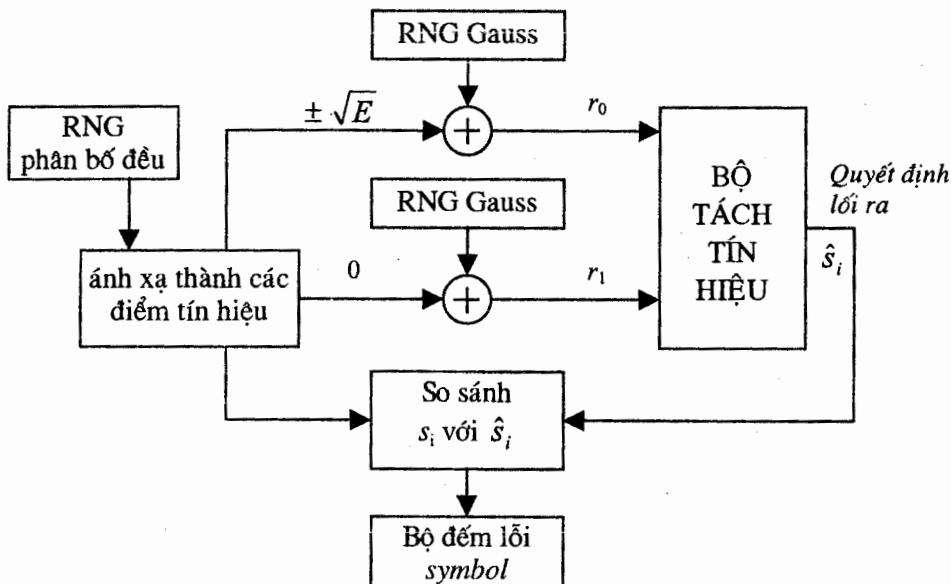
Giống như đã được chỉ dẫn, chúng ta mô phỏng việc tạo các biến ngẫu nhiên  $r_0$  và  $r_1$  là các lỗi vào bộ tách tín hiệu. Chúng ta bắt đầu bằng việc tạo ra một chuỗi nhị phân các con số 0 và 1 mà chúng xuất hiện với xác suất như nhau và độc lập thống kê với nhau, giống như trong bài tập minh họa 5.4. Chuỗi nhị phân này được nhóm thành các cặp bít mà sau đó chúng được ánh xạ thành các thành phần tín hiệu

tương ứng như sau:

$$00 \rightarrow s_0 = (\sqrt{E}, 0); \quad 01 \rightarrow s_1 = (0, \sqrt{E})$$

$$10 \rightarrow s_2 = (0, -\sqrt{E}); \quad 11 \rightarrow s_3 = (-\sqrt{E}, 0)$$

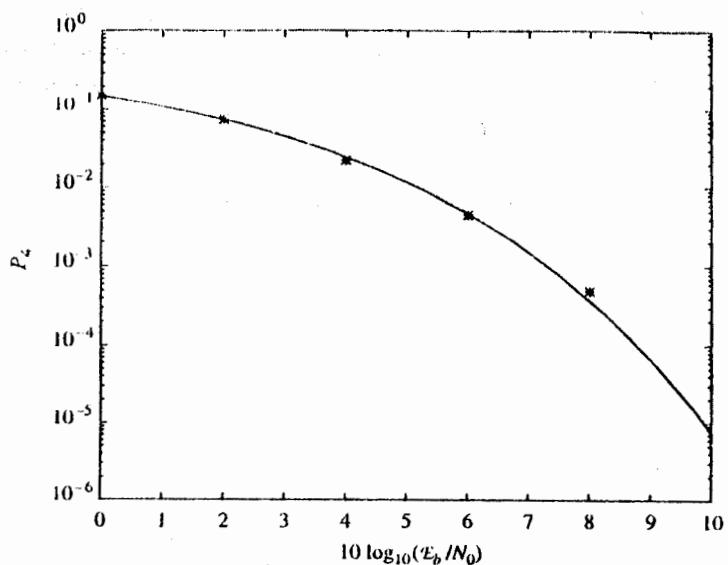
Một cách khác, chúng ta có thể sử dụng phương pháp trong bài tập minh họa 5.8 để trực tiếp tạo ra các *symbol* 2-bít.



Hình 5.34 Sơ đồ khối hệ thống với  $M=4$  tín hiệu trực giao đôi để mô phỏng Monte-Carlo.

Do  $s_2 = -s_1$  và  $s_3 = -s_0$ , việc giải điều chế chỉ cần hai bộ tách quan tín hiệu hay 2 mạch lọc phối hợp, với các đầu ra của chúng là  $r_0$  và  $r_1$ . Các thành phần tạp âm cộng  $n_0$  và  $n_1$  được tạo ra bằng hai bộ tạo tạp âm Gauss, mỗi trong chúng có kỳ vọng 0 và phương sai  $\sigma^2 = EN_0/2$ . Để thuận tiện, chúng ta có thể chuẩn hóa năng lượng *symbol* thành  $E=1$  và cho  $\sigma^2$  biến thiên. Do  $E=2E_b$  nên  $E_b=1/2$ . Lối ra bộ tách tín hiệu được so sánh với chuỗi bít đã được truyền đi và một bộ đếm lõi được sử dụng để đếm số lõi *symbol* và số lõi bít.

Hình 5.35 minh họa các kết quả của mô phỏng này đối với việc truyền dẫn 20 000 bít tại các giá trị khác nhau của SNR  $E_b/N_0$ . Hãy chú ý đến sự thống nhất giữa các kết quả mô phỏng và giá trị lý thuyết của  $P_4$  được cho bởi (5.4.26) và (5.4.24).



Hình 5.35 Xác suất lỗi symbol đối với  $M=4$  tín hiệu trực giao đôi nhận được từ mô phỏng Monte-Carlo, so sánh với xác suất lỗi lý thuyết.

Các MATLAB *script* cho bài tập này được cho dưới đây.

### m-file

% MATLAB script for Illustrative Problem 11, Chapter 5.

echo on

SNRindB=0:2:10;

for i=1:length(SNRindB),

  % simulated error rate

  smld\_err\_prb(i)=smldP511(SNRindB(i));

  echo off;

end;

echo on ;

  % Plotting commands follow

### m-file

function [p]=smldP511(snr\_in\_dB)

  % [p]=smldP511(snr\_in\_dB)

  %     SMLDP511 simulates the probability of error for the given

  %     snr\_in\_dB, signal to noise ratio in dB, for the system

  %     described in illustrated problem 11, Chapter 5.

M=4; % quarternary orthogonal signalling

E=1;

```
SNR=exp(snr_in_dB*log(10)/10); % signal to noise ratio per bit
sgma=sqrt(E^2/(4*SNR)); % sigma, standard deviation of noise
N=10000; % number of symbols being simulated
% generation of the quarternary data source
for i=1:N,
    temp=rand; % uniform random variable over (0,1)
    if (temp<0.25),
        dsource(i)=0;
    elseif (temp<0.5),
        dsource(i)=1;
    elseif (temp<0.75),
        dsource(i)=2;
    else
        dsource(i)=3;
    end
end;
% detection, and error probability computation
numoferr=0;
for i=1:N,
    % The matched filter outputs
    if (dsource(i)==0)
        r0=sqrt(E)+gngauss(sgma);
        r1=gngauss(sgma);
    elseif (dsource(i)==1)
        r0=gngauss(sgma);
        r1=sqrt(E)+gngauss(sgma);
    elseif (dsource(i)==2)
        r0=-sqrt(E)+gngauss(sgma);
        r1=gngauss(sgma);
    else
        r0=gngauss(sgma);
        r1=-sqrt(E)+gngauss(sgma);
    end;
    % detector follows
    if (r0>abs(r1)),
        decis=0;
    elseif (r1>abs(r0)),
        decis=1;
    elseif (r0<-abs(r1)),
        decis=2;
    else
        decis=3;
    end;
```

```

if (decis~=dsource(i)),           % if it is an error, increase to the error counter
    numoferr=numoferr+1;
end;
end;
p=numoferr/N;                   % bit error probability estimate

```

## CÁC BÀI TẬP

**5.1** Giả sử rằng hai tín hiệu trực giao cho trên hình 5.2 được sử dụng để truyền thông tin nhị phân qua một kênh AWGN. Tín hiệu thu được trong từng khoảng thời gian tồn tại của một bít  $T_b$  được cho bởi (5.2.1). Giả sử rằng dạng sóng tín hiệu thu được được lấy mẫu với tốc độ  $10/T_b$ , nghĩa là với 10 mẫu trên một khoảng thời gian một bít. Do đó, trong thời gian rời rạc, dạng sóng tín hiệu  $s_0(t)$  với biên độ  $A$  được biểu diễn bằng 10 mẫu ( $A, A, \dots, A$ ) và dạng sóng tín hiệu  $s_1(t)$  được biểu diễn bằng 10 mẫu ( $A, A, A, A, A, -A, -A, -A, -A, -A$ ). Hệ quả là, phiên bản đã lấy mẫu của chuỗi nhận được khi  $s_0(t)$  được truyền là

$$r_k = A + n_k, \quad k=1, 2, \dots, 10$$

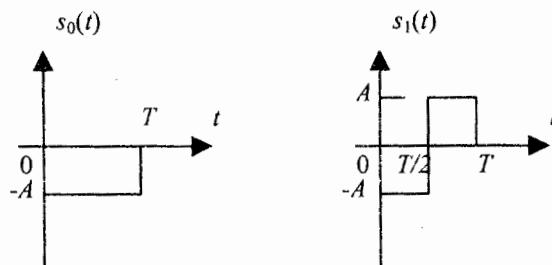
và khi  $s_1(t)$  được truyền đi là

$$r_k = \begin{cases} A + n_k, & 1 \leq k \leq 5 \\ -A + n_k, & 6 \leq k \leq 10 \end{cases}$$

trong đó chuỗi  $\{n_k\}$  là chuỗi các biến Gauss, kỳ vọng 0 cùng phân bố và độc lập thống kê, với mỗi một biến ngẫu nhiên đều có phương sai  $\sigma^2$ . Hãy viết một thuật toán MATLAB tạo ra chuỗi  $\{r_k\}$  cho mỗi một trong hai tín hiệu thu được có thể có, và hãy thực hiện việc tính tương quan thời gian rời rạc chuỗi  $\{r_k\}$  với mỗi một trong hai tín hiệu có thể có  $s_0(t)$  và  $s_1(t)$  (được biểu diễn bằng các phiên bản đã lấy mẫu của chúng) đối với các giá trị khác nhau của phương sai tần âm cộng Gauss  $\sigma^2=0, \sigma^2=0.1, \sigma^2=1.0$  và  $\sigma^2=2.0$ . Biên độ tín hiệu có thể được chuẩn hóa thành  $A=1$ . Hãy vẽ đồ thị tín hiệu các lối ra bộ tương quan tại các thời điểm  $k=1, 2, 3, \dots, 10$ .

**5.2** Lặp lại bài tập 5.1 đối với hai tín hiệu  $s_0(t)$  và  $s_1(t)$  được minh họa trên hình P5.2.

Hãy mô tả những cái tương tự và những điều khác biệt giữa tập hai tín hiệu này và tập tín hiệu trên hình 5.2. Có một tập nào tốt hơn tập kia không theo quan điểm truyền dẫn một chuỗi các tín hiệu mang tín nhị phân?



Hình P5.2.

**5.3** Trong bài tập này, mục tiêu là thay hai bộ lọc phối hợp vào vị trí của hai bộ tương quan tín hiệu trong bài tập 5.1. Điều kiện đối với việc tạo các tín hiệu thì hoàn toàn như bài tập 5.1.

Hãy viết một thủ tục MATLAB tạo ra chuỗi  $\{r_k\}$  đối với mỗi một trong hai tín hiệu thu được có thể có và thực hiện việc lọc phối hợp thời gian rời rạc của chuỗi  $\{r_k\}$  với mỗi một trong hai tín hiệu có thể có  $s_0(t)$  và  $s_1(t)$  (được biểu diễn bằng các phiên bản đã lấy mẫu của chúng) đối với các giá trị khác nhau của phương sai tạp âm cộng Gauss  $\sigma^2=0$ ,  $\sigma^2=0.1$ ,  $\sigma^2=1.0$  và  $\sigma^2=2.0$ . Biên độ tín hiệu có thể được chuẩn hoá thành  $A=1$ . Hãy vẽ đồ thị các tín hiệu lối ra các bộ lọc phối hợp tại các thời điểm tương ứng với  $k=1, 2, 3, \dots, 10$ .

**5.4** Hãy lặp lại bài tập 5.3 đối với các dạng sóng tín hiệu cho trên hình P5.2.

**5.5** Hãy chạy chương trình MATLAB thực hiện một mô phỏng Monte-Carlo đối với hệ thống truyền tin nhị phân như đã trình bày ở hình 5.8, dựa trên các tín hiệu trực giao. Hãy thực hiện việc mô phỏng với 10 000 bít và hãy đánh giá xác suất lỗi đối với  $\sigma^2=0$ ,  $\sigma^2=0.1$ ,  $\sigma^2=1.0$  và  $\sigma^2=2.0$ . Hãy vẽ tỷ lệ lỗi lý thuyết và tỷ lệ lỗi xác định được bằng mô phỏng Monte-Carlo, hãy so sánh hai kết quả đó. Hãy vẽ cả đồ thị 1000 mẫu tín hiệu thu được cộng với tạp âm, tại lối vào của bộ tách tín hiệu đối với từng giá trị nói trên của  $\sigma^2$ .

**5.6** Hãy lặp lại bài tập 5.5 đối với hệ thống truyền tin nhị phân được trình bày trên hình 5.13, dựa trên các tín hiệu đối cực.

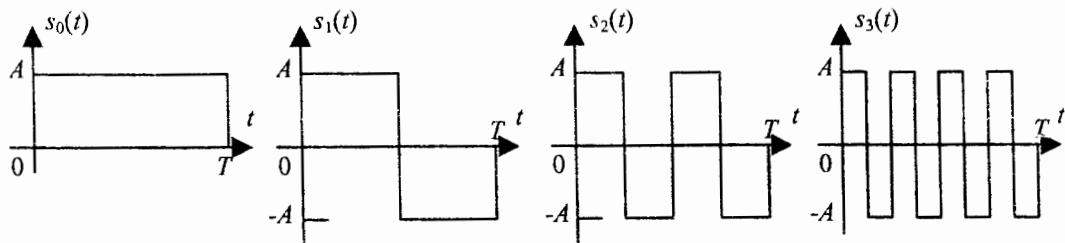
**5.7** Hãy lặp lại bài tập 5.5 đối với hệ thống truyền tin nhị phân sử dụng các tín hiệu đóng-ngắt.

**5.8** Hãy chạy chương trình MATLAB thực hiện một mô phỏng Monte-Carlo đối với một hệ thống truyền tin PAM bốn mức. Hãy thực hiện mô phỏng đó với 10 000 *symbol* (20 000 bít), và hãy đánh giá xác suất lỗi *symbol* đối với  $\sigma^2=0$ ,  $\sigma^2=0.1$ ,  $\sigma^2=1.0$  và  $\sigma^2=2.0$ . Hãy vẽ các đồ thị tỷ lệ lỗi theo lý thuyết và các tỷ lệ lỗi xác định được bằng mô phỏng rồi so sánh các kết quả đó. Hãy vẽ cả đồ thị 1000 mẫu tín hiệu thu được cộng với tạp âm, tại lỗi vào của bộ tách tín hiệu đối với từng giá trị nói trên của  $\sigma^2$ .

**5.9** Hãy sửa đổi chương trình MATLAB trong bài tập 5.8 để mô phỏng  $M=8$  tín hiệu PAM và thực hiện các mô phỏng đã được ấn định trong bài tập 5.8.

**5.10** Hãy chạy chương trình MATLAB thực hiện mô phỏng Monte-Carlo một hệ thống truyền tin số áp dụng  $M=4$  tín hiệu trực giao, như đã được mô tả trong bài tập minh họa 5.10. Hãy thực hiện mô phỏng đối với 10 000 *symbol* (20 000 bít) và đánh giá xác suất lỗi bít đối với  $\sigma^2=0.1$ ,  $\sigma^2=1.0$  và  $\sigma^2=2.0$ . Hãy vẽ các đồ thị xác suất lỗi theo lý thuyết và các tỷ lệ lỗi xác định được bằng mô phỏng Monte-Carlo rồi so sánh các kết quả đó.

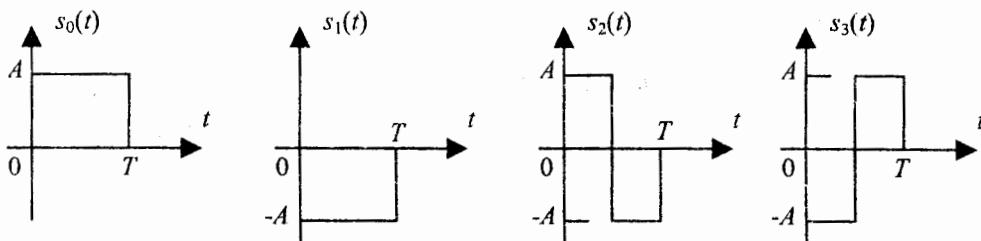
**5.11** Hãy xét 4 dạng sóng tín hiệu được cho thấy trên hình P5.11. Hãy chứng tỏ rằng 4 dạng sóng tín hiệu này trực giao lẫn nhau. Các kết quả của mô phỏng Monte-Carlo trong bài tập 5.10 sẽ áp dụng được không cho các tín hiệu này? Tại sao?



Hình P5.11

**5.12** Hãy chạy chương trình MATLAB thực hiện mô phỏng Monte-Carlo một hệ thống truyền tin số áp dụng  $M=4$  tín hiệu trực giao đôi, như đã được mô tả trong bài tập minh họa 5.11. Hãy thực hiện các mô phỏng đối với 10 000 *symbol* (20 000 bít) và đánh giá xác suất lỗi *symbol* đối với  $\sigma^2=0.1$ ,  $\sigma^2=1.0$  và  $\sigma^2=2.0$ . Hãy vẽ các đồ thị xác suất lỗi *symbol* lý thuyết và tỷ lệ lỗi xác định được bằng mô phỏng Monte-Carlo rồi so sánh các kết quả đó. Hãy vẽ đồ thị cả 1000 mẫu tín hiệu thu được cộng với tạp âm, tại lỗi vào của bộ tách tín hiệu đối với từng giá trị nói trên của  $\sigma^2$ .

**5.13** Hãy xét 4 dạng sóng tín hiệu được cho thấy trên hình P5.13. Hãy chứng tỏ rằng chúng là trực giao đôi. Các kết quả của mô phỏng Monte-Carlo trong bài tập 5.12 sẽ áp dụng được không cho tập 4 dạng sóng tín hiệu này? Tại sao?



Hình P5.13

## CHƯƠNG 6

# TRUYỀN DẪN SỐ QUA KÊNH CÓ BĂNG THÔNG HỮU HẠN

### 6.1 GIỚI THIỆU CHUNG

Trong chương này, chúng ta đề cập đến một số khía cạnh của truyền dẫn số qua kênh có độ rộng băng thông hữu hạn. Ta sẽ bắt đầu bằng việc mô tả các đặc trưng phổ của các tín hiệu PAM. Thứ đến, chúng ta sẽ xem xét đặc trưng của các kênh có băng thông hữu hạn và vấn đề thiết kế các dạng sóng tín hiệu đối với các kênh như thế. Sau đó chúng ta sẽ đề cập tới vấn đề thiết kế các bộ san bằng để bù méo gây bởi các kênh có độ rộng băng thông hạn chế. Chúng ta sẽ thấy rằng méo kênh gây ra xuyên nhiễu giữa các ký hiệu (ISI: *InterSymbol Interference*), mà nó gây nên các lỗi trong giải điều chế tín hiệu. Một bộ san bằng kênh là một thiết bị làm giảm xuyên nhiễu giữa các ký hiệu và do đó làm giảm tỷ lệ lỗi trong chuỗi số liệu sau giải điều chế.

### 6.2 PHỔ CÔNG SUẤT CỦA MỘT TÍN HIỆU SỐ PAM

Trong chương trước, chúng ta đã xét việc truyền dẫn thông tin số bằng điều chế biên độ xung (PAM). Trong mục này, chúng ta sẽ nghiên cứu các đặc trưng phổ của các tín hiệu như thế.

Một tín hiệu số PAM tại đầu vào một kênh thông tin được biểu diễn một cách tổng quát theo:

$$v(t) = \sum_{n=-\infty}^{\infty} a_n g(t - nT) \quad (6.2.1)$$

với  $\{a_n\}$  là chuỗi các biên độ tương ứng với các *symbol* thông tin đến từ nguồn tin,  $g(t)$  là một dạng sóng còn  $T$  là nghịch đảo của tốc độ *symbol*.  $T$  cũng còn được gọi là khoảng thời gian của một *symbol*. Mỗi một phần tử của chuỗi  $\{a_n\}$  được chọn từ một trong các giá trị biên độ có thể có là

$$A_m = (2m - M + 1)d, \quad m = 0, 1, \dots, M-1 \quad (6.2.2)$$

trong đó  $d$  là hệ số tỷ lệ mà nó xác định khoảng cách Euclide giữa một cặp bất kỳ nào đó của các biên độ tín hiệu. ( $2d$  là khoảng cách Euclide giữa hai mức biên độ tín hiệu cạnh nhau).

Vì chuỗi thông tin là một chuỗi ngẫu nhiên nên chuỗi biên độ  $\{a_n\}$  tương ứng với các *symbol* thông tin đến từ nguồn cũng ngẫu nhiên. Do đó, tín hiệu PAM  $v(t)$  là một hàm mẫu của một quá trình ngẫu nhiên  $V(t)$ . Để xác định các đặc tính phổ của quá trình ngẫu nhiên  $V(t)$ , chúng ta phải xác định phổ công suất.

Trước hết, chúng ta hãy để ý rằng giá trị trung bình của  $V(t)$  là

$$E[V(t)] = \sum_{n=-\infty}^{\infty} E(a_n)g(t - nT) \quad (6.2.3)$$

Bằng cách chọn các biên độ tín hiệu để chúng đối xứng quanh 0 như đã cho trong (6.2.2) và đồng khả năng, chúng ta có  $E(a_n) = 0$ , và vì vậy  $E[V(t)] = 0$ .

Hàm tự tương quan của  $V(t)$  là

$$R_v(t+\tau; t) = E[V(t)V(t+\tau)] \quad (6.2.4)$$

Người ta chỉ ra trong nhiều tài liệu chuẩn mực về thông tin số rằng, hàm tự tương quan của này là một hàm tuần hoàn theo biến  $t$  với chu kì  $T$ . Các quá trình ngẫu nhiên có giá trị trung bình và hàm tự tương quan tuần hoàn được gọi là quá trình *dừng tuần hoàn* (*periodically stationary*) hoặc *dừng vòng* (*cyclostationary*). Biến thời gian  $t$  có thể loại bỏ được bằng cách lấy trung bình  $R_v(t+\tau; t)$  trên một chu kì, nghĩa là

$$\bar{R}_v(\tau) = \frac{1}{T} \int_{-T/2}^{T/2} R_v(t+\tau; t) dt \quad (6.2.5)$$

Hàm tự tương quan trung bình đối với các tín hiệu PAM có thể biểu diễn theo:

$$\bar{R}_v(\tau) = \frac{1}{T} \sum_{m=-\infty}^{\infty} R_a(m) R_g(\tau - mT) \quad (6.2.6)$$

trong đó  $R_a(m) = E(a_n a_{n+m})$  là hàm tự tương quan của dãy  $\{a_n\}$  còn  $R_g(\tau)$  được xác định theo

$$R_g(\tau) = \int_{-\infty}^{\infty} g(t)g(t+\tau) dt \quad (6.2.7)$$

Phổ công suất của  $V(t)$  đơn giản là biến đổi Fourier của hàm tự tương quan trung bình  $\bar{R}_v(\tau)$ , nghĩa là

$$S_v(f) = \int_{-\infty}^{\infty} \bar{R}_v(\tau) e^{-j2\pi f\tau} d\tau = \frac{1}{T} S_a(f) |G(f)|^2 \quad (6.2.8)$$

với  $S_a(f)$  là phổ công suất của dãy biên độ  $\{a_n\}$  còn  $G(f)$  là biến đổi Fourier của xung  $g(t)$ ,  $S_a(f)$  được xác định theo

$$S_a(f) = \sum_{m=-\infty}^{\infty} R_a(m) e^{-j2\pi fmT} \quad (6.2.9)$$

Từ (6.2.8), chúng ta thấy rằng phổ công suất của tín hiệu PAM là một hàm của phổ công suất của dãy *symbol* thông tin  $\{a_n\}$  và phổ công suất của xung  $g(t)$ . Trong trường hợp đặc biệt trong đó dãy  $\{a_n\}$  không tương quan, tức là

$$R_a(m) = \begin{cases} \sigma_a^2, & m=0 \\ 0, & m \neq 0 \end{cases} \quad (6.2.10)$$

trong đó  $\sigma_a^2 = E(a_n^2)$ , thì  $S_a(f) = \sigma_a^2$  trên mọi tần số và

$$S_v(f) = \frac{\sigma_a^2}{T} |G(f)|^2 \quad (6.2.11)$$

Trong trường hợp này, phổ công suất của  $V(t)$  phụ thuộc hoàn toàn vào các đặc tính phổ của xung  $g(t)$ .

### BÀI TẬP MINH HỌA

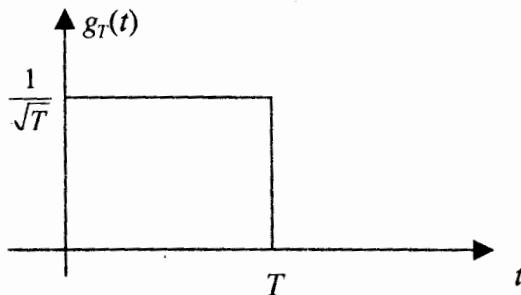
**Bài tập minh họa 6.1 [Phổ công suất tín hiệu PAM]** Xác định phổ công suất của  $V(t)$  khi  $\{a_n\}$  là một dãy không tương quan và  $g(t)$  là xung chữ nhật được cho trên hình 6.1.

**Lời giải**

Biến đổi Fourier của  $g(t)$  là

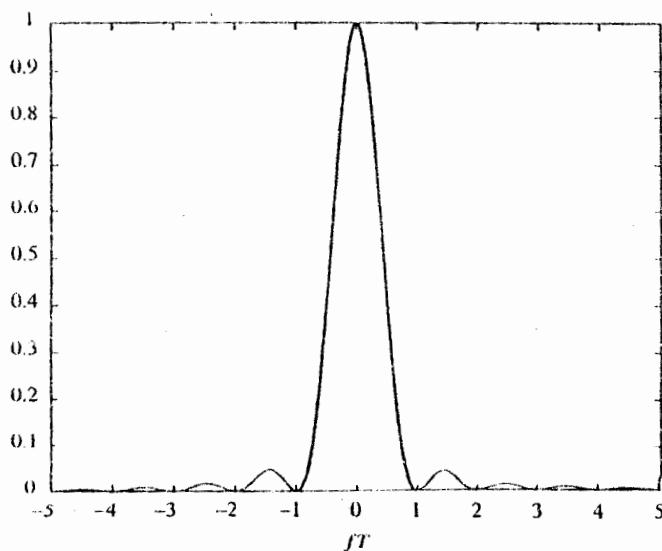
$$G(f) = \int_{-\infty}^{\infty} g(t) e^{-j2\pi ft} dt = \sqrt{T} \frac{\sin \pi f T}{\pi f T} e^{-j\pi f T} \quad (6.2.12)$$

và:  $S_v(f) = \sigma_a^2 \left( \frac{\sin \pi f T}{\pi f T} \right)^2 \quad (6.2.13)$



Hình 6.1. Xung của máy phát.

Phổ công suất này được minh họa trên hình 6.2.



Hình 6.2. Phổ công suất tín hiệu phát trong ví dụ 6.1 (trường hợp  $\sigma_a^2 = 1$ )

Dưới đây là MATLAB script cho tính toán này.

**m-file**

```
% MATLAB script for Illustrative Problem 1, Chapter 6.
echo on
```

```

T=1;
delta_f=1/(100*T);
f=-5/T:delta_f:5/T;
sgma_a=1;
Sv=sgma_a^2*sinc(f*T).^2;
% plotting command follows
plot(f,Sv);

```

## BÀI TẬP MINH HOA

**Bài tập minh họa 6.2 [Phổ công suất tín hiệu PAM]** Giả sử hàm tự tương quan của dãy  $\{a_n\}$  là

$$R_a(m) = \begin{cases} 1, & m = 0 \\ \frac{1}{2}, & m = \pm 1 \\ 0, & \text{với các giá trị khác của } m \end{cases} \quad (6.2.14)$$

và  $g(t)$  là một xung chữ nhật được thể hiện trên hình 6.1. Hãy tính  $S_v(f)$  trong trường hợp này.

### Lời giải

Phổ công suất của dãy biên độ  $\{a_n\}$  được tính theo công thức (6.2.9):

$$S_a(f) = 1 + \cos 2\pi fT = 2\cos^2 \pi fT \quad (6.2.15)$$

Hệ quả là

$$S_v(f) = 2\cos^2 \pi fT \left( \frac{\sin \pi fT}{\pi fT} \right)^2 \quad (6.2.16)$$

Đồ thị phổ công suất này được cho trên hình 6.3.

Dưới đây là chương trình MATLAB để thực hiện tính toán này. Trong trường hợp này, phổ công suất toàn phần của tín hiệu phát  $V(t)$  hẹp hơn một cách đáng kể so với phổ trên hình 6.3.

### m-file

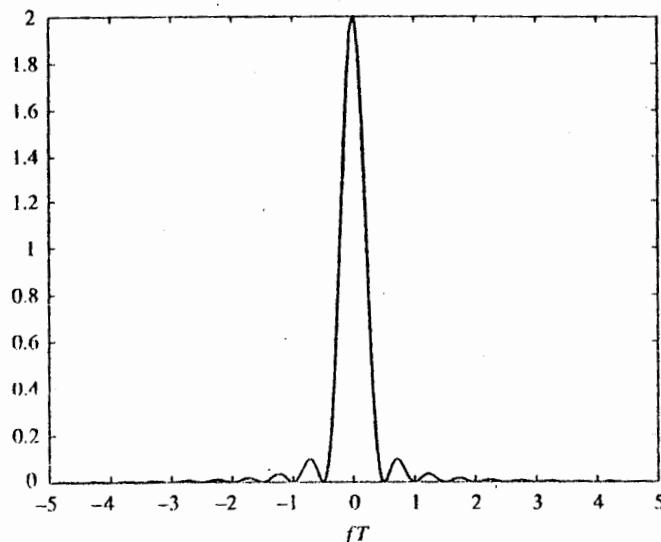
```

% MATLAB script for Illustrative Problem 2, Chapter 6.
echo on
T=1;
delta_f=1/(100*T);

```

```

f=-5/T:delta_f:5/T;
Sv=2*(cos(pi*f*T).*sinc(f*T)).^2;
% plotting command follows
plot(f,Sv);
    
```



Hình 6.3. Phổ công suất tín hiệu phát trong ví dụ 6.2 (trường hợp  $\sigma_a^2 = 1$ ).

### 6.3 ĐẶC TRƯNG CỦA CÁC KÊNH CÓ ĐỘ RỘNG BĂNG THÔNG HỮU HẠN VÀ MÉO KÊNH

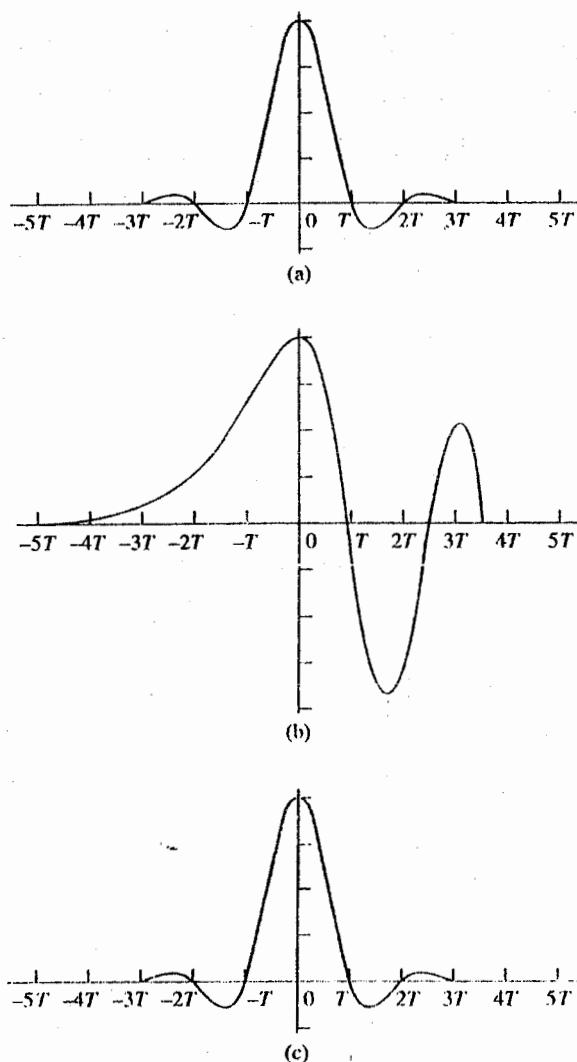
Nhiều kênh thông tin, bao gồm các kênh điện thoại và một số kênh vô tuyến, có thể đặc trưng được một cách tổng quát như các bộ lọc tuyến tính có băng thông hữu hạn. Do đó, các kênh như thế mô tả được bằng đáp ứng tần số của chúng  $C(f)$  mà nó có thể biểu diễn được theo

$$C(f) = A(f) e^{j\theta(f)} \quad (6.3.1)$$

trong đó  $A(f)$  được gọi là *đáp ứng biên độ* còn  $\theta(f)$  được gọi là *đáp ứng pha*. Một đặc trưng khác đôi khi cũng được sử dụng thay thế cho đáp ứng pha là *trễ đường bao* (*envelope delay*) hay *trễ nhóm* (*group delay*) được định nghĩa theo

$$\tau(f) = -\frac{1}{2\pi} \frac{d\theta(f)}{df} \quad (6.3.2)$$

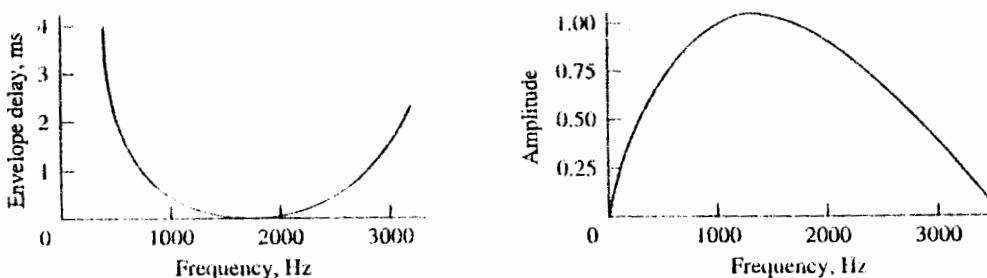
Một kênh được gọi là không méo hoặc lí tưởng nếu  $A(f)$  bằng hằng số còn  $\alpha(f)$  là một hàm tuyến tính của tần số [hoặc trễ đường bao  $\tau(f)$  không đổi] trong độ rộng băng chiếm bởi tín hiệu phát  $W$ . Ngược lại, nếu  $A(f)$  và  $\tau(f)$  không phải là hằng số trong độ rộng băng chiếm bởi tín hiệu phát, kênh sẽ gây ra méo tín hiệu. Nếu  $A(f)$  không phải là hằng số, méo được gọi là *méo biên độ* còn nếu  $\tau(f)$  thay đổi, méo đổi với tín hiệu đã truyền đi được gọi là *méo trễ*.



Hình 6.4. Ảnh hưởng méo kênh: (a) Đầu vào kênh; (b) Đầu ra kênh; (c) Đầu ra bộ cân bằng

Như một hệ quả của méo biên độ và méo trễ gây bởi đặc tính đáp ứng tần số

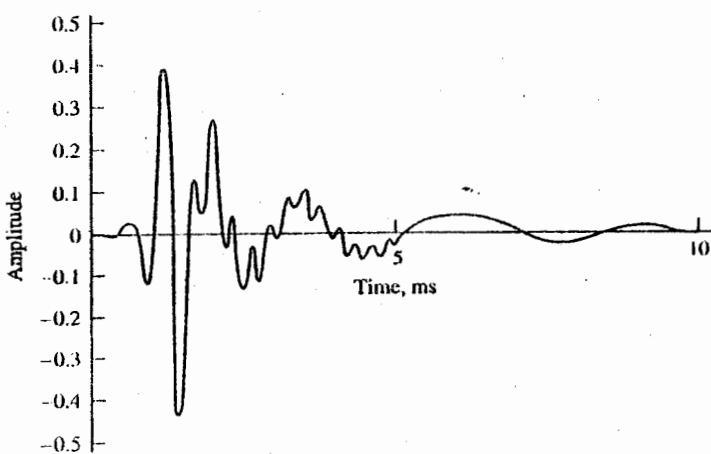
của kênh không lý tưởng  $C(f)$ , một chuỗi các xung được truyền qua kênh ở tốc độ tương ứng với độ rộng băng  $W$  bị nhòe đến nỗi chúng không thể phân biệt được như các xung được xác định đúng tại máy đầu cuối thu. Thay vào đó, chúng lấn lên nhau vì vậy chúng ta có xuyên nhiễu giữa các dấu (ISI: *InterSymbol Interference*). Như một thí dụ về tác động của méo trễ tới một xung được truyền, hình 6.4(a) minh họa một xung có độ rộng băng tần hữu hạn có các trị bằng không tại các điểm được đánh dấu  $\pm T$ ,  $\pm 2T$ , vân vân. Khi thông tin được tải bằng biên độ xung, như trong PAM chẵng hạn, ta có thể truyền một dãy các xung, mỗi một trong chúng đều có giá trị đỉnh tại các điểm bằng không tuân hoà của các xung khác. Tuy nhiên, việc truyền xung qua một kênh được mô hình hoá là có một đặc tính trễ đường bao  $\tau(f)$  tuyến tính [tức là có đặc tính pha  $\theta(f)$  bậc hai] sẽ gây ra xung thu được như được thể hiện trên hình 6.4(b) có các điểm cắt qua giá trị bằng không không còn được phân bố một cách tuân hoà nữa. Hệ quả là, một dãy các xung liên tiếp sẽ nhòe lấn lên nhau và các giá trị đỉnh của các xung sẽ không còn phân biệt được nữa. Như vậy, méo trễ của kênh gây ra xuyên nhiễu giữa các dấu (ISI). Như sẽ được bàn bạc trong chương này, có thể bù được đặc tính đáp ứng tần số không lý tưởng của kênh bằng cách sử dụng một mạch lọc hay mạch san bằng tại bộ giải điều chế. Hình 6.4(c) minh họa lỗi ra của một bộ san bằng tuyến tính mà nó bù khử đối với méo tuyến tính trong kênh.



Hình 6.5 Các đặc tuyến biên độ và trễ trung bình của một kênh điện thoại cự ly trung bình

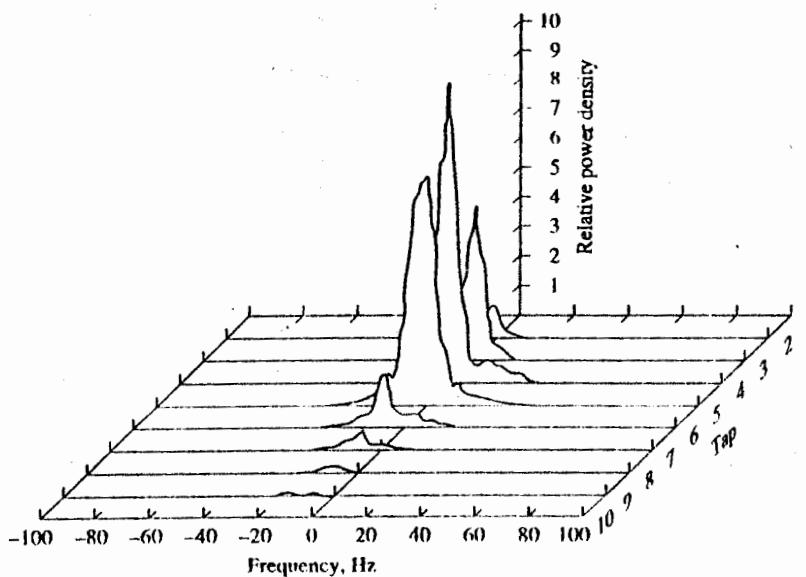
Như một ví dụ, chúng ta hãy xét ISI trên một kênh thoại. Hình 6.5 biểu diễn biên độ và trễ đo được trung bình theo tần số đối với một kênh thoại của mạng viễn

thông có chuyển mạch. Chúng ta thấy rằng băng thông khả dụng của kênh mở ra từ quãng 300 Hz tới quãng 3200 Hz. Đáp ứng xung tương ứng của kênh trung bình này được cho trên hình 6.6. Khoảng thời gian tồn tại của nó vào khoảng 10 ms. Tương ứng, các tốc độ *symbol* truyền trên một kênh như thế có thể vào bậc 2500 xung hay *symbol* trên một giây. Do đó, ISI có thể trải rộng trên 20-30 *symbol*.



Hình 6.6 Đáp ứng xung của kênh trung bình với biên độ và trễ được cho trên hình 6.5

Bên cạnh các kênh điện thoại còn có các kênh vật lý khác cũng biểu hiện một dạng nào đó của phân tán về thời gian (*time dispersive*) và do vậy cũng gây nên ISI. Các kênh vô tuyến, như các kênh sóng ngắn tầng điện ly (HF), các kênh đối lưu và các kênh vô tuyến di động tế bào, là ba thí dụ về các kênh vô tuyến phân tán về thời gian. Trong các kênh này, sự phân tán về thời gian - và do đó là ISI - là kết quả của nhiều đường truyền lan sóng có độ trễ đường truyền khác nhau. Số lượng các đường truyền và độ trễ thời gian tương đối giữa các đường thay đổi theo thời gian, do đó các kênh vô tuyến này thường được gọi là các kênh đa đường biến đổi theo thời gian. Các điều kiện truyền dẫn đa đường biến đổi theo thời gian làm cho các đặc tính đáp ứng tần số thay đổi mạnh. Do đó, đặc trưng đáp ứng tần số được sử dụng cho kênh thoại thì không thích hợp đối với các kênh đa đường biến đổi theo thời gian. Thay vào đó, các kênh vô tuyến này được đặc trưng thống kê bằng hàm tán xạ mà nói một cách ngắn gọn thì nó là một biểu diễn hai chiều của công suất tín hiệu nhận được trung bình như một hàm của trễ thời gian tương đối và độ trãi phổ Doppler.



Hình 6.7. Hàm tán xạ của một kênh tán xạ tầng đối lưu cự ly trung bình

Nhằm các mục đích minh họa, một hàm tán xạ đo được trên một kênh vô tuyến đối lưu cự ly trung bình (150 dặm) được cho thấy trên hình 6.7. Thời gian tồn tại tổng cộng trung bình (trải đa đường) của đáp ứng kênh là xấp xỉ  $0.7 \mu\text{s}$  còn độ trải tần số Doppler giữa hai điểm nửa công suất thì nhỏ hơn 1 Hz một chút đối với đường truyền dài nhất và lớn hơn thế đôi chút đối với các đường truyền lan khác. Một cách tiêu biểu, nếu việc truyền dẫn diễn ra với một tốc độ là  $10^7 \text{ symbol/s}$  trên một kênh như thế thì độ trải đa đường  $0.7 \mu\text{s}$  sẽ gây nên ISI trãi rộng trên khoảng bảy symbol.

### BÀI TẬP MINH HOA

**Bài tập minh họa 6.3 [Kênh như một bộ lọc].** Như đã được chỉ ra ở trên, một kênh truyền tin có băng tần hạn chế có thể mô hình hoá được như một mạch lọc tuyến tính mà các đặc tính đáp ứng tần số của nó phù hợp với các đặc tính đáp ứng tần số của kênh. MATLAB có thể sử dụng được để thiết kế các bộ lọc số có thời gian tồn tại đáp ứng xung hữu hạn (FIR: *Finite-duration Impulse Response*) hoặc vô hạn (IIR: *Infinite-duration Impulse Response*) mà chúng xấp xỉ với các đặc tính đáp ứng tần số của các kênh truyền tin *analog*. Giả sử chúng ta muốn mô

hình hoá một kênh lý tưởng có một đáp ứng biên độ  $A(f)=1$  đối với  $|f| \leq 2000$  Hz và  $A(f)=0$  đối với  $|f| > 2000$  Hz còn độ trễ thì bằng hằng số (pha tuyến tính) đối với mọi  $f$ . Tốc độ lấy mẫu đối với bộ lọc số được chọn là  $F_s=10$  ngàn Hz. Vì đáp ứng pha mong muốn là tuyến tính, chỉ bộ lọc FIR đáp ứng được yêu cầu này. Tuy nhiên, không thể đạt được đáp ứng bằng không tại các tần số trong băng chấn. Thay vào đó chúng ta chọn đáp ứng tại băng chấn bằng  $-40$  dB và tần số bắt đầu băng chấn là  $2500$  Hz.Thêm vào đó, chúng ta cho phép một lượng gợn sóng nhỏ,  $0.5$  dB, trong băng thông. Hãy thiết kế một bộ lọc với các đặc tính này.

### Lời giải

Đáp ứng xung và đáp ứng tần số của một bộ lọc FIR có độ dài  $N=41$  thỏa mãn các chỉ tiêu này được minh họa trên hình 6.8. Do  $N$  lẻ, trễ qua bộ lọc là  $(N+1)/2$  khâu mà nó tương ứng với một trễ thời gian là  $(N+1)/20$  ms với tốc độ lấy mẫu là  $10$  kHz. Trong ví dụ này, bộ lọc FIR đã được thiết kế bằng MATLAB theo phương pháp xấp xỉ Chebyshev (thuật toán Remez).

### m-file

```
% MATLAB script for Illustrative Problem 3, Chapter 6.
echo on
f_cutoff=2000; % the desired cut-off frequency
f_stopband=2500; % the actual stopband frequency
fs=10000; % the sampling frequency
f1=2*f_cutoff/fs; % the normalized passband frequency
f2=2*f_stopband/fs; % the normalized stopband frequency
N=41; % this number is found by experiment
F=[0 f1 f2 1];
M=[1 1 0 0]; % describes the low-pass filter
B=remez(N-1,F,M); % returns the FIR tap coefficients
% plotting command follows
figure(1);
[H,W]=freqz(B);
H_in_dB=20*log10(abs(H));
plot(W/(2*pi),H_in_dB);
figure(2);
plot(W/(2*pi),(180/pi)*unwrap(angle(H)));
```

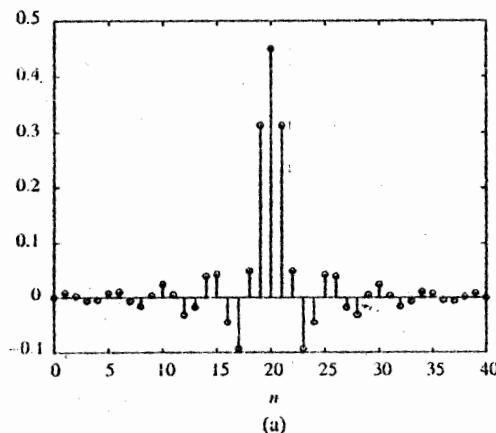
% plot of the impulse response follows

figure(3);

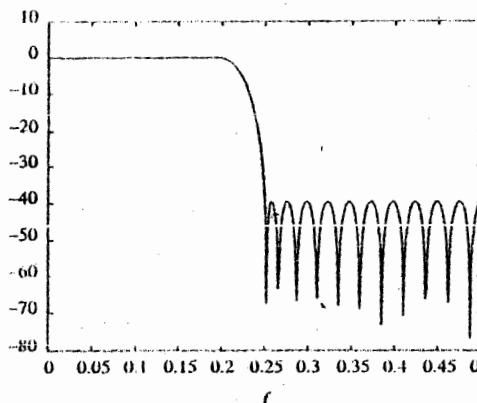
plot(zeros(size([0:N-1])));

hold;

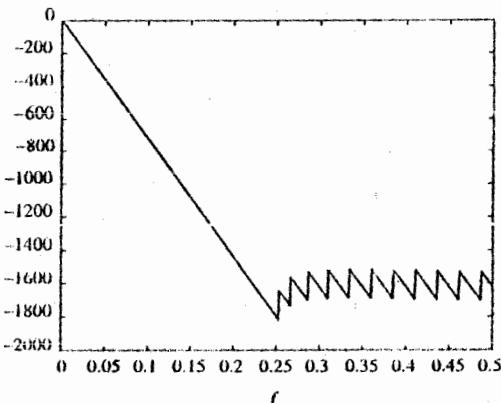
stem([0:N-1],B);



(a)



(b)



(c)

Hình 6.8 (a) Đáp ứng xung; (b, c) đáp ứng tần số của bộ lọc FIR pha tuyến tính trong bài tập minh họa 6.3

## BÀI TẬP MINH HỌA

**Bài tập minh họa 6.4 [Kênh như một bộ lọc]** Một phương pháp khác để thiết kế một bộ lọc FIR xấp xỉ các đặc tính của kênh mong muốn là dựa trên phương pháp cửa sổ. Trường hợp riêng, nếu đáp ứng kênh mong muốn là  $C(f)$  với  $|f| \leq W$  và  $C(f)=0$

với  $|f| > W$ , thì đáp ứng xung của kênh là

$$h(t) = \int_{-W}^W C(f) e^{j2\pi ft} df \quad (6.3.3)$$

Thí dụ, nếu kênh lý tưởng, thì  $C(f)=1$  với  $|f| \leq W$  và do vậy

$$h(t) = \frac{\sin 2\pi Wt}{\pi t} \quad (6.3.4)$$

Một bộ lọc số tương đương có thể được thực hiện bằng cách lấy mẫu  $h(t)$  tại  $t=nT_s$ , trong đó  $T_s$  là khoảng lấy mẫu còn  $n=0, \pm 1, \pm 2, \dots$ . Hãy thiết kế một bộ lọc FIR với  $W=2000$  Hz và  $F_s=1/T_s=10$  kHz.

### Lời giải

Phiên bản lấy mẫu của  $h(t) - h_n \equiv h(nT_s)$  – được minh họa trên hình 6.9. Do  $\{h_n\}$  có độ dài vô hạn, chúng ta có thể cắt nó tại một độ dài  $N$  nào đó. Việc cắt bỏ như vậy tương đương với việc nhân dãy  $\{h_n\}$  với một dãy cửa sổ chữ nhật  $w_n=1$  đối với  $|n| \leq (N-1)/2$  và  $w_n=0$  đối với  $|n| \geq (N+1)/2$ . Đáp ứng xung  $\{h_n^1 = h_n w_n\}$  và đáp ứng tần số tương ứng của bộ lọc FIR (đã được cắt) được minh họa trên hình 6.10 cho trường hợp  $N=51$ . Hãy để ý rằng bộ lọc được cắt có các búp phụ lớn trong băng chẵn. Do đó, bộ lọc FIR này là một xấp xỉ tồi đối với các đặc tính của kênh mong muốn. Kích thước của các búp phụ có thể giảm đáng kể được bằng cách áp dụng một hàm cửa sổ tròn hơn, chẳng hạn như một hàm cửa sổ Hanning hoặc một hàm cửa sổ Hamming. Hình 6.11 minh họa đáp ứng xung và đáp ứng tần số của  $\{h_n^1 = h_n w_n\}$  khi hàm cửa sổ là một cửa sổ Hanning có độ dài  $N=51$ . MATLAB có sẵn các thủ tục để thực hiện một số kiểu hàm cửa sổ khác nhau.

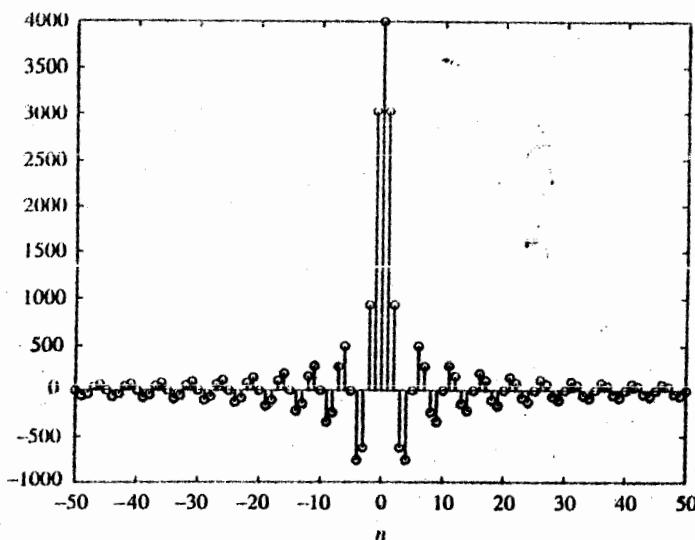
### m-file

```
% MATLAB script for Illustrative Problem 4, Chapter 6.
echo on
Length=101;
Fs=10000;
W=2000;
Ts=1/Fs;
n=-(Length-1)/2:(Length-1)/2;
```

```

t=Ts*n;
h=2*W*sinc(2*W*t);
% The rectangular windowed version follows...
N=61;
rec_windowed_h=h((Length-N)/2+1:(Length+N)/2);
% Frequency response of rec_windowed_h follows
[rec_windowed_H,W1]=freqz(rec_windowed_h,1);
% to normalize the magnitude
rec_windowed_H_in_dB=20*log10(abs(rec_windowed_H)/abs(rec_windowed_H(1)));
% The Hanning windowed version follows...
hanning_window=hanning(N);
hanning_windowed_h=h((Length-N)/2+1:(Length+N)/2).*hanning_window.';
[hanning_windowed_H,W2]=freqz(hanning_windowed_h,1);
hanning_windowed_H_in_dB=20*log10(abs(hanning_windowed_H)/abs(hannin g_windowed_H(1)));
% the plotting commands follow

```



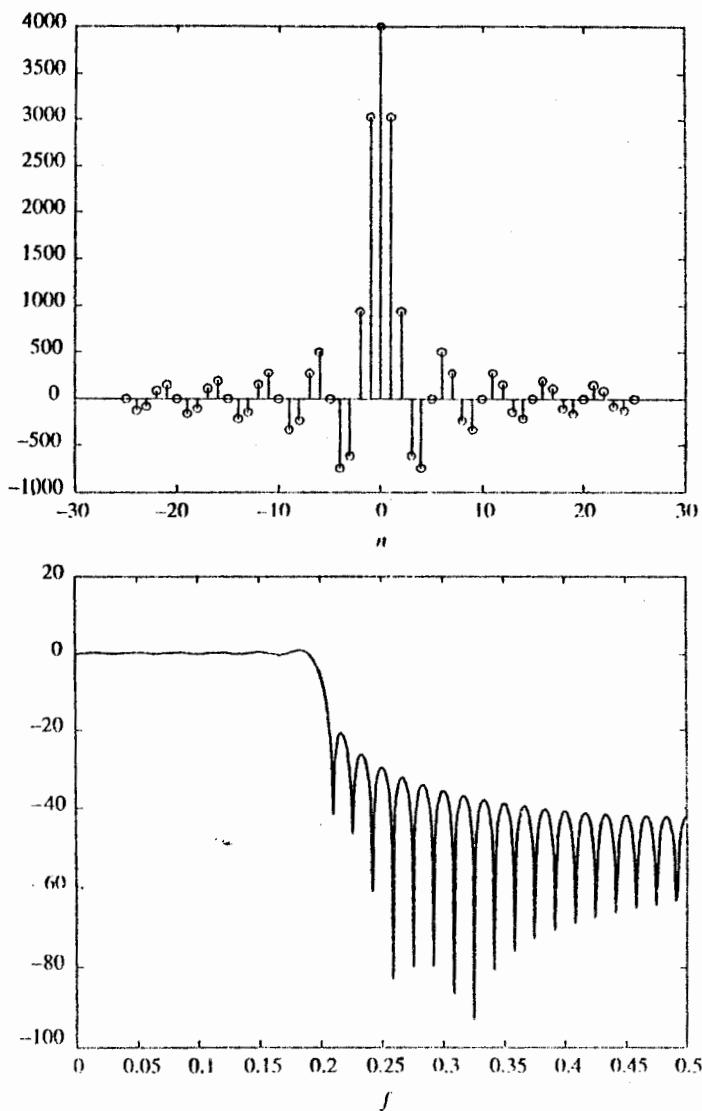
Hình 6.9 Các mẫu của  $h(n)$  trong bài tập minh họa 6.4

## BÀI TẬP MINH HOA

**Bài tập minh họa 6.5 [Mô phỏng kênh đa đường]** Một kênh vô tuyến (kênh đa đường) hai tia có thể mô hình hoá được trên miền thời gian như được minh họa trên hình 6.12. Đáp ứng xung của kênh có thể biểu diễn được theo

$$c(t, \tau) = b_1(t)\delta(t) + b_2(t)\delta(t-\tau_d) \quad (6.3.5)$$

trong đó  $b_1(t)$  và  $b_2(t)$  là các quá trình ngẫu nhiên biểu thị quá trình truyền dẫn sóng biến đổi theo thời gian của kênh còn  $\tau_d$  là trễ giữa hai thành phần đa đường đó. Hãy mô phỏng một kênh như thế trên máy tính.



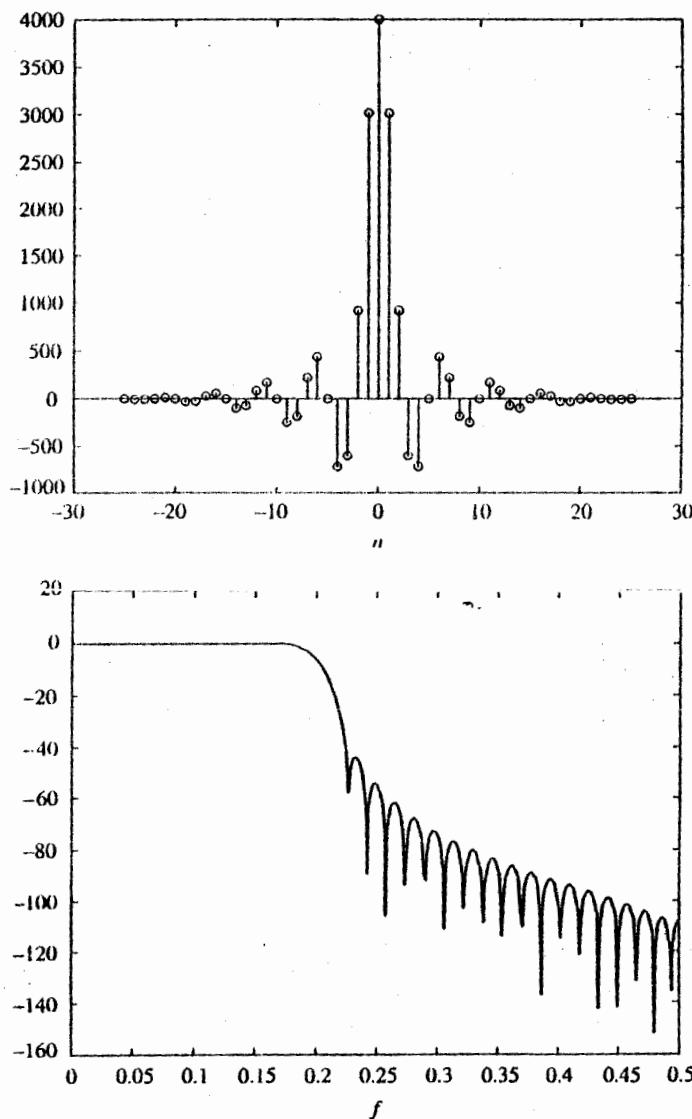
Hình 6.10. Đáp ứng xung và đáp ứng tần số của bộ lọc được cắt với một cửa sổ chữ nhật trong bài tập minh họa 6.4

### Lời giải

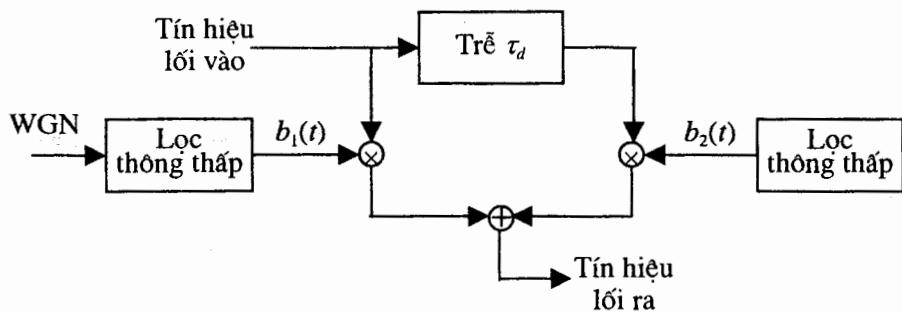
Chúng ta mô hình hóa  $b_1(t)$  và  $b_2(t)$  như các quá trình ngẫu nhiên Gauss được tạo ra bằng cách cho các quá trình tạp âm trắng Gauss qua các bộ lọc thông thấp.

Trong miền thời gian rời rạc, chúng ta có thể sử dụng các bộ lọc số IIR tương đối đơn giản được kích thích bởi các dãy tạp âm trắng Gauss (WGN). Thí dụ, một bộ lọc số thông thấp đơn giản có hai cực như nhau được mô tả bằng biến đổi z

$$H(z) = \frac{(1-p)^2}{(1-pz^{-1})^2} = \frac{(1-p)^2}{1-2pz^{-1}+p^2z^{-2}} \quad (6.3.6)$$



Hình 6.11 Đáp ứng xung và đáp ứng tần số của bộ lọc  
được cắt với cửa sổ Hanning trong bài tập minh họa 6.4

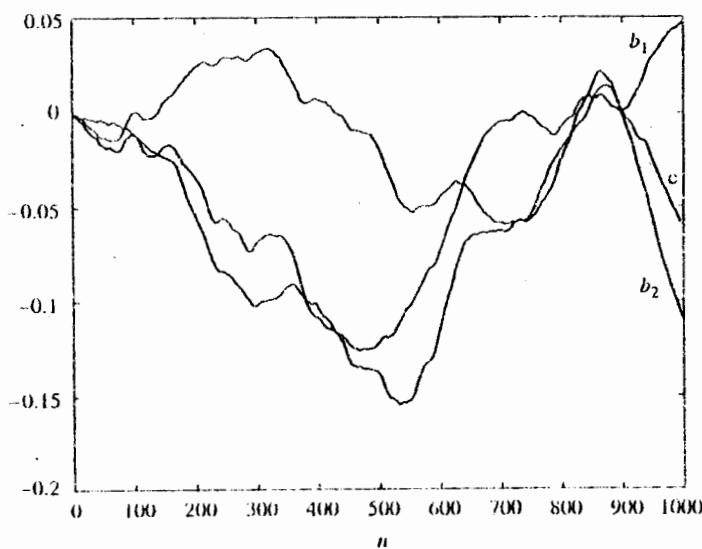


Hình 6.12. Mô hình kênh vô tuyến hai tia

Hay bằng phương trình vi phân tương ứng

$$b_n = 2pb_{n-1} - p^2b_{n-2} + (1-p)^2w_n \quad (6.3.7)$$

trong đó  $\{w_n\}$  là dãy lối vào WGN,  $\{b_n\}$  là dãy lối ra, còn  $0 < p < 1$  là vị trí cực. Vị trí của cực điều khiển độ rộng băng của bộ lọc và do vậy điều khiển tốc độ biến thiên của  $\{b_n\}$ . Khi  $p$  gần tới 1 (sát vòng tròn đơn vị) thì độ rộng băng của bộ lọc hẹp, ngược lại, khi  $p$  tiến đến 0, độ rộng băng lớn. Do đó, khi  $p$  tiến gần tới vòng tròn đơn vị trên mặt phẳng  $z$  thì chuỗi lối ra của bộ lọc biến thiên chậm hơn so với trường hợp khi cực ở gần gốc.

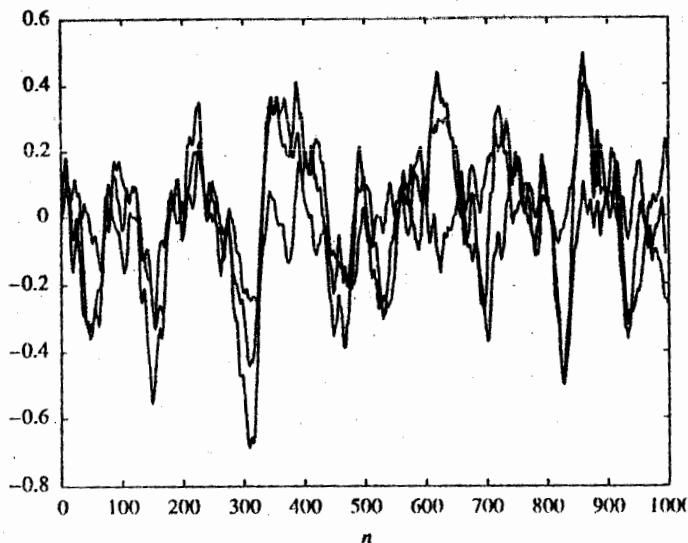
Hình 6.13 Các lối ra  $b_{1,n}$  và  $b_{2,n}$  của một bộ lọc thông thấp và đáp ứng xung  $c_n$  với  $p = 0.99$ 

Hình 6.13 minh họa các dãy đầu ra  $\{b_{1,n}\}$  và  $\{b_{2,n}\}$  tạo được bằng cách cho các dãy WGN độc lập thống kê qua một bộ lọc có  $p=0.99$ . Đáp ứng xung rời rạc của kênh

$$c_n = b_{1,n} + b_{2,n-d} \quad (6.3.8)$$

cũng được thể hiện, với trễ  $d=5$  mẫu.

Hình 6.14 minh họa các dãy  $\{b_{1,n}\}$  và  $\{b_{2,n}\}$  và  $\{c_n\}$  khi  $p=0.9$ .



Hình 6.14. Các lối ra  $b_{1,n}$ ,  $b_{2,n}$ , và  $c_n$  với cực tại  $p=0.9$  (từ đỉnh tới đáy tại  $n=1000$ ,  $b_{1,n}$ ,  $c_n$  và  $b_{2,n}$ )

#### 6.4 ĐẶC TRUNG CỦA XUYÊN NHIỀU GIỮA CÁC DẤU

Trong một hệ thống thông tin số, méo kênh gây ra xuyên nhiễu giữa các dấu (ISI). Trong mục này, chúng tôi trình bày một mô hình đặc trưng cho ISI. Để đơn giản, chúng ta giả thiết rằng tín hiệu phát là một tín hiệu PAM băng gốc. Tuy thế, cách đề cập này có thể mở rộng được cho các tín hiệu được điều chế (tuyến tính) sóng mang như sẽ được thảo luận trong chương sau.

Tín hiệu phát PAM biểu diễn được theo

$$s(t) = \sum_{n=0}^{\infty} a_n g(t - nT) \quad (6.4.1)$$

trong đó  $g(t)$  là dạng xung cơ sở được chọn để kiểm soát các đặc tính phổ của tín hiệu phát,  $\{a_n\}$  là dãy các *symbol* thông tin được truyền đi, được chọn từ một biểu đồ sao tín hiệu (*constellation*) gồm  $M$  điểm, còn  $T$  là thời khoảng của tín hiệu ( $1/T$  là tốc độ *symbol*).

Tín hiệu  $s(t)$  được phát qua một kênh băng gốc có thể đặc trưng được bởi một đáp ứng tần số  $C(f)$ . Hệ quả là, tín hiệu thu được có thể biểu diễn được theo

$$r(t) = \sum_{n=0}^{\infty} a_n h(t - nT) + w(t) \quad (6.4.2)$$

trong đó  $h(t) = g(t) * c(t)$ ,  $c(t)$  là đáp ứng xung của kênh, dấu  $*$  ký hiệu cho tích chập, còn  $w(t)$  biểu thị tạp âm cộng trong kênh. Để đặc trưng ISI, hãy giả sử rằng tín hiệu thu được được cho qua một bộ lọc thu rồi lấy mẫu với tốc độ  $1/T$  mẫu/s. Nói chung, bộ lọc tối ưu tại máy thu thì phối hợp với xung tín hiệu thu  $h(t)$ . Do đó, đáp ứng tần số của bộ lọc này là  $H^*(t)$ . Chúng ta ký hiệu lối ra của nó theo

$$y(t) = \sum_{n=0}^{\infty} a_n x(t - nT) + v(t) \quad (6.4.3)$$

trong đó  $x(t)$  là đáp ứng xung tín hiệu của bộ lọc thu, nghĩa là  $X(f) = H(f)H^*(f) = |H(f)|^2$ , còn  $v(t)$  là đáp ứng của bộ lọc thu đối với tạp âm  $w(t)$ . Bây giờ, nếu  $y(t)$  được lấy mẫu tại các thời điểm  $t = kT$ ,  $k = 0, 1, 2, \dots$ , chúng ta có

$$\begin{aligned} y(kT) &= \sum_{n=0}^{\infty} a_n x(kT - nT) + v(kT) \\ y_k &= \sum_{n=0}^{\infty} a_n x_{k-n} + v_k, \quad k = 0, 1, 2, \dots \end{aligned} \quad (6.4.4)$$

Các giá trị mẫu  $\{y_k\}$  có thể biểu diễn được theo

$$y_k = x_0 \left( a_k + \sum_{\substack{n=0 \\ n \neq k}}^{\infty} a_n x_{k-n} \right) + v_k, \quad k = 0, 1, 2, \dots \quad (6.4.5)$$

$x_0$  là một hệ số tỷ lệ tùy ý, để đơn giản chúng ta cho nó bằng đơn vị. Khi đó:

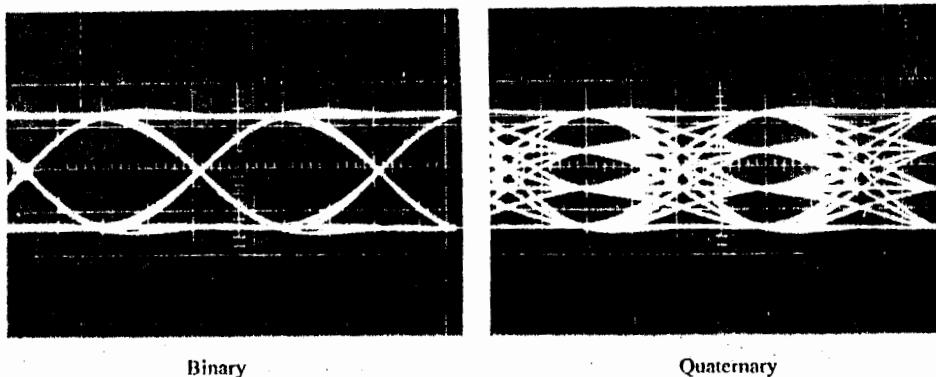
$$y_k = a_k + \sum_{\substack{n=0 \\ n \neq k}}^{\infty} a_n x_{k-n} + v_k \quad (6.4.6)$$

Số hạng  $a_k$  biểu thị *symbol* thông tin mong muốn tại thời điểm lấy mẫu thứ  $k$ , còn số hạng

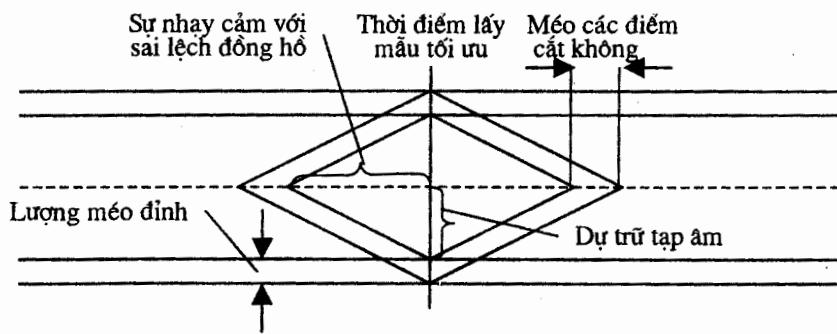
$$\sum_{\substack{n=0 \\ n \neq k}}^{\infty} a_n x_{k-n} \quad (6.4.7)$$

biểu thị ISI, còn  $v(k)$  là tạp âm cộng tại thời điểm lấy mẫu thứ  $k$ .

Đại lượng ISI và tạp âm trong một hệ thống truyền tin số có thể quan sát được trên một máy hiện sóng. Đối với các tín hiệu PAM, chúng ta có thể hiển thị tín hiệu thu được  $y(t)$  trên lối vào trực đứng với tốc độ quét trực nằm ngang  $1/T$ . Hình hiển thị trên máy hiện sóng được gọi là một *mẫu mắt* (*eye pattern*) do giống mắt người. Để làm thí dụ, hình 6.15 minh họa các mẫu mắt đối với điều chế PAM nhị phân (*binary*) và 4 mức (*quaternary*). Ảnh hưởng của ISI là làm cho mắt đóng dần lại, do đó làm giảm độ dự trữ đối với tạp âm cộng dẫn đến các lối. Hình 6.16 minh họa bằng đồ thị tác động của ISI làm giảm độ mở của mắt đối với tín hiệu nhị phân. Hãy để ý rằng ISI làm méo lệch vị trí của các điểm cắt qua không và gây giảm độ mở mẫu mắt. Vì thế, nó làm cho hệ thống nhạy cảm hơn đối với lỗi về đồng bộ.



Hình 6.15 Các thí dụ về các mẫu mắt đối với ASK (hay PAM ) nhị phân và 4 mức



Hình 6.16 Ảnh hưởng của ISI lên độ mở mẫu mắt

## BÀI TẬP MINH HOA

**Bài tập minh họa 6.6 [Xuyên nhiễu giữa các dấu (ISI)]** Trong bài tập này, chúng ta xét ảnh hưởng của ISI lên dãy tín hiệu thu được  $\{y_k\}$  đối với hai kênh mà chúng gay nên các dãy  $\{x_k\}$  như sau

### Kênh 1

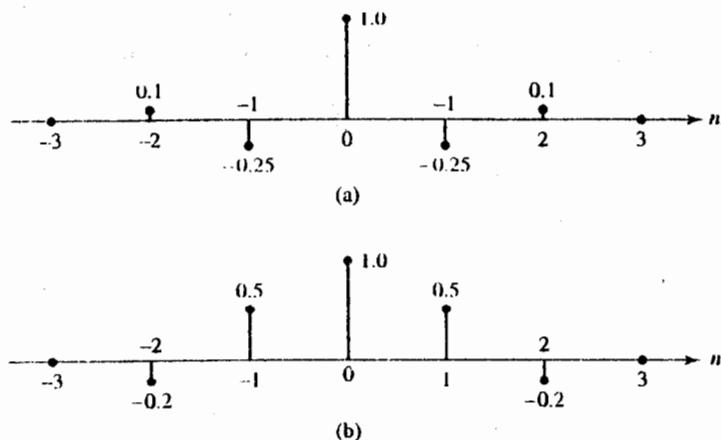
$$x_n = \begin{cases} 1, & n = 0 \\ -0.25, & n = \pm 1 \\ 0.1, & n = \pm 2 \\ 0, & \text{với các giá trị khác của } n \end{cases}$$

### Kênh 2

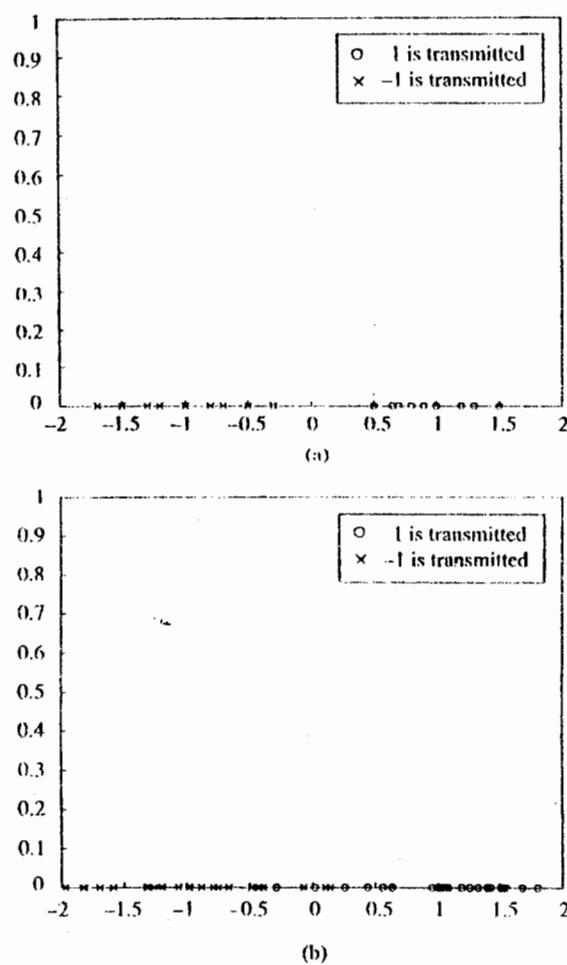
$$x_n = \begin{cases} 1, & n = 0 \\ 0.5, & n = \pm 1 \\ -0.2, & n = \pm 2 \\ 0, & \text{với các giá trị khác của } n \end{cases}$$

Hãy chú ý rằng trong các kênh này, ISI được giới hạn lại chỉ còn từ hai *symbol* về phía bên này hay phía bên kia của tín hiệu đã được truyền mong muốn. Do đó, việc mắc nối tiếp các bộ lọc thu, phát và kênh tại các thời điểm lấy mẫu biểu diễn được bằng bộ lọc kênh FIR rồi rạc về thời gian tương đương như được thể hiện trên hình 6.17. Bây giờ, hãy giả thiết rằng dãy tín hiệu phát là nhị phân, nghĩa là  $\{a_n = \pm 1\}$ . Khi đó, đối với kênh 1, dãy tín hiệu thu được  $\{y_n\}$  khi không có tạp âm được biểu diễn trên hình 6.18(a), còn khi có tạp âm cộng với phương sai  $\sigma^2=0.1$  thì dãy tín hiệu thu được biểu diễn trên hình 6.18(b). Chúng ta để ý rằng, khi không có tạp âm, chỉ riêng ISI không gây ra các lỗi tại bộ tách tín hiệu mà nó thực hiện so sánh dãy tín hiệu thu  $\{y_n\}$  với mức ngưỡng được đặt bằng 0. Do vậy, mẫu mắt vẫn mở khi không có tạp âm. Tuy nhiên, khi tạp âm cộng đủ lớn, các lỗi sẽ xảy ra.

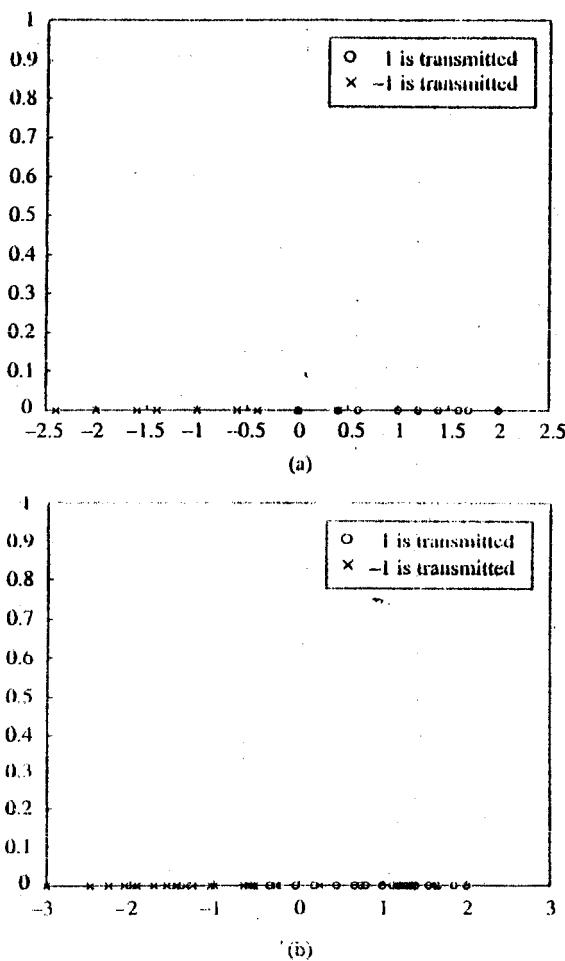
Trong trường hợp của kênh 2, các dãy  $\{y_n\}$  không có tạp âm và có tạp âm ( $\sigma^2=0.1$ ) được minh họa trên hình 6.19. Bây giờ, chúng ta thấy rằng ISI có thể gây nên các lỗi tại bộ tách tín hiệu mà nó thực hiện việc so sánh dãy thu được  $\{y_n\}$  với mức ngưỡng được đặt bằng 0, thậm chí ngay cả khi không có tạp âm. Như thế, đối với đặc tính kênh này, mẫu mắt hoàn toàn bị đóng.



Hình 6.17. Các mô hình kênh FIR với ISI: (a) kênh 1; (b) kênh 2



Hình 6.18. Đầu ra mô hình kênh 1, có và không có AWGN:  
(a) không có tạp âm; (b) AWGN với  $\sigma^2=0.1$



Hình 6.19. Đầu ra mô hình kênh 2, có và không có AWGN:  
 (a) không có tạp âm; (b) AWGN với  $\sigma^2=0.1$

## 6.5 THIẾT KẾ HỆ THỐNG TRUYỀN TIN ĐỐI VỚI CÁC KÊNH CÓ BĂNG THÔNG HỮU HẠN

Trong mục này, chúng ta sẽ xem xét việc thiết kế các bộ lọc phát và thu thích hợp với một kênh băng gốc có băng thông hữu hạn. Hai trường hợp sẽ được xem xét. Trong trường hợp thứ nhất, việc thiết kế dựa trên các mạch lọc phát và thu dẫn đến ISI bằng 0. Trong trường hợp thứ hai, việc thiết kế dựa trên các mạch lọc thu và phát mà chúng có một lượng ISI xác định (định trước được). Như thế, giải pháp thiết kế thứ hai dẫn đến một lượng ISI kiểm soát được. Các tín hiệu phát tương ứng đối với trường hợp này được gọi là các tín hiệu đáp ứng một phần (*partial response signals*).

## 6.5.1 THIẾT KẾ TÍN HIỆU ĐỂ ISI BẰNG KHÔNG

Việc thiết kế các tín hiệu có độ rộng băng hạn chế với ISI bằng không đã là một vấn đề được Nyquist nghiên cứu vào quãng 70 năm về trước. Ông đã phát biểu rằng một điều kiện cần và đủ để một tín hiệu  $x(t)$  có ISI bằng không, nghĩa là

$$x(nT) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \quad (6.5.1)$$

là biến đổi Fourier  $X(f)$  của nó thoả mãn

$$\sum_{m=-\infty}^{\infty} X\left(f + \frac{m}{T}\right) = T \quad (6.5.2)$$

trong đó  $1/T$  là tốc độ *symbol*.

Nói chung, nhiều tín hiệu có thể thiết kế được để có tính chất này. Một trong những tín hiệu hay được dùng nhất trong thực tế là tín hiệu có đặc tính tần số dạng cosin nâng được xác định theo

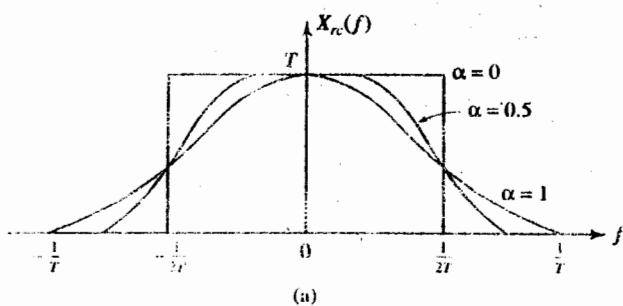
$$X_{rc}(f) = \begin{cases} T, & 0 \leq |f| \leq \frac{1-\alpha}{2T} \\ \frac{T}{2} \left[ 1 + \cos \frac{\pi T}{\alpha} \left( |f| - \frac{1-\alpha}{2T} \right) \right], & \frac{1-\alpha}{2T} < |f| \leq \frac{1+\alpha}{2T} \\ 0, & |f| > \frac{1+\alpha}{2T} \end{cases} \quad (6.5.3)$$

trong đó  $\alpha$  được gọi là hệ số uốn (*roll-off factor*), nhận các giá trị trong khoảng  $0 \leq \alpha \leq 1$ , còn  $1/T$  là tốc độ *symbol*. Đáp ứng tần số  $X_{rc}(f)$  được minh họa trên hình 6.20(a) đối với  $\alpha=0$ ,  $\alpha=\frac{1}{2}$  và  $\alpha=1$ . Chú ý rằng khi  $\alpha=0$ ,  $X_{rc}(f)$  suy biến thành một đáp ứng tần số lý tưởng dạng “tường gạch” (*brick wall*) không thể thực hiện được về mặt vật lý, với độ rộng băng thông  $1/2T$ . Tần số  $1/2T$  được gọi là *tần số Nyquist*. Với  $\alpha>0$ , độ rộng băng tần chiếm bởi tín hiệu bên ngoài tần số Nyquist  $1/2T$  được gọi là *độ rộng băng dôi (excess bandwidth)*, thường được biểu diễn theo phần trăm của tần số Nyquist. Ví dụ, với  $\alpha=\frac{1}{2}$ , độ rộng băng dôi bằng 50% và với  $\alpha=1$ , độ rộng băng dôi bằng 100%. Xung tín hiệu  $x_{rc}(t)$  có phổ cosin nâng là

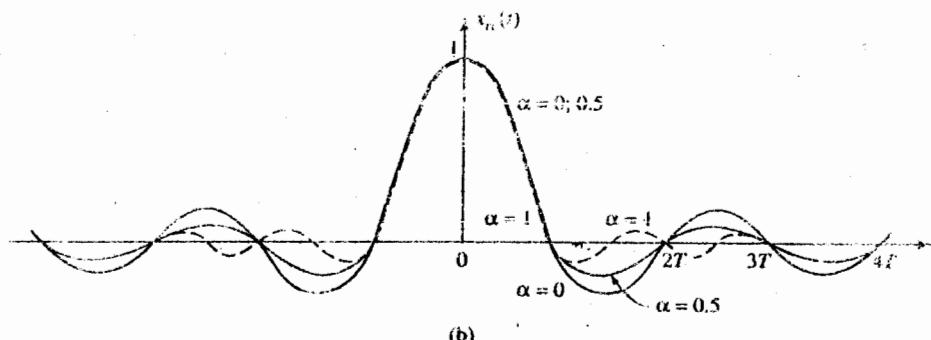
$$x_{rc}(t) = \frac{\sin \pi t/T}{\pi t/T} \frac{\cos(\pi \alpha t/T)}{1 - 4\alpha^2 t^2/T^2} \quad (6.5.4)$$

Hình 6.20(b) minh họa  $x_{rc}(t)$  đối với  $\alpha=0, \frac{1}{2}, 1$ . Do  $X_{rc}(f)$  thoả mãn (6.5.2),

chúng ta chú ý rằng  $x_{rc}(t)=1$  tại  $t=0$  và  $x_{rc}(t)=0$  tại  $t=kT$  ( $k=\pm 1, \pm 2, \dots$ ). Như vậy, tại các thời điểm lấy mẫu  $t=kT$  ( $k \neq 0$ ) không có ISI từ các symbol lân cận khi không có méo kênh. Tuy nhiên, khi có méo kênh, ISI được cho bởi (6.4.7) sẽ không bằng không nữa, và một bộ san bằng kênh là cần thiết để giảm thiểu ảnh hưởng của méo tới chất lượng hệ thống. Các bộ san bằng kênh sẽ được xem xét trong mục 6.6.



(a)



(b)

Hình 6.20. (a) Đáp ứng tần số cosin nâng; (b) các dạng xung đối với đáp ứng tần số cosin nâng

Trong một kênh lý tưởng, các bộ lọc phát và thu được thiết kế cùng nhau để có ISI bằng không tại các thời điểm lấy mẫu mong muốn  $t=nT$ . Như thế, nếu  $G_T(f)$  là đáp ứng tần số của bộ lọc phát còn  $G_R(f)$  là đáp ứng tần số của bộ lọc thu, thì tích (mắc nối tiếp hai bộ lọc này)  $G_T(f) G_R(f)$  được thiết kế để đạt được ISI bằng không. Thí dụ, nếu tích  $G_T(f) G_R(f)$  được chọn theo

$$G_T(f) G_R(f) = X_{rc}(f) \quad (6.5.5)$$

Trong đó  $X_{rc}(f)$  là đặc tính đáp ứng tần số cosin nâng, thì ISI tại các thời điểm lấy mẫu  $t=nT$  sẽ bằng 0.

### BÀI TẬP MINH HỌA

**Bài tập minh họa 6.7 [Các bộ lọc thu và phát tối ưu]** Hãy thiết kế một thực hiện bằng số các bộ lọc phát và thu  $G_T(f)$ ,  $G_R(f)$  sao cho tích của chúng thỏa mãn (6.5.5) và  $G_R(f)$  là bộ lọc phôi hợp đối với  $G_T(f)$ .

#### Lời giải

Cách đơn giản nhất để thiết kế và thực hiện các bộ lọc thu và phát ở dạng số là áp dụng các bộ lọc FIR với pha tuyến tính (đáp ứng xung đối xứng). Đáp ứng biên độ mong muốn là

$$|G_T(f)| = |G_R(f)| = \sqrt{X_{rc}(f)} \quad (6.5.6)$$

trong đó  $X_{rc}(f)$  được cho bởi (6.5.3). Đáp ứng tần số quan hệ với đáp ứng xung của bộ lọc số theo phương trình

$$G_T(f) = \sum_{n=-(N-1)/2}^{(N-1)/2} g_T(n) e^{-j2\pi fnT_s} \quad (6.5.7)$$

trong đó  $T_s$  là khoảng lấy mẫu, còn  $N$  là độ dài của bộ lọc. Hãy để ý rằng  $N$  lẻ. Do  $G_T(f)$  có băng thông hữu hạn, chúng ta có thể chọn tần số lấy mẫu  $F_s$  ít nhất là  $2/T$ . Lựa chọn của chúng ta là

$$F_s = \frac{1}{T_s} = \frac{4}{T}$$

hay, một cách tương đương là  $T_s=T/4$ . Do vậy, tần số gập phổ là  $F_s/2=2/T$ . Vì  $G_T(f)=\sqrt{X_{rc}(f)}$ , chúng ta có thể lấy mẫu  $X_{rc}(f)$  tại các điểm tần số cách đều nhau, với phân cách tần số  $\Delta f = F_s/N$ . Như thế, chúng ta có

$$\sqrt{X_{rc}(m\Delta f)} = \sqrt{X_{rc}\left(\frac{mF_s}{N}\right)} = \sum_{n=-(N-1)/2}^{(N-1)/2} g_T(n) e^{-j2\pi mn/N} \quad (6.5.8)$$

Quan hệ biến đổi ngược là

$$g_T(n) = \sum_{m=-(N-1)/2}^{(N-1)/2} \sqrt{X_{rc}\left(\frac{4m}{NT}\right)} e^{j2\pi mn/N}, \quad n = 0, \pm 1, \dots, \pm \frac{N-1}{2} \quad (6.5.9)$$

Do  $g_T(n)$  đối xứng, đáp ứng xung của bộ lọc phát pha tuyến tính mong muốn nhận được bằng cách làm trễ  $g_T(n)$  đi  $\frac{N-1}{2}$  mẫu.

Dưới đây là các MATLAB script cho tính toán này.

### m-file

```
% MATLAB script for Illustrative Problem 7, Chapter 6.  
echo on  
N=31;  
T=1;  
alpha=1/4;  
n=-(N-1)/2:(N-1)/2; % The indices for g_T  
% The expression for g_T is obtained next  
for i=1:length(n),  
    g_T(i)=0;  
    for m=-(N-1)/2:(N-1)/2,  
        g_T(i)=g_T(i)+sqrt(xrc(4*m/(N*T),alpha,T))*exp(j*2*pi*m*n(i)/N);  
    echo off ;  
    end;  
end;  
echo on ;  
g_T=real(g_T); % The imaginary part is due to the finite machine precision  
% derive g_T(n-(N-1)/2)  
n2=0:N-1;  
% get the frequency response characteristics  
[G_T,W]=freqz(g_T,1);  
% normalized magnitude response  
magG_T_in_dB=20*log10(abs(G_T)/max(abs(G_T)));  
% Impulse response of the cascade of the transmitter and the receiver filters..  
g_R=g_T;  
imp_resp_of_cascade=conv(g_R,g_T);  
% plotting commands follow
```

### m-file

```
function [y] = xrc(f,alpha,T);  
% [y]=xrc(f,alpha,T)  
% evaluates the expression Xrc(f). The parameters alpha and T  
% must also be given as inputs to the function.  
if (abs(f) > ((1+alpha)/(2*T))),  
    y=0;  
elseif (abs(f) > ((1-alpha)/(2*T))),
```

```

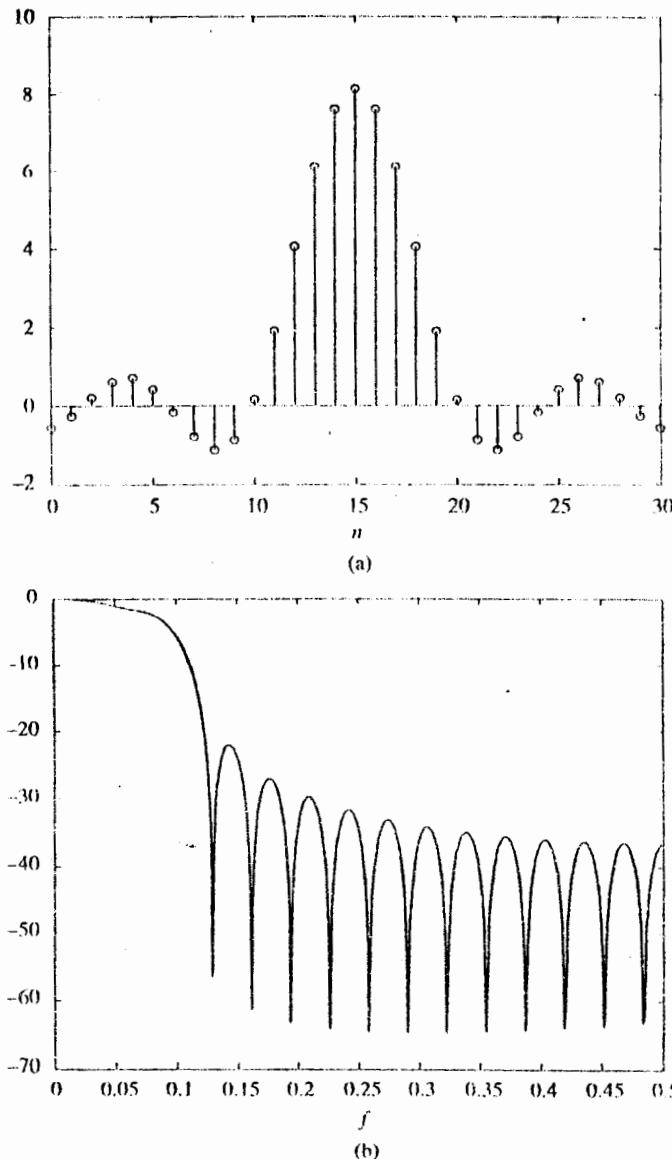
y=(T/2)*(1+cos((pi*T/alpha)*(abs(f)-(1-alpha)/(2*T))));  

else  

    y=T;  

end;

```



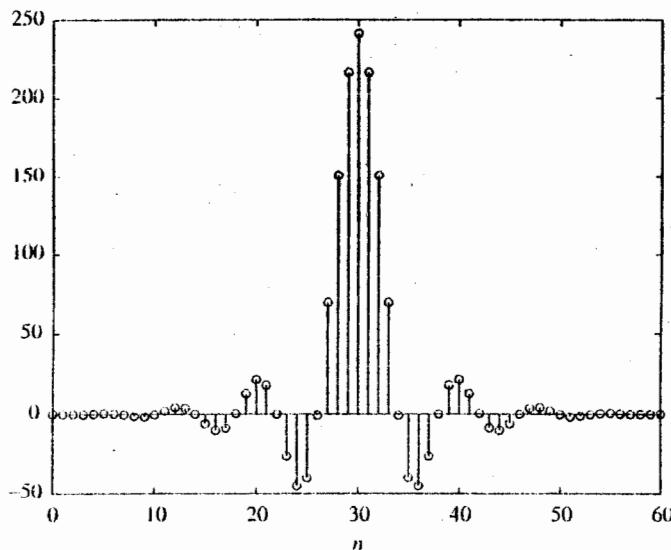
Hình 6.21 (a) Đáp ứng xung và (b) đáp ứng tần số của bộ lọc phát FIR rời rạc được cắt

Hình 6.21(a) minh họa  $g_T(n - \frac{N-1}{2})$ , trong đó  $n=0, 1, 2, \dots, N-1$ , đối với  $\alpha = \frac{1}{4}$

và  $N=31$ . Các đặc tuyến đáp ứng tần số tương ứng được thể hiện trên hình 6.21(b).  
Chú ý rằng đáp ứng tần số không còn bằng không nữa khi  $|f| \geq (1 + \alpha)/T$  do bộ lọc

số có khoảng thời gian tồn tại đáp ứng xung hữu hạn. Tuy nhiên, các búp phụ của phổ khá nhỏ. Việc giảm nhỏ hơn nữa các búp phụ có thể đạt được bằng cách tăng  $N$ .

Cuối cùng, trên hình 6.22, chúng tôi thể hiện đáp ứng xung của mạch mắc nối tiếp các bộ lọc FIR thu và phát. Đáp ứng xung này có thể so sánh được với đáp ứng xung lí tưởng nhận được bằng cách lấy mẫu  $x_{rc}(t)$  với tốc độ  $F_s=4/T$ .



Hình 6.22 Đáp ứng xung của dãy nối tiếp bộ lọc phát và bộ lọc phối hợp tại máy thu

### 6.5.2 THIẾT KẾ TÍN HIỆU ĐỂ CÓ ISI KIỂM SOÁT ĐƯỢC

Như chúng ta đã thấy được từ các thảo luận của chúng ta về việc thiết kế tín hiệu để ISI bằng không, một bộ lọc phát với độ rộng băng đôi có thể áp dụng được để thực hiện các bộ lọc phát và thu thực tế đối với các kênh có băng thông hạn chế. Mặt khác, giả sử chúng ta chọn cách làm nhẹ điều kiện ISI bằng không và nhờ thế truyền dẫn *symbol* được trong một độ rộng băng  $W=1/2T$ , nghĩa là không có độ rộng băng đôi. Bằng việc tính đến một lượng kiểm soát được của ISI, chúng ta có thể đạt được tốc độ  $2W$  *symbol/s*.

Chúng ta đã thấy được rằng điều kiện để không có ISI là  $x(nT)=0$  đối với  $n \neq 0$ . Tuy nhiên, hãy giả thiết rằng chúng ta đi thiết kế tín hiệu có độ rộng băng hạn chế có ISI kiểm soát được tại một thời điểm. Điều này có nghĩa là chúng ta chấp nhận một lượng phụ khác không được cộng thêm vào các mẫu  $\{x(nT)\}$ . ISI

mà chúng ta đưa vào này là “tất định” hay “kiểm soát được”; do đó, nó có thể tính đến được tại máy thu như sẽ được bàn đến sau.

Nói chung, một tín hiệu  $x(t)$  được hạn chế độ rộng băng tới  $W$  Hz, tức là

$$X(f)=0, \quad |f| > W \quad (6.5.10)$$

có thể biểu diễn được theo

$$x(t) = \sum_{n=-\infty}^{\infty} x\left(\frac{n}{2W}\right) \frac{\sin 2\pi W(t - n/2W)}{2\pi W(t - n/2W)} \quad (6.5.11)$$

Cách biểu diễn này xuất phát từ định lí lấy mẫu đối với tín hiệu có độ rộng băng hữu hạn. Phổ của tín hiệu có độ rộng băng hữu hạn là

$$\begin{aligned} X(f) &= \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \\ &= \begin{cases} \frac{1}{2W} \sum_{n=-\infty}^{\infty} x\left(\frac{n}{2W}\right) e^{-jn\pi f/W}, & |f| \leq W \\ 0, & |f| > W \end{cases} \end{aligned} \quad (6.5.12)$$

Một trường hợp đặc biệt, dẫn đến các bộ lọc thu và phát có thể thực hiện được về mặt vật lý, được xác định bởi các mẫu

$$x\left(\frac{n}{2W}\right) \equiv x(nT) = \begin{cases} 1, & n = 0, 1 \\ 0, & \text{với các giá trị khác của } n \end{cases} \quad (6.5.13)$$

Phổ của tín hiệu ưng ứng là

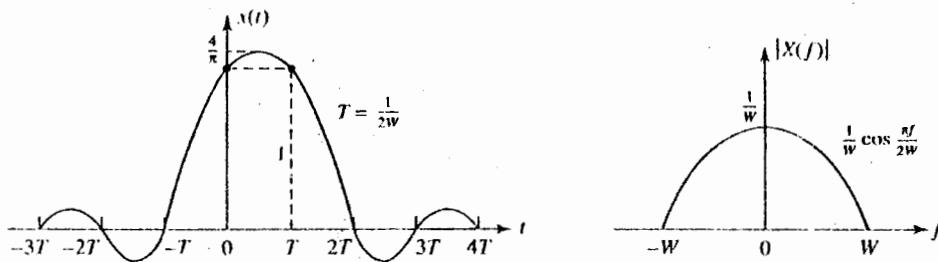
$$\begin{aligned} X(f) &= \begin{cases} \frac{1}{2W} (1 + e^{-j\pi f/W}), & |f| < W \\ 0, & \text{với các giá trị khác của } f \end{cases} \\ &= \begin{cases} \frac{1}{W} e^{-j\pi f/2W} \cos\left(\frac{\pi f}{2W}\right), & |f| < W \\ 0, & \text{với các giá trị khác của } f \end{cases} \end{aligned} \quad (6.5.14)$$

Do đó,  $x(t)$  được cho bởi

$$x(t) = \operatorname{sinc}(2Wt) + \operatorname{sinc}(2Wt - 1) \quad (6.5.15)$$

trong đó  $\operatorname{sinc}(t) = \sin(\pi t)/(\pi t)$ . Xung này được gọi là *xung tín hiệu nhị phân đối (duobinary signal pulse)*. Xung này được minh họa trên hình 6.23 cùng với

phổ biên độ của nó. Ta để ý rằng phổ này giảm một cách trơn chu tới không, mà điều đó có nghĩa các bộ lọc thực hiện được về mặt vật lý có thể thiết kế được xấp xỉ rất gần với phổ này. Như vậy, đạt được một tốc độ *symbol* bằng  $2W$ .



Hình 6.23 Xung tín hiệu nhị phân đôi và phổ của nó

Một trường hợp đặc biệt khác dẫn tới các bộ lọc thu và phát thực hiện được về mặt vật lý được xác định bởi các mẫu

$$x\left(\frac{n}{2W}\right) = x(nT) = \begin{cases} 1, & n = 1 \\ -1, & n = -1 \\ 0, & \text{với các } n \text{ khác} \end{cases} \quad (6.5.16)$$

Xung  $x(t)$  tương ứng được cho theo

$$x(t) = \text{sinc}(2Wt + 1) - \text{sinc}(2Wt - 1) \quad (6.5.17)$$

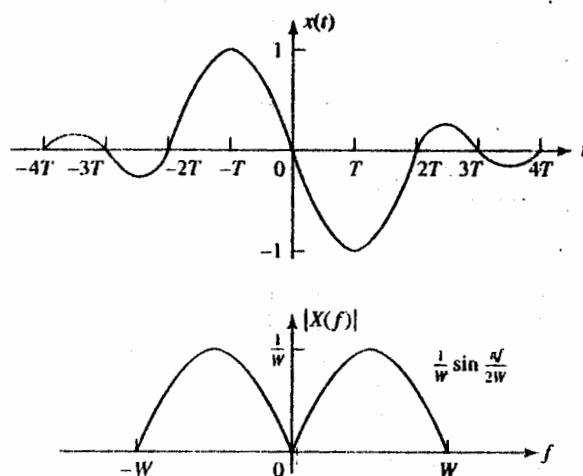
Phổ của nó là

$$X(f) = \begin{cases} \frac{1}{2W} (e^{j\pi f/W} - e^{-j\pi f/W}), & |f| \leq W \\ 0, & |f| > W \end{cases} = \frac{j}{W} \sin \frac{\pi f}{W}, \quad |f| \leq W \quad (6.5.18)$$

Xung này và phổ biên độ của nó được minh họa trên hình 6.24. Nó được gọi là một xung tín hiệu nhị phân đôi cải biến (*modified duobinary signal pulse*). Đáng lưu ý rằng phổ của tín hiệu này có một trị bằng không tại  $f = 0$ , làm cho nó trở nên thích hợp để truyền dẫn qua một kênh không cho qua thành phần một chiều.

Cũng có thể nhận được các đặc tính lọc có thể thực hiện được về mặt vật lý đáng quan tâm khác nữa bằng cách lựa chọn các giá trị khác nhau đối với các mẫu  $\{x(n/2W)\}$  và chọn có hơn hai mẫu khác không. Tuy nhiên, khi chúng ta càng chọn nhiều mẫu khác không hơn nữa, vấn đề loại bỏ ISI kiểm soát được sẽ trở nên càng

phức tạp hơn và không thực tế.



Hình 6.24 Xung tín hiệu nhị phân đôi cải biên và phổ của nó

Các tín hiệu nhận được khi ISI kiểm soát được đưa vào một cách có chủ ý bằng cách chọn hai hoặc hơn các mẫu khác không từ tập  $\{x(n/2W)\}$  được gọi là các tín hiệu đáp ứng một phần (*partial response signal*). Các xung tín hiệu có được cho phép chúng ta truyền các *symbol* thông tin với tốc độ Nyquist là  $2W$  symbol/s. Nhờ thế, có được hiệu quả sử dụng băng tần lớn hơn so với các xung tín hiệu cosin nâng.

### BÀI TẬP MINH HOẠ

**Bài tập minh họa 6.8 [Truyền tín hiệu nhị phân đôi]** Hãy thiết kế một thực hiện dạng số các bộ lọc máy thu và phát  $G_T(f)$  và  $G_R(f)$  sao cho tích của chúng bằng với phổ của một xung nhị phân đôi còn  $G_R(f)$  thì phối hợp với  $G_T(f)$ .

#### Lời giải

Để thoả mãn chỉ tiêu trên miền tần số, chúng ta có

$$|G_T(f)||G_R(f)| = \begin{cases} \frac{1}{W} \cos\left(\frac{\pi f}{2W}\right), & |f| \leq W \\ 0, & |f| > W \end{cases} \quad (6.5.19)$$

Và do đó,

$$|G_T(f)| = \begin{cases} \sqrt{\frac{1}{W} \cos\left(\frac{\pi f}{2W}\right)}, & |f| \leq W \\ 0, & |f| > W \end{cases} \quad (6.5.20)$$

Bây giờ, chúng ta làm theo cùng một giải pháp như trong bài tập minh họa 6.7 để nhận được các đáp ứng xung đối với việc thực hiện FIR của các mạch lọc phát và thu. Như vậy, với  $W=1/2T$  còn  $F_s=4/T$ , chúng ta sẽ có

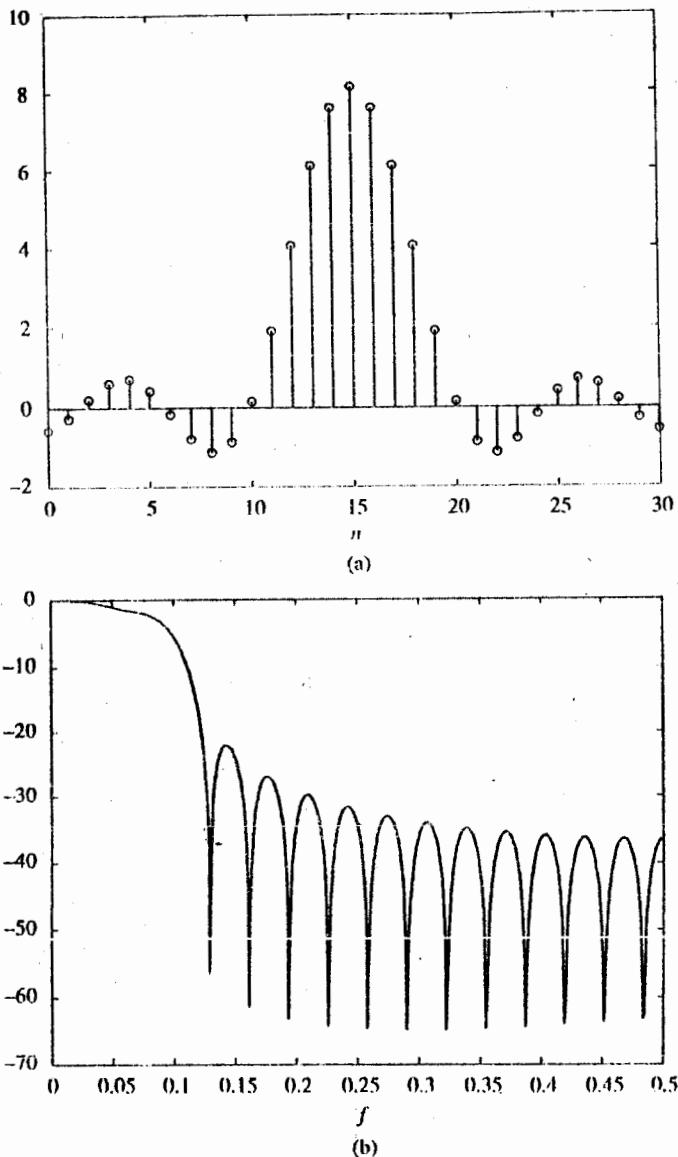
$$g_T(n) = \sum_{m=-(N-1)/2}^{(N-1)/2} \left| G_T \left( \frac{4m}{NT} \right) \right| e^{j2\pi mn/N}, \quad n=0, \pm 1, \dots, \pm \frac{N-1}{2} \quad (6.5.21)$$

và  $g_R(n)=g_T(n)$ .

MATLAB script đối với tính toán này được cho dưới đây.

### m-file

```
% MATLAB script for Illustrative Problem 8, Chapter 6.
echo on
N=31;
T=1;
W=1/(2*T);
n=-(N-1)/2:(N-1)/2; % The indeces for g_T
% The expression for g_T is obtained next
for i=1:length(n),
    g_T(i)=0;
    for m=-(N-1)/2:(N-1)/2,
        if ( abs((4*m)/(N*T)) <= W ),
            g_T(i)=g_T(i)+sqrt((1/W)*cos((2*pi*m)/(N*T*W)))*exp(j*2*pi*m*n(i)/N);
        end;
        echo off ;
    end;
end;
echo on ;
g_T=real(g_T); % The imaginary part is due to the finite machine precision
% obtain g_T(n-(N-1)/2)
n2=0:N-1;
% obtain the frequency response characteristics
[G_T,W]=freqz(g_T,1);
% normalized magnitude response
magG_T_in_dB=20*log10(abs(G_T)/max(abs(G_T)));
% Impulse response of the cascade of the transmitter and the receiver filters
g_R=g_T;
imp_resp_of_cascade=conv(g_R,g_T);
% plotting commands follow
```

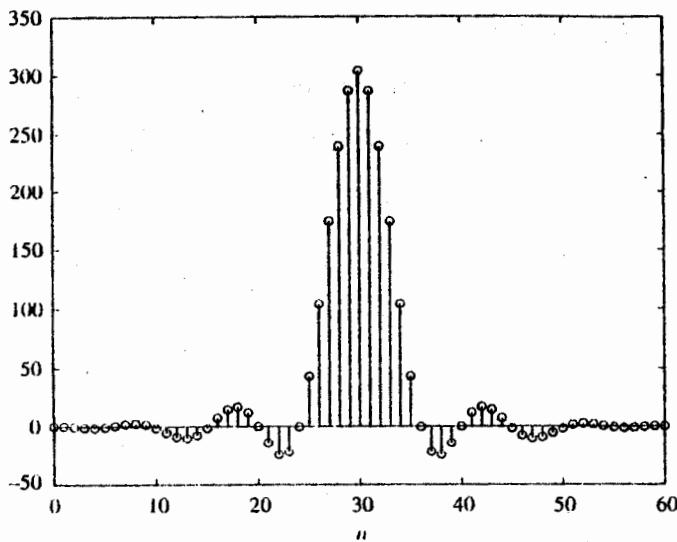


Hình 6.25 (a) Đáp ứng xung và (b) đáp ứng tần số của bộ lọc FIR nhị phân đồi rời rạc theo thời gian được cắt, tại máy phát

Hình 6.25(a) minh họa  $g_T\left(\frac{N-1}{2}\right)$ ,  $n=0, 1, \dots, N-1$  với  $N=31$ . Đặc tuyến đáp ứng tần số tương ứng được thể hiện trên hình 6.25(b). Hãy chú ý rằng đặc tuyến không triệt tiêu khi  $|f| \geq W$  do bộ lọc số có thời gian tồn tại đáp ứng xung hữu hạn.

Tuy nhiên, các búp phụ của phổ tương đối nhỏ. Cuối cùng, trên hình 6.26, chúng tôi thể hiện đáp ứng xung của mạch mắc nối tiếp các bộ lọc FIR phát và

thu này. Đáp ứng xung này có thể so sánh được với đáp ứng xung lý tưởng nhận được bằng cách lấy mẫu  $x(t)$  được cho bởi (6.5.17) với tốc độ  $F_s=4/T=8W$ .



Hình 6.26 Đáp ứng xung tương đương của mạch măc nối tiếp bộ lọc phát với bộ lọc phối hợp tại máy thu

### 6.5.3 TIỀN MÃ HOÁ ĐỐI VỚI TÁCH CÁC TÍN HIỆU ĐÁP ỦNG MỘT PHẦN

Đối với xung tín hiệu nhị phân đối,  $x(nT)=1$  khi  $n=0, 1$  và  $x(nT)=0$  với các giá trị khác của  $n$ . Do đó, các mẫu của lõi ra bộ lọc máy thu  $G_R(f)$  biểu diễn được theo

$$y_k = a_k + a_{k-1} + v_k = b_k + v_k \quad (6.5.22)$$

trong đó  $\{a_k\}$  là dãy các biên độ được truyền đi,  $\{v_k\}$  là một dãy các mẫu tạp âm cộng Gauss, còn  $b_k = a_k + a_{k-1}$ . Tạm thời chúng ta hãy bỏ qua không xét đến tạp âm và chỉ xét trường hợp truyền tin nhị phân, trong đó  $a_k = \pm 1$  với xác suất như nhau. Khi đó  $b_k$  nhận một trong ba giá trị có thể có, cụ thể là  $b_k = -2, 0, 2$  với các xác suất tương ứng là  $\frac{1}{4}, \frac{1}{2}, \frac{1}{4}$ . Nếu  $a_{k-1}$  là tín hiệu tách được từ khoảng thời gian truyền tín hiệu thứ  $(k-1)$  thì tác động của nó lên  $b_k$  – là tín hiệu thu được trong khoảng thời gian truyền tín hiệu thứ  $k$  – có thể khử được nhờ phép trừ, nhờ thế cho phép  $a_k$  tách được. Quá trình này có thể được lặp đi lặp lại tuần tự đối với từng *symbol* thu được.

Vấn đề chính đối với thủ tục này là ở chỗ các lõi nảy sinh do tạp âm cộng có

xu hướng lan truyền ra. Thí dụ, nếu  $a_{k-1}$  bị tách sai thì tác động của nó tới  $a_k$  không bị loại bỏ; thực tế, tác động này được cung cấp thêm bởi việc trừ không đúng. Hệ quả là, việc tách  $a_k$  cũng y như bị tách sai.

Hiện tượng lan sai có thể tránh được bằng cách mã hoá trước (tiền mã hoá) số liệu tại máy phát thay cho việc loại bỏ ISI kiểm soát được bằng cách trừ ở máy thu. Mã hoá trước được thực hiện trên dãy số liệu nhị phân trước khi điều chế. Một dãy mới  $\{p_k\}$ , được gọi là *dãy được tiền mã hoá*, được tạo ra từ dãy dữ liệu  $\{D_k\}$  gồm các con 1 và 0 cần truyền. Đối với tín hiệu nhị phân đối, dãy được tiền mã hoá được xác định theo

$$p_k = D_k - p_{k-1} \pmod{2}, \quad k = 1, 2, \dots \quad (6.5.23)$$

Cần nói thêm ở đây rằng, mặc dù phép trừ modulo 2 hoàn toàn đồng nhất với phép cộng modulo 2 song việc xét thuật toán mã hoá trước đối với tín hiệu nhị phân đối sẽ thuận tiện hơn với phép trừ modulo 2.

Khi độ biến đổi tín hiệu phát là  $a_k = -1$  nếu  $p_k = 0$  và  $a_k = 1$  nếu  $p_k = 1$ , tức là

$$a_k = 2p_k - 1 \quad (6.5.24)$$

Các mẫu không có tạp âm tại lối ra của bộ lọc thu được cho bởi

$$b_k = a_k + a_{k-1} = (2p_k - 1) + (2p_{k-1} - 1) = 2(p_k + p_{k-1} - 1) \quad (6.5.25)$$

Hệ quả là,

$$p_k + p_{k-1} = \frac{1}{2}b_k + 1 \quad (6.5.26)$$

Do  $D_k = p_k \oplus p_{k-1}$  nên dãy số liệu  $\{D_k\}$  có được từ  $\{b_k\}$  bằng cách sử dụng quan hệ

$$D_k = \frac{1}{2}b_k + 1 \pmod{2} \quad (6.5.27)$$

Vì vậy,  $D_k = 0$  nếu  $b_k = \pm 2$  và  $D_k = 1$  nếu  $b_k = 0$ . Một thí dụ minh họa các thuật toán tiền mã hoá và giải mã được nêu trong bảng 6.1.

Khi có tạp âm cộng, các lối ra được lấy mẫu từ bộ lọc thu được tính theo công thức (6.5.22). Trong trường hợp này,  $y_k = b_k + v_k$  được so sánh với hai mức ngưỡng được đặt là +1 và -1. Dãy dữ liệu  $\{D_k\}$  nhận được theo luật quyết định

$$D_k = \begin{cases} 1, & |y_k| < 1 \\ 0, & |y_k| \geq 1 \end{cases} \quad (6.5.28)$$

Như vậy, việc tiền mã hoá dữ liệu cho phép chúng ta thực hiện việc tách tín hiệu lần lượt từng *symbol* tại máy thu mà không cần phải trừ các *symbol* đã tách được trước đó.

Bảng 6.1. Truyền tín hiệu với các xung nhị phân đôi

Dãy dữ liệu $D_k$	-	1	1	1	0	1	0	0	1	0	0	0	1
Dãy tiền mã hoá $p_k$	0	1	0	1	1	0	0	0	1	1	1	1	0
Dãy phát $a_k$	-1	1	-1	1	1	-1	-1	-1	1	1	1	1	-1
Dãy thu $b_k$	-	0	0	0	2	0	-2	-2	0	2	2	2	0
Dãy đã giải mã $D_k$	-	1	1	1	0	1	0	0	1	0	0	0	1

Việc mở rộng từ PAM nhị phân cho PAM nhiều mức sử dụng các xung nhị phân đôi thật đơn giản. Dãy phát  $M$  mức  $\{a_k\}$  dẫn đến một dãy thu (không có tạp âm)

$$b_k = a_k + a_{k-1}, \quad k = 1, 2, \dots \quad (6.5.29)$$

mà nó có  $2M-1$  mức biên độ có thể có, phân cách đều nhau. Các mức biên độ đối với dãy  $\{a_k\}$  được xác định theo quan hệ

$$a_k = 2p_k - (M-1) \quad (6.5.30)$$

trong đó  $\{p_k\}$  là dãy tiền mã hoá nhận được từ một dãy dữ liệu  $M$  mức  $\{D_k\}$  theo quan hệ

$$p_k = D_k - p_{k-1} \pmod{M} \quad (6.5.31)$$

trong đó các giá trị có thể có của dãy dữ liệu  $\{D_k\}$  là  $0, \dots, M-1$ .

Trong trường hợp không có tạp âm, các mẫu tại lối ra bộ lọc thu có thể biểu diễn được theo

$$\begin{aligned} b_k &= a_k + a_{k-1} = [2p_k - (M-1)] + [2p_{k-1} - (M-1)] \\ &= 2[p_k + p_{k-1} - (M-1)] \end{aligned} \quad (6.5.32)$$

$$\text{Do đó: } p_k + p_{k-1} = \frac{1}{2}b_k + (M-1) \quad (6.5.33)$$

Do  $D_k = p_k + p_{k-1} \pmod{M}$ , điều này dẫn đến việc số liệu đã truyền  $\{D_k\}$  khôi phục lại được từ dãy thu được  $\{b_k\}$  theo quan hệ

$$D_k = \frac{1}{2} b_k + (M-1) \pmod{M} \quad (6.5.34)$$

Trong trường hợp xung nhị phân đôi cải biên, các mẫu tín hiệu thu được tại lối ra bộ lọc thu  $G_R(f)$  biểu diễn được theo

$$y_k = a_k - a_{k-2} + v_k = b_k + v_k \quad (6.5.35)$$

Bộ tiền mã hoá đổi với xung nhị phân đôi cải biên cho ra dãy  $\{p_k\}$  từ dãy số liệu  $\{D_k\}$  theo quan hệ

$$p_k = D_k \oplus p_{k-2} \pmod{M} \quad (6.5.36)$$

Từ các quan hệ này, dễ dàng chỉ ra rằng luật tách tín hiệu để khôi phục dãy dữ liệu  $\{D_k\}$  từ  $\{b_k\}$  khi không có tạp âm là

$$D_k = \frac{1}{2} b_k \pmod{M} \quad (6.5.37)$$

### BÀI TẬP MINH HOA

**Bài tập minh họa 6.9 [Tiền mã hoá tín hiệu nhị phân đôi]** Hãy viết một chương trình MATLAB mà nó nhận một dãy dữ liệu  $\{D_k\}$ , thực hiện tiền mã hoá nó cho một hệ thống truyền dẫn bằng xung nhị phân đôi để cho ra  $\{p_k\}$ , rồi ánh xạ dãy đã tiền mã hoá thành các mức biên độ phát  $\{a_k\}$ . Sau đó, từ dãy tín hiệu phát  $\{a_k\}$  hãy tạo ra dãy tín hiệu thu không có tạp âm  $\{b_k\}$  rồi bằng cách sử dụng quan hệ trong (6.5.34) hãy khôi phục lại dãy dữ liệu  $\{D_k\}$ .

### Lời giải

MATLAB script được cho dưới đây. Bằng cách sử dụng chương trình này, chúng ta có thể kiểm chứng các kết quả trong bảng 6.1 cho trường hợp  $M=2$ .

#### m-file

```
% MATLAB script for Illustrative Problem 9, Chapter 6.
echo on
d=[1 1 1 0 1 0 0 1 0 0 0 1];
p(1)=0;
```

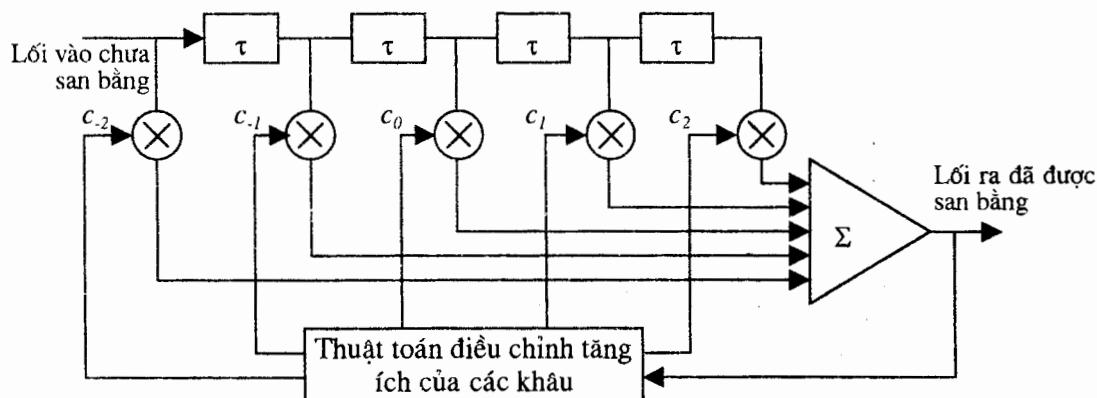
```

for i=1:length(d)
    p(i+1)=rem(p(i)+d(i),2);
    echo off ;
end
echo on ;
a=2.*p-1;
b(1)=0;
dd(1)=0;
for i=1:length(d)
    b(i+1)=a(i+1)+a(i);
    d_out(i+1)=rem(b(i+1)/2+1,2);
    echo off ;
end
echo on ;
d_out=d_out(2:length(d)+1);

```

## 6.6 CÁC BỘ SAN BẰNG TUYẾN TÍNH

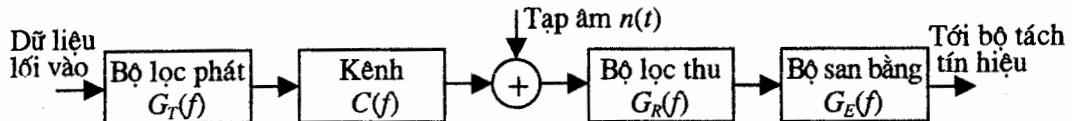
Kiểu chung nhất của bộ san bằng kênh được sử dụng trong thực tế để giảm ISI là một bộ lọc FIR tuyến tính với các hệ số có thể điều chỉnh được  $\{c_i\}$  như được thể hiện trên hình 6.27.



Hình 6.27. Bộ lọc dàn tuyến tính

Trên các kênh mà các đặc tính đáp ứng tần số của chúng không biết được song không biến đổi theo thời gian, chúng ta có thể đo đặc các đặc tính kênh và điều chỉnh các tham số của bộ san bằng; một khi đã được điều chỉnh, các tham số đó giữ nguyên cố định trong suốt thời gian truyền đưa số liệu. Các bộ san bằng như thế được gọi là các *bộ san bằng thiết lập trước* (*preset equalizer*). Trái lại, các *bộ san*

bằng thích nghi (*adaptive equalizer*) cập nhật các tham số của chúng một cách đều đặn trong quá trình truyền dẫn dữ liệu, do đó chúng có khả năng *bám* (*tracking*) theo một đáp ứng kênh biến đổi chậm theo thời gian.



Hình 6.28. Sơ đồ khối một hệ thống với một bộ san bằng

Trước tiên, chúng ta hãy xem xét các đặc tính thiết kế đối với một bộ san bằng tuyến tính xét trên miền tần số. Hình 6.28 trình bày một sơ đồ khối của một hệ thống sử dụng một bộ lọc tuyến tính làm bộ san bằng kênh.

Bộ giải điều chế bao gồm một bộ lọc thu có đáp ứng tần số  $G_R(f)$  nối tiếp với một bộ lọc san bằng kênh có đáp ứng tần số  $G_E(f)$ . Như đã trình bày trong phần trước, đáp ứng của bộ lọc thu phối hợp với đáp ứng của bộ lọc phát, nghĩa là  $G_R(f) = G_T^*(f)$ , và tích  $G_R(f).G_T(f)$  thường được thiết kế sao cho thành phần ISI tại các thời điểm lấy mẫu hoặc là bị triệt tiêu ví dụ như khi  $G_R(f).G_T(f) = X_{rc}(f)$ , hoặc là ISI kiểm soát được như đối với các tín hiệu đáp ứng một phần.

Đối với hệ thống được thể hiện trên hình 6.28, trong đó đáp ứng tần số của kênh không lí tưởng, điều kiện mong muốn để triệt tiêu ISI là

$$G_T(f).C(f).G_R(f).G_E(f) = X_{rc}(f) \quad (6.6.1)$$

trong đó  $X_{rc}(f)$  là đặc tính phổ cosin nâng mong muốn. Vì theo thiết kế,  $G_T(f).G_R(f)=X_{rc}(f)$  nên đáp ứng tần số của bộ san bằng bù méo kênh là

$$G_E(f) = \frac{1}{C(f)} = \frac{1}{|C(f)|} e^{-j\theta_c(f)} \quad (6.6.2)$$

Như vậy, đáp ứng biên độ của bộ san bằng là  $|G_E(f)| = \frac{1}{|C(f)|}$  và đáp ứng

pha của nó là  $\theta_E(f) = -\theta_c(f)$ . Trong trường hợp này, bộ san bằng được gọi là *bộ lọc kênh nghịch đảo* (*inverse channel filter*) đối với đáp ứng của kênh.

Chúng ta chú ý rằng bộ lọc kênh nghịch đảo loại bỏ hoàn toàn ISI do kênh gây ra. Do nó buộc ISI bằng không tại các thời điểm lấy mẫu  $t=kT$  ( $k=0, 1, \dots$ ), bộ san bằng được gọi là bộ *san bằng cưỡng ép không* (*zero-forcing equalizer*). Vì vậy, lối vào bộ tách tín hiệu đơn giản là

$$z_k = a_k + \eta_k, \quad k = 0, 1, \dots \quad (6.6.3)$$

trong đó  $\eta_k$  biểu thị tạp âm cộng còn  $a_k$  là *symbol* mong muốn.

Trên thực tế, ISI gây ra do méo kênh thường hạn chế chỉ trên một số hữu hạn *symbol* nằm cả hai bên *symbol* mong muốn. Do đó, số các số hạng hình thành nên ISI trong tổng được cho bởi (6.4.7) là hữu hạn. Như là một hệ quả, trên thực tế, bộ san bằng kênh được thực hiện như một bộ lọc có đáp ứng xung với thời gian tồn tại hữu hạn (FIR), hoặc *bộ lọc dàn* (*transversal filter*) với các hệ số khâu điều chỉnh được  $\{c_n\}$ , như được minh họa trên hình 6.27. Trễ thời gian  $\tau$  giữa các khâu nằm cạnh nhau có thể được lựa chọn lớn bằng với  $T$  là thời khoảng của một *symbol* mà trong trường hợp đó bộ san bằng FIR được gọi là *bộ san bằng trễ khâu từng symbol* (*symbol-spaced equalizer*). Trong trường hợp này, đầu vào bộ san bằng là dãy được lấy mẫu được cho bởi (6.4.6). Tuy nhiên, chúng ta cần chú ý rằng khi tốc độ *symbol*  $\frac{1}{T} < 2W$ , các tần số lớn hơn tần số gấp phỏ  $\frac{1}{T}$  trong tín hiệu thu bị thiên dịch về các tần số nhỏ hơn  $\frac{1}{T}$ . Trong trường hợp này, bộ san bằng bù cho tín hiệu bị méo kênh thiên dịch (*aliased channel-distorted signal*).

Trong trường hợp, khi trễ thời gian  $\tau$  giữa các khâu nằm cạnh nhau được lựa chọn sao cho  $\frac{1}{\tau} \geq 2W > \frac{1}{T}$ , sự thiên dịch không xảy ra; do đó, bộ san bằng kênh nghịch đảo thực hiện bù cho méo thực sự của kênh. Vì  $\tau < T$  nên bộ san bằng kênh được nói là có các khâu trễ một phần và nó được gọi là *bộ san bằng phân cách từng phần* (*fractionally spaced equalizer*). Thực tế,  $\tau$  thường được chọn là  $\tau = T/2$ . Hãy chú ý rằng trong trường hợp này tốc độ lấy mẫu tại lối vào bộ lọc  $G_E(f)$  là  $2/T$ .

Đáp ứng xung của bộ san bằng FIR là

$$g_E(t) = \sum_{n=-K}^K c_n \delta(t - n\tau) \quad (6.6.4)$$

và đáp ứng tần số tương ứng là

$$G_E(f) = \sum_{n=-K}^K c_n e^{-j2\pi f n\tau} \quad (6.6.5)$$

Trong đó  $\{c_n\}$  là  $2K+1$  hệ số bộ san bằng, còn  $K$  được chọn đủ lớn sao cho bộ san bằng bắc ngang độ dài của ISI – nghĩa là  $2K+1 \geq L$ , trong đó  $L$  là số các mẫu tín hiệu liên quan đến nhau bởi ISI. Vì  $X(f) = G_T(f)C(f)G_R(f)$  và  $x(t)$  là xung tín hiệu tương ứng với  $X(f)$  nên xung tín hiệu lõi ra đã được sửa (được san bằng méo) là

$$q(t) = \sum_{n=-K}^K c_n x(t - n\tau) \quad (6.6.6)$$

Bây giờ, điều kiện *cưỡng ép không* có thể áp dụng được cho các mẫu của  $q(t)$  lấy tại các thời điểm  $t = mT$ . Các mẫu này là

$$q(mT) = \sum_{n=-K}^K c_n x(mT - n\tau), \quad m = 0, \pm 1, \pm 2, \dots, \pm K \quad (6.6.7)$$

Do có cả thảy  $2K+1$  hệ số bộ san bằng, chúng ta chỉ có thể điều khiển  $2K+1$  giá trị đã được lấy mẫu của  $q(t)$ . Cụ thể, chúng ta ràng buộc các điều kiện

$$q(mT) = \sum_{n=-K}^K c_n x(mT - n\tau) = \begin{cases} 1, & m = 0 \\ 0, & m = \pm 1, \pm 2, \dots, \pm K \end{cases} \quad (6.6.8)$$

mà chúng có thể biểu diễn được dưới dạng ma trận theo  $Xc=q$ , trong đó  $X$  là một ma trận  $(2K+1) \times (2K+1)$  với các phần tử  $x(mT - n\tau)$ ,  $c$  là véc-tơ  $(2K+1)$  hệ số, còn  $q$  là véc-tơ cột  $(2K+1)$  chiều với một phần tử khác không. Như vậy, chúng ta nhận được một hệ gồm  $2K+1$  phương trình tuyến tính đối với các hệ số của bộ san bằng *cưỡng ép không*.

Chúng ta cần nhấn mạnh rằng bộ san bằng *cưỡng ép không* FIR không loại bỏ hoàn toàn ISI được vì nó có độ dài hữu hạn. Tuy nhiên, khi  $K$  tăng, phần ISI còn dư có thể giảm được, và tới giới hạn  $K \rightarrow \infty$ , ISI sẽ bị triệt tiêu hoàn toàn.

**BÀI TẬP MINH HỌA**

**Bài tập minh họa 6.10 [Thiết kế bộ san bằng]** Xét một xung chịu méo kênh là  $x(t)$  tại đầu vào bộ cân bằng, được cho bởi biểu thức

$$x(t) = \frac{1}{1 + (2t/T)^2}$$

với  $1/T$  là tốc độ symbol. Xung được lấy mẫu với tốc độ  $2/T$  và được sửa bằng một bộ san bằng cưỡng ép không. Hãy xác định các hệ số của bộ san bằng cưỡng ép không 5 khâu.

**Lời giải**

Theo (6.6.8), bộ san bằng cưỡng ép không phải thoả mãn phương trình

$$q(mT) = \sum_{n=-2}^2 c_n x\left(mT - \frac{nT}{2}\right) = \begin{cases} 1, & m = 0 \\ 0, & m = \pm 1, \pm 2 \end{cases}$$

Ma trận  $X$  với các phần tử  $x(mT - nT/2)$  tính được là

$$X = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ \frac{1}{5} & \frac{1}{10} & \frac{1}{17} & \frac{1}{26} & \frac{1}{37} \\ 1 & \frac{1}{2} & \frac{1}{5} & \frac{1}{10} & \frac{1}{17} \\ \frac{1}{5} & \frac{1}{2} & 1 & \frac{1}{2} & \frac{1}{5} \\ \frac{1}{17} & \frac{1}{10} & \frac{1}{5} & \frac{1}{2} & 1 \\ \frac{1}{37} & \frac{1}{26} & \frac{1}{17} & \frac{1}{10} & \frac{1}{5} \end{bmatrix} \quad (6.6.9)$$

Véc-tơ hệ số  $c$  và véc-tơ  $q$  được cho như sau

$$c = \begin{bmatrix} c_{-2} \\ c_{-1} \\ c_0 \\ c_1 \\ c_2 \end{bmatrix}; q = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad (6.6.10)$$

Khi đó các phương trình  $Xc=q$  có thể giải được bằng cách nghịch đảo ma trận  $X$ . Như thế, chúng ta nhận được

$$\mathbf{C}_{\text{opt}} = \mathbf{X}^{-1} \mathbf{q} = \begin{bmatrix} -2,2 \\ 4,9 \\ -3 \\ 4,9 \\ -2,2 \end{bmatrix} \quad (6.6.11)$$

Hình 6.29 minh họa xung  $x(t)$  ban đầu và xung đã được sửa. Hãy để ý đến lượng ISI nhỏ còn dư trong xung đã được sửa.

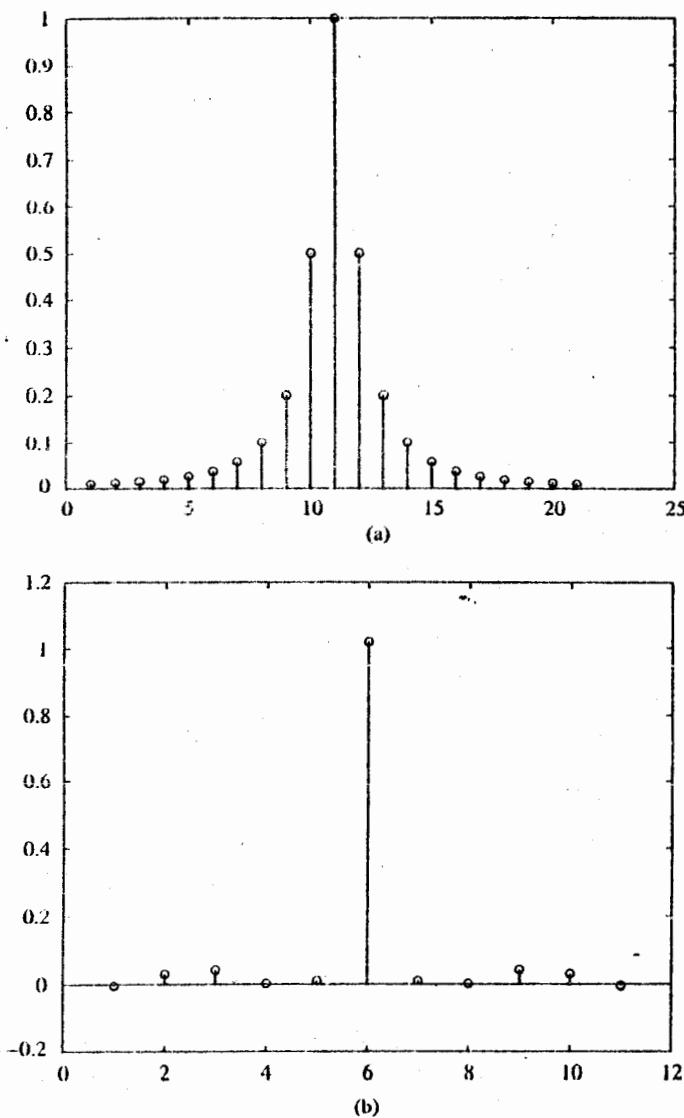
MATLAB script cho tính toán này được cho dưới đây.

### m-file

```
% MATLAB script for Illustrative Problem 10, Chapter 6.
echo on
T=1;
Fs=2/T;
Ts=1/Fs;
c_opt=[-2.2 4.9 -3 4.9 -2.2];
t=-5*T:T/2:5*T;
x=1./(1+((2/T)*t).^2); % sampled pulse
equalized_x=filter(c_opt,1,[x 0 0]); % since there will be a delay of two
samples at the output
% to take care of the delay
equalized_x=equalized_x(3:length(equalized_x));
% Now, let us downsample the equalizer output
for i=1:2:length(equalized_x),
    downsampled_equalizer_output((i+1)/2)=equalized_x(i);
    echo off;
end;
echo on ;
% plotting commands follow
```

Một mặt hạn chế đối với bộ san bằng cường ép không là ở chỗ nó bỏ qua không tính đến sự có mặt của tạp âm cộng. Như một hệ quả, việc sử dụng của nó có thể tăng cường đáng kể tạp âm. Có thể dễ dàng nhận thấy điều đó khi để ý rằng trong một dải tần số trong đó  $C(f)$  nhỏ, bộ san bằng kênh  $G_E(f) = 1/C(f)$  sẽ bù bằng cách có một tăng ích lớn trong dải tần số này. Kết quả là, tạp âm trong dải tần số này được tăng mạnh. Một cách khác có thể áp dụng là giảm nhẹ điều kiện

ISI bằng không và chọn đặc tính bộ san bằng sao cho công suất hỗn hợp của ISI dư và tạp âm cộng tại lõi ra của bộ san bằng là nhỏ nhất. Một bộ san bằng được tối ưu hoá dựa trên tiêu chí sai số bình phương trung bình cực tiểu (MMSE: *Minimum Mean-Square Error*) sẽ đạt được mục đích này.



Hình 6.29 Các đồ thị của (a) xung ban đầu và (b) xung đã sửa trong bài tập minh họa 6.10

Một cách chi tiết, ta hãy xét tín hiệu lõi ra bị nhiễu loạn bởi tạp âm của bộ san bằng FIR, biểu diễn được theo

$$z(t) = \sum_{n=-K}^K c_n y(t - n\tau) \quad (6.6.12)$$

với  $y(t)$  là tín hiệu lối vào bộ san bằng, được cho bởi (6.4.3). Lối ra bộ san bằng được lấy mẫu tại các thời điểm  $t=mT$ . Như thế, chúng ta nhận được

$$z(mT) = \sum_{n=-K}^K c_n y(mT - n\tau) \quad (6.6.13)$$

Đáp ứng mong muốn tại lối ra bộ san bằng tại  $t=mT$  là *symbol* đã được truyền đi  $a_m$ . Sai số được định nghĩa là độ chênh lệch giữa  $a_m$  và  $z(mT)$ . Khi đó sai số bình phương trung bình giữa mẫu thực tế lối ra  $z(mT)$  và các giá trị mong muốn  $a_m$  là như sau (trong phát triển tính toán này, chúng tôi để cho các tín hiệu  $z(t)$  và  $y(t)$  là các tín hiệu phức và dãy dữ liệu cũng là phức):

$$\begin{aligned} \text{MSE} &= E|z(mT) - a_m|^2 = E\left[\left|\sum_{n=-K}^K c_n y(mT - n\tau) - a_m\right|^2\right] \\ &= \sum_{n=-K}^K \sum_{k=-K}^K c_n c_k R_y(n-k) - 2 \sum_{k=-K}^K c_k R_{ay}(k) + E(|a_m|^2) \end{aligned} \quad (6.6.14)$$

trong đó các hàm tương quan được xác định theo

$$R_y(n-k) = E[y^*(mT - n\tau)y(mT - k\tau)], \quad R_{ay}(k) = E[y(mT - k\tau)a_m^*] \quad (6.6.15)$$

và giá trị kỳ vọng được lấy theo dãy thông tin ngẫu nhiên  $\{a_m\}$  và tạp âm cộng.

Lời giải MSE cực tiểu nhận được bằng cách lấy vi phân (6.6.14) theo các hệ số bộ san bằng  $\{c_n\}$ . Như thế, chúng ta nhận được các điều kiện cần để có MSE cực tiểu là

$$\sum_{n=-K}^K c_n R_y(n-k) = R_{ay}(k), \quad k = 0, \pm 1, \pm 2, \dots, \pm K \quad (6.6.16)$$

Đây là  $2K+1$  phương trình tuyến tính đối với các hệ số của bộ san bằng. Trái với giải pháp cưỡng ép không đã được trình bày trên đây, các phương trình này phụ thuộc vào các thuộc tính thống kê (hàm tự tương quan) của tạp âm cũng như ISI thông qua hàm tự tương quan  $R_y(n)$ .

Trên thực tế, ma trận tự tương quan  $R_y(n)$  và véc-tơ tương quan chéo  $R_{ay}(n)$  không

biết trước được. Tuy nhiên, các dãy tương quan này có thể được ước lượng bằng cách truyền một tín hiệu thử qua kênh và sử dụng các ước lượng trung bình theo thời gian

$$\hat{R}_y(n) = \frac{1}{K} \sum_{k=1}^K y^*(kT - n\tau) y(kT)$$

$$R_{ay}(n) = \frac{1}{K} \sum_{k=1}^K y(kT - n\tau) a_k^* \quad (6.6.17)$$

thay cho các trị trung bình toàn thể để giải tìm các hệ số của bộ san bằng cho bởi (6.6.16).

### BÀI TẬP MINH HOA

**Bài tập minh họa 6.11 [Thiết kế bộ cân bằng]** Xét xung chịu méo kênh  $x(t)$  như trong bài tập minh họa 6.10, song bây giờ hãy thiết kế bộ san bằng 5 khâu dựa trên tiêu chí MSE cực tiểu. Các *symbol* thông tin có kì vọng bằng 0, phương sai bằng 1 và không tương quan, tức là  $E(a_n) = 0$ ,  $E(a_n a_m) = 0$  ( $n \neq m$ ),  $E(|a_n|^2) = 1$ .

Tạp âm cộng  $v(t)$  có kì vọng 0 và hàm tự tương quan

$$\varphi_{vv}(\tau) = \frac{N_0}{2} \delta(\tau)$$

#### Lời giải

Các hệ số khâu của bộ san bằng nhận được bằng cách giải (6.6.16) với  $K=2$  và  $\tau=T/2$ . Ma trận với các phần tử  $R_y(n-k)$  đơn giản là

$$R_y = X^t X + \frac{N_0}{2} I$$

trong đó  $X$  được cho bởi (6.6.9) còn  $I$  là ma trận đơn vị. Véc-tơ chứa các

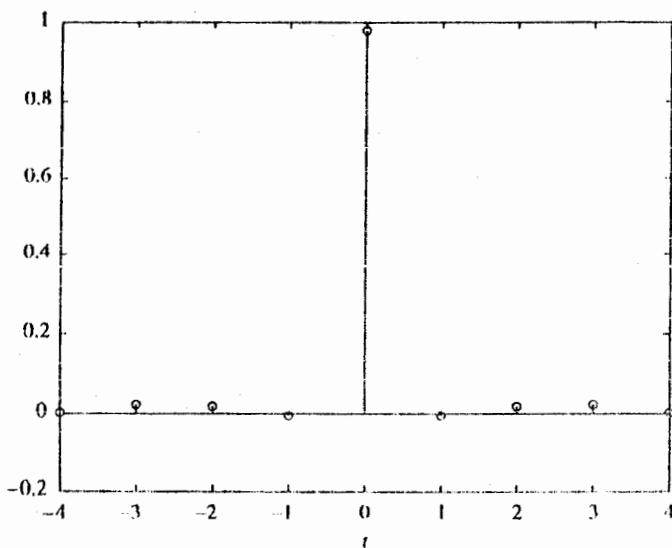
phần tử  $R_{ay}(k)$  được cho theo:

$$R_{ay} = \begin{bmatrix} 1 \\ \frac{1}{5} \\ 5 \\ \frac{1}{2} \\ 2 \\ 1 \\ \frac{1}{2} \\ 2 \\ \frac{1}{5} \end{bmatrix}$$

Các hệ số bộ san bằng thu được bằng cách giải (6.6.16) là như sau

$$c_{\text{opt}} = \begin{bmatrix} 0,0956 \\ -0,7347 \\ 1,6761 \\ -0,7347 \\ 0,0956 \end{bmatrix}$$

Đồ thị xung đã được sửa được thể hiện trên hình 6.30.



Hình 6.30 Dạng xung sau khi được sửa trong bài tập minh họa 6.11

MATLAB *script* cho tính toán này được cho dưới đây.

### m-file

```
% MATLAB script for Illustrative Problem 11, Chapter 6.
echo on
T=1;
for n=-2:2,
    for k=-2:2,
        temp=0;
        for i=-2:2, temp=temp+(1/(1+(n-i)^2))*(1/(1+(k-i)^2));
        end;
        X(k+3,n+3)=temp;
        echo off ;
    end;
end;
```

```

echo on;
N0=0.01; % assuming that N0=0.01
Ry=X+(N0/2)*eye(5);
Riy=[1/5 1/2 1 1/2 1/5].';
c_opt=inv(Ry)*Riy; % optimal tap coefficients
% find the equalized pulse...
t=-3:1/2:3;
x=1./(1+(2*t/T).^2); % sampled pulse
equalized_pulse=conv(x,c_opt);
% decimate the pulse to get the samples at the symbol rate
decimated_equalized_pulse=equalized_pulse(1:2:length(equalized_pulse));
% plotting command follows

```

### 6.6.1 CÁC BỘ SAN BẰNG TUYẾN TÍNH THÍCH NGHI

Chúng tôi đã cho thấy rằng các hệ số khâu của một bộ san bằng tuyến tính có thể xác định được bằng cách giải một hệ các phương trình tuyến tính. Trong tiêu chuẩn tối ưu cưỡng ép không, các phương trình tuyến tính được cho bởi (6.6.8). Mặt khác, nếu tiêu chuẩn tối ưu lại dựa trên việc cực tiểu hóa MSE thì các hệ số bộ san bằng tối ưu được xác định bằng cách giải hệ các phương trình tuyến tính được cho bởi (6.6.16).

Trong cả hai trường hợp, chúng ta có thể biểu diễn hệ các phương trình tuyến tính dưới dạng ma trận

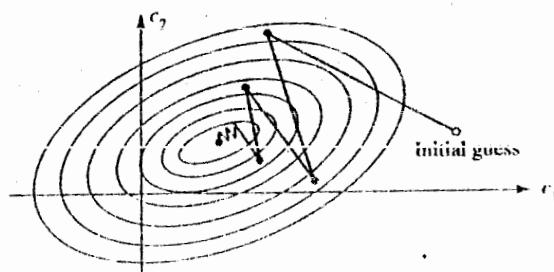
$$Bc = d \quad (6.6.18)$$

với  $B$  là một ma trận  $(2K+1) \times (2K+1)$ ,  $c$  là một véc-tơ cột biểu diễn  $(2K+1)$  hệ số của bộ san bằng, còn  $d$  là một véc-tơ cột  $(2K+1)$  chiều. Việc giải (6.6.18) cho chúng ta

$$c_{\text{opt}} = B^{-1}d \quad (6.6.19)$$

Trong thực tế thực hiện các bộ san bằng, lời giải của (6.6.18) đối với véc-tơ hệ số tối ưu thường nhận được bằng một thủ tục lặp mà nó tránh được việc tính toán nghịch đảo của ma trận  $B$  một cách tường minh. Thủ tục lặp đơn giản nhất là phương pháp giảm với độ dốc lớn nhất (giảm dốc nhất) (*steepest descent*), trong đó có thể bắt đầu bằng cách chọn tùy ý một véc-tơ hệ số  $c$ , cho là  $c_0$ . Lựa chọn

ban đầu này của véc-tơ hệ số  $c_0$  tương ứng với một điểm trên hàm tiêu chí đang được tối ưu hoá. Thí dụ, trong trường hợp tiêu chuẩn MSE, sự phỏng đoán ban đầu  $c_0$  tương ứng với một điểm trên mặt cong MSE bậc hai trong không gian  $(2K+1)$  chiều của các hệ số. Véc-tơ *gradient*, được ký hiệu là  $g_0$ , là đạo hàm của MSE theo  $2K+1$  hệ số của bộ lọc, khi đó được tính tại điểm này trên mặt cong tiêu chí và mỗi một hệ số khâu được thay đổi theo chiều hướng ngược với thành phần *gradient* tương ứng của nó. Thay đổi của hệ số khâu thứ  $j$  thì tỷ lệ với độ lớn của thành phần *gradient* thứ  $j$ .



Hình 6.31. Ví dụ về các đặc tính hội tụ của một thuật toán *gradient*

Thí dụ đối với bộ tiêu chuẩn MSE thì véc-tơ *gradient*, ký hiệu là  $g_0$  tìm được bằng cách lấy đạo hàm (riêng) của MSE theo từng hệ số trong  $2K+1$  hệ số, sẽ là

$$g_k = Bc_k - d, \quad k = 0, 1, 2, \dots \quad (6.6.20)$$

Sau đó véc-tơ hệ số  $c_k$  được cập nhật theo quan hệ

$$c_{k+1} = c_k - \Delta \cdot g_k \quad (6.6.21)$$

trong đó  $\Delta$  là *tham số cỡ bước* đối với thủ tục lặp. Để đảm bảo sự hội tụ của thủ tục lặp,  $\Delta$  phải được lựa chọn là một số dương đủ nhỏ. Trong trường hợp này, véc-tơ *gradient*  $g_k$  hội tụ về không, nghĩa là  $g_k \rightarrow 0$  khi  $k \rightarrow \infty$ , và véc-tơ hệ số  $c_k \rightarrow c_{opt}$ , như được minh họa trên hình 6.31 dựa trên một tối ưu hóa hai chiều. Một cách tổng quát, sự hội tụ của các hệ số khâu bộ san bằng đến  $c_{opt}$  không thể đạt được sau một số hữu hạn bước lặp với phương pháp giảm dốc nhất. Tuy nhiên nghiệm tối ưu  $c_{opt}$  có thể đạt đến được như mong muốn sau khoảng vài trăm bước lặp. Trong các hệ thống

thông tin số có sử dụng các bộ san bằng kênh, mỗi bước lặp tương đương với khoảng thời gian để truyền một *symbol*, do đó một vài trăm bước lặp để đạt được sự hội tụ đến  $c_{opt}$  tương ứng với thời gian chỉ bằng một phần của giây.

Việc san bằng kênh thích nghi là cần thiết đối với các kênh mà các đặc tính của chúng thay đổi theo thời gian. Trong trường hợp đó, ISI biến đổi theo thời gian. Bộ san bằng kênh nhất thiết phải bám theo các biến thiên theo thời gian như thế của đáp ứng kênh và điều chỉnh để các hệ số của nó một cách thích ứng nhằm làm giảm ISI. Trong phạm vi vấn đề đã được bàn trên đây, véc-tơ hệ số tối ưu  $c_{opt}$  thay đổi theo thời gian do những biến thiên theo thời gian trong ma trận  $B$ , còn đối với trường hợp tiêu chuẩn MSE, thì do các biến thiên theo thời gian trong véc-tơ  $d$ . Với các điều kiện này, phương pháp lặp nói trên có thể được cải biến để sử dụng các ước lượng của các thành phần gradient. Như thế, thuật toán điều chỉnh các hệ số khâu bộ san bằng có thể biểu diễn được theo

$$\hat{c}_{k+1} = \hat{c}_k - \Delta \hat{g}_k \quad (6.6.22)$$

$\hat{g}_k$  ký hiệu cho một ước lượng của véc-tơ *gradient*  $g_k$ , còn  $\hat{c}_k$  ký hiệu cho ước lượng của véc-tơ hệ số khâu  $c_k$ .

Trong trường hợp tiêu chuẩn MSE, véc-tơ *gradient*  $g_k$  cho bởi (6.6.20) có thể biểu diễn được theo

$$g_k = -E(e_k y_k^*)$$

Một ước lượng  $\hat{g}_k$  của véc-tơ *gradient* tại bước lặp thứ  $k$  tính được theo

$$\hat{g}_k = -e_k y_k^* \quad (6.6.23)$$

trong đó  $e_k$  biểu thị chênh lệch giữa đầu ra mong muốn của bộ san bằng tại thời điểm thứ  $k$  và giá trị lối ra thực tế  $z(kT)$ , còn  $y_k$  ký hiệu cho véc-tơ cột gồm  $2K+1$  giá trị tín hiệu thu được đang chứa trong bộ san bằng tại thời điểm thứ  $k$ .

Tín hiệu lõi  $e_k$  biểu diễn được theo

$$e_k = a_k - z_k \quad (6.6.24)$$

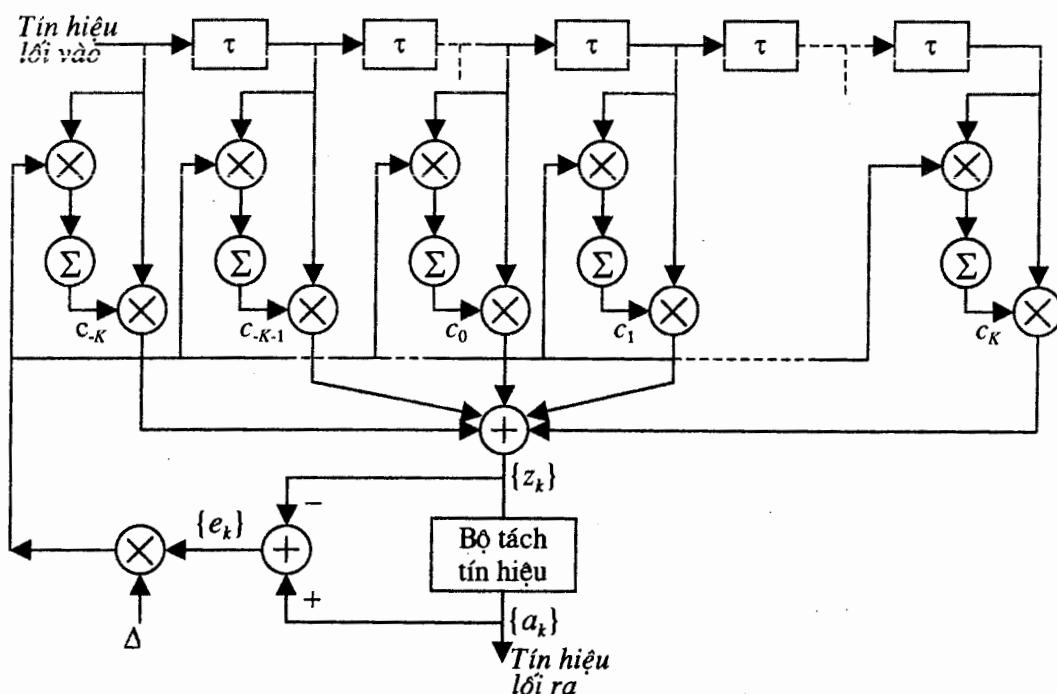
trong đó  $z_k = z(kT)$  là lối ra bộ san bằng được cho bởi (6.6.13), còn  $a_k$  là *symbol*

mong muốn. Do đó, bằng cách thay (6.6.23) vào (6.6.22), chúng ta có được thuật toán thích nghi để tối ưu hóa các hệ số các khâu (dựa trên tiêu chuẩn MSE) như sau

$$\hat{c}_{k+1} = \hat{c}_k + \Delta e_k y_k^* \quad (6.6.25)$$

Do ước lượng của véc-tơ gradient được sử dụng trong (6.6.25), thuật toán trên được gọi là *thuật toán gradient ngẫu nhiên (stochastic gradient algorithm)*. Nó cũng còn được biết đến với tên gọi *thuật toán LMS (Least Mean Square)*.

Một sơ đồ khái của một bộ san bằng thích nghi, mà nó biến đổi thích nghi các hệ số khâu theo (6.6.25), được minh họa trên hình 6.32. Cân lưu ý rằng sai lệch giữa lối ra mong muốn  $a_k$  và lối ra thực  $z_k$  từ bộ san bằng được sử dụng để tạo nên tín hiệu lỗi  $e_k$ . Lỗi này được phóng tỷ lệ bằng tham số cỡ bước  $\Delta$ , và tín hiệu lỗi đã đổi tỷ lệ  $\Delta e_k$  được nhân với các giá trị tín hiệu thu được  $\{y(kT-n\tau)\}$  tại  $2K+1$  khâu. Các tích  $\Delta e_k y^*(kT-n\tau)$  tại  $2K+1$  khâu sau đó được cộng với giá trị trước đó của các hệ số khâu để có được các hệ số nhánh mới, theo như (6.6.25). Tính toán như trên được lặp lại khi thu được mỗi mẫu tín hiệu mới. Nhờ đó, các hệ số bộ san bằng được cập nhật (làm mới lại) với tốc độ bằng tốc độ symbol.



Hình 6.32. Bộ san bằng thích nghi tuyến tính dựa trên tiêu chuẩn MSE

Khởi đầu, bộ san bằng thích nghi được huấn luyện bằng việc truyền dẫn một dãy giả ngẫu nhiên biết trước  $\{a_m\}$  qua kênh. Tại bộ giải điều chế, bộ san bằng sử dụng dãy đã biết để điều chỉnh các hệ số của mình. Với điều chỉnh khởi đầu này, bộ san bằng thích nghi chuyển từ *phương thức tập huấn (training mode)* sang *phương thức điều khiển theo quyết định (decision-directed mode)*, trong trường hợp đó các quyết định tại đầu ra bộ tách tín hiệu là đủ tin cậy sao cho tín hiệu lỗi tạo được bằng cách tính sai lệch giữa lỗi ra bộ tách tín hiệu và lỗi ra bộ san bằng, tức là

$$e_k = \hat{a}_k - z_k \quad (6.6.26)$$

trong đó  $\hat{a}_k$  là lỗi ra bộ tách tín hiệu. Nói chung, các lỗi quyết định tại lỗi ra bộ tách tín hiệu xảy ra không thường xuyên; hệ quả là, các lỗi này có ảnh hưởng nhỏ tới việc thực hiện thuật toán bám cho bởi (6.6.25).

Kinh nghiệm chọn tham số cỡ bước nhằm đảm bảo các khả năng hội tụ và bám tốt trong các kênh biến đổi chậm là

$$\Delta = \frac{1}{5(2K+1)P_R} \quad (6.6.27)$$

trong đó  $P_R$  ký hiệu cho công suất tín hiệu thu cộng với tạp âm mà nó có thể ước lượng được từ tín hiệu thu được.

### BÀI TẬP MINH HOẠ

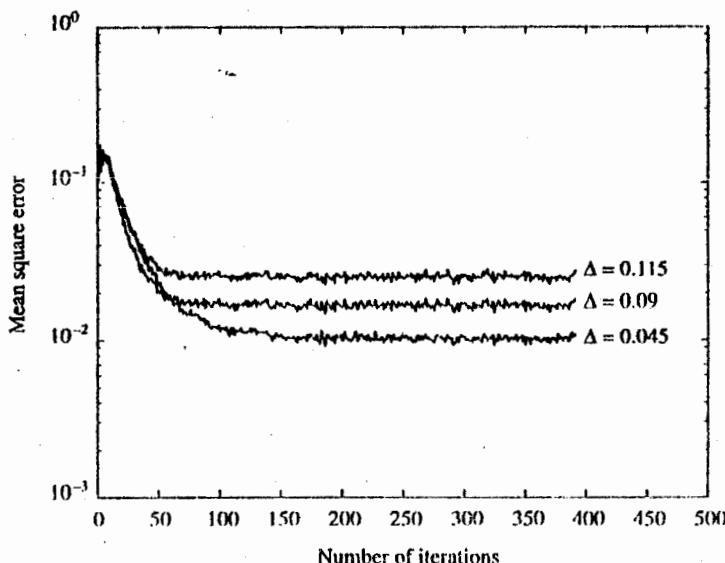
**Bài tập minh họa 6.12 [Bộ san bằng thích nghi]** Hãy thực hiện một bộ san bằng thích nghi dựa trên thuật toán LMS được cho trong (6.6.25). Số khâu của kênh được chọn cho bộ san bằng là  $2K+1=11$ . Công suất tín hiệu thu cộng với tạp âm  $P_R$  được chuẩn hoá bằng 1. Đặc tuyến của kênh được cho bởi véc-tơ  $x$  như sau

$$x=[0.05 \ -0.063 \ 0.088 \ -0.126 \ -0.25 \ 0.9047 \ 0.25 \ 0 \ 0.126 \ 0.038 \ 0.088]$$

### Lời giải

Sự hội tụ của thuật toán *gradient* ngẫu nhiên trong (6.6.25) được minh họa trên hình 6.33. Các đồ thị này đã nhận được từ mô phỏng máy tính một bộ san bằng thích nghi 11 khâu. Các đồ thị biểu thị lỗi trung bình bình *phương* được tính

trung bình trên một số lần thực hiện. Như được chỉ ra trên đồ thị, khi  $\Delta$  giảm, sự hội tụ chậm hơn đôi chút song đạt được một MSE thấp hơn, cho thấy rằng các hệ số ước lượng được gần với  $c_{opt}$  hơn.



Hình 6.33. Các đặc tính hội tụ ban đầu của thuật toán LMS với các cỡ bước khác nhau

MATLAB script đối với thí dụ này được cho dưới đây.

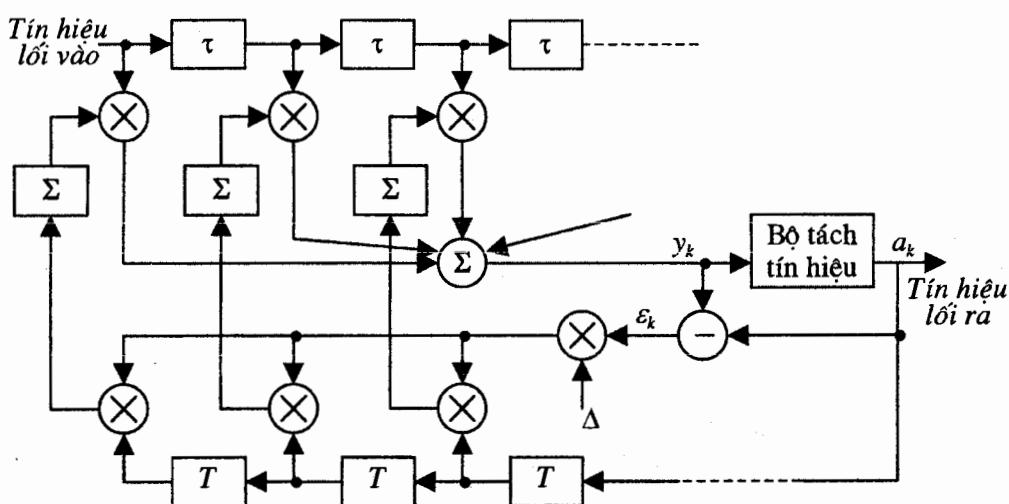
#### m-file

```
% MATLAB script for Illustrative Problem 12, Chapter 6.
echo on
N=500; % length of the information sequence
K=5;
actual_isi=[0.05 -0.063 0.088 -0.126 -0.25 0.9047 0.25 0 ... 0.126 0.038 0.088];
sigma=0.01;
delta=0.115;
Num_of_realizations=1000;
mse_av=zeros(1,N-2*K);
for j=1:Num_of_realizations, % compute the average over a number of realizations
    % the information sequence
    for i=1:N,
        if (rand<0.5),
            info(i)=-1;
        else
            info(i)=1;
        end;
    end;
    % the channel
    for i=1:N-K+1,
        r(i)=0;
        for j=i:i+K-1
            r(i)=r(i)+actual_isi(j)*x(j);
        end;
        r(i)=r(i)+n(i);
    end;
    % the estimate
    for i=1:N-K+1,
        if i==1
            e(i)=r(i)-y(i);
            x(i)=x(i)+delta*e(i);
        else
            e(i)=r(i)-y(i);
            x(i)=x(i)+delta*(e(i)-e(i-1));
        end;
    end;
    % the error
    for i=1:N-K+1,
        mse_av(j)=mse_av(j)+(y(i)-x(i))^2;
    end;
end;
mse_av=mse_av/Num_of_realizations;
```

```

echo off ;
end;
if (j==1) ; echo on ; end
% the channel output
y=filter(actual_isi,1,info);
for i=1:2:N, [noise(i) noise(i+1)]=gngauss(sigma); end;
y=y+noise;
% now the equalization part follows
estimated_c=[0 0 0 0 0 1 0 0 0 0 0]; % initial % estimate of ISI
for k=1:N-2*K,
y_k=y(k:k+2*K);
z_k=estimated_c*y_k.';
e_k=info(k)-z_k;
estimated_c=estimated_c+delta*e_k*y_k;
mse(k)=e_k^2;
echo off ;
end;
if (j==1) ; echo on ; end
mse_av=mse_av+mse;
echo off ;
end;
echo on ;
mse_av=mse_av/Num_of_realizations; % mean squared error
% versus iterations
% plotting commands follow

```

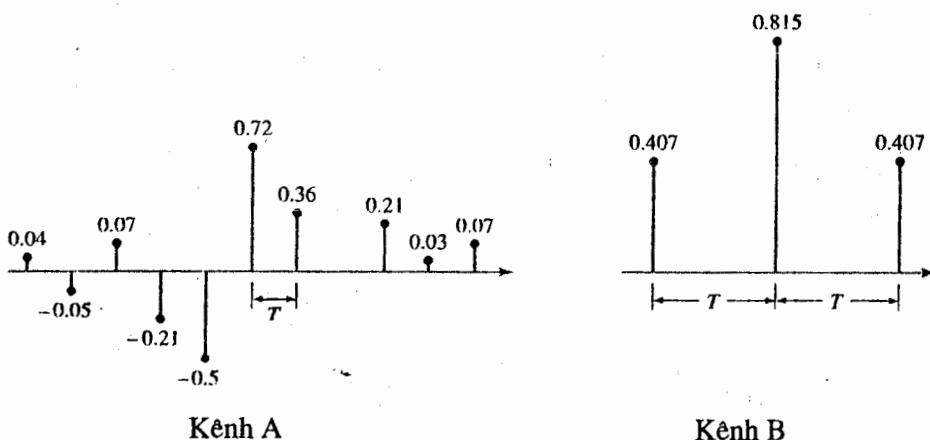


Hình 6.34 Một bộ san bằng thích nghi cưỡng ép không

Mặc dù chúng ta đã mô tả tương đối chi tiết hoạt động của một bộ san bằng thích nghi trên cơ sở tiêu chuẩn MSE, hoạt động của một bộ san bằng thích nghi dựa trên phương pháp cưỡng ép không thì cũng tương tự như vậy. Điểm khác nhau chính là ở phương pháp ước lượng các véc-tơ *gradient*  $g_k$  tại mỗi bước lặp. Một sơ đồ khái của một bộ san bằng thích nghi cưỡng ép không được thể hiện trên hình 6.34.

## 6.7 BỘ SAN BẰNG PHI TUYẾN

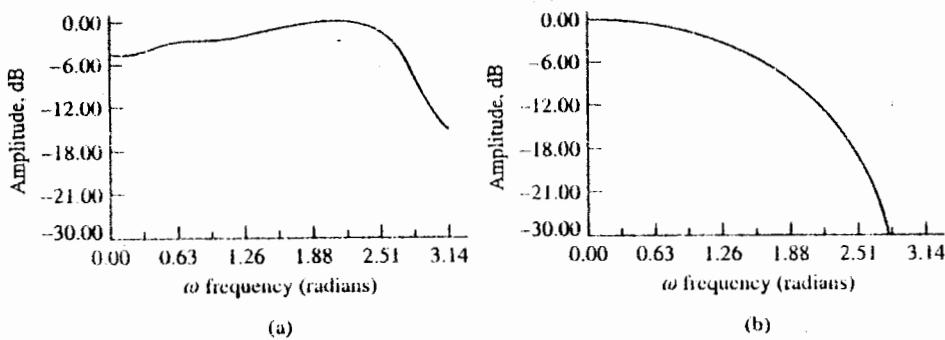
Các bộ san bằng tuyến tính đã được mô tả ở trên rất hiệu quả trên các kênh như thế các kênh điện thoại hữu tuyến, trong đó ISI không quá trầm trọng. Mức độ nghiêm trọng của ISI liên quan trực tiếp với các đặc tính phổ của kênh và không nhất thiết liên quan tới sai thời gian của ISI. Thí dụ, hãy xem xét ISI gây ra từ hai kênh được minh họa trên hình 6.35. Sai thời gian đối với ISI trong kênh A là năm khoảng *symbol* về mỗi bên của thành phần tín hiệu mong muốn mà nó có giá trị là 0.72. Trong khi đó sai thời gian đối với ISI trong kênh B là một khoảng *symbol* về mỗi bên của thành phần tín hiệu mong muốn, mà nó có giá trị là 0.815. Năng lượng của đáp ứng xung tổng cộng được chuẩn hoá bằng một đối với cả hai kênh.



Hình 6.35 Hai kênh với ISI

Mặc dù sai ISI ngắn hơn, kênh B gây nên ISI nghiêm trọng hơn. Điều này thấy rõ được trên các đặc tính đáp tuyến tần số của các kênh này mà chúng được

vẽ trên hình 6.36. Chúng ta thấy rằng kênh B có một trị bằng không của phổ [đáp tuyến tần số  $C(f)=0$  đối với một số tần số trong dải  $|f| \leq W$ ] tại tần số  $f=1/2T$ , trong khi đó điều này không xảy ra trong trường hợp kênh A. Hệ quả là, một bộ san bằng tuyến tính sẽ đưa vào đáp ứng tần số của nó một tăng ích lớn nhằm bù lại trị bằng không của đáp tuyến tần số kênh. Như thế, tạp âm trong kênh B sẽ được tăng cường mạnh hơn nhiều so với kênh A. Điều này hàm nghĩa rằng chất lượng của bộ san bằng tuyến tính đối với kênh B sẽ đủ tồi hơn so với chất lượng của kênh A. Nói chung, hạn chế cơ bản của một bộ san bằng tuyến tính là ở chỗ nó hoạt động tồi trên các kênh có các điểm phổ bằng không. Các kênh như thế thường gặp phải trong thông tin vô tuyến chẳng hạn như truyền dẫn tầng điện ly với các tần số dưới 30 MHz hay các kênh vô tuyến di động như các kênh được sử dụng đối với thông tin vô tuyến tần số.



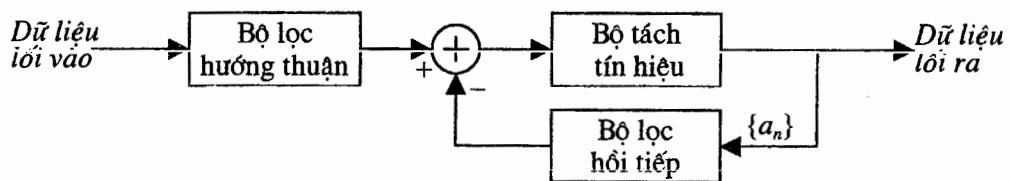
Hình 6.36 Các phổ biên độ đối với (a) kênh A và (b) kênh B từ hình vẽ 6.35

Một bộ san bằng hồi tiếp quyết định (DFE: Decision-Feedback Equalizer) là một bộ san bằng phi tuyến mà nó sử dụng các quyết định trước đó để loại bỏ ISI gây bởi các *symbol* đã được tách trước đó ảnh hưởng tới *symbol* hiện đang được tách. Một sơ đồ khái niệm đơn giản đối với một DFE được thể hiện trên hình 6.37. DFE bao gồm hai bộ lọc. Bộ lọc thứ nhất được gọi là *bộ lọc hướng thuận* (*feedforward filter*); nó nhìn chung là một bộ lọc FIR phân cách một phần với các hệ số khâu có thể điều chỉnh được. Bộ lọc này hoàn toàn giống về dạng với bộ san bằng tuyến tính đã được mô tả trước đây. Lối vào của nó là tín hiệu thu được đã qua lọc  $y(t)$  được lấy mẫu với

một tốc độ nào đó là một bội số của tốc độ symbol, thí dụ như với tốc độ  $2/T$ . Bộ lọc thứ hai là *bộ lọc hồi tiếp (feedback filter)*. Nó được thực hiện như một bộ lọc FIR với các khâu phân cách từng symbol có các hệ số khâu cũng có khả năng điều chỉnh được. Lối vào của nó là tập các symbol đã tách được trước. Lối ra của bộ lọc hồi tiếp được đem để trừ đi khỏi lối ra của bộ lọc hướng thuận để tạo nên lối vào bộ tách tín hiệu. Như thế, chúng ta có

$$z_m = \sum_{n=1}^{N_1} c_n y(mT - n\tau) - \sum_{n=1}^{N_2} b_n \tilde{a}_{m-n}$$

trong đó  $\{c_n\}$  và  $\{b_n\}$  lần lượt là các hệ số điều chỉnh được của các bộ lọc hướng thuận và bộ lọc hồi tiếp;  $\tilde{a}_{m-n}$  là các *symbol* đã tách được trước đó,  $n=1, 2, \dots, N_2$ ;  $N_1$  là độ dài của bộ lọc hướng thuận còn  $N_2$  là độ dài của bộ lọc hồi tiếp. Dựa trên lối vào  $z_m$ , bộ tách tín hiệu xác định xem *symbol* nào (trong số các *symbol* có thể có sẵn truyền) là gần nhất về khoảng cách đối với tín hiệu lối vào  $a_m$ . Theo đó, nó thực hiện việc quyết định của mình và cho ra  $\tilde{a}_m$ . Cái làm cho DFE trở nên phi tuyến là đặc tính phi tuyến của bộ tách tín hiệu tạo ra tín hiệu lối vào cho bộ lọc hồi tiếp.

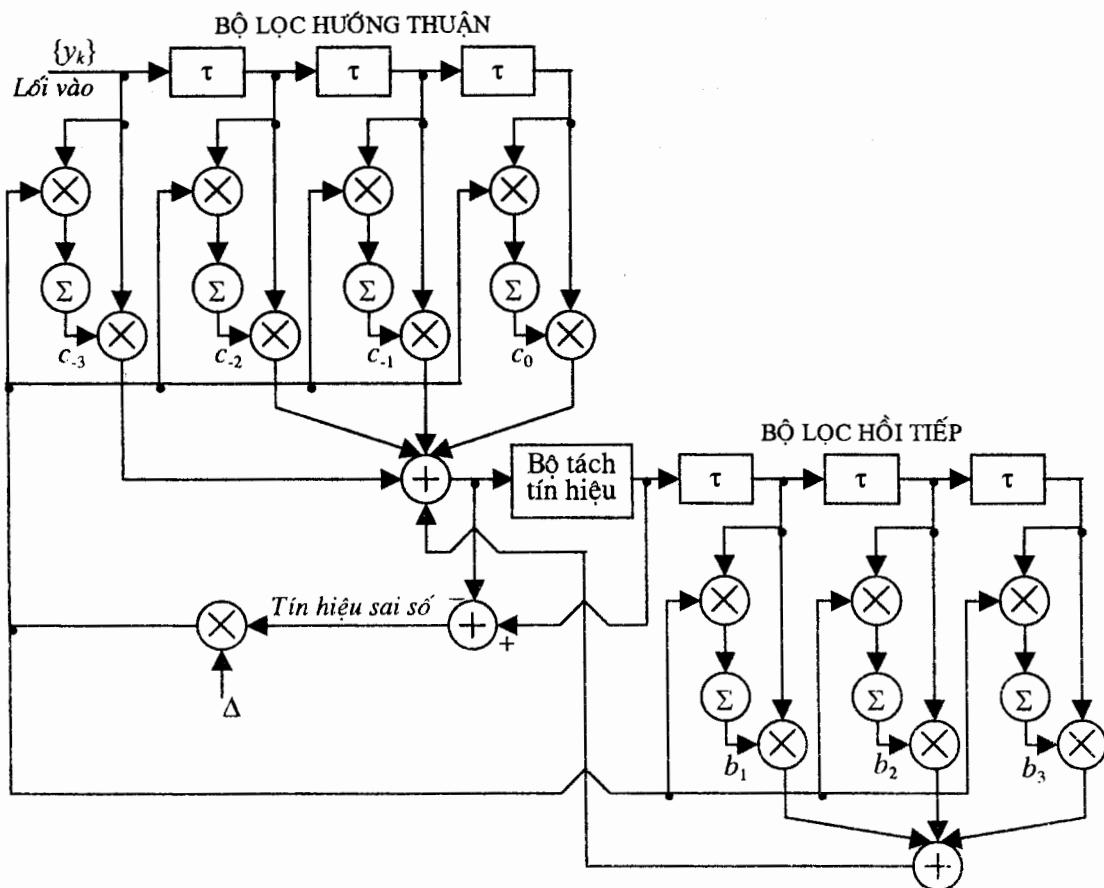


Hình 6.37 Sơ đồ khối của một DFE

Các hệ số khâu của bộ lọc hướng thuận và bộ lọc hồi tiếp được lựa chọn để tối ưu hoá một độ đo chất lượng mong muốn nào đó. Để đơn giản về mặt toán học, tiêu chuẩn MSE thường được áp dụng và thuật toán *gradient* ngẫu nhiên được sử dụng rộng rãi để thực hiện DFE thích nghi. Hình 6.38 minh họa sơ đồ khối của một DFE thích nghi mà các hệ số khâu của nó được điều chỉnh theo thuật toán *gradient* ngẫu nhiên LMS.

Chúng ta cần chú ý rằng các lối quyết định từ bộ tách tín hiệu khi được cấp tới bộ lọc hồi tiếp thì có tác động nhỏ tới thao tác của DFE. Nói chung, một tổn thất nhỏ

về chất lượng vào cỡ từ 1 đến 2 dB là có thể có tại các giá trị tỷ lệ lỗi dưới  $10^{-2}$ , song các lỗi quyết định trong các bộ lọc hồi tiếp thì không đến nỗi là thảm họa.



Hình 6.38 Sơ đồ khối của một DFE thích nghi

Mặc dù DFE tốt hơn một bộ san bằng tuyến tính song nó không phải là tối ưu theo quan điểm tối thiểu hóa xác suất lỗi trong việc quyết định dãy thông tin  $\{a_k\}$  từ các mẫu tín hiệu thu được  $\{y_k\}$ , được cho bởi (6.4.6). Trong một hệ thống truyền tin số truyền thông tin qua một kênh gây ra ISI, bộ tách tín hiệu tối ưu là một bộ tách dãy *symbol* có khả năng cực đại, mà nó cho ra tại lỗi ra của mình dãy *symbol* có khả năng nhất  $\{\tilde{a}_k\}$  đối với dãy thu đã được lấy mẫu đã cho  $\{y_k\}$ . Nghĩa là, bộ tách tín hiệu tìm dãy  $\{\tilde{a}_k\}$  cực đại hóa hàm khả năng (*likelihood function*)

$$\Lambda(\{a_k\}) = \ln p(\{y_k\} | \{a_k\})$$

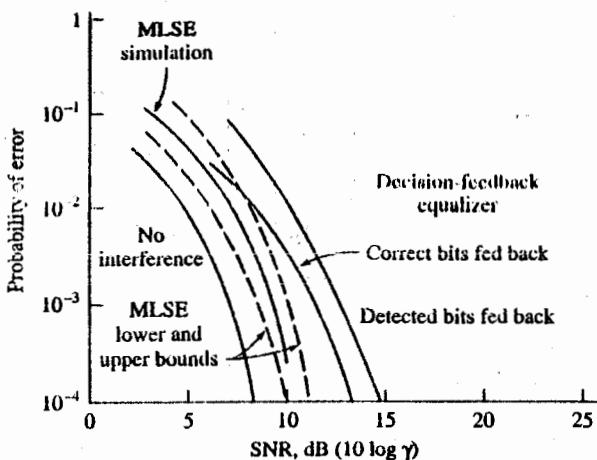
trong đó  $p(\{y_k\} | \{a_k\})$  là xác suất cùng nhau của dãy thu được  $\{y_k\}$  với điều kiện

$\{a_k\}$ . Bộ tách tín hiệu, thực hiện tách dãy *symbol*  $\{\tilde{a}_k\}$  cực đại hoá xác suất điều kiện cùng nhau này, được gọi là bộ tách tín hiệu theo dãy có khả năng cực đại (*maximum-likelihood sequence detector*).

Một thuật toán thực hiện tách dãy có khả năng cực đại (MLSD: *maximum-likelihood sequence detection*) là thuật toán Viterbi vốn được đặt ra để giải các mã chập như được mô tả trong mục 8.3.2. Để có được một mô tả về thuật toán này trong khuôn khổ vấn đề tách dãy trong điều kiện có ISI, bạn đọc nên tham khảo các tài liệu [3, 4].

Hạn chế chính của MLSD đối với các kênh có ISI là độ phức tạp tính toán tăng theo hàm mũ của sải ISI. Do đó, MLSD chỉ thực tế đối với các kênh trong đó sải ISI trải ra chỉ trên vài *symbol* song ISI nghiêm trọng theo nghĩa nó gây nên suy giảm nghiêm trọng về chất lượng của một bộ san bằng tuyến tính hay bộ san bằng hồi tiếp quyết định. Để lấy làm ví dụ, hình 6.39 minh họa chất lượng xác suất lỗi của thuật toán Viterbi đối với một tín hiệu PAM nhị phân được truyền qua kênh B (xem hình 6.35). Để so sánh, chúng tôi cũng minh họa xác suất lỗi đối với một DFE. Cả hai kết quả đều nhận được thông qua mô phỏng máy tính. Chúng ta thấy rằng chất lượng của MLSD thì tốt hơn vào khoảng 4.5 dB so với của DFE xét tại giá trị xác suất lỗi bằng  $10^{-4}$ . Do đó, đây chính là một thí dụ trong đó trên một kênh có sải ISI tương đối ngắn, MLSD mang lại lợi ích đáng kể về chất lượng.

Tóm lại, các bộ san bằng kênh được sử dụng rộng rãi trong các hệ thống thông tin số nhằm hạn chế ảnh hưởng của ISI gây bởi méo kênh. Các bộ san bằng tuyến tính được sử dụng rộng rãi cho các *modem* tốc độ cao truyền dữ liệu qua các kênh điện thoại. Đối với truyền dẫn vô tuyến, như trong thông tin di động tế bào, việc truyền lan sóng đa đường của tín hiệu phát gây ra ISI rất nghiêm trọng. Các kênh như thế đòi hỏi phải sử dụng các bộ san bằng mạnh hơn nhằm chống lại ISI nghiêm trọng đó. Bộ san bằng hồi tiếp quyết định và MLSD là hai bộ san bằng phi tuyến thích hợp đối với các kênh vô tuyến với ISI nghiêm trọng.



Hình 6.39 Xác suất lỗi của thuật toán Viterbi đối với một tín hiệu PAM được truyền qua kênh B trên hình 6.35

## CÁC BÀI TẬP

**6.1** Biến đổi Fourier của xung chữ nhật trong bài tập minh họa 6.1 và phổ công suất  $S_v(f)$  có thể tính được theo phương pháp số với MATLAB bằng việc dùng biến đổi Fourier rời rạc (DFT) hoặc thuật toán biến đổi Fourier nhanh. Hãy chuẩn hoá  $T=1$  và  $\sigma_a^2 = 1$ . Sau đó hãy lấy mẫu xung chữ nhật  $g(t)$  tại  $t=k/10$  với  $k=0, 1, \dots, 127$ . Điều này đưa đến dãy  $\{g_k\}$  gồm các mẫu của  $g(t)$ . Hãy sử dụng MATLAB để tính DFT 128 điểm của  $\{g_k\}$  và vẽ các giá trị  $|G_m|^2$  với  $m=0, 1, \dots, 127$ . Hãy vẽ cả phổ chính xác  $|G(f)|^2$  được cho bởi (6.2.13) và so sánh hai kết quả.

**6.2** Hãy lặp lại tính toán trong bài tập 6.1 khi xung  $g(t)$  được cho theo

$$g(t) = \begin{cases} \frac{1}{2} \left( 1 - \cos \frac{2\pi t}{T} \right), & 0 \leq t \leq T \\ 0, & \text{t còn lại} \end{cases}$$

Hãy cho  $T=1$  đối với tính toán này.

**6.3** Hãy viết một chương trình MATLAB để tính phổ công suất  $S_v(f)$  của tín hiệu  $V(t)$  khi xung  $g(t)$  là

$$g(t) = \begin{cases} \frac{1}{2} \left( 1 - \cos \frac{2\pi t}{T} \right), & 0 \leq t \leq T \\ 0, & t \text{ còn lại} \end{cases}$$

còn dãy  $\{a_n\}$  của các biên độ tín hiệu có hàm lượng quan được cho bởi (6.2.14).

**6.4** Hãy sử dụng MATLAB để thiết kế một bộ lọc pha tuyến tính FIR để mô hình hoá một kênh thông thấp có độ rộng băng hữu hạn, có độ nhấp nhô  $\frac{1}{2}$  dB trong dải thông  $|f| \leq 3000$  Hz và suy hao băng chấn là  $-40$  dB với  $|f| \geq 3500$  Hz. Hãy vẽ đáp ứng xung và đáp ứng tần số.

**6.5** Hãy viết một chương trình MATLAB thiết kế một bộ lọc pha tuyến tính FIR để mô hình hoá một kênh thông thấp có độ rộng băng hữu hạn với đáp ứng biên

độ mong muốn       $A(f) = \begin{cases} 1, & |f| \leq 3000 \\ 0, & |f| > 3000 \end{cases}$

bằng phương pháp cửa sổ, sử dụng một cửa sổ Hanning.

**6.6** Hãy viết một chương trình MATLAB tạo ra đáp ứng xung đối với kênh hai tia (đa đường) như trong bài tập minh họa 6.5 và vẽ đáp ứng xung trong trường hợp  $p=0,95$  và độ trễ là 5 mẫu.

**6.7** Hãy viết một chương trình mô phỏng MATLAB để tạo ra kênh 1 trong bài tập minh họa 6.6 và hãy đo tỉ lệ lỗi khi 10 000 bit dữ liệu nhị phân  $\{\pm 1\}$  được truyền qua kênh này. Kênh bị nhiễu loạn bởi AWGN với phương sai  $\sigma^2=0$ ,  $\sigma^2=0.1$ ,  $\sigma^2=0.2$ ,  $\sigma^2=0.5$  và  $\sigma^2=1.0$ .

**6.8** Lặp lại bài tập 6.7 đối với kênh sau

$$x_n = \begin{cases} 1, & n = 0 \\ 0,25, & n = \pm 1 \\ 0, & n \text{ còn lại} \end{cases}$$

**6.9** Hãy viết một chương trình MATLAB tạo ra một phiên bản đã lấy mẫu của đáp ứng xung mạch lọc phát  $g_T(t)$  được cho bởi (6.5.9) đối với một giá trị tuỳ ý của hệ số uốn  $\alpha$ . Hãy ước lượng và vẽ  $g_T(n)$  đối với  $\alpha=1/2$  và  $N=31$ . Hãy ước lượng và vẽ

cả các đặc tuyến biên độ của đáp ứng tần số đối với bộ lọc này. [Hãy sử dụng một DFT 4 điểm của  $g_T(n)$  bằng cách nhồi thêm cho  $g_T(n)$  3N điểm không].

**6.10** Hãy viết một chương trình MATLAB tính toán đáp ứng xung tổng cộng của mạch mắc nối tiếp một bộ lọc phát bất kỳ  $g_T(n)$  với bộ lọc phối hợp của nó tại máy thu. Tính toán này có thể thực hiện được bằng cách sử dụng DFT như sau. Nhồi thêm cho  $g_T(n)$  bằng  $N-1$  (hoặc hơn thế) điểm không và tính DFT ( $2N-1$ ) (hoặc hơn thế) điểm. Tính toán đó sẽ cho chúng ta  $G_T(k)$ . Sau đó tạo ra  $|G_T(k)|^2$  rồi tính DFT ngược ( $2N-1$ ) điểm của  $|G_T(k)|^2$ . Hãy ước lượng đáp ứng xung tổng cộng này đối với bộ lọc trong bài tập 6.9 và so sánh kết quả này với đáp ứng xung lý tưởng nhận được bằng cách lấy mẫu  $x_{rc}(t)$  với tốc độ lấy mẫu  $F_s=4/T$ .

**6.11** Lặp lại bài tập 6.9 đối với  $N=21$  và  $N=41$ . Hãy vẽ và so sánh đáp ứng tần số của các bộ lọc thời gian rời rạc này với các bộ lọc trong bài tập 6.9. Hãy mô tả các sai khác chủ yếu.

**6.12** Hãy viết một chương trình MATLAB mà nó nhận một dãy dữ liệu  $\{D_k\}$ , tiền mã hoá nó cho một hệ thống truyền dẫn xung nhị phân đôi cải biên, rồi ánh xạ dãy đã tiền mã hoá thành các mức biên độ phát  $\{a_k\}$ . Sau đó, từ dãy được phát đi hãy tạo ra dãy thu được không có tạp âm  $\{b_k=a_k-a_{k-2}\}$  và bằng cách sử dụng quan hệ được cho bởi (6.5.37), hãy khôi phục lại tín hiệu dữ liệu  $\{D_k\}$ . Hãy chạy chương trình này đối với dãy dữ liệu  $\{D_k\}$  giả ngẫu nhiên tùy ý với  $M=2$  và  $M=4$  các mức biên độ phát, và hãy kiểm tra các kết quả.

**6.13** Hãy viết một chương trình MATLAB thực hiện một mô phỏng Monte-Carlo cho một hệ thống truyền tin PAM nhị phân sử dụng xung tín hiệu nhị phân đôi, trong đó dãy biên độ và tiền mã  $\{a_k\}$  được hình thành như trong bài tập minh họa 6.9. Hãy cộng thêm tạp âm Gauss vào dãy tín hiệu thu được  $\{b_k\}$  như đã chỉ ra trong (6.5.22) để tạo nên lỗi vào mạch tách tín hiệu, rồi sử dụng quy tắc tách tín hiệu trong (6.5.28) để khôi phục dữ liệu. Hãy thực hiện mô phỏng đối với 10 000 bit và hãy đo xác suất lỗi bít đối với  $\sigma^2=0.1$ ,  $\sigma^2=0.5$  và  $\sigma^2=1.0$ . Hãy vẽ xác suất

lỗi lý thuyết đối với PAM nhị phân khi không có ISI rồi so sánh các kết quả mô phỏng Monte-Carlo với chất lượng lý tưởng này. Bạn sẽ thấy một suy giảm nhỏ nào đó trong chất lượng của hệ thống nhị phân đôi.

**6.14** Hãy lặp lại bài tập 6.9 đối với một tốc độ lấy mẫu  $F_s=8/T$ ,  $\alpha=1/2$  và  $N=61$ . Tốc độ lấy mẫu cao hơn có tạo ra một đặc tuyến đáp ứng tần số tốt hơn, nghĩa là có cho một đặc tuyến sát hợp hơn với  $X_{rc}(f)$ ?

**6.15** Đối với bộ lọc đã thiết kế trong bài tập 6.14, hãy tính và vẽ lỗi ra của mạch mắc nối tiếp bộ lọc này với bộ lọc phối hợp của nó bằng cách sử dụng thủ tục đã được mô tả trong bài tập 6.10. Hãy so sánh đáp ứng xung đã lấy mẫu này với đáp ứng xung lý tưởng nhận được bằng cách lấy mẫu  $x_{rc}(t)$  với tốc độ  $F_s=8/T$ . Tốc độ lấy mẫu cao hơn có cho một xấp xỉ tốt hơn của đáp ứng xung bộ lọc thời gian rời rạc đối với đáp ứng xung bộ lọc lý tưởng?

**6.15** Đối với bộ lọc đã thiết kế trong bài tập 6.14, hãy tính và vẽ lỗi ra của mạch mắc nối tiếp của bộ lọc này và mạch lọc phối hợp với nó bằng cách sử dụng thủ tục đã được mô tả trong bài tập 6.10. Hãy so sánh đáp ứng xung đã được lấy mẫu này với đáp ứng xung lý tưởng nhận được bằng cách lấy mẫu  $x_{rc}(t)$  với tốc độ  $F_s=8/T$ . Tốc độ lấy mẫu cao hơn này có cho một xấp xỉ tốt hơn của đáp ứng xung bộ lọc thời gian rời rạc so với đáp ứng xung bộ lọc lý tưởng không?

**6.16** Hãy viết một chương trình MATLAB tạo ra một phiên bản đã lấy mẫu của đáp ứng xung bộ lọc phát  $g_T(t)$  cho bởi (6.5.21) đối với tín hiệu nhị phân đôi cài biên xác định theo (6.5.18). Hãy ước lượng và vẽ  $g_T(n)$  với  $N=31$ . Hãy ước lượng và vẽ cả biên độ của đáp ứng tần số của bộ lọc này.

**6.17** Lặp lại bài tập 6.10 đối với bộ lọc đã thiết kế được trong bài tập 6.16.

**6.18** Lặp lại bài tập 6.16 đối với  $N=21$  và  $N=41$ . Hãy so sánh các đáp ứng tần số của các bộ lọc này với đáp ứng tần số của bộ lọc đã thiết kế được trong bài tập 6.16. Các sai biệt chính trong các đặc tuyến đáp ứng tần số này là gì?

**6.19** Hãy xét xung  $x(t)$  bị méo bởi kênh, như đã cho trong bài tập minh họa 6.10. Xung này được lấy mẫu với tốc độ  $2/T$  và được sửa (san bằng) bằng một bộ san

bằng cưỡng ép không với  $2K+1$  khâu. Hãy viết một chương trình MATLAB để giải tìm các hệ số khâu của bộ san bằng cưỡng ép không. Hãy ước lượng và vẽ lối ra của bộ san bằng cưỡng ép không này đối với 50 giá trị đã lấy mẫu.

**6.20** Lặp lại bài tập 6.19 đối với một bộ san bằng MSE với  $\sigma^2=0.01$ ,  $\sigma^2=0.1$  và  $\sigma^2=1.0$ . Hãy so sánh các hệ số của bộ san bằng này với các hệ số đã nhận được trong bài tập 6.19 và hãy bình luận về các kết quả khi  $\sigma^2$  được cho biến thiên.

**6.21** Hãy viết một chương trình MATLAB tổng quát để tính các hệ số khâu của một bộ san bằng FIR với độ dài  $2K+1$  tùy ý dựa trên tiêu chuẩn MSE, với các dữ liệu đầu vào của chương trình là các giá trị đã lấy mẫu của xung  $x(t)$  với tốc độ lấy mẫu bằng tốc độ *symbol* và mật độ phổ của tạp âm cộng  $\sigma^2$ . Hãy sử dụng chương trình đó để ước lượng các hệ số của một bộ san bằng 11 khâu khi các giá trị đã lấy mẫu của  $x(t)$  là

$$x(nT) = \begin{cases} 1, & n = 0 \\ 0.5, & n = \pm 1 \\ 0.3, & n = \pm 3 \\ 0.1, & n = \pm 4 \end{cases}$$

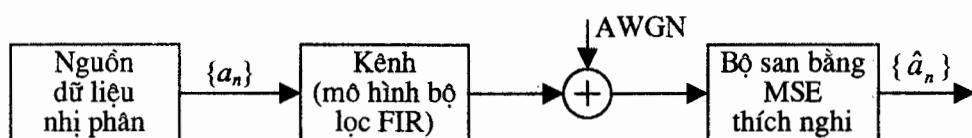
$\sigma^2=0.01$ , và  $\sigma^2=0.1$ . Hãy ước lượng cả MSE tối thiểu đối với các hệ số tối ưu của bộ san bằng.

**6.22** Đối với các đặc tính kênh đã được cho trong bài tập 6.21, hãy ước lượng các hệ số của bộ san bằng MSE và MSE tối thiểu khi số khâu của bộ san bằng là 21. Hãy so sánh các hệ số của bộ san bằng này với các giá trị của các hệ số đã nhận được trong bài tập 6.21 và hãy bình luận xem độ giảm MSE nhận được với bộ san bằng dài hơn có lớn đến mức đáng để sử dụng nó hay không.

**6.23** Lượng ISI còn dư tại lối ra của một bộ san bằng có thể ước lượng được bằng cách nhân chập đáp ứng xung của kênh với các hệ số của bộ san bằng và quan sát dãy lối ra kết quả của tích chập. Hãy viết một chương trình MATLAB tính toán lối ra của bộ san bằng có một độ dài xác định khi lối vào là đặc tính kênh đã lấy mẫu. Để đơn giản, hãy xét trường hợp trong đó bộ san bằng là một bộ san

bằng phân cách từng *symbol* và đáp ứng xung đã lấy mẫu của kênh cũng gồm các mẫu phân cách nhau cùng độ dài của *symbol*. Hãy sử dụng chương trình đó để ước lượng lỗi ra của một bộ san bằng cưỡng ép không, phân cách từng *symbol* đối với đáp ứng xung kênh đã được cho trong bài tập 6.21.

**6.24** Hãy viết một chương trình mô phỏng Monte-Carlo thực hiện mô phỏng hệ thống truyền tin số đọc mô hình hoá như trên hình vẽ P6.24. Kênh được mô hình hoá như một bộ lọc FIR với các giá trị phân cách nhau từng *symbol*. Bộ san bằng MSE cũng là một bộ lọc FIR với các hệ số khâu phân cách nhau từng *symbol*. Các *symbol* huấn luyện thoát tiên được truyền để huấn luyện bộ san bằng. Trong phương thức dữ liệu (*data mode*) - là phương thức truyền đưa dữ liệu sau khi đã kết thúc huấn luyện, chú thích của người dịch - bộ san bằng sử dụng lỗi ra của bộ tách tín hiệu (*detector*) để tạo nên tín hiệu sai số. Hãy thực hiện một mô phỏng Monte-Carlo đối với hệ thống này bằng cách sử dụng 1000 *symbol* (nhị phân) huấn luyện và 10 000 *symbol* dữ liệu nhị phân đối với mô hình kênh đã được cho trong bài tập 6.21. Hãy sử dụng  $\sigma^2=0.01$ ,  $\sigma^2=0.1$  và  $\sigma^2=1.0$ . Hãy so sánh tỷ lệ lỗi thu được từ mô phỏng với tỷ lệ lỗi của một kênh lý tưởng không có ISI.



Hình P6.24

## CHƯƠNG 7

**TRUYỀN DẪN SỐ THÔNG QUA ĐIỀU CHẾ SÓNG MANG****7.1 GIỚI THIỆU CHUNG**

Trong hai chương trước chúng ta đã xem xét việc truyền dẫn thông tin số thông qua các kênh băng gốc. Trong trường hợp như thế, tín hiệu mang thông tin được truyền trực tiếp qua kênh mà không sử dụng một sóng mang hình sin. Tuy vậy, hầu hết các kênh thông tin đều là các kênh thông dài; do vậy, cách duy nhất để truyền các tín hiệu thông qua các kênh như thế là nhờ việc dịch chuyển tần số của tín hiệu mang thông tin tới băng tần của kênh.

Trong chương này, chúng ta sẽ xem xét bốn kiểu tín hiệu đã điều chế sóng mang thích hợp cho các kênh thông dài: Các tín hiệu điều chế biên độ, các tín hiệu điều chế biên độ vuông góc, các tín hiệu điều chế pha số và các tín hiệu điều chế tần số số.

**7.2 ĐIỀU CHẾ BIÊN ĐỘ SÓNG MANG**

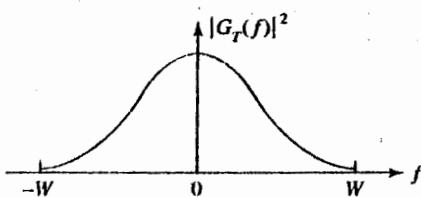
Trong PAM số băng gốc, các dạng sóng tín hiệu có dạng

$$s_m(t) = A_m g_T(t) \quad (7.2.1)$$

trong đó  $A_m$  là biên độ của dạng sóng thứ  $m$  và  $g_T(t)$  là một xung mà dạng của nó xác định các đặc tính phổ của tín hiệu được truyền đi. Phổ của các tín hiệu băng gốc được giả sử là được chứa trong dải tần số  $|f| \leq W$ , trong đó  $W$  là độ rộng băng của  $|G_T(f)|^2$  như được minh họa trên hình 7.1. Ta hãy nhớ lại rằng biên độ của tín hiệu nhận các giá trị

$$A_m = (2m - 1 - M)d, \quad m = 1, 2, \dots, M \quad (7.2.2)$$

trong đó  $2d$  là khoảng cách Euclide giữa hai điểm tín hiệu lân cận nhau.



Hình 7.1 Phổ mật độ năng lượng của tín hiệu được truyền đi  $g_T(t)$ .

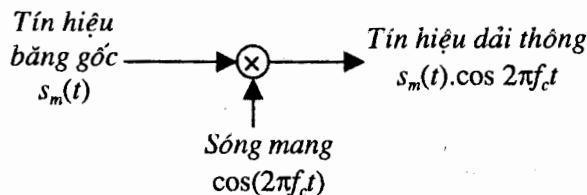
Để truyền các dạng sóng tín hiệu số thông qua một kênh thông dài, các dạng sóng tín hiệu bằng gốc  $s_m(t)$ ,  $m = 1, 2, \dots, M$ , được nhân với một sóng mang hình sin có dạng  $\cos 2\pi f_c t$  như được trình bày trên hình 7.2, trong đó  $f_c$  là tần số sóng mang ( $f_c > W$ ) và ứng với tần số trung tâm trong dải thông của kênh. Do đó, các dạng sóng tín hiệu được truyền đi biểu diễn được theo

$$u_m(t) = A_m g_T(t) \cos 2\pi f_c t, \quad m = 1, 2, \dots, M \quad (7.2.3)$$

Trong trường hợp đặc biệt, khi dạng xung được truyền là chữ nhật, nghĩa là

$$g_T(t) = \begin{cases} \sqrt{\frac{2}{T}}, & 0 \leq t \leq T \\ 0, & \text{với các giá trị còn lại của } t \end{cases}$$

thì tín hiệu sóng mang được điều chế biên độ thường được gọi là khoá dịch biên độ (ASK: Amplitude-Shift Keying). Trong trường hợp này, tín hiệu PAM không phải là có bång hạn chế.



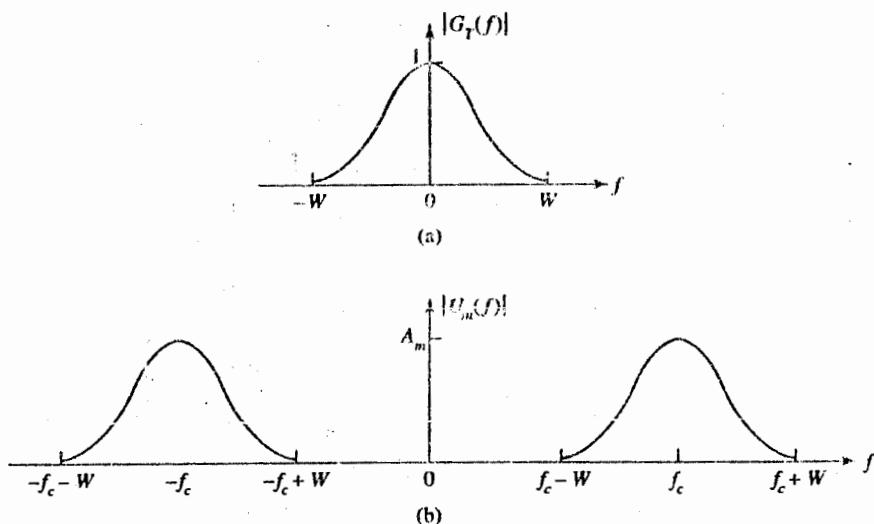
Hình 7.2 Điều chế biên độ một sóng mang bằng tín hiệu PAM bằng gốc.

Việc điều chế biên độ của sóng mang  $\cos 2\pi f_c t$  bằng các dạng sóng bằng gốc  $s_m(t)$  làm dịch phổ của tín hiệu bằng gốc một lượng  $f_c$  và do đó đưa tín hiệu lên

dải thông của kênh. Chúng ta hãy nhớ lại rằng biến đổi Fourier của sóng mang là  $[\delta(f - f_c) + \delta(f + f_c)]/2$ . Do việc nhân hai tín hiệu trên miền thời gian thì tương ứng với tích chập của các phô của chúng trên miền tần số nên phô của tín hiệu được điều chế biên độ là

$$U_m(f) = \frac{A_m}{2} [G_T(f - f_c) + G_T(f + f_c)] \quad (7.2.4)$$

Như thế, phô của tín hiệu băng gốc  $s_m(t) = A_m g_T(t)$  được dịch về tần số một lượng  $f_c$ . Tín hiệu thông dải là một tín hiệu AM hai băng biên sóng mang bị nén (DSB-SC: Double-SideBand Suppressed-Carrier), như được minh họa trên hình 7.3.



Hình 7.3 Các phô của các tín hiệu (a) băng gốc và (b) tín hiệu đã điều chế.

Chúng ta hãy để ý rằng việc đóng tín hiệu băng gốc  $s_m(t)$  lên biên độ của tín hiệu sóng mang  $\cos 2\pi f_c t$  không làm thay đổi biểu diễn hình học cơ sở của các dạng sóng tín hiệu PAM số. Các dạng sóng tín hiệu PAM thông dải có thể biểu diễn được theo

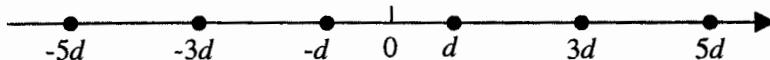
$$u_m(t) = s_m \psi(t) \quad (7.2.5)$$

trong đó dạng sóng tín hiệu  $\psi(t)$  được xác định theo

$$\psi(t) = g_T(t) \cos 2\pi f_c t \quad (7.2.6)$$

$$\text{còn } s_m = A_m, \quad m = 1, 2, \dots, M \quad (7.2.7)$$

ký hiệu các điểm tín hiệu mà chúng nhận  $M$  giá trị trên đường thẳng trực thực như thấy được trên hình 7.4.



Hình 7.4 Biểu đồ sao tín hiệu đối với tín hiệu PAM.

Dạng sóng tín hiệu  $\psi(t)$  được chuẩn hoá để có năng lượng bằng đơn vị; tức là

$$\int_{-\infty}^{\infty} \psi^2(t) dt = 1 \quad (7.2.8)$$

Hệ quả là

$$\int_{-\infty}^{\infty} g_T^2(t) \cos^2 2\pi f_c t dt = \frac{1}{2} \int_{-\infty}^{\infty} g_T^2(t) dt + \frac{1}{2} \int_{-\infty}^{\infty} g_T^2(t) \cos 4\pi f_c t dt = 1 \quad (7.2.9)$$

Song

$$\int_{-\infty}^{\infty} g_T^2(t) \cos 4\pi f_c t dt = 0 \quad (7.2.10)$$

do độ rộng băng  $W$  của  $g_T(t)$  thì rất nhỏ hơn tần số sóng mang, nghĩa là  $f_c >> W$ . Trong trường hợp như thế,  $g_T(t)$  về căn bản là hằng số trong bất kỳ một chu kỳ nào của  $\cos 4\pi f_c t$ , vì vậy tích phân trong (7.2.10) bằng không đối với từng chu kỳ của biểu thức dưới dấu tích phân. Do (7.2.10) ta có

$$\frac{1}{2} \int_{-\infty}^{\infty} g_T^2(t) dt = 1 \quad (7.2.11)$$

Như vậy,  $g_T(t)$  nhất thiết phải được tính tỷ lệ một cách thích hợp sao cho các biểu thức (7.2.8) và (7.2.11) được thoả mãn.

## 7.2.1 GIẢI ĐIỀU CHẾ CÁC TÍN HIỆU PAM

Việc giải điều chế của một tín hiệu PAM số thông dải có thể hoàn thành được theo một trong nhiều cách bằng các tính tương quan hay lọc phối hợp. Với mục đích minh họa, chúng ta xét một bộ giải điều chế kiểu tính tương quan.

Tín hiệu thu được có thể biểu diễn được theo

$$r(t) = A_m g_T(t) \cos 2\pi f_c t + n(t) \quad (7.2.12)$$

trong đó  $n(t)$  là một quá trình tạp âm thông dải biểu diễn được theo

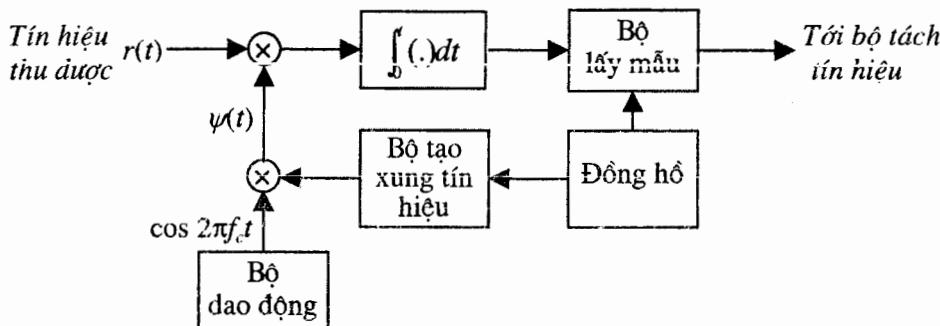
$$n(t) = n_c(t) \cos 2\pi f_c t - n_s(t) \sin 2\pi f_c t \quad (7.2.13)$$

và ở đây  $n_c(t)$  và  $n_s(t)$  là các thành phần vuông góc với nhau của tạp âm.

Bằng cách tính tương quan chéo tín hiệu thu được với  $\psi(t)$  được cho bởi (7.2.6), như trên hình 7.5, chúng ta nhận được lối ra

$$\int_{-\infty}^{\infty} r(t) \psi(t) dt = A_m + n = s_m + n \quad (7.2.14)$$

trong đó  $n$  biểu thị thành phần tạp âm cộng tại lối ra của bộ tương quan.



Hình 7.5 Giải điều chế tín hiệu PAM số thông dải.

Thành phần tạp âm có kỳ vọng bằng 0. Phương sai của nó có thể biểu diễn theo

$$\sigma_n^2 = \int_{-\infty}^{\infty} |\Psi(f)|^2 S_n(f) df \quad (7.2.15)$$

trong đó  $\Psi(f)$  là biến đổi Fourier của  $\psi(t)$ , còn  $S_n(f)$  là mật độ phổ công suất của tạp âm cộng. Biến đổi Fourier của  $\psi(t)$  là

$$\Psi(f) = \frac{1}{2} [G_T(f - f_c) + G_T(f + f_c)] \quad (7.2.16)$$

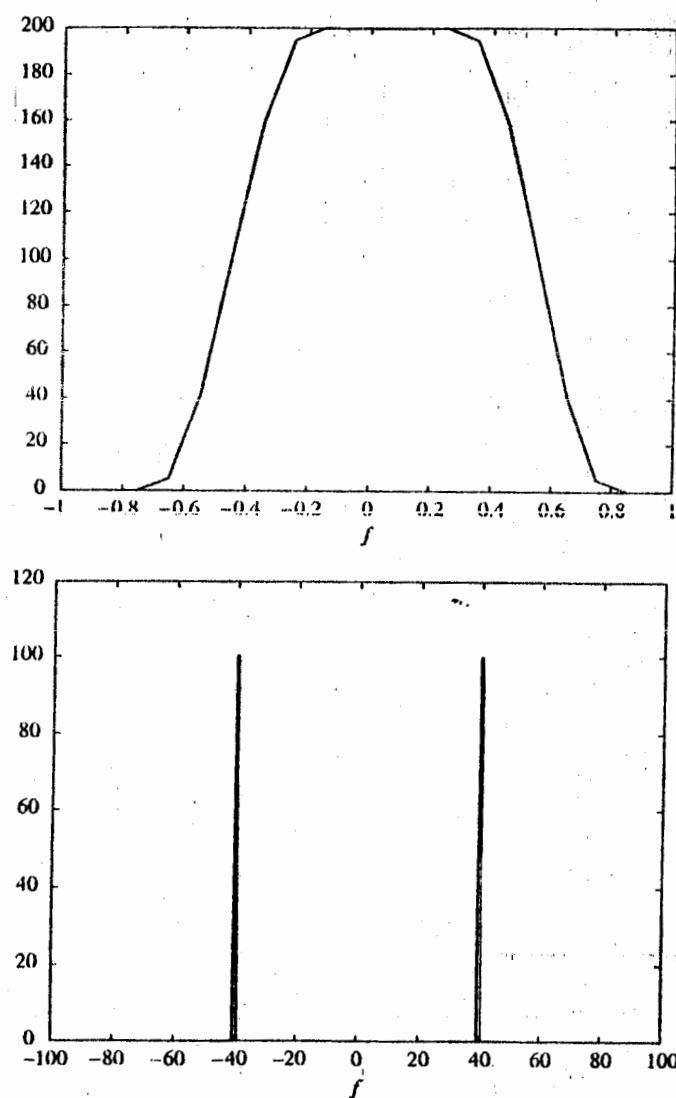
và mật độ phổ công suất của quá trình tạp âm cộng thông dải là

$$S_n(f) = \begin{cases} \frac{N_0}{2}, & |f - f_c| \leq W \\ 0, & \text{với các giá trị khác của } f \end{cases} \quad (7.2.17)$$

Bằng cách thay (7.2.16) và (7.2.17) vào (7.2.15) và tính tích phân, chúng ta nhận được  $\sigma_n^2 = N_0 / 2$ .

Từ (7.2.14) là biểu thức của lỗi vào bộ tách biên độ, rõ ràng rằng xác suất lỗi của bộ tách tín hiệu tối ưu đối với tín hiệu PAM điều chế biên độ thì hoàn toàn giống như xác suất lỗi của bộ tách biên độ đối với PAM băng gốc. Nghĩa là

$$P_M = \frac{2(M-1)}{M} Q\left(\sqrt{\frac{6E_{avb} \log_2 M}{(M^2 - 1)N_0}}\right) \quad (7.2.18)$$



Hình 7.6 Các phổ của tín hiệu băng gốc và của tín hiệu điều chế biên độ (thông dài).

## BÀI TẬP MINH HỌA

**Bài tập minh họa 7.1 [Phổ tín hiệu PAM]** Trong một hệ thống PAM số điều chế biên độ, bộ lọc máy phát với phản ứng xung  $g_T(t)$  có đặc tính phô căn bậc hai cosin nâng như được mô tả trong bài tập minh họa 6.7, với hệ số uốn lõi (*roll-off factor*)  $\alpha = 0.5$ . Tân số sóng mang là  $f_c = 40/T$ . Hãy tính và vẽ đồ thị phổ của tín hiệu băng gốc và của tín hiệu đã được điều chế biên độ.

### Lời giải

Hình 7.6 minh họa hai đặc tính phổ này. MATLAB *script* đối với tính toán này được cho dưới đây.

#### m-file

```
% MATLAB script for Illustrated Problem 1, Chapter 7.
echo on
T=1;
delta_T=T/200; % sampling interval
alpha=0.5; % roll-off factor
fc=40/T; % carrier frequency
A_m=1; % amplitude
t=-5*T+delta_T:delta_T:5*T; % time axis
N=length(t);
for i=1:N,
    if (abs(t(i))~=T/(2*alpha)),
        g_T(i) = sinc(t(i)/T)*(cos(pi*alpha*t(i)/T)/(1-4*alpha^2*t(i)^2/T^2));
    else
        g_T(i) = 0; % the value of g_T is 0 at t=T/(2*alpha)
    end; % and, at t=-T/(2*alpha)
    echo off ;
end;
echo on;
G_T=abs(fft(g_T)); % spectrum of g_T
u_m=A_m*g_T.*cos(2*pi*fc*t); % the modulated signal
U_m=abs(fft(u_m)); % spectrum of the modulated signal
% actual frequency scale
f=-0.5/delta_T:1/(delta_T*(N-1)):0.5/delta_T;
% plotting commands follow
figure(1);
plot(f,fftshift(G_T));
axis([-1/T 1/T 0 max(G_T)]);
```

```
figure(2);
plot(f,fftshift(U_m));
```

### 7.3 ĐIỀU CHẾ PHA SÓNG MANG

Trong điều chế pha sóng mang, thông tin truyền qua một kênh thông tin được đóng trên pha của sóng mang. Do giải của pha sóng mang là  $0 \leq \theta < \pi$ , các pha của sóng mang được sử dụng để truyền thông tin số thông qua điều chế pha số là  $\theta_m = 2\pi m/M$  đối với  $m = 0, 1, 2, \dots, M-1$ . Như thế, đối với điều chế pha nhị phân ( $M = 2$ ), hai góc pha của sóng mang là  $\theta_0 = 0$  và  $\theta_1 = \pi$  rad. Đối với điều chế pha số  $M$  mức,  $M = 2^k$ , trong đó  $k$  là số bít thông tin trên một *symbol* được truyền đi.

Biểu diễn tổng quát của một tập gồm  $M$  dạng sóng tín hiệu được điều chế pha sóng mang là

$$u_m(t) = A g_T(t) \cos\left(2\pi f_c t + \frac{2\pi m}{M}\right), \quad m = 0, 1, 2, \dots, M-1 \quad (7.3.1)$$

trong đó  $g_T(t)$  là dạng xung bộ lọc phát mà nó xác định đặc tính phổ của tín hiệu được truyền đi, còn  $A$  là biên độ của tín hiệu. Kiểu điều chế pha này được gọi là khoá dịch pha (PSK: *Phase-Shift Keying*). Chúng ta hãy để ý rằng các tín hiệu PSK có cùng năng lượng như nhau, nghĩa là

$$E_m = \int_{-\infty}^{\infty} u_m^2(t) dt \quad (7.3.2)$$

$$= \int_{-\infty}^{\infty} A^2 g_T^2(t) \cos^2\left(2\pi f_c t + \frac{2\pi m}{M}\right) dt$$

$$= \frac{1}{2} \int_{-\infty}^{\infty} A^2 g_T^2(t) dt + \frac{1}{2} \int_{-\infty}^{\infty} A^2 g_T^2(t) \cos\left(4\pi f_c t + \frac{4\pi m}{M}\right) dt$$

$$= \frac{A^2}{2} \int_{-\infty}^{\infty} g_T^2(t) dt \quad (7.3.3)$$

$$\equiv E_s, \text{ đối với mọi } m \quad (7.3.4)$$

trong đó  $E_s$  ký hiệu cho năng lượng trên một *symbol* được truyền đi. Số hạng liên quan tới thành phần tần số gấp đôi trong (7.3.4) tính trung bình ra đạt tới không khi  $f_c \gg W$ , trong đó  $W$  là độ rộng băng của  $g_T(t)$ .

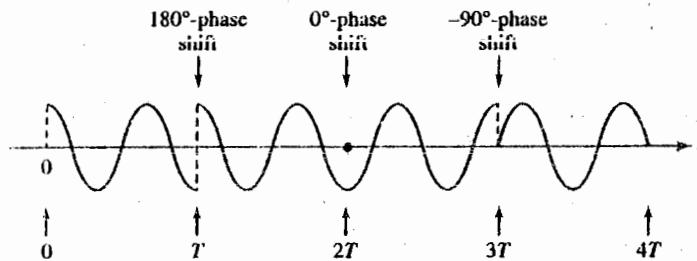
Khi  $g_T(t)$  là một xung chữ nhật thì nó được xác định theo

$$g_T(t) = \sqrt{\frac{2}{T}}, \quad 0 \leq t \leq T \quad (7.3.5)$$

Trong trường hợp này, các dạng sóng tín hiệu được truyền đi trong khoảng thời gian một symbol  $0 \leq t \leq T$  có thể biểu diễn được theo (với  $A = \sqrt{E_s}$ )

$$u_m(t) = \sqrt{\frac{2E_s}{T}} \cos(2\pi f_c t + \frac{2\pi m}{M}), \quad m = 0, 1, \dots, M-1 \quad (7.3.6)$$

Hãy chú ý rằng các tín hiệu được truyền đi được cho bởi (7.3.6) có một đường bao hằng số còn pha của sóng mang thay đổi đột biến tại điểm đầu của mỗi khoảng tín hiệu. Hình 7.7 minh họa một dạng sóng tín hiệu PSK bốn pha ( $M = 4$ ).



Hình 7.7 Thí dụ về một tín hiệu PSK bốn pha.

Bằng cách xem góc pha của hàm cosin trong (7.3.6) như một tổng của hai góc, chúng ta có thể biểu diễn các dạng sóng trong (7.3.1) theo

$$\begin{aligned} u_m(t) &= \sqrt{E_s} g_T(t) \cos\left(\frac{2\pi m}{M}\right) \cos 2\pi f_c t - \sqrt{E_s} g_T(t) \sin\left(\frac{2\pi m}{M}\right) \sin 2\pi f_c t \\ &= s_{mc} \psi_1(t) + s_{ms} \psi_2(t) \end{aligned} \quad (7.3.7)$$

trong đó

$$\begin{aligned} s_{mc} &= \sqrt{E_s} \cos \frac{2\pi m}{M} \\ s_{ms} &= \sqrt{E_s} \sin \frac{2\pi m}{M} \end{aligned} \quad (7.3.8)$$

còn  $\psi_1(t)$  và  $\psi_2(t)$  là các hàm cơ sở trực giao được xác định theo

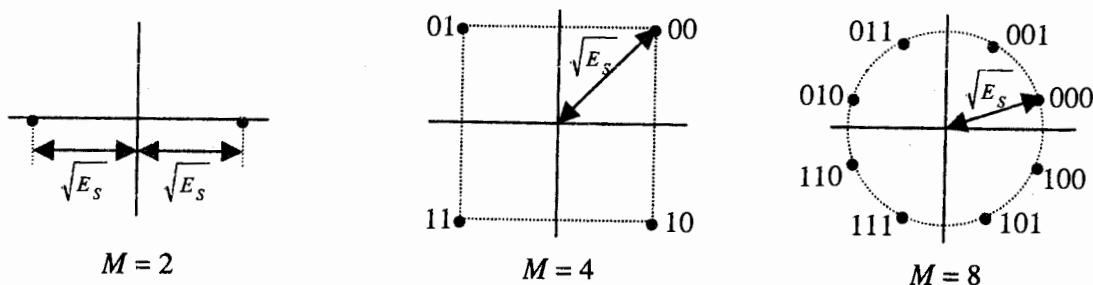
$$\begin{aligned} \psi_1(t) &= g_T(t) \cos 2\pi f_c t \\ \psi_2(t) &= g_T(t) \sin 2\pi f_c t \end{aligned} \quad (7.3.9)$$

Bằng cách chuẩn hoá thích hợp dạng xung  $g_T(t)$ , chúng ta có thể chuẩn hoá năng lượng của các hàm cơ sở trực giao này thành 1. Như thế, một tín hiệu điều chế pha có thể xem được như hai sóng mang vuông góc với các biên độ phụ thuộc vào pha được truyền đi trong từng khoảng thời gian của tín hiệu. Do đó, các tín hiệu điều chế pha số biểu diễn được một cách hình học như các véc-tơ hai chiều với các thành phần  $s_{mc}$  và  $s_{ms}$ , nghĩa là

$$\mathbf{s}_m = \left( \sqrt{E_s} \cos \frac{2\pi m}{M} \quad \sqrt{E_s} \sin \frac{2\pi m}{M} \right) \quad (7.3.10)$$

Biểu đồ sao các điểm tín hiệu đối với  $M = 2, 4$  và  $8$  được minh họa trên hình 7.8. Chúng ta thấy được rằng điều chế pha nhị phân thì đồng nhất với PAM nhị phân (các tín hiệu nhị phân đối cực).

Việc ánh xạ hay gán  $k$  bít thông tin thành  $M=2^k$  pha có thể có thể thực hiện được bằng một số cách. Cách gán thông dụng là sử dụng mã hoá Gray, trong đó các góc pha cạnh nhau thì khác nhau một *digit* nhị phân (một bít), như được minh họa trên hình vẽ 7.8. Hệ quả là, chỉ một lỗi bít đơn xảy ra trong một dãy  $k$  bít có mã Gray khi tạp nhiễu gây nên việc chọn nhầm một pha bên cạnh làm pha đã được truyền đi.



Hình 7.8 Các biểu đồ sao của tín hiệu PSK.

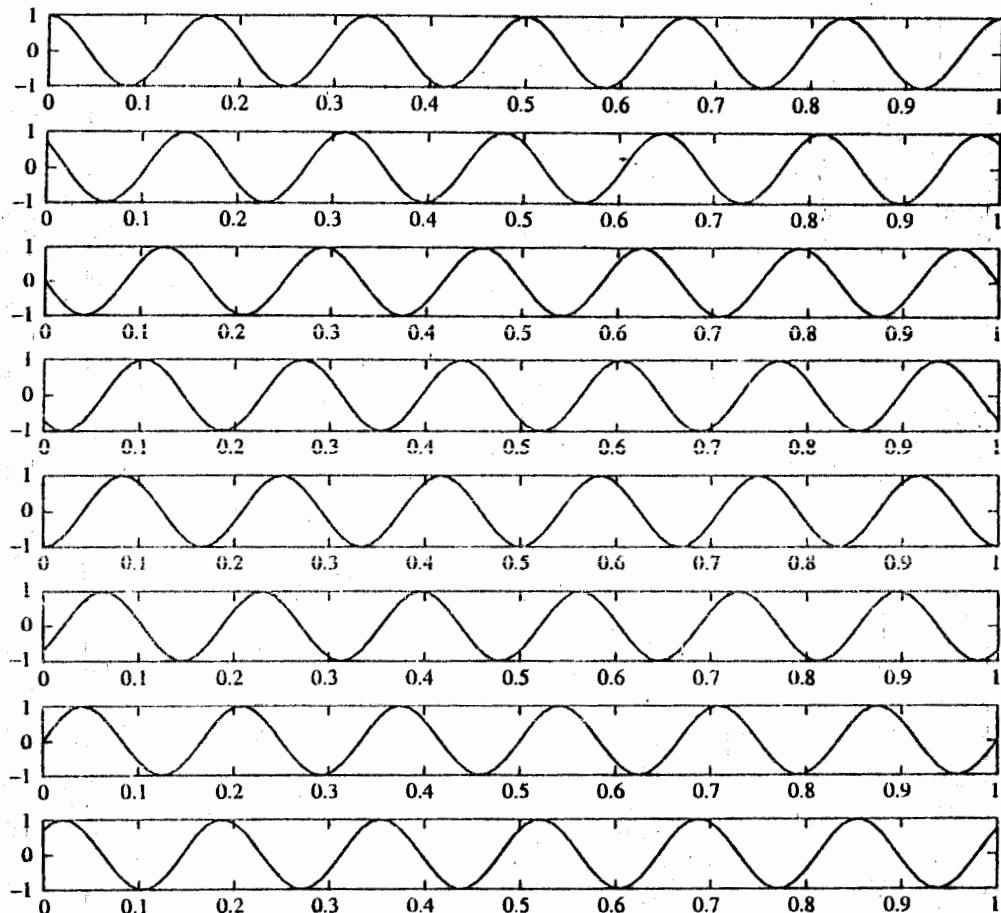
### BÀI TẬP MINH HỌA

**Bài tập minh họa 7.2 [Dạng sóng PSK]** Hãy tạo ra các dạng sóng tín hiệu PSK có đường bao không đổi được cho bởi (7.3.6) đối với  $M = 8$ . Để thuận tiện, biên độ tín hiệu được chuẩn hoá bằng 1.

**Lời giải**

Hình 7.9 minh họa 8 dạng sóng đối với trường hợp trong đó  $f_c = 6/T$ .

MATLAB script cho tính toán này được cho dưới đây.



Hình 7.9  $M = 8$  dạng sóng PSK có đường bao không đổi.

**m-file**

```
% MATLAB script for Illustrative Problem 2, Chapter 7.
```

```
echo on
```

```
T=1;
```

```
M=8;
```

```
Es=T/2;
```

```
fc=6/T;
```

% carrier frequency

```
N=100;
```

% number of samples

```
delta_T=T/(N-1);
```

```

t=0:delta_T:T;
u0=sqrt(2*Es/T)*cos(2*pi*fc*t);
u1=sqrt(2*Es/T)*cos(2*pi*fc*t+2*pi/M);
u2=sqrt(2*Es/T)*cos(2*pi*fc*t+4*pi/M);
u3=sqrt(2*Es/T)*cos(2*pi*fc*t+6*pi/M);
u4=sqrt(2*Es/T)*cos(2*pi*fc*t+8*pi/M);
u5=sqrt(2*Es/T)*cos(2*pi*fc*t+10*pi/M);
u6=sqrt(2*Es/T)*cos(2*pi*fc*t+12*pi/M);
u7=sqrt(2*Es/T)*cos(2*pi*fc*t+14*pi/M);
% plotting commands follow
subplot(8,1,1);
plot(t,u0);
subplot(8,1,2);
plot(t,u1);
subplot(8,1,3);
plot(t,u2);
subplot(8,1,4);
plot(t,u3);
subplot(8,1,5);
plot(t,u4);
subplot(8,1,6);
plot(t,u5);
subplot(8,1,7);
plot(t,u6);
subplot(8,1,8);
plot(t,u7);

```

### 7.3.1 GIẢI ĐIỀU CHẾ PHA VÀ TÁCH TÍN HIỆU

Tín hiệu thông dải nhận được từ một kênh AWGN trong một khoảng thời gian truyền tín hiệu có thể biểu diễn được theo

$$r(t) = u_m(t) + n(t) = u_m(t) + n_c(t)\cos 2\pi f_c t - n_s(t)\sin 2\pi f_c t \quad (7.3.11)$$

trong đó  $n_c(t)$  và  $n_s(t)$  là hai thành phần vuông góc của tạp âm cộng.

Tín hiệu nhận được có thể được tính tương quan với  $\psi_1(t)$  và  $\psi_2(t)$  được cho bởi (7.3.9). Các lối ra của hai bộ tương quan có các thành phần tín hiệu bị nhiễu loạn bởi tạp âm mà chúng có thể biểu diễn được theo

$$r = s_m + n = \left( \sqrt{E_s} \cos \frac{2\pi m}{M} + n_c - \sqrt{E_s} \cos \frac{2\pi m}{M} + n_s \right) \quad (7.3.12)$$

trong đó  $n_c$  và  $n_s$  được xác định theo

$$\begin{aligned} n_c &= \frac{1}{2} \int_{-\infty}^{\infty} g_T(t) n_c(t) dt \\ n_s &= \frac{1}{2} \int_{-\infty}^{\infty} g_T(t) n_s(t) dt \end{aligned} \quad (7.3.13)$$

Các thành phần tạp âm vuông góc  $n_c(t)$  và  $n_s(t)$  là các quá trình ngẫu nhiên Gauss kỳ vọng không khôngh tương quan với nhau. Như một hệ quả,  $E(n_c) = E(n_s) = 0$  và  $E(n_c n_s) = 0$ . Phương sai của  $n_c$  và  $n_s$  là

$$E(n_c^2) = E(n_s^2) = \frac{N_0}{2} \quad (7.3.14)$$

Bộ tách tín hiệu tối ưu chiếu véc-tơ tín hiệu nhận được lên từng véc-tơ trong số  $M$  véc-tơ tín hiệu phát có thể có  $\{s_m\}$  và chọn ra véc-tơ tương ứng với hình chiếu lớn nhất. Như thế, chúng ta nhận được các *metric*<sup>\*)</sup> tương quan

$$C(r, s_m) = r \cdot s_m, \quad m=0, 1, 2, \dots, M-1 \quad (7.3.15)$$

Do tất cả các tín hiệu đều có năng lượng như nhau, nên một *metric* của bộ tách tín hiệu tương đương đối với điều chế pha số là việc tính góc pha của véc-tơ tín hiệu nhận được  $r=(r_c, r_s)$  theo

$$\theta_r = \tan^{-1} \frac{r_s}{r_c} \quad (7.3.16)$$

và từ tập  $\{s_m\}$  chọn ra tín hiệu có góc pha gần nhất với  $\theta_r$ .

Xác suất lỗi tại bộ tách tín hiệu đối với điều chế pha trong một kênh AWGN có thể tìm thấy được trong bất kỳ giáo trình thông tin số nào. Do điều chế pha nhị phân đồng nhất với PAM nhị phân nên xác suất lỗi là

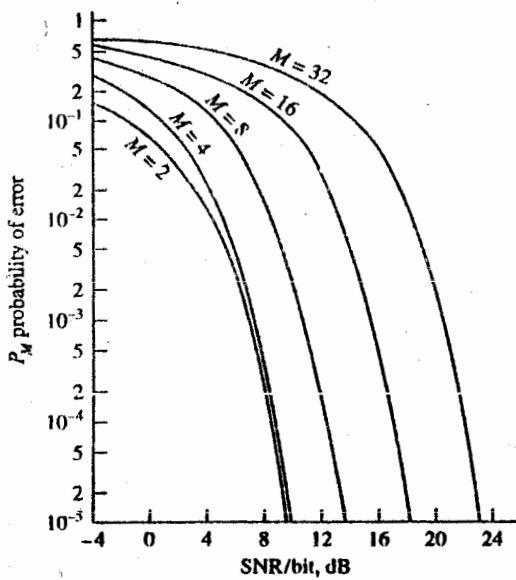
$$P_2 = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (7.3.17)$$

<sup>\*)</sup> *metric* trên một tập X: Một hàm giá trị thực, không âm, xác định trên tích Descartes  $X \times X$ , có các tính chất tương tự các tính chất của khoảng cách giữa hai điểm trên đường thẳng trục thực: khoảng cách giữa hai điểm không phụ thuộc thứ tự của các điểm, khoảng cách giữa hai điểm bằng không khi và chỉ khi hai điểm trùng nhau và khoảng cách giữa hai điểm thì nhỏ hơn hoặc bằng tổng các khoảng cách từ từng điểm tới một điểm thứ ba. Đây là khái niệm khái quát hoá khoảng cách hình học thông thường. (Webster's Encyclopedic Unabridged Dictionary. Gramercy Books, NY, 1993 và Từ điển toán học. Nxb. KH&KT, Hà Nội, 1977 - ND.)

trong đó  $E_b$  là năng lượng trên một bít. Điều chế bốn pha có thể xem được như hai hệ thống điều chế pha nhị phân trên các sóng mang vuông góc (trục giao) với nhau. Hệ quả là xác suất của một bít lỗi thì đúng bằng xác suất một bít lỗi đối với điều chế pha nhị phân. Đối với  $M > 4$ , không có một biểu thức đơn giản dạng tường nào cho xác suất một lỗi bít. Một gần đúng tốt đối với  $P_M$  là

$$P_M \approx 2Q\left(\sqrt{\frac{2E_s}{N_0}} \sin \frac{\pi}{M}\right) \approx 2Q\left(\sqrt{\frac{2kE_b}{N_0}} \sin \frac{\pi}{M}\right) \quad (7.3.18)$$

trong đó  $k = \log_2 M$  là số bít trong một *symbol*. Hình 7.10 minh họa xác suất lỗi *symbol* như là một hàm của SNR  $E_b/N_0$ .



Hình 7.10 Xác suất một lỗi *symbol* đối với PSK  $M$  mức.

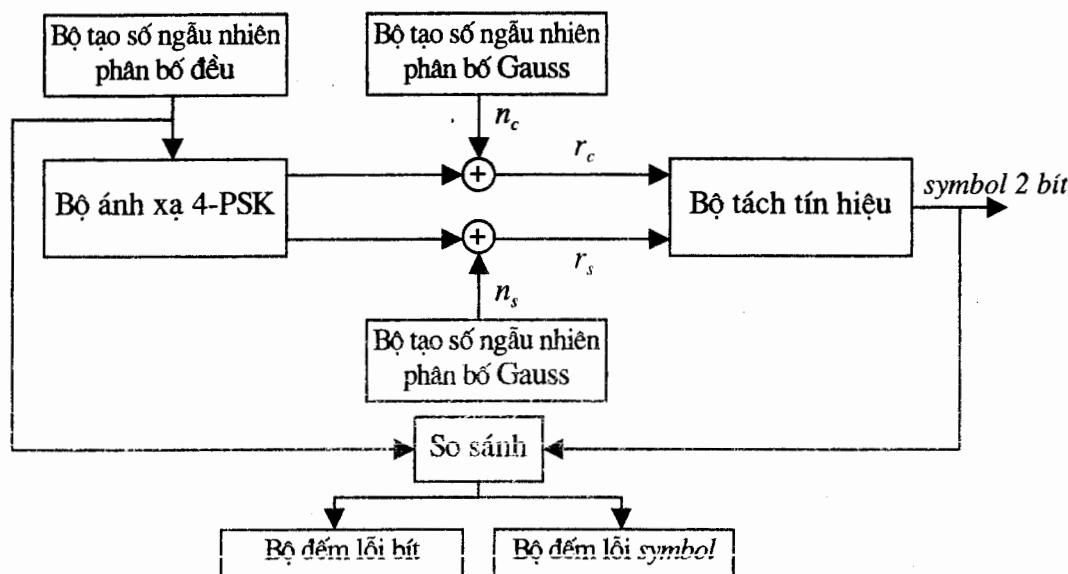
Xác suất lỗi bít tương đương đối với điều chế pha  $M$  mức cũng khó mà tìm ra được do phụ thuộc vào việc ánh xạ các *symbol*  $k$  bít thành các góc pha tín hiệu tương ứng. Khi một mã Gray được sử dụng trong việc ánh xạ này thì các *symbol*  $k$  bít tương ứng với các pha của các tín hiệu lân cận nhau sẽ chỉ khác nhau một bít. Bởi vì các lỗi dễ xảy ra nhất gây bởi tạp âm dẫn đến việc chọn nhầm một pha lân cận thành pha đúng nên hầu hết các lỗi *symbol*  $k$  bít đều chỉ chứa có một lỗi bít đơn. Do đó, xác suất lỗi bít tương đương đối với điều chế pha  $M$  mức làm gần đúng

tốt được theo

$$P_b \approx \frac{1}{k} P_M \quad (7.3.19)$$

### BÀI TẬP MINH HỌA

**Bài tập minh họa 7.3 [Mô phỏng PSK]** Hãy thực hiện một mô phỏng Monte-Carlo về một hệ thống truyền tin PSK có  $M = 4$  mà nó mô hình hóa bộ tách tín hiệu như mô hình tính các *metric* tương quan được cho trong (7.3.15). Mô hình đối với hệ thống được đưa vào mô phỏng được thể hiện trên hình 7.11.



Hình 7.11 Sơ đồ khối của một hệ thống PSK với  $M = 4$  dùng để mô phỏng Monte-Carlo.

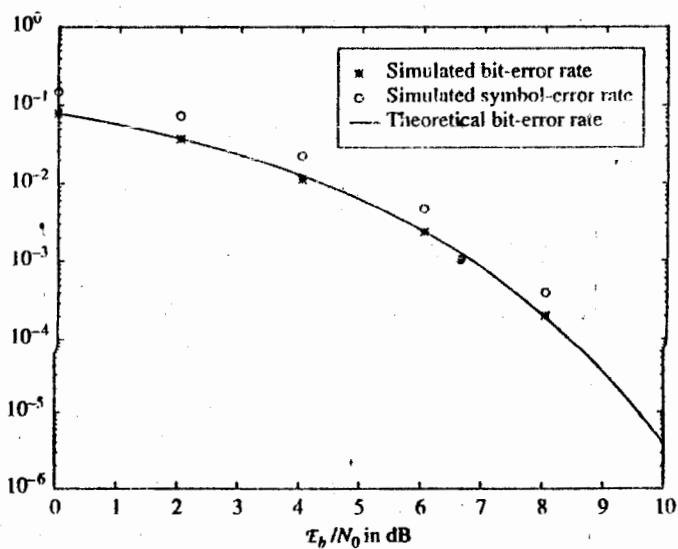
### Lời giải

Như đã chỉ dẫn, chúng ta mô phỏng việc tạo véc-tơ ngẫu nhiên  $r$  được cho bởi (7.3.12) mà nó là lối ra của bộ tương quan tín hiệu và là lối vào tới bộ tách tín hiệu. Chúng ta bắt đầu bằng việc tạo một chuỗi các *symbol* bốn mức (2 bít) mà chúng được ánh xạ thành các điểm tín hiệu 4 pha như đã cho thấy trên hình 7.8 đối với  $M = 4$ . Để hoàn thành nhiệm vụ này, chúng ta sử dụng một bộ tạo số ngẫu nhiên tạo ra một số ngẫu nhiên phân bố đều trên dải  $(0, 1)$ . Dải này được chia nhỏ thành 4 dải đều nhau  $(0, 0.25), (0.25, 0.5), (0.5, 0.75)$  và  $(0.75, 1.0)$ , ở đây các dải nhỏ lần lượt ứng với các cặp bít thông tin 00, 01, 10 và 11. Các cặp bít này được sử

dụng để chọn véc-tơ pha tín hiệu  $s_m$ .

Các thành phần tạp âm cộng  $n_c$  và  $n_s$  là các biến ngẫu nhiên Gauss kỳ vọng 0 độc lập thống kê với phương sai  $\sigma^2$ . Để thuận tiện, chúng ta có thể chuẩn hoá phương sai này thành  $\sigma^2 = 1$  và điều khiển SNR trong tín hiệu thu được bằng cách cho tham số năng lượng tín hiệu  $E_s$  chạy, hoặc ngược lại.

Bộ tách tín hiệu quan sát véc-tơ tín hiệu thu được  $r = s_m + n$  như đã cho trong (7.3.12) và tính toán hình chiếu (tích vô hướng) của  $r$  lên 4 véc-tơ tín hiệu  $s_m$  có thể có. Quyết định của nó dựa trên việc chọn ra điểm tín hiệu tương ứng với hình chiếu lớn nhất. Các quyết định lối ra từ bộ tách tín hiệu được so sánh với các *symbol* đã được truyền đi và các lối *symbol* cũng như các lối bít sẽ được đếm.



Hình 7.12 Chất lượng hoạt động của một hệ thống PSK bốn pha thu được từ mô phỏng Monte-Carlo.

Hình 7.12 minh họa các kết quả của mô phỏng Monte-Carlo đối với việc truyền dẫn của  $N = 10\,000$  *symbol* tại các giá trị khác nhau của tham số SNR  $E_b/N_0$ , trong đó  $E_b = E_s/2$  là năng lượng một bít. Cũng được thể hiện trên hình vẽ 7.12 là cả tỷ lệ lối bít được xác định theo  $P_b \approx P_M/2$ , lẫn xác suất lối lý thuyết tương ứng được cho bởi (7.3.18).

Các MATLAB script đối với mô phỏng Monte-Carlo này được cho dưới đây.

### m-file

```
% MATLAB script for Illustrative Problem 3, Chapter 7.  
echo on  
SNRindB1=0:2:10;  
SNRindB2=0:0.1:10;  
for i=1:length(SNRindB1),  
    [pb,ps]=cm_sm32(SNRindB1(i));           % simulated bit and symbol error rates  
    smld_bit_err_prb(i)=pb;  
    smld_symbol_err_prb(i)=ps;  
    echo off ;  
end;  
echo on;  
for i=1:length(SNRindB2),  
    SNR=exp(SNRindB2(i)*log(10)/10);      % signal to noise ratio  
    theo_err_prb(i)=Qfunct(sqrt(2*SNR));   % theoretical bit error rate  
    echo off ;  
end;  
echo on ;  
% Plotting commands follow  
semilogy(SNRindB1,smld_bit_err_prb,'*');  
hold  
semilogy(SNRindB1,smld_symbol_err_prb,'o');  
semilogy(SNRindB2,theo_err_prb);
```

### m-file

```
function [pb,ps]=cm_sm32(snr_in_dB)  
% [pb,ps]=cm_sm32(snr_in_dB)  
% CM_SM32 finds the probability of bit error and symbol error for the  
% given value of snr_in_dB, signal to noise ratio in dB.  
N=10000;  
E=1;                                % energy per symbol  
snr=10^(snr_in_dB/10);                % signal to noise ratio  
sgma=sqrt(E/snр)/2;                  % noise variance  
% the signal mapping  
s00=[1 0];  
s01=[0 1];  
s11=[-1 0];  
s10=[0 -1];  
% generation of the data source
```

```

for i=1:N,
    temp=rand; % a uniform random variable between 0 and 1
    if (temp<0.25),
        dsource1(i)=0; % with probability 1/4, source output is "00"
        dsource2(i)=0;
    elseif (temp<0.5),
        dsource1(i)=0; % with probability 1/4, source output is "01"
        dsource2(i)=1;
    elseif (temp<0.75),
        dsource1(i)=1; % with probability 1/4, source output is "10"
        dsource2(i)=0;
    else
        dsource1(i)=1; % with probability 1/4, source output is "11"
        dsource2(i)=1;
    end;
end;
% detection and the probability of error calculation
numofsymbolerror=0;
numofbiterror=0;
for i=1:N,
    % the received signal at the detector, for the ith symbol, is:
    n(1)=gngauss(sgma);
    n(2)=gngauss(sgma);
    if ((dsource1(i)==0) & (dsource2(i)==0)),
        r=s00+n;
    elseif ((dsource1(i)==0) & (dsource2(i)==1)),
        r=s01+n;
    elseif ((dsource1(i)==1) & (dsource2(i)==0)),
        r=s10+n;
    else
        r=s11+n;
    end;
    % The correlation metrics are computed below
    c00=dot(r,s00);
    c01=dot(r,s01);
    c10=dot(r,s10);
    c11=dot(r,s11);
    % The decision on the ith symbol is made next
    c_max=max([c00 c01 c10 c11]);
    if (c00==c_max),
        decis1=0; decis2=0;
    elseif (c01==c_max),

```

```

decis1=0; decis2=1;
elseif (c10==c_max),
  decis1=1; decis2=0;
else
  decis1=1; decis2=1;
end;
% increment the error counter, if the decision is not correct
symbolerror=0;
if (decis1~=dsource1(i)),
  numofbiterror=numofbiterror+1;
  symbolerror=1;
end;
if (decis2~=dsource2(i)),
  numofbiterror=numofbiterror+1;
  symbolerror=1;
end;
if (symbolerror==1),
  numofsymbolerror = numofsymbolerror+1;
end;
end;
ps=numofsymbolerror/N;           % since there are totally N symbols
pb=numofbiterror/(2*N);         % since 2N bits are transmitted

```

### 7.3.2 ĐIỀU CHẾ VÀ GIẢI ĐIỀU CHẾ PHA VI SAI

Việc giải điều chế tín hiệu sóng mang được điều chế pha, như đã được mô tả trên đây, đòi hỏi rằng các thành phần pha  $\psi_1(t)$  và  $\psi_2(t)$  của sóng mang (ở máy thu) phải được khoá pha với tín hiệu được điều chế sóng mang thu được. Nói chung, điều này nghĩa là máy thu nhất thiết phải ước lượng lượng dịch pha sóng mang của tín hiệu thu được gây bởi một trễ truyền dẫn qua kênh và bù lượng dịch pha này trong lúc tính tương quan chéo giữa tín hiệu thu được với hai thành phần tham chiếu  $\psi_1(t)$  và  $\psi_2(t)$ . Việc ước lượng lượng dịch pha sóng mang thường được thực hiện bằng cách sử dụng một mạch vòng khoá pha (PLL: *Phase-Locked Loop*). Như thế, chúng ta có được giải điều chế pha kết hợp (*coherent*).

Một kiểu điều chế pha sóng mang khác là điều chế pha vi sai, trong đó dữ liệu được truyền đi được mã hoá vi sai trước khi được đưa tới bộ điều chế. Trong mã hoá vi sai, thông tin được tải bằng các lượng dịch pha tương đối so với khoảng

thời gian tín hiệu trước đó. Thí dụ, trong điều chế pha nhị phân, bít thông tin 1 có thể được truyền đi bằng cách dịch pha của sóng mang đi  $180^\circ$  so với pha sóng mang trong khoảng tín hiệu trước đó, trong khi đó bít thông tin 0 được truyền đi bằng cách dịch pha của sóng mang đi  $0^\circ$  so với pha sóng mang trong khoảng truyền một tín hiệu trước đó. Trong điều chế bốn pha, các lượng dịch pha tương đối giữa các khoảng thời gian truyền các tín hiệu kế tiếp nhau là  $0^\circ, 90^\circ, 180^\circ$  và  $270^\circ$ , tương ứng lần lượt với các cặp bít thông tin 00, 01, 10 và 11. Việc tổng quát hoá việc mã vi sai đối với  $M > 4$  thấy được hoàn toàn dễ dàng. Các tín hiệu được điều chế pha có được từ quá trình mã hoá này được mô tả là *được mã hoá vi sai*.

Việc giải điều chế và tách tín hiệu của tín hiệu được điều chế pha có mã hoá vi sai có thể thực hiện được như sau. Pha của tín hiệu nhận được  $\theta_r = \tan^{-1}(r_s/r_c)$  tại bộ tách tín hiệu được ánh xạ thành một trong  $M$  pha tín hiệu phát có thể có  $\{\theta_m\}$  mà nó gần nhất với  $\theta_r$ . Tiếp sau bộ tách tín hiệu là một bộ so pha tương đối, thực hiện so sánh các pha của tín hiệu đã tách được trên hai khoảng tín hiệu liên tiếp nhằm tách lấy thông tin đã phát.

Chúng ta thấy rằng việc giải điều chế tín hiệu điều chế pha có mã hoá vi sai không đòi hỏi phải ước lượng pha của sóng mang. Để nói một cách chi tiết, ta hãy giả sử rằng chúng ta giải điều chế tín hiệu được mã hoá vi sai bằng cách tính tương quan chéo  $r(t)$  với  $g_T(t)\cos 2\pi f_c t$  và  $-g_T(t)\sin 2\pi f_c t$ . Tại khoảng truyền tín hiệu thứ  $k$ , hai thành phần của lối ra bộ giải điều chế có thể biểu diễn được dưới dạng phức theo

$$r_k = \sqrt{E_s} e^{j(\theta_k - \phi)} + n_k \quad (7.3.20)$$

trong đó  $\theta_k$  là góc pha của tín hiệu phát tại khoảng tín hiệu thứ  $k$ ,  $\phi$  là pha của sóng mang còn  $n_k = n_{kc} + jn_{ks}$  là tạp âm. Tương tự như vậy, véc-tơ tín hiệu thu được tại lối ra của bộ giải điều chế trong khoảng tín hiệu trước đó là đại lượng phức

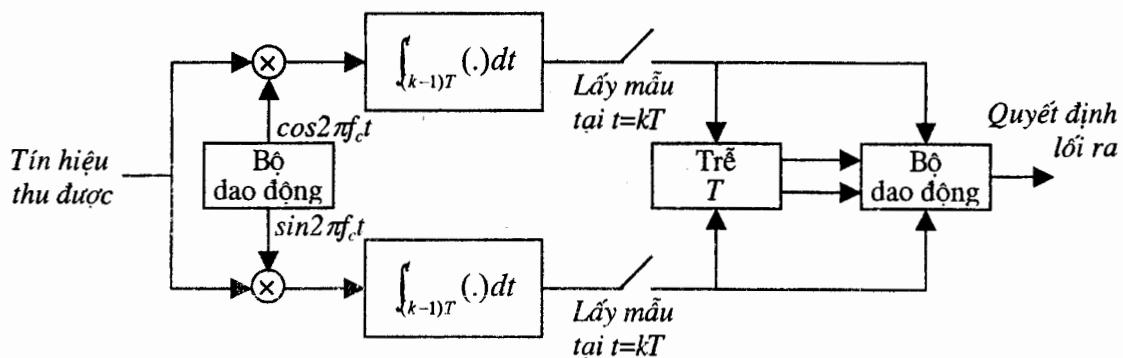
$$r_{k-1} = \sqrt{E_s} e^{j(\theta_{k-1} - \phi)} + n_{k-1} \quad (7.3.21)$$

Biến quyết định đối với bộ tách pha là độ chênh lệch pha giữa hai số phức này. Tương đương vậy, chúng ta có thể chiếu  $r_k$  lên  $r_{k-1}$  và sử dụng góc pha của số

phức kết quả của phép chiếu, tức là

$$r_k r_{k-1}^* = E_s e^{j(\theta_k - \theta_{k-1})} + \sqrt{E_s} e^{j(\theta_k - \phi)} n_{k-1} + \sqrt{E_s} e^{-j(\theta_{k-1} - \phi)} n_k + n_k n_{k-1}^* \quad (7.3.22)$$

mà khi không có tạp âm, nó cho chúng ta sai lệch pha  $\theta_k - \theta_{k-1}$ . Như vậy thì, giá trị trung bình của  $r_k r_{k-1}^*$  độc lập với pha của sóng mang. Việc truyền tín hiệu PSK có mã hoá vi sai được điều chế và giải điều chế như đã mô tả trên đây được gọi là PSK vi sai (DPSK: Differential PSK). Việc giải điều chế và tách tín hiệu của DPSK được minh họa trên hình 7.13.



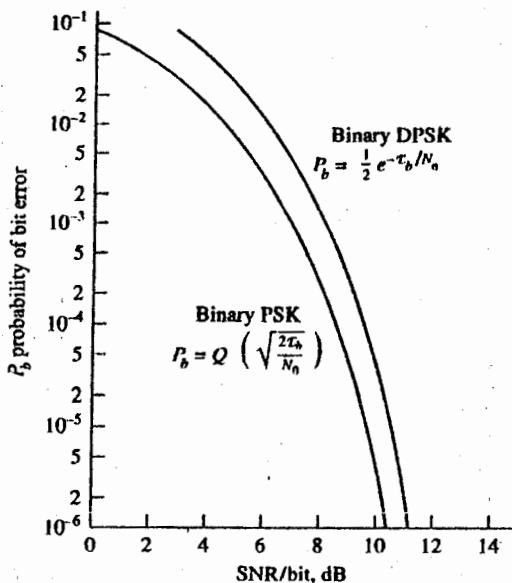
Hình 7.13 Sơ đồ khối của bộ giải điều chế DPSK.

Xác suất lỗi đối với DPSK trong một kênh AWGN thì tìm được tương đối đơn giản với trường hợp điều chế pha nhị phân ( $M = 2$ ). Kết quả là

$$P_2 = \frac{1}{2} e^{-E_b / N_0} \quad (7.3.23)$$

Đồ thị của biểu thức (7.2.23) được trình bày trên hình 7.14. Cũng được thể hiện trên hình này là xác suất lỗi đối với PSK nhị phân. Chúng ta quan sát thấy rằng tại các xác suất lỗi dưới  $10^{-4}$ , sai lệch về SNR giữa PSK nhị phân và DPSK nhị phân nhỏ hơn 1 dB.

Đối với  $M > 2$ , chất lượng về xác suất lỗi của một bộ giải điều chế và bộ tách tín hiệu DPSK cực kỳ khó đánh giá được. Khó khăn chính gấp phải là trong việc xác định hàm mật độ xác suất lỗi đối với pha của biến ngẫu nhiên  $r_k r_{k-1}^*$ , được cho trong (7.3.22). Tuy nhiên, một xấp xỉ đối với chất lượng hoạt động của DPSK lại dễ dàng nhận được như sẽ trình bày ngay sau đây.



Hình 7.14 Xác suất lỗi đối với PSK nhị phân và DPSK nhị phân.

Không làm mất đi tính tổng quát, ta hãy giả sử rằng sai lệch pha  $\theta_k - \theta_{k-1} = 0$ .

Hơn thế nữa, các thừa số mũ  $e^{-j(\theta_{k-1}-\phi)}$  và  $e^{j(\theta_k-\phi)}$  trong (7.3.22) có thể hấp thụ được vào trong các thành phần tạp âm Gauss  $n_{k-1}$  và  $n_k$  mà không làm thay đổi các đặc tính thống kê của chúng. Do vậy,  $r_k r_{k-1}^*$  trong (7.3.22) có thể biểu diễn được theo

$$r_k r_{k-1}^* = E_s + \sqrt{E_s} (n_k + n_{k-1}^*) + n_k n_{k-1}^* \quad (7.3.24)$$

Điều phức tạp trong việc xác định hàm mật độ xác suất lỗi của pha là số hạng  $n_k n_{k-1}^*$ . Tuy nhiên, tại các SNR thực tế mà chúng ta quan tâm, số hạng  $n_k n_{k-1}^*$  khá nhỏ so với số hạng tạp âm vượt trội  $\sqrt{E_s} (n_k + n_{k-1}^*)$ . Nếu chúng ta bỏ qua số hạng  $n_k n_{k-1}^*$  đồng thời chúng ta cũng chuẩn hoá  $r_k r_{k-1}^*$  bằng cách chia tất cả cho  $\sqrt{E_s}$  thì tập các *metric* quyết định mới trở thành

$$\begin{aligned} x &= \sqrt{E_s} + \operatorname{Re}(n_k + n_{k-1}^*) \\ y &= \operatorname{Im}(n_k + n_{k-1}^*) \end{aligned} \quad (7.3.25)$$

Các biến  $x$  và  $y$  là các biến ngẫu nhiên Gauss không tương quan có cùng phương sai  $\sigma^2$ . Góc pha là

$$\theta_r = \tan^{-1} \frac{y}{x} \quad (7.3.26)$$

Tại bước này, chúng ta có một bài toán hoàn toàn như bài toán đối với giải điều chế kết hợp pha. Điểm khác biệt duy nhất là phương sai của tạp âm bây giờ lớn hơn gấp hai lần so với trường hợp PSK. Như thế, chúng ta kết luận được rằng chất lượng hoạt động của DPSK kém hơn chất lượng hoạt động của PSK 3 dB. Kết quả này khá tốt đối với  $M \geq 4$ , song nó lại là kết quả bi quan (kém hơn thực tế) đối với  $M = 2$  theo nghĩa rằng tổn thất trong DPSK nhị phân so với PSK nhị phân thật ra thì nhỏ hơn 3 dB tại những giá trị SNR lớn.

### BÀI TẬP MINH HỌA

**Bài tập minh họa 7.4 [Bộ mã hoá DPSK]** Hãy thực hiện một bộ mã hoá vi sai cho trường hợp DPSK có  $M=8$ .

#### Lời giải

Các điểm tín hiệu thì giống như các điểm tín hiệu đối với PSK được trình bày trên hình 7.8. Tuy nhiên, đối với DPSK thì các điểm tín hiệu biểu thị sự thay đổi pha tương đối so với pha của các điểm tín hiệu đã phát ngay trước đó.

MATLAB script cho việc thực hiện bộ mã hoá vi sai được cho dưới đây.

#### m-file

```
% MATLAB script for Illustrative Problem 4, Chapter 7.
mapping=[0 1 3 2 7 6 4 5]; % For Gray mapping
M=8;
E=1;
sequence=[0 1 0 0 1 1 0 0 1 1 1 1 0 0 0];
[e]=cm_dpske(E,M,mapping,sequence); % e is the differential encoder output
```

### BÀI TẬP MINH HỌA

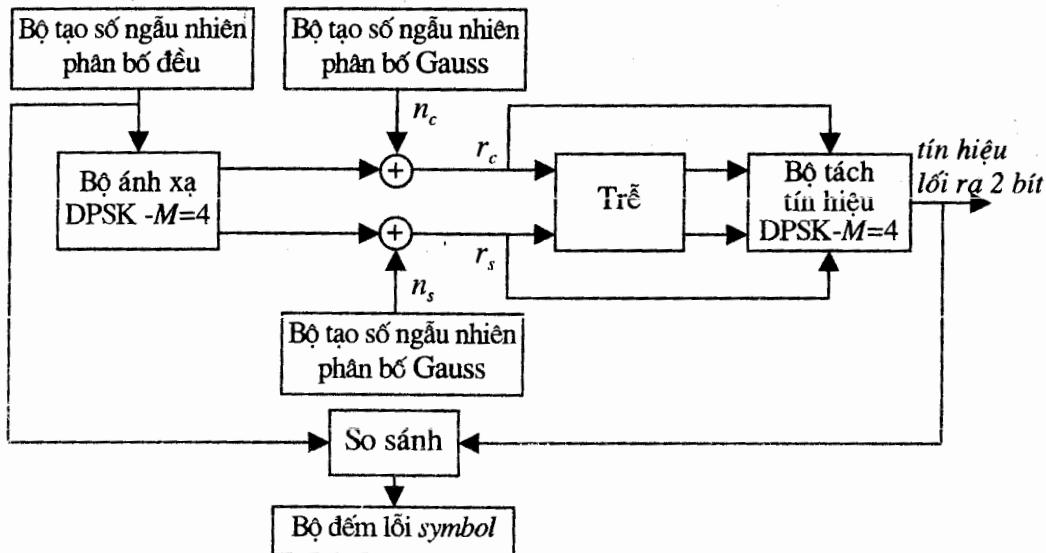
**Bài tập minh họa 7.5** Hãy tiến hành một mô phỏng Monte-Carlo của một hệ thống truyền tin 4-DPSK. Mô hình đối với hệ thống được mô phỏng được cho trên hình 7.15.

#### Lời giải

Bộ tạo số ngẫu nhiên phân bố đều được sử dụng để tạo các cặp bit [00, 01, 10, 11]

như đã được mô tả trong bài tập minh họa 7.3. Từng *symbol* 2 bít được ánh xạ thành một trong 4 điểm tín hiệu  $s_m = [\cos \pi m/2 \quad \sin \pi m/2]$ ,  $m = 0, 1, 2, 3$  bằng một bộ mã hoá vi sai. Hai bộ tạo số ngẫu nhiên Gauss được sử dụng để tạo ra các thành phần tạp âm  $[n_c \quad n_s]$ . Khi đó véc-tơ tín hiệu-cộng-tạp âm thu được là

$$\mathbf{r} = \left[ \cos \frac{\pi m}{2} + n_c \quad \sin \frac{\pi m}{2} + n_s \right] = [r_c \quad r_s]$$



Hình 7.15 Sơ đồ khái niệm của hệ thống DPSK với  $M = 4$ , dùng cho mô phỏng Monte-Carlo.

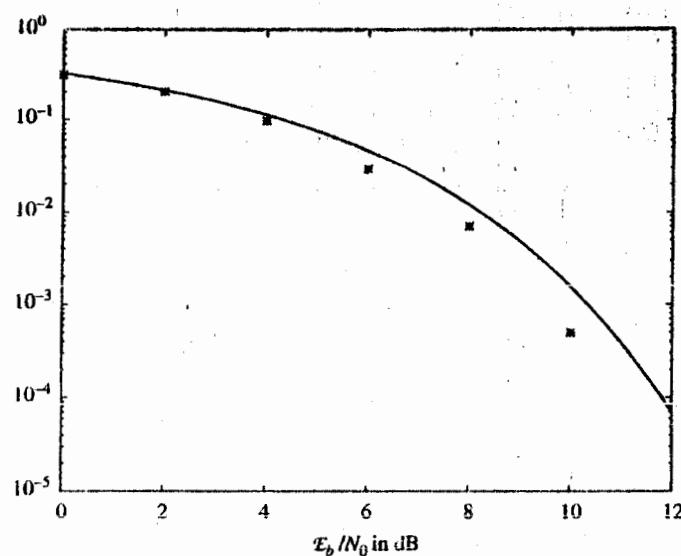
Bộ tách tín hiệu vi sai về căn bản tính sai lệch pha giữa  $r_k$  và  $r_{k-1}$ . Về mặt toán học, tính toán này có thể thực hiện được như trong (7.3.22), nghĩa là

$$\begin{aligned} r_k r_{k-1}^* &= (r_{ck} + j r_{sk})(r_{ck-1} - j r_{sk-1}) \\ &= r_{ck} r_{ck-1} + r_{sk} r_{sk-1} + j(r_{sk} r_{ck-1} - r_{ck} r_{sk-1}) \\ &= x_k + j y_k \end{aligned}$$

còn  $\theta_k = \tan^{-1}(y_k/x_k)$  là sai lệch pha. Giá trị của  $\theta_k$  được so sánh với các giá trị sai lệch pha có thể có  $\{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$ , và một quyết định được đưa ra về góc pha nào gần nhất với  $\theta_k$ . Góc pha tách được sau đó được ánh xạ thành cặp bít thông tin. Bộ đếm lỗi đếm các lỗi *symbol* trong chuỗi đã tách được.

Hình 7.16 minh họa các kết quả mô phỏng Monte-Carlo đối với việc truyền dẫn  $N = 10\,000$  *symbol* tại các giá trị khác nhau của tham số SNR  $E_b/N_0$ , trong đó

$E_b = E_s/2$  là năng lượng của một bít. Cùng được trình bày trên hình 7.16 là giá trị lý thuyết của lỗi *symbol* dựa trên xấp xỉ rằng số hạng  $n_k n_{k-1}^*$  bỏ qua được. Chúng ta thấy được từ hình 7.16 rằng việc xấp xỉ dẫn đến một giới hạn chặn trên đối với xác suất lỗi.



Hình 7.16 Chất lượng của hệ thống DPSK bốn pha thu được từ mô phỏng Monte-Carlo.  
(Đường liền là một giới hạn chặn trên dựa trên xấp xỉ với việc bỏ qua thành phần tạp âm  $n_k n_{k-1}^*$ )

Các MATLAB *script* cho mô phỏng Monte-Carlo này được cho dưới đây.

### m-file

```
% MATLAB script for Illustrative Problem 5, Chapter 7.
echo on
SNRindB1=0:2:12;
SNRindB2=0:0.1:12;
for i=1:length(SNRindB1),
    smld_err_prb(i)=cm_sm34(SNRindB1(i));      % simulated error rate
    echo off ;
end;
echo on ;
for i=1:length(SNRindB2),
    SNR=exp(SNRindB2(i)*log(10)/10);          % signal to noise ratio
    theo_err_prb(i)=2*Qfunct(sqrt(SNR));        % theoretical symbol error rate
    echo off ;
end;
```

```

echo on ;
% Plotting commands follow
semilogy(SNRindB1,smld_err_prb,'*');
hold
semilogy(SNRindB2,theo_err_prb);

```

### m-file

```

function [p]=cm_sm34(snr_in_dB)
% (p)=cm_sm34(snr_in_dB)
%           CM_SM34 finds the probability of error for the given
%           value of snr_in_dB, signal to noise ratio in dB.
N=10000;
E=1;                                % energy per symbol
snr=10^(snr_in_dB/10);                % signal to noise ratio
sgma=sqrt(E/(4*snr));                % noise variance
% generation of the data source follows
for i=1:2*N,
    temp=rand;                      % a uniform random variable between 0 and 1
    if (temp<0.5),
        dsource(i)=0;                % with probability 1/2, source output is "0"
    else
        dsource(i)=1;                % with probability 1/2, source output is "1"
    end;
end;
% Differential encoding of the data source follows
mapping=[0 1 3 2];
M=4;
[diff_enc_output] = cm_dpske(E,M,mapping,dsource);
% received signal is then
for i=1:N,
    [n(1) n(2)]=gngauss(sgma);
    r(i,:)=diff_enc_output(i,:)+n;
end;
% detection and the probability of error calculation
numoferr=0;
prev_theta=0;
for i=1:N,
    theta=angle(r(i,1)+j*r(i,2));
    delta_theta=mod(theta-prev_theta,2*pi);
    if ((delta_theta<pi/4) | (delta_theta>7*pi/4)),
        decis=[0 0];
    else

```

```

elseif (delta_theta<3*pi/4),
decis=[0 1];
elseif (delta_theta<5*pi/4)
decis=[1 1];
else
decis=[1 0];
end;
prev_theta=theta;
% increase the error counter, if the decision is not correct
if ((decis(1)~=dsource(2*i-1)) | (decis(2)~=dsource(2*i))),
numoferr=numoferr+1;
end;
end;
p=numoferr/N;

```

**m-file**

```

function [enc_comp] = cm_dpske(E,M,mapping,sequence);
% [enc_comp] = cm_dpske(E,M,mapping,sequence)
% CM_DPSKE differentially encodes a sequence.
% E is the average energy, M is the number of constellation points and
% mapping is the vector defining how the constellation points are
% allocated. Finally, ``sequence'' is the uncoded binary data sequence.
k=log2(M);
N=length(sequence);
% If N is not divisible by k, append zeros, so that it is...
remainder=rem(N,k);
if (remainder~=0),
for i=N+1:N+k-remainder,
sequence(i)=0;
end;
N=N+k-remainder;
end;
theta=0; % Initially, assume that theta=0
for i=1:k:N,
index=0;
for j=i:i+k-1,
index=2*index+sequence(j);
end;
index=index+1;
theta=mod(2*pi*mapping(index)/M+theta,2*pi);
enc_comp((i+k-1)/k,1)=sqrt(E)*cos(theta);

```

```

enc_comp((i+k-1)/k,2)=sqrt(E)*sin(theta);
end;

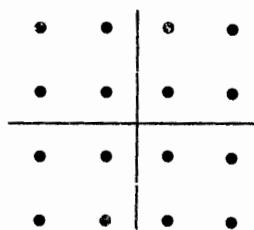
```

## 7.4 ĐIỀU CHẾ BIÊN ĐỘ VUÔNG GÓC

Một tín hiệu điều chế biên độ vuông góc (QAM: *Quadrature-Amplitude-Modulated signal*) sử dụng hai sóng mang vuông góc là  $\cos 2\pi f_c t$  và  $\sin 2\pi f_c t$ , mỗi một trong chúng được điều chế bởi một chuỗi độc lập các bít thông tin. Các dạng sóng tín hiệu được truyền đi có dạng

$$u_m(t) = A_{mc}g_T(t) \cos 2\pi f_c t + A_{ms}g_T(t) \sin 2\pi f_c t, \quad m=1, 2, \dots, M \quad (7.4.1)$$

trong đó  $\{A_{mc}\}$  và  $\{A_{ms}\}$  là các tập các mức biên độ nhận được bằng cách ánh xạ các chuỗi  $k$  bít thành các biên độ tín hiệu. Thí dụ, hình 7.17 minh họa một biểu đồ sao tín hiệu 16-QAM mà nó nhận được bằng cách điều chế biên độ từng sóng mang bằng 4-PAM. Nói chung, các biểu đồ sao tín hiệu hình vuông được sinh ra khi từng sóng mang một trong hai sóng mang được điều chế bởi PAM.



Hình 7.17 Biểu đồ sao tín hiệu 16-QAM.

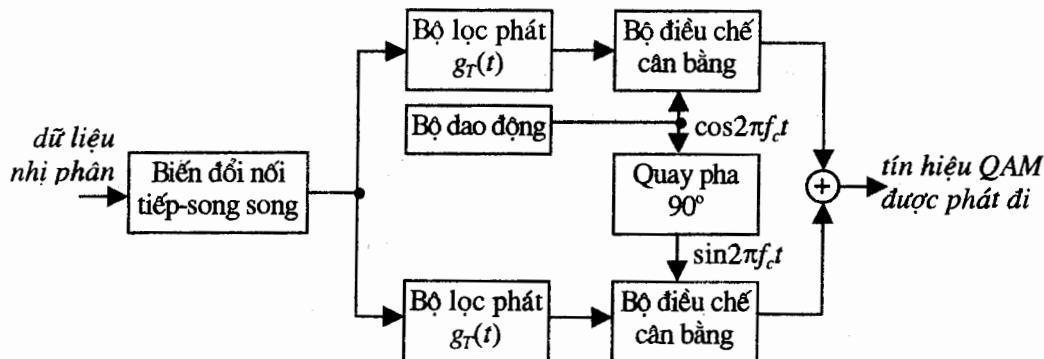
Tổng quát hơn, QAM có thể xem được như một dạng hỗn hợp của điều chế biên độ số và điều chế pha số. Như thế các dạng sóng tín hiệu QAM được truyền đi có thể biểu diễn được theo

$$u_{mn}(t) = A_{m}g_T(t) \cos(2\pi f_c t + \theta_n), \quad m = 1, 2, \dots, M_1, \quad n = 1, 2, \dots, M_2 \quad (7.4.2)$$

Nếu  $M_1 = 2^{k_1}$  còn  $M_2 = 2^{k_2}$ , thì phương pháp điều chế biên độ và pha kết hợp dẫn đến việc truyền dẫn đồng thời  $k_1 + k_2 = \log_2 M_1 M_2$  digit nhị phân xảy ra với một tốc độ *symbol* là  $R_b/(k_1+k_2)$ . Hình 7.18 minh họa sơ đồ khối chức năng của một bộ điều chế QAM. Rõ ràng là việc biểu diễn hình học các tín hiệu được cho bởi (7.4.1)

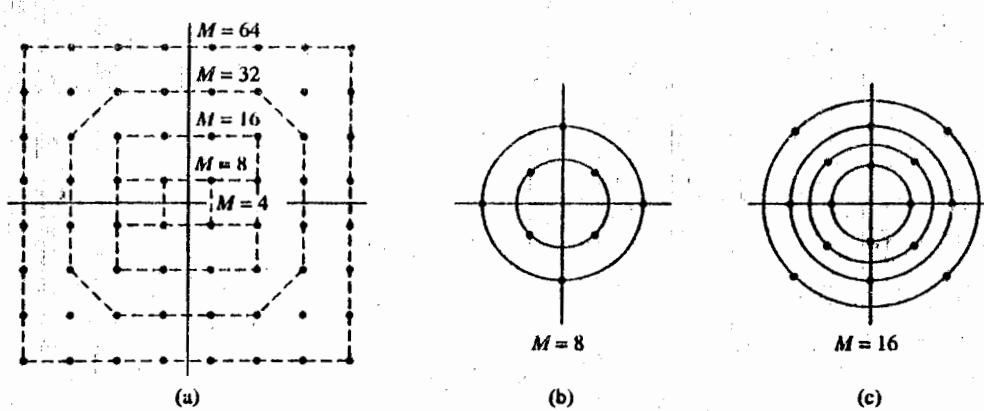
và (7.4.2) là biểu diễn bằng các véc-tơ tín hiệu hai chiều có dạng

$$\mathbf{s}_m = \begin{pmatrix} \sqrt{E_s} A_{mc} \\ \sqrt{E_s} A_{ms} \end{pmatrix}, \quad m=1, 2, \dots, M \quad (7.4.3)$$



Hình 7.18 Sơ đồ khối chức năng một bộ điều chế QAM.

Các thí dụ về các biểu đồ sao khôn gian tín hiệu đối với QAM được thể hiện trên hình vẽ 7.19. Hãy chú ý rằng 4-QAM thì trùng với 4-PSK.



Hình 7.19 Các biểu đồ sao tín hiệu QAM (a) hình vuông và (b, c) hình tròn.

#### 7.4.1 GIẢI ĐIỀU CHẾ VÀ TÁCH TÍN HIỆU QAM

Ta hãy giả sử rằng một lượng dịch pha sóng mang được đưa vào trong quá trình truyền dẫn tín hiệu qua kênh. Thêm vào đó, tín hiệu thu được bị nhiễu loạn bởi tạp âm cộng Gauss. Vì vậy,  $r(t)$  có thể biểu diễn được theo

$$r(t) = A_{mc}g_T(t)\cos(2\pi f_c t + \phi) + A_{ms}g_T(t)\sin(2\pi f_c t + \phi) + n(t) \quad (7.4.4)$$

trong đó  $\phi$  là lượng dịch pha sóng mang và

$$n(t) = n_c(t) \cos 2\pi f_c t - n_s(t) \sin 2\pi f_c t$$

Tín hiệu thu được được tính tương quan với hai hàm cơ sở trực giao đã được dịch pha

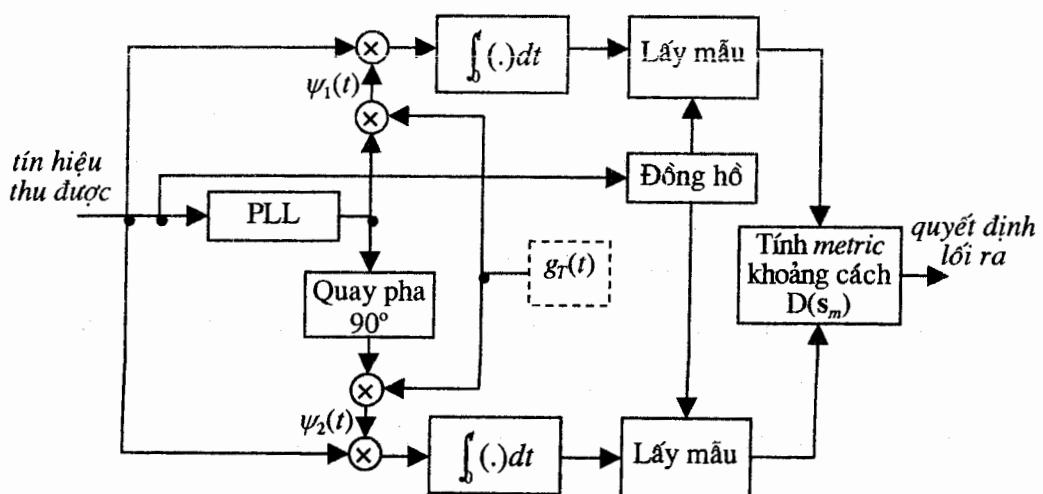
$$\begin{aligned}\psi_1(t) &= g_T(t) \cos(2\pi f_c t + \phi) \\ \psi_2(t) &= g_T(t) \sin(2\pi f_c t + \phi)\end{aligned}\quad (7.4.5)$$

nó như được minh họa trên hình 7.20, còn các lối ra của các bộ tương quan được lấy mẫu rồi được đưa tới bộ tách tín hiệu. Mạch vòng khoá pha (PLL) trên hình 7.20 ước lượng lượng dịch pha mang  $\phi$  của tín hiệu thu được và bù lượng dịch pha này bằng cách dịch pha  $\psi_1(t)$  và  $\psi_2(t)$  như đã chỉ ra trong (7.4.5). Đồng hồ trên hình 7.20 được giả thiết là được đồng bộ với tín hiệu thu được sao cho các lối ra của các bộ tương quan được lấy mẫu tại các thời điểm lấy mẫu chính xác. Với các điều kiện này, các lối ra từ hai bộ tương quan là

$$\begin{aligned}r_c &= A_{mc} + n_c \cos \phi - n_s \sin \phi \\ r_s &= A_{ms} + n_c \sin \phi + n_s \cos \phi\end{aligned}\quad (7.4.6)$$

trong đó

$$\begin{aligned}n_c &= \frac{1}{2} \int_0^T n_c(t) g_T(t) dt \\ n_s &= \frac{1}{2} \int_0^T n_s(t) g_T(t) dt\end{aligned}\quad (7.4.7)$$



Hình 7.20 Giải điều chế và tách các tín hiệu QAM.

Các thành phần tạp âm là các biến ngẫu nhiên Gauss kỳ vọng 0, không btương quan, với phương sai  $N_0/2$ .

Bộ tách tín hiệu tối ưu tính các *metric* khoảng cách

$$D(\mathbf{r}, \mathbf{s}_m) = |\mathbf{r} - \mathbf{s}_m|^2, \quad m = 1, 2, \dots, M \quad (7.4.8)$$

trong đó  $\mathbf{r} = (r_c, r_s)$  còn  $\mathbf{s}_m$  được cho bởi (7.4.3).

#### 7.4.2 XÁC SUẤT LỖI ĐỐI VỚI QAM TRONG MỘT KÊNH AWGN

Trong mục này, chúng ta xem xét chất lượng hoạt động của các hệ thống QAM áp dụng biểu đồ sao tín hiệu hình vuông. Các biểu đồ sao tín hiệu QAM hình vuông có ưu điểm rõ rệt là dễ tạo ra được do các tín hiệu PAM được đóng lên các sóng mang vuông góc mà mỗi một trong chúng gồm  $\sqrt{M} = 2^{k/2}$  điểm tín hiệu. Do các tín hiệu trên các thành phần trực giao về pha được phân cách với nhau một cách hoàn hảo nhờ tách tín hiệu kết hợp, xác suất lỗi đối với QAM dễ dàng xác định được từ xác suất lỗi đối với PAM. Đặc biệt, xác suất của một quyết định đúng đối với hệ thống QAM  $M$  mức là

$$P_C = (1 - P_{\sqrt{M}})^2 \quad (7.4.9)$$

trong đó  $P_{\sqrt{M}}$  là xác suất lỗi của một PAM  $\sqrt{M}$  mức với một nửa công suất trung bình trên mỗi một tín hiệu vuông góc của hệ thống QAM tương đương. Bằng cách biến đổi thích hợp xác suất lỗi đối với PAM  $M$  mức, chúng ta nhận được

$$P_{\sqrt{M}} = 2 \left( 1 - \frac{1}{\sqrt{M}} \right) Q \left( \sqrt{\frac{3}{M-1}} \frac{E_{av}}{N_0} \right) \quad (7.4.10)$$

trong đó  $E_{av}/N_0$  là SNR trung bình trên *symbol*. Do đó, xác suất lỗi của một *symbol* đối với QAM  $M$  mức là

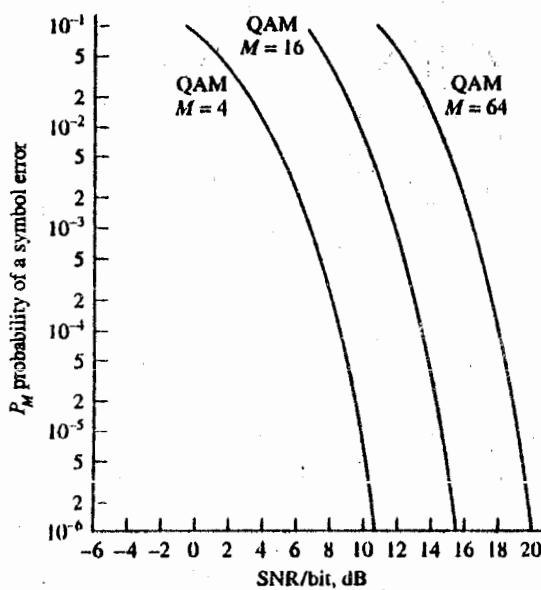
$$P_M = 1 - (1 - P_{\sqrt{M}})^2 \quad (7.4.11)$$

Chúng ta hãy chú ý rằng kết quả này là chính xác đối với  $M=2^k$  khi  $k$  chẵn. Mặt khác, khi  $k$  lẻ, không có một hệ thống PAM  $\sqrt{M}$  mức tương đương. Tuy thế, không có vấn đề gì vì việc xác định tỷ lệ lỗi đối với một tập tín hiệu vuông góc thì

khá dễ. Nếu chúng ta áp dụng bộ tách tín hiệu tối ưu đưa ra các quyết định của nó dựa trên các *metric* khoảng cách tối ưu được cho bởi (7.4.8) thì tương đối dễ để chứng tỏ rằng xác suất lỗi *symbol* được chặn trên chặt theo

$$P_M \leq 1 - \left[ 1 - 2Q\left(\sqrt{\frac{3E_{av}}{(M-1)N_0}}\right) \right]^2 \leq 4Q\left(\sqrt{\frac{3kE_{avb}}{(M-1)N_0}}\right) \quad (7.4.12)$$

đối với bất kỳ  $k \geq 1$  nào, trong đó  $E_{avb}/N_0$  là SNR trung bình trên một bít. Xác suất lỗi một *symbol* được vẽ trên hình 7.21 như một hàm của SNR trung bình trên một bít.



Hình 7.21 Xác suất của một *symbol* lỗi đối với QAM.

## BÀI TẬP MINH HỌA

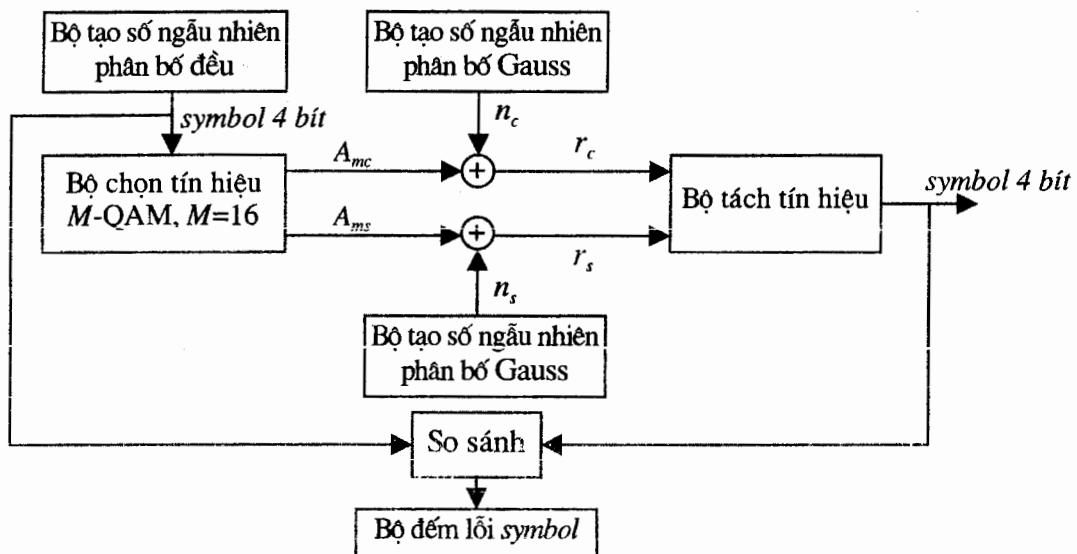
Bài tập minh họa 7.6 [Mô phỏng QAM] Hãy thực hiện một mô phỏng Monte-Carlo của một hệ thống truyền tin 16-QAM sử dụng biểu đồ sao tín hiệu hình vuông. Mô hình của hệ thống để mô phỏng được trình bày trên hình 7.22.

### Lời giải

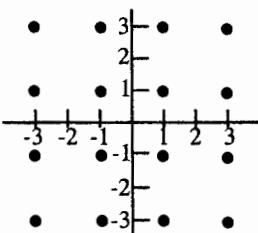
Bộ tạo số ngẫu nhiên phân bố đều được sử dụng để tạo ra chuỗi các *symbol* thông tin tương ứng với 16 tổ hợp 4 bít có thể có gồm các bít  $b_1, b_2, b_3, b_4$ . Các

symbol thông tin được ánh xạ thành các điểm tín hiệu tương ứng như được minh họa trên hình 7.23 mà chúng có các toạ độ  $[A_{mc}, A_{ms}]$ . Hai bộ tạo số ngẫu nhiên Gauss được sử dụng để tạo ra các thành phần tạp âm  $[n_c, n_s]$ . Để thuận tiện, lượng dịch pha  $\phi$  được đặt bằng không. Hệ quả là, véc-tơ tín hiệu-cộng-tập âm là

$$R = [A_{mc} + n_c \quad A_{ms} + n_s]$$

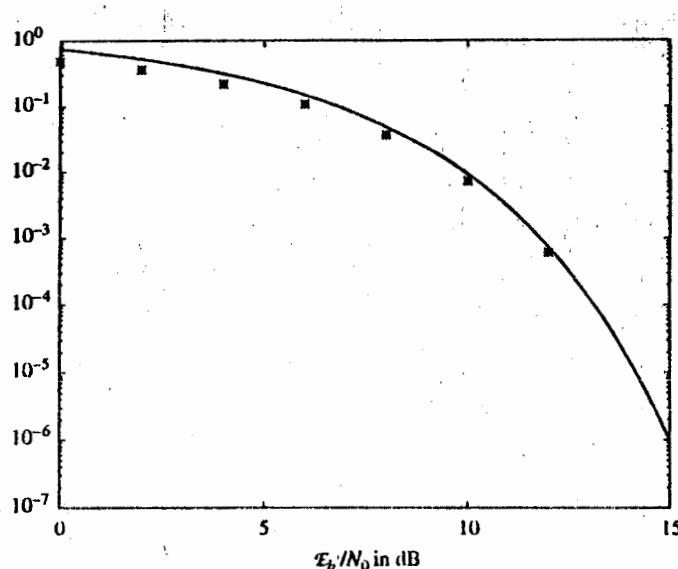


Hình 7.22 Sơ đồ khối một hệ thống 16-QAM dùng cho mô phỏng Monte-Carlo.



Hình 7.23 Biểu đồ sao tín hiệu 16-QAM dùng cho mô phỏng Monte-Carlo.

Bộ tách tín hiệu tính các metric khoảng cách được cho bởi (7.4.8) và biểu quyết cho điểm tín hiệu gần nhất với véc-tơ thu được  $r$ . Bộ đếm lỗi đếm các lỗi symbol trong chuỗi đã tách được. Hình 7.24 minh họa các kết quả của mô phỏng Monte-Carlo đối với việc truyền dẫn  $N=10\,000$  symbol tại các giá trị khác nhau của tham số SNR  $E_b/N_0$ , trong đó  $E_b=E_s/4$  là năng lượng tính trên một bít. Cũng được vẽ trên hình 7.24 là giá trị lý thuyết của xác suất lỗi symbol được cho bởi (7.4.10) và (7.4.11).



Hình 7.24 Chất lượng của hệ thống 16-QAM thu được từ mô phỏng Monte-Carlo.

Các MATLAB *script* đối với bài tập này được cho dưới đây.

### m-file

```
% MATLAB script for Illustrative Problem 6, Chapter 7.
echo on
SNRindB1=0:2:15;
SNRindB2=0:0.1:15;
M=16;
k=log2(M);
for i=1:length(SNRindB1),
    smld_err_prb(i)=cm_sm41(SNRindB1(i));      % simulated error rate
    echo off;
end;
echo on ;
for i=1:length(SNRindB2),
    SNR=exp(SNRindB2(i)*log(10)/10);           % signal to noise ratio
    % theoretical symbol error rate
    theo_err_prb(i)=4*Qfunct(sqrt(3*k*SNR/(M-1)));
    echo off ;
end;
echo on ;
% Plotting commands follow
semilogy(SNRindB1,smld_err_prb,'*');
hold
```

semilogy(SNRindB2,theo\_err\_prb);

### m-file

```

function [p]=cm_sm41(snr_in_dB)
% [p]=cm_sm41(snr_in_dB)
% CM_SM41 finds the probability of error for the given
% value of snr_in_dB, SNR in dB.

N=10000;
d=1; % min. distance between symbols
Eav=10*d^2; % energy per symbol
snr=10^(snr_in_dB/10); % SNR per bit (given)
sgma=sqrt(Eav/(8*snr)); % noise variance

M=16;
% generation of the data source follows
for i=1:N,
    temp=rand; % a uniform R.V. between 0 and 1
    dsource(i)=1+floor(M*temp); % a number between 1 and 16, uniform
end;
% Mapping to the signal constellation follow
mapping=[-3*d 3*d;
          -d 3*d;
          d 3*d;
          3*d 3*d;
          -3*d d;
          -d d;
          d d;
          3*d d;
          -3*d -d;
          -d -d;
          d -d;
          3*d -d;
          -3*d -3*d;
          -d -3*d;
          d -3*d;
          3*d -3*d];
for i=1:N,
    qam_sig(i,:)=mapping(dsource(i,:));
end;
% received signal
for i=1:N,
    [n(1) n(2)]=gngauss(sgma);
    r(i,:)=qam_sig(i,:)+n;

```

```
end;
% detection and error probability calculation
numoferr=0;
for i=1:N,
    % metric computation follow
    for j=1:M,
        metrics(j)=(r(i,1)-mapping(j,1))^2+(r(i,2)-mapping(j,2))^2;
    end;
    [min_metric decis]=min(metrics);
    if (decis~=dsource(i)),
        numoferr=numoferr+1;
    end;
end;
p=numoferr/(N);
```

## 7.5 ĐIỀU CHẾ TẦN SỐ SÓNG MANG

Chúng tôi đã mô tả các phương pháp truyền dẫn thông tin số bằng cách điều chế hoặc là biên độ của sóng mang, hoặc là pha của sóng mang, hoặc là cả pha và biên độ hỗn hợp. Thông tin số cũng có thể truyền được bằng việc điều chế tần số của sóng mang.

Như chúng ta sẽ thấy từ nghiên cứu dưới đây, truyền dẫn số bằng điều chế tần số là một phương pháp điều chế thích hợp đối với các kênh thiếu sự ổn định về pha cần thiết cho việc thực hiện ước lượng pha sóng mang. Ngược lại, các phương pháp điều chế tuyến tính mà chúng ta đã giới thiệu, cụ thể là PAM, PSK kết hợp và QAM, lại đòi hỏi phải ước lượng pha sóng mang để thực hiện tách tín hiệu một cách kết hợp về pha.

### 7.5.1 KHOÁ DỊCH TẦN SỐ

Dạng đơn giản nhất của điều chế tần số là khoá dịch tần số (FSK: *Frequency Shift Keying*) nhị phân. Trong FSK nhị phân, chúng ta sử dụng hai tần số khác nhau, ví dụ là  $f_1$  và  $f_2 = f_1 + \Delta f$ , để truyền một chuỗi thông tin nhị phân. Việc chọn phân cách tần số  $\Delta f = f_2 - f_1$  sẽ được xét sau. Như thế, hai dạng sóng tín hiệu có thể biểu diễn được theo

$$\begin{aligned} u_1(t) &= \sqrt{\frac{2E_b}{T_b}} \cos 2\pi f_1 t, & 0 \leq t \leq T \\ u_2(t) &= \sqrt{\frac{2E_b}{T_b}} \cos 2\pi f_2 t, & 0 \leq t \leq T \end{aligned} \quad (7.5.1)$$

trong đó  $E_b$  là năng lượng trên một bít, còn  $T_b$  là độ dài thời gian của một khoảng bít.

Tổng quát hơn nữa, FSK  $M$  mức có thể sử dụng được để truyền một khối gồm  $k = \log_2 M$  bít trên một dạng sóng tín hiệu. Trong trường hợp này  $M$  dạng sóng tín hiệu có thể biểu diễn được theo

$$u_m(t) = \sqrt{\frac{2E_s}{T}} \cos(2\pi f_c t + 2\pi m \Delta f t), \quad m = 0, 1, \dots, M-1, \quad 0 \leq t \leq T \quad (7.5.2)$$

trong đó  $E_s = kE_b$  và là năng lượng tính trên một symbol,  $T = kT_b$  là khoảng thời gian một symbol còn  $\Delta f$  là phân cách tần số giữa các tần số nằm cạnh nhau, nghĩa là  $\Delta f = f_m - f_{m-1}$  đối với mọi  $m = 1, 2, \dots, M-1$ , ở đây  $f_m = f_c + m\Delta f$ .

Hãy để ý rằng  $M$  dạng sóng FSK có cùng năng lượng, ký hiệu là  $E$ . Khoảng phân cách tần số  $\Delta f$  xác định mức độ mà với nó chúng ta có thể phân biệt tách bạch  $M$  dạng sóng phát có thể có. Giống như một độ đo về tính tương tự (hay tính không tương tự) giữa một cặp dạng sóng tín hiệu, chúng ta sử dụng hệ số tương quan  $\gamma_{mn}$ :

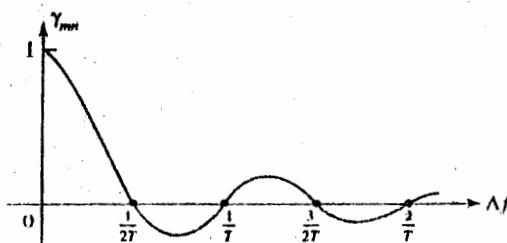
$$\gamma_{mn} = \frac{1}{E_s} \int_0^T u_m(t) u_n(t) dt \quad (7.5.3)$$

Bằng cách thay các biểu thức đổi với  $u_m(t)$  và  $u_n(t)$  vào (7.5.3), chúng ta được

$$\begin{aligned} \gamma_{mn} &= \frac{1}{E_s} \int_0^T \frac{2E_s}{T} \cos(2\pi f_c t + 2\pi m \Delta f t) \cos(2\pi f_c t + 2\pi n \Delta f t) dt \\ &= \frac{1}{T} \int_0^T \cos 2\pi(m-n)\Delta f t dt + \frac{1}{T} \int_0^T \cos[4\pi f_c t + 2\pi(m+n)\Delta f t] dt \quad (7.5.4) \\ &= \frac{\sin 2\pi(m-n)\Delta f T}{2\pi(m-n)\Delta f T} \end{aligned}$$

trong đó tích phân thứ hai triệt tiêu khi  $f_c \gg 1/T$ . Một đồ thị về  $\gamma_{mn}$  như một hàm của độ phân cách tần số  $\Delta f$  được cho trên hình vẽ 7.25. Chúng ta thấy được

rằng các dạng sóng tín hiệu là trực giao với nhau khi  $\Delta f$  là bội của  $1/2T$ . Như vậy, khoảng phân cách tần số tối thiểu giữa các tần số nằm cạnh nhau để có được tính trực giao là  $1/2T$ . Chúng ta cũng chú ý rằng giá trị cực tiểu của hệ số tương quan là  $\gamma_{mn} = -0.217$  mà nó xảy ra với phân cách tần số  $\Delta f = 0.715/T$ .



Hình 7.25 Hệ số tương quan chéo như một hàm của phân cách tần số đối với các tín hiệu FSK.

Các dạng sóng FSK trực giao  $M$  mức có biểu diễn hình học như  $M$  véc tơ  $M$  chiều được cho theo

$$s_0 = (\sqrt{E_s}, 0, \dots, 0) \quad (7.5.5)$$

$$s_1 = (0, \sqrt{E_s}, \dots, 0) \quad (7.5.6)$$

$$\dots \quad (7.5.7)$$

$$s_{M-1} = (0, 0, \dots, \sqrt{E_s}) \quad (7.5.8)$$

trong đó các hàm cơ sở là  $\psi_m(t) = \sqrt{2/T} \cos(f_c + m\Delta f)t$ . Khoảng cách giữa các cặp véc tơ là  $d = \sqrt{2E_s}$  đối với mọi  $m, n$  và nó cũng là khoảng cách cực tiểu giữa  $M$  tín hiệu với nhau. Hãy để ý rằng các tín hiệu này thì tương đương với các tín hiệu trực giao bằng gốc  $M$  mức đã được mô tả trong mục 5.4.

Việc giải điều chế và tách tín hiệu của các tín hiệu FSK  $M$  mức sẽ được xem xét dưới đây.

### 7.5.2 GIẢI ĐIỀU CHẾ VÀ TÁCH CÁC TÍN HIỆU FSK

Chúng ta hãy giả sử rằng các tín hiệu FSK được truyền qua một kênh AWGN. Hơn nữa, chúng ta giả sử rằng mỗi một tín hiệu đều bị giữ chậm (trễ) trong quá trình truyền dẫn qua kênh. Hệ quả là, các tín hiệu thu được tại lối vào bộ

giải điều chế có thể biểu diễn được theo

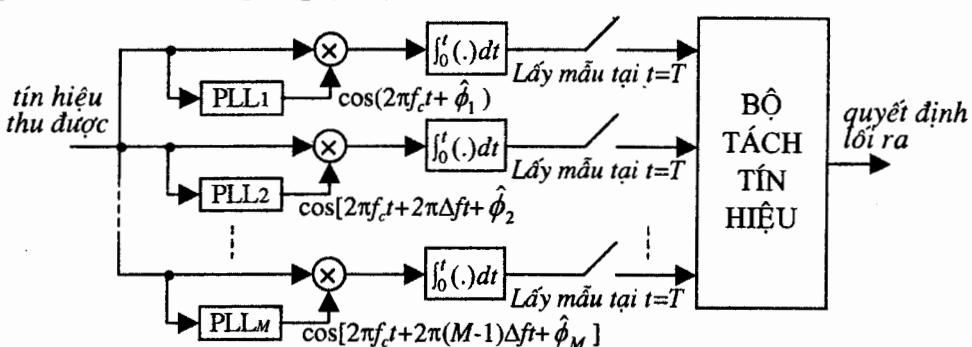
$$r(t) = \sqrt{\frac{2E_s}{T}} \cos(2\pi f_c t + 2\pi m \Delta f t + \phi_m) + n(t) \quad (7.5.9)$$

trong đó  $\phi_m$  ký hiệu cho lượng dịch pha của tín hiệu thứ  $m$  (do trễ truyền dẫn) còn  $n(t)$  biểu thị tạp âm cộng thông dải mà nó có thể biểu diễn được theo

$$n(t) = n_c(t) \cos 2\pi f_c t - n_s(t) \sin 2\pi f_c t \quad (7.5.10)$$

Việc giải điều chế và tách đối với  $M$  tín hiệu FSK có thể hoàn thành được bằng một trong hai phương pháp. Một giải pháp là ước lượng  $M$  lượng dịch pha sóng mang  $\{\phi_m\}$  và thực hiện giải điều chế và tách tín hiệu kết hợp về pha. Một cách khác là các góc pha sóng mang được bỏ qua không quan tâm đến trong quá trình giải điều chế và tách tín hiệu.

Trong giải điều chế kết hợp về pha, tín hiệu thu được  $r(t)$  được tính tương quan với từng tín hiệu một trong số  $M$  tín hiệu có thể có là  $\cos(2\pi f_c t + 2\pi m \Delta f t + \hat{\phi}_m)$ , đối với  $m=0, 1, \dots, M-1$ , trong đó  $\{\hat{\phi}_m\}$  là các ước lượng pha sóng mang. Một sơ đồ khái minh họa kiểu giải điều chế này được trình bày trên hình 7.26. Điều thú vị đáng quan tâm ở đây là khi  $\hat{\phi}_m \neq \phi_m$  đối với  $m = 0, 1, \dots, M-1$  (các ước lượng pha không hoàn hảo), thì phân cách tần số cần thiết để có được tính trực giao tín hiệu ở bộ giải điều chế sẽ là  $\Delta f = 1/T$ , gấp đôi phân cách tối thiểu để có được tính trực giao đối với trường hợp  $\hat{\phi} = \phi$ .



Hình 7.26 Giải điều chế kết hợp về pha đối với các tín hiệu FSK  $M$  mức.

Đòi hỏi về ước lượng  $M$  pha sóng mang làm cho việc giải điều chế của các tín

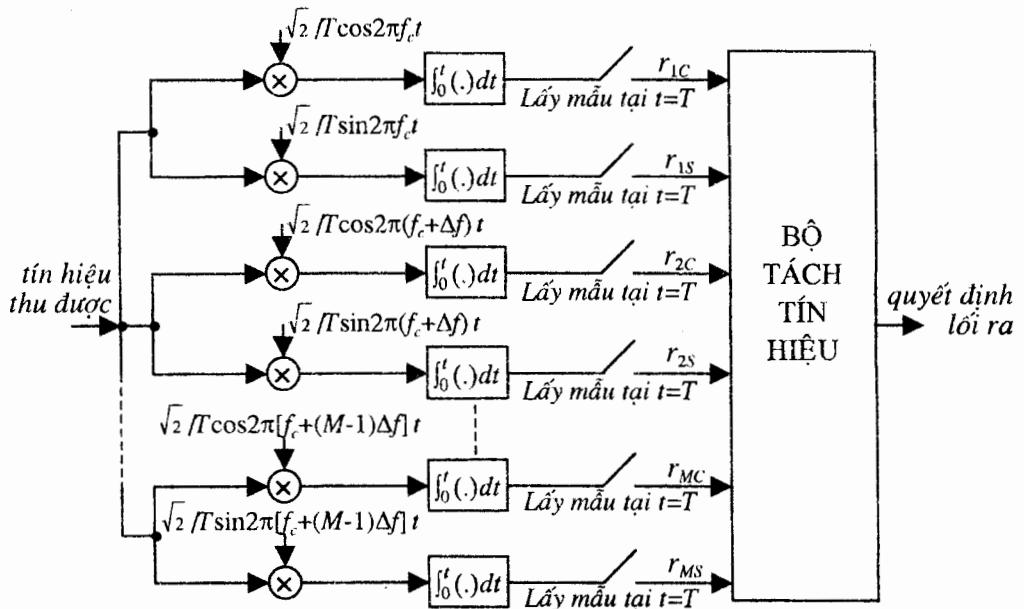
hiệu FSK trở nên cực kỳ phức tạp và không thực tế, đặc biệt khi số các tín hiệu lớn. Vì vậy, chúng ta sẽ không xét đến việc tách một cách kết hợp các tín hiệu FSK.

Thay vào đó, chúng ta xem xét một phương pháp giải điều chế và tách tín hiệu mà nó không đòi hỏi phải biết về các góc pha của sóng mang. Việc giải điều chế có thể hoàn thành được như được thể hiện trên hình vẽ 7.27. Trong trường hợp này, có hai bộ tương quan trên một dạng sóng tín hiệu hay nói chung có tổng cộng  $2M$  bộ tương quan. Tín hiệu thu được được tính tương quan với các hàm cơ sở (các sóng mang trực giao) là  $\sqrt{2/T} \cos(2\pi f_c t + 2\pi m \Delta f t)$  và  $\sqrt{2/T} \sin(2\pi f_c t + 2\pi m \Delta f t)$  đối với  $m = 0, 1, \dots, M-1$ .  $2M$  lối ra của các bộ tương quan được lấy mẫu tại cuối của khoảng tín hiệu và được đưa tới bộ tách tín hiệu. Như thế, nếu tín hiệu thứ  $m$  được truyền đi thì  $2M$  giá trị lấy mẫu tới bộ tách tín hiệu có thể biểu diễn được theo

$$r_{kc} = \sqrt{E_s} \left[ \frac{\sin 2\pi(k-m)\Delta f T}{2\pi(k-m)\Delta f T} \cos \phi_m - \frac{\cos 2\pi(k-m)\Delta f T - 1}{2\pi(k-m)\Delta f T} \sin \phi_m \right] + n_{kc} \quad (7.5.11)$$

$$r_{ks} = \sqrt{E_s} \left[ \frac{\cos 2\pi(k-m)\Delta f T - 1}{2\pi(k-m)\Delta f T} \cos \phi_m + \frac{\sin 2\pi(k-m)\Delta f T}{2\pi(k-m)\Delta f T} \sin \phi_m \right] + n_{ks}$$

trong đó  $n_{kc}$  và  $n_{ks}$  ký hiệu cho các thành phần tạp âm Gauss trong các tín hiệu lối ra đã được lấy mẫu.



Hình 7.27 Giải điều chế các tín hiệu  $M$  mức đối với tách tín hiệu không kết hợp.

Chúng ta thấy rằng khi  $k = m$ , các giá trị đã lấy mẫu đưa tới bộ tách tín hiệu là

$$\begin{aligned} r_{mc} &= \sqrt{E_s} \cos \phi_m + n_{mc} \\ r_{ms} &= \sqrt{E_s} \sin \phi_m + n_{ms} \end{aligned} \quad (7.5.12)$$

Hơn thế, chúng ta thấy rằng khi  $k \neq m$ , các thành phần tín hiệu trong các mẫu  $r_{kc}$  và  $r_{ks}$  sẽ triệt tiêu, bất luận các giá trị của lượng dịch pha  $\phi_k$  thế nào, miễn là phân cách tần số giữa các tần số cạnh nhau là  $\Delta f = 1/T$ . Trong trường hợp như thế,  $2(M-1)$  đầu ra của các bộ tương quan khác chỉ gồm có tạp âm, nghĩa là

$$n_{kc} = n_{kc}, \quad r_{ks} = n_{ks}, \quad k \neq m \quad (7.5.13)$$

Trong tính toán tiếp đây, chúng ta giả sử rằng  $\Delta f = 1/T$  sao cho các tín hiệu là trực giao.

Có thể chỉ ra được rằng  $2M$  mẫu tạp âm  $\{n_{kc}\}$  và  $\{n_{ks}\}$  là các biến ngẫu nhiên Gauss kỳ vọng 0 và không tương quan với nhau với cùng phương sai  $\sigma^2 = N_0/2$ . Hệ quả là hàm mật độ xác suất cùng nhau đối với  $r_{mc}$  và  $r_{ms}$  với điều kiện  $\phi_m$  là

$$f_{r_m}(r_{mc}, r_{ms} | \phi_m) = \frac{1}{2\pi\sigma^2} e^{-[(r_{mc} - \sqrt{E_s} \cos \phi_m)^2 + (r_{ms} - \sqrt{E_s} \sin \phi_m)^2] / 2\sigma^2} \quad (7.5.14)$$

và đối với  $k \neq m$ , chúng ta có

$$f_{r_k}(r_{kc}, r_{ks}) = \frac{1}{2\pi\sigma^2} e^{-(r_{kc}^2 + r_{ks}^2) / 2\sigma^2} \quad (7.5.15)$$

Với  $2M$  biến ngẫu nhiên đã quan sát được  $\{r_{kc}, r_{ks}\}_{k=0}^{M-1}$ , bộ tách tín hiệu tối ưu chọn ra tín hiệu tương ứng với các xác suất hậu nghiệm cực đại, nghĩa là

$$P[s_m \text{ đã được truyền đi} | \mathbf{r}] \equiv P(s_m | \mathbf{r}), \quad m = 0, 1, \dots, M-1 \quad (7.5.16)$$

Trong đó  $\mathbf{r}$  là véc-tơ  $2M$  chiều với các phân tử  $\{r_{kc}, r_{ks}\}_{k=0}^{M-1}$ . Khi các tín hiệu đồng khả năng, bộ tách tín hiệu tối ưu có tính chất theo (7.5.16) sẽ tính toán các đường bao của các tín hiệu, được xác định theo

$$r_m = \sqrt{r_{mc}^2 + r_{ms}^2}, \quad m = 0, 1, \dots, M-1 \quad (7.5.17)$$

và chọn ra tín hiệu ứng với đường bao lớn nhất của tập  $\{r_m\}$ . Trong trường hợp này, bộ tách tín hiệu được gọi là một bộ tách sóng đường bao.

Một bộ tách tín hiệu tương đương là bộ thực hiện tính các đường bao trung bình

$$r_m^2 = r_{mc}^2 + r_{ms}^2, \quad m = 0, 1, \dots, M-1 \quad (7.5.18)$$

và chọn ra tín hiệu tương ứng với giá trị lớn nhất của tập  $\{r_m^2\}$ . Trong trường hợp này, bộ tách tín hiệu tối ưu được gọi là một bộ tách sóng luật bình phương.

### BÀI TẬP MINH HỌA

**Bài tập minh họa 7.7 [Truyền tin bằng tín hiệu FSK]** Hãy xét một hệ thống truyền tin nhị phân áp dụng hai dạng sóng tín hiệu FSK được cho theo

$$u_1(t) = \cos 2\pi f_1 t, \quad 0 \leq t \leq T_b$$

$$u_2(t) = \cos 2\pi f_2 t, \quad 0 \leq t \leq T_b$$

trong đó  $f_1 = 1000/T_b$  và  $f_2 = f_1 + 1/T_b$ . Kênh gây nên một lượng dịch pha  $\phi = 45^\circ$  đối với từng tín hiệu đã được truyền đi sao cho tín hiệu thu được khi không có tạp âm là

$$r(t) = \cos \left( 2\pi f_i t + \frac{\pi}{4} \right), \quad i = 1, 2, \quad 0 \leq t \leq T_b$$

Hãy thông qua tính toán bằng số thực hiện bộ giải điều chế kiểu tương quan đối với các tín hiệu FSK.

#### Lời giải

Chúng ta hãy lấy mẫu tín hiệu thu được  $r(t)$  với tốc độ  $F_s = 5000/T_b$  trong khoảng bít  $T_b$ . Như thế, tín hiệu thu được  $r(t)$  được biểu diễn bằng 5000 mẫu  $\{r(n/F_s)\}$ . Bộ giải điều chế tương quan thực hiện nhân  $\{r(n/F_s)\}$  với phiên bản thông qua lấy mẫu của  $u_1(t) = \cos 2\pi f_1 t$ ,  $v_1(t) = \sin 2\pi f_1 t$ ,  $u_2(t) = \cos 2\pi f_2 t$  và  $v_2(t) = \sin 2\pi f_2 t$ , như được minh họa trên hình vẽ 7.27. Như thế, các lối ra bộ tương quan là

$$r_{1c}(k) = \sum_{n=0}^k r\left(\frac{n}{F_s}\right) u_1\left(\frac{n}{F_s}\right), \quad k = 1, 2, \dots, 5000$$

$$r_{1s}(k) = \sum_{n=0}^k r\left(\frac{n}{F_s}\right) v_1\left(\frac{n}{F_s}\right), \quad k = 1, 2, \dots, 5000$$

$$r_{2c}(k) = \sum_{n=0}^k r\left(\frac{n}{F_s}\right) u_2\left(\frac{n}{F_s}\right), \quad k = 1, 2, \dots, 5000$$

$$r_{2s}(k) = \sum_{n=0}^k r\left(\frac{n}{F_s}\right) v_2\left(\frac{n}{F_s}\right), \quad k = 1, 2, \dots, 5000$$

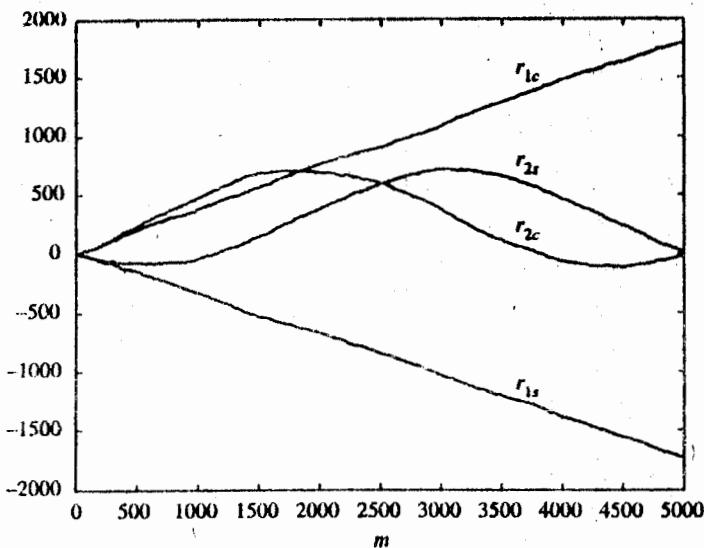
Bộ tách tín hiệu là một bộ tách sóng luật bình phương mà nó thực hiện tính hai biến quyết định

$$r_1 = r_{1c}^2(5000) + r_{1s}^2(5000)$$

$$r_2 = r_{2c}^2(5000) + r_{2s}^2(5000)$$

và chọn ra bít thông tin tương ứng với biến quyết định lớn nhất.

Một chương trình MATLAB thực hiện bằng số các phép tương quan được cho dưới đây. Các đồ thị của các lối ra bộ tương quan được trình bày trên hình 7.28, với giả thiết là tín hiệu  $u_1(t)$  được truyền đi.



Hình 7.28 Tín hiệu lối ra các bộ tương quan đối với giải điều chế FSK nhị phân.

### m-file

```
% MATLAB script for Illustrative Problem 7, Chapter 7.
echo on
Tb=1;
f1=1000/Tb;
f2=f1+1/Tb;
phi=pi/4;
N=5000; % number of samples
t=0:Tb/(N-1):Tb;
u1=cos(2*pi*f1*t);
u2=cos(2*pi*f2*t);
% assuming that u1 is transmitted, the received signal r is
```

```

sgma=1; % noise variance
for i=1:N,
    r(i)=cos(2*pi*f1*t(i)+phi)+gngauss(sgma);
    echo off;
end;
echo on ;
% the correlator outputs are computed next
v1=sin(2*pi*f1*t);
v2=sin(2*pi*f2*t);
r1c(1)=r(1)*u1(1);
r1s(1)=r(1)*v1(1);
r2c(1)=r(1)*u2(1);
r2s(1)=r(1)*v2(1);
for k=2:N,
    r1c(k)=r1c(k-1)+r(k)*u1(k);
    r1s(k)=r1s(k-1)+r(k)*v1(k);
    r2c(k)=r2c(k-1)+r(k)*u2(k);
    r2s(k)=r2s(k-1)+r(k)*v2(k);
    echo off;
end;
echo on;
% decision variables
r1=r1c(5000)^2+r1s(5000)^2;
r2=r2c(5000)^2+r2s(5000)^2;
% plotting commands follow

```

### 7.5.3 XÁC SUẤT LỖI ĐỐI VỚI TÁCH KHÔNG KẾT HỢP TÍN HIỆU FSK

Việc tìm ra chất lượng hoạt động của bộ tách sóng đường bao tối ưu đối với các tín hiệu FSK  $M$  mức có thể tìm thấy được trong hầu hết các giáo trình về thông tin số. Xác suất của một lỗi *symbol* có thể biểu diễn được theo

$$P_M = \sum_{n=1}^{M-1} (-1)^{n+1} \binom{M-1}{n} \frac{1}{n+1} e^{-nkE_b / N_0(n+1)} \quad (7.5.19)$$

Khi  $M = 2$ , biểu thức này rút gọn thành xác suất lỗi đối với FSK nhị phân là

$$P_2 = \frac{1}{2} e^{-E_b / 2N_0} \quad (7.5.20)$$

Đối với  $M > 2$ , xác suất lỗi bít có thể nhận được từ xác suất lỗi *symbol* theo quan hệ

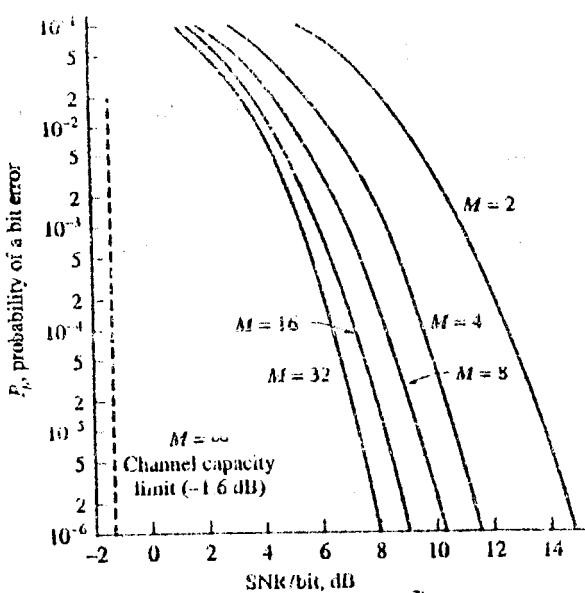
$$P_b = \frac{2^{k-1}}{2^k - 1} P_M \quad (7.5.21)$$

Xác suất lỗi bít được vẽ trên hình 7.29 như một hàm của SNR tính theo bít đối với  $M = 2, 4, 8, 16$  và  $32$ . Chúng ta thấy rằng đối với bất kỳ xác suất lỗi bít bằng bao nhiêu thì SNR tính theo bít cũng giảm khi  $M$  tăng. Về giới hạn khi  $M \rightarrow \infty$ , xác suất lỗi có thể làm cho nhỏ bao nhiêu tùy ý miễn là SNR tính trên bít vượt quá  $-1.6$  dB. Đây là giới hạn dung lượng kênh hay còn gọi là *giới hạn Shannon*, đối với bất kỳ hệ thống thông tin số nào truyền thông tin qua một kênh AWGN.

Cái giá phải trả khi tăng  $M$  là yêu cầu về băng thông cần thiết để truyền dẫn các tín hiệu. Do phân cách tần số giữa các tần số nằm cạnh nhau là  $\Delta f = 1/T$  để có được tính trực giao của các tín hiệu, độ rộng băng thông yêu cầu đối với  $M$  tín hiệu là  $W = M/T$ . Tốc độ bít là  $R = k/T$ , trong đó  $k = \log_2 M$ . Như vậy, tỷ lệ tốc độ bít trên độ rộng băng thông là

$$\frac{R}{W} = \frac{\log_2 M}{M} \quad (7.5.22)$$

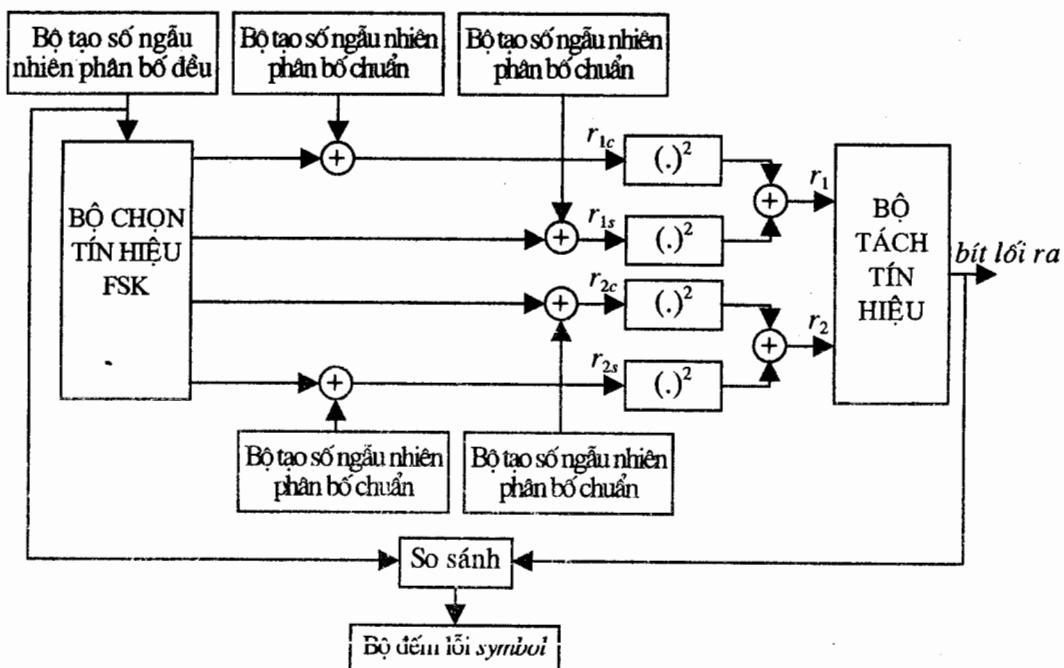
Chúng ta thấy rằng  $R/W \rightarrow 0$  khi  $M \rightarrow \infty$ .



Hình 7.29 Xác suất của một lỗi bít đối với tách sóng không kết hợp các tín hiệu FSK trực giao.

## BÀI TẬP MINH HỌA

**Bài tập minh họa 7.8 [Mô phỏng FSK nhị phân]** Hãy thực hiện một mô phỏng Monte-Carlo của một hệ thống truyền tin FSK nhị phân trong đó các dạng sóng tín hiệu được cho bởi (7.5.1), trong đó  $f_2 = f_1 + 1/T_b$  còn bộ tách tín hiệu là một bộ tách sóng luật bình phương. Sơ đồ khối của hệ thống FSK nhị phân dùng cho mô phỏng được cho trên hình 7.30.



Hình 7.30 Sơ đồ khối của một hệ thống FSK nhị phân dùng cho mô phỏng Monte-Carlo.

### Lời giải

Do các tín hiệu trực giao với nhau nên khi  $u_1(t)$  được phát đi thì lối ra của bộ giải điều chế thứ nhất là

$$\begin{aligned}r_{1c} &= \sqrt{E_b} \cos \phi + n_{1c} \\r_{1s} &= \sqrt{E_b} \sin \phi + n_{1s}\end{aligned}$$

còn lối ra của bộ giải điều chế thứ hai là

$$\begin{aligned}r_{2c} &= n_{2c} \\r_{2s} &= n_{2s}\end{aligned}$$

trong đó  $n_{1c}, n_{1s}, n_{2c}$  và  $n_{2s}$  là các biến ngẫu nhiên Gauss kỳ vọng không, độc

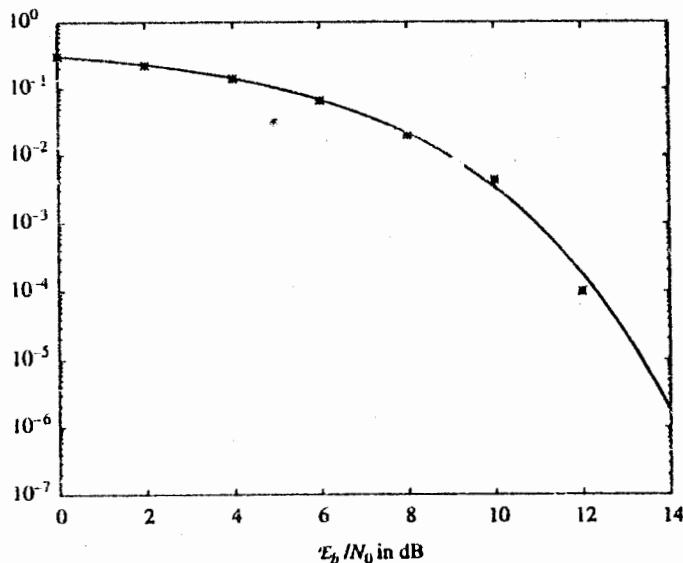
lập thống kê với nhau, có phương sai bằng  $\sigma^2$  còn  $\phi$  biểu thị lượng dịch pha do kênh truyền.

Trong biểu thức trên đây, lượng dịch pha do kênh truyền  $\phi$  có thể cho bằng không cho đơn giản. Bộ tách sóng luật bình phương thực hiện tính

$$\begin{aligned}r_1 &= r_{1c}^2 + r_{1s}^2 \\r_2 &= r_{2c}^2 + r_{2s}^2\end{aligned}$$

và chọn ra bít thông tin ứng với biến nào lớn hơn trong hai biến trên. Một bộ đếm lỗi thực hiện đo tỷ lệ lỗi bằng cách so sánh chuỗi đã truyền với đầu ra của bộ tách sóng.

Các chương trình MATLAB thực hiện mô phỏng Monte-Carlo này được cho dưới đây. Hình 7.31 minh họa tỷ lệ lỗi đo được và so sánh nó với xác suất lỗi lý thuyết được cho bởi (7.5.20).



Hình 7.31 Chất lượng của một hệ thống FSK nhị phân thu được từ mô phỏng Monte-Carlo

### m-file

```
% MATLAB script for Illustrative Problem 8, Chapter 7.
echo on
SNRindB1=0:2:15;
SNRindB2=0:0.1:15;
```

```

for i=1:length(SNRindB1),
    smld_err_prb(i)=cm_sm52(SNRindB1(i));      % simulated error rate
    echo off ;
end;
echo on ;
for i=1:length(SNRindB2),
    SNR=exp(SNRindB2(i)*log(10)/10);           % signal to noise ratio
    theo_err_prb(i)=(1/2)*exp(-SNR/2);          % theoretical symbol error rate
    echo off;
end;
echo on;
% Plotting commands follow
semilogy(SNRindB1,smld_err_prb,'*');
hold
semilogy(SNRindB2,theo_err_prb);

```

### m-file

```

function [p]=cm_sm52(snr_in_dB)
% [p]=cm_sm52(snr_in_dB)
%     CM_SM52 Returns the probability of error for the given
%     value of snr_in_dB, signal to noise ratio in dB.
N=10000;
Eb=1;
d=1;
snr=10^(snr_in_dB/10);                      % signal to noise ratio per bit
sgma=sqrt(Eb/(2*snr));                       % noise variance
phi=0;
% generation of the data source follows
for i=1:N,
    temp=rand;                                % a uniform random variable between 0 and 1
    if (temp<0.5),
        dsoure(i)=0;
    else
        dsoure(i)=1;
    end;
end;
% detection and the probability of error calculation
numoferr=0;
for i=1:N,
    % demodulator output
    if (dsoure(i)==0),

```

```

r0c=sqrt(Eb)*cos(phi)+gngauss(sgma);
r0s=sqrt(Eb)*sin(phi)+gngauss(sgma);
r1c=gngauss(sgma);
r1s=gngauss(sgma);
else
    r0c=gngauss(sgma);
    r0s=gngauss(sgma);
    r1c=sqrt(Eb)*cos(phi)+gngauss(sgma);
    r1s=sqrt(Eb)*sin(phi)+gngauss(sgma);
end;
% square law detector outputs
r0=r0c^2+r0s^2;
r1=r1c^2+r1s^2;
% decision is made next
if (r0>r1),
    decis=0;
else
    decis=1;
end;
% Let's increment the error counter, if the decision is not correct
if (decis~=dsource(i)),
    numoferr=numoferr+1;
end;
end;
p=numoferr/(N);

```

## 7.6 ĐỒNG BỘ TRONG CÁC HỆ THỐNG THÔNG TIN

Trong mục 3.3 chúng ta đã mô tả quá trình giải điều chế đối với các tín hiệu AM. Cụ thể, chúng ta đã chỉ ra rằng chúng ta có thể phân loại các sơ đồ giải điều chế thành các sơ đồ kết hợp và không kết hợp. Trong giải điều chế kết hợp, tín hiệu AM được nhân với một sóng hình sin có cùng tần số và pha của sóng mang rồi được đem giải điều chế. Trong giải điều chế không kết hợp, chỉ áp dụng được với sơ đồ AM truyền thống, việc giải điều chế đường bao được áp dụng và tại máy thu không cần phải bám chính xác pha và tần số của sóng mang. Hơn thế nữa, chúng ta đã chỉ ra trong bài tập minh họa 3.6 rằng việc đồng bộ chính xác về pha trong giải điều chế kết hợp là rất quan trọng và các sai số về pha có thể dẫn đến sự suy giảm chất lượng đáng kể.

Trong chương này, chúng ta đã bàn bạc về các sơ đồ giải điều chế đối với các hệ thống điều chế số sóng mang. Trong giải điều chế của PAM, PSK và QAM, chúng ta đã giả sử rằng chúng ta biết đầy đủ về pha và tần số của sóng mang.

Trong mục này, chúng ta sẽ bàn đến các phương pháp tạo ra các sóng hình sin tại bộ giải điều chế có cùng tần số và pha của sóng mang. Các phương pháp này được nghiên cứu dưới tiêu đề *đồng bộ sóng mang* và áp dụng cả cho các hệ thống điều chế số lẫn điều chế tương tự sóng mang, đã được thảo luận trong chương 3 và chương này. Một loại đồng bộ khác, được gọi là *đồng bộ định thời, đồng bộ đồng hồ* hay *khôi phục định thời*, chỉ gặp trong các hệ thống truyền tin số. Trong mục này chúng ta cũng sẽ thảo luận một cách vắn tắt về vấn đề đồng bộ kiểu này.

### 7.6.1 ĐỒNG BỘ SÓNG MANG

Một hệ thống đồng bộ sóng mang gồm một bộ dao động nội mà pha của nó được điều khiển cho đồng bộ với tín hiệu sóng mang. Điều này đạt được bằng cách áp dụng một mạch vòng khoá pha (PLL: *Phase-Locked Loop*). Một PLL là một hệ thống điều khiển phi tuyến liên kết ngược nhằm điều khiển pha của bộ dao động nội. Trong nghiên cứu dưới đây, để đơn giản chúng ta sẽ chỉ xem xét các hệ thống điều chế PSK nhị phân.

PLL được dẫn bởi một tín hiệu hình sin tại tần số sóng mang (hay một bội của nó). Để nhận được tín hiệu hình sin để dẫn PLL, tín hiệu được điều chế hai băng biên

$$u(t) = A_c m(t) \cos[2\pi f_c t - \phi(t)] \quad (7.6.1)$$

trong đó  $m(t) = \pm 1$ , được bình phương để có được

$$\begin{aligned} u^2(t) &= A_c^2 m^2(t) \cos^2[2\pi f_c t - \phi(t)] \\ &= \frac{A_c^2}{2} m^2(t) + \frac{A_c^2}{2} m^2(t) \cos[4\pi f_c t - 2\phi(t)] \\ &= \frac{A_c^2}{2} + \frac{A_c^2}{2} \cos[4\pi f_c t - 2\phi(t)] \end{aligned} \quad (7.6.2)$$

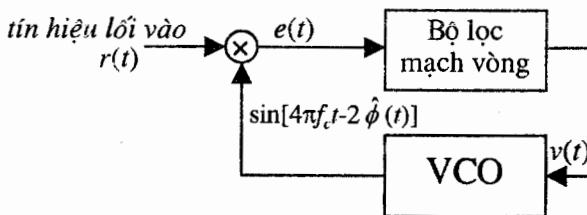
Hiển nhiên là tín hiệu này có một thành phần tại tần số  $2f_c$ . Lý do của việc ta

không giải quyết trực tiếp với  $u(t)$  là ở chỗ quá trình  $m(t)$  thường có kỳ vọng bằng không cho nên công suất của thành phần tín hiệu tại  $f_c$  của  $u(t)$  bằng 0. Bây giờ, nếu tín hiệu  $u^2(t)$  được cho qua một mạch lọc thông dải điều hướng tại  $2f_c$  thì lối ra sẽ là một tín hiệu hình sin có tần số trung tâm  $2f_c$ , góc pha là  $-2\phi(t)$  còn biên độ thì bằng 1; nghĩa là tín hiệu lối vào PLL là

$$r(t) = \cos[4\pi f_c t - 2\phi(t)] \quad (7.6.3)$$

PLL gồm một mạch nhân, một bộ lọc mạch vòng và một bộ dao động điều khiển bằng điện áp (VCO: *Voltage-Controlled Oscillator*) như được cho thấy trên hình 7.32. nếu chúng ta giả sử rằng lối ra của VCO là  $\sin[4\pi f_c t - 2\hat{\phi}(t)]$  thì tại lối vào bộ lọc mạch vòng chúng ta có

$$\begin{aligned} e(t) &= \cos[4\pi f_c t - 2\phi(t)] \sin[4\pi f_c t - 2\hat{\phi}(t)] \\ &= \frac{1}{2} \sin[2\phi(t) - 2\hat{\phi}(t)] + \frac{1}{2} \sin[8\pi f_c t - 2\phi(t) - 2\hat{\phi}(t)] \end{aligned} \quad (7.6.4)$$



Hình 7.32 Mạch vòng khoá pha.

Hãy chú ý rằng  $e(t)$  có một thành phần cao tần và một thành phần thấp tần. Vai trò của bộ lọc mạch vòng là loại bỏ thành phần cao tần và bảo đảm để  $\hat{\phi}(t)$  theo sát các biến thiên của  $\phi(t)$ . Một mạch lọc vòng đơn giản là một bộ lọc thông thấp bậc một với một hàm truyền có dạng

$$G(s) = \frac{1 + \tau_1 s}{1 + \tau_2 s} \quad (7.6.5)$$

trong đó  $\tau_2 \gg \tau_1$ . Nếu chúng ta ký hiệu tín hiệu lối vào VCO là  $v(t)$  thì lối ra của VCO sẽ là một sóng hình sin mà độ lệch tần số tức thời của nó khỏi tần số  $2f_c$  thì tỷ lệ với  $v(t)$ . Song tần số tức thời của lối ra VCO là

$$2f_c + \frac{1}{\pi} \frac{d}{dt} \hat{\phi}(t)$$

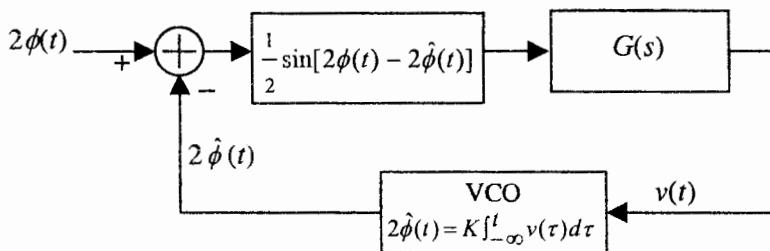
do vậy

$$\frac{d}{dt} \hat{\phi}(t) = \frac{K}{2} v(t) \quad (7.6.6)$$

hay tương đương là

$$2\hat{\phi}(t) = K \int_{-\infty}^t v(\tau) d\tau \quad (7.6.7)$$

trong đó  $K$  là một hằng số tỷ lệ nào đó. Sau khi loại bỏ các hài bậc hai và bậc bốn, PLL suy biến thành sơ đồ như được trình bày trên hình 7.33.

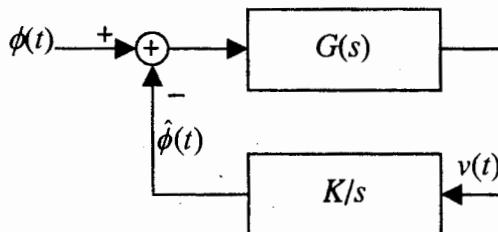


Hình 7.33 PLL sau khi loại các thành phần tần số cao.

Khi giả sử rằng  $\hat{\phi}(t)$  theo sát các thay đổi trong  $\phi(t)$ , thì sai số  $2\phi(t)-2\hat{\phi}(t)$  rất nhỏ và chúng ta có thể sử dụng xấp xỉ

$$\frac{1}{2} \sin[2\phi(t) - 2\hat{\phi}(t)] \approx \phi(t) - \hat{\phi}(t) \quad (7.6.8)$$

Với xấp xỉ này, phần tử phi tuyến duy nhất trên hình vẽ 7.33 được thay thế bằng một phân tử tuyến tính, dẫn đến mô hình PLL được tuyến tính hóa được trình bày trên hình 7.34. Hãy chú ý rằng mô hình này biểu diễn được trên miền biến đổi và bộ tích phân được thay bằng tương đương trên miền biến đổi của nó là  $1/s$ .



Hình 7.34 Mô hình tuyến tính hóa đối với một mạch vòng khóa pha.

Mô hình được thể hiện trên hình 7.34 là một hệ thống điều khiển tuyến tính với một tăng ích hướng thuận là  $G(s)$  và một tăng ích mạch hồi tiếp là  $K/s$ ; do vậy, hàm truyền của hệ thống này được cho bởi

$$H(s) = \frac{\hat{\Phi}(s)}{\Phi(s)} = \frac{KG(s)/s}{1 + KG(s)/s} \quad (7.6.9)$$

và với mô hình bậc nhất đôi với  $G(s)$  được giả thiết trên đây

$$G(s) = \frac{1 + \tau_1 s}{1 + \tau_2 s} \quad (7.6.10)$$

thì  $H(s)$  được cho theo

$$H(s) = \frac{1 + \tau_1 s}{1 + (\tau_1 + 1/K)s + \tau_2 s^2 / K} \quad (7.6.11)$$

Với  $H(s)$  như một hàm truyền, nếu lối vào PLL là  $\Phi(s)$  thì sai số sẽ là

$$\begin{aligned} \Delta\Phi(s) &= \Phi(s) - \hat{\Phi}(s) = \Phi(s) - \Phi(s)H(s) \\ &= [1 - H(s)]\Phi(s) = \frac{(1 + \tau_2 s)s}{K + (1 + K\tau_1)s + \tau_2 s^2} \Phi(s) \end{aligned} \quad (7.6.12)$$

Bây giờ chúng ta hãy giả sử rằng cho tới một thời điểm nhất định nào đó  $\phi(t) \approx \hat{\phi}(t)$ , vì thế  $\Delta\phi(t) \approx 0$ . Tại thời điểm này, một thay đổi đột ngột nào đó gây nên một cú nhảy trong  $\phi(t)$  mà nó có thể mô hình hoá được như một bậc thang, nghĩa là  $\Phi(s) = K_1/s$ . Với sự thay đổi này, chúng ta có

$$\Delta\Phi(s) = \frac{(1 + \tau_2 s)s}{K + (1 + K\tau_1)s + \tau_2 s^2} \frac{K_1}{s} = \frac{K_1(1 + \tau_2 s)}{K + (1 + K\tau_1)s + \tau_2 s^2} \quad (7.6.13)$$

Bây giờ, bằng cách sử dụng định lý giá trị cuối của biến đổi Laplace rằng

$$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sF(s) \quad (7.6.14)$$

chúng nào các cực của  $sF(s)$  còn có các phần thực âm, chúng ta kết luận được

$$\lim_{t \rightarrow \infty} \Delta\phi(t) = \lim_{s \rightarrow 0} s\Phi(s) = \lim_{s \rightarrow 0} \frac{K_1(1 + \tau_2 s)}{K + (1 + K\tau_1)s + \tau_2 s^2} = 0 \quad (7.6.15)$$

Nói một cách khác, một bộ lọc mạch vòng bậc nhất đơn giản làm cho một PLL có thể bám được các cú nhảy về pha của lối vào.

Hàm truyền (7.6.11) có thể viết lại được dưới dạng chính tắc

$$H(s) = \frac{(2\zeta\omega_n - \omega_n^2/K)s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (7.6.16)$$

Ở đây

$$\omega_n = \sqrt{\frac{K}{\tau_2}}, \quad \zeta = \frac{\omega_n(\tau_1 + 1/K)}{2}$$

trong đó  $\omega_n$  là tần số tự nhiên và  $\zeta$  là hệ số làm nhụt.

### BÀI TẬP MINH HỌA

**Bài tập minh họa 7.9 [PLL bậc nhất]** Giả sử rằng

$$G(s) = \frac{1+0.01s}{1+s}$$

và  $K = 1$ , hãy xác định và vẽ đồ thị đáp ứng của PLL với một thay đổi đột ngột có độ cao bậc thang bằng 1 đối với pha lối vào.

**Lời giải**

Ở đây,  $\tau_1 = 0.01$  và  $\tau_2 = 1$ ; do đó  $\omega_n = 1$  và  $\zeta = 0.505$ . Do vậy

$$H(s) = \frac{0.01s + 1}{s^2 + 1.01s + 1}$$

Như thế, đáp ứng đối với  $\phi(t) = u(t)$ , nghĩa là  $\Phi(s) = 1/s$ , được cho bởi

$$\hat{\Phi}(s) = \frac{0.01s + 1}{s^3 + 1.01s^2 + s + 1}$$

*Để xác định và vẽ đáp ứng thời gian  $\hat{\phi}(t)$  đối với lối vào  $u(t)$ , chúng ta hãy* để ý rằng chúng ta phải xác định lối ra của một hệ thống có hàm truyền  $H(s)$  đối với lối vào  $u(t)$ . Điều này có thể làm được dễ nhất bằng cách sử dụng kỹ thuật không gian trạng thái. Chúng ta áp dụng hàm MATLAB **tf2ss.m** mà nó cho chúng ta mô hình không gian trạng thái của một hệ thống được mô tả bằng hàm truyền của nó. Sau khi xác định biểu diễn không gian trạng thái của hệ thống, chúng ta nhận được đáp ứng bậc thang bằng cách tính bằng số.

*Hàm **tf2ss.m** nhận tử số (numerator, trong MATLAB script cho bài tập này thì tử số được viết tắt trong hàm **tf2ss.m** là num – chú thích của người dịch) và mẫu*

số (*denominator*, viết tắt là *den* trong *script* cho bài tập này – *chú thích của người dịch*) của một hàm truyền  $H(s)$  và cho ra  $A$ ,  $B$ ,  $C$  và  $D$ , là mô tả không gian trạng thái của nó, dưới dạng

$$\begin{cases} \frac{d}{dt}x(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

Biểu diễn này có thể xấp xỉ được bằng

$$\begin{cases} x(t + \Delta t) = x(t) + Ax(t)\Delta t + Bu(t)\Delta t \\ y(t) = Cx(t) + Du(t) \end{cases}$$

hay tương đương là

$$\begin{cases} x(i + 1) = x(i) + Ax(i)\Delta t + Bu(i)\Delta t \\ y(i) = Cx(i) + Du(i) \end{cases}$$

Đối với bài tập này, việc chọn  $u(t)$  là một hàm bậc thang còn các véc-tơ tử số và mẫu số của  $H(s)$  lần lượt là  $[0.01 \quad 1]$  và  $[1 \quad 1.01 \quad 1]$  là đủ. Với lựa chọn này về các véc-tơ tử số và mẫu số, các tham số không gian trạng thái của hệ thống sẽ là

$$A = \begin{bmatrix} -1.01 & -1 \\ 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$C = [0.01 \quad 1]$$

$$D = 0$$

Đồ thị của tín hiệu lối ra của PLL được cho trên hình 7.35.

Như chúng ta có thể thấy từ hình 7.35, lối ra của PLL rốt cục cũng bám theo lối vào; tuy nhiên tốc độ mà nó bám theo lối vào phụ thuộc vào các tham số của bộ lọc mạch vòng và  $K$  là hằng số tỷ lệ của VCO.

MATLAB *script* cho bài tập này được cho dưới đây.

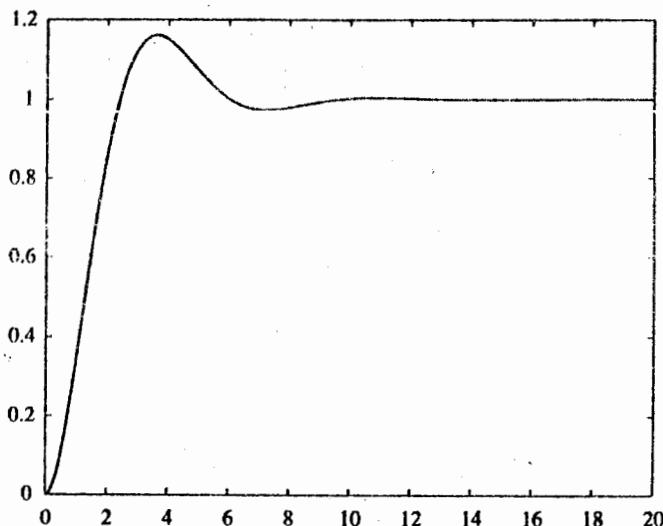
### m-file

```
% MATLAB script for Illustrative Problem 9, Chapter 7
echo on
num=[0.01 1];
```

```

den=[1 1.01 1];
[a,b,c,d]=tf2ss(num,den);
dt=0.01;
u=ones(1,2000);
x=zeros(2,2001);
for i=1:2000
    x(:,i+1)=x(:,i)+dt.*a*x(:,i)+dt.*b*u(i);
    y(i)=c*x(:,i);
    echo off;
end
echo on;
t=[0:dt:20];
plot(t(1:2000),y)

```



Hình 7.35 Đáp ứng của PLL đối với một thay đổi đột ngột trong pha lối vào trong bài tập minh họa 7.9.

### 7.6.2 ĐỒNG BỘ ĐỒNG HỒ

Trong chương 5 và trong chương này chúng ta đã thấy rằng một thực hiện phổ biến của máy thu tối ưu là sử dụng các mạch lọc phoi hợp và các bộ lấy mẫu tại lối ra mạch lọc phoi hợp. Trong tất cả các trường hợp này, chúng ta đã giả thiết rằng máy thu biết một cách đầy đủ về thời điểm lấy mẫu và có thể lấy mẫu hoàn hảo tại thời điểm này. Các hệ thống nhằm có được loại đồng bộ này giữa máy phát và máy thu được gọi là các hệ thống *đồng bộ symbol*, *đồng bộ đồng hồ*.

hay đồng bộ định thời.

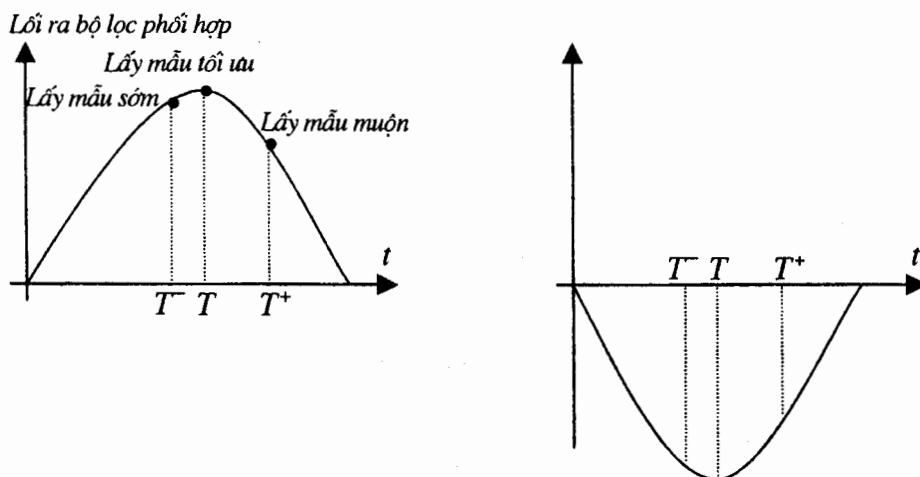
Một thực hiện đơn giản việc đồng bộ đồng hồ là áp dụng một *cổng sorm-muộn*. Hoạt động của một cổng sorm muộn dựa trên thực tế rằng trong một hệ thống truyền tin PAM thì lối ra của bộ lọc phối hợp là hàm tự tương quan của tín hiệu xung cơ sở được sử dụng trong hệ thống PAM (có thể có sự xê dịch nào đó). Hàm tự tương quan đạt cực đại tại thời điểm lấy mẫu tối ưu và là một hàm đối xứng. Điều này có nghĩa là, khi không có tạp nhiễu, tại các thời điểm lấy mẫu  $T^+ = T + \delta$  và  $T^- = T - \delta$ , lối ra của bộ lấy mẫu sẽ như nhau, nghĩa là

$$y(T^+) = y(T^-) \quad (7.6.17)$$

Trong trường hợp này, thời điểm lấy mẫu tối ưu hiển nhiên là điểm giữa nằm giữa các thời điểm lấy mẫu sớm và muộn:

$$T = \frac{T^+ + T^-}{2} \quad (7.6.18)$$

Bây giờ ta hãy giả sử rằng ta đang lấy mẫu không phải tại điểm lấy mẫu tối ưu  $T$  mà thay vào đó ta đang lấy mẫu tại  $T_1$ . Nếu ta lấy thêm hai mẫu nữa tại  $T^+ = T_1 + \delta$  và  $T^- = T_1 - \delta$ , các mẫu này sẽ không đối xứng qua thời điểm lấy mẫu tối ưu  $T$  và do vậy sẽ không bằng nhau. Một hàm tự tương quan tiêu biểu đối với các xung tới âm và dương và 3 giá trị mẫu được thể hiện trên hình 7.36.



Hình 7.36 Lối ra bộ lọc phối hợp và các mẫu sớm và muộn.

Ở đây,  $T^- = T - \delta_1$ ,  $T^+ = T + \delta_2$ , trong đó

$$\delta_1 < \delta_2 \quad (7.6.19)$$

và, như hình vẽ cho thấy, điều này dẫn đến

$$|y(T^-)| > |y(T^+)| \quad (7.6.20)$$

Cũng như vậy, trong trường hợp này

$$T < T_1 = \frac{T^- + T^+}{2} \quad (7.6.21)$$

Do vậy, khi  $|y(T^-)| > |y(T^+)|$ , thời điểm lấy mẫu chính xác sẽ nằm trước thời điểm lấy mẫu đã được giả định và việc lấy mẫu cần phải được thực hiện sớm lên. Ngược lại, khi  $|y(T^-)| < |y(T^+)|$ , thời điểm lấy mẫu cần phải chậm lại. Hiển nhiên, khi  $|y(T^-)| = |y(T^+)|$  thì thời điểm lấy mẫu là đúng và không cần phải hiệu chỉnh gì cả.

Hệ thống đồng bộ công sớm-muộn như vậy nhận ba giá trị mẫu tại  $T_1$ ,  $T^- = T - \delta$  và  $T^+ = T + \delta$  rồi so sánh  $|y(T^-)|$  và  $|y(T^+)|$ , tùy theo tương quan giữa các giá trị của chúng mà đưa ra tín hiệu để hiệu chỉnh thời điểm lấy mẫu.

### BÀI TẬP MINH HỌA

**Bài tập minh họa 7.10 [Đồng bộ đồng hồ]** Một hệ thống truyền tin PAM nhị phân sử dụng một dạng sóng cosin nâng với hệ số uốn 0.4. Tốc độ truyền dẫn là 4800 b/s. Hãy viết một file MATLAB mô phỏng hoạt động của một công sớm-muộn cho hệ thống này.

#### Lời giải

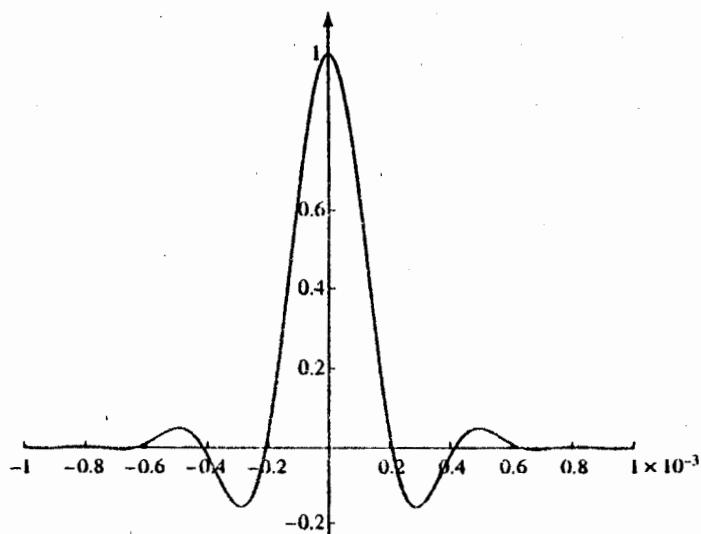
Do tốc độ là 4800 b/s, chúng ta có

$$T = \frac{1}{4800} \quad (7.6.22)$$

và với  $\alpha = 0.4$ , biểu thức đổi với một dạng sóng cosin nâng trở thành

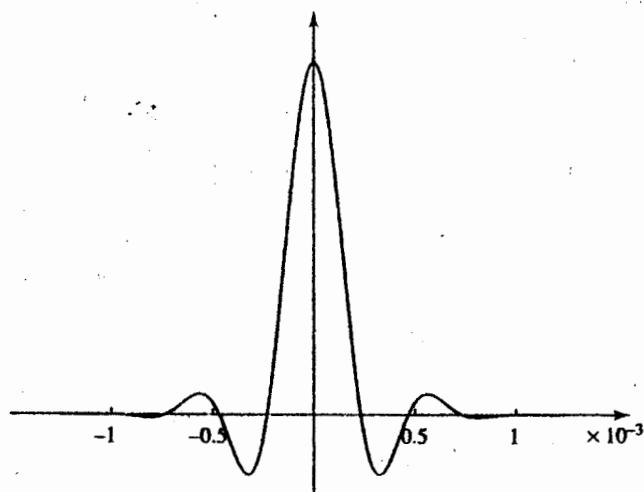
$$\begin{aligned} x(t) &= \sin c(4800t) \frac{\cos(4800 \times 0.4\pi t)}{1 - 4 \times 0.16 \times 4800^2 t^2} \\ &= \sin c(4800t) \frac{\cos 1920\pi t}{1 - 1.4746 \times 10^7 t^2} \end{aligned} \quad (7.6.23)$$

Tín hiệu này hiển nhiên trải từ  $-\infty$  tới  $+\infty$ . Đồ thị của tín hiệu này được thể hiện trên hình 7.37.



Hình 7.37 Tín hiệu cosin nâng trong bài tập minh họa 7.10.

Từ hình vẽ 7.37, rõ ràng là đối với mọi mục đích thực tế việc chỉ xét khoảng thời gian với  $|t| \leq 0.6 \cdot 10^{-3}$  là đủ, khoảng này vào quãng  $[-3T, 3T]$ . Việc cắt xung cosin nâng còn khoảng này và tính hàm tự tương quan dẫn đến dạng sóng được thể hiện trên hình vẽ 7.38.



Hình 7.38 Hàm tự tương quan của tín hiệu cosin nâng.

Trong MATLAB script được cho dưới đây, trước tiên tín hiệu cosin nâng và hàm tự tương quan sẽ được tính và vẽ đồ thị. Trong thí dụ cụ thể này, độ dài hàm tự tương quan là 1201 và trị cực đại (tức là thời điểm lấy mẫu tối ưu) xảy ra tại phần tử thứ 600. Hai trường hợp được kiểm tra: một là khi thời điểm lấy mẫu sai là 700 và một là khi thời điểm lấy mẫu sai là 500. Trong cả hai trường hợp, cồng sóm-muộn đều hiệu chỉnh thời điểm lấy mẫu về thời điểm tối ưu 600.

### m-file

```
% MATLAB script for Illustrative Problem 10, Chapter 7.  
echo on  
alpha=0.4;  
T=1/4800;  
t=[-3*T:1.001*T/100:3*T];  
x=sinc(t./T).*(cos(pi*alpha*t./T)./(1-4*alpha^2*t.^2/T^2));  
pause % Press any key to see a plot of x(t).  
plot(t,x)  
y=xcorr(x);  
ty=[t-3*T,t(2:length(t))+3*T];  
pause % Press any key to see a plot of the autocorrelation of x(t).  
plot(ty,y);  
d=60; % Early and late advance and delay  
ee=0.01; % Precision  
e=1; % Step size  
n=700; % The incorrect sampling time  
while abs(abs(y(n+d))-abs(y(n-d)))>=ee  
    if abs(y(n+d))-abs(y(n-d))>0  
        n=n+e;  
    elseif abs(y(n+d))-abs(y(n-d))<0  
        n=n-e;  
    end  
    echo off ;  
end  
echo on ;  
pause % Press any key to see the corrected sampling time  
n  
n=500; % Another incorrect sampling time  
while abs(abs(y(n+d))-abs(y(n-d)))>=ee  
    if abs(y(n+d))-abs(y(n-d))>0  
        n=n+e;  
    elseif abs(y(n+d))-abs(y(n-d))<0
```

```

n=n-e;
end
echo off ;
end
echo on ;
pause % Press any key to see the corrected sampling time
n

```

## CÁC BÀI TẬP

**7.1** Trong một hệ thống PAM điều chế biên độ sóng mang, bộ lọc phát có đặc tính phổ căn bậc hai cosin nâng với hệ số uốn lắc  $\alpha = 1$ . Tần số sóng mang là  $f_c = 40/T$ . Hãy ước lượng và vẽ đồ thị phổ của tín hiệu băng gốc và của tín hiệu PAM được điều chế biên độ.

**7.2** Lặp lại bài tập 7.1 khi tần số sóng mang là  $f_c = 80/T$ .

**7.3** Lặp lại bài tập 7.1 khi máy phát có đặc tính phổ căn bậc hai nhị phân đôi (*duobinary*).

**7.4** Mục đích của bài tập này là để thể hiện rằng (7.2.9) và (7.2.10) vẫn đúng thông qua cách tính toán số với (7.2.9) bằng MATLAB. Xung  $g_T(t)$  có thể giả sử được là xung chữ nhật, nghĩa là

$$g_T(t) = \begin{cases} 1, & 0 \leq t \leq 2 \\ 0, & \text{với các giá trị khác của } t \end{cases}$$

Cho tần số sóng mang  $f_c = 2000$  Hz. Hãy sử dụng tốc độ lấy mẫu  $F_s = 20\,000$  mẫu trên giây với dạng sóng tín hiệu  $\psi(t)$  được cho bởi (7.2.6) và hãy tính năng lượng của  $\psi(t)$  bằng cách xấp xỉ tích phân trong (7.2.8) bằng tổng

$$\frac{1}{N} \sum_{n=0}^{N-1} \psi^2(nT_s) = \frac{1}{N} \sum_{n=0}^{N-1} \psi^2\left(\frac{n}{F_s}\right)$$

trong đó  $N = 40\,000$  mẫu. Hãy viết một chương trình MATLAB để tạo ra các mẫu  $\psi(n/F_s)$ , và thực hiện việc tính toán năng lượng tín hiệu như đã mô tả ở trên.

**7.5** Việc tính tương quan chéo của tín hiệu thu được  $r(t)$  với  $\psi(t)$  như đã cho bởi (7.2.14) có thể thực hiện được bằng số nhờ sử dụng MATLAB. Hãy viết một chương

trình MATLAB dùng để tính lối ra bộ tương quan

$$y(n) = \sum_{k=0}^n r\left(\frac{k}{F_s}\right) \psi\left(\frac{k}{F_s}\right), \quad n = 0, 1, \dots, N-1$$

trong đó  $F_s$  là tần số lấy mẫu. Hãy tính và vẽ đồ thị  $y(n)$  khi  $r(t) = \psi(t)$ , trong đó  $\psi(t)$  là dạng sóng được mô tả trong bài tập 7.4 còn  $F_s = 20\,000$  Hz.

**7.6** Hãy tính và vẽ đồ thị hàm tương quan  $\{y(n)\}$  trong bài tập 7.5 khi tín hiệu  $g_T(t)$  là

$$g_T(t) = \begin{cases} \frac{1}{2}(1 - \cos \pi t), & 0 \leq t \leq 2 \\ 0, & \text{với các giá trị } t \text{ khác} \end{cases}$$

đối với cùng các tham số đã cho trong bài tập 7.4.

**7.7** Trong bài tập minh họa 7.2, tám dạng sóng PSK đã có một đường bao là hằng số. Thay cho xung  $g_T(t)$  hình chữ nhật, giả sử rằng dạng xung tín hiệu là

$$g_T(t) = \begin{cases} \frac{1}{2}(1 - \cos 2\pi t/T), & 0 \leq t \leq T \\ 0, & \text{với các giá trị } t \text{ khác} \end{cases}$$

Hãy viết một chương trình MATLAB để tính và vẽ đồ thị các dạng sóng 8-PSK đối với trường hợp trong đó  $f_c = 6/T$ .

**7.8** Hãy viết một chương trình MATLAB để tính toán bằng số tương quan chéo của tín hiệu thu được  $r(t)$  đối với một tín hiệu PSK với hai hàm cơ sở được cho bởi (7.3.9). Nghĩa là hãy tính

$$y_c(n) = \sum_{k=0}^n r\left(\frac{k}{F_s}\right) \psi_1\left(\frac{k}{F_s}\right), \quad n = 0, 1, \dots, N-1$$

$$y_s(n) = \sum_{k=0}^n r\left(\frac{k}{F_s}\right) \psi_2\left(\frac{k}{F_s}\right), \quad n = 0, 1, \dots, N-1$$

trong đó  $N$  là số mẫu của  $r(t)$ ,  $\psi_1(t)$  và  $\psi_2(t)$ . Hãy tính và vẽ các dãy giá trị hàm tương quan này khi

$$r(t) = s_{mc} \psi_1(t) + s_{ms} \psi_2(t)$$

$$g_T(t) = \begin{cases} 2, & 0 \leq t \leq 2 \\ 0, & \text{với các giá trị } t \text{ khác} \end{cases}$$

$f_c = 1\ 000$  Hz,  $F_s = 10\ 000$  mẫu trên giây còn điểm tín hiệu được truyền đi thì như được cho như sau

a)  $s_m = (s_{mc}, s_{ms}) = (1, 0)$ ; b)  $s_m = (-1, 0)$ ; c)  $s_m = (0, 1)$ .

**7.9** Hãy viết một chương trình MATLAB để thực hiện một mô phỏng Monte-Carlo một hệ thống truyền tin 4-PSK như đã được mô tả trong bài tập minh họa 7.3 song hãy cải biến bộ tách tín hiệu sao cho nó tính toán pha của tín hiệu thu được  $\theta_r$  như đã cho bởi (7.3.16) và chọn ra điểm tín hiệu nào có góc pha gần với  $\theta_r$  nhất.

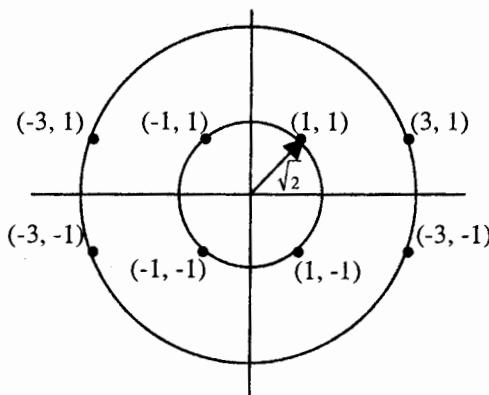
**7.10** Viết một chương trình MATLAB mà nó thực hiện một bộ mã hoá vi sai và một bộ giải mã vi sai đối với một hệ thống 4-DPSK. Hãy kiểm tra hoạt động của bộ mã hoá và giải mã bằng cách cho một chuỗi các *symbol* 2 bít qua bộ mã hoá và giải mã mắc nối tiếp nhau rồi xác minh rằng chuỗi lỗi ra trùng với chuỗi lỗi vào.

**7.11** Hãy viết một chương trình MATLAB mà nó thực hiện một mô phỏng Monte-Carlo một hệ thống truyền tin DPSK nhị phân. Trong trường hợp này, các pha của tín hiệu được truyền đi là  $\theta = 0$  và  $\theta = 180^\circ$ . Một thay đổi pha  $\theta = 0$  tương ứng với việc truyền một con 0. Một thay đổi pha  $\theta = 180^\circ$  tương ứng với việc truyền một con 1. Hãy tiến hành mô phỏng đối với  $N = 10\ 000$  bít tại các giá trị khác nhau của tham số SNR  $E_b/N_0$ . Để thuận tiện  $E_b$  được chuẩn hoá bằng 1. Khi đó, với  $\sigma^2 = N_0/2$ , thì SNR là  $E_b/N_0 = 1/2\sigma^2$ , trong đó  $\sigma^2$  là phương sai của thành phần tạp âm cộng. Do đó, SNR có thể điều khiển chạy được bằng cách thay đổi tỷ lệ phương sai của thành phần tạp âm cộng. Hãy vẽ và so sánh tỷ lệ lỗi của DPSK nhị phân nhận được từ mô phỏng và giá trị lý thuyết của xác suất lỗi được cho bởi (7.3.23).

**7.12** Hãy viết một chương trình MATLAB tạo ra và vẽ các dạng sóng 8-QAM được cho theo (7.4.2) đối với biểu đồ sao tín hiệu được biểu diễn trên hình P7.12. Hãy giả sử rằng dạng sóng xung  $g_T(t)$  là hình chữ nhật, nghĩa là

$$g(t) = \begin{cases} 1, & 0 \leq t \leq T \\ 0, & \text{với các giá trị } t \text{ khác} \end{cases}$$

còn tần số sóng mang là  $f_c = 8/T$ .

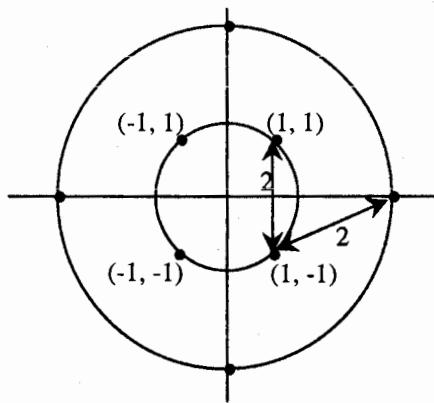


Hình P7.12

**7.13** Lặp lại bài tập 7.12 khi dạng sóng xung  $g_T(t)$  được cho theo

$$g_T(t) = \begin{cases} \frac{1}{2}(1 - \cos 2\pi t/T), & 0 \leq t \leq T \\ 0, & \text{với các giá trị } t \text{ khác} \end{cases}$$

**7.14** Hãy viết một chương trình MATLAB mà nó thực hiện một mô phỏng Monte-Carlo của một hệ thống 8-QAM với biểu đồ sao tín hiệu như trên hình P7.12. Hãy tiến hành mô phỏng đối với  $N = 10\,000$  symbol 3 bít tại các giá trị khác nhau của tham số SNR  $E_{avb}/N_0$ . Việc chuẩn hoá  $E_{avb}$  thành đơn vị là để cho tiện. Khi đó, với  $\sigma^2 = N_0/2$ , thì SNR là  $E_{avb}/N_0 = 1/2\sigma^2$ , trong đó  $\sigma^2$  là phương sai của mỗi một trong hai thành phần tạp âm cộng. Hãy vẽ đồ thị và so sánh tỷ lệ lỗi tính được của hệ thống QAM này với giới hạn chặn trên đối với xác suất lỗi lý thuyết được cho bởi (7.4.12).



Hình P7.15.

7.15 Lặp lại mô phỏng trong bài tập 7.14 song đổi với biểu đồ sao 8 tín hiệu được cho thấy trên hình P7.15. Hãy so sánh các xác suất lỗi đối với hai biểu đồ sao tín hiệu 8-QAM này và chỉ ra biểu đồ sao tín hiệu nào cho chất lượng tốt hơn.

7.16 Hãy xét các tín hiệu FSK nhị phân có dạng

$$u_1(t) = \sqrt{\frac{2E_b}{T_b}} \cos 2\pi f_1 t, \quad 0 \leq t \leq T_b$$

$$u_2(t) = \sqrt{\frac{2E_b}{T_b}} \cos 2\pi f_2 t, \quad 0 \leq t \leq T_b$$

$$f_2 = f_1 + \frac{1}{2T_b}$$

Cho  $f_1 = 1000/T_b$ . Bằng việc lấy mẫu hai dạng sóng nói trên với tốc độ lấy mẫu  $F_s = 5000/T_b$ , chúng ta nhận được 5000 mẫu trong khoảng bít  $0 \leq t \leq T_b$ . Hãy viết một chương trình MATLAB mà nó tạo ra 5000 mẫu cho từng tín hiệu  $u_1(t)$  và  $u_2(t)$  rồi tính hàm tương quan chéo

$$\sum_{n=0}^{N-1} u_1\left(\frac{n}{F_s}\right) u_2\left(\frac{n}{F_s}\right)$$

rồi qua đó xác minh bằng số điều kiện về tính trực giao đối với  $u_1(t)$  và  $u_2(t)$ .

7.17 Hãy sử dụng chương trình MATLAB đã cho trong bài tập minh họa 7.7 để tính và vẽ đồ thị các lỗi ra bộ tương quan khi tín hiệu thu được là

$$r(t) = \cos(2\pi f_1 t + \frac{\pi}{2}), \quad 0 \leq t \leq T_b$$

7.18 Hãy sử dụng chương trình MATLAB đã cho trong bài tập minh họa 7.7 để tính và vẽ đồ thị các lỗi ra bộ tương quan khi tín hiệu đã truyền là  $u_2(t)$  và tín hiệu thu được là

$$r(t) = \cos(2\pi f_2 t + \frac{\pi}{4}), \quad 0 \leq t \leq T$$

7.19 Hãy viết một chương trình MATLAB mà nó thực hiện một mô phỏng Monte-Carlo của một hệ thống truyền tin FSK bốn mức và áp dụng các tần số

$$f_k = f_1 + \frac{k}{T}, \quad k = 0, 1, 2, 3$$

Bộ tách tín hiệu là một bộ tách sóng luật bình phương. Hãy tiến hành mô phỏng đổi với  $N=10\,000$  symbol 2 bít tại các giá trị khác nhau của tham số SNR  $E_b/N_0$  và ghi lại số các lỗi symbol và lỗi bít. Hãy vẽ đồ thị và so sánh các tỷ lệ lỗi bít và tỷ lệ lỗi symbol đã nhận được với các xác suất lỗi bít và xác suất lỗi symbol lý thuyết được cho trong các biểu thức (7.5.19) và (7.5.21).

**7.20** Trong bài tập minh họa 7.9, chúng ta đã giả sử rằng pha lỗi vào có một bước nhảy đột ngột và việc mô phỏng đã cho thấy rằng một bộ lọc mạch vòng bậc một có thể bám được một sự thay đổi pha như thế. Nay giờ hãy giả sử rằng các thay đổi lỗi vào tương ứng với một đường dốc, nghĩa là bắt đầu tăng một cách tuyến tính. Hãy mô phỏng hoạt động của một PLL bậc nhất trong trường hợp này và xác định xem liệu mạch vòng khoá pha có khả năng bám được một thay đổi như thế không.

**7.21** Lặp lại bài tập minh họa 7.10 với một dạng xung chữ nhật trong trường hợp có tạp âm cộng trăng Gauss (AWGN) đổi với các giá trị SNR là 20, 10, 5 và 0 dB.

## CHƯƠNG 8

# DUNG LƯỢNG KÊNH VÀ MÃ HOÁ KÊNH

### 8.1 GIỚI THIỆU CHUNG

Mục đích của mọi hệ thống thông tin là truyền thông tin được tạo bởi một nguồn tin từ nơi này đến nơi khác. Môi trường qua đó thông tin được truyền được gọi là kênh thông tin. Chúng ta đã thấy trong chương 4 rằng lượng thông tin của một nguồn tin được đánh giá bởi *entropy* của nguồn tin đó, và đơn vị phổ biến nhất của đại lượng này là bit. Chúng ta cũng đã thấy rằng một mô hình toán học thích hợp đối với một nguồn tin là một quá trình ngẫu nhiên.

Trong chương này, chúng ta sẽ xem xét các mô hình toán học phù hợp cho các kênh thông tin. Chúng ta cũng sẽ thảo luận về một đại lượng được gọi là dung lượng kênh, nó xác định đối với một kênh thông tin bất kỳ và cho ta một giới hạn cơ bản về lượng tin có thể truyền qua kênh. Cụ thể, chúng ta sẽ xét hai loại kênh là kênh nhị phân đối xứng (BSC: *Binary Symmetric Channel*) và kênh với tạp âm cộng trăng Gauss (AWGN).

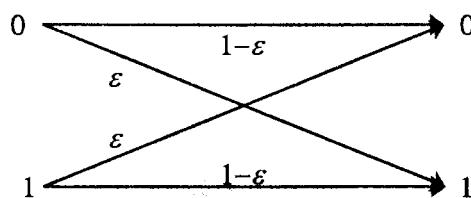
Phần hai của chương sẽ được dành cho các kỹ thuật mã hoá để truyền tin tin cậy trên các kênh thông tin. Chúng ta sẽ bàn tới hai phương pháp mã hoá được sử dụng rộng rãi nhất, cụ thể là mã hoá khối và mã hoá chập. Các kỹ thuật mã hoá và giải mã đối với các mã này và chất lượng của chúng được thảo luận chi tiết trong các mục cuối của chương này.

## 8.2 MÔ HÌNH KÊNH VÀ DUNG LƯỢNG KÊNH

Một kênh thông tin truyền dẫn tín hiệu mang thông tin tới đích nhận tin. Trong quá trình truyền dẫn này, tín hiệu mang thông tin bị biến đổi nhiều. Một số biến đổi này là tất định về bản chất - thí dụ như suy giảm, méo tuyến tính và méo phi tuyến; một số biến đổi khác là ngẫu nhiên - thí dụ như tạp âm cộng, phad-đing đa đường... Do các biến đổi tất định có thể xem được như các trường hợp đặc biệt của các biến đổi ngẫu nhiên nên theo nghĩa tổng quát nhất mô hình toán học cho một kênh thông tin là một sự phụ thuộc ngẫu nhiên giữa tín hiệu lối vào và tín hiệu lối ra.

### 8.2.1 MÔ HÌNH KÊNH

Trong trường hợp đơn giản nhất, một kênh có thể được mô hình như một xác suất có điều kiện liên hệ giữa mỗi đầu ra của kênh với đầu vào tương ứng của nó. Mô hình như thế được gọi là một kênh không nhớ rời rạc (DMC: *Discrete-Memoryless Channel*) và được biểu diễn đầy đủ bởi bảng chữ cái đầu vào  $X$  của kênh, bảng chữ cái đầu ra  $Y$  của kênh, và *ma trận các xác suất chuyển*  $p(y|x)$ , được cho với mọi  $x \in X, y \in Y$ . Một trường hợp đặc biệt của kênh không nhớ rời rạc là kênh đối xứng nhị phân (BSC: *Binary Symmetric Channel*) mà nó có thể xem được như một mô hình toán học đối với việc truyền dẫn nhị phân trên một kênh Gauss với các quyết định cứng ở đầu ra. Một kênh đối xứng nhị phân tương ứng với trường hợp  $X=Y=\{0, 1\}$  và  $p(y=0|x=1) = p(y=1|x=0) = \varepsilon$ . Một mô hình giản lược đối với kênh này được thể hiện trên hình 8.1. Tham số  $\varepsilon$  được gọi là xác suất chéo (*crossover probability*) của kênh.



Hình 8.1 Một kênh đối xứng nhị phân (BSC)

### 8.2.2 DUNG LƯỢNG KÊNH

Về mặt định nghĩa, dung lượng kênh là tốc độ truyền tin tin cậy lớn nhất có thể đạt được qua kênh. Việc truyền tin tin cậy có thể có được khi tồn tại một dãy các mã với độ dài khối tăng lên, mà đối với nó xác suất lỗi tiến đến không khi độ dài khối tăng. Dung lượng kênh được ký hiệu là  $C$ ; về mặt định nghĩa, với các tốc độ  $R < C$  việc truyền tin tin cậy qua kênh là có thể; với các tốc độ  $R > C$  thì không thể truyền tin tin cậy qua kênh.

Kết quả có tính nền tảng của Shannon về lý thuyết thông tin nói rằng đối với các kênh không nhớ rời rạc thì dung lượng kênh được cho bởi biểu thức sau:

$$C = \max_{p(x)} I(X; Y) \quad (8.2.1)$$

trong đó  $I(X; Y)$  ký hiệu cho lượng tin tương hő giữa  $X$  (lối vào của kênh) và  $Y$  (lối ra của kênh), còn việc lấy cực đại được thực hiện trên mọi phân bố xác suất lỗi vào của kênh.

Lượng tin tương hő giữa hai biến ngẫu nhiên  $X$  và  $Y$  được xác định theo

$$I(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x)p(y|x) \log \frac{p(x, y)}{p(x)p(y)} \quad (8.2.2)$$

trong đó, lượng tin tương hő được tính theo bit và logarit tính theo cơ số 2.

Đối với trường hợp kênh đối xứng nhị phân, dung lượng kênh được cho bởi quan hệ đơn giản sau:

$$C = 1 - H_b(\varepsilon) \quad (8.2.3)$$

trong đó,  $\varepsilon$  là xác suất chéo của kênh và  $H_b(\cdot)$  ký hiệu cho hàm *entropy* nhị phân:

$$H_b(x) = -x \log(x) - (1-x) \log(1-x) \quad (8.2.4)$$

Một mô hình kênh quan trọng khác là kênh AWGN có băng thông hạn chế với một ràng buộc về công suất lỗi vào. Kênh này được mô hình hóa như được thể hiện trên hình 8.2.

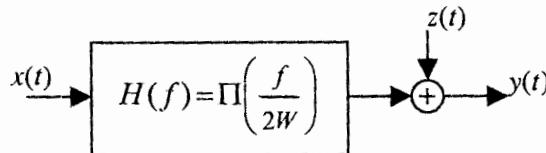
Kênh này bị hạn chế về băng thông còn  $[-W, W]$ , tạp âm là Gauss và trắng với mật độ phổ công suất (2 phía) là  $N_0/2$ , còn lỗi vào kênh là một quá trình thoả mãn

ràng buộc công suất lối vào  $P$ . Shannon đã chỉ ra rằng dung lượng kênh này, tính theo bit/s, được cho bởi

$$C = W \log \left( 1 + \frac{P}{N_0 W} \right) \quad \text{bit / s} \quad (8.2.5)$$

Đối với một kênh AWGN thời gian rời rạc với ràng buộc công suất đầu vào  $P$  và phương sai của tạp âm là  $\sigma^2$ , dung lượng tính theo bit/lần truyền được cho bởi

$$C = \frac{1}{2} \log \left( 1 + \frac{P}{\sigma^2} \right) \quad (8.2.6)$$



Hình 8.2 Kênh có băng thông hạn chế với tạp âm cộng trắng Gauss

## BÀI TẬP MINH HỌA

**Bài tập minh họa 8.1 [Dung lượng kênh đối xứng nhị phân]** Dữ liệu nhị phân được truyền qua một kênh AWGN bằng cách sử dụng điều chế BPSK và giải mã quyết định cứng tại lối ra nhờ sử dụng tách tín hiệu lọc phôi hợp tối ưu.

1. Hãy vẽ đồ thị xác suất lỗi của kênh như một hàm của

$$\gamma = \frac{E}{N_0} \quad (8.2.7)$$

trong đó,  $E$  là năng lượng của mỗi tín hiệu BPSK còn  $N_0/2$  là mật độ phổ công suất của tạp âm. Hãy giả thiết rằng  $\gamma$  thay đổi từ  $-20\text{dB}$  đến  $20\text{ dB}$ .

2. Vẽ đồ thị của dung lượng kênh như một hàm của  $\gamma$ .

### Lời giải

1. Xác suất lỗi của BPSK với tách tín hiệu tối ưu được cho bởi

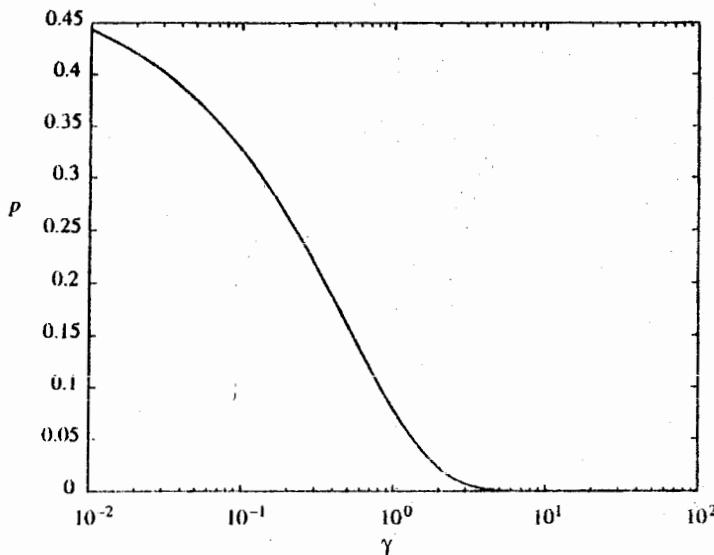
$$P = Q(\sqrt{2\gamma}) \quad (8.2.8)$$

Đồ thị tương ứng được thể hiện trên hình 8.3.

2. Ở đây chúng ta sử dụng quan hệ

$$C = 1 - H_b(p) = 1 - H_b(Q(\sqrt{2\gamma})) \quad (8.2.9)$$

để có được một đồ thị của  $C$  theo  $\gamma$ . Đồ thị này được thể hiện trên hình 8.4.

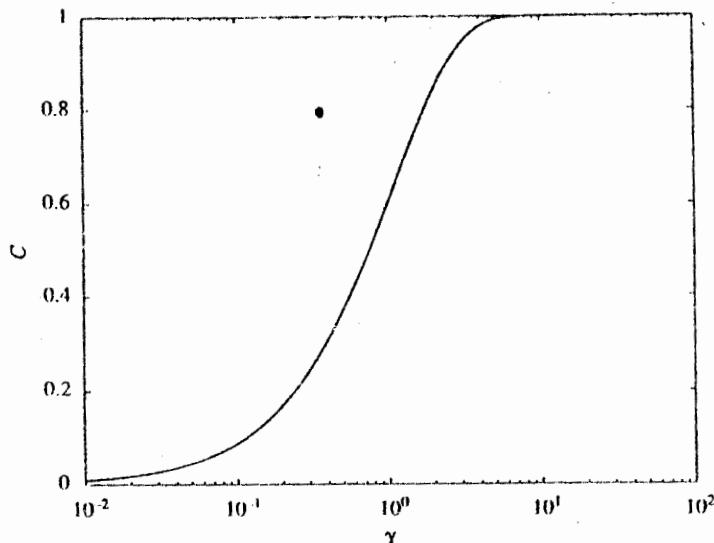


Hình 8.3 Xác suất lỗi của BPSK theo  $\gamma=E/N_0$

MATLAB script cho bài tập này được cho dưới đây.

#### m-file

```
% Chương trình MatLab minh họa cho bài tập 8.1
echo on
gamma_db=[-20:0.1:20];
gamma=10.^gamma_db./10;
p_error=q(sqrt(2.*gamma));
capacity=1.-entropy2(p_error);
pause % Bấm phím bất kỳ để xem đồ thị Pe theo SNR/bit
clf
semilogx(gamma,p_error)
xlabel('SNR/bit')
title('Xác suất lỗi theo SNR/bit')
ylabel('Xác suất lỗi')
pause % Bấm phím bất kỳ để xem đồ thị của dung lượng kênh theo SNR/bit
clf
semilogx(gamma,capacity)
xlabel('SNR/bit')
title('Dung lượng kênh theo SNR/bit')
ylabel('Dung lượng kênh')
```

Hình 8.4 Dung lượng kênh theo  $\gamma = E/N_0$ 

## BÀI TẬP MINH HỌA

### Bài tập minh họa 8.2 [Dung lượng kênh Gauss]

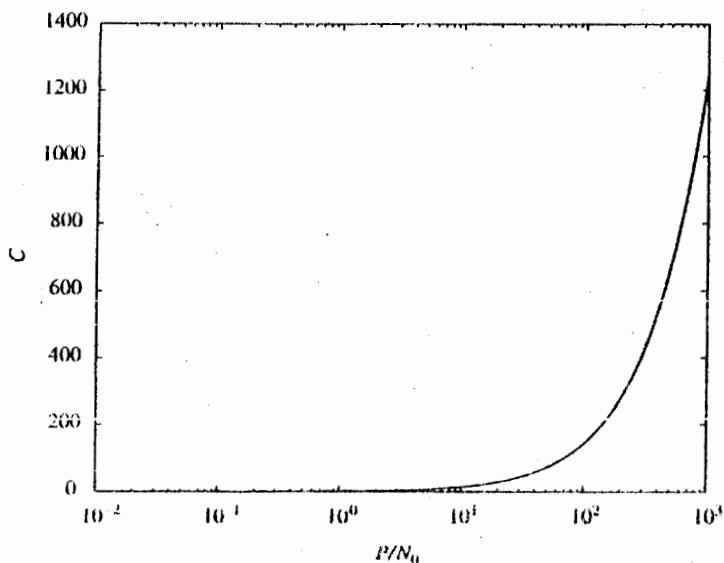
- Hãy vẽ đồ thị dung lượng của kênh AWGN có độ rộng băng  $W = 3000$  Hz như một hàm của  $P/N_0$  với các giá trị của  $P/N_0$  nằm trong khoảng từ  $-20$ dB đến  $30$ dB.
- Vẽ đồ thị dung lượng của một kênh AWGN với  $P/N_0 = 25$ dB như là một hàm của  $W$ . Đặc biệt, khi  $W$  tăng vô hạn thì dung lượng kênh là bao nhiêu?

#### Lời giải

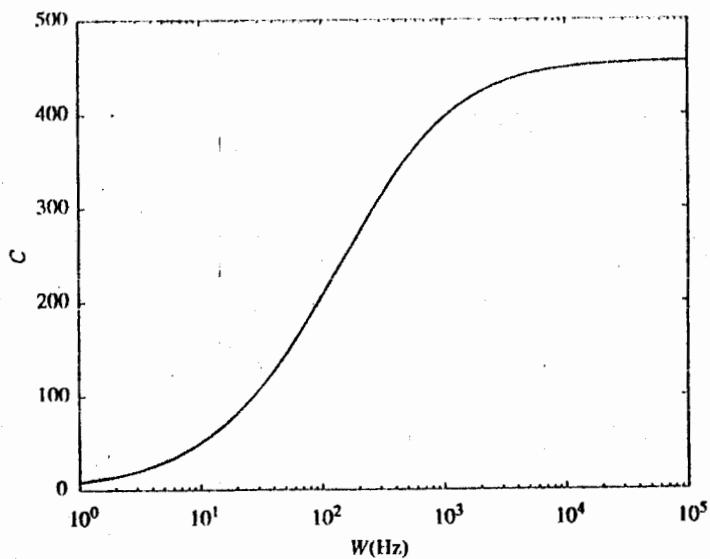
- Đồ thị cần vẽ được cho trên hình 8.5.
- Dung lượng kênh như một hàm của độ rộng băng được vẽ trên hình 8.6. Như thấy được trên các đồ thị, khi  $P/N_0$  hoặc  $W$  tiến tới không, dung lượng của kênh cũng tiến tới không. Tuy nhiên, khi  $P/N_0$  hoặc  $W$  tiến tới vô cùng thì dung lượng kênh thay đổi khác nhau. Trong trường hợp  $P/N_0$  tiến tới vô cùng, dung lượng kênh cũng tiến tới vô cùng như được thể hiện trên hình 8.5. Tuy nhiên, khi  $W$  tiến tới vô cùng thì dung lượng kênh chỉ tiến tới một giới hạn nhất định được xác định bởi  $P/N_0$ . Để xác định giá trị giới hạn này, chúng ta có

$$\lim_{W \rightarrow \infty} W \log_2 \left( 1 + \frac{P}{N_0 W} \right) = \frac{P}{N_0 \ln 2} \quad (8.2.10)$$

$$= 1.4427 \frac{P}{N_0} \quad (8.2.11)$$



Hình 8.5 Dung lượng của một kênh với  $W=3000$  Hz, như một hàm của  $P/N_0$



Hình 8.6 Dung lượng kênh như một hàm của độ rộng băng trong một kênh AWGN ( $P/N_0=25$  dB)

MATLAB script cho bài tập này được cho dưới đây.

### m-file

```
% Chương trình MatLab minh họa cho bài tập 8.2.
echo on
pn0_db=[-20:0.1:30];
pn0=10.^pn0_db./10;
capacity=3000.*log2(1+pn0/3000);
pause % Bấm phím bất kỳ để vẽ đồ thị dung lượng kênh theo P/N0
clf
semilogx(pn0,capacity)
title('Đồ thị dung lượng kênh theo P/N0 trong kênh AWGN')
xlabel('P/N0')
ylabel('Dung lượng kênh (bit/s)')
clear
w=[1:10,12:2:100,105:5:500,510:10:5000,5025:25:20000,20050:50:100000];
pn0_db=25;
pn0=10^(pn0_db/10);
capacity=w.*log2(1+pn0./w);
pause % Bấm phím bất kỳ để vẽ đồ thị dung lượng kênh theo dải tần
clf
semilogx(w,capacity)
title('Đồ thị dung lượng kênh theo dải tần trong kênh AWGN ')
xlabel('Dải tần (Hz)')
ylabel('Dung lượng kênh (bit/s)')
```

## BÀI TẬP MINH HỌA

### Bài tập minh họa 8.3 [Dung lượng kênh AWGN với đầu vào nhị phân]

Một kênh AWGN với đầu vào nhị phân được mô hình hoá bằng các mức lối vào nhị phân  $A$  và  $-A$ , và tạp âm Gauss cộng có trị trung bình bằng không với phương sai  $\sigma^2$ . Trong trường hợp này,  $X=\{-A, A\}$ ,  $Y = \mathbf{R}$  (là trường số thực),  $p(y|X=A) \sim N(A, \sigma^2)$  (là một biến ngẫu nhiên phân bố chuẩn có kỳ vọng bằng  $A$  và phương sai bằng  $\sigma^2$ ), và  $p(y|X=-A) \sim N(-A, \sigma^2)$ . Hãy vẽ đồ thị dung lượng của kênh này như một hàm của  $A/\sigma$ .

### Lời giải

Do tính đối xứng của bài toán, dung lượng kênh đạt được đối với phân bố lối vào là đều, có nghĩa là  $p(X=A) = p(X=-A) = 0.5$ . Đối với phân bố đầu vào này, phân bố

đầu ra được cho bởi

$$p(y) = \frac{1}{2\sqrt{2\pi\sigma^2}} e^{-(y+A)^2/2\sigma^2} + \frac{1}{2\sqrt{2\pi\sigma^2}} e^{-(y-A)^2/2\sigma^2} \quad (8.2.12)$$

và lượng tin tương hỗ giữa đầu vào và đầu ra được cho bởi

$$\begin{aligned} I(X; Y) &= \frac{1}{2} \int_{-\infty}^{\infty} p(y | X = A) \log_2 \frac{p(y | X = A)}{p(y)} dy \\ &\quad + \frac{1}{2} \int_{-\infty}^{\infty} p(y | X = -A) \log_2 \frac{p(y | X = -A)}{p(y)} dy \end{aligned} \quad (8.2.13)$$

Việc lấy tích phân và đổi biến đơn giản dẫn đến

$$I(X; Y) = \frac{1}{2} f\left(\frac{A}{\sigma}\right) + \frac{1}{2} f\left(-\frac{A}{\sigma}\right) \quad (8.2.14)$$

trong đó

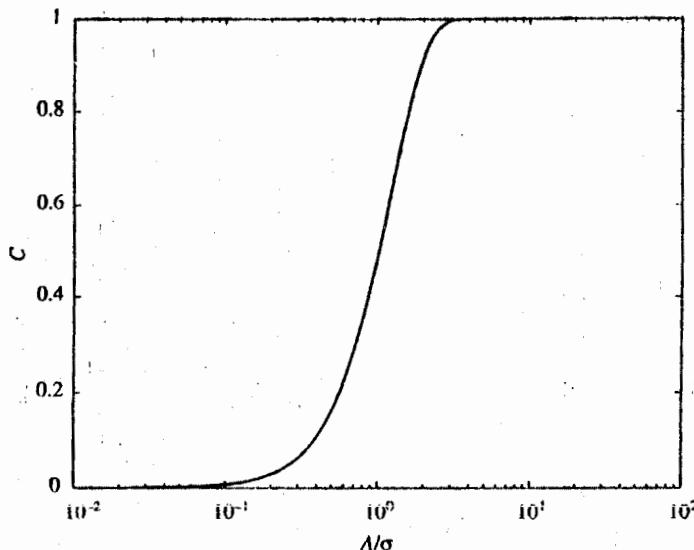
$$f(a) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-(u-a)^2/2} \log_2 \frac{2}{1+e^{-2au}} du \quad (8.2.15)$$

Sử dụng quan hệ này, ta có thể tính được  $I(X; Y)$  đối với một loạt các giá trị của  $A/\sigma$  và vẽ được đồ thị kết quả. Một đồ thị của đường cong tính được được thể hiện trên hình 8.7.

MATLAB script cho bài tập này như sau.

### m-file

```
% Chương trình MatLab cho bài tập minh họa 8.3.
echo on
a_db=[-20:0.2:20];
a=10.^a_db/10;
for i=1:201
    f(i)=quad('il3_8fun',a(i)-5,a(i)+5,1e-3,[],a(i));
    g(i)=quad('il3_8fun',-a(i)-5,-a(i)+5,1e-3,[],-a(i));
    c(i)=0.5*f(i)+0.5*g(i);
    echo off ;
end
echo on ;
pause % Bấm phím bất kỳ để vẽ đồ thị dung lượng kênh theo SNR
semilogx(a,c)
title('Dung lượng kênh theo SNR trong kênh AWGN đầu vào nhị phân')
xlabel('SNR')
ylabel('Dung lượng kênh (số bit/ lần truyền)')
```



Hình 8.7 Dung lượng của một kênh AWGN lối vào nhị phân như một hàm của SNR =  $A/\sigma$ .

### BÀI TẬP MINH HỌA

**Bài tập minh họa 8.4 [So sánh các sơ đồ quyết định cứng và quyết định mềm]** Một kênh đầu vào nhị phân dùng 2 mức lối vào  $A$  và  $-A$ . Đầu ra của kênh là tổng của tín hiệu lối vào và tạp âm cộng trăng Gauss có trị trung bình bằng không và phương sai  $\sigma^2$ . Kênh này được sử dụng với hai điều kiện khác nhau. Trong một trường hợp, lối ra được sử dụng trực tiếp không lượng tử hoá (quyết định mềm), và trong trường hợp kia thì quyết định tối ưu được thực hiện theo từng mức lối vào (quyết định cứng). Hãy vẽ đồ thị của dung lượng kênh như một hàm của  $A/\sigma$  cho mỗi trường hợp.

#### Lời giải

Trường hợp quyết định mềm tương tự như bài tập minh họa 8.3. Đối với trường hợp quyết định cứng, xác suất chéo của kênh đối xứng nhị phân có tác động của tạp âm là  $Q(A/\sigma)$ , và do vậy dung lượng kênh được cho bởi

$$C_H = 1 - H_b\left(Q\left(\frac{A}{\sigma}\right)\right)$$

Cả  $C_H$  lẫn  $C_S$  theo  $A/\sigma$  được thể hiện trên hình 8.8. Như mong đợi, giải mã

quyết định mềm thì tốt hơn giải mã quyết định cứng tại mọi giá trị của  $A/\sigma$ .

MATLAB script cho bài tập này được cho dưới đây.

### m-file

% Chương trình MatLab cho bài tập minh họa 8.4.

echo on

a\_db=[-13:0.5:13];

a=10.^ (a\_db/10);

c\_hard=1-entropy2(Q(a));

for i=1:53

f(i)=quad('il3\_8fun',a(i)-5,a(i)+5,1e-3,[],a(i));

g(i)=quad('il3\_8fun',-a(i)-5,-a(i)+5,1e-3,[],-a(i));

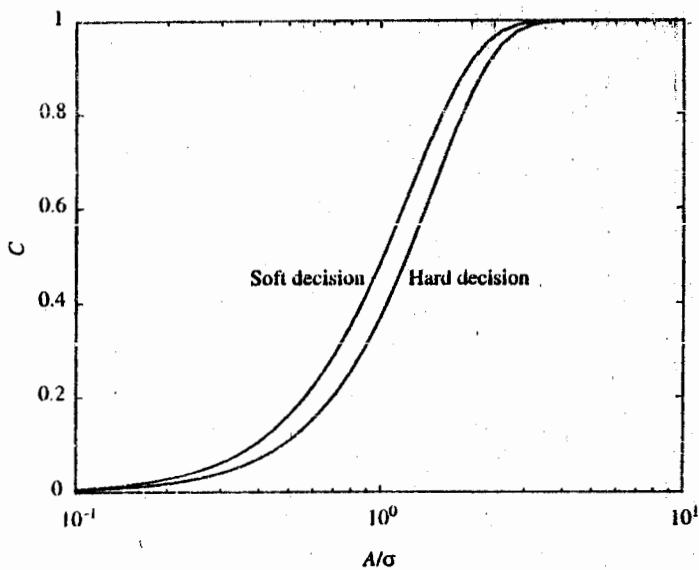
c\_soft(i)=0.5\*f(i)+0.5\*g(i);

echo off ;

end

echo on ;

pause % Bấm phím bất kỳ để xem các đường cong dung lượng kênh  
semilogx(a,c\_soft,a,c\_hard)



Hình 8.8 Các đồ thị của  $C_H$  và  $C_S$  theo  $\text{SNR} = A/\sigma$

### BÀI TẬP MINH HỌA

**Bài tập minh họa 8.5 [Dung lượng kênh theo dải tần và SNR]** Dung lượng của một kênh AWGN có băng thông hạn chế với ràng buộc công suất đầu vào  $P$  và

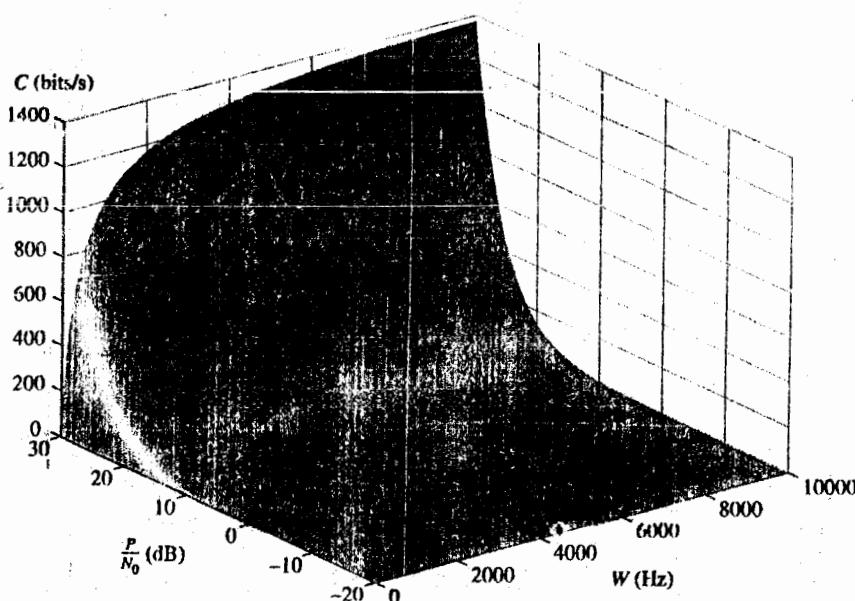
độ rộng băng  $W$  được cho bởi

$$C = W \log_2 \left( 1 + \frac{P}{N_0 W} \right)$$

Hãy vẽ đồ thị của dung lượng kênh như một hàm của cả  $W$  và  $P/N_0$ .

### Lời giải

Đồ thị cần vẽ được chỉ ra trên hình 8.9. Hãy lưu ý rằng với  $P/N_0$  không đổi, đồ thị này suy biến thành đường cong đã được thể hiện trên hình 8.6. Với độ rộng băng không đổi, đồ thị dung lượng kênh như một hàm của  $P/N_0$  giống như đường cong đã được thể hiện trên hình 8.5.



Hình 8.9 Dung lượng của kênh như là một hàm của độ rộng băng thông và SNR trong một kênh AWGN

MATLAB script cho bài tập này được cho dưới đây.

### m-file

% Chương trình MatLab cho bài tập minh họa 8.5.

echo off

```
w=[1:5:20,25:20:100,130:50:300,400:100:1000,1250:250:5000,5500:500:10000];
pn0_db=[-20:1:30];
pn0=10.^((pn0_db/10));
```

```

for i=1:45
    for j=1:51
        c(i,j)=w(i)*log2(1+pn0(j)/w(i));
    end
end
echo on
pause % Bấm phím bất kỳ để xem đường cong của C theo W và P/N0.
k=[0.9,0.8,0.5,0.6];
s=[-70,35];
surf(w,pn0_db,c',s,k)
title('Dung lượng kênh theo dải tần và SNR')

```

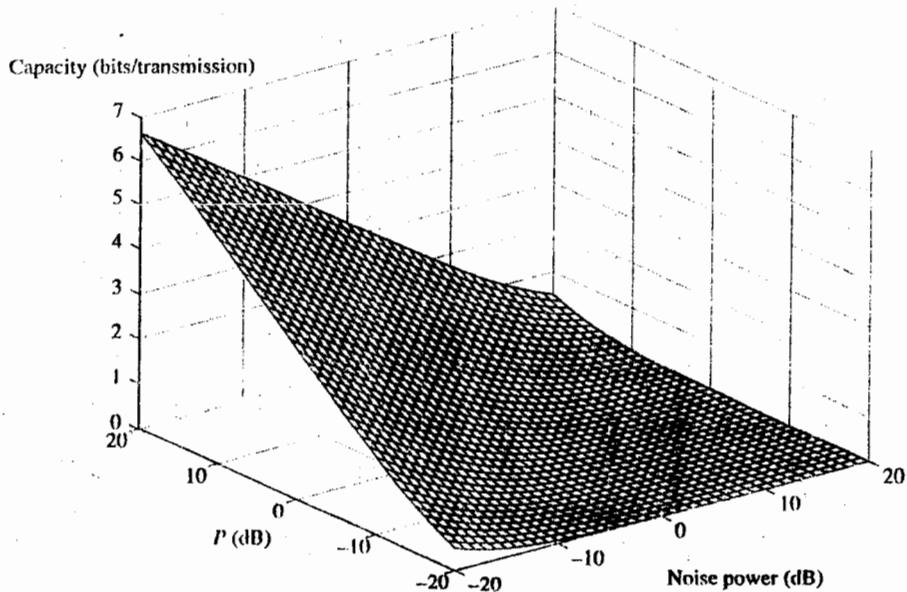
### BÀI TẬP MINH HỌA

#### Bài tập minh họa 8.6 [Dung lượng của kênh AWGN thời gian rời rạc]

Hãy vẽ đồ thị dung lượng của kênh AWGN thời gian rời rạc như một hàm của công suất lối vào và phương sai của tạp âm.

#### Lời giải

Đồ thị mong muốn được cho trên hình 8.10.



Hình 8.10 Dung lượng của kênh AWGN thời gian rời rạc  
như một hàm của công suất tín hiệu ( $P$ ) và công suất của tạp âm ( $\sigma^2$ )

MATLAB *script* cho bài tập này được cho dưới đây.

### m-file

```
% Chương trình MatLab cho bài tập minh họa 8.6.
```

```
echo on
```

```
p_db=[-20:1:20];
```

```
np_db=p_db;
```

```
p=10.^ (p_db/10);
```

```
np=p;
```

```
for i=1:41
```

```
    for j=1:41
```

```
        c(i,j)=0.5*log2(1+p(i)/np(j));
```

```
        echo off ;
```

```
    end
```

```
end
```

```
echo on ;
```

```
pause % Bấm phím bất kỳ để quan sát đồ thị
```

```
surf1(np_db,p_db,c)
```

## 8.3 MÃ HOÁ KÊNH

Việc truyền tin qua các kênh có tạp nhiễu thì bị các lõi. Để giảm tác động của các lõi và truyền tin tin cậy, cần phải truyền các dãy càng khác nhau càng tốt, sao cho tạp nhiễu trên kênh sẽ không làm cho một dãy này biến đổi thành một dãy khác. Điều này có nghĩa là một độ dư nào đó phải được đưa thêm vào để tăng độ tin cậy của quá trình truyền tin. Việc đưa thêm vào độ dư dẫn đến việc phải truyền các bít thêm vào và dẫn đến một sự giảm tốc độ truyền dẫn.

Các sơ đồ mã hóa kênh nói chung có thể phân chia được thành hai lớp: các mã khối và các mã chập. Trong mã hóa khối, các dãy lõi ra nguồn tin nhị phân có chiều dài  $k$  được ánh xạ thành các dãy lõi vào kênh nhị phân có chiều dài  $n$ ; do đó, tỉ lệ của mã là  $k/n$  bít/một lần truyền. Một mã như thế được gọi là một mã khối  $(n,k)$  và gồm  $2^k$  từ mã có độ dài  $n$ , thường được ký hiệu là  $c_1, c_2, \dots, c_{2^k}$ . Việc ánh xạ các lõi ra của nguồn thông tin thành các lõi vào kênh được thực hiện một cách độc lập, và lõi ra của bộ mã hóa chỉ phụ thuộc dãy lõi vào hiện thời có chiều dài  $k$  và không phụ thuộc vào các dãy lõi vào trước đó. Trong mã hóa chập, các lõi

ra nguồn tin có chiều dài  $k_0$  được ánh xạ thành  $n_0$  bít lối vào kênh, song các bít lối vào kênh phụ thuộc không chỉ vào  $k_0$  bít lối ra nguồn ngay lúc này mà còn phụ thuộc cả vào  $(L-1)k_0$  bít lối vào cuối cùng của bộ mã hoá.

Một trong những mã khối đơn giản nhất là *mã lắp đơn giản*, trong đó, có hai bản tin được truyền đi qua một kênh nhị phân đối xứng, nhưng thay vì truyền đi một bit 0 và một bít 1 cho 2 bản tin thì 2 dãy sẽ được truyền đi, một dãy bao gồm toàn bit 0 và một dãy bao gồm toàn bit 1. Chiều dài của 2 dãy được chọn là một số lẻ  $n$  nào đó. Quá trình mã hoá được thể hiện như sau:

$$0 \rightarrow \overbrace{00\dots00}^{n \text{ lối}} \quad (8.3.1)$$

$$1 \rightarrow \overbrace{11\dots11}^{n \text{ lối}} \quad (8.3.2)$$

Việc giải mã chỉ là một phép giải mã biểu quyết theo đa số; nghĩa là, nếu đa số các *symbol* thu được là 1 thì bộ giải mã sẽ quyết định là một bít 1; còn nếu đa số là 0 thì bộ giải mã sẽ quyết định là một bít 0.

Một lỗi xảy ra nếu có ít nhất  $(n+1)/2$  các *symbol* đã truyền bị nhận lỗi. Do kênh là một kênh đối xứng nhị phân với xác suất chéo  $\varepsilon$ , xác suất lỗi có thể biểu diễn được theo

$$p_e = \sum_{k=(n+1)/2}^n \binom{n}{k} \varepsilon^k (1-\varepsilon)^{n-k} \quad (8.3.3)$$

Ví dụ, với  $n = 5$  và  $\varepsilon = 0.001$ , ta có

$$p_e = \sum_{k=3}^5 \binom{5}{k} 0.001^k (0.999)^{5-k} \approx 10^{-9} \quad (8.3.4)$$

Điều này có nghĩa là bằng cách sử dụng kênh 5 lần thay cho chỉ sử dụng một lần, chúng ta đã có thể giảm xác suất lỗi từ 0.001 xuống còn  $10^{-9}$ . Tất nhiên là phải trả giá cho chất lượng tin cậy hơn này; giá đó là một sự giảm tốc độ truyền và độ phức tạp của hệ thống thì tăng lên. Tốc độ truyền tin đã giảm từ một bản tin nhị phân trong một lần sử dụng kênh xuống thành một bản tin nhị phân trong 5 lần sử dụng kênh. Độ phức tạp của hệ thống tăng là do bây giờ, chúng ta phải sử

dụng một bộ mã hoá (mà nó có cấu trúc rất đơn giản) và một bộ giải mã mà nó thực hiện giải mã biểu quyết theo đa số. Việc truyền còn tin cậy hơn nữa trong bài toán này nếu chúng ta tăng  $n$ . Ví dụ, với  $n = 9$ , chúng ta có

$$p_e = \sum_{k=5}^9 \binom{9}{k} 0.001^k (0.999)^{9-k} \approx 10^{-15} \quad (8.3.5)$$

Từ trên, có vẻ như là nếu ta muốn giảm xác suất lỗi xuống bằng không thì ta phải tăng  $n$  đến vô cùng, và do đó, làm giảm tốc độ truyền tin xuống bằng không. Tuy nhiên, điều này không hoàn toàn đúng và Shannon đã chỉ ra rằng có thể đạt được truyền tin tin cậy (nghĩa là  $p_e \rightarrow 0$ ) một cách tiệm cận bằng cách giữ cho tốc độ truyền tin thấp hơn dung lượng kênh, mà trong trường hợp trên là

$$C = 1 - H_b(0.001) = 1 - 0.0114 = 0.9886 \text{ bit/lần truyền} \quad (8.3.6)$$

Tuy vậy, điều này đạt được bằng cách áp dụng các sơ đồ mã hoá và giải mã phức tạp hơn nhiều so với mã lặp đơn giản.

### BÀI TẬP MINH HỌA

**Bài tập minh họa 8.7 [Xác suất lỗi khi sử dụng các mã lặp đơn giản]** Giả thiết rằng  $\varepsilon = 0.3$  trong một kênh đối xứng nhị phân, hãy vẽ đồ thị của xác suất lỗi  $p_e$  như một hàm của chiều dài khối  $n$ .

#### Lời giải

Chúng ta sẽ tìm các giá trị của  $p_e$  đối với các giá trị của  $n$  từ 1 tới 61. Xác suất lỗi được cho bởi

$$p_e = \sum_{k=(n+1)/2}^n \binom{n}{k} 0.3^k 0.7^{n-k}$$

và đồ thị có được được thể hiện trên hình 8.11.

MATLAB script cho bài tập này được cho dưới đây.

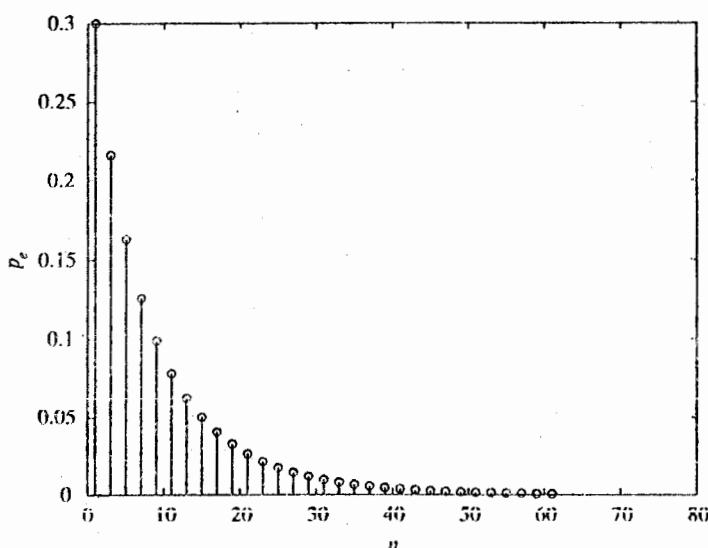
#### m-file

```
% Chương trình MatLab cho bài tập minh họa 8.7, chương 8.
echo on
ep=0.3;
for i=1:2:61
```

```

p(i)=0;
for j=(i+1)/2:i
    p(i)=p(i)+prod(1:i)/(prod(1:j)*prod(1:(i-j)))*ep^j*(1-ep)^(i-j);
    echo off ;
end
end
echo on ;
pause % Bấm phím bất kỳ để quan sát đồ thị
stem((1:2:61),p(1:2:61))
xlabel('n')
ylabel('pe')
title('Đồ thị của xác suất lỗi theo n cho mã lặp đơn giản')

```



Hình 8.11 Xác suất lỗi của một mã lặp đơn giản đối với  $\varepsilon = 0.3$  và  $n = 1, 3, \dots, 61$

### 8.3.1 MÃ KHỐI TUYẾN TÍNH

Các mã khối tuyến tính là lớp các mã khối quan trọng nhất và được ứng dụng nhiều nhất. Một mã khối là tuyến tính nếu bất cứ tổ hợp tuyến tính nào của hai từ mã cũng là một từ mã. Trong trường hợp nhị phân, điều này có nghĩa là tổng của hai từ mã bất kỳ cũng là một từ mã. Trong các mã khối tuyến tính, các từ mã tạo thành một không gian con  $k$ -chiều của một không gian  $n$ -chiều. Các mã khối tuyến tính được mô tả thông qua ma trận sinh  $\mathbf{G}$  của chúng, là ma trận nhị phân  $k \times n$  sao cho mỗi từ mã  $c$  có thể viết được dưới dạng

$$\mathbf{c} = \mathbf{u}\mathbf{G} \quad (8.3.7)$$

trong đó,  $\mathbf{u}$  là dãy dữ liệu nhị phân chiều dài  $k$  (lõi vào bộ mã hoá). Hiển nhiên, dãy toàn các bit 0 có độ dài  $n$  luôn luôn là một từ mã của một mã khối tuyến tính  $(n,k)$ .

Một tham số quan trọng trong một mã khối tuyến tính, xác định các khả năng sửa lỗi của nó, là khoảng cách (Hamming) tối thiểu của mã, được định nghĩa là khoảng cách Hamming nhỏ nhất giữa hai từ mã khác nhau bất kỳ. Khoảng cách tối thiểu của một mã được ký hiệu là  $d_{\min}$ , và ta có

$$d_{\min} = \min_{i \neq j} d_H(\mathbf{c}_i, \mathbf{c}_j) \quad (8.3.8)$$

Đối với các mã tuyến tính, khoảng cách tối thiểu bằng trọng số tối thiểu của mã, được xác định bởi

$$w_{\min} = \min_{c_i \neq 0} w(\mathbf{c}_i) \quad (8.3.9)$$

đó là số nhỏ nhất các bít 1 trong từ mã khác 0 bất kỳ.

### BÀI TẬP MINH HỌA

**Bài tập minh họa 8.8 [Các mã khối tuyến tính]** Ma trận sinh đối với một mã khối tuyến tính  $(10,4)$  được cho bởi

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Hãy xác định tất cả các từ mã và trọng số tối thiểu của mã này.

### Lời giải

Để có thể nhận được tất cả các từ mã, chúng ta phải sử dụng tất cả các dãy thông tin có chiều dài 4 và tìm các dãy được mã tương ứng. Do có cả thảy 16 dãy nhị phân có chiều dài 4 nên sẽ có 16 từ mã. Cho  $\mathbf{U}$  là ký hiệu một ma trận  $2^k \times k$  có các hàng là tất cả các dãy thông tin có thể có với chiều dài  $k$ , bắt đầu từ dãy gồm toàn 0 và kết thúc với dãy toàn 1. Các hàng được chọn sao cho giá trị thập phân của mỗi hàng thì nhỏ hơn giá trị thập phân của tất cả các hàng nằm dưới nó. Đối với trường hợp  $k=4$ , ma trận  $\mathbf{U}$  được cho bởi

$$U = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Ta có

$$C = UG \quad (8.3.10)$$

trong đó,  $C$  là ma trận các từ mã, trong trường hợp này nó là một ma trận  $16 \times 10$  với các hàng là các từ mã. Ma trận các từ mã được cho bởi

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Việc xem xét kỹ các từ mã cho thấy rằng khoảng cách tối thiểu của mã là  $d_{\min} = 2$ .

MATLAB script cho bài tập này được cho dưới đây.

**m-file**

```
% Chương trình MatLab cho bài tập minh họa 8, chương 8.
% Tao ma trận U, là tất cả các dãy thông tin
k=4;
for i=1:2^k
    for j=k:-1:1
        if rem(i-1,2^(-j+k+1))>=2^(-j+k)
            u(i,j)=1;
        else
            u(i,j)=0;
        end
        echo off ;
    end
end
echo on ;
% Định nghĩa G, ma trận sinh.
g=[1 0 0 1 1 1 0 1 1 1;
   1 1 1 0 0 0 1 1 1 0;
   0 1 1 0 1 1 0 1 0 1;
   1 1 0 1 1 1 1 0 0 1];
% Tao các từ mã
c=rem(u*g,2);
% Tìm khoảng cách tối thiểu.
w_min=min(sum((c(2:2^k,:))'));
```

Một mã khởi tuyến tính có dạng hệ thống nếu ma trận sinh của nó có dạng như sau:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & \dots & 0 & p_{1,1} & p_{1,2} & \dots & p_{1,n-k} \\ 0 & 1 & \dots & 0 & p_{2,1} & p_{2,2} & \dots & p_{2,n-k} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & p_{k,1} & p_{k,2} & \dots & p_{k,n-k} \end{bmatrix} \quad (8.3.11)$$

hoặc

$$\mathbf{G} = [\mathbf{I}_k \mid \mathbf{P}] \quad (8.3.12)$$

trong đó,  $\mathbf{I}_k$  là ma trận đơn vị  $k \times k$  và  $\mathbf{P}$  là ma trận  $k \times (n-k)$ . Trong một mã hệ thống,  $k$  symbol nhị phân đầu tiên trong một từ mã là các bit thông tin, còn  $(n-k)$  symbol nhị phân còn lại là các symbol kiểm tra (*parity-check symbols*).

*Ma trận kiểm tra* của một mã là một ma trận nhị phân  $\mathbf{H}$   $(n-k) \times n$  bất kỳ sao

cho với mọi từ mã  $\mathbf{c}$ , ta có

$$\mathbf{c}\mathbf{H}^t = \mathbf{0} \quad (8.3.13)$$

Hiển nhiên, chúng ta sẽ có

$$\mathbf{G}\mathbf{H}^t = \mathbf{0} \quad (8.3.14)$$

và nếu  $\mathbf{G}$  ở dạng hệ thống thì

$$\mathbf{H} = [\mathbf{P}' | \mathbf{I}_k] \quad (8.3.15)$$

### Các mã Hamming

Các mã Hamming là các mã khối tuyến tính  $(2^m-1, 2^m-m-1)$  với khoảng cách tối thiểu bằng 3 và ma trận kiểm tra rất đơn giản. Ma trận kiểm tra, là một ma trận  $m \times (2^m-1)$ , có tất cả các dãy nhị phân chiều dài  $m$  (ngoại trừ dãy toàn 0) làm các cột. Ví dụ, với  $m = 3$ , ta có một mã  $(7,4)$  mà ma trận kiểm tra của nó ở dạng hệ thống là

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (8.3.16)$$

Từ đó chúng ta có

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (8.3.17)$$

### BÀI TẬP MINH HỌA

**Bài tập minh họa 8.9 [Các mã Hamming]** Hãy tìm tất cả các từ mã của mã Hamming  $(15,11)$ , và hãy chứng tỏ rằng khoảng cách tối thiểu của nó là 3.

#### Lời giải

Ở đây

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (8.3.18)$$

và do đó

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Có tất cả  $2^{11} = 2048$  từ mã, mỗi từ mã có chiều dài 15. Tỉ lệ mã của mã này là  $11/15 = 0.733$ . Để kiểm tra lại chiều dài tối thiểu của mã, ta sử dụng một MATLAB script tương tự như script đã được sử dụng trong bài tập minh họa 8.8. MATLAB script này được cho dưới đây và nó cho kết quả  $d_{\min} = 3$ .

### m-file

```
% Chương trình MatLab cho bài tập minh họa 8.9.
```

```
echo on
k=11;
for i=1:2^k
    for j=k:-1:1
        if rem(i-1,2^(-j+k+1))>=2^(-j+k)
            u(i,j)=1;
        else
            u(i,j)=0;
        end
        echo off ;
    end
end
echo on;

g=[1 0 0 0 0 0 0 0 0 0 1 1 0 0;
   0 1 0 0 0 0 0 0 0 0 1 1 0;
   0 0 1 0 0 0 0 0 0 0 0 1 1;
   0 0 0 1 0 0 0 0 0 0 1 0 1 0;
   0 0 0 0 1 0 0 0 0 0 1 0 0 1;
   0 0 0 0 0 1 0 0 0 0 1 0 1;
   0 0 0 0 0 0 1 0 0 0 1 1 1 0;
```

```

0 0 0 0 0 0 0 1 0 0 0 0 1 1 1;
0 0 0 0 0 0 0 0 1 0 0 1 0 1 1;
0 0 0 0 0 0 0 0 0 1 0 1 1 0 1;
0 0 0 0 0 0 0 0 0 1 1 1 1 1];

```

```

c=rem(u*g,2);
w_min=min(sum((c(2:2^k,:))'));

```

### Chất lượng của các mã khôi tuyển tính

Các mã khôi tuyển tính có thể được giải mã bằng cách sử dụng hoặc giải mã quyết định cứng hoặc giải mã quyết định mềm. Trong một sơ đồ giải mã quyết định cứng, thoát tiên việc quyết định theo từng bit được thực hiện trên các phân tử của từ mã, rồi sau đó việc giải mã được thực hiện với tiêu chí khoảng cách Hamming tối thiểu. Chất lượng của sơ đồ giải mã này tùy thuộc vào cấu trúc khoảng cách của mã, song một chyện trên chặng (đặc biệt là với các giá trị SNR lớn) có thể nhận được theo khoảng cách tối thiểu của mã.

Xác suất lỗi (bản tin) của một mã khôi tuyển tính với khoảng cách tối thiểu  $d_{\min}$ , trong giải mã quyết định cứng, được chặn trên bởi

$$p_e \leq (M - 1)[4p(1 - p)]^{d_{\min}/2} \quad (8.3.19)$$

trong đó  $p$  là xác suất lỗi của kênh nhị phân (xác suất lỗi trong giải điều chế) và  $M$  là số các từ mã ( $M = 2^k$ ).

Trong giải mã quyết định mềm, tín hiệu thu được sẽ được ánh xạ thành từ mã mà tín hiệu tương ứng của nó có khoảng cách Euclid tới tín hiệu thu được là nhỏ nhất. Xác suất lỗi bản tin trong trường hợp này được chặn trên bởi

$$p_e \leq (M - 1)Q\left(\frac{d^E}{\sqrt{2N_0}}\right) \quad (8.3.20)$$

trong đó,  $M = 2^k$  là số các từ mã,  $N_0$  là mật độ phổ công suất một phía của tạp âm, và  $d^E$  là khoảng cách Euclid tối thiểu của mã, được cho bởi

$$d^E = \begin{cases} \sqrt{2d_{\min}E} & \text{truyền bằng các tín hiệu trực giao} \\ \sqrt{4d_{\min}E} & \text{truyền bằng các tín hiệu đối cực} \end{cases} \quad (8.3.21)$$

mà điều này dẫn đến

$$p_e \leq \begin{cases} (M-1)Q\left(\sqrt{\frac{d_{min}E}{N_0}}\right) & \text{truyền bằng các tín hiệu trực giao} \\ (M-1)Q\left(\sqrt{\frac{2d_{min}E}{N_0}}\right) & \text{truyền bằng các tín hiệu đối cực} \end{cases} \quad (8.3.22)$$

Trong các bất đẳng thức này,  $d_{min}$  là khoảng cách Hamming tối thiểu của mã còn  $E$  là năng lượng tính trên mỗi phần tử của các từ mã. Do mỗi từ mã có  $n$  phần tử nên năng lượng tính trên mỗi từ mã là  $nE$ , và do mỗi từ mã mang  $k$  bit thông tin nên năng lượng tính trên mỗi bit  $E_b$  được cho bởi

$$E_b = \frac{nE}{k} = \frac{E}{R_c} \quad (8.3.23)$$

trong đó  $R_c = k/n$  là tỉ lệ mã. Do vậy, các quan hệ trên có thể viết được theo

$$p_e \leq \begin{cases} (M-1)Q\left(\sqrt{\frac{d_{min}R_c E_b}{N_0}}\right) & \text{truyền bằng các tín hiệu trực giao} \\ (M-1)Q\left(\sqrt{\frac{2d_{min}R_c E_b}{N_0}}\right) & \text{truyền bằng các tín hiệu đối cực} \end{cases} \quad (8.3.24)$$

Các giới hạn chặn thu được thường chỉ khả dụng đối với các giá trị lớn của  $\gamma_b = E_b/N_0$ . Đối với các giá trị  $\gamma_b$  nhỏ thì các giới hạn chặn này trở nên rất lỏng và thậm chí có thể vượt quá 1.

### BÀI TẬP MINH HỌA

**Bài tập minh họa 8.10 [Chất lượng giải mã quyết định cứng]** Giả thiết rằng mã Hamming (15, 11) được sử dụng với việc truyền tin được thực hiện bằng các tín hiệu đối cực (*antipodal*) và giải mã quyết định cứng, hãy vẽ đồ thị xác suất lỗi bản tin như một hàm của  $\gamma_b = E_b/N_0$ .

### Lời giải

Do truyền tin bằng tín hiệu đối cực, xác suất lỗi của kênh nhị phân được cho bởi

$$p = Q\left(\sqrt{\frac{2E}{N_0}}\right) \quad (8.3.25)$$

trong đó,  $E$  là năng lượng trên một phần tử của mã (năng lượng trên một chiêu) và được tính từ  $E_b$  theo

$$E = E_b R_c \quad (8.3.26)$$

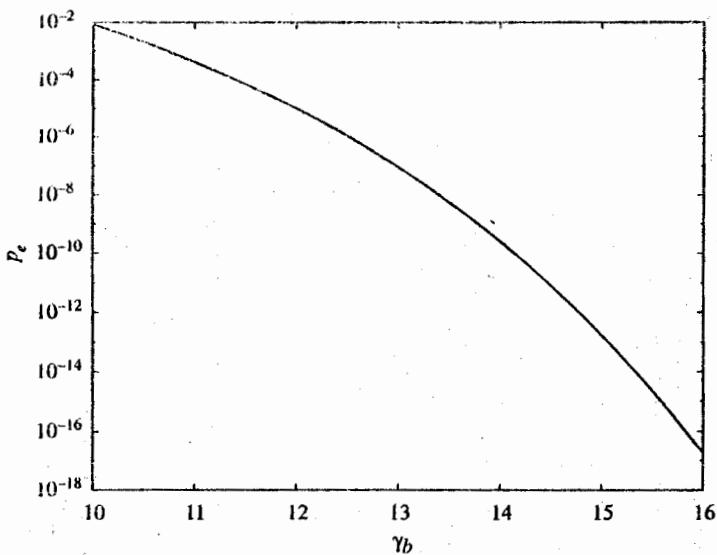
Do đó,

$$P = Q\left(\sqrt{\frac{2R_c E_b}{N_0}}\right) \quad (8.3.27)$$

trong đó  $R_c = k/n = 11/15 = 0.73333$ . Do khoảng cách tối thiểu của các mã Hamming là 3, chúng ta có

$$\begin{aligned} p_e &\leq (2^{11} - 1)[4p(1-p)]^{d_{min}/2} \\ &= 2047 \left[ 4Q\left(\frac{\sqrt{1.466E_b}}{N_0}\right) \left(1 - Q\left(\frac{\sqrt{1.466E_b}}{N_0}\right)\right) \right]^{1.5} \end{aligned} \quad (8.3.28)$$

Đồ thị tính được được vẽ ở hình 8.12.



Hình 8.12 Xác suất lỗi như một hàm của  $\gamma_b$  đối với một mã Hamming với giải mã quyết định cứng và sử dụng tín hiệu đổi cực.

Hàm MATLAB để tính giới hạn chặn của xác suất lỗi bản tin của một mã khối tuyến tính khi sử dụng giải mã quyết định cứng và truyền tin bằng tín hiệu đổi cực được cho dưới đây.

**m-file**

```

function [p_err,gamma_db]=p_e_hd_a(gamma_db_l, gamma_db_h,k,n,d_min)
% p_e_hd_a.m      Matlab function for computing error probability in
%                  hard decision decoding of a linear block code
%                  when antipodal signaling is used.
% [p_err,gamma_db]=p_e_hd_a(gamma_db_l, gamma_db_h,k,n,d_min)
% gamma_db_l=lower E_b/N_0
% gamma_db_h=higher E_b/N_0
% k=number of information bits in the code
% n=code block length
% d_min=minimum distance of the code

gamma_db=[gamma_db_l:(gamma_db_h-gamma_db_l)/20:gamma_db_h];
gamma_b=10.^{gamma_db/10};
R_c=k/n;
p_b=q(sqrt(2.*R_c.*gamma_b));
p_err=(2^{k-1}).*(4*p_b.*(1-p_b)).^{(d_min/2)};

```

Trong MATLAB *script* được cho dưới đây, hàm MATLAB trên đây được áp dụng để vẽ xác suất lỗi theo  $\gamma_b$ .

**m-file**

```

% Chương trình MatLab cho bài tập minh họa 10, chương 8.
[p_err_ha,gamma_b]=p_e_hd_a(10,16,11,15,3);
semilogy(gamma_b,p_err_ha)

```

**BÀI TẬP MINH HỌA**

**Bài tập minh họa 8.11 [Giải mã quyết định cứng]** Nếu mã Hamming (15,11) được sử dụng với một sơ đồ điều chế nhị phân trực giao thay cho sơ đồ tín hiệu đối cực, hãy vẽ đồ thị xác suất lỗi bản tin theo  $\gamma_b = E_b/N_0$ .

**Lời giải**

Bài tập này tương tự với bài tập minh họa 8.10, chỉ khác ở chỗ xác suất chéo của kênh đối xứng nhị phân tương đương (sau khi giải mã quyết định cứng) được cho bởi

$$p = Q\left(\sqrt{\frac{E}{N_0}}\right) \quad (8.3.29)$$

## Sử dụng quan hệ

$$E = E_b R_c \quad (8.3.30)$$

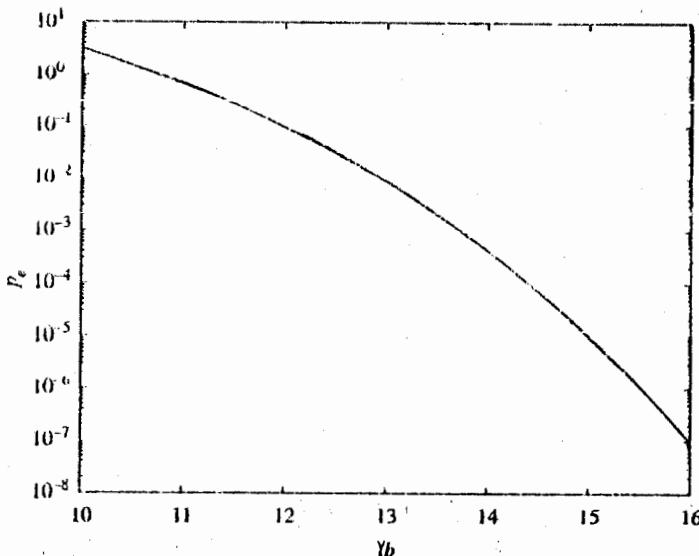
chúng ta được

$$p = Q\left(\sqrt{\frac{R_c E_b}{N_0}}\right) \quad (8.3.31)$$

và cuối cùng là

$$\begin{aligned} p_e &\leq (2^{11} - 1)[4p(1-p)]^{d_{min}/2} \\ &= 2047 \left[ 4Q\left(\sqrt{\frac{0.733 E_b}{N_0}}\right) \left(1 - Q\left(\sqrt{\frac{0.733 E_b}{N_0}}\right)\right) \right]^{1.5} \end{aligned} \quad (8.3.32)$$

Đồ thị  $p_e$  theo  $E_b/N_0$  được thể hiện trên hình 8.13.

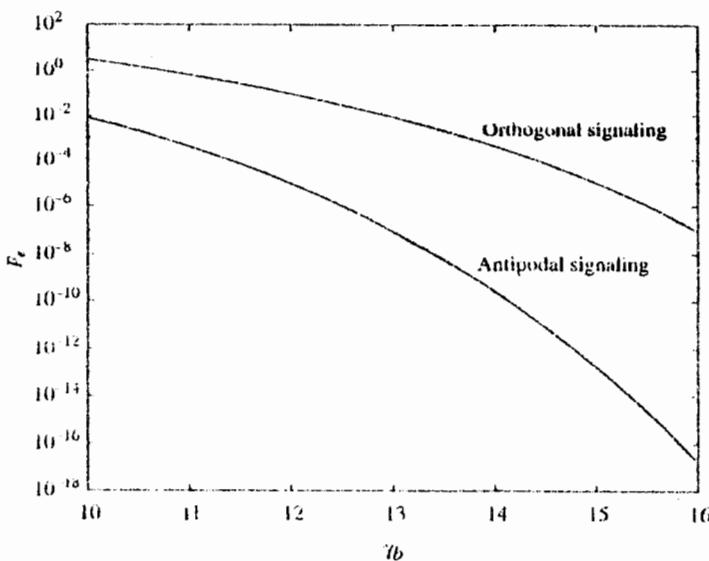


Hình 8.13 Xác suất lỗi theo  $\gamma_b$  của một mã (15, 11) với giải mã quyết định cứng và truyền tin bằng các tín hiệu trực giao.

File MATLAB cho bài tập này được cho dưới đây.

```
% Chương trình MatLab cho bài tập minh họa 11, chương 8.
echo on
gamma_b_db=[-4:1:14];
gamma_b=10.^ (gamma_b_db/10);
qq=q(sqrt(0.733.*gamma_b));
p_err=2047*qq.^2.* (3-2.*qq);
pause % Bấm một phím bất kỳ để quan sát đường cong của p_err theo gamma_b
loglog(gamma_b,p_err)
```

Như ta thấy từ hình 8.13, đối với các giá trị thấp của  $\gamma_b$  thì giới hạn chặn là quá lỏng. Thực vậy, đối với những giá trị này của  $\gamma_b$ , giới hạn chặn đó của xác suất lỗi lớn hơn cả 1. Việc vẽ đồ thị 2 giới hạn chặn của xác suất lỗi với việc truyền tin bằng tín hiệu đối cực và tín hiệu trực giao trên cùng một hình vẽ cũng cho chúng ta thấy rõ thêm. Điều này được thực hiện trên hình 8.14. Chất lượng hơn hẳn của việc truyền tin bằng tín hiệu đối cực so với truyền tin bằng tín hiệu trực giao thấy ngay được nhờ so sánh hai đồ thị này.



Hình 8.14 So sánh truyền tin bằng tín hiệu đối cực và bằng tín hiệu trực giao

Hàm MATLAB dùng để tính xác suất lỗi bản tin trong trường hợp giải mã quyết định cứng với truyền tin bằng tín hiệu trực giao được cho dưới đây.

### m-file

```
function [p_err,gamma_db]=p_e_hd_o(gamma_db_l, gamma_db_h,k,n,d_min)
% p_e_hd_o.m      Matlab function for computing error probability in
%                  hard decision decoding of a linear block code
%                  when orthogonal signaling is used.
% {p_err,gamma_db}=p_e_hd_o(gamma_db_l, gamma_db_h,k,n,d_min)
% gamma_db_l=lower E_b/N_0
% gamma_db_h=higher E_b/N_0
% k=number of information bits in the code
% n=code block length
```

%  $d_{\min} = \text{minimum distance of the code}$

```
gamma_db=[gamma_db_l:(gamma_db_h-gamma_db_l)/20:gamma_db_h];
gamma_b=10.^((gamma_db/10));
R_c=k/n;
p_b=q(sqrt(R_c.*gamma_b));
p_err=(2^k-1).*(4*p_b.*((1-p_b)).^(d_min/2);
```

## BÀI TẬP MINH HỌA

**Bài tập minh họa 8.12 [Giải mã quyết định mềm]** Hãy giải bài tập minh họa 8.11 khi giải mã quyết định mềm được sử dụng thay cho quyết định cứng.

### Lời giải

Trong trường hợp này, ta phải sử dụng biểu thức (8.3.24) để tìm một giới hạn chặn trên đối với xác suất lỗi. Trong bài tập đang xét,  $d_{\min} = 3$ ,  $R_c = 11/15$  và  $M = 2^{11}-1 = 2047$ . Do đó, chúng ta có

$$p_e \leq \begin{cases} (M-1)Q\left(\sqrt{\frac{d_{\min} R_c E_b}{N_0}}\right) & \text{truyền bằng tín hiệu trực giao} \\ (M-1)Q\left(\sqrt{\frac{2d_{\min} R_c E_b}{N_0}}\right) & \text{truyền bằng tín hiệu đối cực} \end{cases}$$

$$\leq \begin{cases} 2047Q\left(\sqrt{\frac{11 E_b}{5 N_0}}\right) & \text{truyền bằng tín hiệu trực giao} \\ 2047Q\left(\sqrt{\frac{22 E_b}{5 N_0}}\right) & \text{truyền bằng tín hiệu đối cực} \end{cases}$$

Các đồ thị tương ứng được vẽ trên hình 8.15. Từ các đồ thị này, chất lượng hơn hẳn của việc truyền tin bằng tín hiệu đối cực là rõ ràng.

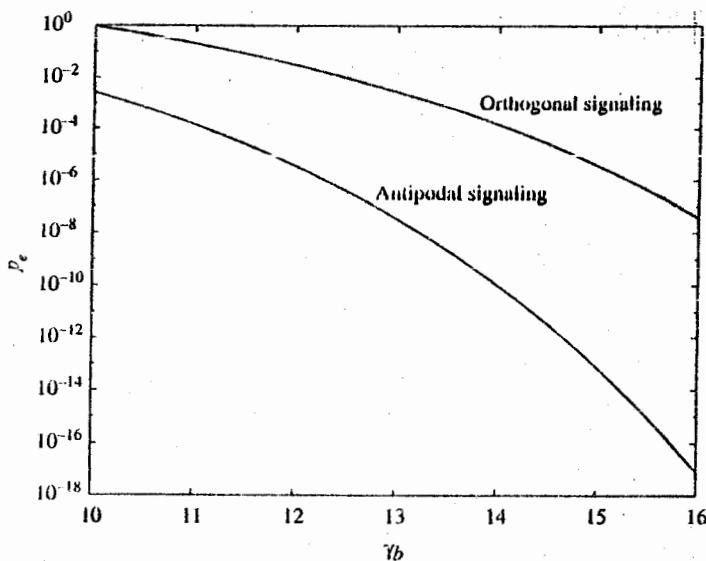
Hai hàm MATLAB - một để tính xác suất lỗi đối với truyền tin bằng tín hiệu đối cực và một dùng để tính xác suất lỗi đối với truyền tin bằng tín hiệu trực giao, khi giải mã quyết định mềm được áp dụng - được cho dưới đây.

### m-file

```
function [p_err,gamma_db]=p_e_sd_a(gamma_db_l,gamma_db_h,k,n,d_min)
% p_e_sd_a.m      Matlab function for computing error probability in soft decision
% decoding of a linear block code when antipodal signaling is used.
```

```
% [p_err,gamma_db]=p_e_sd_a(gamma_db_l, gamma_db_h,k,n,d_min)
% gamma_db_l=lower E_b/N_0
% gamma_db_h=higher E_b/N_0
% k=number of information bits in the code
% n=code block length
% d_min=minimum distance of the code
```

```
gamma_db=[gamma_db_l:(gamma_db_h-gamma_db_l)/20:gamma_db_h];
gamma_b=10.^{gamma_db/10};
R_c=k/n;
p_err=(2^{k-1}.*q(sqrt(2.*d_min.*R_c.*gamma_b)));
```



Hình 8.15 Xác suất lỗi bản tin theo  $\gamma_b$  đổi với giải mã quyết định mềm

### m-file

```
function [p_err,gamma_db]=p_e_sd_o(gamma_db_l, gamma_db_h,k,n,d_min)
% p_e_sd_o.m      Matlab function for computing error probability in
%                  soft decision decoding of a linear block code
%                  when orthogonal signaling is used.
% [p_err,gamma_db]=p_e_sd_o(gamma_db_l, gamma_db_h,k,n,d_min)
% gamma_db_l=lower E_b/N_0
% gamma_db_h=higher E_b/N_0
% k=number of information bits in the code
% n=code block length
% d_min=minimum distance of the code
```

```
gamma_db=[gamma_db_l:(gamma_db_h-gamma_db_l)/20:gamma_db_h];
gamma_b=10.^{gamma_db/10};
R_c=k/n;
p_err=(2^{k-1}).*q(sqrt(d_min.*R_c.*gamma_b));
```

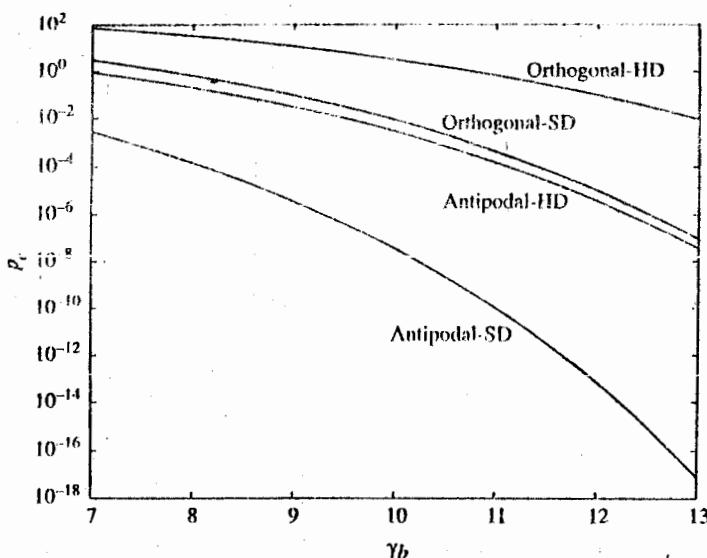
Hình 8.16 thể hiện bốn đồ thị tương ứng với truyền tin bằng tín hiệu đối cực và tín hiệu trực giao, với giải mã quyết định cứng và quyết định mềm.

MATLAB *script* tạo ra hình vẽ này được cho dưới đây.

### m-file

*% Chương trình MatLab cho bài tập minh họa 12, chương 8.*

```
[p_err_ha,gamma_b]=p_e_hd_a(7,13,11,15,3);
[p_err_ho,gamma_b]=p_e_hd_o(7,13,11,15,3);
[p_err_so,gamma_b]=p_e_sd_o(7,13,11,15,3);
[p_err_sa,gamma_b]=p_e_sd_a(7,13,11,15,3);
semilogy(gamma_b,p_err_sa,gamma_b,p_err_so,gamma_b,p_err_ha,gamma_b,p_err_ho)
```



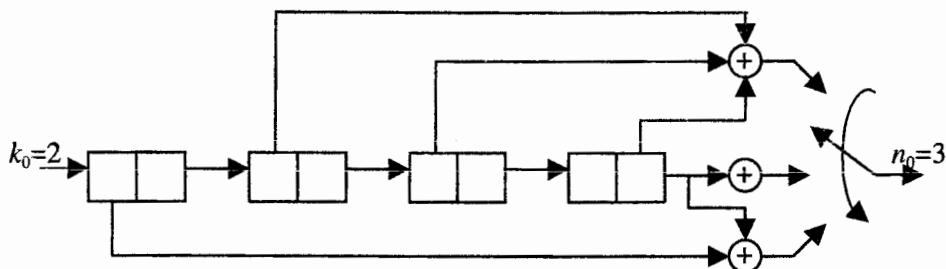
Hình 8.16 So sánh việc truyền bằng tín hiệu đối cực (*antipodal*) với truyền bằng tín hiệu trực giao (*orthogonal*) và giải mã quyết định mềm (SD) hay giải mã quyết định cứng (HD)

### 8.3.2 CÁC MÃ CHẬP

Trong các mã khối, mỗi dãy  $k$  bít thông tin được ánh xạ thành một dãy  $n$  bít lối vào kênh theo một cách cố định, bất luận các bít thông tin trước đó ra sao.

Trong các mã chập, mỗi dãy  $k_0$  bít thông tin được ánh xạ thành một dãy lối vào kênh có độ dài  $n_0$ , song dãy lối vào kênh không chỉ phụ thuộc vào  $k_0$  bít thông tin ngay lúc đó mà còn phụ thuộc vào cả  $(L-1)k_0$  bít lối vào cuối cùng của bộ mã hoá. Như vậy, bộ mã hoá có cấu trúc của một máy trạng thái hữu hạn (*finite-state machine*), trong đó tại mỗi thời điểm, dãy lối ra không chỉ phụ thuộc vào dãy đầu vào mà còn phụ thuộc cả vào trạng thái của bộ mã hoá được quyết định bởi  $(L-1)k_0$  bít lối vào cuối cùng tính đến thời điểm đang xét. Tham số  $L$  được gọi là chiều dài ràng buộc của mã chập<sup>1</sup>. Do vậy, một mã chập nhị phân chính là một máy trạng thái hữu hạn có tất cả  $2^{k_0(L-1)}$  trạng thái. Hình 8.17 là sơ đồ của một bộ mã hoá chập với  $k_0 = 2$ ,  $n_0 = 3$  và  $L = 4$ .

Trong bộ mã hoá chập này, các bít thông tin được tải vào bộ ghi dịch cứ mỗi lần 2 bít, và 2 bít thông tin cuối cùng trong bộ ghi dịch sẽ chạy ra. Ba bít mã được tính khi đó như được thể hiện trên hình vẽ và được truyền đi qua kênh. Tỉ lệ mã là  $R = 2/3$ . Chú ý rằng 3 bít lối ra bộ mã hoá được truyền đi qua kênh phụ thuộc vào 2 bít thông tin đã được tải vào bộ ghi dịch cũng như phụ thuộc cả vào nội dung của cả 3 cấp đầu tiên (6 bít) của bộ ghi dịch. Nội dung của cấp cuối cùng (2 bít) không ảnh hưởng tới lối ra bởi vì chúng rời khỏi bộ ghi dịch ngay khi có 2 bít thông tin được tải vào.



Hình 8.17 Một mã chập với  $k_0 = 2$ ,  $n_0 = 3$  và  $L = 4$

Một mã chập thường được xác định theo *các dãy sinh* của mã chập, ký hiệu

1) Một số tác giả định nghĩa  $m = Lk_0$  và một số lại coi  $(L-1)k_0$  là độ dài ràng buộc.

là  $\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_n$ . Thành phần thứ  $i$  của  $\mathbf{g}_j$ , với  $1 \leq i \leq k_0L$  và  $1 \leq j \leq n$ , là 1 nếu thành phần thứ  $i$  của bộ ghi dịch được nối với bộ cộng tương ứng với bít thứ  $j$  của đầu ra, và bằng 0 trong trường hợp ngược lại. Ví dụ, trong mã chập đã được vẽ trên hình 8.17, ta có

$$\begin{aligned}\mathbf{g}_1 &= [0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1] \\ \mathbf{g}_2 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1] \\ \mathbf{g}_3 &= [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]\end{aligned}$$

Mã chập xác định duy nhất khi  $\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_n$  được ấn định.

Chúng ta cũng định nghĩa ma trận sinh của mã chập theo

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \vdots \\ \mathbf{g}_n \end{bmatrix}$$

mà nó nói chung là một ma trận  $n \times k_0L$ . Đối với mã chập đã được thể hiện trên hình 8.17, chúng ta có

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Việc giả thiết rằng bộ ghi dịch tạo ra mã chập được tải đầy vào với các bít 0, trước khi bít thông tin đầu tiên đi vào (tức là bộ mã hoá khởi động ở trạng thái 0), là rất có lợi và giả thiết đó nghĩa là chuỗi bít thông tin được nhồi thêm  $(L-1)k_0$  bit 0 để đưa bộ mã hoá chập trở về trạng thái toàn 0 ban đầu. Chúng ta cũng giả thiết rằng chiều dài của dãy bít thông tin (lối vào bộ mã hoá chập) là một bội số của  $k_0$ . Nếu độ dài của dãy lối vào không phải là một bội của  $k_0$  thì ta có thể nhồi thêm vào dãy đó các bít 0 sao cho chiều dài của dãy lối vào trở thành một bội số của  $k_0$ . Điều này được thực hiện trước khi nhồi thêm  $(L-1)k_0$  bít 0 đã nói ở trên. Nếu sau khi nhồi thêm các bít 0 độ dài của dãy lối vào là  $nk_0$  thì chiều dài của dãy lối ra sẽ là  $(n+L-1)n_0$ , do đó tỉ lệ của mã sẽ là

$$\frac{nk_0}{(n+L-1)n_0}$$

Trên thực tế,  $n$  rất lớn hơn  $L$  nên biểu thức trên xấp xỉ khá tốt được bởi

$$R_c = \frac{k_0}{n_0}$$

Hàm MATLAB **cnv\_encl.m** được cho sau đây sẽ tạo ra dãy lối ra của bộ mã hoá chập khi  $G$ ,  $k_0$  và dãy lối vào được cho trước. Chú ý rằng việc nhồi thêm vào các bít 0 cho dãy lối vào được thực hiện bằng hàm MATLAB này. Dãy lối vào (được ký hiệu bằng tham số "input") bắt đầu với bít thông tin đầu tiên được đẩy vào bộ mã hoá. Các tham số  $n_0$  và  $L$  được tính ra từ ma trận  $G$ .

### m-file

```

function output=cnv_encl(g,k0,input)
%   cnv_encl(g,k0,input)
% Xác định dãy ra của bộ mã hoá chập nhị phân
% g là ma trận sinh của mã chập với n0 hàng và l*k0 cột
% Các cột là g1,g2,...,gn.
% k0 là số bit đưa vào bộ mã hoá trong 1 nhịp thời gian.
% input Dãy đầu vào nhị phân.

% Kiểm tra xem có cần nhồi thêm bit 0 hay không
if rem(length(input),k0) > 0
    input=[input,zeros(size(1:k0-rem(length(input),k0))]];
end
n=length(input)/k0;
% Kiểm tra kích cỡ của ma trận g
if rem(size(g,2),k0) > 0
    error('Lỗi, ma trận g có kích cỡ sai.')
end
% Tính l và n0
l=size(g,2)/k0;
n0=size(g,1);
% Thêm các bit 0 vào cuối dãy đầu vào
u=zeros(size(1:(l-1)*k0)),input,zeros(size(1:(l-1)*k0)));
% Tạo ma trận uu, đây là ma trận có các cột là nội dung của
% các phần tử nhớ trong bộ mã hoá chập tại các nhịp thời gian
u1=u(l*k0:-1:1);
for i=1:n+1-2

```

```

u1=[u1,u((i+l)*k0:-1:i*k0+1)];
end
uu=reshape(u1,l*k0,n+l-1);
% Tính dãy đầu ra
output=reshape(rem(g*uu,2),1,n0*(l+n-1));

```

### BÀI TẬP MINH HỌA

**Bài tập minh họa 8.13 [Bộ mã hoá chập]** Hãy xác định dãy lối ra của bộ mã hoá chập hình 8.17 khi dãy thông tin là

1 0 0 1 1 1 0 0 1 1 0 0 0 0 1 1 1

#### Lời giải

Ở đây chiều dài của dãy thông tin là 17, không là bội của  $k_0=2$ . Do đó, việc nhồi thêm các bít 0 sẽ được thực hiện. Trong trường hợp này, thêm một bít 0 là đủ, khi đó chiều dài của dãy sẽ là 18. Như vậy, chúng ta có dãy thông tin sau đây

1 0 0 1 1 1 0 0 1 1 0 0 0 0 1 1 1 0

Bây giờ, do chúng ta có

$$G = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

ta có được  $n_0 = 3$  và  $L = 4$ . (Điều này cũng thấy rõ được từ hình 8.17) Do đó, chiều dài của dãy lối ra là

$$\left(\frac{18}{2} + 4 - 1\right) \times 3 = 36$$

Việc nhồi các bít 0 cần thiết để bảo đảm rằng bộ mã hoá khởi động từ trạng thái toàn 0 và quay trở về trạng thái toàn 0 sẽ thực hiện thêm  $(L-1)k_0$  bít 0 vào đầu và cuối dãy thông tin. Do đó, dãy đang xét trở thành

0 0 0 0 0 0 1 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0

Sử dụng hàm **cnv\_encl.m**, chúng ta tìm được dãy lối ra là

0 0 0 0 0 1 1 0 1 1 1 1 0 1 0 1 1 1 0 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 1

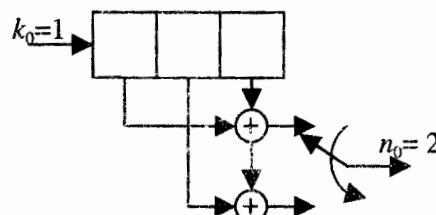
MATLAB script để giải bài tập này được cho dưới đây.

**m-file**

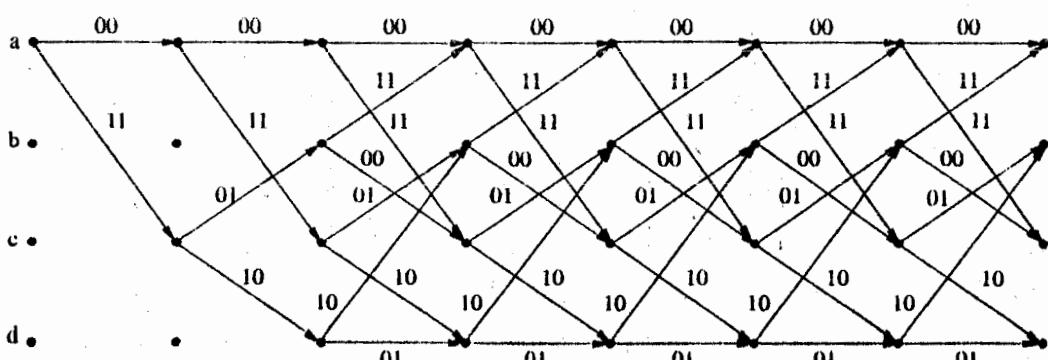
```
k0=2;
g=[0 0 1 0 1 0 0 1;0 0 0 0 0 0 0 1;1 0 0 0 0 0 0 1];
input=[1 0 0 1 1 1 0 0 1 1 0 0 0 0 1 1 1];
output=cnv_encl(g,k0,input);
```

**Biểu diễn các mã chập**

Chúng ta đã thấy rằng một mã chập có thể biểu diễn được hoặc bằng cấu trúc của bộ mã hoá hoặc bằng ma trận sinh  $G$ . Chúng ta cũng đã thấy rằng một bộ mã hoá chập có thể biểu diễn được như một máy hữu hạn trạng thái và do đó có thể mô tả được bằng một lược đồ chuyển trạng thái biểu diễn máy hữu hạn trạng thái. Một phương pháp biểu diễn các mã chập khác thông dụng hơn là bằng đồ hình lưới (*trellis diagram*). Một đồ hình lưới là một lược đồ chuyển trạng thái được vẽ theo thời gian. Do vậy, một đồ hình lưới thực chất là một dãy gồm  $2^{(L-1)k_0}$  trạng thái, được thể hiện bằng các điểm đối với mỗi chu kỳ đồng hồ, và các nhánh thì tương ứng với các chuyển dịch giữa các trạng thái này.



Hình 8.18 Một bộ mã hoá chập với  $k_0=1$ ,  $n_0=2$  và  $L=3$



Hình 8.19 Đồ hình lưới đối với mã chập trên hình 8.18

Xét mã chập với  $k_0=1$ ,  $n_0=2$ , và  $L=3$  như hình 8.18. Hiển nhiên, mã này có thể biểu diễn được bởi một máy hữu hạn trạng thái có 4 trạng thái tương ứng với các nội dung có thể có khác nhau của 2 phân tử đầu tiên của bộ ghi dịch, cụ thể là 00, 01, 10 và 11. Ta hãy biểu diễn 4 trạng thái này là  $a$ ,  $b$ ,  $c$ ,  $d$  một cách tương ứng. Để vẽ đồ hình lưới cho mã này chúng ta phải vẽ 4 điểm tương ứng với mỗi một trạng thái cho mỗi một chu kỳ đồng hồ rồi sau đó nối chúng tùy theo các chuyển đổi khác nhau giữa các trạng thái. Đồ hình lưới cho mã này được thể hiện trên hình 8.19.

Như chúng ta có thể thấy trên hình 8.19, trên trục thời gian tương ứng với các chu kỳ nhịp đồng hồ bốn trạng thái được ký hiệu bởi các điểm đen và các chuyển đổi giữa các trạng thái được chỉ thị bằng các nhánh nối các điểm đó. Trên mỗi một nhánh nối hai trạng thái, hai *symbol* nhị phân chỉ thị lối ra bộ mã hoá tương ứng với chuyển đổi đó. Cũng cần lưu ý rằng chúng ta luôn bắt đầu từ trạng thái toàn 0 (trạng thái  $a$ ), chuyển dịch qua lưới theo các nhánh tương ứng với dây lối vào cho trước, rồi trở về trạng thái toàn 0. Do đó, các từ mã của một mã chập thì tương ứng với các đường dẫn qua lưới, khởi đầu tại trạng thái toàn 0 và rồi lại quay trở về trạng thái toàn 0.

Số các trạng thái trên lưới sẽ tăng theo luỹ thừa của độ dài ràng buộc của mã chập. Ví dụ, với mã chập trên hình 8.17, số trạng thái là  $2^6=64$ ; do đó, cấu trúc của lưới phức tạp hơn nhiều.

### Hàm truyền của một mã chập

Đối với mỗi mã chập, hàm truyền cung cấp thông tin về các đường dẫn khác nhau đi qua lưới, bắt đầu từ trạng thái toàn 0 và trở về trạng thái này lần đầu tiên. Theo qui ước mã hoá đã được mô tả trên đây, một từ mã bất kỳ của một bộ mã hoá chập tương ứng với một đường dẫn qua lưới mà nó bắt đầu từ trạng thái toàn không và quay trở về trạng thái toàn 0. Như chúng ta sẽ thấy sau, hàm truyền của một mã chập giữ một vai trò chính yếu trong ấn định giới hạn chặn đối với xác suất lỗi của mã. Để nhận được hàm truyền của một mã chập, chúng ta chia trạng

thái toàn 0 thành hai trạng thái, một biểu thị trạng thái khởi đầu và một biểu thị lần đầu tiên trở về trạng thái toàn 0. Tất cả các trạng thái khác được biểu hiện như các trạng thái nằm ở giữa. Tương ứng với mỗi nhánh nối hai trạng thái, một hàm có dạng  $D^\alpha B^\beta J$  xác định được, trong đó  $\alpha$ , biểu thị số các bít 1 trong dãy bít lối ra còn  $\beta$  biểu thị số các bít 1 trong dãy lối vào tương ứng đối với nhánh đó. Hàm truyền của mã chập khi đó là hàm truyền của lưu đồ giữa trạng thái toàn 0 khởi đầu và trạng thái toàn 0 cuối cùng và được kí hiệu là  $T(D, N, J)$ . Mỗi số hạng của  $T(D, N, J)$  tương ứng với một đường dẫn qua lưới bắt đầu từ trạng thái toàn 0 và kết thúc ở trạng thái toàn 0. Số mũ của  $J$  chỉ thị số các nhánh của đường dẫn, số mũ của  $D$  cho thấy số các bit 1 trong từ mã tương ứng với đường dẫn này (hay tương đương là khoảng cách Hamming của từ mã này với từ mã toàn 0), và số mũ của  $N$  chỉ thị số bít 1 trong dãy thông tin lối vào.  $T(D, N, J)$  thể hiện các tính chất của tất cả các đường dẫn qua lưới bắt đầu từ trạng thái toàn 0 và trở lại trạng thái toàn 0 lần đầu tiên, cho nên để có được nó thì mọi vòng tự quay lại tại trạng thái toàn 0 sẽ được lờ đi không xét đến. Để nhận được hàm truyền của một mã chập, chúng ta có thể sử dụng mọi qui tắc có thể sử dụng được để có được hàm truyền của một lưu đồ. Để biết chi tiết hơn nữa về tính toán hàm truyền của một mã chập xin xem tài liệu tham khảo [1].

Theo các qui tắc xác định hàm truyền, chúng ta có thể dễ dàng chỉ ra rằng hàm truyền đối với mã chập được thể hiện trên hình 8.18 được cho bởi

$$T(D, N, J) = \frac{D^5 NJ^3}{1 - DNJ - DNJ^2}$$

mà nó khi khai triển ra có thể biểu diễn được theo

$$T(D, N, J) = D^5 NJ^3 + D^6 N^2 J^4 + D^6 N^2 J^5 + D^7 N^3 J^5 + \dots$$

Từ biểu thức này đối với  $T(D, N, J)$ , chúng ta có thể thấy rằng tồn tại một từ mã với trọng số Hamming bằng 5, hai từ mã với trọng số Hamming bằng 6, vân vân. Nó cũng cho thấy, thí dụ, rằng từ mã với trọng số Hamming bằng 5 thì tương ứng với một dãy lối vào có trọng số Hamming bằng 1 và có chiều dài bằng 3. Số

mũ nhõ nhất của  $D$  trong khai triển của  $T(D, N, J)$  được gọi là *khoảng cách tự do* của mã chập và được ký hiệu là  $d_{free}$ . Trong ví dụ này,  $d_{free} = 5$ .

### Giải các mã chập

Có nhiều thuật toán giải các mã chập. Thuật toán Viterbi có lẽ là phương pháp giải mã chập được sử dụng rộng rãi nhất. Thuật toán này đặc biệt đáng quan tâm vì nó là một thuật toán giải mã theo khả năng cực đại, dựa trên việc thu tín hiệu lối ra của kênh thực hiện dò tìm suốt lưới để tìm ra đường dẫn có khả năng lớn nhất tạo ra dãy thu được. Nếu việc giải mã quyết định cứng được sử dụng, thuật toán này tìm ra đường dẫn có khoảng cách Hamming nhỏ nhất tới dãy thu được, và nếu giải mã quyết định mềm được áp dụng thì thuật toán Viterbi tìm ra đường dẫn có khoảng cách Euclidean nhỏ nhất tới dãy thu được.

Trong giải mã quyết định cứng các mã chập, chúng ta muốn chọn ra một đường dẫn suốt qua lưới mà từ mã của nó, ký hiệu là  $c$ , có khoảng cách Hamming cực tiểu tới dãy thu được đã lượng tử hóa  $y$ . Trong giải mã quyết định cứng, kênh là kênh không nhớ nhị phân. (Kênh là không nhớ xuất phát từ giả thiết tạp âm trên kênh là tạp âm trắng). Do đường dẫn mong đợi khởi đầu từ trạng thái toàn không và lại quay trở về trạng thái toàn không, chúng ta giả sử rằng đường dẫn này trải ra trên  $m$  nhánh cả thảy, và do mỗi một nhánh tương ứng với  $n_0$  bít lối ra bộ mã hoá nên tổng số bít trong  $c$  và  $y$  là  $mn_0$ . Chúng ta ký hiệu dãy các bít tương ứng với nhánh thứ  $i$  là  $c_i$  và  $y_i$ , trong đó  $1 \leq i \leq m$ , và mỗi một  $c_i$  và  $y_i$  đều có độ dài là  $n_0$ . Khoảng cách Hamming giữa  $c$  và  $y$  do vậy là

$$d(c, y) = \sum_{i=1}^m d(c_i, y_i) \quad (8.3.33)$$

Trong giải mã quyết định mềm, chúng ta có một tình huống tương tự với ba khác biệt sau:

Thay vì  $y$ , chúng ta giải quyết trực tiếp với véc-tơ  $r$ , là véc-tơ lối ra của bộ giải điều chế số tối ưu (dạng bộ lọc phổi hợp hay bộ tương quan).

Thay vì dãy  $c$  gồm các bít 0 và 1, chúng ta giải quyết dãy tương ứng  $c'$ , với

$$c_p' = \begin{cases} \sqrt{E}, & \text{nếu } c_p = 1 \\ -\sqrt{E}, & \text{nếu } c_p = 0 \end{cases}$$

đối với  $1 \leq i \leq m$  và  $1 \leq j \leq n$ .

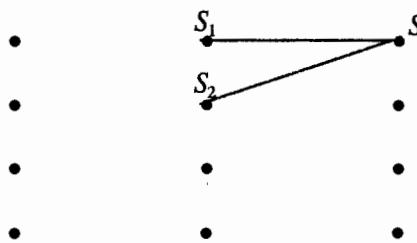
Thay vì khoảng cách Hamming, chúng ta sử dụng khoảng cách Euclidean. Điều này là hệ quả của việc kênh đang nghiên cứu là một kênh AWGN.

Từ trên chúng ta được

$$d_E^2(c', r) = \sum_{i=1}^m d_E^2(c_i, r_i) \quad (8.3.34)$$

Từ các biểu thức (8.3.33) và (8.3.34), chúng ta thấy rằng dạng tổng quát của vấn đề chúng ta phải giải quyết là: với một véc-tơ  $a$  đã cho, hãy tìm một đường dẫn suốt qua lưới khởi đầu từ trạng thái toàn không và kết thúc tại trạng thái toàn không sao cho độ đo khoảng cách nào đó giữa  $a$  và  $b$  tương ứng với đường dẫn đó là nhỏ nhất. Điều quan trọng làm cho bài toán này giải được dễ dàng là ở chỗ khoảng cách giữa  $a$  và  $b$  trong cả hai trường hợp mà ta đang quan tâm đều có thể viết được là tổng các khoảng cách ứng với các nhánh thành phần của đường dẫn. Điều này dễ dàng thấy được từ (8.3.33) và (8.3.34).

Bây giờ ta hãy giả sử rằng ta đang giải quyết với một mã chập với  $k_0 = 1$ . Điều này có nghĩa là cả thảy chỉ có hai nhánh đi vào mỗi một trạng thái trong lưới. Nếu đường dẫn tối ưu tại một điểm nhất định nào đó đi qua trạng thái  $S$  thì sẽ có hai đường dẫn nối từ hai trạng thái  $S_1$  và  $S_2$  trước đó tới trạng thái này. (Xem hình 8.20).



Hình 8.20 Dẫn giải thuật toán Viterbi

Nếu chúng ta muốn xem xét đường dẫn nào trong hai đường dẫn đó là một đề cử tốt để tối thiểu hóa khoảng cách tổng cộng thì chúng ta phải cộng các metric

tổng cộng (nhỏ nhất) tại các trạng thái  $S_1$  và  $S_2$  với các *metric* của các nhánh nối hai trạng thái này tới trạng thái  $S$ . Sau đó, hiển nhiên là nhánh có tích luỹ *metric* tổng cộng cho tới trạng thái  $S$  nhỏ nhất sẽ là một đề cử để xem xét đối với trạng thái sau trạng thái  $S$ . Nhánh này được gọi là một nhánh sống còn (*survivor*) tại trạng thái  $S$ , còn nhánh kia đơn giản không phải là một đề cử thích hợp và được xoá đi. Böyle giờ, sau khi nhánh sống còn tại trạng thái  $S$  xác định được, chúng ta cũng lưu lại *metric* nhỏ nhất cho tới trạng thái này và ta có thể chuyển sang trạng thái tiếp theo. Thủ tục này được tiếp tục cho tới khi ta đạt tới trạng thái toàn không ở cuối của lưới. Đối với các trường hợp  $k_0 > 1$ , điều khác biệt duy nhất là tại mỗi trạng thái chúng ta phải chọn ra một đường dẫn sống còn trong số  $2^{k_0}$  nhánh dẫn tới trạng thái  $S$ .

Thủ tục trên đây có thể tóm tắt lại được trong thuật toán sau đây, được biết đến với tên gọi *thuật toán Viterbi*.

1. Phân dãy nhận được thành  $m$  dãy con có độ dài  $n_0$ .
2. Vẽ một lưới có độ sâu  $m$  đối với mã đang xét. Đối với  $L-1$  trạng thái cuối cùng của lưới, chỉ vẽ các đường dẫn tương ứng với các dãy toàn không lối vào. [Điều này thực hiện được do chúng ta biết rằng dãy lối vào được nhồi thêm  $k_0(L-1)$  con 0].
3. Cho  $l = 1$  và đặt *metric* của trạng thái toàn không ban đầu là 0.
4. Tìm khoảng cách của dãy con thứ  $l$  của dãy nhận được với tất cả các nhánh nối các trạng thái thứ  $l$  tới các trạng thái tầng thứ  $(l+1)$  của lưới.
5. Cộng các khoảng cách này với các *metric* của các trạng thái tầng thứ  $l$  để nhận được các đề cử *metric* đối với các trạng thái tầng thứ  $(l+1)$ . Đối với mỗi một trạng thái của tầng thứ  $(l+1)$ , có cả thảy  $2^{k_0}$  *metric* đề cử, mỗi một trong chúng tương ứng với một nhánh kết cuối tại trạng thái đó.
6. Đối với mỗi một trạng thái tại tầng thứ  $(l+1)$ , chọn ra giá trị tối thiểu của các *metric* đề cử và gán cho nhánh tương ứng với giá trị tối thiểu đó nhãn *nhánh sống còn*, rồi gán giá trị tối thiểu của các giá trị đề cử *metric* làm các *metric* của các trạng thái tầng thứ  $(l+1)$ .

7. Nếu  $l = m$ , chuyển sang bước tiếp theo, nếu không thì chuyển về bước thứ 4.

8. Khởi đầu với trạng thái toàn không tại tầng thứ  $(m+1)$ , đi lần ngược trở lại qua lưới theo các nhánh sống còn cho tới trạng thái toàn không ban đầu. Đường dẫn này là đường dẫn tối ưu và dãy bít lối vào tương ứng với nó là dãy thông tin được giải mã khả năng cực đại. Để nhận được lời phỏng đoán tốt nhất về dãy bít lối vào, hãy loại bỏ đi  $k_0(L-1)$  con không cuối cùng khỏi dãy này.

Như chúng ta có thể thấy từ thuật toán trên đây, độ trễ giải mã và số lượng bộ nhớ cần thiết cho việc giải mã một dãy thông tin dài thì không thể chấp nhận được. Việc giải mã chưa thể bắt đầu chừng nào toàn bộ dãy (mà nó có thể rất dài trong trường hợp của các mã chập) chưa thu nhận được, thêm vào đó, các đường dẫn sống còn tổng cộng phải được trữ lại. Trong thực tế, cần có một giải pháp cận tối ưu mà nó không gây nên các vấn đề đó. Một giải pháp như thế, thường được gọi là *cắt ngắn độ nhớ đường dẫn*, là giải pháp theo đó bộ giải mã tại mỗi một tầng chỉ dò tìm ngược lại  $\delta$  tầng trong lưới chứ không dò ngược lại cho tới tận điểm khởi đầu của lưới. Với giải pháp này, tại tầng thứ  $(\delta+1)$ , bộ giải mã tiến hành quyết định trên các bít lối vào tương ứng với tầng thứ nhất của lưới ( $k_0$  bít đầu tiên), và các bít thu được trong tương lai thì không làm thay đổi quyết định này. Điều này có nghĩa là độ trễ giải mã sẽ là  $\delta k_0$  bít, và cần giữ lại các đường dẫn sống còn tương ứng với  $\delta$  tầng cuối cùng. Các mô phỏng máy tính đã chỉ ra rằng nếu  $\delta \geq 5L$  thì thiệt hại về chất lượng do cắt ngắn độ nhớ đường dẫn có thể bỏ qua được.

### BÀI TẬP MINH HỌA

**Bài tập minh họa 8.14 [Giải mã Viterbi]** Ta hãy giả sử rằng trong giải mã quyết định cứng, dãy nhận được đã lượng tử hoá là

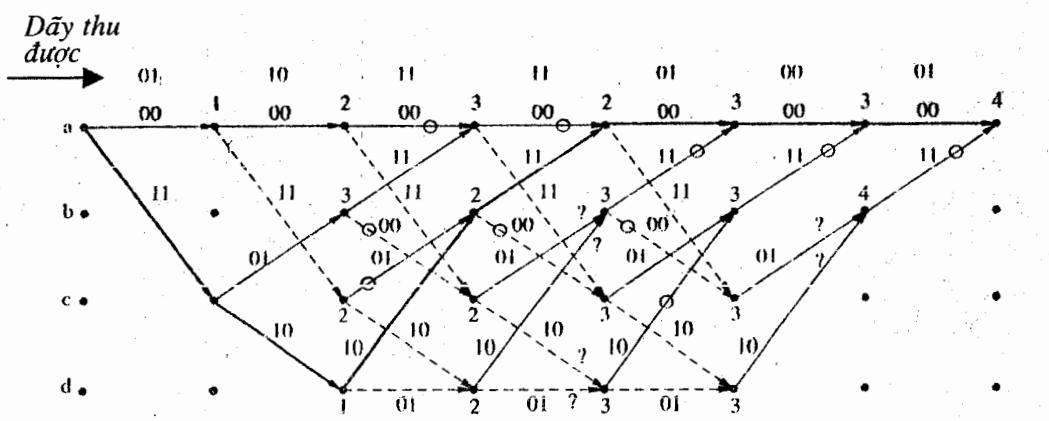
$$y = 01101111010001$$

Mã chập là mã đã được cho trong hình 8.18. Hãy tìm dãy tin tức có khả năng cực đại và số các lối.

### Lời giải

Mã này là một mã  $(2, 1)$  với  $L = 3$ . Độ dài của dãy nhận được  $y$  là 14. Điều đó

có nghĩa là  $m=7$ , và chúng ta cần phải vẽ một lưới có độ sâu bằng 7. Ta cũng hãy chú ý rằng do dãy thông tin lối vào được nhồi thêm  $k_0(L-1) = 2$  con 0, đối với hai tầng cuối cùng của lưới chúng ta sẽ chỉ vẽ các nhánh tương ứng với các lối vào toàn không. Điều đó có nghĩa là độ dài thực của dãy lối vào chỉ là 5, mà nó sau khi được nhồi thêm 2 con 0 đã tăng lên thành 7. Biểu đồ lưới cho trường hợp này được thể hiện trên hình 8.21.



Hình 8.21 Biểu đồ lưới đồi với giải mã Viterbi của dãy 01101111010001

Dãy thu nhận được  $y$  cũng được thể hiện trên hình vẽ này. Hãy để ý rằng khi vẽ lưới trong hai tầng cuối cùng, chúng ta đã xét chỉ các lối vào 0 tới bộ giải mã. (Chú ý rằng trong hai tầng cuối, không có các đường đứt nét ứng với các lối vào 1). Nay giờ, *metric* của trạng thái đầu toàn không được đặt là 0, và các *metric* của tầng tiếp theo được tính toán. Trong bước này, chỉ có một nhánh đi vào mỗi trạng thái; do đó không có sự so sánh, và các *metric* (mà chúng là các khoảng cách Hamming giữa đoạn của dãy nhận được và các nhánh của lưới) được cộng với *metric* của trạng thái trước đó. Trong tầng tiếp theo, cũng không có sự so sánh nào. Trong tầng thứ tư, trước tiên chúng ta có hai nhánh đi vào mỗi trạng thái. Điều này nghĩa là tại đây phải thực hiện một so sánh và các nhánh sống còn được chọn ra. Từ hai nhánh đi vào mỗi trạng thái, một nhánh ứng với *metric* tích luỹ tổng cộng cực tiểu sẽ được giữ lại làm nhánh sống còn, còn các nhánh khác được

loại đi (được đánh dấu bằng một vòng tròn nhỏ trên lưới). Nếu như tại một tầng nào đó, hai đường dẫn đều dẫn đến cùng một *metric* thì mỗi một trong chúng đều có thể là nhánh sống còn. Những trường hợp như vậy được đánh dấu bằng dấu hỏi trong lưới. Thủ tục này được tiếp diễn cho tới trạng thái toàn không cuối cùng của lưới; sau đó thì khởi đầu từ trạng thái đó, chúng ta chuyển dịch theo các nhánh sống còn về trạng thái toàn không ban đầu. Đường dẫn này, được chỉ ra bằng đường liền đậm xuyên qua lưới, là đường dẫn tối ưu. Dãy bít lỗi vào tương ứng với đường dẫn này là 1100000, trong đó hai số 0 cuối cùng không phải là các bít thông tin mà là các bít 0 đã được nhồi thêm vào để đưa bộ mã hoá về trạng thái toàn không ban đầu. Do đó, dãy thông tin là 11000. Từ mã tương ứng đối với đường dẫn đã được chọn là 11101011000000, có khoảng cách Hamming tính từ dãy nhận được là 4. Mọi đường dẫn khác đi qua lưới đều ứng với các từ mã có khoảng cách Hamming lớn hơn tính từ dãy nhận được.

Đối với giải mã quyết định mềm, một thủ tục tương tự được tuân theo song với các bình phương khoảng cách Euclide thay cho các khoảng cách Hamming.

Hàm MATLAB **viterbi.m** được cho dưới đây áp dụng thuật toán Viterbi để giải mã một chuỗi lỗi ra của kênh. Thuật toán này có thể sử dụng được cho giải mã các mã chập cả với quyết định cứng lẫn quyết định mềm. File **metric.m** tách riêng được dùng để xác định *metric* sử dụng trong quá trình giải mã. Đối với giải mã quyết định cứng, *metric* này là khoảng cách Hamming; đối với giải mã quyết định mềm thì đó là khoảng cách Euclidean. Đối với các trường hợp trong đó lỗi ra kênh được lượng tử hoá, *metric* thường là âm của logarit khả năng,  $-\log p(\text{lỗi ra kênh}|\text{lỗi vào kênh})$ . Một số các *m-file* ngắn được hàm **viterbi.m** gọi đến cũng được cho dưới đây.

### **m-file**

```
function [decoder_output,survivor_state,cumulated_metric]=viterbi(G,k,channel_output)
%VITERBI  The Viterbi decoder for convolutional codes
% [decoder_output,survivor_state,cumulated_metric]=viterbi(G,k,channel_output)
% G is a n x Lk matrix each row of which determines the connections from
% the shift register to the n-th output of the code, k/n is the rate of the code.
% survivor_state is a matrix showing the optimal path through the trellis.
```

% The metric is given in a separate function metric( $x,y$ ) and can be specified  
 % to accomodate hard and soft decision. This algorithm minimizes the metric  
 % rather than maximizing the likelihood.

```

n=size(G,1);
% check the sizes
if rem(size(G,2),k) ~=0
    error('Size of G and k do not agree')
end
if rem(size(channel_output,2),n) ~=0
    error('channel output not of the right size')
end
L=size(G,2)/k;
number_of_states=2^((L-1)*k);
% generate state transition matrix, output matrix, and input matrix
for j=0:number_of_states-1
    for l=0:2^k-1
        [next_state,memory_contents]=nxt_stat(j,l,L,k);
        input(j+1,next_state+1)=l;
        branch_output=rem(memory_contents*G',2);
        nextstate(j+1,l+1)=next_state;
        output(j+1,l+1)=bin2deci(branch_output);
    end
end
state_metric=zeros(number_of_states,2);
depth_of_trellis=length(channel_output)/n;
channel_output_matrix=reshape(channel_output,n,depth_of_trellis);
survivor_state=zeros(number_of_states,depth_of_trellis+1);
% start decoding of non-tail channel outputs
for i=1:depth_of_trellis-L+1
    flag=zeros(1,number_of_states);
    if i <= L
        step=2^((L-i)*k);
    else
        step=1;
    end
    for j=0:step:number_of_states-1
        for l=0:2^k-1
            branch_metric=0;
            binary_output=deci2bin(output(j+1,l+1),n);
            for ll=1:n
                branch_metric=branch_metric+metric(channel_output_matrix(ll,i),binary_output(ll));
            end
        end
    end
end

```

```

if((state_metric(nextstate(j+1,l+1)+1,2) > state_metric(j+1,1)...
    +branch_metric) | flag(nextstate(j+1,l+1)+1)==0)
    state_metric(nextstate(j+1,l+1)+1,2) = state_metric(j+1,1)+branch_metric;
    survivor_state(nextstate(j+1,l+1)+1,i+1)=j;
    flag(nextstate(j+1,l+1)+1)=1;
end
end
state_metric=state_metric(:,2:-1:1);
end
% start decoding of the tail channel-outputs
for i=depth_of_trellis-L+2:depth_of_trellis
    flag=zeros(1,number_of_states);
    last_stop=number_of_states/(2^(i-depth_of_trellis+L-2)*k));
    for j=0:last_stop-1
        branch_metric=0;
        binary_output=deci2bin(output(j+1,1),n);
        for ll=1:n
            branch_metric=branch_metric+metric(channel_output_matrix(ll,i),binary_output(ll));
        end
        if((state_metric(nextstate(j+1,1)+1,2) > state_metric(j+1,1)...
            +branch_metric) | flag(nextstate(j+1,1)+1)==0)
            state_metric(nextstate(j+1,1)+1,2) = state_metric(j+1,1)+branch_metric;
            survivor_state(nextstate(j+1,1)+1,i+1)=j;
            flag(nextstate(j+1,1)+1)=1;
        end
    end
    state_metric=state_metric(:,2:-1:1);
end
% generate the decoder output from the optimal path
state_sequence=zeros(1,depth_of_trellis+1);
state_sequence(1,depth_of_trellis)=survivor_state(1,depth_of_trellis+1);
for i=1:depth_of_trellis
    state_sequence(1,depth_of_trellis-i+1)=survivor_state((state_sequence(1,depth_of_trellis+2-i)...+1),depth_of_trellis-i+2);
end
decoder_output_matrix=zeros(k,depth_of_trellis-L+1);
for i=1:depth_of_trellis-L+1
    dec_output_deci=input(state_sequence(1,i)+1,state_sequence(1,i+1)+1);
    dec_output_bin=deci2bin(dec_output_deci,k);
    decoder_output_matrix(:,i)=dec_output_bin(k:-1:1)';
end

```

```
decoder_output=reshape(decoder_output_matrix,1,k*(depth_of_trellis-L+1));  
cumulated_metric=state_metric(1,1);
```

**m-file**

```
function distance=metric(x,y)  
if x==y  
    distance=0;  
else  
    distance=1;  
end
```

**m-file**

```
function [next_state,memory_contents]=nxt_stat(current_state,input,L,k)  
binary_state=deci2bin(current_state,k*(L-1));  
binary_input=deci2bin(input,k);  
next_state_binary=[binary_input,binary_state(1:(L-2)*k)];  
next_state=bin2deci(next_state_binary);  
memory_contents=[binary_input,binary_state];
```

**m-file**

```
function y=bin2deci(x)  
l=length(x);  
y=(l-1:-1:0);  
y=2.^y;  
y=x.*y;
```

**m-file**

```
function y=deci2bin(x,l)  
y = zeros(1,l);  
i = 1;  
while x>=0 & i<=l  
    y(i)=rem(x,2);  
    x=(x-y(i))/2;  
    i=i+1;  
end  
y=y(l:-1:1);
```

**BÀI TẬP MINH HỌA**

**Bài tập minh họa 8.15** Hãy lặp lại bài tập minh họa 8.14 bằng cách sử dụng

hàm MATLAB **viterbi.m**.

### Lời giải

Việc dùng các lối vào sau đây là đủ để sử dụng *m-file viterbi.m*

$$G = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$k = 1$$

$$\text{channel\_output} = [0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1]$$

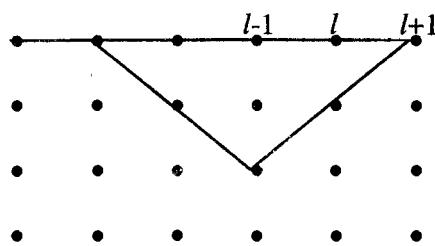
mà nó đưa đến lối ra bộ giải mã *decoder\_output*=[1 1 0 0 0] và một *metric* tích luỹ bằng 4.

### Các giới hạn chặn về xác suất lỗi đối với các mã chập

Việc tìm các giới hạn chặn của chất lượng tính theo lỗi đối với các mã chập thì khác với phương pháp đã được sử dụng đối với các mã khồi do ở đây chúng ta đang giải quyết với các dãy có độ dài rất lớn; vì khoảng cách tự do của các mã này thường là nhỏ nên rút cục một số lỗi sẽ có thể xảy ra. Số các lỗi này là một biến ngẫu nhiên tùy thuộc vào cả các đặc tính của kênh (tỷ số tín trên tạp trong giải mã quyết định mềm và xác suất chuyển trong giải mã quyết định cứng) lẫn độ dài của dãy lỗi vào. Dãy lỗi vào càng dài, khả năng gây lỗi càng lớn. Do vậy, việc chuẩn hoá số lỗi trên độ dài của dãy lỗi vào là có ý nghĩa. Một độ đo thường được chấp nhận để so sánh chất lượng lỗi của các mã chập là kỳ vọng của số bít thu được bị lỗi trên số bít lỗi vào. Để tìm một giới hạn chặn đối với số bít lỗi trung bình trên một bít lỗi vào, chúng ta trước tiên tìm giới hạn chặn của số bít lỗi trung bình đối với mỗi một dãy bít lỗi vào có độ dài  $k$ . Để xác định giới hạn chặn này, ta hãy giả sử rằng dãy toàn 0 được phát đi<sup>2</sup> và không có lỗi nào cho tới tầng thứ  $l$  của việc giải mã. Bây giờ,  $k$  bít thông tin đi vào bộ mã hoá và gây ra việc chuyển tới tầng tiếp theo trong lưới. Chúng ta muốn tìm ra một giới hạn chặn đối với số bít lỗi trung bình có thể xảy ra do khối lỗi vào có độ dài  $k$  này. Do chúng ta đang giả sử rằng

2) Do tính tuyến tính của các mã chập nên chúng ta có thể giả thiết như vậy mà không mất đi tính tổng quát.

không có lỗi nào cho tới tầng thứ  $l$  nên cho tới tầng này đường dẫn toàn 0 xuyên qua lưới đang có *metric* nhỏ nhất. Nay giờ, khi ta chuyển sang tầng tiếp theo (tầng thứ  $l+1$ ), có thể có một đường dẫn khác xuyên qua lưới sẽ có một *metric* còn nhỏ hơn đường dẫn toàn 0 và do đó sẽ gây ra các lỗi. Nếu điều này xảy ra, chúng ta nhất định phải có một đường dẫn xuyên qua lưới mà nó thoát đầu hoà lắn vào đường dẫn toàn 0 tại tầng thứ  $(l+1)$  và có một *metric* nhỏ hơn đường dẫn toàn 0. Sự kiện như thế được gọi là *sự kiện lỗi đầu tiên*, và xác suất tương ứng được gọi là *xác suất sự kiện lỗi đầu tiên*. Tình thế này được vẽ trên hình 8.22.



Hình 8.22 Đường dẫn ứng với sự kiện lỗi đầu tiên

Bước thứ nhất của chúng ta là tìm giới hạn chặn đối với xác suất sự kiện lỗi đầu tiên. Đặt  $P_2(d)$  là xác suất của việc một đường dẫn qua lưới có khoảng cách Hamming bằng  $d$  tính từ đường dẫn toàn 0 là một đường sống còn tại tầng thứ  $(l+1)$ . Do  $d$  thì lớn hơn  $d_{free}$ , chúng ta có thể chặn xác suất sự kiện lỗi đầu tiên bởi

$$P_e \leq \sum_{d=d_{free}}^{\infty} a_d P_2(d)$$

trong đó bên vẽ phải, chúng ta đã tính đến tất cả các đường dẫn qua lưới mà chúng hoà lắn với đường dẫn toàn 0 tại tầng thứ  $(l+1)$ .  $P_2(d)$  ký hiệu cho xác suất lỗi đối với một đường dẫn có khoảng cách Hamming là  $d$  tính từ đường dẫn toàn 0, còn  $a_d$  ký hiệu cho số các đường dẫn có khoảng cách  $d$  tính từ đường dẫn toàn 0. Giá trị của  $P_2(d)$  phụ thuộc vào việc giải mã quyết định cứng hay mềm được áp dụng.

Đối với giải mã quyết định mềm, nếu việc truyền tin bằng tín hiệu đối cực được sử dụng (PSK nhị phân) thì chúng ta có

$$P_2(d) = Q\left(\frac{d^E}{\sqrt{2N_0}}\right) = Q\left(\sqrt{\frac{2Ed}{N_0}}\right) = Q\left(\sqrt{2R_c d \frac{E_b}{N_0}}\right)$$

và do vậy

$$P_e \leq \sum_{d=d_{free}}^{\infty} a_d Q\left(\sqrt{2R_c d \frac{E_b}{N_0}}\right)$$

Sử dụng giới hạn chẵn trên đã biết đối với hàm  $Q$

$$Q(x) \leq \frac{1}{2} e^{-x^2/2}$$

ta nhận được

$$Q\left(\sqrt{2R_c d \frac{E_b}{N_0}}\right) \leq \frac{1}{2} e^{-R_c d E_b / N_0}$$

Bây giờ, để ý rằng

$$e^{-R_c d E_b / N_0} = D^d \Big|_{D=e^{-R_c E_b / N_0}}$$

cuối cùng ta nhận được

$$P_e \leq \frac{1}{2} \sum_{d=d_{free}}^{\infty} a_d D^d \Big|_{D=e^{-R_c E_b / N_0}} = \frac{1}{2} T_1(D) \Big|_{D=e^{-R_c E_b / N_0}}$$

trong đó

$$T_1(D) = T(D, N, J) \Big|_{N=J=1}$$

Đây là một giới hạn chẵn đối với xác suất sự kiện lỗi đầu tiên. Để tìm một giới hạn chẵn cho số bít lỗi trung bình đối với  $k$  bít lỗi vào,  $\bar{P}_b(k)$ , chúng ta để ý rằng mỗi một đường dẫn qua lưới đều gây nên một số nhất định các bít lỗi vào bị giải mã sai. Đối với một  $D^d N^{f(d)} J^{g(d)}$  tổng quát trong khai triển của  $T(D, N, J)$ ,<sup>3)</sup> cả thảy có  $f(d)$  bít lỗi vào khác không. Điều này có nghĩa là số bít lỗi vào trung bình bị lỗi có thể nhận được bằng cách nhân xác suất của việc chọn mỗi một đường dẫn với tổng số các lỗi lỗi vào mà chúng sẽ gặp phải nếu đường dẫn đó được

3) Ở đây chúng ta đã tuỳ tiện đổi chút trong ký hiệu. Số mũ  $N$  không hẳn là một hàm của  $d$ , song chúng ta đang ký hiệu nó bằng  $f(d)$ . Điều này dù sao cũng không ảnh hưởng gì tới kết quả cuối cùng.

chọn. Do đó số bít bị lỗi trung bình, trong trường hợp quyết định mềm, có thể chặn được bởi

$$\bar{P}_b(k) \leq \sum_{d=d_{free}}^{\infty} a_d f(d) Q\left(\sqrt{2R_c d \frac{E_b}{N_0}}\right) \leq \frac{1}{2} \sum_{d=d_{free}}^{\infty} a_d f(d) e^{-R_c d E_b / N_0} \quad (8.3.35)$$

Nếu chúng ta định nghĩa

$$T_2(D, N) = T(D, N, J)|_{J=1} = \sum_{d=d_{free}}^{\infty} a_d D^d N^{f(d)}$$

thì chúng ta sẽ có

$$\frac{\partial T_2(D, N)}{\partial N}|_{N=1} = \sum_{d=d_{free}}^{\infty} a_d f(d) D^d \quad (8.3.36)$$

Do đó, sử dụng (8.3.35) và (8.3.36) chúng ta được

$$\bar{P}_b(k) \leq \frac{1}{2} \frac{\partial T_2(D, N)}{\partial N} \Big|_{N=1, D=e^{-R_c E_b / N_0}}$$

Để nhận được trung bình số bít bị lỗi đối với mỗi một bít lỗi vào, chúng ta phải chia giới hạn chặn này cho  $k$ . Như thế, kết quả cuối cùng là

$$\bar{P}_b = \frac{1}{2k} \frac{\partial T_2(D, N)}{\partial N} \Big|_{N=1, D=e^{-R_c E_b / N_0}}$$

Đối với giải mã quyết định cứng, thủ tục cơ bản cũng tuân theo như trên. Điểm khác biệt duy nhất là giới hạn chặn đối với  $P_2(d)$ . Có thể chỉ ra được (xem [1]) rằng  $P_2(d)$  có thể chặn được bởi

$$P_2(d) \leq [4p(1-p)]^{d/2}$$

Từ kết quả này dễ dàng chỉ ra được rằng trong giải mã quyết định cứng, xác suất lỗi bị chặn trên theo

$$\bar{P}_b = \frac{1}{k} \frac{\partial T_2(D, N)}{\partial N} \Big|_{N=1, D=\sqrt{4p(1-p)}}$$

Việc so sánh giải mã quyết định cứng và giải mã quyết định mềm đối với các mã chấp cho thấy rằng ở đây, như trong trường hợp đối với các mã khối tuyến tính, việc giải mã quyết định mềm vượt trội hơn giải mã quyết định cứng bởi một lượng dự trữ cỡ 2 dB trong các kênh có tạp âm cộng trắng chuẩn.

## CÁC BÀI TẬP

- 8.1** Hãy viết một MATLAB *script* để vẽ đồ thị dung lượng của một kênh nhị phân đối xứng với xác suất chuyển  $p$  như một hàm của  $p$ , với  $0 \leq p \leq 1$ . Dung lượng kênh cực tiểu đối với các giá trị nào của  $p$  và giá trị cực tiểu đó là bao nhiêu?
- 8.2** Một kênh nhị phân không đối xứng được đặc trưng bởi các xác suất điều kiện  $p(0|1) = 0.2$  và  $p(1|0) = 0.4$ . Hãy vẽ đồ thị của lượng tin đồng thời  $I(X; Y)$  giữa lối vào và lối ra của kênh này như một hàm của  $p = P(X=1)$ . Với giá trị nào của  $p$  thì lượng tin tức cùng nhau này cực đại và lượng tin cực đại đó là bao nhiêu?
- 8.3** Một kênh Z là một kênh có đầu vào nhị phân, đầu ra nhị phân, với các chữ cái lối vào và lối ra  $X = Y = \{0, 1\}$  và được đặc trưng bằng  $p(0|1) = \varepsilon$  và  $p(1|0) = 0$ . Hãy vẽ đồ thị  $I(X; Y)$  như một hàm của  $p = P(X=1)$  đối với  $\varepsilon = 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 0.9, 1$ . Hãy xác định dung lượng của kênh trong từng trường hợp.
- 8.4** Một kênh lối vào nhị phân và lối ra ba mức được đặc trưng bằng các chữ cái lối vào  $X = \{0, 1\}$  và các chữ cái lối ra  $Y = \{0, 1, 2\}$  và các xác suất chuyển  $p(0|0) = 0.05$ ,  $p(1|0) = 0.2$ ,  $p(0|1) = 0.1$  và  $p(1|1) = 0.1$ . Hãy vẽ đồ thị  $I(X; Y)$  như một hàm của  $p = P(X=1)$  và hãy xác định dung lượng của kênh.
- 8.5** Một kênh lối vào ba mức, lối ra nhị phân được đặc trưng bởi các chữ cái lối vào và lối ra  $X = \{0, 1, 2\}$  và  $Y = \{0, 1\}$  và các xác suất chuyển  $p(0|0) = 0.05$ ,  $p(1|1) = 0.2$ , và  $p(0|2) = 0.1$ . Hãy vẽ đồ thị  $I(X; Y)$  như một hàm của  $p_1 = P(X=1)$  và  $p_2 = P(X=2)$ , và hãy xác định dung lượng của kênh.
- 8.6** Hãy vẽ đồ thị dung lượng của một kênh nhị phân đối xứng, trong đó tín hiệu trực giao nhị phân được sử dụng để truyền tin, như là một hàm của  $E_b/N_0$ .
- 8.7** Lặp lại bài tập minh họa 8.3, song giả sử rằng hai tín hiệu được truyền đi là có cùng năng lượng và trực giao nhau. Các kết quả của bạn khác thế nào so với các kết quả đã nhận được trong bài tập minh họa 8.3?
- 8.8** Hãy so sánh các đồ thị dung lượng kênh với quyết định cứng và quyết định mềm khi các tín hiệu trực giao được áp dụng. Hãy so sánh các kết quả này với các

kết quả nhận được đối với các tín hiệu đối cực.

**8.9** Hãy vẽ đồ thị dung lượng của một kênh đối xứng nhị phân có sử dụng các tín hiệu trực giao như một hàm số của  $E_b/N_0$ . Hãy làm một lần nữa với giả thiết rằng việc tách tín hiệu kết hợp và một lần nữa với giả thiết rằng việc tách tín hiệu không kết hợp được áp dụng.

**8.10** Hãy viết một MATLAB *script* tạo ra ma trận sinh của một mã Hamming ở dạng hệ thống đối với một  $m$  bất kỳ đã cho.

**8.11** Hãy lặp lại bài tập minh họa 8.10 có sử dụng tín hiệu trực giao để truyền tin với tách tín hiệu kết hợp (*coherent*) và không kết hợp (*noncoherent*).

**8.12** Hãy sử dụng mô phỏng Monte-Carlo để vẽ đồ thị xác suất lỗi theo  $\gamma_b$  trong bài tập minh họa 8.10.

**8.13** Lặp lại bài tập 8.12, song thay vì sử dụng tín hiệu trực giao để truyền tin lại sử dụng các tín hiệu đối cực, hãy so sánh chất lượng giải điều chế kết hợp và không kết hợp các tín hiệu trực giao khi giải mã quyết định mềm.

**8.14** Hãy sử dụng mô phỏng Monte-Carlo để vẽ đồ thị xác suất lỗi theo  $\gamma_b$  trong bài tập minh họa 8.12.

**8.15** Hãy sử dụng MATLAB để tìm dãy lỗi ra của bộ mã hóa chập đã được thể hiện trên hình 8.18 khi dãy lỗi vào là

110010101010010111101011111010

**8.16** Một mã chập được mô tả bằng

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

a. Nếu  $k=1$ , hãy xác định dãy lỗi ra của bộ mã hóa khi dãy lỗi vào là

110010101010010111101011111010

b. Lặp lại phần (a) với  $k = 2$ .

**8.17** Trong bài tập 8.15, sau khi nhận được dãy lỗi ra bộ mã hóa, hãy thay đổi sáu

bít đầu tiên của dãy nhận được, và hãy giải mã dãy có được bằng cách sử dụng giải mã Viterbi. Hãy so sánh dãy lỗi ra bộ giải mã với dãy đã phát đi. Bao nhiêu lỗi xảy ra? Hãy lặp lại bài tập này một lần nữa bằng cách thay đổi sáu bít cuối trong dãy nhận được và một lần nữa bằng cách thay đổi 3 bít đầu và 3 bít cuối, rồi sau đó hãy so sánh các kết quả. Trong mọi trường hợp kể trên đều sử dụng *metric Hamming*.

**8.18** Hãy tạo ra một dãy nhị phân đồng xác suất có độ dài 1000. Hãy mã hoá dãy này bằng cách sử dụng mã chập đã được cho trên hình 8.18. Hãy tạo ra bốn dãy lỗi nhị phân ngẫu nhiên, mỗi dãy lỗi đó dài 2000 với xác suất của con 1 lần lượt là 0.01, 0.05, 0.1 và 0.2. Hãy cộng (*modulo 2*) từng dãy trong bốn dãy lỗi đó với dãy đã được mã rồi sử dụng thuật toán giải mã Viterbi để giải mã các dãy có được. Trong từng trường hợp, hãy so sánh dãy đã giải mã với dãy lỗi vào bộ mã hoá và xác định tỷ lệ lỗi bít.

**8.19** Hãy sử dụng mô phỏng Monte-Carlo để vẽ đồ thị tỷ lệ lỗi bít theo  $\gamma_b$  trong một bộ mã hoá chập sử dụng mã đã nêu trên hình vẽ 8.18. Giả sử rằng sơ đồ điều chế là đối cực nhị phân, một lần với giải mã quyết định cứng và một lần với giải mã quyết định mềm. Hãy để cho  $\gamma_b$  nằm trong khoảng từ 3 dB tới 9 dB và hãy chọn độ dài dãy thông tin của bạn một cách thích hợp. Hãy so sánh các kết quả của bạn với các giới hạn chặn lý thuyết.

**8.20** Bộ mã hoá trên hình vẽ 8.18 được sử dụng để truyền thông tin qua một kênh với hai lỗi vào và ba lỗi ra. Các lỗi ra được ký hiệu bằng các số 0, 1 và 2. Đây là trường hợp trong đó lỗi ra của một kênh Gauss được lượng tử hoá thành ba mức. Các xác suất điều kiện của kênh này được cho là  $p(0|0) = p(1|1) = 0.9$ ,  $p(2|0) = p(2|1) = 0.09$ . Hãy sử dụng thuật toán Viterbi để giải mã dãy nhận được

02020110202112002220110101020220111112

## CHƯƠNG 9

# CÁC HỆ THỐNG THÔNG TIN TRẢI PHỔ

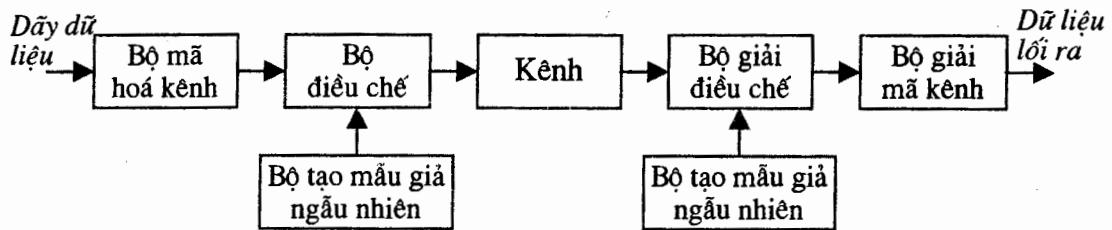
### 9.1 GIỚI THIỆU CHUNG

Các tín hiệu trải phổ đối với các hệ thống thông tin số nguyên thuỷ được phát triển và ứng dụng cho thông tin quân sự để (1) cung cấp khả năng chống lại việc gây nhiễu của địch; (2) để che dấu tín hiệu bằng cách phát nó với mức công suất thấp, nhờ đó, làm cho một người nghe không mong muốn khó phát hiện ra được sự có mặt của tín hiệu trong nền tạp âm; hoặc để (3) làm cho nhiều người sử dụng có thể truyền tin qua cùng một kênh. Ngày nay, tuy thế, các tín hiệu trải phổ đang được sử dụng nhằm cho phép truyền tin tin cậy trong một loạt các ứng dụng thương mại, bao gồm cả thông tin trên xe di động và thông tin vô tuyến liên sở (*interoffice*).

Các phần tử cơ sở của một hệ thống thông tin số trải phổ được minh họa trên hình 9.1. Chúng ta thấy rằng bộ mã hoá và giải mã, bộ điều chế và giải điều chế là các phần tử cơ sở của một hệ thống thông tin số truyền thống. Ngoài các phần tử này, một hệ thống thông tin trải phổ còn áp dụng hai bộ tạo chuỗi giả ngẫu nhiên như nhau, một trong chúng giao tiếp với bộ điều chế ở đầu phát và bộ kia thì giao tiếp với bộ giải điều chế ở đầu thu. Hai bộ này tạo ra một chuỗi nhị phân giả ngẫu nhiên hay giả tạp (PN: *PseudoNoise*) được sử dụng tại bộ điều chế để trải tín hiệu được phát đi về phổ và giải trải tín hiệu thu được tại bộ giải điều chế.

Chuỗi PN được tạo ra ở máy thu phải đồng bộ với chuỗi PN hàm chứa trong tín hiệu thu được để giải trải chính xác tín hiệu thu được đã được trải phổ. Trong một hệ thống thực tế, việc đồng bộ được thiết lập trước khi truyền đưa thông tin

bằng cách truyền một mẫu bít PN cố định được thiết kế sao cho máy thu sẽ tách được nó với một xác suất cao ngay cả khi có nhiễu. Sau khi việc đồng bộ thời gian của các bộ tạo chuỗi PN đã thực hiện xong, việc truyền thông tin bắt đầu. Trong chế độ truyền dữ liệu, hệ thống truyền tin thường bám định thời với tín hiệu thu được và giữ cho bộ tạo chuỗi PN được đồng bộ.



Hình 9.1 Mô hình một hệ thống thông tin số trải phổ.

Trong chương này, chúng ta sẽ xem xét hai loại cơ bản các tín hiệu trải phổ dùng cho truyền tin số, đó là trải phổ chuỗi trực tiếp (DS: *Direct-Sequence*) và trải phổ nhảy tần (FH: *Frequency-Hopped*).

Hai loại điều chế số sẽ được xem xét liên quan với trải phổ, đó là PSK và FSK. Điều chế PSK nói chung được sử dụng với trải phổ DS và thích hợp cho các ứng dụng trong đó sự kết hợp về pha giữa tín hiệu phát và tín hiệu thu có thể duy trì được trên một khoảng thời gian bao trùm vài khoảng *symbol* (hay bít). Về phương diện khác, điều chế FSK nói chung được sử dụng cùng với trải phổ nhảy tần và thích hợp với các ứng dụng trong đó sự kết hợp pha sóng mang không thể duy trì được do đặc tính truyền dẫn của kênh truyền tin biến đổi theo thời gian.

## 9.2 CÁC HỆ THỐNG TRẢI PHỔ CHUỖI TRỰC TIẾP

Ta hãy xem xét việc truyền dẫn một dãy thông tin nhị phân bằng điều chế PSK. Tốc độ thông tin là  $R$  bít trong một giây, và độ rộng một bít là  $T_b = 1/R$  giây. Độ rộng băng kênh có sẵn là  $B_c$  (Hz), trong đó  $B_c \gg R$ . Tại bộ điều chế, độ rộng băng của tín hiệu thông tin được trải rộng ra thành  $W = B_c$  (Hz) nhờ dịch pha của sóng mang một cách giả ngẫu nhiên với một tốc độ  $W$  lần trong một giây theo

mẫu tín hiệu ra của bộ tạo chuỗi PN. Tín hiệu đã điều chế có được gọi là *tín hiệu trai phổ chuỗi trực tiếp* (DS).

Tín hiệu băng gốc mang thông tin được ký hiệu là  $v(t)$  và biểu diễn được theo

$$v(t) = \sum_{n=-\infty}^{\infty} a_n g_T(t - nT_b) \quad (9.2.1)$$

trong đó  $\{a_n = \pm 1, -\infty < n < \infty\}$  còn  $g_T(t)$  là một xung chữ nhật có độ dài  $T_b$ . Tín hiệu này được nhân với tín hiệu đến từ bộ tạo chuỗi PN mà nó có thể biểu diễn được theo

$$c(t) = \sum_{n=-\infty}^{\infty} c_n p(t - nT_c) \quad (9.2.2)$$

trong đó  $\{c_n\}$  biểu thị chuỗi mã PN nhị phân gồm các dấu mã  $\pm 1$ , còn  $p(t)$  là một xung chữ nhật có độ dài  $T_c$ , như được minh họa trên hình 9.2. Thuật toán nhân này thực hiện việc trai độ rộng băng tín hiệu mang thông tin (mà độ rộng băng của nó vốn xấp xỉ  $R$  Hz) trở thành độ rộng băng rộng hơn chiếm bởi tín hiệu của bộ tạo PN (mà độ rộng băng của nó vốn xấp xỉ  $1/T_c$ ). Việc trai rộng phổ được minh họa trên hình 9.3 (trên đó các phổ hình chữ nhật được sử dụng cho đơn giản), mà hình vẽ này cho thấy tích chập của hai phổ: phổ hẹp tương ứng với tín hiệu mang thông tin và phổ rộng tương ứng với tín hiệu từ lối ra bộ tạo PN.

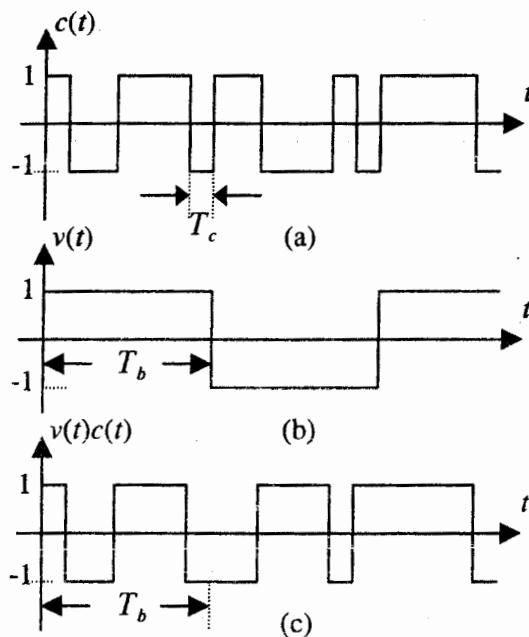
Tín hiệu tích  $v(t)c(t)$ , cũng đã minh họa trên hình vẽ 9.2, được sử dụng để điều chế biên độ sóng mang  $A_c \cos 2\pi f_c t$  và do vậy tạo nên tín hiệu DSB-SC (*Double-SideBand Supressed-Carrier: Tín hiệu điều chế hai biên có sóng mang được nén*) có dạng

$$u(t) = A_c v(t)c(t) \cos 2\pi f_c t \quad (9.2.3)$$

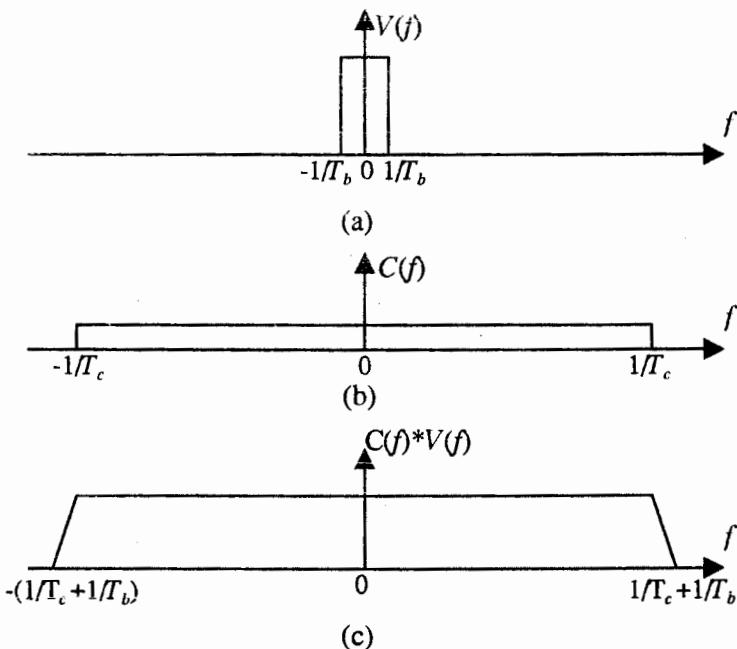
Do  $v(t)c(t) = \pm 1$  với mọi  $t$  nên dẫn đến rằng tín hiệu phát được điều chế sóng mang cũng còn có thể biểu diễn được theo

$$u(t) = A_c \cos[2\pi f_c t + \theta(t)] \quad (9.2.4)$$

trong đó  $\theta(t) = 0$  khi  $v(t)c(t) = 1$  và  $\theta(t) = \pi$  khi  $v(t)c(t) = -1$ . Do vậy, tín hiệu phát là một tín hiệu PSK nhị phân với pha của nó thay đổi với tốc độ  $1/T_c$ .



Hình 9.2 Sự tạo thành một tín hiệu trai phổ DS: (a) tín hiệu PN;  
(b) tín hiệu dữ liệu; (c) tín hiệu tích



Hình 9.3 Tích chập các phổ của (a) tín hiệu dữ liệu với (b) tín hiệu mã PN

Xung chữ nhật  $p(t)$  thường được gọi là một *chip* và độ dài thời gian của nó,  $T_c$ , được gọi là *khoảng chip*. Đại lượng nghịch đảo  $1/T_c$  được gọi là *tốc độ chip* và

tương ứng (gần đúng) với độ rộng băng  $W$  của tín hiệu phát đi. Tỷ lệ giữa khoảng bít  $T_b$  với khoảng chip  $T_c$  thường được chọn là một số nguyên trong các hệ thống trai phổ thực tế. Ta định nghĩa tỷ lệ này là

$$L_c = \frac{T_b}{T_c} \quad (9.2.5)$$

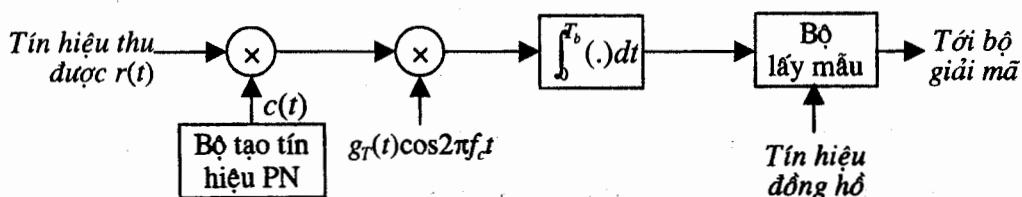
Khi đó,  $L_c$  là số chip của chuỗi mã PN có trong một bit. Một cách hiểu khác nữa là  $L_c$  biểu thị số các chuyển dịch pha  $180^\circ$  có thể có trong tín hiệu phát trong một khoảng bít  $T_b$ .

### 9.2.1 GIẢI ĐIỀU CHẾ TÍN HIỆU

Việc giải điều chế tín hiệu được thực hiện như đã được minh họa trên hình vẽ 9.4. Tín hiệu nhận được trước hết được nhân với một bản sao của dạng sóng  $c(t)$ , tạo ra bởi bộ tạo chuỗi mã PN ở máy thu được đồng bộ với mã PN có trong tín hiệu thu được. Thao tác này được gọi là giải trai (phổ), do tác động của việc nhân với  $c(t)$  tại máy thu sẽ huỷ bỏ tác động trai ở phần phát. Như thế, chúng ta có

$$A_c v(t) c^2(t) \cos 2\pi f_c t = A_c v(t) \cos 2\pi f_c t \quad (9.2.6)$$

vì  $c^2(t) = 1$  với mọi  $t$ . Tín hiệu nhận được từ thao tác giải trai  $A_c v(t) \cos 2\pi f_c t$  chiếm một độ rộng băng (gần đúng) là  $R$  Hz, mà đó chính là độ rộng băng của tín hiệu mang thông tin. Do đó, đối với tín hiệu đã giải trai thì bộ giải điều chế đơn thuần chỉ là một bộ tính tương quan chéo hay một bộ lọc phoi hợp truyền thống như đã được mô tả trong chương 5 và chương 7. Do bộ giải điều chế có một độ rộng băng trùng với độ rộng băng của tín hiệu đã giải trai, tạp nhiễu cộng gây nhiễu tín hiệu tại bộ giải điều chế chỉ là tạp âm rơi vào trong độ rộng băng thông tin của tín hiệu thu được.



Hình 9.4 Giải điều chế tín hiệu trai phổ DS

**Tác động của việc giải trại đối với một nhiễu băng hẹp**

Việc khảo sát tác động của một tín hiệu nhiễu tới việc giải điều chế tín hiệu mang tin mong muốn là rất đáng quan tâm. Giả sử rằng tín hiệu thu được là

$$r(t) = A_c v(t)c(t) \cos 2\pi f_c t + i(t) \quad (9.2.7)$$

trong đó  $i(t)$  ký hiệu cho nhiễu. Thao tác giải trại tại máy thu cho ta

$$r(t)c(t) = A_c v(t) \cos 2\pi f_c t + i(t)c(t) \quad (9.2.8)$$

Tác động của việc nhân nhiễu  $i(t)$  với  $c(t)$  là làm trại rộng độ rộng băng của  $i(t)$  thành  $W$  Hz.

Để lấy làm thí dụ, ta hãy xét một tín hiệu nhiễu hình sin có dạng

$$i(t) = A_J \cos 2\pi f_J t \quad (9.2.9)$$

trong đó  $f_J$  là một tần số nằm trong độ rộng băng của tín hiệu phát. Việc nhân nó với  $c(t)$  tạo ra một nhiễu băng rộng có mật phỏ công suất  $J_0 = P_J/W$ , trong đó  $P_J = A_J^2 / 2$  là công suất trung bình của nhiễu. Do tín hiệu mong muốn được giải điều chế bằng một bộ lọc phổi hợp (hay một bộ tính tương quan) mà nó có một độ rộng băng bằng  $R$  nên công suất tổng cộng của nhiễu tại lối ra bộ giải điều chế là

$$J_0 R = \frac{P_J R}{W} = \frac{P_J}{W/R} = \frac{P_J}{T_b/T_c} = \frac{P_J}{L_c} \quad (9.2.10)$$

Do vậy, công suất của tín hiệu nhiễu được giảm đi một lượng bằng với hệ số trại rộng độ rộng băng  $W/R$ . Hệ số  $W/R = T_b/T_c = L_c$  được gọi là tăng ích xử lý (*processing gain*) của hệ thống trại phổi. Sự suy giảm của công suất nhiễu là lý do cơ bản để sử dụng các tín hiệu trại phổi để truyền dẫn thông tin số trên các kênh có nhiễu.

Tóm lại, chuỗi mã PN được sử dụng ở máy phát để trại tín hiệu mang tin thành một băng rộng để truyền dẫn qua kênh. Khi tín hiệu thu được được nhân với một bản sao đồng bộ của tín hiệu mã PN, tín hiệu mong muốn được giải trại ngược lại thành có độ rộng băng hẹp, trong khi các tín hiệu nhiễu bất kỳ lại được trại rộng phổi ra trên một độ rộng băng lớn. Hiệu quả thực sự là một sự suy giảm công suất nhiễu theo một hệ số  $W/R$ , là tăng ích xử lý của hệ thống trại phổi.

Chuỗi mã PN  $\{c_n\}$  được giả thiết là đã biết trước chỉ đối với máy thu đã định. Bất kỳ máy thu nào khác mà không biết gì về chuỗi mã PN sẽ không thể giải điều chế được tín hiệu. Hệ quả là, việc sử dụng một chuỗi mã PN cho chúng ta một mức độ riêng tư (hay an ninh) mà nó không thể có được bằng cách điều chế truyền thống. Cái giá chủ yếu phải trả để có được độ an ninh cũng như tăng ích chống nhiễu này là việc tăng độ rộng băng sử dụng của kênh và sự tăng mức độ phức tạp của hệ thống truyền tin.

### 9.2.2 XÁC SUẤT LỖI

Trong một kênh AWGN, xác suất lỗi đối với một hệ thống trai phỏ DS áp dụng điều chế PSK nhị phân thì hoàn toàn giống như xác suất lỗi đối với PSK nhị phân truyền thống (không trai phỏ), nghĩa là

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (9.2.11)$$

Về một phương diện khác, nếu nhiễu là một tín hiệu hình sin được cho bởi (9.2.9) với công suất  $P_J$  thì xác suất lỗi xấp xỉ là

$$P_b = Q\left(\sqrt{\frac{2E_b}{P_J/W}}\right) = Q\left(\sqrt{\frac{2E_b}{J_0}}\right) \quad (9.2.12)$$

Như vậy, công suất nhiễu bị giảm bởi yếu tố độ rộng băng  $W$  của tín hiệu trai phỏ. Trong trường hợp này, chúng ta đã bỏ qua AWGN được giả thiết là bỏ qua được - nghĩa là,  $N_0 << P_J/W$ . Nếu chúng ta tính đến cả AWGN trên kênh thì xác suất lỗi biểu diễn được theo

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0 + P_J/W}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0 + J_0}}\right) \quad (9.2.13)$$

#### *Độ đứt chống nhiễu cốt ý*

Khi tín hiệu nhiễu là một tín hiệu gây nhiễu cốt ý của đối phương, chúng ta có thể biểu diễn  $E_b/J_0$  theo

$$\frac{E_b}{J_0} = \frac{P_s T_b}{P_J / W} = \frac{P_s / R}{P_J / W} = \frac{W / R}{P_J / P_s} \quad (9.2.14)$$

Bây giờ, giả sử chúng ta cần định một  $E_b/J_0$  yêu cầu để có được một chất lượng liên lạc mong muốn. Khi đó, bằng cách sử dụng một thang logarit chúng ta có thể biểu diễn (9.2.14) theo

$$10 \log \frac{P_J}{P_s} = 10 \log \frac{W}{R} - 10 \log \frac{E_b}{J_0} \left( \frac{P_J}{P_s} \right)_{\text{dB}} = \left( \frac{W}{R} \right)_{\text{dB}} - \left( \frac{E_b}{J_0} \right)_{\text{dB}} \quad (9.2.15)$$

Tỷ số  $(P_J/P_s)_{\text{dB}}$  được gọi là độ dự trữ chống nhiễu cốt ý. Đây là ưu thế công suất tương đối mà kẻ gây nhiễu cốt ý phải có để có thể làm gián đoạn việc liên lạc của hệ thống.

### BÀI TẬP MINH HỌA

**Bài tập minh họa 9.1 [Tăng ích xử lý và độ dự trữ nhiễu cốt ý]** Giả sử rằng chúng ta cần  $E_b/J_0 = 10$  dB để có được việc truyền tin tin cậy với PSK nhị phân. Hãy xác định tăng ích xử lý cần thiết để có được một lượng dự trữ nhiễu cốt ý 20 dB.

#### Lời giải

Bằng cách sử dụng (9.2.15), chúng ta tìm ra được tăng ích xử lý  $(W/R)_{\text{dB}} = 30$  dB, nghĩa là  $W/R = L_c = 1000$ . Điều đó có nghĩa là với  $W/R = 1000$  thì công suất nhiễu cốt ý trung bình tại máy thu có thể lớn đến 100 lần công suất  $P_s$  của tín hiệu mong muốn mà ta vẫn duy trì được việc truyền tin tin cậy.

#### Chất lượng của các tín hiệu trải phổ có mã hoá

Như đã nêu trong chương 8, khi thông tin được truyền được mã hoá bằng một mã tuyến tính nhị phân (mã khối hay mã chập) thì SNR tại lối ra của một bộ giải mã quyết định mềm được tăng lên bởi tăng ích mã được xác định theo

$$\text{Tăng ích mã (CG)} = R_c d_{\min}^H \quad (9.2.16)$$

trong đó  $R_c$  là tốc độ mã còn  $d_{\min}^H$  là khoảng cách Hamming nhỏ nhất của mã. Do vậy, tác động của việc mã hoá là làm tăng lượng dự trữ chống nhiễu cốt ý

lên một lượng bằng tăng ích mã. Như thế, (9.2.15) có thể sửa đổi thành

$$\left(\frac{P_J}{P_s}\right)_{dB} = \left(\frac{W}{R}\right)_{dB} + (CG)_{dB} - \left(\frac{E_b}{J_0}\right)_{dB} \quad (9.2.17)$$

### 9.2.3 HAI ỨNG DỤNG CỦA CÁC TÍN HIỆU TRẢI PHỔ DS

Trong mục nhỏ này, chúng tôi sẽ mô tả một cách ngắn gọn việc sử dụng các tín hiệu trải phổ DS trong hai ứng dụng. Trước tiên chúng ta sẽ xem xét một ứng dụng, trong đó tín hiệu được phát đi với công suất rất thấp đến nỗi một người thu nghe sẽ gặp phải khó khăn rất lớn trong việc cố phát hiện xem có tín hiệu hay không. Ứng dụng thứ hai là thông tin vô tuyến đa truy nhập.

#### *Việc truyền dẫn tín hiệu có khả năng phát hiện thấp*

Trong ứng dụng này, tín hiệu mang thông tin được truyền với một mức công suất rất thấp so với tạp âm nền của kênh và tạp âm nhiệt phát sinh trong máy thu đầu cuối. Nếu tín hiệu trải phổ DS chiếm một độ rộng băng  $W$  và mật độ phổ công suất của tạp âm cộng là  $N_0$  ( $W/Hz$ ) thì công suất tạp âm trung bình trong độ rộng băng  $W$  sẽ là  $P_N = WN_0$ .

Công suất tín hiệu thu được trung bình tại máy thu đã định trước là  $P_R$ . Nếu chúng ta muốn che dấu sự hiện diện của tín hiệu này đối với các máy thu hiện đang có mặt ở vùng lân cận với máy thu đã định thì tín hiệu phải được phát đi ở một mức công suất sao cho  $P_R/P_N \ll 1$ . Máy thu đã định có thể khôi phục lại được tín hiệu mang tin yếu từ tạp âm nền với sự trợ giúp của tăng ích xử lý và tăng ích mã. Bất kỳ máy thu nào khác không biết gì về chuỗi mã PN lại không có khả năng có được tăng ích xử lý và tăng ích mã. Hệ quả là việc tín hiệu mang tin đang có hay đang không có rất khó mà phát hiện ra được. Chúng ta nói rằng tín hiệu đã phát có xác suất bị chặn thu thấp (LPI: *Low Probability of being Intercepted*) và nó được gọi là một tín hiệu LPI.

Xác suất lỗi đã được cho trong mục 9.2.2 cũng áp dụng được cho việc giải điều chế và giải mã của các tín hiệu LPI tại máy thu đã định.

## BÀI TẬP MINH HỌA

**Bài tập minh họa 9.2 [Thiết kế hệ thống trai phổ DS]** Một hệ thống trai phổ DS cần được thiết kế sao cho tỷ số công suất tại máy thu đã định là  $P_R/P_N = 0.01$  đối với một kênh AWGN. Giá trị mong muốn  $E_b/N_0$  là nhằm có chất lượng chấp nhận được. Hãy xác định giá trị cực tiểu của tăng ích xử lý cần thiết để đạt được một tỷ số  $E_b/N_0 = 10$ .

### Lời giải

Chúng ta có thể viết  $E_b/N_0$  theo

$$\frac{E_b}{N_0} = \frac{P_R T_b}{N_0} = \frac{P_R L_c T_c}{N_0} = \left( \frac{P_R}{W N_0} \right) L_c = \left( \frac{P_R}{P_N} \right) L_c \quad (9.2.18)$$

Do  $E_b/N_0 = 10$  còn  $P_R/P_N = 10^{-2}$  nên tăng ích xử lý cần thiết là  $L_c = 1000$ .

### Đa truy nhập phân chia theo mã

Sự cải thiện về chất lượng nhận được nhờ tín hiệu trai phổ DS thông qua tăng ích xử lý và tăng ích mã có thể sử dụng được để nhiều tín hiệu trai phổ DS có thể chiếm chung một độ rộng băng kênh, miễn là mỗi một tín hiệu có một chuỗi giả tạp (một chữ ký) riêng. Như thế, có thể có một số người sử dụng đồng thời truyền các bản tin trên cùng một độ rộng băng kênh. Kiểu thông tin này, trong đó mỗi một cặp người sử dụng máy thu/máy phát có mã chữ ký riêng của mình để truyền tin trên một độ rộng băng kênh chung, được gọi là đa truy nhập phân chia theo mã (CDMA: *Code Division Multiple Access*).

Trong thông tin tế bào số, một trạm gốc truyền các tín hiệu tới  $N_u$  máy thu di động bằng cách sử dụng  $N_u$  chuỗi PN trực giao, mỗi một mã dành cho một máy thu đã định.  $N_u$  tín hiệu này được truyền hoàn toàn đồng bộ sao cho chúng tới từng máy thu di động một cách đồng bộ. Hệ quả là, do tính trực giao của  $N_u$  chuỗi PN, mỗi một máy thu đã định có thể giải điều chế tín hiệu của riêng mình mà không bị nhiễu từ các tín hiệu đã phát khác đang cùng chia sẻ độ rộng băng. Tuy nhiên, loại đồng bộ như vậy không thể duy trì được đối với các tín hiệu được

phát đi từ các máy phát di động tới trạm gốc (đường lên hay đường về). Khi giải điều chế từng tín hiệu trải phổ DS tại trạm gốc, các tín hiệu từ các người sử dụng đồng thời khác nhau xuất hiện như nhiễu cộng. Ta hãy xác định số các tín hiệu đồng thời có thể hoà được trong một hệ thống CDMA. Ta giả sử rằng tất cả các tín hiệu đều có công suất trung bình như nhau tại trạm gốc. Trong nhiều hệ thống thực tế, mức công suất tín hiệu thu được từ từng người sử dụng được giám sát tại trạm gốc và điều khiển công suất được thực hiện đối với mọi người sử dụng đồng thời bằng cách sử dụng một kênh điều khiển nhằm chỉ dẫn cho các người sử dụng để tăng hay giảm mức công suất của họ. Với điều khiển công suất như thế, nếu có  $N_u$  người sử dụng đồng thời thì tỷ số công suất tín hiệu trên tạp nhiễu tại một máy thu đã cho là

$$\frac{P_s}{P_N} = \frac{P_s}{(N_u - 1)P_s} = \frac{1}{N_u - 1} \quad (9.2.19)$$

Từ mối quan hệ này, chúng ta có thể xác định được số người sử dụng có thể hoà đồng thời.

Khi xác định số người sử dụng cực đại của kênh, chúng ta đã giả thiết cho đơn giản rằng các chuỗi mã PN được sử dụng bởi những người sử dụng khác nhau thì trực giao với nhau còn nhiễu từ các người sử dụng khác thì được cộng lại chỉ trên một cơ sở công suất. Tuy nhiên, tính trực giao của các chuỗi giả ngẫu nhiên giữa  $N_u$  người sử dụng thì rất khó mà đạt được, đặc biệt khi  $N_u$  lớn. Sự thực thì việc thiết kế một tập lớn các chuỗi giả ngẫu nhiên với các tính chất tương quan tốt là một vấn đề quan trọng thu hút sự chú ý đáng kể trong các tài liệu kỹ thuật. Chúng ta sẽ đề cập một cách ngắn gọn tới vấn đề này trong mục 9.3.

### BÀI TẬP MINH HỌA

**Bài tập minh họa 9.3 [Số người sử dụng cực đại trong CDMA]** Giả sử rằng mức chất lượng mong muốn đối với một người sử dụng trong một hệ thống CDMA đạt được khi  $E_b/J_0 = 10$ . Hãy xác định số người sử dụng đồng thời cực đại có thể hoà được trong hệ thống CDMA nếu tỷ số độ rộng băng trên tốc độ bít là 100 và tăng ích mã là 6 dB.

### Lời giải

Từ quan hệ cơ sở được cho trong (9.2.17), chúng ta có

$$\left( \frac{P_N}{P_S} \right)_{dB} = \left( \frac{W}{R} \right)_{dB} + (CG)_{dB} - \left( \frac{E_b}{J_0} \right)_{dB} = 20 + 6 - 10 = 16 \text{ dB}$$

Hệ quả là,

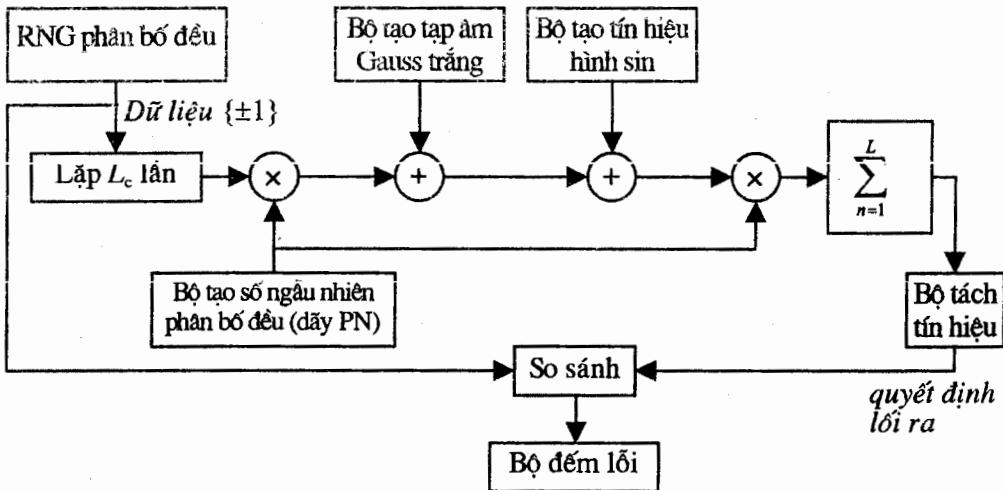
$$\frac{1}{N_u - 1} = \frac{P_S}{P_N} = \frac{1}{40}$$

và vì vậy

$$N_u = 41 \text{ người sử dụng.}$$

### BÀI TẬP MINH HỌA

**Bài tập minh họa 9.4 [Mô phỏng trải phổ DS]** Mục tiêu của bài tập minh họa này là biểu diễn tính hiệu quả của một tín hiệu trải phổ DS trong việc nén một nhiễu hình sin thông qua mô phỏng Monte-Carlo. Sơ đồ khối của hệ thống dùng cho mô phỏng được minh họa trên hình vẽ 9.5.



Hình 9.5 Mô hình hệ thống trải phổ DS dùng cho mô phỏng Monte-Carlo

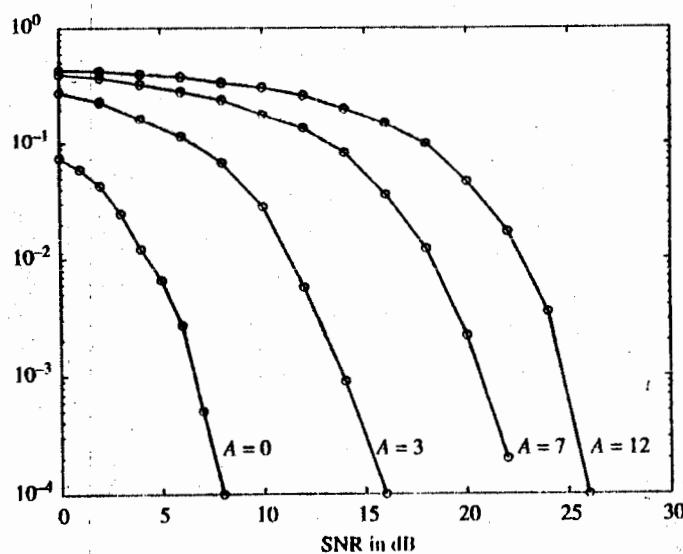
### Lời giải

Một bộ tạo số ngẫu nhiên (RNG: Random Number Generator) phân bố đều được sử dụng để tạo ra một chuỗi *symbol* thông tin nhị phân ( $\pm 1$ ). Mỗi một bit

thông tin được lắp  $L_c$  lần, trong đó  $L_c$  tương ứng với số chip trong một bít thông tin. Chuỗi thu được, gồm  $L_c$  lần lắp trên một bít, được nhân với một chuỗi PN  $c(n)$  được tạo bởi một bộ tạo số ngẫu nhiên phân bố đều khác. Ta cộng chuỗi tích đó với tạp âm Gauss trắng có phương sai  $\sigma^2 = N_0/2$  và nhiều hình sin có dạng

$$i(n) = A \sin \omega_0 n$$

trong đó  $0 < \omega_0 < \pi$  còn biên độ của hình sin được chọn thỏa mãn  $A < L_c$ . Bộ giải điều chế thực hiện tính tương quan chéo với chuỗi PN và tính tổng (lấy tích phân) các khối  $L_c$  mẫu tín hiệu chứa trong mỗi bít thông tin. Lối ra của bộ tổng được cấp đến bộ tách tín hiệu mà nó so sánh tín hiệu này với ngưỡng bằng 0 và quyết định xem bít đã phát đi là +1 hay -1. Bộ đếm lối đếm số các lối do bộ tách tín hiệu tạo nên. Các kết quả của mô phỏng Monte-Carlo được thể hiện trên hình 9.6 cho ba giá trị khác nhau của biên độ nhiễu hình sin, với  $L_c = 20$ . Trên hình 9.6 cũng thể hiện tỷ lệ lối đo được khi nhiễu hình sin được loại đi. Phương sai của tạp âm cộng được giữ cố định trong các mô phỏng này, còn mức của tín hiệu mong muốn thì được thay đổi tỷ lệ để đạt cho được SNR mong muốn đối với từng lần chạy mô phỏng.



Hình 9.6 Chất lượng tỷ lệ lối của hệ thống nhận được từ mô phỏng Monte-Carlo

Các MATLAB script dùng cho chương trình mô phỏng được cho dưới đây.

### m-file

```
% MATLAB script for Illustrative Problem 4, Chapter 9.  
echo on  
Lc=20; % Number of chips per bit  
A1=3; % amplitude of the first sinusoidal interference  
A2=7; % amplitude of the second sinusoidal interference  
A3=12; % amplitude of the third sinusoidal interference  
A4=0; % fourth case: no interference  
w0=1; % frequency of the sinusoidal interference in radians  
SNRindB=0:2:30;  
for i=1:length(SNRindB),  
    % measured error rates  
    smld_err_prb1(i)=ss_Pe94(SNRindB(i),Lc,A1,w0);  
    smld_err_prb2(i)=ss_Pe94(SNRindB(i),Lc,A2,w0);  
    smld_err_prb3(i)=ss_Pe94(SNRindB(i),Lc,A3,w0);  
    echo off ;  
end;  
echo on ;  
SNRindB4=0:1:8;  
for i=1:length(SNRindB4),  
    % measured error rate when there is no interference  
    smld_err_prb4(i)=ss_Pe94(SNRindB4(i),Lc,A4,w0);  
    echo off ;  
end;  
echo on ;  
% Plotting commands follow
```

### m-file

```
function [p]=ss_Pe94(snr_in_dB, Lc, A, w0)  
% [p]=ss_Pe94(snr_in_dB, Lc, A, w0)  
% SS PE94 finds the measured error rate. The function  
% that returns the measured probability of error for the given value of  
% the snr_in_dB, Lc, A and w0.  
snr=10^(snr_in_dB/10);  
sgma=1; % noise standard deviation is fixed  
Eb=2*sgma^2*snr; % signal level required to achieve the given  
% signal to noise ratio  
E_chip=Eb/Lc; % energy per chip  
N=10000; % number of bits transmitted
```

```

% The generation of the data, noise, interference, decoding process and error
% counting is performed all together in order to decrease the run time of the
% program. This is accomplished by avoiding very large sized vectors.
num_of_err=0;
for i=1:N,
    % generate the next data bit
    temp=rand;
    if (temp<0.5),
        data=-1;
    else
        data=1;
    end;
    % repeat it Lc times, i.e. divide it into chips
    for j=1:Lc,
        repeated_data(j)=data;
    end;
    % pn sequence for the duration of the bit is generated next
    for j=1:Lc,
        temp=rand;
        if (temp<0.5),
            pn_seq(j)=-1;
        else
            pn_seq(j)=1;
        end;
    end;
    % the transmitted signal is
    trans_sig=sqrt(E_chip)*repeated_data.*pn_seq;
    % AWGN with variance sigma^2
    noise=sgma*randn(1,Lc);
    % interference
    n=(i-1)*Lc+1:i*Lc;
    interference=A*sin(w0*n);
    % received signal
    rec_sig=trans_sig+noise+interference;
    % determine the decision variable from the received signal
    temp=rec_sig.*pn_seq;
    decision_variable=sum(temp);
    % making decision
    if (decision_variable<0),
        decision=-1;
    else
        decision=1;
    end;
end;

```

```

end;
% if it is an error, increment the error counter
if (decision~=data),
    num_of_err=num_of_err+1;
end;
end;
% then the measured error probability is
p=num_of_err/N;

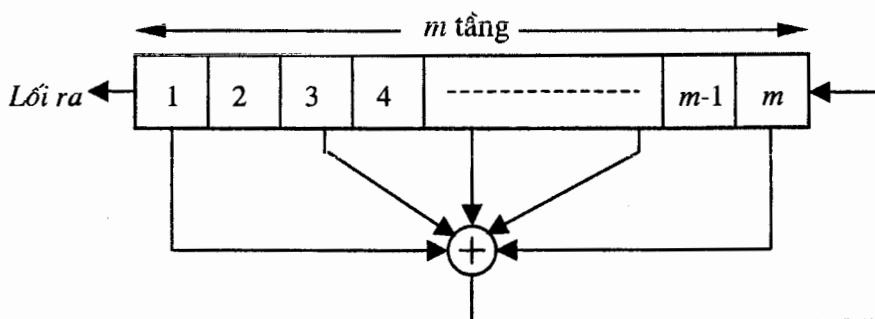
```

### 9.3 TẠO CÁC CHUỖI PN

Một chuỗi giả ngẫu nhiên, hay chuỗi PN, là một chuỗi mã gồm các con 0 và 1 mà hàm tự tương quan của nó có các tính chất tương tự như của tệp âm trắng. Trong mục này, chúng tôi mô tả một cách vắn tắt cấu trúc của một vài chuỗi PN và hàm tự tương quan cũng như các tính chất tương quan chéo của chúng.

Xin nói trước, hầu hết các chuỗi mã PN nhị phân đã được biết rộng rãi đều là các chuỗi ghi dịch độ dài cực đại. Một chuỗi ghi dịch độ dài cực đại, hay ngắn gọn là *chuỗi-m*, có độ dài  $L = 2^m - 1$  bit và được tạo ra bởi một bộ ghi dịch  $m$  tầng có hồi tiếp như được minh họa trên hình 9.7. Chuỗi này tuần hoàn với chu kỳ  $L$ . Mỗi một chu kỳ có một chuỗi gồm  $2^{m-1}$  con một và  $2^{m-1} - 1$  con không. Bảng 9.1 liệt kê các kết nối bộ ghi dịch để tạo ra các chuỗi độ dài cực đại.

Trong các ứng dụng trải phổ DS, chuỗi nhị phân với các phần tử  $\{0, 1\}$  được ánh xạ thành một chuỗi nhị phân tương ứng với các phần tử  $\{1, -1\}$ . Chúng ta sẽ gọi chuỗi tương đương  $\{c_n\}$  với các phần tử  $\{1, -1\}$  là một *chuỗi lưỡng cực*.



Hình 9.7 Bộ ghi dịch  $m$  tầng có hồi tiếp tổng quát

Bảng 9.1 Các kết nối bộ ghi dịch để tạo các chuỗi độ dài cực đại

$m$	các tầng được nối tới bộ cộng modulo 2	$m$	các tầng được nối tới bộ cộng modulo 2	$m$	các tầng được nối tới bộ cộng modulo 2
2	1, 2	13	1, 10, 11, 13	24	1, 18, 23, 24
3	1, 3	14	1, 5, 9, 14	25	1, 23
4	1, 4	15	1, 15	26	1, 21, 25, 26
5	1, 4	16	1, 5, 14, 16	27	1, 23, 26, 27
6	1, 6	17	1, 15	28	1, 26
7	1, 7	18	1, 12	29	1, 28
8	1, 5, 6, 7	19	1, 15, 18, 19	30	1, 8, 29, 30
9	1, 6	20	1, 18	31	1, 29
10	1, 8	21	1, 20	32	1, 11, 31, 32
11	1, 10	22	1, 22	33	1, 21
12	1, 7, 9, 12	23	1, 19	34	1, 8, 33, 34

Một đặc tính quan trọng của một chuỗi PN tuần hoàn là hàm tự tương quan của nó, thường được định nghĩa với các chuỗi lưỡng cực  $\{c_n\}$  theo

$$R_c(m) = \sum_{n=1}^L c_n c_{n+m}, \quad 0 \leq m \leq L-1 \quad (9.3.1)$$

trong đó  $L$  là chu kỳ của chuỗi. Do chuỗi  $\{c_n\}$  tuần hoàn với chu kỳ  $L$ , chuỗi tự tương quan  $\{R_c(m)\}$  cũng tuần hoàn với chu kỳ  $L$ .

Một cách lý tưởng, một chuỗi PN cần phải có một hàm tự tương quan có tính chất tương quan tương tự như của tệp âm trắng. Nghĩa là, chuỗi tự tương quan lý tưởng đối với  $\{c_n\}$  là  $R_c(0) = L$  và  $R_c(m) = 0$  đối với  $1 \leq m \leq L-1$ . Trong trường hợp các chuỗi- $m$ , chuỗi tự tương quan là

$$R_c(m) = \begin{cases} L, & m = 0 \\ -1, & 1 \leq m \leq L-1 \end{cases} \quad (9.3.2)$$

Đối với các chuỗi- $m$  dài, cỡ của các giá trị không phải đỉnh của  $R_c(m)$  so với giá trị đỉnh  $R_c(0)$ , nghĩa là tỷ lệ  $R_c(m)/R_c(0) = -1/L$ , thì nhỏ, chứ còn theo quan điểm thực tế thì chưa phải đã mỹ mãn. Do vậy, các chuỗi- $m$  thì rất gần với các chuỗi PN lý tưởng khi xét theo hàm tự tương quan của chúng.

Trong một số ứng dụng, các tính chất tương quan chéo của các chuỗi PN thì cũng quan trọng như các tính chất tự tương quan. Thí dụ, trong CDMA mỗi một người sử dụng được gán cho một chuỗi PN riêng. Một cách lý tưởng, các chuỗi

PN giữa các người sử dụng phải trực giao cùng nhau sao cho mức nhiễu mà một người sử dụng phải chịu từ việc truyền tin của các người sử dụng khác thì bằng không. Tuy vậy, trong thực tế các chuỗi PN được sử dụng bởi các người sử dụng khác nhau lại tỏ ra có tương quan đôi chút.

Để cụ thể, chúng ta hãy xét lớp các chuỗi- $m$ . Ta đã biết rằng hàm tương quan chéo giữa một cặp chuỗi- $m$  có cùng chu kỳ có thể có các đỉnh khá lớn. Bảng 9.2 liệt kê các độ lớn đỉnh  $R_{max}$  của mức tự tương quan giữa các cặp chuỗi- $m$  đối với  $3 \leq m \leq 12$ . Số các chuỗi- $m$  có độ dài  $L = 2^m - 1$  đối với  $3 \leq m \leq 12$  cũng được liệt kê trong bảng 9.2. Chúng ta thấy rằng số các chuỗi- $m$  có độ dài  $L$  tăng theo  $m$  khá nhanh. Chúng ta cũng thấy rằng, đối với hầu hết các chuỗi, độ lớn đỉnh  $R_{max}$  của hàm tương quan chéo thì bằng một số phần trăm lớn của giá trị đỉnh của hàm tự tương quan. Hệ quả là các chuỗi- $m$  thì không thích hợp đối với các hệ thống thông tin CDMA. Mặc dù có thể chọn lấy được một tập con nhỏ của các chuỗi- $m$  có các giá trị tương quan chéo đỉnh tương đối nhỏ hơn  $R_{max}$ , số các chuỗi trong tập đó thường quá nhỏ đối với các ứng dụng CDMA.

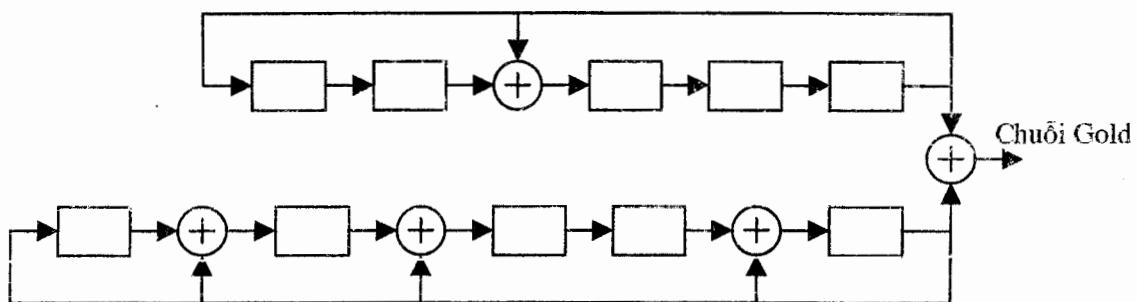
Bảng 9.2 Các giá trị tương quan chéo của các chuỗi- $m$  và các chuỗi Gold

$m$	$L=2^m-1$	chuỗi- $m$			chuỗi Gold	
		Số	$R_{max}$	$R_{max}/R(0)$	$R_{max}$	$R_{max}/R(0)$
3	7	2	5	0.71	5	0.71
4	15	2	9	0.60	9	0.60
5	31	6	11	0.35	9	0.29
6	63	6	23	0.36	17	0.27
7	127	18	41	0.32	17	0.13
8	255	16	95	0.37	33	0.13
9	511	48	113	0.22	33	0.06
10	1023	60	383	0.37	69	0.06
11	2047	176	287	0.14	69	0.03
12	4095	144	1407	0.34	129	0.03

Phương pháp tạo các chuỗi PN với các tính chất tương quan chéo tuần hoàn

tốt hơn các chuỗi- $m$  đã được Gold [5], [6] và Kasami [7] phát triển. Các chuỗi Gold được xây dựng bằng cách dùng một cặp chuỗi- $m$  được chọn lựa đặc biệt, được gọi là các chuỗi- $m$  ưa dùng hơn, rồi tạo tổng modulo 2 của hai chuỗi này đối với mỗi một trong  $L$  phiên bản dịch chu kỳ của một chuỗi với chuỗi kia. Như thế,  $L$  chuỗi Gold được tạo ra như được minh họa trên hình vẽ 9.8. Đối với  $L$  lớn và  $m$  là số lẻ, giá trị cực đại của hàm tương quan chéo giữa một cặp chuỗi Gold bất kỳ là  $R_{max} = \sqrt{2L}$ . Đối với  $m$  chẵn,  $R_{max} = \sqrt{L}$ .

Kasami [7] đã mô tả một phương pháp xây dựng các chuỗi PN bằng cách chiết một chuỗi- $m$ . Trong phương pháp của Kasami, mọi bít thứ  $(2^{m/2}+1)$  của một chuỗi- $m$  được chọn lấy. Phương pháp xây dựng này thu được một tập nhỏ hơn các chuỗi PN so với các chuỗi Gold song giá trị tương quan chéo cực đại của chúng là  $R_{max} = \sqrt{L}$ .



Hình 9.8 Tạo các chuỗi Gold có độ dài 31

Việc so sánh giá trị đỉnh của hàm tương quan chéo đối với các chuỗi Gold và các chuỗi Kasami là đáng quan tâm, bằng cách sử dụng một giới hạn chặn dưới đã biết đối với trị tương quan chéo cực đại giữa một cặp chuỗi nhị phân bất kỳ có độ dài  $L$ . Với một tập đã cho gồm  $N$  chuỗi có chu kỳ  $L$ , một giới hạn chặn dưới đối với trị tương quan chéo cực đại của chúng là

$$R_{max} \geq L \sqrt{\frac{N-1}{NL-1}} \quad (9.3.3)$$

mà nó, đối với các giá trị  $L$  và  $N$  lớn, xấp xỉ tốt được theo  $R_{max} \geq \sqrt{L}$ . Do đó, chúng ta thấy được rằng các chuỗi Kasami thoả mãn giới hạn chặn dưới, và do

vậy chúng là tối ưu. Trong khi đó, các chuỗi Gold với  $m$  lẻ thì có  $R_{max} = \sqrt{2L}$ . Do vậy, chúng là kém tối ưu đôi chút.

Bên cạnh các chuỗi Gold và Kasami nổi tiếng, còn có các chuỗi nhị phân khác thích hợp với các ứng dụng CDMA. Bạn đọc nào muốn biết thêm xin tham khảo các bài báo của Scholtz [8] và của Sarwate và Pursley [9].

### BÀI TẬP MINH HỌA

**Bài tập minh họa 9.5 [Tạo chuỗi Gold]** Hãy tạo các chuỗi Gold có  $L = 31$  mà chúng có được nhờ lấy tổng modulo 2 các tín hiệu lõi ra của hai bộ ghi dịch đã được thể hiện trên hình 9.8.

Bảng 9.3 Bảng các chuỗi Gold trong bài tập minh họa 9.5

0111000010000110010010111100000
0101101110001000001001000110001
000011011001010011110110010011
1010000110101101010001011010111
1111100111011110001110001011110
0100100100111000110000101001100
0010100011110101001101101101001
1110101101101110110111100100011
0110110001011001000011110110110
0110001000110110101011010011101
0111111011101001111010011001011
010001110101011011000001100111
0011010000101010011100100111111
1101001011010000010101110001111
0001111100100100000111011101110
1000010011001100100010000101101
1011001100011101101000110101010
110111001011111111101010100100
000000111111011010110010111000
101111010111001000000010000001
1100000001100000101100011110010
0011101001000101110100000010100
1100111000001111000100111011001
0010011010011010100101001000010
1111011110110001100110101110101
010101011100111100001100011010
000100010100101110111111000101
1001100000010011110011001111011
1000101010100011001010100000110
10101111100001011100111111100
11100101000000010111100001000

## Lời giải

Các MATLAB *script* để thực hiện tính toán này được cho dưới đây. 31 chuỗi tạo được được trình bày trong bảng 9.3. Giá trị trị tương quan chéo cực đại đối với các chuỗi này là  $R_{max} = 9$ .

### m-file

```
% MATLAB script for Illustrative Problem 5, Chapter 9.
echo on
% first determine the maximal length shift register sequences
% We'll take the initial shift register content as "00001".
connections1=[1 0 1 0 0];
connections2=[1 1 1 0 1];
sequence1=ss_mlsrs(connections1);
sequence2=ss_mlsrs(connections2);
% cyclically shift the second sequence and add it to the first one
L=2^length(connections1)-1;;
for shift_amount=0:L-1,
    temp=[sequence2(shift_amount+1:L) sequence2(1:shift_amount)];
    gold_seq(shift_amount+1,:)=(sequence1+temp)
    floor((sequence1+temp)./2).*2;
    echo off ;
end;
echo on ;
% find the max value of the cross correlation for these sequences
max_cross_corr=0;
for i=1:L-1,
    for j=i+1:L,
        % equivalent sequences
        c1=2*gold_seq(i,:)-1;
        c2=2*gold_seq(j,:)-1;
        for m=0:L-1,
            shifted_c2=[c2(m+1:L) c2(1:m)];
            corr=abs(sum(c1.*shifted_c2));
            if (corr>max_cross_corr),
                max_cross_corr=corr;
            end;
            echo off ;
        end;
        end;
    end;
% note that max_cross_corr turns out to be 9 in this example...
```

**m-file**

```

function [seq]=ss_mlsrs(connections);
% [seq]=ss_mlsrs(connections)
%   SS_MLSRS generates the maximal length shift register sequence when the
%   shift register connections are given as input to the function. A "zero"
%   means not connected, whereas a "one" represents a connection.
m=length(connections);
L=2^m-1;                                % length of the shift register sequence requested
registers=[zeros(1,m-1) 1];                % initial register contents
seq(1)=registers(m);                     % first element of the sequence
for i=2:L,
    new_reg_cont(1)=connections(1)*seq(i-1);
    for j=2:m,
        new_reg_cont(j)=registers(j-1)+connections(j)*seq(i-1);
    end;
    registers=new_reg_cont;               % current register contents
    seq(i)=registers(m);                 % the next element of the sequence
end;

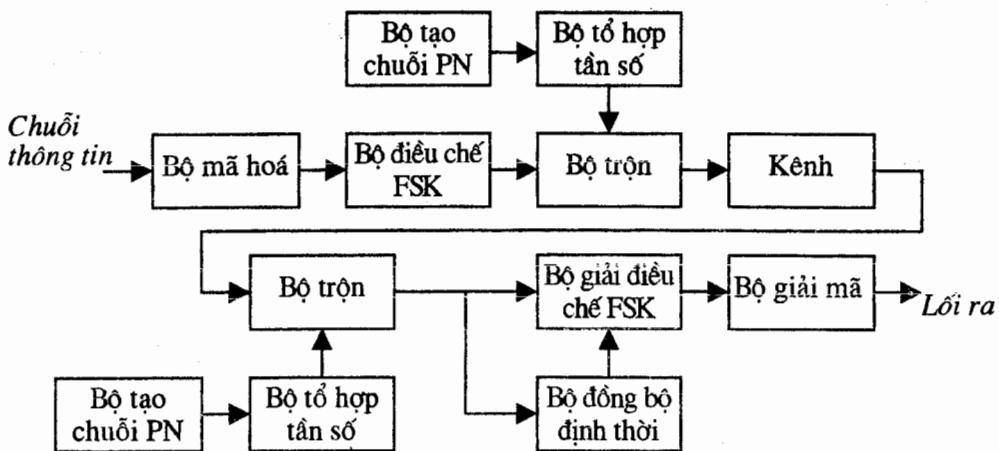
```

**9.4 TRẢI PHỔ NHảy TẦN**

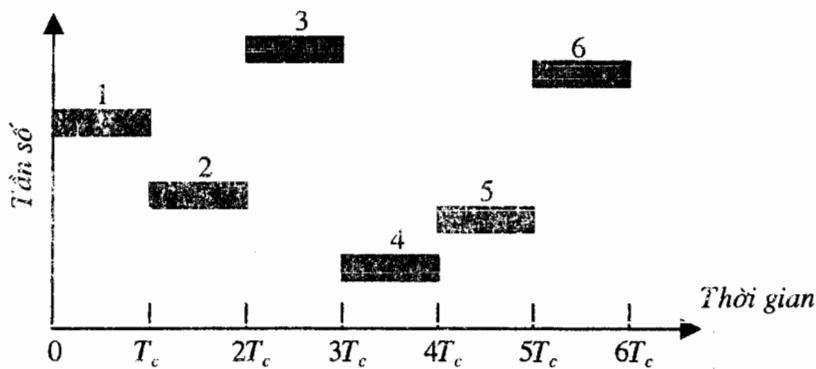
Trong trải phổ nhảy tần (FH), độ rộng băng kênh sẵn có  $W$  được chia nhỏ thành một số lớn các khe tần số không lấn lên nhau. Trong bất kỳ khoảng thời gian truyền tin nào, tín hiệu được truyền đi đều chiếm một (hay nhiều hơn một) khe tần số nói trên. Việc chọn khe (hay các khe) tần số nào trong mỗi một khoảng thời gian truyền tín hiệu được thực hiện một cách giả ngẫu nhiên theo tín hiệu lối ra của một bộ tạo chuỗi PN.

Một sơ đồ khái của máy phát và máy thu đối với một hệ thống trải phổ FH được trình bày trên hình vẽ 9.9. Việc điều chế thì hoặc là FSK nhị phân hay FSK  $M$  mức (MFSK). Thí dụ, nếu FSK nhị phân được áp dụng, bộ điều chế chọn một trong hai tần số, chẳng hạn như  $f_0$  hay  $f_1$ , tương ứng với việc truyền một con 0 hay một con 1. Tín hiệu FSK nhị phân có được đó được chuyển dịch về tần số một lượng được quy định bởi chuỗi lối ra lấy từ một bộ tạo chuỗi PN, bộ tạo PN này được sử dụng để chọn một tần số  $f_c$  được tổng hợp bằng một bộ tổ hợp tần số. Tín hiệu đã được dịch chuyển tần số này được truyền qua kênh. Thí dụ, bằng cách lấy

$m$  bít từ bộ tạo chuỗi PN chúng ta có thể ấn định  $2^m - 1$  tần số sóng mang có thể có. Hình vẽ 9.10 minh họa một mẫu tín hiệu FH.



Hình 9.9 Sơ đồ khối của một hệ thống trai phổ FH



Hình 9.10 Một thí dụ của một mẫu FH

Tại máy thu, có một bộ tạo chuỗi PN y như vậy (được đồng bộ với tín hiệu thu được) được sử dụng để điều khiển lối ra của bộ tổ hợp tần số. Như thế, sự chuyển dịch tần số giả ngẫu nhiên đã được đưa vào ở máy phát sẽ được loại bỏ tại bộ giải điều chế nhờ việc trộn tín hiệu lối ra của bộ tổ hợp tần số với tín hiệu thu được. Tín hiệu sản phẩm của việc trộn sau đó được giải điều chế bằng một bộ giải điều chế FSK. Một tín hiệu dùng cho việc duy trì đồng bộ bộ tạo chuỗi PN với tín hiệu FH thu được thường được tách từ tín hiệu thu được.

Mặc dù điều chế PSK nhị phân nói chung cho chất lượng tốt hơn điều chế FSK nhị phân, song khó mà duy trì được sự kết hợp về pha trong quá trình tổng

hợp các tần số được sử dụng trong mẫu nhảy tần và cả trong quá trình truyền lan tín hiệu qua kênh, do tín hiệu được nhảy từ tần số này sang tần số khác trên một giải tần rộng. Hệ quả là điều chế FSK với giải điều chế không kết hợp thường được áp dụng trong các hệ thống trải phổ FH.

Tốc độ nhảy tần, được ký hiệu là  $R_h$ , có thể được lựa chọn bằng hay thấp hơn, hoặc cao hơn tốc độ *symbol*. Nếu  $R_h$  bằng với hay thấp hơn tốc độ *symbol*, hệ thống FH được gọi là một hệ thống nhảy tần chậm. Nếu  $R_h$  cao hơn tốc độ *symbol*, nghĩa là có nhiều bước nhảy trên một *symbol*, thì hệ thống FH được gọi là một hệ thống nhảy tần nhanh. Chúng ta sẽ chỉ xét một tốc độ nhảy tần bằng với tốc độ *symbol*.

#### 9.4.1 XÁC SUẤT LỖI ĐỐI VỚI CÁC TÍN HIỆU FH

Ta hãy xét một hệ thống FH trong đó điều chế FSK nhị phân được sử dụng để truyền thông tin số. Tốc độ nhảy tần là một bước nhảy trên một bít. Việc giải điều chế và tách tín hiệu là không kết hợp. Trên một kênh AWGN, xác suất lỗi của một hệ thống như thế là

$$P_b = \frac{1}{2} e^{-E_b/2N_0} \quad (9.4.1)$$

Cùng một kết quả như thế được áp dụng nếu nhiều là một tín hiệu băng rộng hay là một nhiễu cố ý với phổ băng phẳng che phủ toàn bộ băng nhảy tần rộng  $W$ . Trong trường hợp như thế,  $N_0$  được thay thế bằng  $N_0+J_0$ , trong đó  $J_0$  là mật độ phổ nhiễu.

Cũng như trong trường hợp một hệ thống trải phổ DS chúng ta thấy rằng  $E_b$ , là năng lượng trên một bít, có thể biểu diễn được theo  $E_b = P_s T_b = P_s / R$ , trong đó  $P_s$  là công suất tín hiệu trung bình còn  $R$  là tốc độ bít. Tương tự như thế,  $J_0 = P_J / W$ , trong đó  $P_J$  là công suất trung bình của nhiễu băng rộng còn  $W$  là độ rộng băng kênh đã có. Do đó, giả sử rằng  $J_0 \gg N_0$ , chúng ta có thể biểu diễn SNR theo

$$\frac{E_b}{J_0} = \frac{W / R}{P_J / P_s} \quad (9.4.2)$$

trong đó  $W/R$  là tăng ích xử lý còn  $P_J/P_S$  là lượng dự trữ chống nhiễu cố ý của tín hiệu trải phổ FH.

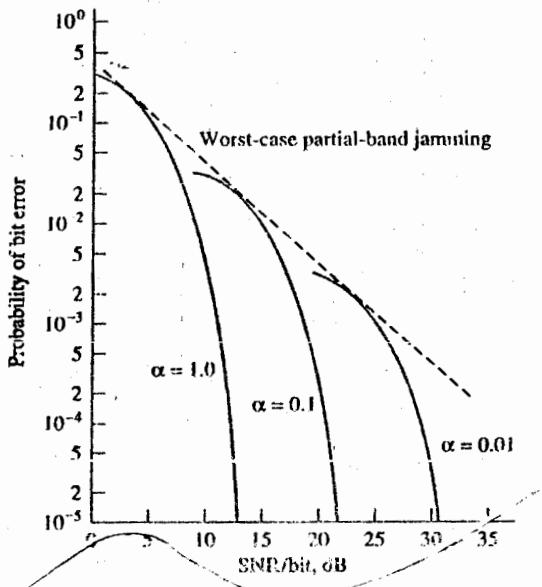
Các hệ thống trải phổ FH chậm đặc biệt dễ bị tổn thương do một nhiễu một phần băng có thể sinh ra từ một hoạt động gây nhiễu cố ý hoặc trong các hệ thống FH CDMA. Cụ thể, ta hãy giả sử rằng nhiễu một phần băng mô hình hoá được như một quá trình ngẫu nhiên Gauss có kỳ vọng bằng không với một mật độ phổ công suất bằng phẳng trên một phần của độ rộng băng tổng cộng  $W$  và bằng không trên phần còn lại của giải tần số đó. Trong miền hay trong các miền mà tại đó mật độ phổ công suất (của nhiễu) khác không, giá trị của nó là  $S_J(f) = J_0/\alpha$ , trong đó  $0 < \alpha \leq 1$ . Nói một cách khác đi, công suất trung bình của nhiễu  $P_J$  được giả sử là hằng số, còn  $\alpha$  là tỷ lệ của phần băng tần mà nhiễu chiếm trên băng tần tổng cộng.

Giả sử rằng nhiễu một phần băng tới từ một trạm gây nhiễu cố ý mà người gây nhiễu chọn  $\alpha$  để tối ưu hoá tác động tới hệ thống truyền tin. Trong một hệ thống nhảy tần chậm không mã hoá với điều chế FSK, nhị phân và tách tín hiệu không kết hợp thì các tần số phát được chọn với xác suất như nhau trong băng tần số  $W$ . Hệ quả là, tín hiệu thu được sẽ bị gây nhiễu với xác suất  $\alpha$  và sẽ không bị gây nhiễu với xác suất  $1-\alpha$ . Khi nó bị gây nhiễu, xác suất lỗi là  $\frac{1}{2} \exp(-\alpha \rho_b / 2)$ , còn khi không bị gây nhiễu thì việc tách tín hiệu được giả sử là không có lỗi, ở đây  $\rho_b \equiv E_b/J_0$ . Do đó, xác suất lỗi trung bình là

$$P_2(\alpha) = \frac{\alpha}{2} \exp\left(-\frac{\alpha \rho_b}{2}\right) = \frac{\alpha}{2} \exp\left(-\frac{\alpha W / R}{2P_J / P_S}\right) \quad (9.4.3)$$

Hình 9.11 minh họa tỷ lệ lỗi như một hàm của SNR  $\rho_b$  đối với một số giá trị  $\alpha$ . Người gây nhiễu được giả sử là đã tối ưu hoá chiến lược gây nhiễu của mình bằng cách chọn  $\alpha$  nhằm cực đại hóa xác suất lỗi. Bằng cách lấy đạo hàm  $P_2(\alpha)$  và giải nhằm tìm giá trị  $\alpha$  làm cho  $P_2(\alpha)$  cực đại, chúng ta tìm ra được rằng lựa chọn tốt nhất về  $\alpha$  của người gây nhiễu là

$$\alpha^* = \begin{cases} 2/\rho_b, & \rho_b \geq 2 \\ 1, & \rho_b < 2 \end{cases} \quad (9.4.4)$$



Hình 9.11 Chất lượng của FSK nhị phân với nhiễu một phần băng

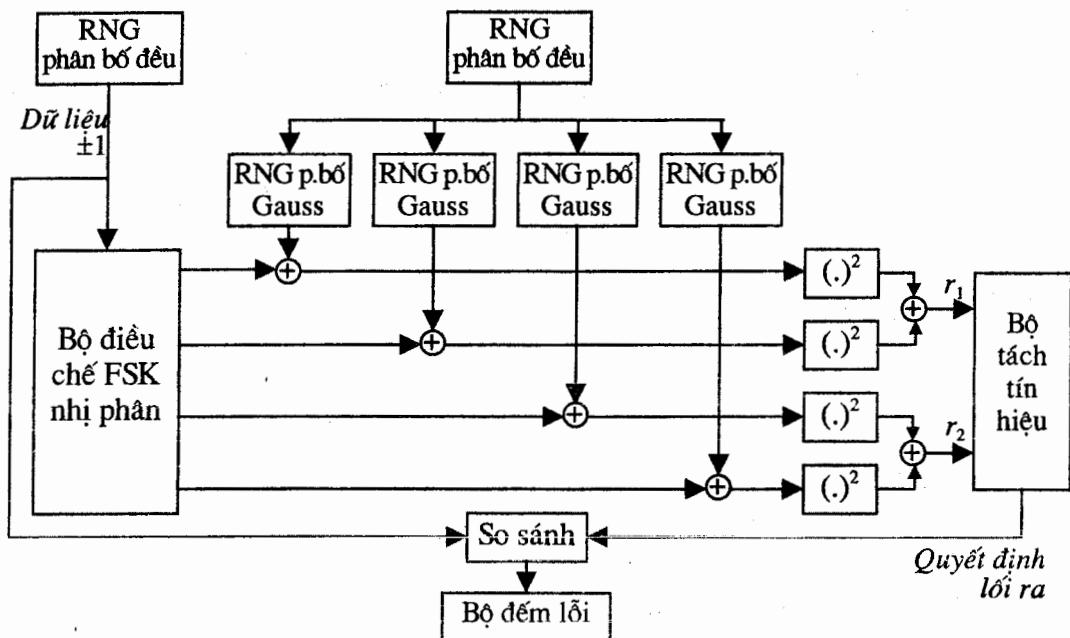
Xác suất lỗi tương ứng khi có người gây nhiễu một phần băng trong trường hợp xấu nhất là

$$P_2 = \begin{cases} e^{-1/\rho_b}, & \rho_b \geq 2 \\ \frac{1}{2}e^{-1/\rho_b}, & \rho_b < 2 \end{cases} \quad (9.4.5)$$

mà nó cũng được thể hiện trên hình vẽ 9.11. Trong khi xác suất lỗi giảm theo hàm mũ đối với nhiễu cố ý toàn băng, như đã cho trong (9.4.3), xác suất lỗi khi gây nhiễu cố ý một phần băng trong trường hợp xấu nhất chỉ giảm nghịch đảo theo  $E_b/J_0$ .

### BÀI TẬP MINH HỌA

**Bài tập minh họa 9.6 [Mô phỏng hệ thống FH]** Thông qua mô phỏng Monte-Carlo, hãy trình diễn chất lượng của một hệ thống thông tin số FH áp dụng FSK nhị phân và bị phá bởi một nhiễu cố ý một phần băng trong trường hợp xấu nhất. Sơ đồ khái của hệ thống để mô phỏng được trình bày trên hình vẽ 9.12.



Hình 9.12 Mô hình của một hệ thống FSK nhị phân có nhiều một phần băng dùng cho mô phỏng Monte-Carlo

### Lời giải

Một bộ tạo số ngẫu nhiên (RNG) phân bố đều được sử dụng để tạo một chuỗi thông tin nhị phân làm tín hiệu lối vào của bộ điều chế FSK. Tín hiệu lối ra của bộ điều chế FSK bị nhiễu loạn bởi tạp âm cộng Gauss với xác suất  $\alpha$ , trong đó  $0 < \alpha \leq 1$ . Một bộ tạo số ngẫu nhiên phân bố đều thứ hai được sử dụng để xác định khi nào tạp âm cộng Gauss gây nhiễu loạn tín hiệu và khi nào thì không gây nhiễu. Khi có tạp âm, giả sử rằng một bít 0 đã được phát đi, tín hiệu lối vào bộ tách tín hiệu là

$$r_1 = (\sqrt{E_b} \cos \phi + n_c)^2 + (\sqrt{E_b} \sin \phi + n_s)^2$$

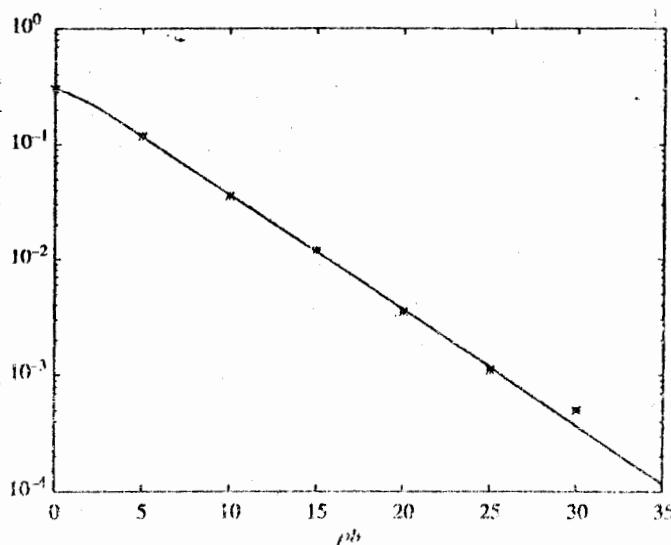
$$r_2 = n_{2c}^2 + n_{2s}^2$$

trong đó  $\phi$  biểu thị lượng dịch pha gây bởi kênh còn  $n_c$ ,  $n_s$ ,  $n_{2c}$ ,  $n_{2s}$  biểu thị các thành phần tạp âm cộng. Khi không có tạp âm, chúng ta có

$$r_1 = E_b, \quad r_2 = 0$$

và do vậy không có lối nào xảy ra với bộ tách tín hiệu. Phương sai của từng thành phần tạp âm là  $\sigma^2 = J_0/2\alpha$ , trong đó  $\alpha$  được cho theo (9.4.4). Để đơn giản, ta có

thể đặt  $\phi = 0$  và chuẩn hoá  $J_0$  bằng cách cho nó bằng 1. Khi đó  $\rho_b = E_b/J_0 = E_b$ . Do  $\sigma^2 = J_0/2\alpha$  còn  $\alpha = 2/\rho_b$ , nên khi có nhiều một phần băng,  $\sigma^2 = E_b/4$  và  $\alpha = 2/E_b$ , trong đó  $E_b$  bị ràng buộc theo  $E_b \geq 2$ . Hình vẽ 9.13 minh họa tỷ lệ lỗi có được từ mô phỏng Monte-Carlo. Giá trị lý thuyết của xác suất lỗi cho bởi (9.4.5) cũng được thể hiện trên hình vẽ này.



Hình 9.13 Chất lượng tỷ lệ lỗi của một hệ thống FH điều chế FSK  
nhị phân với nhiều một phần băng - mô phỏng Monte-Carlo

Các MATLAB *script* cho chương trình mô phỏng được cho dưới đây.

### m-file

```
% MATLAB script for Illustrative Problem 6, Chapter 9.
echo on
rho_b1=0:5:35; % rho in dB for the simulated error rate
rho_b2=0:0.1:35; % rho in dB for theoretical error rate computation
for i=1:length(rho_b1),
    smld_err_prb(i)=ss_pe96(rho_b1(i)); % simulated error rate
    echo off ;
end;
echo on ;
for i=1:length(rho_b2),
    temp=10^(rho_b2(i)/10);
    if (temp>2)
        theo_err_rate(i)=1/(exp(1)*temp); % theoretical error rate if rho>2
```

```

else
    theo_err_rate(i)=(1/2)*exp(-temp/2); % theoretical error rate if rho<2
end;
echo off ;
end;
echo on ;
% Plotting command follow

```

**m-file**

```

function [p]=ss_Pe96(rho_in_dB)
% [p]=ss_Pe96(rho_in_dB)
%     SS_PE96 finds the measured error rate. The value of signal per
%     interference ratio in dB is given as an input to the function.
rho=10^(rho_in_dB/10);
Eb=rho; % energy per bit
if (rho>2),
    alpha=2/rho; % optimal alpha if rho>2
else
    alpha=1; % optimal alpha if rho<2
end;
sgma=sqrt(1/(2*alpha)); % noise standard deviation
N=10000; % number of bits transmitted
% generation of the data sequence
for i=1:N,
    temp=rand;
    if (temp<0.5)
        data(i)=1;
    else
        data(i)=0;
    end;
end;
% find the received signals
for i=1:N,
    % the transmitted signal
    if (data(i)==0),
        r1c(i)=sqrt(Eb);
        r1s(i)=0;
        r2c(i)=0;
        r2s(i)=0;
    else
        r1c(i)=0;
        r1s(i)=0;
    end;
    % calculate the received signals
    r1c(i)=r1c(i)+sqrt(2*Eb)*randn();
    r1s(i)=r1s(i)+sqrt(2*Eb)*randn();
    r2c(i)=r2c(i)+sqrt(2*Eb)*randn();
    r2s(i)=r2s(i)+sqrt(2*Eb)*randn();
end;
% calculate the error rate
err_rate=0;
for i=1:N,
    if (data(i)~=r1c(i)>0.5),
        err_rate=err_rate+1;
    end;
end;
err_rate=err_rate/N;

```

```

r2c(i)=sqrt(Eb);
r2s(i)=0;
end;
% the received signal is found by adding noise with probability alpha
if (rand<alpha),
    r1c(i)=r1c(i)+gngauss(sgma);
    r1s(i)=r1s(i)+gngauss(sgma);
    r2c(i)=r2c(i)+gngauss(sgma);
    r2s(i)=r2s(i)+gngauss(sgma);
end;
end;
% make the decisions and count the number of errors made.
num_of_err=0;
for i=1:N,
    r1=r1c(i)^2+r1s(i)^2;           % first decision variable
    r2=r2c(i)^2+r2s(i)^2;           % second decision variable
    % decision is made next
    if (r1>r2),
        decis=0;
    else
        decis=1;
    end;
    % increment the counter if this is an error
    if (decis~=data(i)),
        num_of_err=num_of_err+1;
    end;
end;
% measured bit error rate is then
p=num_of_err/N;

```

#### 9.4.2 SỬ DỤNG PHÂN TẬP TÍN HIỆU ĐỂ KHẮC PHỤC NHIỀU MỘT PHẦN BĂNG

Chất lượng của hệ thống FH bị nhiễu bởi nhiều một phân băng như đã mô tả ở mục trước là rất tồi. Thí dụ, đối với hệ thống muốn đạt được một xác suất lỗi  $10^{-6}$  thì SNR yêu cầu tại bộ tách tín hiệu gần như phải bằng 60 dB khi có nhiễu trong trường hợp xấu nhất. Nếu so với trường hợp không có nhiễu một phân băng thì SNR yêu cầu đối với kênh AWGN chỉ là quãng 10 dB. Hệ quả là, thiệt hại về SNR gây bởi sự có mặt của nhiễu một phân băng sẽ là xấp xỉ 50 dB, quá lớn.

Phương pháp giảm tác động của nhiễu một phân băng lên hệ thống trai phỏ FH

là sử dụng phân tập tín hiệu; nghĩa là, cùng một bít thông tin được truyền đi trên các bước nhảy đa tần và các tín hiệu từ việc bội dẫn (*multiple transmission*) như thế sẽ được tính trọng số rồi tổ hợp lại với nhau tại lối vào bộ tách tín hiệu. Một cách chi tiết, giả sử rằng mỗi một bít thông tin được truyền đi trên hai bước nhảy tần liên tiếp. Hệ thống như thế được gọi là một *hệ thống phân tập kép (dual diversity system)*. Trong trường hợp này, giả sử một bít 0 được truyền đi thì hoặc là cả hai tín hiệu lối vào bộ tổ hợp tín hiệu đều bị nhiễu, hoặc là chỉ một trong hai tín hiệu đã truyền bị nhiễu, hoặc không một tín hiệu đã truyền nào bị nhiễu bởi nhiễu trên kênh.

Bộ tổ hợp tín hiệu được giả thiết là biết được mức nhiễu và do đó tạo ra các biến quyết định tổ hợp

$$\begin{aligned} x &= w_1 r_{11} + w_2 r_{12} \\ y &= w_1 r_{21} + w_2 r_{22} \end{aligned} \quad (9.4.6)$$

trong đó  $r_{11}, r_{21}$  là hai tín hiệu lối ra của thiết bị có đặc tuyến công tác theo luật bậc hai (*square-law device*: thiết bị luật bậc hai) đối với tín hiệu đã truyền thứ nhất, còn  $r_{12}, r_{22}$  là các tín hiệu lối ra của thiết bị luật bậc hai có được từ tín hiệu đã truyền thứ hai. Các trọng số  $w_1$  và  $w_2$  được đặt là  $1/\sigma^2$ , ở đây  $\sigma^2$  là phương sai của tạp âm cộng cộng với nhiễu. Do đó, khi  $\sigma^2$  lớn, như thể trong trường hợp có nhiễu, trọng số được tính cho tín hiệu thu được là nhỏ. Mặt khác, khi  $\sigma^2$  nhỏ trong trường hợp không có nhiễu thì trọng số được tính cho các thành phần tín hiệu thu được sẽ là lớn. Bằng cách đó bộ tổ hợp tín hiệu giải nhấn (*de-emphasize*) các thành phần tín hiệu bị nhiễu.

Hai thành phần  $x$  và  $y$  từ bộ tổ hợp được cấp tới bộ tách tín hiệu, bộ này sẽ ra quyết định theo thành phần tín hiệu nào lớn hơn.

Chất lượng của tín hiệu FH với phân tập kép khi này sẽ trội hẳn lên trong trường hợp khi cả hai tín hiệu cùng bị nhiễu. Tuy vậy, xác suất của sự kiện này thì tỷ lệ thuận với  $\alpha^2$ , nhỏ hơn  $\alpha$  một cách đáng kể. Hệ quả là, xác suất lỗi đối với nhiễu một phần bằng trường hợp xấu nhất sẽ có dạng

$$P_2(2) = \frac{K_2}{\rho_b^2}, \quad \rho_b > 2 \quad (9.4.7)$$

trong đó  $K_2$  là một hằng số còn  $\rho_b = E_b/J_0$ . Trong trường hợp này, xác suất lỗi đối với phân tập kép sẽ giảm nghịch đảo theo bình phương của SNR. Nói một cách khác, việc tăng SNR lên một lượng 10 dB sẽ dẫn đến việc giảm xác suất lỗi một trăm lần. Hệ quả là, một xác suất lỗi bằng  $10^{-6}$  có thể đạt được với một SNR vào quãng 30 dB khi có phân tập kép, so với 60 dB (1000 lần lớn hơn) khi không có phân tập kép.

Tổng quát hơn, nếu mỗi một bít thông tin được truyền trên  $D$  bước nhảy tần số, trong đó  $D$  là bậc của phân tập, thì xác suất lỗi có dạng

$$P_2(D) = \frac{K_D}{\rho_b^D}, \quad \rho_b > 2 \quad (9.4.8)$$

trong đó  $K_D$  là một hằng số.

Do phân tập tín hiệu như đã mô tả trên đây là một dạng hiển nhiên của mã hoá (mã hoá lặp), nên không có gì ngạc nhiên khi thấy rằng thay vì lặp lại việc truyền của mỗi bít thông tin  $D$  lần, chúng ta có thể sử dụng một mã với khoảng cách Hamming cực tiểu bằng  $D$  và sử dụng giải mã quyết định mềm các lỗi ra từ đầu ra các thiết bị luật bậc hai.

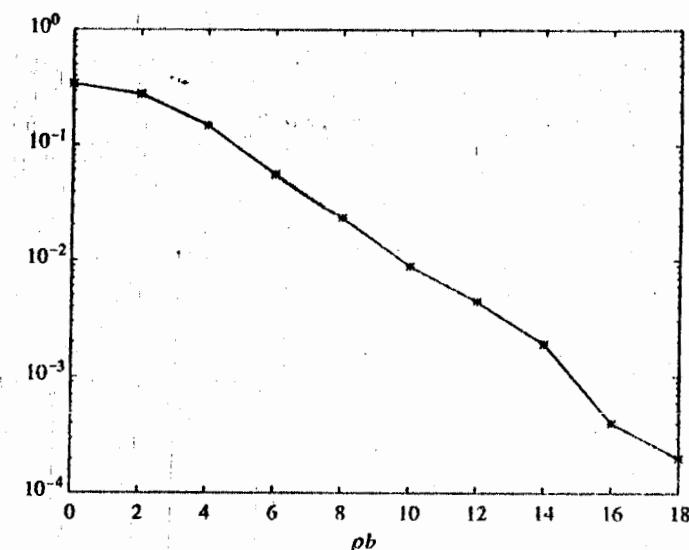
### BÀI TẬP MINH HỌA

**Bài tập minh họa 9.7 [Phân tập trong các hệ thống FH]** Hãy lặp lại mô phỏng Monte-Carlo đối với hệ thống FH đã được xét trong bài tập minh họa 9.6 song bây giờ có áp dụng phân tập kép.

#### Lời giải

Khi không có nhiễu, trọng số được sử dụng tại bộ tổ hợp được đặt là  $w = 10$ , tương ứng với  $\sigma^2 = 0.1$ , một giá trị có thể là tiêu biểu của mức tạp âm cộng Gauss. Mặt khác, khi có nhiễu, trọng số này được đặt là  $w = 1/\sigma^2 = 2/E$ , trong đó  $E$  được ràng buộc theo  $E \geq 4$ . SNR trên một bước nhảy là  $E$  và năng lượng tổng cộng trên một bít trong hai bước nhảy là  $E_b = 2E$ . Do vậy, xác suất lỗi sẽ được vẽ đồ thị như

một hàm của  $E_b/J_0$ . Các kết quả của mô phỏng Monte-Carlo được minh họa trên hình vẽ 9.14.



Hình 9.14 Chất lượng tỷ lệ lỗi của FSK nhị phân phân tần kép FH với nhiều một phần băng - mô phỏng Monte-Carlo

Các MATLAB script dùng cho chương trình mô phỏng được cho dưới đây.  
**m-file**

```
% MATLAB script for Illustrative Problem 7, Chapter 9.
echo on
rho_b=0:2:24; % rho in dB
for i=1:length(rho_b),
    smld_err_prb(i)=ss_Pe97(rho_b(i)); % simulated error rate
    echo off ;
end;
echo on ;
% Plotting commands follow
```

**m-file**

```
function [p]=ss_Pe97(rho_in_dB)
% [p]=ss_Pe97(rho_in_dB)
% SS PE97 finds the measured error rate. The value of signal per
% interference ratio in dB is given as an input to the function.
rho=10^(rho_in_dB/10);
Eb=rho; % energy per information bit
```

```
E=Eb/2; % energy per symbol transmitted
% the optimal value of alpha
if (rho>2),
    alpha=2/rho;
else
    alpha=1;
end;
% the variance of the additive noise
if (E>1),
    sigma=sqrt(E/2);
else
    sigma=sqrt(1/2);
end;
N=10000; % number of bits transmitted
% generation of the data sequence
for i=1:N,
    temp=rand;
    if (temp<0.5)
        data(i)=1;
    else
        data(i)=0;
    end;
end;
% find the transmitted signals
for i=1:N,
    if (data(i)==0),
        tr11c(i)=sqrt(E); tr12c(i)=sqrt(E);
        tr11s(i)=0; tr12s(i)=0;
        tr21c(i)=0; tr22c(i)=0;
        tr21s(i)=0; tr22s(i)=0;
    else
        tr11c(i)=0; tr12c(i)=0;
        tr11s(i)=0; tr12s(i)=0;
        tr21c(i)=sqrt(E); tr22c(i)=sqrt(E);
        tr21s(i)=0; tr22s(i)=0;
    end;
end;

% find the received signals, make the decisions and count the number of errors made.
num_of_err=0;
for i=1:N,
    % determine if there is jamming
```

```
if (rand<alpha),
    jamming1=1; % jamming present on the second transmission
else
    jamming1=0; % jamming not present on the first transmission
end;
if (rand<alpha),
    jamming2=1; % jamming present on the second transmission
else
    jamming2=0; % jamming not present on the second transmission
end;
% the received signals are
if (jamming1==1)
    r11c=tr11c(i)+gngauss(sgma); r11s=tr11s(i)+gngauss(sgma);
    r21c=tr21c(i)+gngauss(sgma); r21s=tr21s(i)+gngauss(sgma);
else
    r11c=tr11c(i); r11s=tr11s(i);
    r21c=tr21c(i); r21s=tr21s(i);
end;
if (jamming2==1)
    r12c=tr12c(i)+gngauss(sgma); r12s=tr12s(i)+gngauss(sgma);
    r22c=tr22c(i)+gngauss(sgma); r22s=tr22s(i)+gngauss(sgma);
else
    r12c=tr12c(i); r12s=tr12s(i);
    r22c=tr22c(i); r22s=tr22s(i);
end;
% compute the decision variables, first the weights
if (jamming1==1),
    w1=1/sgma^2;
else
    w1=10;
end;
if (jamming2==1),
    w2=1/sgma^2;
else
    w2=10;
end;
% the intermediate decision variables are computed as follows
r11=r11c^2+r11s^2;
r12=r12c^2+r12s^2;
r21=r21c^2+r21s^2;
r22=r22c^2+r22s^2;
% finally, the resulting decision variables x and y are computed
```

```
x=w1*r11+w2*r12;
y=w1*r21+w2*r22;
% make the decision
if (x>y),
    decis=0;
else
    decis=1;
end;
% increment the counter if this is an error
if (decis~=data(i)),
    num_of_err=num_of_err+1;
end;
end;
% the measured bit error rate is then
p=num_of_err/N;
```

## CÁC BÀI TẬP

**9.1** Hãy viết một chương trình MATLAB để thực hiện một mô phỏng Monte-Carlo cho một hệ thống trai phổ DS thực hiện truyền thông tin bằng điều chế PSK nhị phân qua một kênh AWGN. Giả sử rằng tăng ích xử lý là 10. Hãy vẽ đồ thị tỷ lệ lỗi đo được theo SNR và nhờ vậy thể hiện được rằng không thu được một tăng ích về chất lượng nào từ tín hiệu trai phổ cả.

**9.2** Hãy viết một chương trình MATLAB để thực hiện một mô phỏng Monte-Carlo cho một hệ thống trai phổ DS hoạt động trong một chế độ LPI. Tăng ích xử lý là 20 (13 dB), còn tỷ số công suất tín/tạp mong muốn tại máy thu trước lúc giải trai phổ là -5 dB hay bé hơn. hãy vẽ đồ thị tỷ lệ lỗi đo được như một hàm của SNR.

**9.3** Lặp lại mô phỏng Monte-Carlo đã được mô tả trong bài tập minh họa 9.4 đối với một tăng ích xử lý bằng 10 và hãy vẽ đồ thị tỷ lệ lỗi đo được.

**9.4** Hãy viết một chương trình MATLAB thực hiện một bộ ghi dịch độ dài cực đại với  $m = 12$  tầng và hãy tạo ra ba chu kỳ của chuỗi. Hãy tính và vẽ đồ thị hàm tự tương quan tuần hoàn của chuỗi lưỡng cực tương đương được cho bởi (9.3.1).

**9.5** Hãy viết một chương trình MATLAB thực hiện một bộ ghi dịch độ dài cực đại  $m=3$  tầng và một bộ ghi dịch độ dài cực đại có  $m = 4$  tầng rồi cộng modulo 2

các chuỗi đầu ra của chúng. Chuỗi nhận được có tuần hoàn không? Nếu tuần hoàn thì chu kỳ của nó là gì? hãy tính và vẽ chuỗi tự tương quan của chuỗi (lưỡng cực) kết quả bằng cách sử dụng (9.3.1).

**9.6** Hãy viết một chương trình MATLAB để tính chuỗi tự tương quan của các dãy Gold có  $L = 31$  đã được tạo ra trong bài tập minh họa 9.5.

**9.7** Một hệ thống FSK nhị phân trực giao FH sử dụng một bộ ghi dịch với  $m = 7$  tầng để tạo ra một chuỗi độ dài cực đại tuần hoàn có độ dài  $L = 127$ . Mỗi một tầng của bộ ghi dịch này chọn lựa một trong 127 băng tần không lấn lên nhau trong mẫu nhảy tần. Hãy viết một chương trình MATLAB mô phỏng việc chọn tần số trung tâm và tạo ra hai tần số trong mỗi một trong  $N = 127$  băng tần số đó. Hãy trình bày mẫu chọn lựa tần số cho 10 khoảng bít đầu tiên.

**9.8** Hãy viết một chương trình Monte-Carlo để mô phỏng một hệ thống thông tin số FH áp dụng sơ đồ điều chế FSK nhị phân với tách tín hiệu không kết hợp (luật bậc hai). Hệ thống này bị nhiễu bởi nhiễu một phần bằng với mật độ phổ công suất  $J_0/\alpha$ , trong đó  $\alpha = 0.1$ . Nhieu này có phổ bằng phẳng trên băng tần số  $0 < \alpha \leq 1$ . Hãy vẽ đồ thị tỷ lệ lỗi đo được cho hệ thống này theo SNR  $E_b/J_0$ .

**9.9** Lặp lại mô phỏng Monte-Carlo trong bài tập minh họa 9.7 khi trọng số được sử dụng tại bộ tổ hợp tín hiệu khi không có nhiễu được đặt là  $w = 100$ , còn khi có nhiễu thì trọng số là  $w = 1/\sigma^2 = 2/E$ , trong đó năng lượng tín hiệu là  $E \geq 4$ . Hãy vẽ đồ thị tỷ lệ lỗi đo được từ mô phỏng Monte-Carlo cho hệ thống phân tần kép này, rồi so sánh chất lượng này với chất lượng nhận được trong bài tập minh họa 9.7.

## TÀI LIỆU THAM KHẢO

1. Proakis, J. G., và Salehi, M.: *Communication Systems Engineering*. Upper Saddle River, NJ. Prentice Hall, 1994.
2. Max, J.: "Quantization for Minimum Distortion", *IRE Transactions on Information Theory*, vol. IT-6 (March 1960).
3. Proakis, J. G.: *Digital Communications* (3<sup>rd</sup> edition). New York. McGraw-Hill, 1995.
4. Forney, G. D., Jr.: "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference", *IEEE Trans. Inform. Theory*, vol. IT-18 (May 1972).
5. Gold, R.: "Optimal Binary Sequences for Spread Spectrum Multiplexing", *IEEE Trans. Inform. Theory*, vol. IT-13 (October 1967).
6. Gold, R.: "Maximal Recursive Sequences with 3-Valued Recursive Cross Correlation Functions", *IEEE Trans. Inform. Theory*, vol. IT-14 (January 1968).
7. Kasami, T.: "Weight Distribution Formular for Some Class of Cyclic Codes", Coordinated Science Laboratory, University of Illinois, Urbana, Ill., Technical Report No. R-285, April 1966.
8. Scholtz, R. A.: "Optimal CDMA Codes", *1979 National Telecommunication Conference Records*, Washington, D. C. (November 1979).
9. Sarwate, D. V. và Pursley, M. B.: "Crosscorrelation Properties of Pseudorandom and Related Sequences", *Proceedings of the IEEE*, vol. 68 (May 1980).