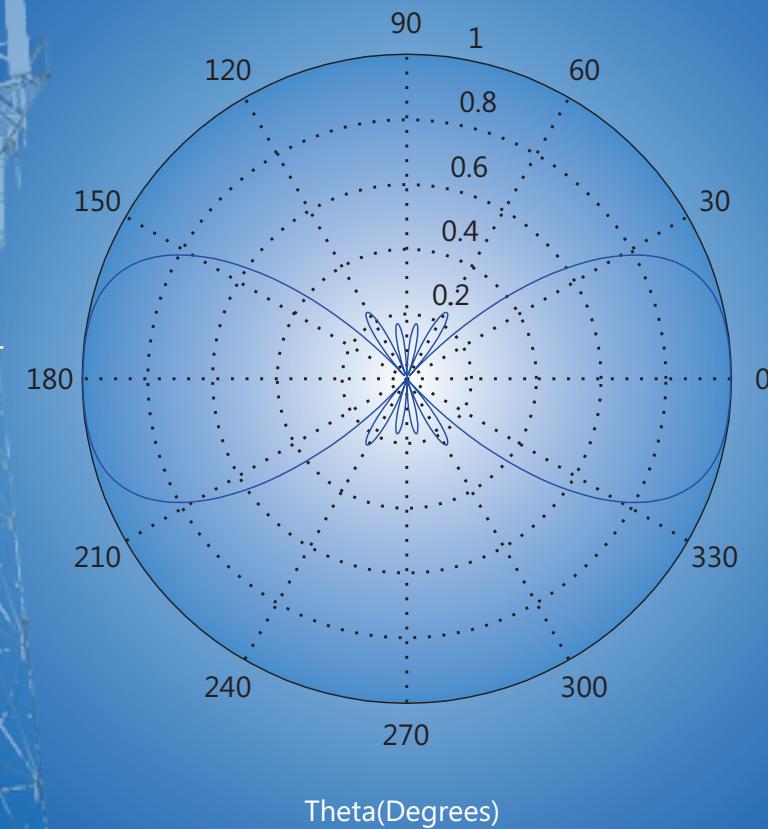




TRẦN XUÂN NAM - LÊ MINH TUẤN

# XỬ LÝ TÍN HIỆU KHÔNG GIAN - THỜI GIAN

Lý thuyết và Mô phỏng



NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT

# **XỬ LÝ TÍN HIỆU KHÔNG GIAN THỜI GIAN**

**Lý thuyết và Mô phỏng**

**Trần Xuân Nam và Lê Minh Tuấn**

PGS.TS. Trần Xuân Nam (*Chủ biên*) - TS. Lê Minh Tuấn

# Xử lý tín hiệu không gian và thời gian

*Lý thuyết và Mô phỏng*



NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT

# Mục lục

Lời nói đầu . . . . .	xiii
<b>I Cơ sở về lập trình MATLAB và lý thuyết mô phỏng</b>	<b>1</b>
<b>1 Giới thiệu MATLAB</b>	<b>3</b>
1.1 MATLAB là gì? . . . . .	3
1.2 Khởi động và Thoát khỏi MATLAB . . . . .	5
1.3 Làm việc với MATLAB Desktop . . . . .	6
1.4 Các lệnh MATLAB cơ bản . . . . .	7
1.5 Các ký hiệu đặc biệt . . . . .	8
Tài liệu tham khảo . . . . .	10
<b>Tài liệu tham khảo</b>	<b>10</b>
<b>2 Tính toán và lập trình sử dụng Matlab</b>	<b>11</b>
2.1 Các phép tính số học . . . . .	11
2.2 Các toán tử so sánh . . . . .	12
2.3 Các toán tử logic . . . . .	14
2.4 Véc-tơ và Ma trận . . . . .	16
2.5 Lập trình với MATLAB . . . . .	23
2.6 Sử dụng đồ họa trong MATLAB . . . . .	33
Tài liệu tham khảo . . . . .	37
<b>Tài liệu tham khảo</b>	<b>37</b>

<b>3 Lý thuyết mô phỏng</b>	<b>39</b>
3.1 Khái niệm và vai trò của mô phỏng . . . . .	39
3.2 Ví dụ về sự cần thiết của mô phỏng trong truyền thông . . . . .	41
3.3 Nguyên lý xây dựng mô hình mô phỏng . . . . .	46
3.4 Các phương pháp mô phỏng . . . . .	47
3.5 Các tham số liên quan đến mô phỏng . . . . .	49
3.6 Mô phỏng Monte-Carlo trong thông tin số . . . . .	53
3.7 Mô phỏng bằng thông và mô phỏng bằng gốc . . . . .	61
3.8 Tóm tắt chương . . . . .	64
Tài liệu tham khảo . . . . .	64
<b>Tài liệu tham khảo</b>	<b>64</b>
<b>II Cơ sở lý thuyết kênh truyền và các phương pháp điều chế</b>	<b>65</b>
<b>4 Kênh thông tin vô tuyến</b>	<b>67</b>
4.1 Kênh tạp âm AWGN . . . . .	67
4.2 Kênh pha-định . . . . .	72
4.3 Kênh pha-định Rayleigh . . . . .	81
4.4 Kênh pha-định Rice . . . . .	83
4.5 Mô phỏng pha-định Rayleigh và pha định Rice . . . . .	85
4.6 Tóm tắt . . . . .	90
Tài liệu tham khảo . . . . .	91
<b>Tài liệu tham khảo</b>	<b>91</b>
<b>5 Các kỹ thuật điều chế số</b>	<b>95</b>
5.1 Các hệ thống thông tin số . . . . .	95
5.2 Các phương pháp điều chế cơ sở . . . . .	98
5.3 Phương pháp điều chế PSK . . . . .	101
5.4 Kỹ thuật điều chế biên độ cầu phương QAM . . . . .	116
5.5 Mô phỏng truyền dẫn BPSK qua kênh AWGN . . . . .	127

5.6	Mô phỏng hệ thống truyền dẫn MPSK . . . . .	130
5.7	Mô phỏng truyền dẫn QAM . . . . .	134
5.8	Tóm tắt . . . . .	137
	Tài liệu tham khảo . . . . .	137
	<b>Tài liệu tham khảo</b>	<b>137</b>
<b>III</b>	<b>Các hệ thống phân tập và ăng-ten mảng thích nghi</b>	<b>157</b>
<b>6</b>	<b>Kỹ thuật thu phát phân tập không gian-thời gian</b>	<b>159</b>
6.1	Các phương pháp phân tập . . . . .	159
6.2	Kỹ thuật kết hợp phân tập không gian thu . . . . .	163
6.3	Kỹ thuật kết hợp phân tập không gian phát . . . . .	178
6.4	Tóm tắt . . . . .	190
	Tài liệu tham khảo . . . . .	191
	<b>Tài liệu tham khảo</b>	<b>191</b>
<b>7</b>	<b>Ăng-ten thích nghi</b>	<b>199</b>
7.1	Các khái niệm cơ bản . . . . .	199
7.2	Mô hình tín hiệu mảng ăng-ten . . . . .	200
7.3	Nguyên lý tạo búp sóng thích nghi . . . . .	204
7.4	Đặc tính búp sóng của ăng-ten thích nghi . . . . .	208
7.5	Tiêu chuẩn tối ưu phẩm chất . . . . .	209
7.6	Các thuật toán thích nghi . . . . .	215
7.7	Ưu điểm của ăng-ten thích nghi . . . . .	220
7.8	Tóm tắt . . . . .	224
	Tài liệu tham khảo . . . . .	224
	<b>Tài liệu tham khảo</b>	<b>224</b>

---

<b>IV Mã không gian-thời gian và các hệ thống MIMO</b>	<b>235</b>
<b>8 Các hệ thống MIMO</b>	<b>237</b>
8.1 Mô hình kênh MIMO . . . . .	237
8.2 Dung lượng kênh truyền MIMO . . . . .	240
8.3 Các phương pháp truyền dẫn trên kênh truyền MIMO	250
8.4 Ghép kênh theo không gian . . . . .	252
8.5 Các bộ tách tín hiệu tuyến tính . . . . .	255
8.6 Các bộ tách tín hiệu phi tuyến . . . . .	263
8.7 Tóm tắt . . . . .	296
Tài liệu tham khảo . . . . .	297
<b>Tài liệu tham khảo</b>	<b>297</b>
<b>9 Mã không gian – thời gian</b>	<b>325</b>
9.1 Giới thiệu . . . . .	325
9.2 Mã khối không gian – thời gian . . . . .	326
9.3 Tóm tắt . . . . .	343
Tài liệu tham khảo . . . . .	344
<b>Tài liệu tham khảo</b>	<b>344</b>

# Danh sách hình vẽ

1.1	Môi trường làm việc của MATLAB . . . . .	6
2.1	Đồ thị $\sin(x)$ và $\cos(x)$ . . . . .	35
2.2	BER của hệ thống BPSK trên kênh pha-định Rayleigh.	36
3.1	Hệ thống truyền thông qua kênh AWGN có thể dễ dàng thực hiện phân tích giải tích. . . . .	43
3.2	Ví dụ về một hệ thống khó thực hiện phân tích giải tích [1]. . . . .	45
3.3	Lược đồ xây dựng mô hình mô phỏng. . . . .	46
3.4	Mối quan hệ giữa sai số, thời gian chạy mô phỏng theo độ phức tạp của mô hình [1]. . . . .	47
3.5	Mô hình mô phỏng Monte-Carlo [3]. . . . .	54
3.6	Biểu diễn phương pháp mô phỏng Monte-Carlo. . .	55
3.7	Ảnh hưởng của số lượng mẫu $N$ tới độ chính xác của kết quả mô phỏng. . . . .	60
4.1	Một ví dụ về tệp âm Gauss với giá trị trung bình 0 và phương sai $\sigma_z^2 = 1$ . . . . .	68
4.2	Hàm mật độ xác suất Gauss với $\sigma_z^2 = 1$ . . . . .	69
4.3	Mật độ phổ công suất và hàm tự tương quan của tệp âm trắng. . . . .	69
4.4	Hàm phân bố của tệp âm AWGN. . . . .	71
4.5	Mô hình truyền sóng đa đường. . . . .	73
4.6	Sự lan truyền của đường $l$ tới một trạm MS. . . . .	75
4.7	Một ví dụ về đáp ứng xung của kênh đa đường. . . .	77

---

4.8	Mô tả phổ tần Doppler có dạng hình chữ U. . . . .	79
4.9	Kênh truyền đa đường được mô hình hóa như một bộ lọc FIR có các hệ số không đổi theo thời gian; $\tau_l = (l + 1)\tau.$ . . . . .	80
4.10	Đáp ứng xung của một bộ lọc FIR. . . . .	80
4.11	Đáp ứng tần số của kênh truyền pha-định: (a) Kênh pha-định phẳng, (b) Kênh pha-định chọn lọc theo tần số. . . . .	81
4.12	Hàm phân bố Rayleigh với $\sigma_h^2 = 1.$ . . . . .	83
4.13	Hàm phân bố Rice cho các giá trị khác nhau của $K$ với $A_p = 1.$ . . . . .	85
4.14	Đường bao tín hiệu bị pha-định Rayleigh, số lượng tia $L = 35$ , tần số Doppler cực đại $f_D = 100$ Hz. . . . .	88
4.15	Đường bao tín hiệu bị pha-định Rice, số lượng tia $L = 35$ , tần số Doppler cực đại $f_D = 100$ Hz, hệ số Rice $K = 5$ và $K = 10.$ . . . . .	90
5.1	Sơ đồ khối của một hệ thống thông tin số điển hình. . . . .	95
5.2	Mô hình hệ thống thông tin số với điều chế và giải điều chế. . . . .	97
5.3	Các ví dụ về điều chế số băng gốc. . . . .	99
5.4	Các phương pháp điều chế sóng mang cơ sở. . . . .	100
5.5	Biểu diễn tín hiệu BPSK trong không gian hai chiều. . . . .	102
5.6	Sơ đồ bộ điều chế/giải điều chế đồng bộ BPSK. . . . .	103
5.7	Chòm sao tín hiệu 8PSK sử dụng mã hóa Gray. . . . .	107
5.8	Sơ đồ điều chế và giải điều chế MPSK. . . . .	108
5.9	Xác suất lỗi ký hiệu của tín hiệu MPSK trong kênh AWGN. . . . .	111
5.10	Xác suất lỗi ký hiệu gần đúng của tín hiệu MPSK. . . . .	112
5.11	Xác suất lỗi ký hiệu của tín hiệu MPSK trong kênh pha-định Rayleigh. . . . .	115
5.12	Một số dạng chòm sao tín hiệu QAM. . . . .	119
5.13	Sơ đồ bộ điều chế và giải điều chế QAM. . . . .	122

5.14	Chòm sao tín hiệu 16-QAM với mã hóa Gray. . . . .	123
5.15	Xác suất lỗi ký hiệu của tín hiệu QAM trong kênh AWGN. . . . .	124
5.16	Xác suất lỗi ký hiệu của tín hiệu QAM trong kênh pha-định Rayleigh. . . . .	126
5.17	Sơ đồ mô phỏng truyền dẫn BPSK trên kênh AWGN. 127	
5.18	Phẩm chất BPSK trên kênh AWGN. . . . .	129
5.19	Mô phỏng truyền MPSK qua kênh pha-định Rayleigh.130	
5.20	Phẩm chất lỗi của hệ thống 8PSK trên kênh AWGN. 133	
5.21	Phẩm chất lỗi của hệ thống 8PSK trên kênh pha-định Rayleigh. . . . .	134
5.22	Phẩm chất lỗi của hệ thống 16QAM trên kênh AWGN.137	
5.23	Phẩm chất lỗi của hệ thống 16QAM trên kênh pha-định Rayleigh. . . . .	138
6.1	Mô hình phương pháp kết hợp chọn lọc. . . . .	164
6.2	Phân phối xác xuất (CDF) của SNR cho phương pháp kết hợp phân tập lựa chọn. (Vẽ bảng chương trình combSNRcdf.m). . . . .	166
6.3	Độ lợi phân tập của các phương pháp kết hợp phân tập. (Vẽ bảng chương trình combSNRcdf.m) . . . . .	167
6.4	Mô hình phương pháp kết hợp tỷ số cực đại. . . . .	169
6.5	Phân phối xác xuất (CDF) của SNR cho phương pháp kết hợp tỉ số cực đại. (Vẽ bảng chương trình combSNRcdf.m) . . . . .	172
6.6	Sơ đồ máy thu với 2 nhánh phân tập MRC và một bộ tách tín hiệu tối ưu. . . . .	175
6.7	Phẩm chất BER trung bình của máy thu MRC với $M$ nhánh phân tập sử dụng điều chế BPSK. . . . .	178
6.8	Sơ đồ phân tập MRT có $N$ nhánh phân tập với các đường phản hồi. . . . .	179
6.9	Sơ đồ cấu hình phân tập phát giữ chậm với $N$ nhánh phân tập. . . . .	180

6.10	Sơ đồ máy phát mã khối STBC Alamouti với 2 ăng-ten phát và 1 ăng-ten thu. . . . .	182
6.11	Sơ đồ Alamouti STBC với 2 ăng-ten phát và 2 ăng-ten thu. . . . .	185
6.12	Phẩm chất BER của các hệ thống Alamouti STBC so sánh với các hệ thống MRC. . . . .	190
7.1	Cấu hình một ăng-ten thích nghi với $M$ phần tử. . . . .	200
7.2	Các cấu hình mảng khác nhau của ăng-ten thích nghi. . . . .	201
7.3	Mô hình tín hiệu của ăng-ten thích nghi. . . . .	201
7.4	Cấu hình một bộ tạo búp sóng băng hẹp. . . . .	205
7.5	Cấu hình bộ tạo búp sóng sử dụng các dây giữ chậm. . . . .	206
7.6	Bộ tạo búp sóng trên miền tần số sử dụng FFT. Ví dụ mô tả cho trường hợp xử lý tín hiệu tại một băng tần con $k$ . . . . .	207
7.7	Búp sóng chuẩn hóa của dàn tuyền tính cách đều với $M = 5$ phần tử và hướng phát xạ $\theta = 60^\circ$ . . . . .	209
7.8	Ví dụ về đường huấn luyện của thuật toán LMS. Ăng-ten mảng tuyền tính với $d = \lambda/2$ , $M = 4$ , $\mu = 0.05$ , và $\text{SNR}_{\text{in}} = 0\text{dB}$ . . . . .	217
7.9	Một ví dụ về đường huấn luyện của thuật toán RLS. Ăng-ten mảng thích nghi với $d = \lambda/2$ , $M = 4$ , $\gamma = 0.999$ , và $\text{SNR}_{\text{in}} = 0\text{dB}$ . . . . .	219
7.10	Tỉ số SNR đầu ra theo số phần tử ăng-ten. . . . .	222
7.11	Cải thiện vùng phủ nhờ sử dụng ăng-ten thích nghi. . . . .	222
8.1	Các mô hình phân tập không gian. . . . .	237
8.2	Mô hình kênh MIMO vô tuyến. . . . .	238
8.3	Mô hình tương đương của kênh truyền SISO. . . . .	241
8.4	Mô hình tương đương của kênh truyền MISO. . . . .	241
8.5	Mô hình tương đương của kênh truyền SIMO. . . . .	242
8.6	Mô hình kênh MIMO tương đương. . . . .	244

8.7	Dung lượng Ergodic ứng với các cấu hình MIMO khác nhau. . . . .	247
8.8	Dung lượng ergodic ứng với các giá trị SNR khác nhau. . . . .	249
8.9	Dung lượng outage $C_{out,5}/W$ ứng với các giá trị SNR khác nhau. . . . .	251
8.10	Phương pháp phân kênh theo không gian. . . . .	253
8.11	Phân loại các bộ tách tín hiệu MIMO-SDM. . . . .	253
8.12	Sơ đồ bộ tách tín hiệu tuyến tính cho MIMO-SDM. .	255
8.13	Sơ đồ nguyên lý hoạt động của bộ tách tín hiệu V-BLAST. . . . .	267
8.14	Phẩm chất của các bộ tách tín hiệu cho hệ thống $4 \times 4$ MIMO-SDM. . . . .	270
8.15	Biểu diễn một dàn 2 chiều. . . . .	271
8.16	Ví dụ biểu diễn thao tác của thuật toán LLL trên một lưới 2 chiều. . . . .	273
8.17	Miền (vùng) quyết định của các bộ tách tín hiệu [18].	276
8.18	Mô hình tương đương của bộ tách tín hiệu có trợ giúp rút gọn cơ sở dàn. . . . .	277
8.19	Minh họa nguyên lý hoạt động của một bộ giải mã cầu. . . . .	283
8.20	BER của hệ thống $4 \times 4$ MIMO-SDM; Điều chế 16QAM.	289
8.21	Nguyên lý tìm các điểm tín hiệu của thuật toán giải mã cầu trong không gian phức. . . . .	291
8.22	Xác định thứ tự thử tối ưu tại lớp $k$ cho các ký hiệu 8-PSK trong $x$ . . . . .	294
8.23	BER của hệ thống $4 \times 4$ MIMO-SDM; Điều chế 8PSK.	296
9.1	Sơ đồ khối của một hệ thống mã hóa STBC. . . . .	326
9.2	Cấu trúc của một khung thông tin gồm các từ mã O-STBC. . . . .	338
9.3	Phẩm chất BER của mã OSTBC thực, $N = 4, M = 1$ ; Điều chế 4PAM. . . . .	342



# Danh sách bảng

2.1	Các tham số định dạng đường đồ thị trong MATLAB	33
6.1	Quy luật mã hóa không gian-thời gian Alamouti . . . . .	181
8.1	Thuật toán tách tín hiệu QR . . . . .	265
8.2	Thuật toán tách tín hiệu V-BLAST . . . . .	268
8.3	Thuật toán rút gọn cơ sở dàn LLL . . . . .	274



# Lời nói đầu

Thông tin vô tuyến đã phát triển mạnh mẽ trong những thập niên vừa qua. Các hệ thống thông tin vô tuyến đã được ứng dụng rộng khắp, đáp ứng nhu cầu truyền thông tốc độ cao và truy nhập “mọi lúc, mọi nơi” của con người. Khác với các hệ thống truyền thông hữu tuyến, các hệ thống thông tin vô tuyến phải làm việc trong các điều kiện băng thông hạn chế và kênh truyền khắc nghiệt chịu ảnh hưởng của các hiện tượng: *pha-định*, *hiệu ứng Doppler* và *nhiễu*. Để hạn chế được các ảnh hưởng này của kênh truyền vô tuyến, các kỹ thuật truyền dẫn tín hiệu đã được phát triển không ngừng. Từ các biện pháp phân tập không gian thu được đề xuất vào những năm 70, một loạt các kỹ thuật truyền dẫn và xử lý tín hiệu mới đã được đề xuất sử dụng và đưa vào các chuẩn giao diện vô tuyến cho các hệ thống thông tin di động đương thời. Phần lớn các kỹ thuật mới tập trung vào khai thác miền không gian còn nhiều tiềm năng, tạo nên kỹ thuật xử lý tín hiệu kết hợp không gian-thời gian. Các kỹ thuật xử lý tín hiệu không gian thời gian có thể kể ra bao gồm: kỹ thuật phân tập không gian thu/phát, kỹ thuật ăng-ten mảng thích nghi (adaptive array antenna), mã không gian thời gian (space-time code), và các hệ thống truyền dẫn đa ăng-ten phát đa ăng-ten thu (MIMO: Multiple Input Multiple Output).

Do có các ưu điểm nổi trội về nâng cao chất lượng tín hiệu thu, loại bỏ ảnh hưởng của hiện tượng truyền sóng đa đường, triệt nhiễu, và nâng cao dung lượng kênh truyền nên các kỹ thuật này đã được tích hợp vào hầu hết các chuẩn giao diện vô tuyến cho các

hệ thống thông tin di động 3G/4G, hệ thống WiMAX (Worldwide Interoperability via Microwave Access), Wi-Fi (Wireless Fidelity), và phát thanh, truyền hình số.

Cuốn sách này tập trung chính vào lý thuyết và phương pháp mô phỏng các kỹ thuật xử lý không gian thời gian nói trên. Mục tiêu của cuốn sách là cung cấp cho người đọc các kiến thức cơ bản nhất về mô hình tín hiệu, các thuật toán xử lý tín hiệu thu/phát, đến phương pháp mô hình hóa và xây dựng chương trình mô phỏng sử dụng công cụ Matlab. Nội dung cuốn sách được chia thành 3 phần:

- Phần 1: Cơ sở về lập trình Matlab và lý thuyết mô phỏng. Phần này trình bày cơ sở về lập trình MATLAB, lý thuyết cơ bản về mô phỏng, và phương pháp mô phỏng Monte-Carlo được sử dụng rộng rãi trong đánh giá các hệ thống truyền thông.
- Phần 2: Cơ sở lý thuyết kênh truyền và các phương pháp điều chế. Nội dung phần này tập trung vào lý thuyết kênh truyền vô tuyến, phương pháp mô phỏng các kênh truyền vô tuyến phổ biến như kênh pha-định Rayleigh, pha-định Rice, và các phương pháp điều chế số.
- Phần 3: Các hệ thống phân tập và ăng-ten mảng thích nghi. Nội dung phần 3 tập trung vào các phương pháp phân tập không gian như các hệ thống phân tập không gian phát (MISO: Multiple-Input Single Input), các hệ thống phân tập không gian thu (SIMO: Single-Input Multiple-Output), các kỹ thuật kết hợp tín hiệu phân tập, và ăng-ten mảng thích nghi.
- Phần 4: Mã không gian-thời gian và các hệ thống MIMO. Phần này tập trung vào các kỹ thuật truyền dẫn không gian thời gian tiên tiến bao gồm: các phương pháp mã hóa và giải mã không gian-thời gian, phương pháp ghép kênh phân chia theo không gian và các kỹ thuật tách tín hiệu.

Cuốn sách được biên soạn nhằm đáp ứng nhu cầu giảng dạy và nghiên cứu về các lĩnh vực liên quan. Nội dung Phần 1 và Phần 2

đã được sử dụng làm giáo trình cho môn học *Mô phỏng các hệ thống viễn thông sử dụng MATLAB* phục vụ cho đào tạo đại học và sau đại học. Nội dung phần 3 và Phần 4 được sử dụng làm giáo trình cho môn học *Xử lý tín hiệu không gian thời gian* phục vụ cho đào tạo sau đại học. Toàn bộ nội dung cuốn sách được tổng hợp từ các kết quả nghiên cứu hơn 10 năm của các tác giả. Hi vọng cuốn sách cũng sẽ là nguồn tài liệu tham khảo có giá trị phục vụ cho đào tạo và nghiên cứu.

Nhóm tác giả

Trần Xuân Nam và Lê Minh Tuấn



## **Phần I**

# **Cơ sở về lập trình MATLAB và lý thuyết mô phỏng**



## Chương 1

# Giới thiệu MATLAB

### 1.1 MATLAB là gì?

MATLAB [1][2] là từ viết tắt của *Matrix Laboratory* với ý nghĩa phần mềm ứng dụng cho tính toán ma trận. MATLAB được mô tả như là một gói phần mềm dùng cho tính toán kỹ thuật trong đó tích hợp các công cụ tính toán, trực quan hóa (visualization), và lập trình. Môi trường làm việc của MATLAB dễ sử dụng và gần gũi với biểu diễn toán học của các phép toán. Các ứng dụng điển hình của MATLAB bao gồm:

- Tính toán toán học,
- Phát triển thuật toán,
- Thu kết dữ liệu (data acquisition),
- Mô hình, mô phỏng và tạo mẫu,
- Phân tích, khai thác và trực giác hóa dữ liệu,
- Đồ họa kỹ thuật,
- Phát triển ứng dụng bao gồm cả việc phát triển giao diện người sử dụng

MATLAB là một hệ thống tương tác trong đó phần tử dữ liệu cơ sở là một mảng không cần định kích thước. Điều này cho phép giải quyết được nhiều vấn đề tính toán, đặc biệt là các vấn đề gắn với các phép toán ma trận hay véc-tơ, mà chỉ tiêu tốn một phần thời gian cần thiết để viết các chương trình sử dụng các ngôn ngữ không tương tác vô hướng (scalar) như C hay FORTRAN.

Hệ thống MATLAB bao gồm năm phần chính:

- **Môi trường phát triển (Development Environment).** Đây là một tập hợp các công cụ và phương tiện hỗ trợ người dùng sử dụng các hàm và tệp MATLAB. Nhiều công cụ là các giao diện đồ họa người dùng (GUI: Graphical User Interface). Tập hợp công cụ này bao gồm Màn hình MATLAB (MATLAB Desktop) và Cửa sổ câu lệnh (Command Window), Lịch sử câu lệnh (Command History), Chương trình soạn thảo và gỡ rối (Editor and Debugger), và một Trình duyệt (Browser) để xem trợ giúp, Không gian làm việc (Workspace), các Tệp, và Đường dẫn Tìm kiếm (Search Path).
- **Thư viện Hàm Toán học (Mathematical Function Library).** Đây là một tập hợp các thuật toán tính toán trải rộng từ các hàm cơ bản như cộng, trừ, sin, cos, các phép tính số học phức, tới các hàm phức tạp hơn như đảo ma trận, tính giá trị riêng (eigenvalue) của ma trận, các hàm Bessel, và các phép biến đổi nhanh.
- **Ngôn ngữ MATLAB (MATLAB Language).** Đây là ngôn ngữ ma trận/mảng bậc cao với các khai báo luồng điều khiển, các hàm số, các cấu trúc dữ liệu, vào/ra, các đặc điểm lập trình hướng đối tượng. Nó cho phép viết cả các chương trình gọn nhẹ hay các chương trình ứng dụng phức tạp.
- **Đồ họa (Graphics.)** MATLAB có nhiều phương tiện hiển thị véc-tơ và ma trận ở dạng đồ thị, cũng như sửa đổi và in các đồ thị này. Nó bao gồm các hàm bậc cao để trực giác hóa các dữ

liệu hai và ba chiều, xử lý ảnh, hình động, và biểu diễn đồ họa. Nó cũng bao gồm cả các hàm bậc thấp cho phép tùy biến hóa đồ họa cũng như xây dựng các giao diện đồ họa hoàn chỉnh cho các ứng dụng MATLAB của người sử dụng.

- **Giao diện chương trình ứng dụng MATLAB (MATLAB Application Program Interface [API]).** Đây là một thư viện cho phép viết các chương trình C và Fortran tương tác với MATLAB. Nó có cả các phương tiện để gọi các thường trình (routine) từ MATLAB, dùng MATLAB như là công cụ tính toán, và dùng để đọc và viết MAT-files<sup>1</sup>.

MATLAB cung cấp một họ các giải pháp theo từng ứng dụng, được gọi là hộp công cụ (toolbox). Hộp công cụ MATLAB bao gồm một tập hợp đầy đủ các hàm MATLAB ở dạng tệp "m" (m-file) dùng để mở rộng môi trường MATLAB cho việc giải quyết các loại vấn đề cụ thể. Các ví dụ về phạm vi ứng dụng của các hộp công cụ MATLAB là xử lý tín hiệu, hệ thống điều khiển, mạng nơ-ron, fuzzy logic, wavelet, mô phỏng, và nhiều ứng dụng khác.

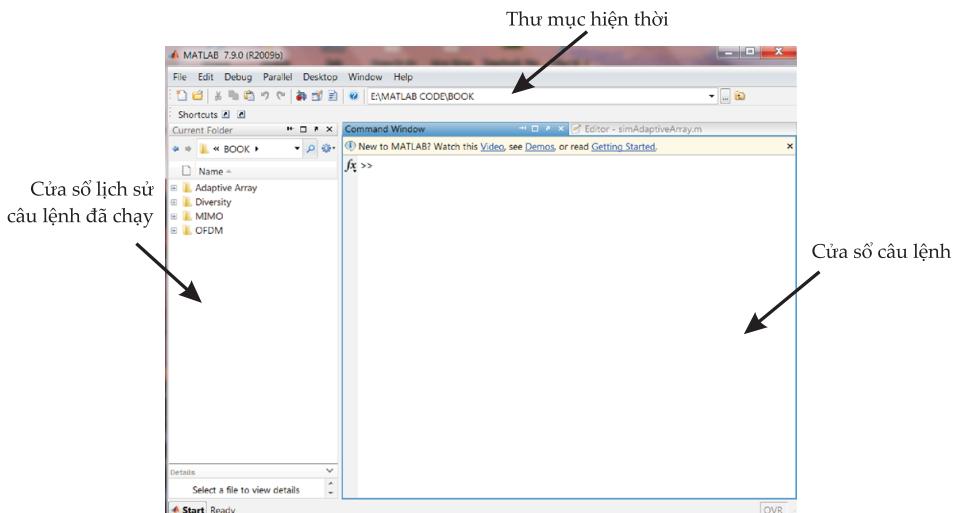
## 1.2 Khởi động và Thoát khỏi MATLAB

Để khởi động MATLAB từ Windows, nhấp đúp (double-click) vào biểu tượng MATLAB  trên màn hình desktop của Windows. Sau khi khởi động xong màn hình sẽ hiện ra cửa sổ Môi trường làm việc của MATLAB gồm 3 phần chính là: Thư mục hiện thời (Current Directory), cửa sổ Lịch sử câu lệnh (Command History) và Cửa sổ câu lệnh (Command Windows) như ở Hình 1.1.

Để kết thúc MATLAB có thể thực hiện bằng cách nhấp phím chuột trái vào ô đóng cửa sổ hình dấu sao ( ) phía trên và bên tay trái cửa sổ MATLAB. Ngoài ra cũng có thể kết thúc MATLAB bằng cách nhập vào câu lệnh

---

<sup>1</sup>Tệp chứa dữ liệu định dạng của MATLAB



**Hình 1.1:** Môi trường làm việc của MATLAB

» quit

ở cửa sổ Command Windows rồi bấm Enter.

### 1.3 Làm việc với MATLAB Desktop

MATLAB Desktop bao gồm một Thanh công cụ (Tool Bar) với các menu File, Edit, Debug, Desktop, Windows và Help. Bên cạnh Thanh công cụ là một menu kéo xuống (Pull-down Menu) cho phép xem và thay đổi thư mục làm việc hiện thời. Nội dung của thư mục làm việc hiện thời được hiển thị ở cửa sổ Current Menu. Phía dưới cửa sổ Current Menu là cửa sổ Command History hiển thị các câu lệnh MATLAB đã được nhập trước đó. Tiếp theo cửa sổ Command History xuống phía dưới có phím Start, cho phép truy nhập nhanh đến các thư viện của MATLAB, Simulink và các cài đặt Desktop Tools hay là các lựa chọn Preferences.

Cửa sổ to nhất trong MATLAB Desktop là cửa sổ câu lệnh Command Window dùng để nhập các câu lệnh MATLAB hay chạy các chương trình cho trước.

## 1.4 Các lệnh MATLAB cơ bản

Các câu lệnh của MATLAB gần giống với các câu lệnh Unix. Một số câu lệnh cơ bản của MATLAB được tóm tắt lại dưới đây:

- » **ls** Liệt kê nội dung của thư mục làm việc hiện thời. Cũng có thể dùng câu lệnh **dir** thay cho câu lệnh **ls**.

**Ví dụ:**

```
» ls  
. . . temp
```

liệt kê thư mục con **temp** bên trong thư mục làm việc hiện thời của MATLAB.

- » **pwd** Hiển thị đường dẫn của thư mục hiện tại.

**Ví dụ:**

```
» pwd  
ans =  
C:\MATLAB701\work
```

chỉ ra đường dẫn của thư mục hiện tại là C:\MATLAB701\work

- » **who** chỉ ra các biến đang được lưu ở bộ nhớ.

**Ví dụ:**

```
>> a=1  
a =  
1  
  
>> b=2  
b =  
2  
  
>> who  
Your variables are:
```

a      b

liệt kê hai biến a và b đang được lưu trữ ở bộ nhớ chương trình.

» clear [tên biến] xóa biến có tên được khai báo khỏi bộ nhớ.

**Ví dụ:**

» clear a

» who

Your variables are:

b

Để xóa hết tất cả các biến đang được lưu tại bộ nhớ, sử dụng lệnh

»clear all

» clc lệnh xóa toàn bộ thông tin trên Command Windows và đưa con trỏ trở về vị trí ban đầu.

## 1.5 Các ký hiệu đặc biệt

- ( ) dấu ngoặc tròn được sử dụng để chỉ ra thứ tự ưu tiên trong các biểu thức số học hoặc bao quanh đối số của một hàm số. Dấu ngoặc tròn cũng được dùng để bao quanh chỉ số phần tử trong một véc-tơ hay ma trận. Ngoài ra, dấu ngoặc tròn này còn được sử dụng để bao quanh các chỉ số dưới (subscript) logic.

**Ví dụ:**

A(2) chỉ ra phần tử thứ 2 của A.

A([1 2 3]) liệt kê các phần tử thứ nhất, hai và ba của A.

A(A>0.5) liệt kê các phần tử của A lớn hơn 0.5.

[ ] dấu ngoặc vuông được sử dụng để tạo các véc-tơ và ma trận

### Ví dụ:

»  $A = [2 \ 6 \ 3]$

$A =$

2    6    3

tạo một véc-tơ hàng với ba phần tử

»  $A = [2 \ 6 \ 3; \ 1 \ 2 \ 3]$

$A =$

2    6    3

1    2    3

định nghĩa một ma trận với sáu phần tử cho trước.

- { } dấu ngoặc móc được sử dụng để tạo ra các mảng tế bào (cell array). Bộ dấu ngoặc mộc này tương tự như bộ ngoặc vuông ngoại trừ các cấp độ lồng ghép (nesting) được bảo toàn.
- ' biểu diễn phép toán chuyển vị liên hợp phức của một ma trận. Ví dụ,  $A'$  là ma trận chuyển vị liên hợp phức của  $A$  còn  $A.'$  là ma trận chuyển vị của  $A$ .
- . dấu chấm biểu diễn phân cách giữa phần nguyên và phần thập phân của một số thập phân. Ví dụ:  $\pi = 3.1416$ .
- ; dấu chấm phẩy dùng để ngăn cách các hàng khi khai báo ma trận, hoặc ngăn không hiển thị kết quả một phép toán trên màn hình.
- % dấu phần trăm dùng để tạo chú thích. Tất cả các câu lệnh viết sau dấu phần trăm này đều bị bỏ qua.
- ... dấu 3 chấm dùng để nối hai phần của một câu lệnh trên 2 dòng với nhau. Một câu lệnh dài có thể viết trên 2 dòng cho tiên theo dõi. Khi đó, dấu 3 chấm được sử dụng để nối 2 dòng với nhau.

## Tài liệu tham khảo

- [1] *Getting started with Matlab.* The Mathworks Inc., 2006.
- [2] A. Biran and M. Breiner, *Matlab for Engineers.* Addison Wesley, 1995.

## Chương 2

# Tính toán và lập trình sử dụng Matlab

## 2.1 Các phép tính số học

Bốn phép tính số học cơ bản gồm cộng, trừ, nhân, chia được thể hiện tương ứng bằng các ký hiệu  $+, -, *, /$ .

Ví dụ:

```
» 2 + 1
```

```
ans=
```

```
3
```

```
» 3 - 1
```

```
ans=
```

```
2
```

```
» 2 * 3
```

```
ans=
```

```
6
```

```
» 6/3
```

```
ans=
```

```
2
```

Với các phép tính phức tạp hơn có dấu ngoặc thì dấu ngoặc tròn (gồm cả mở và đóng) được sử dụng để phân cách thứ tự ưu tiên. Ví

dụ, phép tính

$$\frac{[(1023 + 3012)(150 - 31) - 7](154 - 42)}{2} \quad (2.1)$$

được biểu diễn trong Matlab như sau

```
» ((1023+3012)*(150-31)-7)*(154-42)/2
```

trong đó dấu ngoặc đơn được sử dụng thay cho dấu ngoặc vuông đã được mặc định dùng cho véc-tơ và ma trận trong Matlab.

Trong Matlab phép tính lấy lũy thừa được biểu diễn bởi ký hiệu bởi dấu mũ ^ như:

```
» 5^2
```

```
ans=
```

```
25
```

## 2.2 Các toán tử so sánh

Trong Matlab các toán tử so sánh được biểu diễn như sau: nhỏ hơn (<), lớn hơn (>), nhỏ hơn hoặc bằng (<=), lớn hơn hoặc bằng (>=), bằng (trùng) nhau (==), khác nhau (~=). Khi hai mảng có cùng kích thước được so sánh với nhau thì toán tử so sánh sẽ thực hiện việc so sánh từng phần tử với nhau. Các toán tử <, >, <=, and >= chỉ so sánh phần thực của các toán hạng với nhau. Các toán tử == and ~= thực hiện so sánh cả phần thực và phần ảo của hai toán hạng. Kết quả của phép toán so sánh cho ta 1 nếu phép so sánh là TRUE và ngược lại 0 nếu FALSE. Một số ví dụ về toán tử so sánh được trình bày ở dưới đây.

**Ví dụ:**

```
» 1==2
```

```
ans =
```

```
0
```

```
» 3 > 1
```

```
ans =
```

```
1
```

```
» 4 <= 10
```

```
ans =
```

```
1
```

```
» 3 ~= 7
```

```
ans =
```

```
1
```

Trong trường hợp so sánh 2 véc-tơ hay 2 ma trận với nhau thì toán tử so sánh thực hiện và cho ta kết quả so sánh của từng phần tử tương ứng ở 2 véc-tơ hay ma trận với nhau.

### Ví dụ:

```
» A=[1 3 4; 2 8 7; 6 9 5]
```

```
A =
```

```
1 3 4
```

```
2 8 7
```

```
6 9 5
```

```
» B=[3 1 4; 7 8 2; 6 5 9]
```

```
B =
```

```
3 1 4
```

```
7 8 2
```

```
6 5 9
```

```
» A==B
```

```
ans =
```

```
0 0 1
```

```
0 1 0
```

```
1 0 0
```

```
» A~=B
```

```
ans =
```

```

1   1   0
1   0   1
0   1   1

» A >= B
ans =
0   1   1
0   1   1
1   1   0

» A < B
ans =
1   0   0
1   0   0
0   0   1

```

### 2.3 Các toán tử logic

Các ký hiệu  $\&$ ,  $|$ , và  $\sim$  được sử dụng để biểu diễn toán tử logic AND, OR, và NOT. Các toán tử này làm việc với từng phần tử của mảng, với 0 biểu diễn FALSE còn 1 hay bất kỳ phần tử khác 0 nào biểu diễn TRUE. Các toán tử logic trả lại một mảng logic với các phần tử 0 (FALSE) hoặc 1 (TRUE). Các ký hiệu trên cũng có thể thay thế bằng cách sử dụng các hàm Matlab ở dạng `and(A,B)`, `or(A,B)`, hay `not(A)`. Hàm HỌÂC tuyệt đối được biểu diễn như sau `xor(A,B)`. Trình tự ưu tiên của các toán tử logic là NOT, OR và AND.

Phép toán sử dụng toán tử AND ( $\&$ ) cho kết quả TRUE nếu cả hai toán hạng đều TRUE về mặt logic. Nói theo thuật ngữ số, thì phép toán AND cho ta kết quả TRUE nếu cả hai toán hạng đều khác 0.

**Ví dụ:**

```

» a=[3   5   0   4   0]
a =
3   5   0   4   0

```

```
» b=[3 1 0 0 2]
```

```
b =
```

```
3 1 0 0 2
```

```
» a & b
```

```
ans =
```

```
1 1 0 0 0
```

Các số 1 chỉ ra các phần tử tương ứng khác không của cả a và b.

Phép toán OR ( | ) cho kết quả TRUE nếu một trong hai toán hạng hoặc cả hai toán hạng là TRUE về mặt logic. Nói theo thuật ngữ số thì phép toán OR chỉ cho kết quả FALSE khi cả hai toán hạng đều bằng không.

### Ví dụ:

```
» a | b
```

```
ans =
```

```
1 1 0 1 1
```

Phép toán NOT (~) thực hiện phép đảo toán hạng, cho kết quả FALSE nếu toán hạng là TRUE và cho kết quả TRUE nếu toán hạng là FALSE. Theo thuật ngữ số thì các toán hạng bằng không sẽ bằng một và tất cả các toán hạng khác không sẽ bằng không.

### Ví dụ:

```
» ~a
```

```
ans =
```

0 0 1 0 1

## 2.4 Véc-tơ và Ma trận

### 2.4.1 Tạo véc-tơ và ma trận

Để tạo một véc-tơ cột

$$\mathbf{A} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \quad (2.2)$$

chúng ta chỉ việc nhập vào các phần tử và dấu chấm phẩy (;) để phân chia hàng như sau:

» A=[1; 2; 3]

và thu được

» A=

1

2

3

Để tạo một véc-tơ hàng

$$\mathbf{A} = [1 2 3] \quad (2.3)$$

chúng ta có thể sử dụng dấu cách hoặc dấu phẩy để phân chia cột như sau:

» A=[1, 2, 3]

hoặc

» A=[1 2 3]

đều cho chúng ta kết quả

» A=

1 2 3

Tương tự như véc-tơ để tạo một ma trận, chúng ta nhập trực tiếp các phần tử rồi sử dụng dấu chấm phẩy (;) và dấu cách (hoặc

dấu phẩy) để phân chia các cột và hàng. Ví dụ, để tạo ma trận

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} \quad (2.4)$$

chúng ta nhập vào các phần tử như sau

» A=[1 4 7; 2 5 8; 3 6 9]

và thu được

» A=

$$\begin{array}{ccc} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{array}$$

#### 2.4.2 Các phép toán đối với véc-tơ và ma trận

*Chuyển vị véc-tơ và ma trận:* trong Matlab dấu sắc ('') được sử dụng để biểu diễn phép toán chuyển vị. Ví dụ, để chuyển vị ma trận

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} \quad (2.5)$$

chúng ta chỉ cần thực hiện

» A'

là thu được ma trận chuyển vị của  $\mathbf{A}$

» ans=

$$\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array}$$

Trong trường hợp  $\mathbf{A}$  là một ma trận gồm các phần tử phức thì phép toán '' này còn thực hiện cả việc lấy liên hợp phức của các phần tử trong ma trận. Trong trường hợp chỉ muốn chuyển vị mà không cần lấy liên hợp phức thì nhập thêm dấu chấm vào trước dấu sắc

(.). Ví dụ, ma trận phức

$$A = \begin{bmatrix} -0.43 + 1.06i & -1.14 + 0.29i & 0.32 - 0.69i \\ -1.66 + 0.05i & 1.19 - 1.33i & 0.17 + 0.85i \\ 0.12 - 0.09i & 1.18 + 0.71i & -0.18 + 1.25i \end{bmatrix} \quad (2.6)$$

nếu lấy chuyển vị liên hợp phức cho chúng ta<sup>2</sup>

» A'

» ans=

$$\begin{array}{cccc} -0.43 - 1.06i & -1.66 - 0.05i & 0.12 + 0.09i \\ -1.14 - 0.29i & 1.19 + 1.33i & 1.18 - 0.71i \\ 0.32 + 0.69i & 0.17 - 0.85i & -0.18 - 1.25i \end{array}$$

còn nếu chỉ lấy chuyển vị mà không lấy liên hợp phức sẽ có

» A.'

» ans=

$$\begin{array}{ccc} -0.43 + 1.06i & -1.66 + 0.05i & 0.12 - 0.09i \\ -1.14 + 0.29i & 1.19 - 1.33i & 1.18 + 0.71i \\ 0.32 - 0.69i & 0.17 + 0.85i & -0.18 + 1.25i \end{array}$$

**Cộng và trừ véc-tơ/ma trận:** các phép toán cộng hay trừ véc-tơ/ma trận được thực hiện theo từng phần tử. Phép toán cộng và trừ đòi hỏi các ma trận/véc-tơ được cộng hay trừ phải có kích thước (có cùng số hàng và cột) như nhau. Ví dụ, thao tác cộng hai véc-tơ *a* và *b* được thực hiện như sau

» a=[1; 4; 6]

a =

$$\begin{array}{c} 1 \\ 4 \\ 6 \end{array}$$

» b=[2; 9; 3]

b =

$$\begin{array}{c} 2 \\ 9 \end{array}$$

<sup>2</sup>Chú ý là Matlab sử dụng *i* để hiển thị số ảo, tức là *i* =  $\sqrt{-1}$

```

3
» a + b
ans =

```

```

3
13
9

```

Thao tác trừ hai ma trận  $A$  và  $B$  được biểu diễn như sau:

```
» A=[1 3 7; 4 1 6; 6 4 2]
```

```
A =
```

```

1   3   7
4   1   6
6   4   2

```

```
» B=[5 1 9; 3 8 7; 5 2 3]
```

```
B =
```

```

5   1   9
3   8   7
5   2   3

```

```
» A - B
```

```
ans =
```

```

-4   2   -2
1   -7   -1
1   2   -1

```

**Tích hai véc-tơ:** Hai véc-tơ  $a$  và  $b$  có cùng độ dài có thể nhân với nhau theo cả hai trình tự  $ab$  và  $ba$ . Kết quả là một số vô hướng (scalar) với trường hợp tích trong (inner product) hoặc là một ma trận với trường hợp tích ngoài (outer product).

### Ví dụ:

```
» a = [-1 2 4]
```

```
a =
```

```

-1 2 4
» b = [3; 8; 10]
b =
3
8
10
» a*b
ans =
53
» b*a
ans =
-3    6   12
-8   16   32
-10  20   40

```

**Nhân ma trận:** tích hai ma trận  $C = AB$  được xác định khi chiều của cột  $A$  bằng chiều của hàng  $B$ , hoặc khi một trong số chúng là một số vô hướng. Nếu kích thước của  $A$  là  $m \times p$  và của  $B$  là  $p \times n$ , thì ma trận tích  $C$  của chúng có kích thước là  $m \times n$ . Ví dụ, nhân ma trận  $A$  có kích thước  $2 \times 3$  với ma trận  $B$  có kích thước là  $3 \times 2$  cho ta ma trận  $C$  có kích thước  $2 \times 2$ .

### Ví dụ:

```

» A=[2 5 6; 1 4 3]
A =
2 5 6
1 4 3
» B=[4 9; 7 2; 6 1]
B =
4 9
7 2

```

6 1

» C=A\*B

C =

79 34

50 20

Trong trường hợp nhân một số vô hướng với một ma trận thì kết quả là một ma trận mới với các phần tử là tích của từng phần tử với số vô hướng đó.

### Ví dụ:

» A=[2 5 6; 1 4 3]

A =

2 5 6

1 4 3

» b=5

b =

5

» A\*b

ans =

10 25 30

5 20 15

Đối với trường hợp hai ma trận có cùng kích thước, Matlab cho phép thực hiện nhân từng phần tử với nhau thông qua phép toán nhân từng phần tử (.\*).

### Ví dụ:

» A=[2 4 6; 9 3 5; 1 4 2]

A =

2 4 6

9 3 5

```

    1   4   2
» B=[6   2   5;   7   3   8;   1   0   4]
B =
    6   2   5
    7   3   8
    1   0   4
» C=A.*B
C =
    12   8   30
    63   9   40
    1   0   8

```

**Đảo ma trận:** phép toán đảo ma trận được thực hiện nhờ hàm số có sẵn (built-in function) `inv`. Ma trận đảo `inv(A)` của ma trận A chỉ tồn tại khi A là một ma trận vuông và không gần đơn điệu (nearly singular).

### Ví dụ:

```

» A=rand(4,4)
A =
    0.9077   0.6220   0.2050   0.2539
    0.1772   0.1251   0.5769   0.9528
    0.8046   0.2110   0.4998   0.6918
    0.5592   0.1788   0.0873   0.3441
» B=inv(A)
B =
    -0.4924   -1.2604    1.5160    0.8057
     2.3661    1.4403   -2.5141   -0.6797
     0.5983   -0.5830    3.4275   -5.7183
    -0.5812    1.4479   -2.0271    3.4014

```

trong trường hợp ma trận cần đảo gần với đơn điệu thì Matlab sẽ thông báo lỗi như sau

```

A =
1 2 3
2 4 6
3 5 8
» inv(A) Warning: Matrix is singular to working precision.
(Type "warning off MATLAB:singularMatrix" to suppress this warning.
ans =
Inf Inf Inf
Inf Inf Inf
Inf Inf Inf

```

**Dịnh thức ma trận:** để tính định thức ma trận, chúng ta có thể sử dụng hàm có sẵn (built-in) `det(·)` trong Matlab. Ví dụ, tạo một ma trận ngẫu nhiên với 4 hàng, 4 cột với các phần tử là các số ngẫu nhiên phân bố chuẩn, rồi tính định thức của ma trận đó

```

» randn('seed',0)
» A=randn(4,4)
A =
1.1650 -0.6965 0.2641 1.2460
0.6268 1.6961 0.8717 -0.6390
0.0751 0.0591 -1.4462 0.5774
0.3516 1.7971 -0.7012 -0.3600
» det(A)
ans =
0.0914

```

## 2.5 Lập trình với MATLAB

### 2.5.1 Điều khiển luồng (flow control)

Matlab cung cấp cho người lập trình các cấu trúc luồng sau: `if`, `switch`, `cases`, `for`, `while`, `continue`, và `break`.

- **Câu lệnh if, elseif, và elseif**

Câu lệnh if đánh giá một biểu thức logic và thi hành một nhóm các câu lệnh khi biểu thức đó là TRUE. Các từ khóa tùy chọn khác như elseif hay else thi hành các nhóm câu lệnh khác. Một từ khóa end, khớp với if, được sử dụng ở phần cuối của nhóm câu lệnh cuối cùng.

Ví dụ sau đây hiển thị trên màn hình yêu cầu nhập câu trả lời và nhận câu trả lời ở dạng ký tự (string) từ bàn phím. Nếu câu trả lời là Y thì máy tính in ra màn hình thông báo “Bạn đã đồng ý.” còn nếu câu trả lời là N thì máy tính thông báo “Bạn không đồng ý.” Trong trường hợp chữ nhập vào khác Y hay N máy tính in ra màn hình thông báo “Bạn nhập sai câu trả lời.” và thực hiện ngắt chế độ nhập bàn phím.

### ■ MATLAB Program 2.1 ifCode.m

```
s=input('Nhập "Y" neu dong y hoac "N" neu khong (Y/N):','s');
if s=='Y'
    disp('Ban da dong y.')
elseif s=='N'
    disp('Ban khong dong y.')
else
    disp('Ban nhap sai cau tra loi.')
    return
end
```

Trong ví dụ này, ba trường hợp đều là duy nhất, trường hợp nếu không duy nhất, chỉ có điều kiện đầu tiên được thực hiện.

Điểm cần chú ý khi sử dụng toán tử quan hệ và câu lệnh if với ma trận. Khi muốn kiểm tra sự bằng nhau giữa hai biến, có thể sử dụng

a == b, ...

Đây là câu lệnh hợp lệ của MATLAB khi a and b là các số vô hướng. Tuy nhiên, khi A and B là các ma trận, A == B không kiểm tra xem nếu chúng có bằng nhau không, mà kiểm tra xem chúng bằng nhau ở các vị trí nào, hay nói cách khác, các phần tử nào bằng nhau; và kết quả là một ma trận khác với các phần tử 0 và 1 chỉ ra các phần tử bằng nhau. Nếu A và B không cùng kích thước thì, A == B sẽ trả về một thông báo lỗi.

Cách chuẩn để kiểm tra sự bằng nhau giữa hai biến là sử dụng hàm isequal,

```
if isequal(A,B), ...
```

Ví dụ sau đây cho thấy rõ điểm này. Nếu a và b là các số vô hướng, chương trình sau đây sẽ không bao giờ xảy ra trường hợp không mong muốn. Tuy nhiên, nếu A và B là các ma trận thì phần lớn các trường hợp không xảy ra các điều kiện A > B, A < B, or A == B cho tất cả các phần tử và vì vậy mệnh đề else sẽ được thực hiện

#### ■ MATLAB Program 2.2 elseCode.m

```
if A > B
    'greater'
elseif A < B
    'less'
elseif A == B
    'equal'
else
    error('Unexpected situation')
end
```

- Câu lệnh switch và case

Câu lệnh switch thi hành một nhóm các câu lệnh dựa vào giá trị của một biến hoặc một biểu thức. Các từ khóa case and otherwise được dùng cho các nhóm câu lệnh. Chỉ có trường hợp khớp với điều

kiện đầu tiên là được thi hành, còn các trường hợp sau đó đều bỏ qua (khác với C, vì vậy không cần sử dụng break ). Kết thúc luồng switch là một từ khóa end. Ví dụ sau đây mô tả ví dụ nhập câu trả lời từ bàn phím ở mục trên sử dụng cấu trúc switch.

### ■ MATLAB Program 2.3 switchCode.m

```
s=input('Nhập "Y" nếu đang y hoặc "N" nếu không (Y/N):','s');
switch s
    case 'Y'
        disp('Bạn đã đang y.')
    case 'N'
        disp('Bạn không đang y.')
    otherwise
        disp('Bạn nhập sai câu trả lời.')
        return
end
```

### • Câu lệnh for

Vòng lặp for lặp lại một nhóm câu lệnh một số thời gian cố định cho trước. Cấu trúc vòng lặp được kết thúc bởi từ khóa end. Ví dụ sau đây mô tả việc cách dùng vòng for để tìm các phần tử của một ma trận lớn hơn một giá trị cho trước<sup>3</sup> (bằng 5)

### ■ MATLAB Program 2.4 forCode.m

```
H=[1 8 3; 4 9 6; 7 2 5]
[nRow,nCol]=size(H);
disp('Các phần tử sau >= 5:')
for i=1:nRow
    for k=1:nCol
        if H(i,k)>=5
            disp(['Phần tử ở dòng ', num2str(i), ' cột ', num2str(k)])
        end
    end
end
```

<sup>3</sup>Sau khi đã quen với Matlab có thể dùng hàm có sẵn `find` thay cho vòng `for` này.

```
    end
end
```

---

Kết quả của vòng for này là

```
H =
1     8     3
4     9     6
7     2     5
```

Các phần tử sau  $\geq 5$ :

```
Phần tử ở dòng 1 cột 2
Phần tử ở dòng 2 cột 2
Phần tử ở dòng 2 cột 3
Phần tử ở dòng 3 cột 1
Phần tử ở dòng 3 cột 3
```

Do hàm disp chỉ làm việc với các ký tự (string) nên hàm num2str được dùng ở phần đổi số của hàm disp để biến đổi các chỉ số i và k về dạng ký tự.

- **Câu lệnh while**

Vòng lặp while lặp lại một nhóm câu lệnh một số lần nhất định bằng điều khiển của một điều kiện logic. Cấu trúc vòng lặp while được kết thúc bởi một từ khoá end. Ví dụ sau đây mô tả cách tạo ra một chuỗi dữ liệu  $\{1, -1\}$  trong đó chứa 10 số 1.

---

#### MATLAB Program 2.5 whileCode.m

```
k=1;
noOne=10;
while noOne > 0
    s(k)=2*(rand>0.5)-1;
    if s(k)==1
        noOne=noOne-1;
    end
```

```
k=k+1;
end
s
```

---

Kết quả thu được là một chuỗi 17 số 1, -1 trong đó có chứa 10 số 1

```
s =
Columns 1 through 17
```

```
1     -1      1     -1      1      1     -1     -1      1     -1      1      1      1
```

```
1     -1      -1      1
```

Chú ý là do tính chất ngẫu nhiên của hàm rand nên kết quả có thể khác so với ví dụ. Tuy nhiên, số số 1 vẫn là 10.

- **Câu lệnh break**

Câu lệnh break cho phép thoát sớm khỏi vòng lặp for hay vòng lặp while. Trong trường hợp có nhiều vòng lặp lồng vào nhau thì câu lệnh break chỉ cho phép thoát ra khỏi vòng lặp trong cùng. Ví dụ sau đây mô tả lại phương pháp tạo một chuỗi dữ liệu {1, -1} trong đó chứa 10 bit 1 sử dụng vòng lặp for kết hợp với câu lệnh break.

---

### ■ MATLAB Program 2.6 breakCode.m

```
noOne=0;
k=1;
while k~=0
    n(k)=rand>0.5;
    s(k)=1-2*n(k);
    if s(k)==1
        noOne=noOne+1;
    end
    if noOne==10
        break
    end
    k=k+1;
end
s
```

Kết quả chúng ta cũng thu được một chuỗi gồm 10 bit 1 tương tự như ở ví dụ về câu lệnh while

```
s =
Columns 1 through 12
1     1    -1    -1    -1     1    -1    -1     1     1    -1    -1
Columns 13 through 24
1     1     1     1    -1    -1    -1    -1     1    -1     1
```

### 2.5.2 Tạo chương trình MATLAB bằng tệp .m

Trong phần trước chúng ta đã thấy MATLAB như là một môi trường tính toán tương tác. Ngoài chức năng của một môi trường tính toán tương tác, MATLAB còn cho phép xây dựng các chương trình lập trình như các ngôn ngữ lập trình thông dụng với một kho dữ liệu các hàm xây dựng sẵn (built-in functions). Các chương trình MATLAB được lưu giữ với phần tên tệp mở rộng .m và thường được gọi là các tệp m (m-file). Có hai loại tệp m:

- **Chương trình (script):** thực hiện một tập hợp các câu lệnh và làm việc với các dữ liệu nằm ở không gian công tác (workspace). Các chương trình (script) không nhận đối số đầu vào và cũng không trả các đối số đầu ra.
- **Hàm (function):** thực hiện một tập hợp các câu lệnh chứa trong nó. Tuy nhiên, các function nhận các đối số đầu vào và cho các đối số đầu ra. Các biến khai bao bên trong function chỉ tồn tại trong function.

Để xem hướng dẫn sử dụng các function từ Command Window nhập vào help functionName. Ví dụ, để xem cách sử dụng hàm or chúng ta nhập vào

»help or

và nhận được hướng dẫn sau

| Logical OR. A | B is a matrix whose elements are 1's where either A or B has a non-zero element, and 0's where both have zero elements. A and B must have the same dimensions unless one is a scalar.

C = OR(A,B) is called for the syntax 'A | B' when A or B is an object.  
See also XOR.

Để soạn thảo các script hay các function của MATLAB chúng ta có thể sử dụng bất kỳ một chương trình soạn thảo văn bản nào rồi lưu lại với tên mở rộng .m. Chúng ta cũng có thể sử dụng ngay chương trình soạn thảo của MATLAB (MATLAB Editor). Để xem hay soạn thảo lại một m-file có sẵn chỉ cần sử dụng câu lệnh

» edit fileName

Để tạo một tệp mới chỉ cần sử dụng câu lệnh

» edit

không cần tên tệp.

### • Tạo các hàm MATLAB

Để tạo một function trong MATLAB chúng ta cần tạo một script với dòng đầu tiên có cấu trúc

```
function [Output1,...,OutputM]=functionName(Input1,...,InputN)
```

Trong đó {Input1,...,InputN} là N đối số vào và {Output1,...,OutputM} là M đối số ra. Tên của hàm functionName nhất thiết phải giống với tên tệp m và không trùng với tên các tệp có sẵn.

Ví dụ sau đây hướng dẫn cách viết một function để tính biệt thức  $\Delta = a^2 - 4ac$  của phương trình bậc hai  $ax^2 + bx + c = 0$ .

#### ■ Function 2.1 discriminant.m

```
function delta=discriminant(a,b,c)
% Function to calculate the discriminant of a quadratic equation
% ax^2 + bx + c = 0
delta=b^2 - 4*a*c;
```

Giả sử phương trình bậc hai cần tìm nghiệm số là  $x^2 + 4x + 3 = 0$ . Các hệ số phương trình là  $a = 1$ ,  $b = 4$  và  $c = 3$ . Sử dụng hàm discriminant.m vừa tạo chúng ta có thể tính được ngay biệt thức của phương trình này

```
>>delta=discriminant(1,4,3)
delta =
4
```

#### • Tạo các chương trình MATLAB

Một chương trình MATLAB chứa một tập hợp các câu lệnh. Khi chạy một chương trình MATLAB thì MATLAB tiến hành tất cả các câu lệnh chứa trong nó. Các chương trình MATLAB có thể làm việc với các dữ liệu đang tồn tại ở trong workspace, hoặc cũng có thể tạo ra các dữ liệu mới để làm việc. Tuy các chương trình MATLAB không trả lại các đối số ra (output argument) nhưng các biến do chương trình tạo ra được lưu lại ở workspace và được sử dụng ở các câu lệnh tiếp theo.

Ví dụ chương trình quadEqn.m sau đây mô tả một chương trình MATLAB để tính nghiệm của phương trình bậc hai  $ax^2 + bx + c = 0$

#### ■ MATLAB Program 2.7 quadEqn.m

```
% Chương trình tinh nghiem so cua mot phuong trinh bac hai
% Nhap cac hang so a, b, c
```

```

disp('Chuong trinh tinh nghiem cua phuong trinh bac 2')
a=input('Nhập vào hàng số a=');
b=input('Nhập vào hàng số b=');
c=input('Nhập vào hàng số c=');
% Tính biệt thức delta
delta=discriminant(a, b, c)
% Xét biệt thức và tính nghiệm số
if delta > 0
    x1=(-b + sqrt(delta))/(2*a);
    x2=(-b - sqrt(delta))/(2*a);
    disp(['Hai nghiệm của pt là x1=' num2str(x1) ...
           ' và x2=' num2str(x2)])
elseif delta==0
    x1=-b/(2*a);
    disp(['Phương trình có một nghiệm duy nhất x=' num2str(x1)])
elseif delta < 0
    disp('Phương trình vô nghiệm! ')
end

```

---

Sử dụng chương trình quadEqn.m này tính nghiệm số của phương trình  $x^2 + 4x + 3 = 0$  cho ta kết quả như sau:

```

» quadEqn
Chuong trinh tinh nghiem cua phuong trinh bac 2
Nhập vào hàng số a=1
Nhập vào hàng số b=4
Nhập vào hàng số c=3
Hai nghiệm của phương trình là x1=-1 và x2=-3

```

và của phương trình  $4x^2 + 3x + 2$  là

```

» quadEqn
Chuong trinh tinh nghiem cua phuong trinh bac 2
Nhập vào hàng số a=4
Nhập vào hàng số b=3

```

Nhap vao hang so c=2  
 Phuong trinh vo nghiem!

## 2.6 Sử dụng đồ họa trong MATLAB

### 2.6.1 Vẽ đồ thị

Phương pháp phổ biến nhất để vẽ đồ thị là sử dụng hàm plot có sẵn trong MATLAB. Câu lệnh `plot(x,y)` vẽ đồ thị của  $y$  theo  $x$ , trong đó  $y$  và  $x$  là các véc-tơ có cùng độ dài. Trong trường hợp  $Y$  và  $X$  là các ma trận thì câu lệnh `plot(X,Y)` vẽ ra các đồ thị với các cột (hay hàng) của  $Y$  tương ứng với các cột (hay hàng) của  $X$ .

Trường hợp muốn vẽ nhiều đồ thị trên một hình vẽ chúng ta có thể khai báo theo dạng `plot(x1,y1,x2,y2,...,xN,yN)` với  $\{x_n, y_n\}$  là cặp giá trị của từng đồ thị. Để đánh dấu từng đồ thị chúng ta có thể gán cho chúng các dấu (marker), các màu, kiểu đường đồ thị riêng, hay cũng có thể kết hợp với nhau. Các định dạng đường đồ thị định nghĩa trong MATLAB được liệt kê ở bảng sau

Ví dụ sau đây mô tả phương pháp sử dụng hàm `plot` để vẽ hai đồ thị  $\sin(x)$  và  $\cos(x)$  chung trên một hình vẽ.

#### ■ MATLAB Program 2.8 plotCode.m

```
x = 0:pi/10:2*pi;
y = sin(x);
z= cos(x);
plot(x,y,'r+:', x,z, 'b-.o')
xlabel('x')
ylabel('sin(x)/cos(x)')
title('Do thi ham sin(x) va cos(x)')
legend('sin(x)', 'cos(x)')
grid
```

Trong đó đường đồ thị  $\sin(x)$  được biểu diễn bởi đường đứt nét màu đỏ với các dấu cộng (+), còn đường đồ thị  $\cos(x)$  được biểu

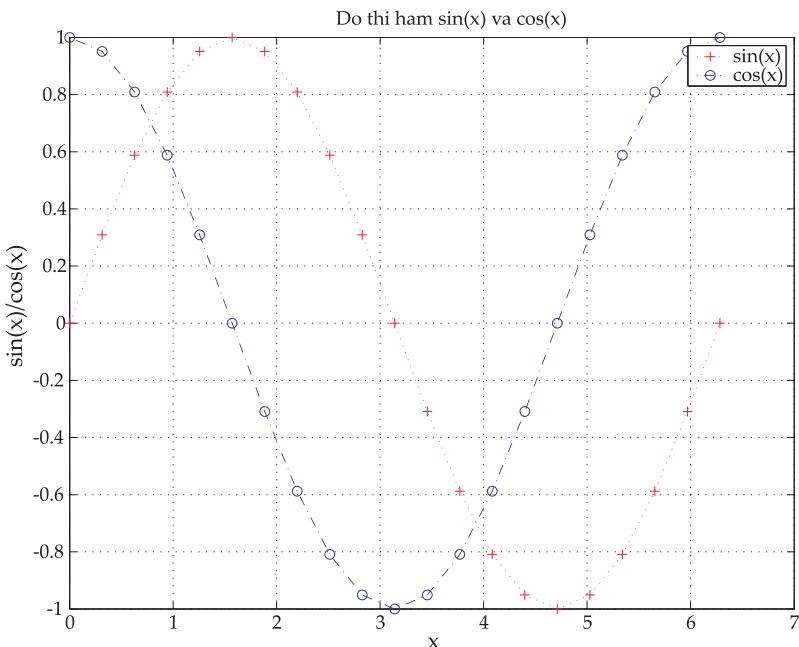
**Bảng 2.1:** Các tham số định dạng đường đồ thị trong MATLAB

Ký hiệu màu	Màu	Ký hiệu dấu	Dấu	Ký hiệu đường	Kiểu đường
b	Xanh nước biển	.	•	-	—
g	Xanh lá cây	o	○	:	.....
r	Đỏ	x	×	-.	----
c	Xanh da trời	+	+	-	----
m	Đỏ thẫm	*	*		
y	Vàng	s	□		
k	Đen	d	◊		
		v	▽		
		^	△		
		<	◀		
		>	▶		
		p	☆		
		h	◊		

điễn bởi đường “gạch và chấm” (.-) màu xanh nước biển với các dấu tròn. Câu lệnh xlabel('x') và xlabel('sin(x)/cos(x)') dùng để đánh dấu trực hoành và trực tung của đồ thị. Câu lệnh title('Đồ thi hàm sin(x) và cos(x)') dùng để đặt tên cho đồ thị. Câu lệnh legend('sin(x)', 'cos(x)') dùng để ghi chú giải cho từng đồ thị và cuối cùng, câu lệnh grid dùng để hiển thị các đường lưới giá trị đồ thị. Kết quả cụ thể được biểu diễn trên Hình vẽ 2.1.

Sử dụng câu lệnh help plot chúng ta có thể biết thêm chi tiết về cách dùng hàm plot. Một câu lệnh vẽ đồ thị khác thường được sử dụng nhiều trong thông tin số để vẽ tỉ số lỗi bit (BER: Bit Error Rate) là câu lệnh semilogy(x,y). Câu lệnh tương tự như câu lệnh plot nhưng cho phép biểu diễn trực tung của đồ thị ở thang logarithm.

Ví dụ, xác suất lỗi của hệ thống truyền dẫn BPSK qua kênh



**Hình 2.1:** Đồ thị  $\sin(x)$  và  $\cos(x)$ .

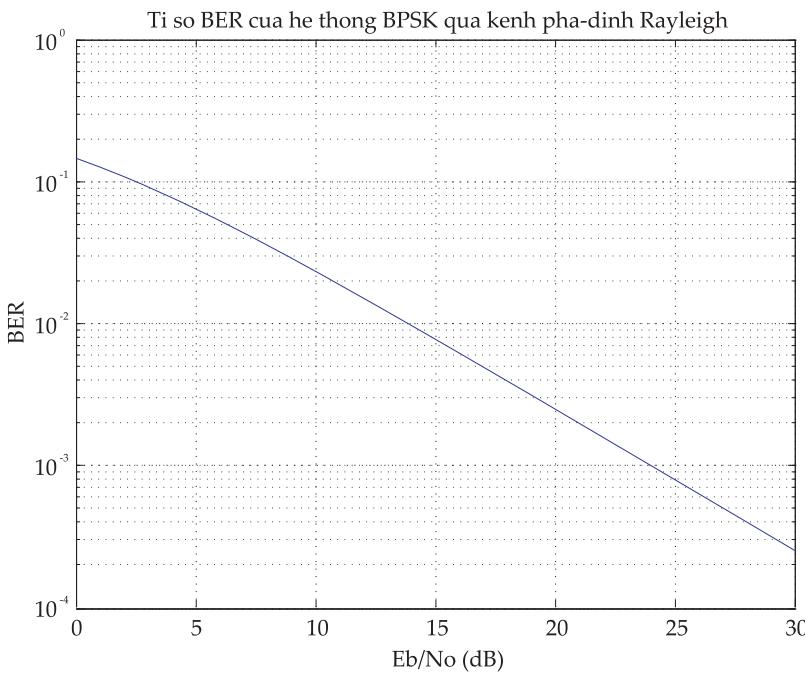
pha-định Rayleigh được cho bởi [1]

$$P_{e_{\text{BPSK-Fading}}} = \frac{1}{2} \left( 1 - \frac{1}{\sqrt{1 + \frac{1}{E_b/N_0}}} \right) \quad (2.7)$$

trong đó  $E_b/N_0$  là tỉ số năng lượng bit tín hiệu trên mật độ phổ tần số tạp âm. Sử dụng chương trình BERBPSKFading.m sau với câu lệnh semilogy(BER, EbNodB) cho phép vẽ đồ thị BER theo tỉ số  $E_b/N_0$  trên thang lô-ga-rít như minh họa trên Hình 2.2.

#### ■ MATLAB Program 2.9 BERBPSKFading.m

```
% Chuong trinh ve do thi BER cua he thong BPSK tren kenh Rayleigh
EbNodB=0:30;
EbNo=10.^{(EbNodB./10)};
BER=1/2*(1-1./sqrt(1+1./EbNo));
semilogy(EbNodB,BER)
xlabel('Eb/No')
ylabel('BER')
```



**Hình 2.2:** BER của hệ thống BPSK trên kênh pha-định Rayleigh.

```
title('Ti so BER cua h/t BPSK qua kenh pha-dinh Rayleigh ')
grid
```

Một số câu lệnh vẽ đồ thị thường dùng khác gồm bar hay hist. Để biết thêm chi tiết về các câu lệnh này sử dụng lệnh help trong MATLAB.

## Bài tập

- 2.1 Sử dụng hàm randn tạo một chuỗi ngẫu nhiên gồm  $N = 1000$  giá trị  $x = \{x_1, x_2, \dots, x_k, \dots, x_N\}$ . Viết chương trình tính giá trị trung bình (kỳ vọng)  $E_x$ , phương sai  $\sigma_x^2$  và độ lệch chuẩn  $\sigma_x$

sử dụng các công thức sau đây

$$E_x = \frac{1}{N} \sum_{k=1}^N x_k \quad (2.8)$$

$$\sigma_x^2 = E(x^2) - E_x^2 = \frac{\sum_{k=1}^N x_k^2}{N} - \left( \frac{\sum_{k=1}^N x_k}{N} \right)^2 \quad (2.9)$$

$$\sigma_x = \sqrt{\sigma_x^2}. \quad (2.10)$$

- 2.2 Viết chương trình tạo hai chuỗi độc lập  $x$  và  $y$  có độ dài  $10^4$  bit  $\{-1, +1\}$ , trong đó các bit  $\{-1, +1\}$  xuất hiện ngẫu nhiên. So sánh  $y$  và  $x$ , và tính toán tỉ số sai số giữa hai chuỗi.
- 2.3 Dựa vào ví dụ vẽ đồ thị BER của hệ thống BPSK trên kênh pha-định Rayleigh ở mục 2.6.1 và hàm sai số bù erfc có sẵn trong MATLAB,

- Viết chương trình vẽ đồ thị BER của hệ thống BPSK trên kênh tạp âm Gauss theo công thức sau [1]

$$\text{BER}_{\text{BPSK-AWGN}} = \frac{1}{2} \text{erfc} \left( \sqrt{E_b/N_0} \right) \quad (2.11)$$

- Vẽ đồ thị BER của hệ thống BPSK trên kênh tạp âm Gauss kết hợp với đồ thị BER của hệ thống BPSK trên kênh pha-định Rayleigh vào một hình vẽ, đánh dấu và đặt chú giải cho từng đồ thị.

- 2.4 Viết một hàm MATLAB  $y=\text{QPSKMap}(x)$  nhận đối số là một chuỗi ngẫu nhiên 1 và 0 có độ dài  $N$  bit,  $N$  chẵn. Hàm QPSKMap thực hiện kiểm tra từng cặp 2 bit liên tiếp  $x_n$  và  $x_{n+1}$  rồi thực hiện phép biến đổi sau:

trong đó  $n = 1, 2, \dots, N$  và  $m = 1, 2, \dots, N/2$ .

- 2.5 Viết một chương trình QPSKMod tạo ra 20 bit 0, 1 ngẫu nhiên. Sử dụng hàm MATLAB QPSKMap ở ví dụ trên để chuyển chuỗi bit {0,1} thành chuỗi các dấu QPSK  $y_m$ .

$x_n x_{n+1}$		$y_m$
00	$\rightarrow$	$\frac{1}{\sqrt{2}} + j \frac{1}{\sqrt{2}}$
01	$\rightarrow$	$-\frac{1}{\sqrt{2}} + j \frac{1}{\sqrt{2}}$
10	$\rightarrow$	$-\frac{1}{\sqrt{2}} - j \frac{1}{\sqrt{2}}$
11	$\rightarrow$	$\frac{1}{\sqrt{2}} - j \frac{1}{\sqrt{2}}$

## Tài liệu tham khảo

- [1] H. Harada and R. Prasad, *Simulation and Software Radio for Mobile Communications*. Artech House, 2002.

## **Chương 3**

# **Lý thuyết mô phỏng**

### **3.1 Khái niệm và vai trò của mô phỏng**

Mô phỏng là phương pháp thiết kế mô hình của một hệ thống thực hoặc lý thuyết để tiến hành các thực nghiệm trên mô hình đó trên máy tính giúp cho việc phân tích để hiểu biết về hoạt động của một hệ thống hoặc các thuật toán liên quan. Mô phỏng đóng vai trò là công cụ hỗ trợ việc thiết kế và phân tích hệ thống và được sử dụng trong các trường hợp sau đây:

- Mô hình hệ thống cần thiết kế hoặc phân tích quá phức tạp, trong đó có chứa nhiều biến số và các phần tử tương tác với nhau.
- Quan hệ giữa các biến cơ bản trong mô hình là phi tuyến.
- Mô hình có chứa các biến số ngẫu nhiên.
- Kết quả mô hình cần hiển thị trên đồ họa 3D.
- Hệ thống cần thiết kế chưa tồn tại trong thực tế.
- Phi thực tiễn hay quá nguy hiểm để có thể thử nghiệm.

Ưu điểm của mô phỏng hơn so với phương pháp phân tích lý thuyết hoặc thử nghiệm trên hệ thống thực bao gồm:

- Cho phép cấu hình lại mô hình cần nghiên cứu và tiến hành thực nghiệm lại một cách dễ dàng thông qua thay đổi trên phần mềm mô phỏng. Điều này không cho phép thực hiện được với các hệ thống thực tế do chi phí quá cao hoặc phi thực tế.
- Cho phép nghiên cứu hoạt động của hệ thống, rút ra kết luận về các đặc tính liên quan đến thao tác của hệ thống thực và các hệ thống con một cách dễ dàng.

Mô phỏng được sử dụng rộng rãi trong nhiều công đoạn của quá trình thiết kế và triển khai các hệ thống truyền dẫn hiện đại. Mục đích chính của mô phỏng là để đánh giá phẩm chất và tối ưu tham số. Ngoài ra, mô phỏng còn được sử dụng để thiết lập các thủ tục kiểm chuẩn (benchmark), dự đoán tuổi thọ, và thẩm định hệ thống sau khi đã được triển khai ra hiện trường. Phương pháp mô phỏng sẽ chịu chi phối hay hướng dẫn do dòng thiết kế tổng quát sử dụng.

Đối với các hệ thống truyền thông hiện đại do nhu cầu về tốc độ và hiệu năng cao đòi hỏi sử dụng các phương thức điều chế bậc cao kết hợp mới mã sửa lỗi và các phương pháp xử lý tín hiệu phức tạp khác ở phía thu, nên hệ thống ngày càng phức tạp. Các hệ thống này lại thường được sử dụng để làm việc trên các kênh truyền phức tạp, như kênh thông tin di động tê bào, nên chịu ảnh hưởng lớn của hiện tượng pha-định đa đường và nhiễu. Điều này dẫn đến việc thiết kế và phân tích giải tích để đánh giá phẩm chất hệ thống trở nên hết sức phức tạp. Trong các trường hợp như vậy thì nhờ có sự hỗ trợ của mô phỏng các chức năng xử lý phức tạp của cả máy phát và máy thu, hay các tác động của pha-định và nhiễu đều có thể thực hiện được dựa trên các máy tính thông thường. Ngoài ra, sự phát triển của máy tính cũng hỗ trợ cho sự phát triển của lý thuyết mô phỏng.

Trong những năm gần đây, ngày càng có nhiều công cụ mô phỏng có tính năng cao hỗ trợ cho việc mô phỏng được thực hiện dễ dàng hơn. Động cơ quan trọng cho mô phỏng là do mô phỏng là

công cụ quý giá cho phép tìm hiểu sâu về hoạt động của hệ thống. Một hệ thống mô phỏng hoàn chỉnh đóng vai trò như một phòng thí nghiệm cho phép kiểm tra tại nhiều vị trí trong hệ thống. Và vì vậy, các tham số nghiên cứu như độ rộng băng tần, tham số bộ lọc hay tỉ số tín hiệu trên tạp âm (SNR: Signal to Noise Ratio) có thể thay đổi được theo mong muốn. Hiệu quả của các thay đổi này có thể dễ dàng quan sát trên màn hình máy tính. Các tham số như dạng sóng, phổ tín hiệu, sơ đồ mẫu mắt, chòm sao tín hiệu, biểu đồ histogram, hay nhiều biểu đồ khác đều có thể được hiển thị trên màn hình máy tính, cho phép người nghiên cứu có thể phân tích, đánh giá và so sánh với các kết quả thực hiện trên phần cứng.

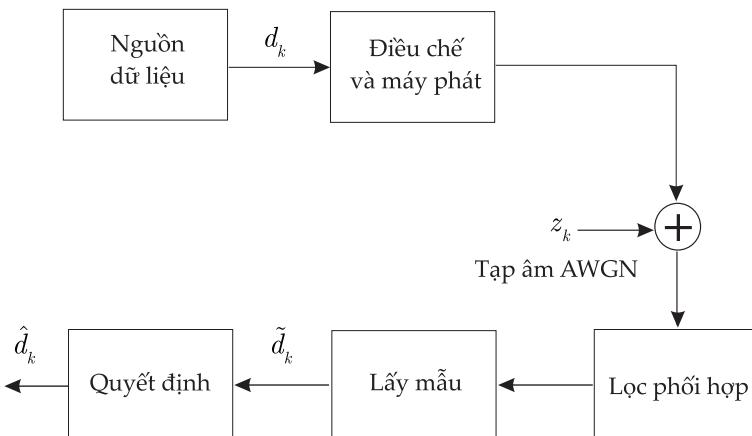
Việc thiết kế của một hệ thống truyền thông phức tạp thường được thực hiện theo phương thức “từ trên xuống dưới”. Tức là, khi thiết kế một hệ thống chúng ta bắt đầu ở mức hệ thống và bắt đầu bổ sung chi tiết của thiết kế từ hệ thống xuống các hệ thống con, và cuối cùng là đến mức phần tử. Khi triển khai hệ thống thì quá trình lại được thực hiện “từ dưới lên trên”. Tức là, các phần tử được chế tạo trước, sau đó chúng được lắp ráp thành các hệ thống con, và cuối cùng toàn bộ hệ thống được xây dựng từ các hệ thống con. Tương tự như thiết kế hệ thống, phát triển mô phỏng cũng được thực hiện theo giải pháp “từ trên xuống dưới”. Chúng ta bắt đầu với một mô phỏng ở mức hệ thống có mức trừu tượng cao, tiếp theo là các mô hình cụ thể hơn và mô phỏng của các hệ thống con và các phần tử. Khi bắt đầu quá trình triển khai, các đặc tính đo được của các phần tử và hệ thống con được bổ sung vào mô hình mô phỏng.

### **3.2 Ví dụ về sự cần thiết của mô phỏng trong truyền thông**

Để hiểu rõ hơn vai trò của mô phỏng, chúng ta xét 3 sơ đồ hệ thống sau đây.

### 3.2.1 Sơ đồ truyền dẫn số qua kênh AWGN

Hình 3.1 mô tả một sơ đồ truyền dẫn thông tin số cơ bản.



**Hình 3.1:** Hệ thống truyền thông qua kênh AWGN có thể dễ dàng thực hiện phân tích giải tích.

Tại phía phát, nguồn dữ liệu tạo ra một chuỗi các bít dữ liệu phát,  $s_k \in \{0, 1\}$ , độc lập với nhau. Các nguồn dữ liệu kiểu này thường được sử dụng phổ biến trong mô phỏng và được gọi là nguồn gián đoạn không nhớ (DMS: Discrete Memoryless Source). Các bít dữ liệu phát được tạo ra sau đó được đưa đến bộ điều chế và ánh xạ thành các dạng sóng phù hợp. Với hệ thống nhị phân, tập dạng sóng được định nghĩa  $\{s_1(t), s_2(t)\}$ . Máy phát (Tx), sau đó, sẽ thực hiện chức năng khuếch đại dạng sóng đầu ra của bộ điều chế để phát đi trên kênh truyền với công suất yêu cầu.

Tín hiệu phát đi qua kênh truyền trước khi đến máy thu. Trong thực tế kênh truyền là một môi trường truyền dẫn phức tạp, gây ảnh hưởng lớn đến chất lượng truyền dẫn tín hiệu. Tuy nhiên, trong ví dụ đơn giản này, chúng ta giả thiết kênh truyền chỉ tạo ra tạp âm trắng cộng tính (Additive White Gaussian Noise: AWGN). Tín hiệu thu nhận được ở đầu vào máy thu sẽ được đưa qua bộ lọc phôi hợp, còn được gọi là máy thu tương quan. Đầu ra bộ lọc phôi hợp được lấy mẫu tại cuối chu kỳ dấu để tạo nên thông kê quyết định (decision statistic),  $\tilde{d}_k$ , rồi so sánh với ngưỡng quyết định  $T_h$  để tạo nên ước lượng  $\hat{d}_k$  của dữ liệu gốc  $d_k$ . Nếu  $\tilde{d}_k > T_h$  thì quyết định được thực hiện theo một trong hai dấu, còn ngược lại,  $\tilde{d}_k < T_h$ , quyết định

theo dấu còn lại. Máy thu kiểu này thường được gọi là máy thu tối ưu do bản chất của việc ước lượng tín hiệu phát là làm tối thiểu hóa xác suất lỗi  $P_e$ .

Hệ thống đề cập đến ở Hình 3.1 là một hệ thống có thể phân tích bằng giải tích một cách dễ dàng nhờ các kiến thức cơ bản về lý thuyết thông tin và giải tích. Thực tế là xác suất lỗi  $P_e$  đã được tính toán một cách dễ dàng và trình bày ở trong hầu hết các tài liệu về thông tin số và cho bởi [1]

$$P_e = Q\left(\sqrt{k \frac{E_s}{N_0}}\right) \quad (3.1)$$

trong đó  $E_s$  là năng lượng trung bình của các dấu phát,  $N_0$  là mật độ phổ công suất đơn phía của tạp âm,  $k$  là hệ số xác định bởi tương quan giữa các dạng sóng  $\{s_1(t), s_2(t)\}$ . Nếu các tần số được chọn một cách chính xác, các tín hiệu không tương quan thì  $k = 1$ . Với tín hiệu điều chế khóa dịch pha (PSK), các tín hiệu điều chế có cùng tần số và công suất, nhưng khác pha ban đầu. Trong trường hợp pha khác nhau  $\pi$ , sao cho  $s_2(t) = -s_1(t)\}$ , thì  $k = 2$ .

Sở dĩ chúng ta nói rằng hệ thống truyền dẫn trên Hình 3.1 là hệ thống có thể phân tích bằng giải tích dễ dàng là do các lý do sau:

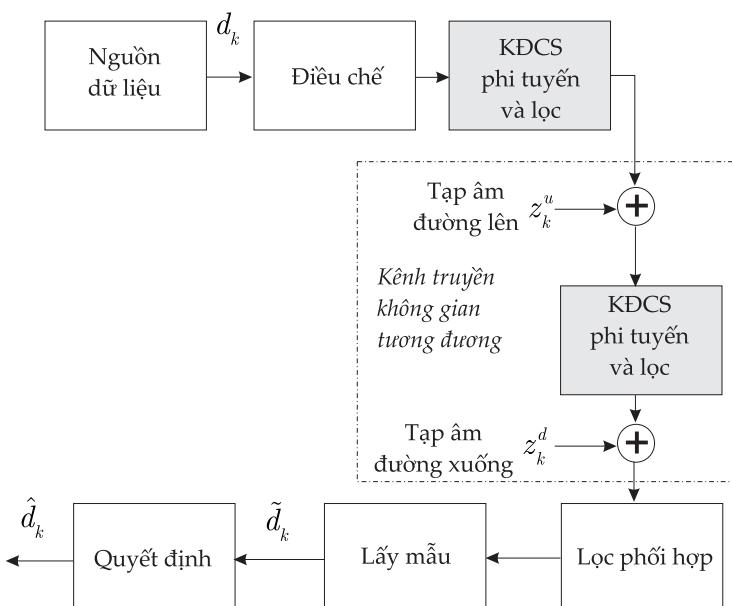
- Do giả thiết kênh truyền AWGN và máy thu tuyến tính. Giả thiết này dẫn đến thông kê quyết định  $\tilde{d}$  trở thành một biến Gauss ngẫu nhiên.
- Do các giả thiết nguồn dữ liệu không có nhớ.
- Do giả thiết đồng bộ dấu được thực hiện lý tưởng nên chúng ta có thể biết chính xác thời điểm bắt đầu và kết thúc của một dấu, vì vậy, cho phép thông kê quyết định được tách ra một cách chính xác.

Mặc dù có thể phân tích được dễ dàng nhưng trong một số trường hợp xây dựng chương trình mô phỏng cho các hệ thống kiểu này vẫn cần thiết. Lý do là do đây là một hệ thống cơ bản nên nó

thường được sử dụng làm cơ sở để mở rộng cho các hệ thống truyền dẫn phức tạp hơn. Ví dụ, nếu thay khói kênh AWGN bằng khói kênh pha-định Rayleigh chúng ta có mô hình truyền dẫn số qua kênh pha-định Rayleigh, hay chúng ta cũng có thể thêm vào khói san bằng kênh ở máy thu để có được sơ đồ truyền dẫn sử dụng bộ san bằng để loại bỏ ảnh hưởng của pha-định chọn lọc theo tần số đối với kênh có trễ. Trong những trường hợp như vậy, việc xây dựng thành công chương trình mô phỏng cho hệ thống truyền dẫn cơ bản được kiểm nghiệm bằng lý thuyết này, cho phép mở rộng nhanh chóng để xây dựng thành công chương trình mô phỏng cho các hệ thống phức tạp.

### 3.2.2 Hệ thống truyền dẫn qua kênh thông tin vệ tinh

Hệ thống mô tả ở Hình 3.2 được biết đến như là một hệ thống không thể phân tích được bằng giải tích.



**Hình 3.2:** Ví dụ về một hệ thống khó thực hiện phân tích giải tích [1].

Hệ thống này là một mô hình truyền dẫn hai chặng qua kênh

vệ tinh, trong đó bộ phát đáp được mô hình hóa bởi một bộ khuếch đại công suất lớn (HPA) phi tuyến và một bộ lọc để loại bỏ méo hài ngoài băng tần do tính phi tuyến gây nên. Quan sát Hình 3.2 chúng ta thấy rằng mô hình hệ thống truyền dẫn trong trường hợp này có hai nguồn tạp âm: một cho đường lên (up-link)  $z_k^u$  và một cho đường xuống (down-link)  $z_k^d$ . Như vậy, tín hiệu ở máy thu chứa hai thành phần tạp âm, trong đó tạp âm tuyến lên đi qua một bộ khuếch đại phi tuyến ở trên bộ phát đáp của vệ tinh. Dù cho chúng ta có giả thiết là cả tạp âm trên đường lên và đường xuống đều là tạp âm AWGN, thì hàm mật độ công suất (PDF) của tạp âm ở máy thu vẫn rất khó xác định, đặc biệt là với tạp âm đường lên. Vì vậy, mô phỏng là giải pháp cần thiết đối với các hệ thống kiểu này.

### 3.3 Nguyên lý xây dựng mô hình mô phỏng

Bước đầu tiên trong việc phát triển một chương trình mô phỏng của một hệ thống truyền thông là phát triển mô hình mô phỏng của hệ thống đó. Mô hình thường được biểu diễn ở dạng công thức toán học mô tả mối quan hệ vào/ra của hệ thống, còn được gọi là phương trình hệ thống. Nghệ thuật của mô hình hóa là phát triển mô hình hoạt động có chứa đầy đủ các tính năng cần thiết nhưng lại không quá phức tạp để có thể thực hiện được bằng các máy tính thông dụng. Yêu cầu này đòi hỏi phải có sự thỏa hiệp giữa tính chính xác, độ phức tạp và yêu cầu tính toán của mô hình. Đối với một quá trình mô phỏng, thông thường có hai mô hình được xây dựng: *mô hình giải tích* và *mô hình mô phỏng* như mô tả ở Hình 3.3.

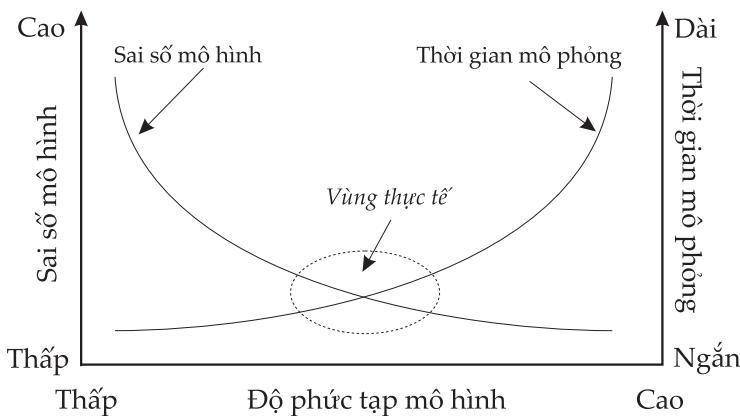
Cả hai mô hình này đều mô tả tính khái quát (abstract) của hệ thống. Mô hình giải tích thường biểu diễn ở dạng công thức toán học hay các hệ phương trình xác định mối quan hệ vào/ra của hệ thống. Các công thức này thường mô tả một phần của hệ thống và có độ chính xác trong một giải điện áp, dòng điện, hay tần số nào đó. Mô hình mô phỏng thường là một tập hợp của các thuật toán



**Hình 3.3:** Lược đồ xây dựng mô hình mô phỏng.

thực hiện giải pháp tính toán bằng số (numerical) của các công thức xác định mô hình giải tích. Các kỹ thuật giải tích số và xử lý tín hiệu số là các công cụ được sử dụng để phát triển các thuật toán này.

Mối quan hệ giữa sai số mô hình, độ phức tạp và thời gian mô phỏng được biểu diễn ở Hình 3.4. Chúng ta có thể thấy rằng một mô hình có độ phức tạp thấp có sai số mô hình hóa lớn, nhưng lại yêu cầu thời gian chạy mô phỏng ngắn. Ngược lại, mô hình có độ phức tạp lớn có sai số nhỏ nhưng lại yêu cầu thời gian mô phỏng dài.



**Hình 3.4:** Mối quan hệ giữa sai số, thời gian chạy mô phỏng theo độ phức tạp của mô hình [1].

### 3.4 Các phương pháp mô phỏng

Có hai loại mô phỏng cơ bản: *mô phỏng xác định* và *mô phỏng ngẫu nhiên*. Mô phỏng xác định thường gặp trong trường hợp mô phỏng

các mạch điện được thiết kế sử dụng một chương trình thiết kế như kiểu SPICE. Chương trình này được sử dụng để tạo ra một mạch điện và cấp dòng đầu vào. Chương trình mô phỏng tạo ra dòng điện chạy ở các nhánh của mạch điện và điện áp qua từng phần tử. Điện áp và dòng điện thường được biểu diễn bởi các dạng sóng. Khoảng thời gian mong muốn của các dạng sóng này được xác định trước khi chạy mô phỏng. Do mạch điện cố định và tín hiệu đầu vào là xác định nên mỗi lần chạy mô phỏng sẽ cho cùng một kết quả giống nhau. Hơn nữa, có thể sử dụng tinh tay để tìm ra cùng các dạng sóng như vậy. Trong trường hợp này, mô phỏng được sử dụng để tiết kiệm thời gian và tránh các lỗi toán học do phải thực hiện các phép toán nhầm chán.

Nếu giả thiết đầu vào của hệ thống là một dạng sóng ngẫu nhiên. Nói chính xác là nếu đầu vào hệ thống là một hàm mẫu của một quá trình ngẫu nhiên, thì một cách tương đương, có thể coi trở kháng của điện trở là một biến ngẫu nhiên xác định bởi một hàm mật độ xác suất xác định. Kết quả của mô phỏng này sẽ không còn là một dạng sóng xác định và các mẫu của dạng sóng này sẽ tạo nên một tập hợp của các biến ngẫu nhiên. Các mô phỏng trong đó xuất hiện các giá trị ngẫu nhiên được gọi là các mô phỏng ngẫu nhiên.

Để lấy ví dụ, chúng ta giả thiết điện áp qua một phần tử mạch điện được ký hiệu là  $u(t)$  và mô phỏng được sử dụng để tạo ra  $u(t)$  trong khoảng thời gian 1ms, tức là  $u(0.01)$ . Trong mô phỏng xác định thì  $u(0.01)$  không đổi và chúng ta có kết quả như nhau sau mỗi lần mô phỏng.

Một ví dụ khác là hệ thống truyền dẫn số trong đó tín hiệu thu bao gồm tín hiệu phát cộng tạp âm ngẫu nhiên. Giả thiết rằng nhiệm vụ của chúng ta là tính xác suất lỗi dấu tại đầu ra máy thu. Từ giáo trình truyền dẫn chúng ta biết rằng đối với truyền dẫn tín hiệu BPSK qua kênh AWGN, thì xác suất lỗi dấu là [2]

$$P_e = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (3.2)$$

trong đó  $E_b$  là năng lượng bít,  $N_0$  là mật độ phổ công suất tạp âm một phía, và  $Q(x)$  là hàm  $Q$  Gauss được định nghĩa bởi

$$Q(x) = \frac{1}{2\pi} \int_x^\infty \exp\left(-\frac{y^2}{2}\right). \quad (3.3)$$

Để ý rằng  $P_e$  là một số chứ không phải một biến ngẫu nhiên, mặc dù tạp âm ngẫu nhiên xuất hiện ở đầu vào máy thu. Số  $P_e$  là một giá trị trung bình sau một số lần thử vô hạn, trong đó một lần thử bao gồm việc gửi một số dấu số đi qua hệ thống và quan sát kết quả đầu ra. Tất nhiên là kết quả sẽ có thể là chính xác hoặc là một lỗi ở đầu ra. Đối với các quá trình dừng (ergodic), chúng ta có thể xác định xác suất lỗi bằng 2 cách. Thứ nhất, chúng ta có thể xem một bit được truyền và tính  $P_e$  như là trung bình tập hợp (ensemble average), trong đó chúng ta có một tập hợp (ensemble) của các dạng sóng tạp âm có cùng tính chất thống kê. Cách thứ hai là chúng ta có thể xác định  $P_e$  như là trung bình thời gian bằng cách truyền vô hạn các dấu nhị phân và sử dụng hàm mẫu đơn của tạp âm. Điểm mẫu chốt trong mô phỏng là việc tính  $P_e$  dựa trên một số vô hạn các dấu nhị phân phát đi có thể thay bằng ước lượng  $P_e$  sử dụng một số hữu hạn các dấu nhị phân phát. Thực tế ước lượng của  $P_e$  là một biến ngẫu nhiên do mỗi hàm mẫu có khoảng hữu hạn sẽ tạo nên một giá trị khác nhau (mong muốn là không lớn lầm) cho xác suất lỗi. Điều này sẽ được trình bày ở phần sau khi chúng ta xem xét kỹ thuật Monte Carlo.

## 3.5 Các tham số liên quan đến mô phỏng

### 3.5.1 Các tham số đánh giá phẩm chất hệ thống

Có nhiều đại lượng được sử dụng để đánh giá phẩm chất một hệ thống. Đối với các hệ thống truyền dẫn thông tin thường tỉ số SNR (hay tỉ số CNR: Carrier-to-Noise Ratio) được sử dụng cho hệ thống tương tự và BER (hoặc có thể SER: Symbol Error Rate hay FER: Frame Error Rate) được sử dụng cho các hệ thống truyền dẫn

số. SNR có thể cũng được sử dụng làm tham số phụ cho đánh giá phẩm chất của hệ thống truyền dẫn số.

Tỉ số SNR được định nghĩa là tỉ số của công suất tín hiệu  $s(t)$  trên công suất của tạp âm  $z(t)$ , tức là

$$\text{SNR} = \frac{P_s}{P_z} = \frac{\text{E}\{s^2(t)\}}{\text{E}\{z^2(t)\}} \quad (3.4)$$

trong đó  $\text{E}\{\cdot\}$  biểu diễn phép toán lấy giá trị kỳ vọng.

Với hệ thống truyền dẫn số, tỉ số lỗi bit BER thường không thể xác định được mà được ước lượng nhờ mô phỏng Monte-Carlo. Giả thiết số  $K$  bit được truyền đi và xảy ra  $K_e$  bit lỗi ở đầu ra, tỉ số BER được định nghĩa là

$$\text{BER} = \frac{K_e}{K}. \quad (3.5)$$

Thực chất đây là ước lượng của xác suất lỗi bít được định nghĩa bởi

$$P_e = \lim_{K \rightarrow \infty} \frac{K_e}{K}. \quad (3.6)$$

Tương tự, tỉ lệ lỗi dấu (SER: Symbol Error Rate) có thể được định nghĩa là tỉ lệ giữa số dấu bị thu lỗi  $K_{s_e}$  trên tổng số dấu truyền  $K_s$

$$\text{SER} = \frac{K_{s_e}}{K_s}. \quad (3.7)$$

Với tỉ số lỗi khung chúng ta cũng có thể định nghĩa tương tự

$$\text{FER} = \frac{K_{f_e}}{K_f} \quad (3.8)$$

trong đó  $K_{f_e}$  là số khung thu lỗi và  $K_f$  là tổng số khung gửi đi.

### 3.5.2 Phân biệt tỉ lệ lỗi bít (BER) và xác suất lỗi bit

Xét hệ thống truyền dẫn số đơn giản ở Hình 3.1 và giả thiết rằng chúng ta cần tính tỉ số lỗi bit. Kỹ thuật mô phỏng cơ bản nhất để xác định đại lượng phẩm chất quan trọng này là gửi một số lớn các dấu số qua hệ thống và tính lỗi thu được ở đầu ra máy thu. Kỹ thuật này được gọi là mô phỏng Monte Carlo. Nếu  $K$  dấu được hệ thống xử lý và  $K_e$  lỗi đếm được ở đầu ra hệ thống thì ước lượng Monte Carlo của xác suất lỗi là [2]

$$\hat{P}_e = \frac{K_e}{K}. \quad (3.9)$$

Đại lượng này được gọi là BER theo  $K$  dấu. Ý nghĩa của BER là nó cho ta ước lượng của xác suất lỗi bít, mà theo định nghĩa tần suất tương đối của xác suất là

$$P_e = \lim_{K \rightarrow \infty} \frac{K_e}{K}. \quad (3.10)$$

Do một mô phỏng theo yêu cầu có thể chỉ cần xử lý một số hữu hạn các bít nên xác suất lỗi bít chỉ có thể xác định xấp xỉ.

Do thuật ngữ tỉ số lỗi bít và xác suất lỗi bit thường được dùng chung, có thể có lúng túng trong việc phân biệt hai khái niệm này. Thực chất hai khái niệm này hoàn toàn khác nhau. BER là ước lượng của xác suất lỗi bit. BER thực chất là một tỉ số (tỷ lệ), do nó mang ý nghĩa  $K_e$  lỗi trong  $K$  dấu truyền. Nếu xét một thí nghiệm truyền  $K$  dấu qua một kênh ngẫu nhiên (tập âm)  $L$  lần, thì số lỗi  $K_e$  đếm được trong mỗi lần thường khác nhau. Xác suất lỗi bit, tuy nhiên, lại là một số chứ không phải một biến ngẫu nhiên. Ví dụ, xác suất lỗi bit cho một hệ thống nhị phân PSK trên kênh AWGN là  $Q(\sqrt{2E_b/N_0})$  là cố định nếu  $E_b$  và  $N_0$  không đổi.

Thực tế là với  $K$  lớn, ước lượng  $\hat{P}_e$  hội tụ đến  $P_e$ , theo định nghĩa tần suất tương đối của xác suất.

### 3.5.3 Năng lượng và Công suất tín hiệu

Các thành phần năng lượng  $E_s$  và công suất tín hiệu  $P_s$  của  $s(t)$  được định nghĩa như sau

$$E_s = \int_{-\infty}^{\infty} s^2(t) dt \quad (3.11)$$

$$P_s = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} s^2(t) dt. \quad (3.12)$$

Trong trường hợp tín hiệu rời rạc (đã lấy mẫu), năng lượng và công suất tín hiệu được định nghĩa như sau

$$E_s = T_s \sum_{k=-\infty}^{\infty} s_k^2 \quad (3.13)$$

$$P_s = \lim_{K \rightarrow \infty} \frac{1}{2K+1} \sum_{k=-K}^{K} s_k^2 \quad (3.14)$$

trong đó  $T_s$  là thời gian của ký hiệu.

Nếu chuỗi tín hiệu là hữu hạn và chuỗi được phát lặp lại, thì

$$E_s = T_s \sum_{k=0}^{K-1} s_k^2 \quad (3.15)$$

$$P_s = \frac{1}{K} \sum_{k=0}^{K-1} s_k^2. \quad (3.16)$$

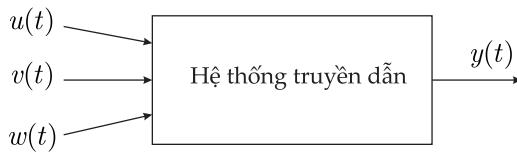
Trong Matlab công suất của chuỗi tín hiệu  $s_k$  được tính nhờ sử dụng hàm `norm.m` như sau: `Ps=norm(s)^2/length(s)`.

### 3.6 Mô phỏng Monte-Carlo trong thông tin số

Mô phỏng Monte-Carlo là một phương pháp mô phỏng ngẫu nhiên được sử dụng để đánh giá *lặp lại* chất lượng của một mô hình xác định sử dụng các tập biến đầu vào ngẫu nhiên. Phương pháp mô phỏng Monte-Carlo được bắt nguồn từ trò chơi may rủi được các casino tạo ra bằng các bộ tạo số ngẫu nhiên cơ khí. Tên gọi Monte-Carlo được lấy từ tên gọi của thành phố quê hương nổi tiếng của các sòng bạc Monte-Carlo. Kỹ thuật Monte-Carlo về bản chất là phương pháp mô phỏng một thí nghiệm ngẫu nhiên sử dụng các công cụ nhân tạo, tức là, không yêu cầu phải lặp lại thí nghiệm vật lý hoàn toàn.

Trong thông tin số, mô phỏng đòi hỏi phải tạo ra các mẫu có các dạng sóng khác nhau, xử lý các mẫu tín hiệu này qua các mô hình với các khối chức năng trong hệ thống, và ước lượng phẩm chất hệ thống từ các mẫu tín hiệu đầu ra tại các điểm trong mô hình. Trường hợp phổ biến trong thông tin số là sử dụng mô phỏng Monte-Carlo để ước lượng tỉ số lỗi bit (BER).

Giả thiết chúng ta có mô hình một hệ thống truyền dẫn được mô tả bởi các tín hiệu đầu vào  $u(t)$ ,  $v(t)$ , và  $w(t)$  là các quá trình ngẫu nhiên như Hình 3.5.



**Hình 3.5:** Mô hình mô phỏng Monte-Carlo [3].

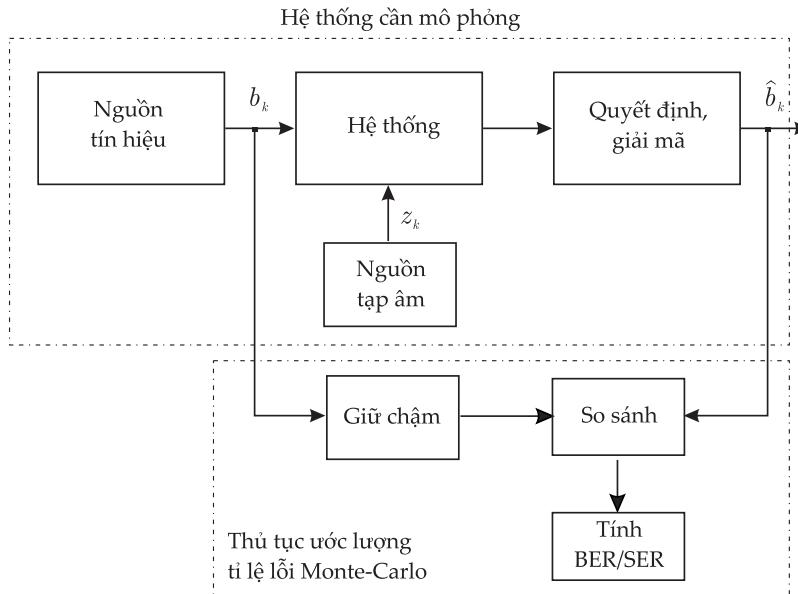
Mục tiêu của chúng ta là đi tìm các đặc tính thống kê của  $y(t)$  hay giá trị kỳ vọng của một hàm  $\mathcal{G}\{y(t)\}$  nào đó của  $y(t)$ . Nếu chúng ta giải bài toán này bằng cách giả lập hệ thống trong đó tất cả các dạng sóng biến đổi theo thời gian, chúng ta sẽ có một mô phỏng "thuần" Monte-Carlo. Điều này hàm ý việc tạo ra các mẫu của tất cả các quá trình đầu vào, cho các mô hình của các khối chức năng trong hệ thống làm việc với các mẫu đầu vào này, rồi quan sát các dạng sóng đầu ra. Trong trường hợp lý tưởng, mô phỏng Monte Carlo tương ứng với hệ thống thực trong phạm vi giới hạn của các giả thiết mô hình hóa và phép tính xấp xỉ. Giá trị kỳ vọng  $E[\mathcal{G}\{y(t)\}]$  được ước lượng từ mô phỏng Monte-Carlo theo công thức sau [3]:

$$E[\hat{\mathcal{G}}\{(y(t))\}] = \frac{1}{K} \sum_{k=1}^K \mathcal{G}\{y_k\} \quad (3.17)$$

trong đó dấu mũ biểu diễn giá trị ước lượng và  $K$  là số mẫu sử dụng mô phỏng.

Trong trường hợp mô hình mô phỏng hệ thống truyền dẫn số điển hình, sơ đồ mô tả việc thực hiện mô phỏng Monte-Carlo để ước lượng tỉ lệ lỗi bit BER hay tỉ lệ lỗi dấu SER được biểu diễn ở Hình 3.6. Mô phỏng Monte-Carlo được tiến hành theo các bước sau đây:

- Tạo các mẫu của chuỗi bít đầu vào  $b_k$ ,  $k = 1, 2, \dots, K$ , và các mẫu tạp âm  $z_k$ ,  $k = 1, 2, \dots, K$ .



**Hình 3.6:** Biểu diễn phương pháp mô phỏng Monte-Carlo.

- Xử lý các mẫu này qua mô hình các khối chức năng của hệ thống cần mô phỏng và tạo một chuỗi đầu ra  $\tilde{b}_k$ .
- Ước lượng tỉ số lỗi theo công thức sau

$$\hat{P}_e = \frac{1}{K} \sum_{k=1}^K \mathcal{G}\{\tilde{b}_k\} \quad (3.18)$$

trong đó  $\mathcal{G}\{\tilde{b}_k\} = 1$  nếu  $\tilde{b}_k \neq b_k$  và  $\mathcal{G}(y_k) = 0$  nếu  $\tilde{b}_k = b_k$ , tức là  $\mathcal{G}\{\tilde{b}_k\}$  là hàm tính lỗi.

Độ chính xác của các ước lượng thu được từ mô phỏng Monte-Carlo sẽ phụ thuộc vào thủ tục ước lượng sử dụng, kích thước mẫu  $K$ , khả năng tái tạo chính xác các mẫu đầu vào, và các giả thiết mô hình hóa và phép tính xấp xỉ.

### 3.6.1 Quan hệ giữa tính chính xác và số lượng mẫu

Để đánh giá phẩm chất của một hệ thống thông tin số, mô phỏng Monte-Carlo thường được sử dụng để ước lượng tỉ số BER. Như đã

thảo luận ở các phần trên, để có ước lượng chính xác của BER, chúng ta cần tăng số lượng bit gửi đi  $K$  đến vô cùng ( $\infty$ ). Tuy nhiên, việc tăng số lượng bit gửi đi đồng nghĩa với thời gian chạy mô phỏng sẽ lâu hơn. Vì vậy, để có được sự thỏa hiệp giữa độ chính xác của kết quả mô phỏng và thời gian chạy mô phỏng, chúng ta mong muốn tìm được số lượng bit cần gửi đi  $K$  tương ứng với giá trị BER mong muốn có độ chính xác cho phép. Chúng ta sẽ nghiên cứu tìm ra số bit  $K$  này thông qua trường hợp dưới đây.

Xét một hệ thống truyền dẫn đơn giản chịu ảnh hưởng của tạp âm. Giả thiết tín hiệu phát đi là  $s$  là một hằng số và tạp âm là  $z$  là một biến ngẫu nhiên Gauss với giá trị trung bình bằng 0 và phương sai  $\sigma_z^2 = 1$ . Do tạp âm  $z$  là ngẫu nhiên nên tín hiệu thu được

$$y = s + z \quad (3.19)$$

là một biến ngẫu nhiên. Để đơn giản ta xét trường hợp  $s > 0$ , tương đương với trường hợp gửi đi một bit thông tin ở mức điện áp cao. Chúng ta có thể dễ dàng nhận thấy  $y$  cũng là một biến ngẫu nhiên Gauss với giá trị trung bình  $s$  và phương sai bằng  $\sigma_y^2 = 1$ .

Giả thiết là chúng ta cần sử dụng mô phỏng Monte-Carlo để ước lượng xác suất thu được  $y < 0$  khi phát đi một tín hiệu  $s > 0$  cho trước, tức là, chúng ta cần ước lượng xác suất

$$P(s) = P(y < 0|s). \quad (3.20)$$

Trường hợp này tương tự như khi chúng ta gửi đi một tín hiệu  $s = +1$ , nhưng do ảnh hưởng của tạp âm Gauss  $z$  trên kênh truyền nên máy thu nhận được một mẫu tín hiệu  $y < 0$ . Để ước lượng tín hiệu phát  $s$ , máy thu sử dụng phương pháp tách tín hiệu hợp lý cực đại (maximum likelihood) thực hiện phép quyết định theo ngưỡng tại giá trị bằng 0 (tức là quyết định  $s = +1$  nếu  $y > 0$  và  $s = -1$  nếu  $y < 0$ ) và cho ta kết quả  $s = -1$ . Trường hợp này xảy ra một lỗi tách tín hiệu và xác suất  $P(s) = P(y < 0|s)$  chính là xác suất thu lỗi của hệ thống.

Để ước lượng được xác suất lỗi  $P(s)$ , chúng ta sử dụng máy tính để tạo ra một chuỗi các biến Gauss ngẫu nhiên, có kỳ vọng

bằng 0 và phương sai bằng 1, độc lập thống kê và có phân bố giống nhau (i.d.d.),  $z_k, k = 1, 2, \dots, N$ . Cộng  $z_k$  với  $s$ , chúng ta có một chuỗi  $K$  biến ngẫu nhiên

$$y_k = s + z_k, \quad k = 1, 2, \dots, K. \quad (3.21)$$

Từ các biến ngẫu nhiên tạo ra bằng máy tính này, chúng ta mong muốn ước lượng xác suất một biến ngẫu nhiên Gauss, với kỳ vọng  $s > 0$  và phương sai đơn vị, nhỏ hơn 0, tức là, ước lượng xác suất  $y_k < 0$ . Để thu được ước lượng xác suất, chúng ta làm phép kiểm tra xem  $y_k < 0$  và định nghĩa một biến ngẫu nhiên mới  $x_k$  có quan hệ với  $y_k$  như sau

$$e_k = \begin{cases} 0, & \text{nếu } y_k \geq 0 \\ 1, & \text{nếu } y_k < 0 \end{cases}. \quad (3.22)$$

Có thể nhận thấy rằng  $e_k$  chính là biến biểu diễn lỗi, và ước lượng của xác suất  $P(y < 0|s)$  có thể biểu diễn như sau

$$\hat{P}(s) = \frac{1}{K} \sum_{k=1}^K e_k. \quad (3.23)$$

Nói cách khác, ước lượng  $\hat{P}(s)$  đơn giản là số các biến ngẫu nhiên  $y_k, k = 1, 2, \dots, K$  nhỏ hơn 0 chia cho tổng số  $K$  biến ngẫu nhiên.

Do ước lượng  $\hat{P}(s)$  là một hàm số của các biến ngẫu nhiên  $e_k, k = 1, 2, \dots, K$ , nên nó cũng là một biến ngẫu nhiên. Để có được ước lượng của xác suất lỗi  $\hat{P}(s)$  gần với giá trị của xác suất lỗi  $P(s)$ , chúng ta mong muốn  $E[\hat{P}(s)] = P(s)$  và  $\sigma_{\hat{P}(s)} \ll P(s)$ .

Chúng ta bắt đầu bằng việc xét giá trị kỳ vọng của  $\hat{P}(s)$

$$E[\hat{P}(s)] = \frac{1}{K} \sum_{k=1}^K E[e_k]. \quad (3.24)$$

Nhưng do

$$E[e_k] = 0 \cdot P(y_k > 0) + 1 \cdot P(y_k < 0) \quad (3.25)$$

nên

$$\text{E}[\hat{P}(s)] = \frac{1}{K} \sum_{k=1}^K P(y_k < 0) \quad (3.26)$$

$$= \frac{1}{K} \sum_{k=1}^K P(s) = P(s). \quad (3.27)$$

Điều này chứng tỏ rằng giá trị kỳ vọng của ước lượng  $\hat{P}(s)$  đúng bằng giá trị thật  $P(s)$ .

Tiếp theo, chúng ta đi tính phương sai của ước lượng  $\hat{P}(s)$ . Phương sai của  $\hat{P}(s)$  được định nghĩa như sau

$$\sigma_{\hat{P}(s)}^2 = \text{E}[\hat{P}(s) - \text{E}[\hat{P}(s)]]^2 \quad (3.28)$$

$$= \text{E}[\hat{P}^2(s)] - P^2(s). \quad (3.29)$$

Tuy nhiên,

$$\text{E}[\hat{P}^2(s)] = \text{E}\left[\frac{1}{K^2} \sum_{k=1}^K \sum_{\ell=1}^K e_k e_\ell\right] \quad (3.30)$$

$$= \frac{1}{K^2} \sum_{k=1}^K \text{E}[e_k^2] + \frac{1}{K^2} \sum_{k=1}^K \sum_{\ell=1, \ell \neq k}^N \text{E}[e_k e_\ell]. \quad (3.31)$$

Để ý rằng chúng ta có

$$\text{E}[e_k^2] = 0 \cdot P(y_k \geq 0) + 1 \cdot P(y_k < 0) = P(s) \quad (3.32)$$

và

$$\text{E}[e_k e_\ell] = \text{E}[e_k] \text{E}[e_\ell] = P^2(s) \quad (3.33)$$

do  $e_k$  và  $e_\ell$  độc lập với nhau. Thay (3.32) và (3.33) vào (3.31), chúng ta thu được phương sai của ước lượng như sau

$$\text{E}[\hat{P}^2(s)] = \frac{1}{K} P(s) + \frac{K(K-1)}{K^2} P^2(s) \quad (3.34)$$

$$= \frac{1}{K} P(s)[1 + (K-1)P(s)]. \quad (3.35)$$

Như vậy, chúng ta có phương sai của  $\hat{P}(s)$

$$\sigma_{\hat{P}(s)}^2 = \frac{1}{K} P(s)[1 + (K-1)P(s)] - P^2(s) \quad (3.36)$$

$$= \frac{1}{K} P(s)[1 - P(s)]. \quad (3.37)$$

Như đã đề cập ở phần trên, khi ước lượng  $P(s)$  dựa vào mô phỏng chúng ta mong muốn độ lệch chuẩn  $\sigma_{P(s)}$  càng nhỏ so với  $P(s)$  càng tốt. Tức là

$$\sigma_{P(s)} = \left[ \frac{1}{K} P(s)[1 - P(s)] \right]^{1/2} \ll P(s) \quad (3.38)$$

hay, một cách tương đương,

$$\sigma_{P(s)}^2 \ll P^2(s)$$

tức là

$$\frac{1}{K} P(s)[1 - P(s)] \ll P^2(s). \quad (3.39)$$

Vì vậy

$$K \gg \frac{1 - P(s)}{P(s)} = \frac{1}{P(s)} - 1. \quad (3.40)$$

Do xác suất lỗi thường rất nhỏ hơn 1 nên  $1/P(s) \gg 1$  và chúng ta có

$$K \gg \frac{1}{P(s)}. \quad (3.41)$$

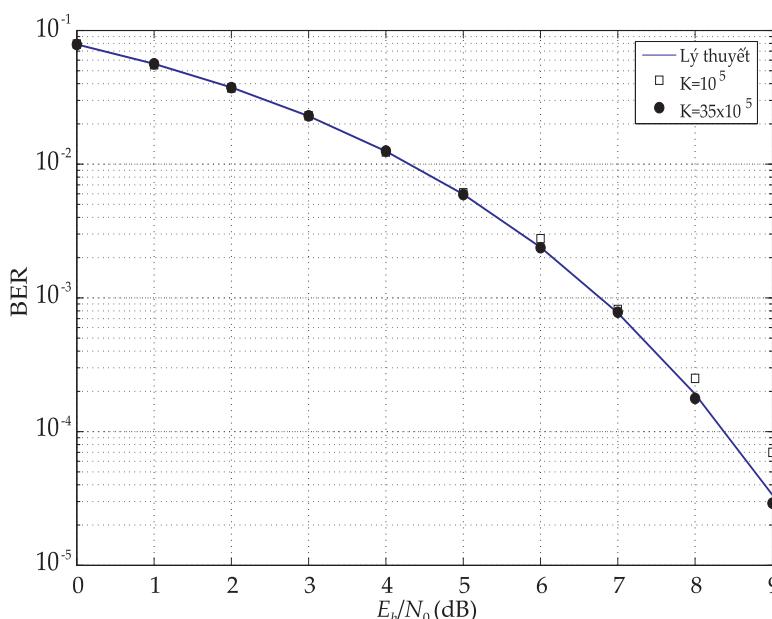
Như vậy, để thu được ước lượng của xác suất lỗi  $\hat{P}(s)$  chính xác so với  $P(s)$ , chúng ta cần chọn  $K \gg 1/P(s)$ . Ví dụ, với  $P(s) = 10^{-3}$ , thì yêu cầu  $K \gg 1000$ . Để ý rằng xác suất lỗi  $P(s) = 10^{-3}$  có nghĩa là khi chúng ta gửi đi  $K = 10^3$  bit thì trung bình sẽ có một bit lỗi. Nếu chúng ta chọn  $K = 10^4$  thì trung bình sẽ có 10 bit lỗi, và tương tự khi tăng số bit  $K$ . Thực tế, theo kinh nghiệm cho thấy kết quả mô phỏng có thể chấp nhận được khi chúng ta đếm được 10 bit lỗi trở lên, và vì vậy, số lượng mẫu nhỏ nhất để thu được ước lượng tin cậy của  $P(s)$  cần thỏa mãn [2]

$$K > \frac{10}{P(s)} \quad (3.42)$$

khi  $P(s) \ll 1$ . Điều kiện này thường đủ để ước lượng xác suất lỗi của một hệ thống truyền dẫn số chịu ảnh hưởng của tạp âm cộng và nhiễu khác.

### 3.6.2 Ý nghĩa của số lượng mẫu trong mô phỏng Monte-Carlo

Từ điều kiện công thức (3.42), chúng ta thấy rằng giả thiết chúng ra cần ước lượng tỉ số lỗi bít bằng  $10^{-3}$  thì, tối thiểu, chúng ta cần gửi đi nhiều hơn  $10 \times 10^3$  bit dữ liệu để có thể thu được kết quả với độ chính xác chấp nhận được. Để chứng minh kết luận này chúng ta xét các kết quả mô phỏng so với lý thuyết cho hệ thống truyền dẫn qua kênh tạp âm AWGN như ở Hình 3.1. Trên Hình 3.7 kết quả đường连线 là kết quả xác suất lỗi tính được bằng giải tích sử dụng công thức (3.1), kết quả đánh dấu bằng hình vuông ( $\square$ ) là kết quả mô phỏng sử dụng  $K = 10^5$  bit, và kết quả đánh dấu bằng các hình tròn ( $\bullet$ ) là kết quả thu được với  $K = 35 \times 10^5$  bit.



Hình 3.7: Ảnh hưởng của số lượng mẫu  $N$  tới độ chính xác của kết quả mô phỏng.

Dựa theo kết quả lý thuyết tại  $E_b/N_0 = 9$  dB chúng ta có tỉ số BER =  $3.5 \times 10^{-5}$ . Sử dụng điều kiện (3.42) thì số bit cần gửi đi tối thiểu là  $K = 35 \times 10^5$  bit. Như vậy, chúng ta có thể thấy ngay trường hợp mô phỏng sử dụng  $K = 10^5$  bit sẽ không cho kết quả chính xác. Điều này thấy rõ trên hình vẽ khi các điểm  $\square$  bị lệch rất xa các giá trị lý

thuyết tại vùng  $E_b/N_0$ . Với trường hợp sử dụng  $K = 35 \times 10^5$  thì kết quả khá chính xác, ngoại trừ kết quả tại  $E_b/N_0 = 9$  dB bị lệch xuông một chút. Nếu sử dụng  $K > 35 \times 10^5$ , ví dụ  $K = 50 \times 10^5$  chúng ta sẽ thu được kết quả gần như chính xác hoàn toàn.

Để xây dựng được chương trình mô phỏng có kết quả chính xác chúng ta có thể sử dụng hai phương pháp sau đây:

- **Sử dụng vòng lặp while:**

- (1) Đặt số lỗi  $K_e > 10$ , VD,  $K_e = 100$ , tổng số bit truyền  $K = 1$ .
- (2) Với mỗi giá trị  $E_b/N_0$  cho trước.
- (3) WHILE  $K_e > 0$ 
  - Tạo các bit dữ liệu ngẫu nhiên và truyền vào hệ thống.
  - Tạo các mẫu tạp âm tương ứng với  $E_b/N_0$ .
  - Kích hoạt thủ tục đếm lỗi.
  - $K_e = K_e - 1$ .
- (4) END.
- (5) Tính BER =  $K_e/K$ .

- **Sử dụng vòng lặp for**

- (1) Đặt số tổng số bit truyền, VD,  $K = 10^5$ .
- (2) Với mỗi giá trị  $E_b/N_0$  cho trước.
- (3) Tạo các bit dữ liệu ngẫu nhiên và truyền vào hệ thống.
- (4) Tạo các mẫu tạp âm tương ứng với  $E_b/N_0$ .
- (5) Kích hoạt thủ tục đếm lỗi  $N_e$ .
- (6) Tính BER =  $K_e/K$ .

### 3.7 Mô phỏng bằng thông và mô phỏng bằng gốc

Một trong các yêu cầu để mô phỏng các hệ thống truyền thông, là cần phải tạo ra dạng sóng của tín hiệu mong muốn giống như hệ thống cần nghiên cứu. Các tín hiệu truyền đi thường là các tín hiệu

điều chế cao tần được xử lý ở các hệ thống băng thông. Tuy nhiên, do máy tính chỉ có khả năng xử lý các giá trị biểu diễn các mẫu tín hiệu rời rạc nên bước đầu tiên trong quá trình tạo lập mô hình mô phỏng là cần biểu diễn chính xác dạng sóng tín hiệu băng máy tính thông qua biến đổi tương tự-số.

Thao tác đầu tiên trong biến đổi tương tự-số được biết đến rộng rãi là lấy mẫu. Từ định lý lấy mẫu của Nyquist chúng ta biết rằng một tín hiệu liên tục chỉ có thể biểu diễn duy nhất bởi một mô hình rời rạc chỉ khi tần số lấy mẫu lớn hơn hoặc bằng hai lần thành phần tần số lớn nhất trong phổ tần tín hiệu. Đối với tín hiệu băng gốc, giả thiết thành phần tần số lớn nhất  $f_{\max}$ , thì tần số lấy mẫu  $f_s \geq 2f_{\max}$ . Đối với tín hiệu băng thông có phổ tần tín hiệu nằm trong dải  $f_c - B/2 \leq f \leq f_c + B/2$  thì thành phần tần số lớn nhất là  $f_c + B/2$ . Dựa theo định lý lấy mẫu băng thông [1] thì tần số lấy mẫu không cần phải thỏa mãn điều kiện  $f_s \geq 2f_c + B$ , mà chỉ cần thỏa mãn điều kiện

$$f_s \geq \frac{2f_{\max}}{m} = \frac{2f_c + B}{m}, \quad (3.43)$$

trong đó

$$m = \left\lceil \frac{f_{\max}}{B} \right\rceil \quad (3.44)$$

tức là  $m$  là kết quả của phép tính  $\frac{f_{\max}}{B}$  lấy làm tròn xuồng [1]. Điều này có nghĩa là  $2B \leq f_s \leq 4B$  và trong trường hợp  $f_0 \gg B$  thì  $f_s \approx 2B$ , tuy nhiên không nhỏ hơn  $2B$ .

Trên cơ sở mô hình tín hiệu rời rạc cho tín hiệu băng gốc và tín hiệu băng thông có thể thực hiện tương ứng các *mô phỏng băng gốc* và *mô phỏng băng thông*. Tuy nhiên, biểu diễn của tín hiệu sử dụng các mẫu của dạng sóng băng thông thường phức tạp hơn các tín hiệu băng gốc [3]. Vì vậy, tín hiệu và hệ thống băng thông thường được biểu diễn ở dạng băng gốc tương đương trong mô phỏng. Phương pháp biểu diễn tín hiệu băng thông về dạng băng gốc tương đương được gọi là phương pháp *đường bao phíc*. Ý tưởng của phương pháp

này là với một tín hiệu điều chế sóng mang  $x(t)$  ở dạng

$$x(t) = a(t) \cos[2\pi f_c t + \phi(t)] \quad (3.45)$$

trong đó  $a(t)$  là biên độ điều chế,  $\phi(t)$  là pha điều chế của tín hiệu, còn  $f_c$  là tần số sóng mang, chúng ta có thể viết lại thành

$$\begin{aligned} x(t) &= \Re[a(t)e^{j(2\pi f_c t + \phi(t))}] \\ &= \Re[a(t)e^{j\phi(t)}e^{j2\pi f_c t}] \end{aligned} \quad (3.46)$$

trong đó  $\Re[\cdot]$  biểu diễn phép tính lấy phần thực của thành phần phức bên trong dấu ngoặc. Để ý rằng thành phần tín hiệu

$$\tilde{x}(t) = a(t)e^{j\phi(t)} \quad (3.47)$$

chứa tất cả các thông tin và là tín hiệu băng gốc. Tín hiệu  $\tilde{x}(t)$  thường được gọi là *biên độ phức* hay *tín hiệu băng gốc tương đương* của tín hiệu băng thông. Trong mô phỏng tín hiệu băng gốc tương đương ở công thức (3.47) thay thế cho tín hiệu băng thông công thức (3.45). Việc mô hình hóa tín hiệu băng thông ở dạng băng gốc tương đương thường liên quan đến biến đổi Hilbert. Việc tính toán biến đổi Hilbert trong miền tần số thường dễ dàng hơn trong miền thời gian. Tuy nhiên, trong thực tế với phần lớn tín hiệu băng thông điều chế biên độ hoặc pha có băng thông hữu hạn và có  $f_c \gg B$ . Vì vậy, để mô hình hóa tín hiệu cho mô phỏng, tín hiệu băng gốc tương đương cho các tín hiệu điều chế này có thể nhận được bằng cách ánh xạ thành phần đồng pha (inphase) của tín hiệu băng thông trở thành phần thực và thành phần trực giao (quadrature) trở thành phần ảo của biên độ phức. Tức là

$$\tilde{x}(t) = a(t)e^{j\phi(t)} = x_I(t) + jx_Q(t) \quad (3.48)$$

trong đó

$$x_I(t) = a(t) \cos \phi(t) \quad (3.49)$$

là thành phần đồng pha, còn

$$x_Q(t) = a(t) \sin \phi(t) \quad (3.50)$$

là thành phần trực giao.

### 3.8 Tóm tắt chương

Trong chương này đã trình bày các khái niệm về mô phỏng, vai trò mô phỏng máy tính, phân loại các phương pháp mô phỏng. Nội dung chương cũng đã trình bày lý thuyết chung về mô phỏng Monte-Carlo và điều kiện số mẫu tín hiệu cần thiết sử dụng để mô phỏng Monte-Carlo cho kết quả chính xác. Các điều kiện và phương pháp thực hiện mô phỏng Monte-Carlo trình bày trong chương này sẽ là cơ sở cho các ví dụ mô phỏng ở các chương sau. Phần cuối của chương đưa ra nguyên lý thực hiện mô phỏng bằng thông so với mô phỏng bằng gốc.

### Tài liệu tham khảo

- [1] W. H. Tranter, K. S. Shanmugan, T. S. Rappaport, and K. L. Kosbar, *Principle of Communication with Wireless Applications*. Prentice-Hall, 2004.
- [2] J. G. Proakis, M. Salehi, and G. Bauch, *Contemporary Communication System using Matlab*. Thomson, 2004.
- [3] M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, Eds., *Simulation of Communication Systems: Modeling, Methodology and Techniques*. Norwell, MA, USA: Kluwer Academic Publishers, 2000.

## **Phần II**

# **Cơ sở lý thuyết kênh truyền và các phương pháp điều chế**



## Chương 4

# Kênh thông tin vô tuyến

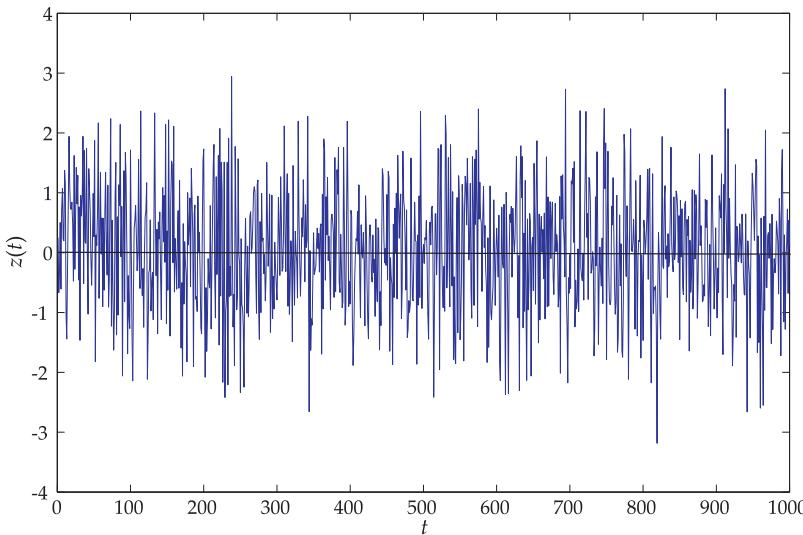
## 4.1 Kênh tạp âm AWGN

### 4.1.1 Tạp âm AWGN

Thuật ngữ *tạp âm* (noise) mô tả các tín hiệu điện không mong muốn xuất hiện trong hệ thống. Sự xuất hiện của tạp âm làm giảm khả năng tách chính xác các tín hiệu phát và, vì vậy, làm giảm tốc độ truyền dẫn thông tin. Tạp âm được tạo ra từ nhiều nguồn khác nhau, nhưng có thể phân loại thành hai nguồn chính là *nhân tạo* và *tự nhiên* [1]. Nguồn tạp âm nhân tạo xuất hiện từ các nguồn đánh lửa, chuyển mạch hay các phát xạ điện tử. Tạp âm tự nhiên gồm tạp âm xuất hiện trong các mạch hay linh kiện điện tử, xáo động khí quyển hay các nguồn thiên hà.

Việc thiết kế tốt các mạch điện, thiết bị hay hệ thống cho phép loại bỏ hoặc giảm thiểu ảnh hưởng của các tạp âm bằng cách nối đất, chọn vị trí đặt thiết bị hay sử dụng các phương pháp lọc. Tuy nhiên, có một nguồn tạp âm tự nhiên không thể loại bỏ là *tạp âm nhiệt*. Tạp âm nhiệt xuất hiện do chuyển động nhiệt của các điện tử ở trong tất cả các linh kiện điện tử như đèn tròn, dây dẫn hay các phần tử dẫn điện khác. Sự chuyển động *ngẫu nhiên* và *độc lập* của vô hạn các điện tử tạo nên các đặc tính thống kê Gauss theo định lý giới hạn trung tâm (central limit theorem). Vì vậy, tạp âm nhiệt có

thể mô tả như một quá trình ngẫu nhiên Gauss có giá trị kỳ vọng bằng không. Một ví dụ về tạp âm Gauss với giá trị trung bình 0 và phương sai  $\sigma_z^2 = 1$  được miêu tả ở Hình vẽ 4.1.



**Hình 4.1:** Một ví dụ về tạp âm Gauss với giá trị trung bình 0 và phương sai  $\sigma_z^2 = 1$ .

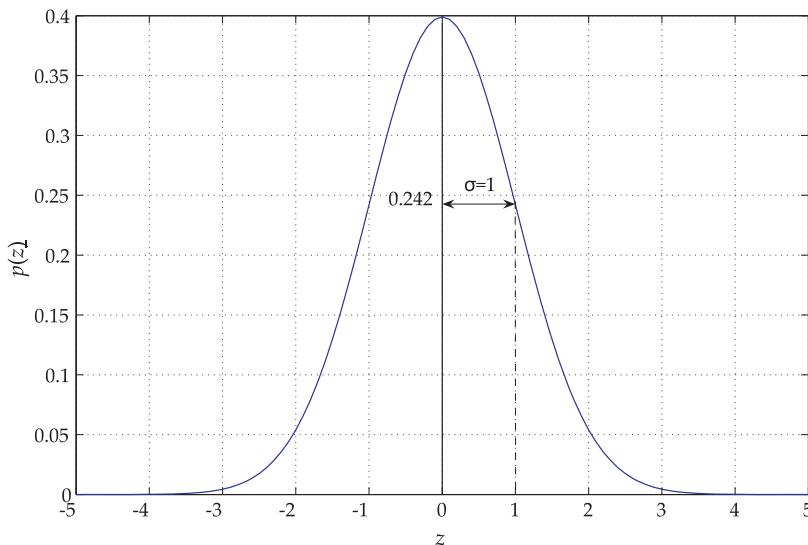
Hàm mật độ xác suất (PDF: Probability Density Function) của một quá trình ngẫu nhiên Gauss  $z(t)$  được biểu diễn như sau

$$p(z) = \frac{1}{\sigma_z \sqrt{2\pi}} \exp\left(-\frac{z^2}{2\sigma_z^2}\right) \quad (4.1)$$

Hình vẽ 4.2 biểu diễn hàm PDF Gauss với giá trị trung bình bằng không và độ lệch chuẩn (standard deviation)  $\sigma_z = 1$ .

Một đặc tính quan trọng của tạp âm Gauss có giá trị trung bình bằng không là phương sai  $\sigma_z^2$  bằng trung bình bình phương của  $z(t)$ , tức là,  $\sigma_z^2 = E\{z^2(t)\}$ .

**Tạp âm trắng:** một đặc tính quan trọng của tạp âm nhiệt là mật độ phổ tần số của nó như nhau tại mọi tần số. Tức là, nó là nguồn tạp âm phát ra một lượng công suất như nhau trên một đơn vị băng

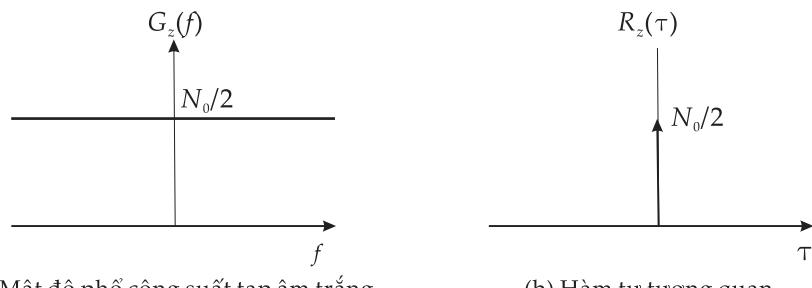


**Hình 4.2:** Hàm mật độ xác suất Gauss với  $\sigma_z^2 = 1$ .

tần tại tất cả các tần số bằng

$$G_z(f) = \frac{N_0}{2} \quad [\text{W/Hz}] \quad (4.2)$$

như mô tả ở [Hình vẽ 4.3\(a\)](#). Hệ số 2 trong công thức trên chỉ thị rằng  $G_z(f)$  là một hàm mật độ phổ công suất 2 phía (two-sided power spectral density function) còn  $N_0$  thì được gọi là mật độ phổ công suất tạp âm. Tạp âm với công suất có mật độ phổ đều như vậy được gọi là *tạp âm trắng* (white noise).



(a) Mật độ phổ công suất tạp âm trắng

(b) Hàm tự tương quan

**Hình 4.3:** Mật độ phổ công suất và hàm tự tương quan của tạp âm trắng.

Hàm tự tương quan của tạp âm trắng là phép biến đổi Fourier

ngược của mật độ phổ công suất tạp âm cho bởi

$$R_z(\tau) = \mathfrak{F}^{-1}\{G_z(f)\} = \int_{-\infty}^{\infty} G_z(f) e^{j2\pi f\tau} df \quad (4.3)$$

$$= \frac{N_0}{2} \delta(\tau) \quad (4.4)$$

như biểu diễn ở Hình vẽ 4.3(b). Tức là, hàm tự tương quan của tạp âm trắng là một hàm xung delta tại  $\tau = 0$  được nhân trọng số với  $N_0/2$ . Để ý rằng  $R_z(\tau) = 0$  với mọi  $\tau \neq 0$  nên bất kỳ hai mẫu khác nhau nào của tạp âm trắng đều không tương quan với nhau bất kể chúng gần nhau đến mức nào. Do tạp âm nhiệt được cộng với tín hiệu nên nó còn được gọi là tạp âm cộng tính (additive noise). Tổng hợp các đặc tính của tạp âm nhiệt ở trên chúng ta có thể tóm tắt lại rằng tạp âm nhiệt trong các hệ thống thông tin là tạp âm Gauss trắng cộng tính (AWGN: Additive White Gaussian Noise).

#### 4.1.2 Mô phỏng tạp âm AWGN

Trong MATLAB, chúng ta có thể sử dụng hàm có sẵn randn để mô phỏng tạp âm AWGN. Hàm randn cho phép tạo ra các biến ngẫu nhiên theo phân bố chuẩn chính tắc (standard normal distribution) với giá trị trung bình 0, phương sai 1 và độ lệch chuẩn 1.

Hình 4.4 biểu diễn hàm PDF của tạp âm AWGN tạo nên bằng mô phỏng sử dụng hàm randn sử dụng chương trình Matlab Program. 4.1.

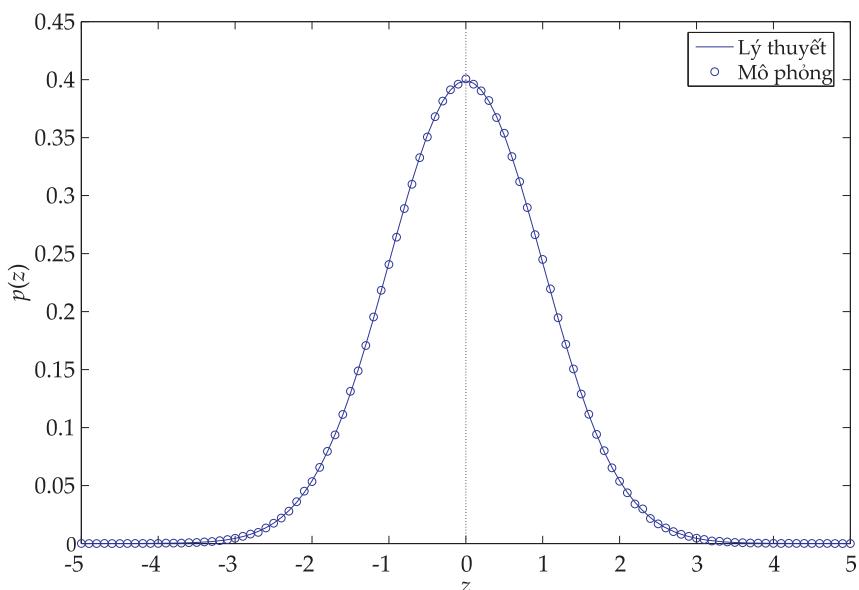
##### ■ MATLAB Program 4.1 SimAWGN.m

```
%% Script to describe AWGN
% Plot time version of AWGN
N=10^6;
t=1:N;
z=randn(1,N);
plot(t,z)
xlabel('t')
ylabel('z(t)')
axis([0 1000 -4 4])
```

```
% Plot PDF of the normal distribution by theory
step=0.1;
x=-5:step:5;
sigma=1;
px=1/sqrt(2*pi)*exp(-1/2*x.^2/sigma^2);
figure(2)
plot(x,px)
xlabel('z')
ylabel('p(z)')
hold on

% Plot PDF of simulated AWGN
z=randn(1,N);
h=hist(z,x);
plot(x,h/N/step,'o');
legend('Theory','Simulated')
%!
```

---



**Hình 4.4:** Hàm phân bố của tạp âm AWGN.

Do phân bố chuẩn chính tắc là một trường hợp đặc biệt của phân bố Gauss trong đó giá trị trung bình bằng 0 và phương sai bằng 1, nên để mô phỏng tạp âm AWGN với phương sai bằng  $N_0$  chúng ta chỉ việc nhân hàm `randn` với  $\sigma_z$ , tức là độ lệch chuẩn mong muốn. Do tạp âm mô phỏng trong các hệ thống băng thông là một biến số phức gồm hai thành phần thực và ảo, nên công suất tạp âm là tổng công suất của từng thành phần, tức là,  $N_0 = 2\sigma_z^2$ . Do đó để tạo được tạp âm AWGN phức với mật độ phổ công suất  $N_0$  chúng ta cần tạo ra tạp âm AWGN có phương sai của từng phần thực và ảo bằng nhau và bằng  $\sigma_z^2 = N_0/2$ . Ví dụ, để tạo được 1000 dấu tạp âm phức có công suất  $N_0$  chúng ta làm như sau:

$$z = \text{sqrt}(N_0/2) * (\text{randn}(1, 1000) + j * \text{randn}(1, 1000))$$

#### 4.1.3 Kênh tạp âm AWGN

Kênh AWGN là kênh truyền thông trong đó ảnh hưởng của AWGN là chính. Kênh tạp âm AWGN thường xuất hiện trong các đường truyền hữu tuyến không chịu ảnh hưởng của xuyên âm (crosstalk) hoặc trong các đường truyền vô tuyến có sử dụng ăng-ten định hướng với búp sóng hẹp như kênh truyền vi-ba hay kênh truyền thông tin vệ tinh.

## 4.2 Kênh pha-định

Hình vẽ 4.5 mô tả một đường liên lạc giữa ăng-ten trạm gốc (BS: Base Station) và ăng-ten trạm di động (MS: Mobile Station). Xung quanh MS có nhiều vật phản xạ như nhà, cây, đồi núi, ...trong khi xung quanh BS lại có rất ít hoặc không có các vật phản xạ do ăng-ten trạm BS được đặt trên cao. Các vật phản xạ này được gọi chung là vật tán xạ. Liên lạc giữa BS và MS thông qua nhiều đường (path), mỗi đường chịu một hay nhiều phản xạ, và tín hiệu đến máy thu là tín hiệu tổng hợp từ tất cả các đường này. Do các đường có biên độ, pha, và độ trễ khác nhau, nên tín hiệu truyền qua các đường có thể

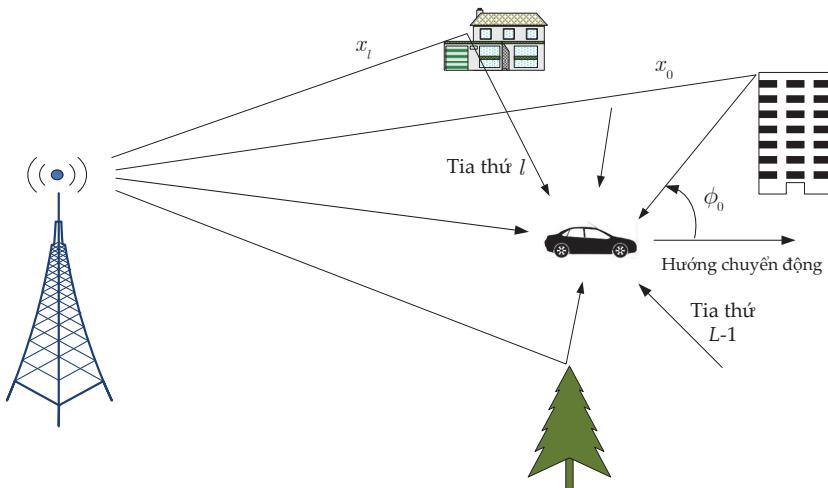
kết hợp với nhau một cách có lợi hoặc không có lợi, tạo nên một sóng đứng ngẫu nhiên. Hiện tượng này được gọi là truyền sóng pha-định đa đường. Kênh truyền sóng kiểu này được gọi là kênh pha-định đa đường.

#### 4.2.1 Mô hình toán học của pha-định

Tín hiệu vô tuyến luôn là tín hiệu băng thông (bandpass) và thường có băng thông hẹp (narrowband). Tín hiệu băng thông phát đi tại tần số sóng mang  $f_c$  với đường bao phức  $\check{s}(t)$ <sup>4</sup> được biểu diễn như sau

$$s(t) = \Re \{ \check{s}(t) e^{j2\pi f_c t} \} \quad (4.5)$$

trong đó  $\Re \{ \cdot \}$  biểu diễn phép toán lấy phần thực.



Hình 4.5: Mô hình truyền sóng đa đường.

Giả thiết có  $L$  vật phản xạ xung quanh, và độ dài của đường  $l$  là  $x_l$ . Ký hiệu  $c$  là tốc độ ánh sáng (cũng chính là tốc độ lan truyền của sóng điện từ trong không khí), thì thời gian truyền sóng từ BS tới MS là  $x_l/c$ . Giả sử độ suy hao của đường  $l$  là  $a_l$ , thì tín hiệu thu

<sup>4</sup>  $\check{s}(t)$  còn được gọi là tín hiệu băng tần gốc tương đương

được tại MS không tính đến tạp âm là

$$y(t) = \sum_{l=0}^{L-1} a_l s\left(t - \frac{x_l}{c}\right). \quad (4.6)$$

Thay (4.5) vào (4.6) chúng ta có

$$y(t) = \sum_{l=0}^{L-1} a_l \Re \left\{ \check{s}\left(t - \frac{x_l}{c}\right) e^{j2\pi f_c(t - \frac{x_l}{c})} \right\} \quad (4.7)$$

$$= \Re \left\{ \sum_{l=0}^{L-1} a_l \check{s}\left(t - \frac{x_l}{c}\right) e^{j2\pi f_c(t - \frac{x_l}{c})} \right\}. \quad (4.8)$$

Viết lại  $y(t)$  ở dạng

$$y(t) = \Re \left\{ \sum_{l=0}^{L-1} a_l \check{s}\left(t - \frac{x_l}{c}\right) e^{-j2\pi f_c \frac{x_l}{c}} e^{j2\pi f_c t} \right\} \quad (4.9)$$

$$= \Re \left\{ \check{y}(t) e^{j2\pi f_c t} \right\} \quad (4.10)$$

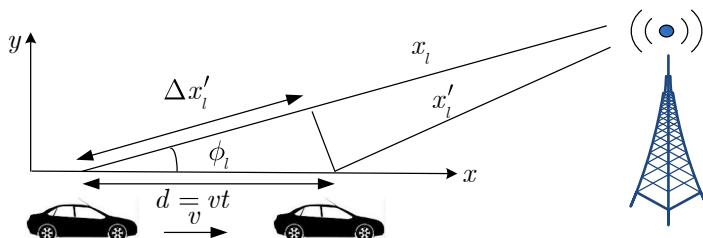
trong đó thành phần đường bao tín hiệu thu  $\check{y}(t)$  là tổng của  $L$  thành phần tín hiệu có độ trễ, suy hao và dịch pha khác nhau

$$\check{y}(t) = \sum_{l=0}^{L-1} a_l \check{s}(t - \tau_l) e^{-j2\pi f_c \tau_l} \quad (4.11)$$

còn  $\tau_l = \frac{x_l}{c}$  là thời gian trễ của đường truyền thứ  $l$ .

Để ý là tổng  $L$  thành phần đa đường có thể bao gồm cả tia truyền thẳng (LOS: Line of Sight) và các tia tán xạ.

#### 4.2.2 Ảnh hưởng của chuyển động của MS



Hình 4.6: Sự lan truyền của đường  $l$  tới một trạm MS.

Hình vẽ 4.6 minh họa sự chuyển động của một trạm MS theo trục  $x$  trong mặt phẳng  $x - y$  với vận tốc  $v$ . Ở đây, chúng ta đã giả thiết rằng ăng ten phát được phân cực một cách phù hợp sao cho véc-tơ điện trường song song với trục  $z$ . Khi MS chuyển động với tốc độ  $v$ , độ dài đường truyền sóng thứ  $l$  thay đổi một khoảng là

$$x'_l = x_l - \Delta x_l. \quad (4.12)$$

Nếu góc tới của đường thứ  $l$  so với hướng chuyển động là  $\phi_l$ , chúng ta có

$$\Delta x_l \approx vt \cos(\phi_l). \quad (4.13)$$

Như vậy, tín hiệu thu được khi không có tạp âm là

$$\begin{aligned} y(t) &= \Re \left\{ \sum_{l=0}^{L-1} a_l e^{j2\pi f_c \left( t - \frac{x_l - \Delta x_l}{c} \right)} \check{s} \left( t - \frac{x_l - \Delta x_l}{c} \right) \right\} \\ &= \Re \left\{ \sum_{l=0}^{L-1} a_l e^{j2\pi f_c t} e^{-j2\pi f_c \frac{x_l}{c}} e^{j2\pi f_c \frac{v \cos(\phi_l)}{c} t} \check{s} \left( t - \frac{x_l}{c} + \frac{v \cos(\phi_l)}{c} t \right) \right\}. \end{aligned} \quad (4.14)$$

Để ý là do thay đổi về trễ tín hiệu  $\frac{v \cos(\phi_l)}{c} t$  thường rất nhỏ so với thang thời gian của tín hiệu điều chế  $\check{s}(t)$  nên có thể bỏ qua thành phần này trong  $\check{s}(t)$  [2]

$$y(t) = \Re \left\{ \sum_{l=0}^{L-1} a_l e^{j2\pi f_c t} e^{-j2\pi f_c \frac{x_l}{c}} e^{j2\pi f_c \frac{v \cos(\phi_l)}{c} t} \check{s} \left( t - \frac{x_l}{c} \right) \right\}. \quad (4.15)$$

Gọi

$$f_D = \frac{f_c}{c} v \quad (4.16)$$

là tần số Doppler lớn nhất. Để ý rằng  $e^{j2\pi f_D \cos(\phi_l) \tau_l} e^{-j2\pi f_D \cos(\phi_l) \tau_l} = 1$  nên chúng ta có thể bổ sung thêm thành phần về trái vào công thức (4.15) để thu được

$$y(t) = \Re \left\{ \sum_{l=0}^{L-1} a_l e^{j2\pi f_D \cos(\phi_l) \tau_l} e^{j2\pi [f_c + f_D \cos(\phi_l)] (t - \tau_l)} \check{s} (t - \tau_l) \right\}. \quad (4.17)$$

Để đơn giản đặt

$$\alpha_l = a_l e^{j2\pi f_D \cos(\phi_l) \tau_l} \quad (4.18)$$

và

$$f_{D,l} = f_D \cos(\phi_l) \quad (4.19)$$

ta có thể viết lại  $\tilde{r}(t)$  ở dạng rút gọn sau

$$y(t) = \Re \left\{ \sum_{l=0}^{L-1} \alpha_l e^{j2\pi(f_c + f_{D,l})(t - \tau_l)} \check{s}(t - \tau_l) \right\} \quad (4.20)$$

trong đó  $\alpha_l$  và  $f_{D,l}$  tương ứng là hệ số suy hao và tần số Doppler của đường thứ  $l$  dưới tác động của sự chuyển động của MS. Từ công thức (4.20) chúng ta có thể thấy rằng vật tán xạ thứ  $l$  đã dịch tín hiệu phát đi  $\tau_l$  về thời gian và  $f_{D,l}$  về tần số.

Ta hoàn toàn có thể viết  $y(t)$  dưới dạng tín hiệu băng thông như trong biểu thức (4.10) trong đó đường bao phức của tín hiệu thu được có dạng

$$\check{y}(t) = \sum_{l=0}^{L-1} \alpha_l e^{-j\varphi_l(t)} \check{s}(t - \tau_l) \quad (4.21)$$

với

$$\varphi_l(t) = 2\pi[(f_c + f_{D,l})\tau_l - f_{D,l}t] \quad (4.22)$$

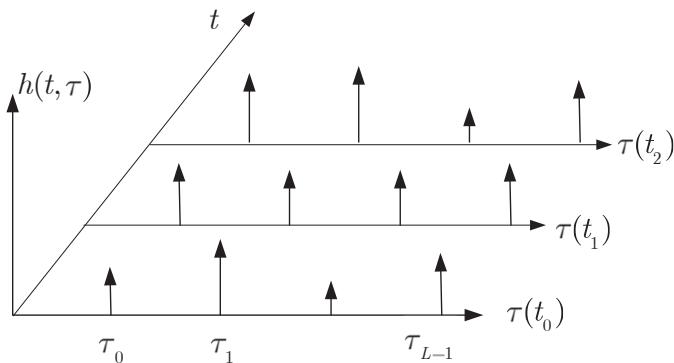
là góc pha của đường tín hiệu thứ  $l$ . Từ biểu thức (4.21), ta có thể mô hình hóa kênh truyền bằng một bộ lọc tuyến tính biến đổi theo thời gian (linear time-variant filter) với đáp ứng xung được cho bởi:

$$h(t, \tau) = \sum_{l=0}^{L-1} \alpha_l e^{-j\varphi_l(t)} \delta(\tau - \tau_l) \quad (4.23)$$

trong đó  $\delta(\cdot)$  là hàm xung delta. Một ví dụ về đáp ứng xung của kênh đa đường được mô tả ở Hình 4.7.

#### 4.2.3 Hậu quả của truyền sóng pha-định đa đường

Hậu quả của truyền sóng pha-định đa đường là làm cho tín hiệu thu [2]



**Hình 4.7:** Một ví dụ về đáp ứng xung của kênh đa đường.

- Hiện tượng pha-định hay tính chất *chọn lọc theo thời gian* (time selective fading) của kênh truyền do trải Doppler (Doppler spread) gây ra bởi chuyển động của MS.
- Hiện tượng trải trễ (delay spread) hay tính chất *chọn lọc theo tần số* do môi trường tán xạ.

**Hiện tượng trải Doppler.** Khi MS chuyển động qua các vùng ngẫu nhiên, tín hiệu thu được tại MS thay đổi cả về cường độ và pha tùy thuộc vào vận tốc chuyển động của MS. Xét trường hợp sự sai khác giữa các thời gian trễ của các đường tín hiệu  $\tau_m - \tau_n$  đủ nhỏ so với độ rộng của một ký hiệu được điều chế (modulated symbol duration). Khi đó các  $\tau_l$  trong (4.23) là xấp xỉ bằng nhau và bằng  $\hat{\tau}$ . Trong trường hợp này, đáp ứng xung  $h(t, \tau)$  có dạng

$$h(t, \tau) = h(t)\delta(\tau - \hat{\tau}) \quad (4.24)$$

và tín hiệu thu được bằng

$$\check{y}(t) = \int_{-\infty}^{+\infty} h(t, \tau) \check{s}(t - \tau) d\tau \quad (4.25)$$

$$= \int_{-\infty}^{+\infty} h(t)\delta(\tau - \hat{\tau}) \check{s}(t - \tau) d\tau \quad (4.26)$$

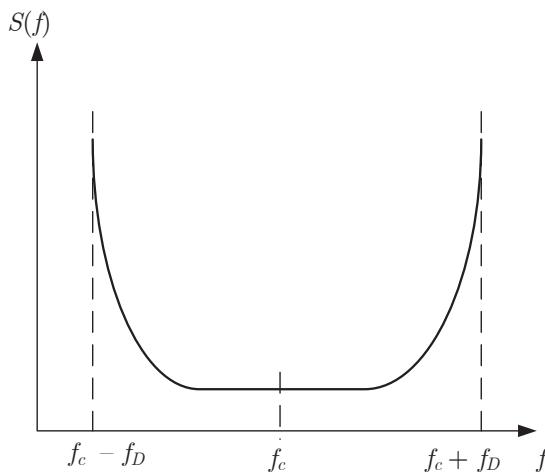
$$= h(t)\check{s}(t - \hat{\tau}). \quad (4.27)$$

Giả sử băng tần của tín hiệu là đủ nhỏ (tốc độ truyền tin đủ nhỏ) sao cho thời gian trễ  $\hat{\tau}$  không làm ảnh hưởng tới tín hiệu, khi đó

$\check{s}(t - \hat{\tau}) \approx s(t)$ , và tín hiệu thu được trở thành

$$\check{y}(t) = h(t)\check{s}(t) \quad (4.28)$$

với độ tăng ích kênh phức (complex channel gain)  $h(t) = \sum_{l=0}^{L-1} \alpha_l e^{-j\varphi_l(t)}$ . Nhớ rằng do tần số sóng mang  $f_c$  có giá trị rất lớn nên dù sự khác biệt giữa các thời gian trễ là nhỏ thì giữa các góc pha  $\varphi_l(t)$  vẫn có sự khác biệt lớn. Kết quả là làm cho độ tăng ích đường truyền  $h(t)$  biến đổi theo thời gian. Điều này có nghĩa là tín hiệu thu được vẫn trải qua hiện tượng pha-định. Hơn nữa, do  $h(t)$  thay đổi nhanh hơn  $\check{s}(t)$  nên tín hiệu thu bị trải (spread) trên thang tần số. Hiện tượng này còn được gọi là Doppler spread. Cụ thể là, một âm đơn  $s(t) = A$  sẽ bị trải thành vài thành phần ở băng tần  $[-f_D, f_D]$ , mỗi thành phần tương ứng với một vật tán xạ. Hình 4.8 mô tả phổ tần Doppler có dạng chữ U do hiện tượng trải Doppler gây nên.



Hình 4.8: Mô tả phổ tần Doppler có dạng hình chữ U.

Thực hiện biến đổi Fourier đối với đáp ứng xung  $h(t, \tau)$ , ta thu được hàm truyền đạt

$$\begin{aligned} H(t, f) &= \int_{-\infty}^{+\infty} h(t, \tau) e^{-j2\pi f\tau} d\tau = \int_{-\infty}^{+\infty} h(t) \delta(\tau - \hat{\tau}) e^{-j2\pi f\tau} d\tau \\ &= h(t) e^{-j2\pi f\hat{\tau}}. \end{aligned} \quad (4.29)$$

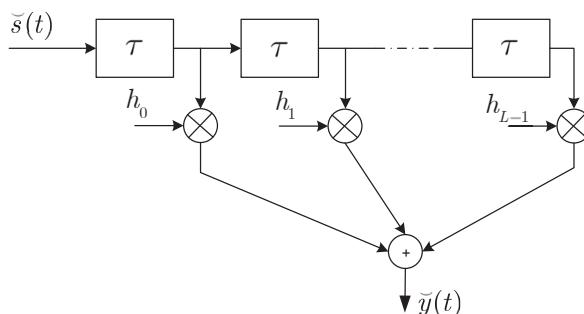
Do biên độ của hàm truyền đạt bằng  $|H(t, f)| = h(t)$  với  $f$  bất kỳ, tất cả các thành phần tần số trong tín hiệu thu được đều có cùng một độ tăng ích là  $h(t)$ . Trong trường hợp này ta nói rằng tín hiệu thu trải qua hiện tượng pha-định phẳng (flat fading). Và kênh truyền như thế này được gọi là kênh pha định-phẳng.

**Hiện tượng trải trễ.** Xét trường hợp các tần số Doppler  $f_{D,l}$  rất nhỏ tương ứng với MS đứng im. Khi đó các góc pha  $\varphi_l(t)$  trong (4.22) xấp xỉ bằng  $\varphi_l(t) \approx 2\pi f_c \tau_l = \psi_l$  và là một hằng số. Như vậy, công thức (4.23) có thể được viết lại như sau

$$h(t, \tau) = \sum_{l=0}^{L-1} h_l \delta(\tau - \tau_l) = h_s(\tau) \quad (4.30)$$

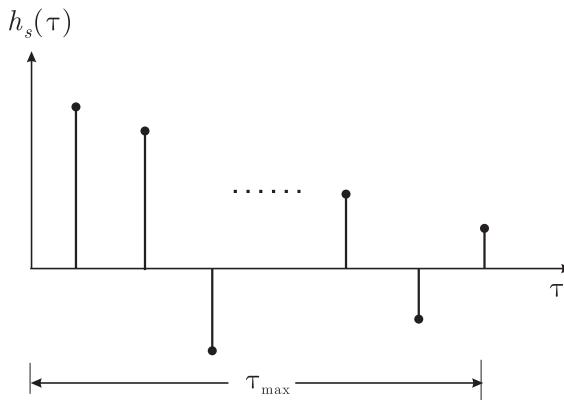
với  $h_l = \alpha_l e^{-j\psi_l}$  không phụ thuộc thời gian.

Từ biểu thức (4.30), ta thấy rằng với giả thiết  $f_{D,l}$  rất nhỏ kênh truyền trở thành một bộ lọc tuyến tính bất biến (linear time-invariant filter) với đáp ứng xung có chiều dài hữu hạn (FIR: Finite Impulse Response)  $h_s(\tau)$ . Hình 4.9 mô tả một ví dụ về bộ lọc FIR.



**Hình 4.9:** Kênh truyền đa đường được mô hình hóa như một bộ lọc FIR có các hệ số không đổi theo thời gian;  $\tau_l = (l + 1)\tau$ .

Hình vẽ 4.10 minh họa đáp ứng xung của một bộ lọc FIR tuyến tính bất biến. Giải  $\tau_D$  được gọi là trải trễ (delay spread). Chúng ta có thể thấy do độ dài các đường truyền sóng khác nhau nên đã làm cho tín hiệu bị dịch chuyển trễ.



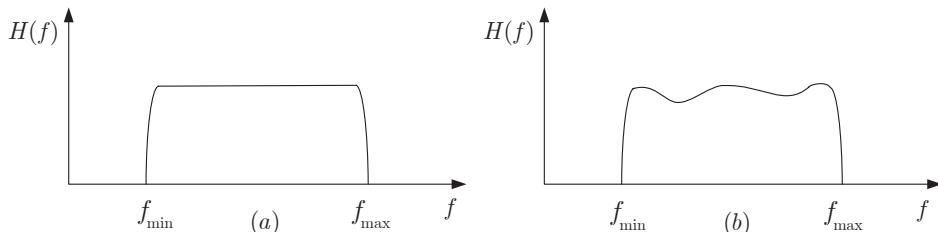
**Hình 4.10:** Đáp ứng xung của một bộ lọc FIR.

Thực hiện biến đổi Fourier lên đáp ứng xung ở (4.30) chúng ta có đáp ứng tần số

$$H(f) = \int_{-\infty}^{+\infty} h_s(\tau) e^{-j2\pi f \tau} d\tau \quad (4.31)$$

$$= \sum_{l=0}^{L-1} h_l e^{-j2\pi f \tau_l}. \quad (4.32)$$

Từ đây chúng ta thấy rằng tại các tần số khác nhau thì biên độ  $|H(f)|$  có giá trị khác nhau. Kênh truyền dẫn trong trường hợp này được gọi là kênh pha-định chọn lọc theo tần số (frequency selective fading). Ví dụ về đáp ứng tần số của một kênh pha-định phẳng và chọn lọc theo tần số được minh họa trên Hình 4.11.



**Hình 4.11:** Đáp ứng tần số của kênh truyền pha-định: (a) Kênh pha-định phẳng, (b) Kênh pha-định chọn lọc theo tần số.

### 4.3 Kênh pha-định Rayleigh

Từ các công thức (4.23) và (4.24), chúng ta viết lại được độ tăng ích kênh trong trường hợp MS chuyển động như sau

$$h(t) = \sum_{l=0}^{L-1} \alpha_l e^{-j\varphi_l(t)}. \quad (4.33)$$

Trong trường hợp không tồn tại tia trực tiếp LOS giữa BS và MS và tín hiệu thu được là tổng hợp của một số lượng lớn các đường tín hiệu, theo luật số lớn ta có thể coi  $h(t) = h_I(t) + jh_Q(t)$ , trong đó  $h_I(t)$  và  $h_Q(t)$  là các số thực với mọi  $t$ , là một quá trình ngẫu nhiên Gauss phức dừng theo nghĩa rộng(wide-sense stationary complex Gaussian random process) [3]. Trong một số môi trường tán xạ,  $h_I(t)$  và  $h_Q(t)$  là các biến ngẫu nhiên Gauss độc lập với nhau, có giá trị trung bình bằng không và cùng phương sai  $\sigma_g^2$  tại bất kỳ thời điểm  $t$  nào, tức là

$$\mathbb{E}\{h_I(t)\} = \mathbb{E}\{h_Q(t)\} = 0 \quad (4.34)$$

và

$$\mathbb{E}\{h_I(t)^2\} = \mathbb{E}\{h_Q(t)^2\} = \sigma_h^2. \quad (4.35)$$

Sử dụng công thức về phân bố Gauss ở (4.1) chúng ta có

$$p(h_I) = \frac{1}{\sigma_h \sqrt{2\pi}} \exp\left(-\frac{h_I^2}{2\sigma_h^2}\right) \quad (4.36)$$

$$p(h_Q) = \frac{1}{\sigma_h \sqrt{2\pi}} \exp\left(-\frac{h_Q^2}{2\sigma_h^2}\right). \quad (4.37)$$

Do  $h_I(t)$  và  $h_Q(t)$  là các quá trình độc lập nên chúng ta có phân bố

$$p(h) = p(h_I)p(h_Q) = \frac{1}{\sigma_h^2 2\pi} \exp\left(-\frac{|h|^2}{2\sigma_h^2}\right) \quad (4.38)$$

với

$$|h(t)|^2 = h_I^2(t) + h_Q^2(t). \quad (4.39)$$

Chuyển  $h(t)$  sang hệ toạ độ cực  $h(t) = \alpha(t)e^{j\theta(t)}$ , với  $\alpha(t) = |h(t)|$  là biên độ của  $h(t)$ , chúng ta có pdf kết hợp [2]

$$p(\alpha, \theta) = \frac{\alpha}{\sigma_h^2 2\pi} \exp\left(-\frac{\alpha^2}{2\sigma_h^2}\right). \quad (4.40)$$

Do  $\alpha$  và  $\theta$  là các biến độc lập nên chúng ta có thể viết

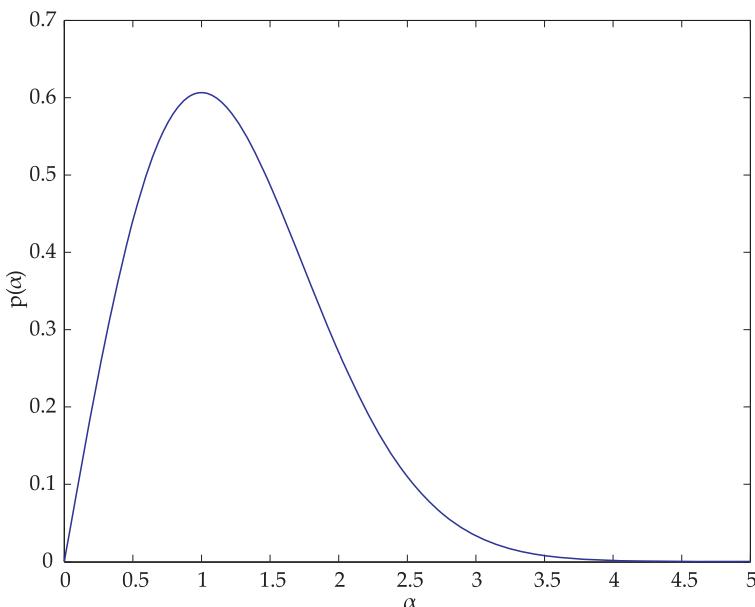
$$p(\alpha, \theta) = p(\alpha) \cdot p(\theta) = \frac{\alpha}{\sigma_h^2} \exp\left(-\frac{\alpha^2}{2\sigma_h^2}\right) \cdot \frac{1}{2\pi} \quad (4.41)$$

hay

$$p(\alpha) = \frac{\alpha}{\sigma_h^2} \exp\left(-\frac{\alpha^2}{2\sigma_h^2}\right), \quad \alpha \geq 0 \quad (4.42)$$

$$p(\theta) = \frac{1}{2\pi}, \quad -\pi \leq \theta < \pi. \quad (4.43)$$

Tức là, PDF của biên độ  $\alpha(t)$  là phân bố Rayleigh, và pha-định kiểu này được gọi là *pha-định Rayleigh*. Hình vẽ 4.12 mô tả phân bố Rayleigh với  $\sigma = 1$ .



**Hình 4.12:** Hàm phân bố Rayleigh với  $\sigma_h^2 = 1$ .

## 4.4 Kênh pha-định Rice

Trong trường hợp môi trường tán xạ tồn tại tia truyền thẳng,  $h_I(t)$  và  $h_Q(t)$  là các biến ngẫu nhiên Gauss độc lập với nhau, có giá trị trung bình bằng  $\mu_I(t)$  và  $\mu_Q(t)$  khác 0. Nếu ta vẫn giả thiết  $h_I(t)$  và  $h_Q(t)$  có cùng phương sai  $\sigma_h^2$  tại bất kỳ thời điểm  $t$  nào, thì biến độ của  $h(t)$ , tức là  $\alpha(t)$ , tại một thời điểm  $t$  bất kỳ sẽ có phân bố Rice được cho bởi [3]

$$p(\alpha) = \frac{\alpha}{\sigma_h^2} \exp\left(-\frac{\alpha^2 + \chi^2}{2\sigma_h^2}\right) I_0\left(\frac{\alpha\chi}{\sigma_h^2}\right), \quad \alpha \geq 0 \quad (4.44)$$

trong đó

$$\chi = \mu_I^2(t) + \mu_Q^2(t) \quad (4.45)$$

được gọi là tham số lệch tâm (non-centrality parameter), và  $I_0(x)$  là hàm số Bessel sửa đổi bậc 0 loại 1 (zero order modified Bessel function of the first kind). Dạng pha định này được gọi là pha định Rice.

Một số mô hình kênh Rice đã được đề xuất trong thực tế giả thiết rằng  $\mu_I(t)$  và  $\mu_Q(t)$  là các hằng số khác không [4, 5]. Một phương pháp có nhiều ưu điểm hơn đã được đề xuất bởi Aulin [6]. Trong phương pháp này các giá trị trung bình  $\mu_I(t)$  và  $\mu_Q(t)$  tương ứng với các thành phần đồng pha (in phase) và trực giao (quadrature) của tia LOS được mô hình là các tham số xác định biến đổi theo thời gian (deterministic time-varying) như sau [3, 6]

$$\mu_I(t) = \chi \cos(2\pi f_D \cos(\phi_0)t + \theta_0) \quad (4.46)$$

$$\mu_Q(t) = \chi \sin(2\pi f_D \cos(\phi_0)t + \theta_0) \quad (4.47)$$

trong đó  $f_D \cos(\phi_0)$  và  $\theta_0$  là tần số Doppler và góc lệch pha ngẫu nhiên ứng với tia LOS.

Một thông số quan trọng của kênh pha-định Rice là *hệ số Rice* (Rice factor),  $K$ , được định nghĩa là tỉ số giữa công suất của tia LOS  $\chi^2$  và công suất của các thành phần tán xạ  $2\sigma_h^2$ , tức là  $K = \frac{\chi^2}{2\sigma_h^2}$ . Ta

thấy rằng, khi  $K = 0$  thì kênh truyền thuận tuý là kênh Rayleigh, và khi  $K = \infty$  kênh truyền sẽ không còn hiện tượng pha-định.

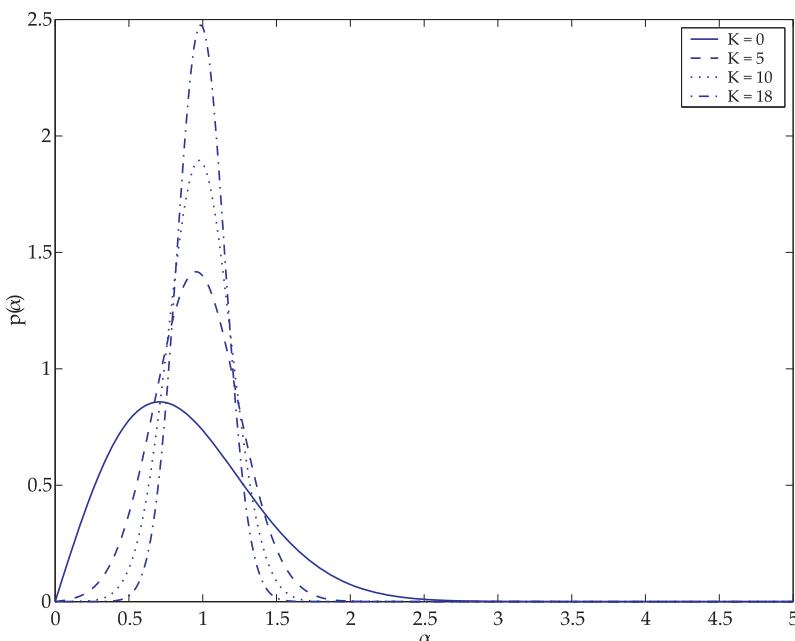
Sử dụng hệ số Rice, ta có thể viết

$$\chi^2 = \frac{KA_p}{K+1}, \quad 2\sigma_h^2 = \frac{A_p}{K+1}, \quad (4.48)$$

trong đó  $A_p = E\{\alpha^2\} = \chi^2 + 2\sigma_h^2$  là công suất trung bình của  $g(t)$ . Khi đó, hàm phân bố Rice trong (4.44) được viết lại như sau

$$p(\alpha) = \frac{2\alpha(K+1)}{A_p} \exp\left(-K - \frac{(K+1)\alpha^2}{A_p}\right) I_0\left(2\alpha\sqrt{\frac{K(K+1)}{A_p}}\right), \quad \alpha \geq 0 \quad (4.49)$$

Hình vẽ 4.13 minh họa phân bố Rice cho một số giá trị khác nhau của  $K$ . Trường hợp  $K = 0$  chính là hàm phân bố Rayleigh. Từ hình vẽ ta thấy rằng khi  $K$  càng lớn thì bề rộng của  $p(\alpha)$  càng có xu hướng co hẹp lại quanh giá trị  $\sqrt{A_p} = 1$ . Điều đó chứng tỏ rằng khi  $K$  càng lớn thì tính ngẫu nhiên của  $\alpha$  càng giảm.



**Hình 4.13:** Hàm phân bố Rice cho các giá trị khác nhau của  $K$  với  $A_p = 1$ .

## 4.5 Mô phỏng pha-định Rayleigh và pha-định Rice

Các chương trình mô phỏng kênh thông tin vô tuyến được sử dụng rất nhiều trong các phòng nghiên cứu vì chúng cho phép ta khảo sát và đánh giá hệ thống một cách thuận tiện với chi phí rất thấp so với việc đo thử trong các môi trường thực tế. Một số phương pháp xây dựng các chương trình mô phỏng pha định đã được đề xuất như phương pháp tổng của các tín hiệu hình sin (SOS: sum of sinusoids method) [7, 8, 9], phương pháp lọc nhiễu Gauss (Gaussian noise filtering method) [3], phương pháp biến đổi Fourier rời rạc (IDFT method) [10], và một số phương pháp khác. Trong cuốn sách này chúng tôi trình bày phương pháp được sử dụng rộng rãi trong mô phỏng pha-định Rayleigh và pha-định Rice, đó là phương pháp tổng của các tín hiệu sin.

### 4.5.1 Mô phỏng pha-định Rayleigh

Để đơn giản ta giả thiết tín hiệu phát đi là  $s(t) = 1$  và số lượng tia tới MS là  $L$ , khi đó từ công thức (4.21) ta có thể viết đường bao phức của tín hiệu như sau

$$\check{y}(t) = \sum_{l=0}^{L-1} \alpha_l e^{-j\varphi_l(t)} \quad (4.50)$$

$$= \sum_{l=0}^{L-1} a_l e^{j2\pi f_{D,l}\tau_l} e^{-j\varphi_l(t)} \quad (4.51)$$

Sử dụng  $\varphi_l(t)$  ở công thức (4.22), sau một số biến đổi toán học, ta có

$$\check{y}(t) = \sum_{l=0}^{L-1} a_l e^{j[2\pi f_D \cos(\phi_l)t + \theta_l]} \quad (4.52)$$

với  $\theta_l = -2\pi f_c \tau_l$  là góc pha ban đầu của tia tới thứ  $l$ . Các góc pha  $\theta_l$  được giả thiết là độc lập thống kê với nhau và có phân bố đều trong khoảng  $[0, 2\pi]$ . Để thuận tiện, ta thực hiện chuẩn hoá  $y(t)$  sao cho công suất của nó bằng 1, tức là ta chọn các hệ số  $a_l = \frac{1}{\sqrt{L}}$ . Lúc này ta

có

$$\check{y}(t) = \frac{1}{\sqrt{L}} \sum_{l=0}^{L-1} e^{j[2\pi f_D \cos(\phi_l)t + \theta_l]}. \quad (4.53)$$

Theo mô hình của Jakes thì tia thứ  $l$  đến đều từ tất cả các hướng nên nếu chọn  $\psi_0$  là một giá trị ngẫu nhiên trong khoảng  $[0, 2\pi)$  thì

$$\phi_l = \psi_0 + 2\pi \frac{l}{L}, \text{ với } l = 0, 1, 2, \dots, L-1. \quad (4.54)$$

Thuật toán mô phỏng được tổng kết dưới đây.

### Thuật toán mô phỏng pha-định Rayleigh

Thuật toán này dựa trên phương pháp của Jakes được mô tả như sau. Để thuận tiện trong việc tích hợp thuật toán với Matlab thì biến chạy  $l = 0, 1, 2, \dots, L-1$  được đổi thành  $l = 1, 2, 3, \dots, L$ .

- (1) Chọn  $L$  lẻ
- (2) Tạo  $\psi_0$  ngẫu nhiên trong khoảng  $[0, 2\pi)$
- (3) For  $l = 1$  to  $L$
- (4) Tạo  $\theta_l$  ngẫu nhiên trong khoảng  $[0, 2\pi)$
- (5) Tính  $\phi_l = \psi_0 + 2\pi \frac{l-1}{L}$
- (6) Tính  $y_l(t) = e^{j[\theta_l + 2\pi f_D \cos(\phi_l)t]}$
- (7) End

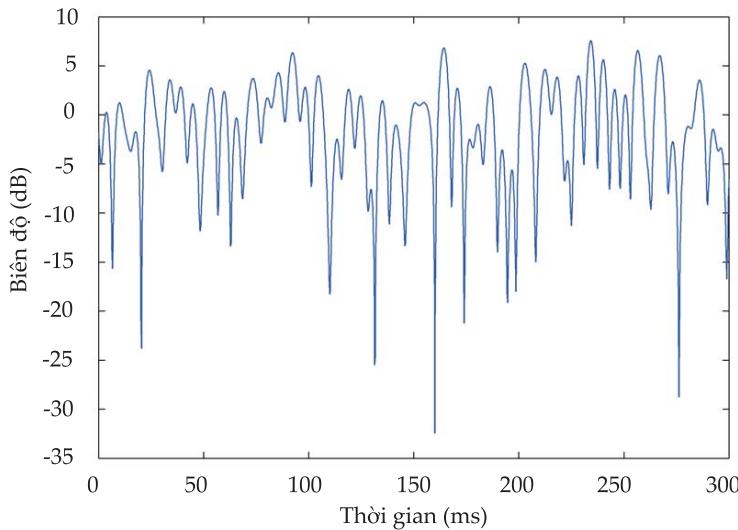
$$(8) \text{ Tính } \check{y}(t) = \frac{1}{\sqrt{L}} \sum_{l=1}^L y_l(t)$$

Thuật toán này tuy dễ hiểu và dễ thực hiện nhưng số lượng phép tính sẽ tăng lên đáng kể nếu ta chọn  $L$  có giá trị lớn. Trong [7] Jakes đã trình bày một phương pháp nhằm làm giảm số lượng các thành phần dao động tần số thấp trong (4.53), nhờ đó giảm được số lượng phép tính. Jakes đã chọn số lượng tia  $L$  thỏa mãn điều kiện

$$L = 4M + 2. \quad (4.55)$$

Khi đó số lượng thành phần tần số thấp trong (4.53) sẽ giảm từ  $L$  xuống còn  $M + 1$ . Trong [8], Pop và các cộng sự cũng đã đề xuất phương pháp nhằm khắc phục những nhược điểm trong phương pháp giảm số lượng thành phần dao động tần thấp của Jakes. Bạn đọc có thể tham khảo [7] và [8] để hiểu rõ hơn về các phương pháp này.

Hình vẽ 4.14 minh họa một đường bao bị pha-định Rayleigh điển hình được tạo ra bởi chương trình mô phỏng pha-định Rayleigh trình bày ở trên. Trong chương trình mô phỏng, số lượng tia đèn được chọn là  $L = 35$  và tần số Doppler cực đại  $f_D = 100$  Hz.



**Hình 4.14:** Đường bao tín hiệu bị pha-định Rayleigh, số lượng tia  $L = 35$ , tần số Doppler cực đại  $f_D = 100$  Hz.

#### 4.5.2 Mô phỏng pha-định Rice

Từ thuật toán mô phỏng pha-định Rayleigh, ta dễ dàng xây dựng được thuật toán mô phỏng pha-định Rice dựa trên kết quả phân tích trình bày ở trên. Một cách cụ thể hơn, pha-định Rice sẽ bao gồm 2 thành phần: một thành phần sinh ra do các tia tán xạ và một thành phần sinh ra do tia LOS. Để mô phỏng hai thành phần này,

ta coi thành phần thứ nhất có phân bố Rayleigh và có công suất  $2\sigma_h^2 = \frac{A_p}{K+1}$  (xem công thức (4.48)). Thành phần thứ hai là thành phần có tham số xác định theo thời gian được cho bởi các biểu thức (4.46) và (4.47). Thành phần này có công suất  $\chi^2 = \frac{A_p K}{K+1}$ . Rõ ràng tổng công suất của hai thành phần này là  $\chi^2 + 2\sigma_h^2 = A_p$ .

Trong chương trình mô phỏng này, ta chọn  $A_p = 1$ . Thành phần tán xạ thứ nhất được chọn là  $r(t)$  trong bởi biểu thức (4.53). Tuy nhiên, để thành phần này có công suất là  $2\sigma_h^2 = \frac{1}{K+1}$ , ta cần phải chuẩn hoá nó bằng cách chia  $r(t)$  cho  $\sqrt{\frac{1}{K+1}}$ . Thành phần LOS được cho bởi biểu thức

$$\tilde{y}(t)_{\text{LOS}} = \mu_I + j\mu_Q \quad (4.56)$$

Kết quả là đường bao tín hiệu tổng hợp bị pha-định Rice được cho bởi [9]

$$\tilde{y}(t)_{\text{RICE}} = [\tilde{y}(t) + \sqrt{K}e^{j[2\pi f_D \cos(\phi_0)t + \theta_0]}]/\sqrt{K+1} \quad (4.57)$$

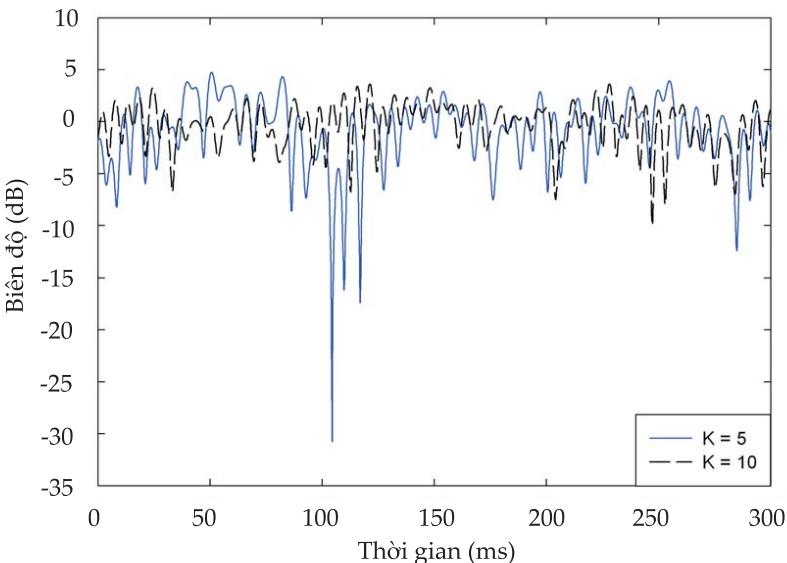
Thuật toán mô phỏng được mô tả như sau.

### Thuật toán mô phỏng pha-định Rice

- (1) Chọn  $L$  lẻ
- (2) Gán cho  $\phi_0$  một giá trị bất kỳ trong khoảng  $[0, 2\pi)$
- (3) Tạo  $\theta_0$  ngẫu nhiên trong khoảng  $[0, 2\pi)$
- (4) Gán cho hệ số Rice  $K$  giá trị mong muốn
- (5) Tạo  $\psi_0$  ngẫu nhiên trong khoảng  $[0, 2\pi)$
- (6) For  $l = 1$  to  $L$
- (7) Tạo  $\theta_l$  ngẫu nhiên trong khoảng  $[0, 2\pi)$
- (8) Tính  $\phi_l = \psi_0 + 2\pi \frac{l-1}{L}$
- (9) Tính  $\tilde{y}_l(t) = e^{j[\theta_l + 2\pi f_D \cos(\phi_l)t]}$
- (10) End

$$(11) \text{ Tính } \check{y}(t) = \frac{1}{\sqrt{L}} \sum_{l=1}^L \check{y}_l(t)$$

$$(12) \text{ Tính } \check{y}(t)_{\text{RICE}} = [\check{y}(t) + \sqrt{K} e^{j[2\pi f_D \cos(\phi_0)t + \theta_0]}] / \sqrt{K + 1}$$



**Hình 4.15:** Đường bao tín hiệu bị pha-định Rice, số lượng tia  $L = 35$ , tần số Doppler cực đại  $f_D = 100$  Hz, hệ số Rice  $K = 5$  và  $K = 10$ .

Hình vẽ 4.15 minh họa các đường bao bị pha-định Rice được tạo ra bởi chương trình Matlab FadingSim.m mô phỏng pha-định Rice viết theo thuật toán trình bày ở trên (xem Matlab Program 4.2). Trong chương trình mô phỏng, số lượng tia đến được chọn để mô phỏng thành phần tán xạ là  $L = 35$ , tần số Doppler cực đại  $f_D = 100$  Hz, hệ số Rice  $K = 5$  và  $K = 10$ . Từ các hình vẽ 4.14 và 4.15 ta thấy rõ ràng là khi hệ số Rice tăng lên, tín hiệu càng ít chịu ảnh hưởng của hiện tượng pha-định.

Để đánh giá tính chính xác của các chương trình mô phỏng trên. Chúng ta cần phải đánh giá những đặc tính thống kê của các mô hình trên, đặc biệt là các đại lượng thống kê bậc nhất (tức là giá trị trung bình) và đại lượng thống kê bậc hai (tức là các hàm tự

tương quan và tương quan chéo). Tuy nhiên trong phạm vi quyển sách này chúng tôi sẽ không đi sâu vào đánh giá các đặc tính thông kê này. Bạn đọc có thể tham khảo những nội dung này trong [3, 8, 9].

## 4.6 Tóm tắt

Trong chương này chúng ta đã xem xét một số mô hình kênh thông tin cơ bản được sử dụng trong mô phỏng đánh giá chất lượng của các hệ thống thông tin vô tuyến. Thông qua các kiến thức của chương này, chúng ta biết được các kiến thức về kênh AWGN, kênh pha-định Rayleigh, kênh pha-định Rice cũng như các thuật toán mô phỏng các loại kênh truyền này. Những kiến thức này sẽ giúp ích cho việc thực hiện mô phỏng các dạng kênh truyền khác nhau trong thông tin vô tuyến.

## Tài liệu tham khảo

- [1] B. Sklar, *Digital Communications Fundamentals and Applications*. Prentice Hall, 1988.
- [2] J. K. Cavers, *Mobile Channel Characteristics*. Kluwer Academic Publishers, 2000.
- [3] G. L. Stüber, *Principles of Mobile Communication*. Kluwer Academic Publishers, 2001.
- [4] S. O. Rice, "Statistical properties of a sine wave plus random noise," *Bell Syst. Tech. J.*, vol. 27, pp. 109–157, Jan. 1948.
- [5] K. W. Yip, "Discrete time model for digital communications over a frequency selective Rician fading WSSUS channel," *IEE Proc. Commun.*, vol. 15, pp. 34–42, Feb. 1996.
- [6] T. Aulin, "A modified model for the fading signal at a mobile radio channel," *IEEE Trans. Veh. Technol.*, vol. 28, pp. 182–203, 1979.
- [7] W. C. Jakes, *Microwave Mobile Communications*. Piscataway, NJ: IEEE Press, 1994.
- [8] M. F. Pop and N. C. Beaulieu, "Limitations of sum of sinusoids fading channel simulators," *IEEE Trans. Commun.*, vol. 49, pp. 699–708, Apr. 2001.

- [9] C. Xiao and Y. R. Zheng, "A statistic simulation model for mobile radio fading channels," *Proc. WCNC*, vol. 1, pp. 144–149, Mar. 2003.
- [10] D. J. Young and N. C. Beaulieu, "The generation of correlated rayleigh random variates by inverse Fourier transform," *IEEE. Trans. Commun.*, vol. 48, pp. 1114–1127, Jul. 2000.

## Phụ lục: Chương trình Matlab mẫu

---

### ■ MATLAB Program 4.2 FadingSim.m

```
%Program to simulate Rayleigh and Rice fading based on SOS method
% (C)2011 Le Minh Tuan, UVISION JSC.
% Date: 16/07/2011

clear all;
clc;

%% Input parameters
L=35; %Number of sinusoids
k=[0:1500]; %Discrete time, i.e, t = kT
fD=100; %Dopper frequency
T=0.02/fD; %Sampling period
t=k*T*1000; %Time in ms
K=input('Enter Rice factor K: ');

%%---Generate scattering part-----
%Randomly generate Theta for L paths
Theta = rand(1,L)*2*pi;
%Randomly generate Phi for L paths
Phi = [0:L-1]*2*pi/L+(rand(1,L))*2*pi/L;
gt=0;
for l=1:L
    gt=gt+exp(j*(2*pi*fD*T*cos(Phi(l))*k+Theta(l)));
end
%Normalization
```

```
gt=gt/sqrt(L);

%%--Generate LOS part and composit signal----
%Initial phase of LOS component
Theta0 = rand(1,1)*2*pi;
%Angle of arrival of LOS component
Phi0 = pi/6;
%Rice fading
gt=(gt+sqrt(K)*exp(j*(2*pi*fD*T*cos(Phi0)*k+Theta0)))/sqrt(K+1);

%%Plot fading
alpha=20*log10(abs(gt));
plot(t,alpha);
xlabel('Time (ms)')
ylabel('Magnitude (dB)')
if K==0
    txt='Rayleigh Fading';
else
    txt=['Rice fading with Rice factor K = ', num2str(K)];
end
title(txt)
%!
```

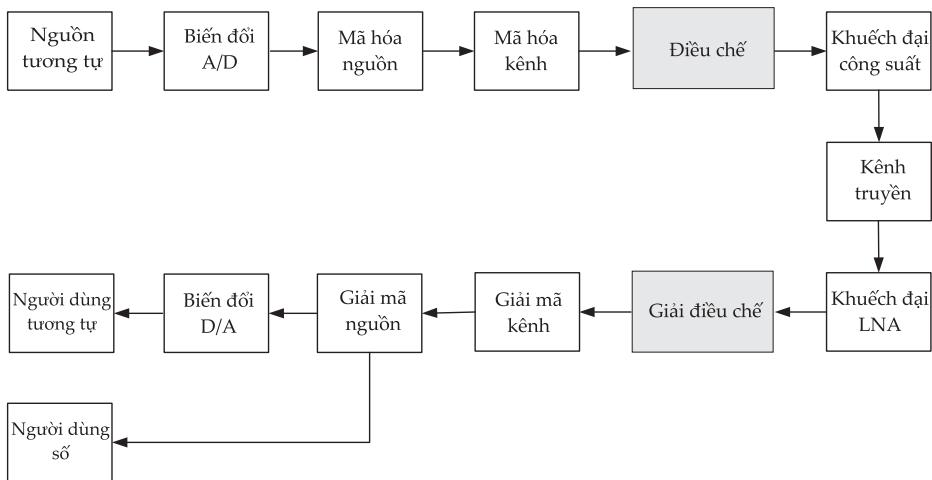
---

## Chương 5

# Các kỹ thuật điều chế số

### 5.1 Các hệ thống thông tin số

Một hệ thống thông tin số điển hình có sơ đồ khối như minh họa trên Hình 5.1 [1].



Hình 5.1: Sơ đồ khối của một hệ thống thông tin số điển hình.

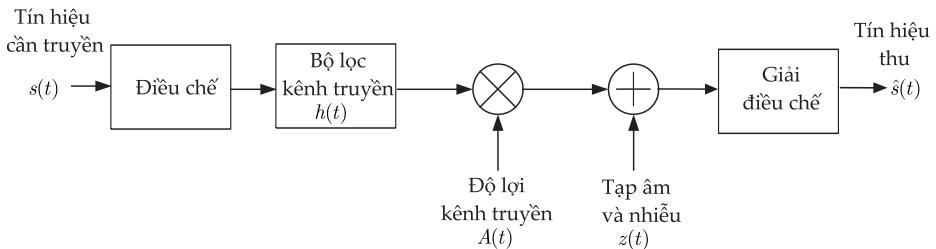
Thông tin cần truyền đi có thể từ một nguồn tương tự (chẳng hạn như tín hiệu thoại) hoặc từ một nguồn số (dữ liệu máy tính). Nếu nguồn tín hiệu là tương tự thì bộ chuyển đổi tương tự số (A/D converter) sẽ chuyển đổi các tín hiệu tương tự thành tín hiệu số (các

bit 0 và 1). Bộ mã hóa nguồn (source encoder) sẽ nhận các tín hiệu số từ bộ chuyển đổi A/D và mã hóa thành các tín hiệu số ngắn hơn (tức là, loại bỏ bớt thông tin dư thừa). Nhờ đó, bộ mã hóa nguồn sẽ giúp làm giảm (tiết kiệm) băng thông cần thiết của hệ thống. Bộ mã hóa kênh (channel encoder) nhận các tín hiệu số từ đầu ra của bộ mã hóa nguồn và mã hóa chúng thành các tín hiệu số dài hơn. Thông tin dư thừa được thêm vào tín hiệu số cần mã hóa một cách có chủ định nhằm giúp máy thu sửa được các lỗi có thể xảy ra trong quá trình truyền tin. Trong hầu hết các trường hợp, thông tin được truyền đi ở dưới dạng một tín hiệu thông dài tần số cao. Khi đó, bộ điều chế (modulator) sẽ điều chế các ký hiệu hay còn gọi là dấu (symbol) đã mã hóa vào một sóng mang. Trong một số trường hợp khác, thông tin được truyền đi ở băng gốc hay băng cơ sở (baseband), bộ điều chế lúc này là bộ điều chế băng gốc hay còn gọi là bộ định dạng (formator) tín hiệu. Nó định dạng các ký hiệu đã mã hóa thành một dạng sóng phù hợp cho truyền dẫn.

Thông thường, tại lối ra của bộ điều chế sẽ có một bộ khuếch đại công suất. Khi truyền dẫn ở tần số cao, quá trình điều chế và giải điều chế thường được thực hiện ở một tần số trung gian được gọi là trung tần (intermediate frequency hay IF). Trong trường hợp này, hệ thống phải có thêm một bộ đổi tần lên (up-convertisor) nằm giữa bộ điều chế và bộ khuếch đại công suất. Đối với các hệ thống vô tuyến, tầng cuối của máy phát là một ăng-ten dùng để bức xạ tín hiệu ra không gian.

Môi trường truyền dẫn thường được gọi là kênh truyền. Khi tín hiệu đi qua kênh truyền, nó bị tác động bởi nhiều cộng, pha định và suy hao. Nhiều ở đây bao gồm các nhân tố tác động ngẫu nhiên bên ngoài và nhân tố tác động nội tại bên trong hệ thống (còn gọi là tạp âm). Kênh truyền thường có một băng thông hữu hạn và có thể được xem như là một bộ lọc. Ở phía máy thu, toàn bộ quá trình xử lý tín hiệu được thực hiện ngược lại với phía máy phát. Trước hết, tín hiệu thu được với cường độ yếu sẽ được khuếch đại (rồi được

đổi tần xuống nếu cần) và giải điều chế. Sau đó, bộ giải mã kênh (channel decoder) loại bỏ những thông tin dư thừa trong luồng bit thu được và bộ giải mã nguồn (source decoder) khôi phục tín hiệu thu được trở về dạng ban đầu của nó trước khi chuyển tới người sử dụng.



**Hình 5.2:** Mô hình hệ thống thông tin số với điều chế và giải điều chế.

Với mục đích mô tả các kỹ thuật điều chế và giải điều chế đồng thời phân tích phẩm chất của chúng, người ta thường sử dụng mô hình hệ thống đơn giản hóa như minh họa trên Hình 5.2. Mô hình này loại bỏ đi những khối không liên quan tới điều chế. Tuy nhiên, các kỹ thuật điều chế và giải điều chế được phát triển gần đây kết hợp cả điều chế và mã hóa kênh. Trong những trường hợp như vậy, bộ mã hóa kênh là một phần của bộ điều chế và bộ giải mã kênh là một phần của bộ giải điều chế. Từ Hình 5.2, tín hiệu thu được tại lối vào bộ giải điều chế được cho bởi:

$$r(t) = A(t)[s(t) * h(t)] + z(t) \quad (5.1)$$

trong đó  $*$  là ký hiệu của tích chập. Trong Hình 5.2, kênh truyền được mô tả bao gồm ba thành phần. Thành phần thứ nhất là bộ lọc kênh truyền. Do tín hiệu  $s(t)$  từ bộ điều chế phải đi qua máy phát, kênh truyền (môi trường truyền dẫn) và máy thu trước khi tới bộ giải điều chế, nên bộ lọc kênh truyền là một bộ lọc tổ hợp với hàm truyền đạt là:

$$H(f) = H_T(f)H_C(f)H_R(f) \quad (5.2)$$

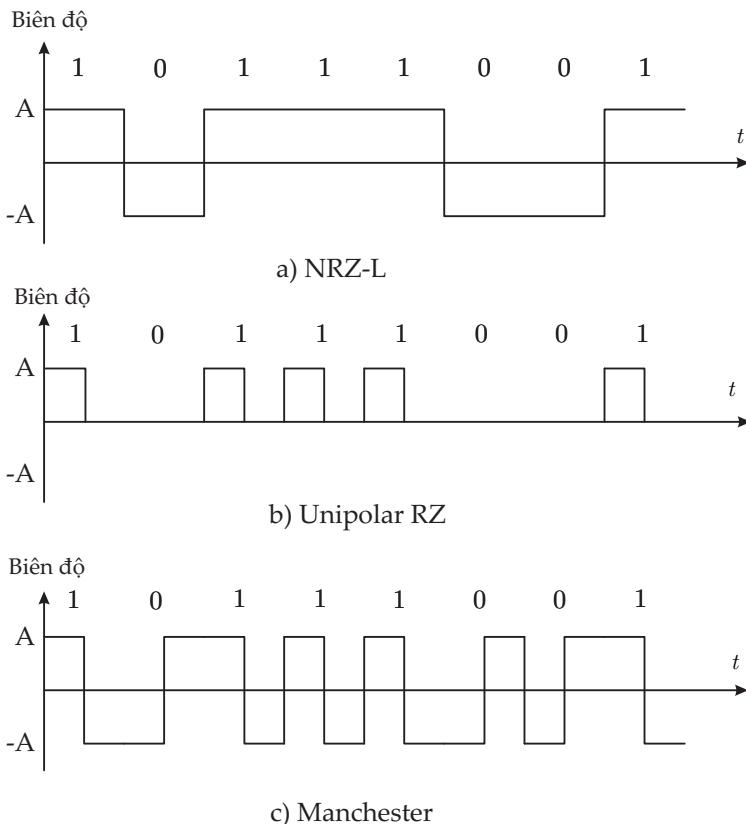
trong đó  $H_T(f)$ ,  $H_C(f)$ , và  $H_R(f)$  tương ứng là hàm truyền đạt của máy phát, kênh truyền và máy thu. Thành phần thứ hai hệ số  $A(t)$ .

Hệ số này thường là số phức. Nó biểu thị cho hiện tượng pha-định trong một số loại kênh truyền, ví dụ như kênh thông tin vô tuyến mà chúng ta đã khảo sát ở chương trước, lúc này  $A(t) \equiv g(t)$ . Thành phần thứ ba là nhiễu và tạp âm cộng  $n(t)$ . Trong thực tế, Cả ba thành phần này luôn tồn tại và gây tác động tới phẩm chất lỗi bit của hệ thống.

## 5.2 Các phương pháp điều chế cơ sở

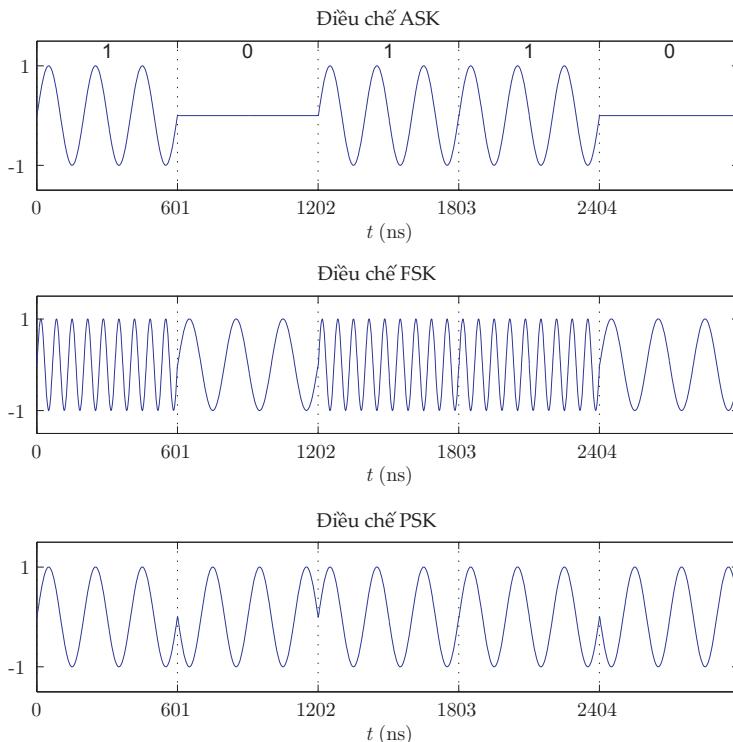
Điều chế số là một quá trình "ghi" một ký hiệu số (digital symbol) vào một tín hiệu phù hợp cho truyền dẫn. Đối với truyền dẫn khoảng cách ngắn, người ta thường sử dụng điều chế băng gốc. Trong điều chế băng gốc, chuỗi các ký hiệu số được sử dụng để tạo một dạng sóng xung vuông với những đặc tính xác định dùng để biểu thị mỗi một loại ký hiệu mà không tạo ra nhầm lẫn trong quá trình khôi phục tín hiệu ở phía thu. Các đặc tính này là các biến thể của biên độ xung, độ rộng xung và vị trí xung. Hình 5.3 minh họa một số dạng sóng điều chế băng gốc. Dạng sóng đầu tiên là điều chế NRZ-L (nonreturn-to-zero level), trong đó bit 1 được biểu diễn bởi một xung vuông dương còn bit 0 được biểu thị bằng một xung vuông âm. Cả xung dương và xung âm có độ rộng xung là  $T$ . Dạng sóng thứ hai là điều chế RZ đơn cực (unipolar return to zero) trong đó bit 1 được biểu thị bằng một xung dương có độ rộng xung  $T/2$  còn bit 0 không có tín hiệu. Dạng sóng thứ ba là điều chế Manchester, trong đó bit 1 biểu thị bằng một dạng sóng bao gồm một xung dương với độ rộng  $T/2$  và một xung âm với độ rộng  $T/2$ , còn bit 0 được biểu diễn ngược lại. Trong phạm vi quyển sách này, chúng ta sẽ không nghiên cứu nhiều về các dạng điều chế băng gốc.

Đối với truyền dẫn vô tuyến và khoảng cách dài, điều chế thông dải (bandpass modulation) thường được sử dụng. Điều chế thông dải còn được gọi là điều chế sóng mang. Trong phương thức điều chế này, người ta sử dụng một chuỗi các ký hiệu để làm thay đổi các



Hình 5.3: Các ví dụ về điều chế số bằng gốc.

thông số của một tín hiệu hình sin cao tần gọi là sóng mang. Chúng ta đều biết rằng, một tín hiệu hình sin có ba thông số là: biên độ, tần số và pha. Vì vậy, điều chế biên độ (điều biên), điều chế tần số (điều tần) và điều chế pha (điều pha) là ba phương pháp điều chế cơ sở trong điều chế sóng mang. Hình 5.4 minh họa ba phương pháp điều chế sóng mang cơ sở là khóa dịch biên độ (ASK), khóa dịch tần số (FSK) và khóa dịch pha (PSK). Trong phương pháp điều chế ASK, bộ điều chế tạo ra một burst sóng mang tương ứng với bit 1 và không phát đi tín hiệu nếu là bit 0. Vì vậy, phương pháp này còn được gọi là khóa đóng-mở (on-off keying hay OOK). Tuy nhiên, trong phương pháp ASK, biên độ tín hiệu ứng với bit 0 không nhất thiết phải bằng 0. Trong phương pháp FSK, tương ứng với bit 1 sẽ



**Hình 5.4:** Các phương pháp điều chế sóng mang cơ sở.

có một burst sóng mang tần số cao được truyền đi và tương ứng với bit 0 sẽ có một burst sóng mang ở tần số thấp hơn được truyền đi hoặc ngược lại.

Trong phương pháp PSK, bit 1 được truyền đi dưới dạng một cụm (burst) sóng mang với góc pha đầu bằng 0, trong khi đó bit 0 được truyền đi dưới dạng một burst sóng mang với góc pha đầu là  $180^\circ$ . Bằng cách kết hợp các phương pháp điều chế cơ sở này, người ta có thể tạo ra được một số phương pháp điều chế khác. Ví dụ, bằng cách kết hợp hai tín hiệu PSK nhị phân (hay BPSK) với các sóng mang trực giao với nhau, người ta tạo ra được phương pháp điều chế mới gọi là khóa dịch pha cầu phương (QPSK). Bằng cách điều chế cả biên độ và pha của sóng mang, người ta tạo ra được phương pháp điều chế mới gọi là điều chế biên độ cầu phương (QAM).

Trong các mục tiếp theo, chúng ta sẽ nghiên cứu kỹ hơn về

phương pháp điều chế PSK và QAM. Đồng thời, chúng ta sẽ xây dựng các chương trình mô phỏng Monte-Carlo để đánh giá phẩm chất lỗi bit (BER) của các hệ thống này trong các kênh truyền AWGN và pha-định Rayleigh.

### 5.3 Phương pháp điều chế PSK

Khóa dịch pha (PSK) là một tập hợp nhiều cấu hình điều chế số khác nhau. PSK được sử dụng rộng rãi trong lĩnh vực thông tin truyền thông. Trong mục này, chúng ta nghiên cứu một số cấu hình điều chế PSK tiêu biểu. Trong mỗi một cấu hình, chúng ta sẽ nghiên cứu cách mô tả tín hiệu, sơ đồ khói điều chế/giải điều chế và công thức tính toán lỗi bit trong các kênh truyền AWGN và pha-định Rayleigh. Chúng ta cũng sẽ xây dựng các chương trình mô phỏng để đánh giá phẩm chất lỗi bit của các hệ thống này dựa trên phương pháp mô phỏng Monte-Carlo.

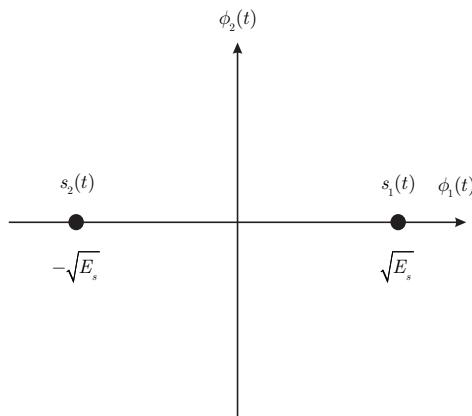
#### 5.3.1 Khóa dịch pha nhị phân BPSK

Trong phương pháp điều chế BPSK, dữ liệu nhị phân được biểu diễn bởi hai tín hiệu với các góc pha khác nhau. Thông thường, các góc pha được chọn là  $0$  và  $\pi$ , và hai tín hiệu này được cho bởi biểu thức sau [1]:

$$\begin{aligned} s_1(t) &= A \cos(2\pi f_c t), \quad 0 \leq t \leq T, \text{ cho bit 1} \\ s_2(t) &= -A \cos(2\pi f_c t), \quad 0 \leq t \leq T, \text{ cho bit 0}. \end{aligned} \quad (5.3)$$

Hai tín hiệu này được gọi là hai tín hiệu đối cực (antipodal). Người ta lựa chọn hai tín hiệu này bởi vì chúng có hệ số tương quan bằng  $-1$ , và do đó dẫn tới xác suất lỗi nhỏ nhất cho cùng một giá trị  $E_b/N_0$ , như chúng ta sẽ xét tới dưới đây. Hai tín hiệu này có cùng tần số cũng như năng lượng.

Bằng cách định nghĩa một hệ tọa độ hai chiều với



**Hình 5.5:** Biểu diễn tín hiệu BPSK trong không gian hai chiều.

$$\phi_1(t) = \sqrt{2/T} \cos(2\pi f_c t), \quad 0 \leq t \leq T \quad (5.4)$$

và

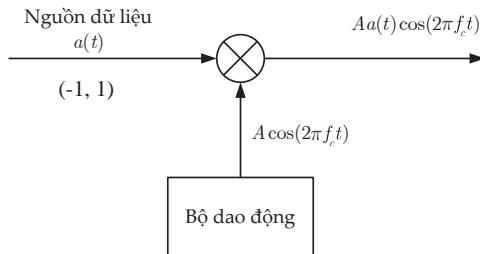
$$\phi_2(t) = -\sqrt{2/T} \sin(2\pi f_c t), \quad 0 \leq t \leq T \quad (5.5)$$

tương ứng là các trục hoành và trục tung của hệ tọa độ. Chúng ta có thể biểu diễn tín hiệu BPSK trong hệ tọa độ này như trên Hình 5.5, trong đó  $E_s$  là năng lượng của một ký hiệu BPSK (symbol energy) được cho bởi:

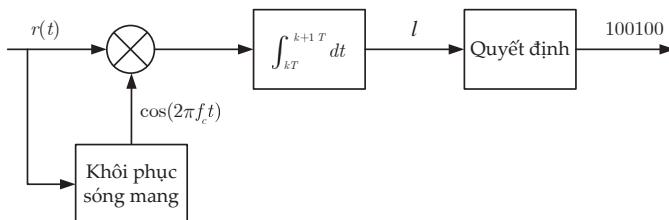
$$E_s = \frac{A^2 T}{2}.$$

Cách biểu diễn như trên Hình 5.5 được gọi là *chòm sao tín hiệu* (signal constellation) của tín hiệu BPSK. Trong cách biểu diễn này, trục hoành  $\phi_1(t)$  còn được gọi là trục I (Inphase), trục tung  $\phi_2(t)$  còn được gọi là trục Q (Quadrature). Tất cả các tín hiệu PSK và tín hiệu QAM đều có thể được biểu diễn dưới dạng tổng của hai hàm số  $\phi_1(t)$  và  $\phi_2(t)$  nên chúng cũng có thể được biểu diễn theo phương pháp đồ thị chòm sao tín hiệu tương tự như tín hiệu BPSK.

Bộ điều chế BPSK tương đối đơn giản (xem minh họa trên Hình 5.6). Trước hết, từ luồng dữ liệu nhị phân đầu vào, người ta tạo ra



a) Bộ điều chế BPSK



b) Bộ giải điều chế BPSK

**Hình 5.6:** Sơ đồ bộ điều chế/giải điều chế đồng bộ BPSK.

một luồng dữ liệu lưỡng cực  $a(t)$  như sau:

$$a(t) = \sum_{k=-\infty}^{\infty} a_k p(t - kT) \quad (5.6)$$

trong đó  $a_k \in \{-1, 1\}$ ,  $p(t)$  là một xung vuông có độ rộng xung  $T$  và biên độ bằng 1. Sau đó,  $a(t)$  được nhân với một sóng mang hình sin  $A \cos(2\pi f_c t)$ . Kết quả thu được chính là tín hiệu BPSK

$$s(t) = A a(t) \cos(2\pi f_c t), \quad -\infty \leq t \leq \infty. \quad (5.7)$$

Bộ giải điều chế đồng bộ BPSK thuộc tập hợp các bộ tách sóng đồng bộ, tức là nó có thể được thiết kế dưới dạng một bộ tương quan (correlator) hoặc một bộ lọc phối hợp (matched filter). Tín hiệu tham chiếu của bộ tương quan là tín hiệu sai phân  $s_d(t) = 2A \cos(2\pi f_c t)$ . Minh họa trên Hình 5.6(b) là máy thu đồng bộ sử dụng một bộ tương quan với tín hiệu tham chiếu là phiên bản của tín hiệu sai phân với biên độ được thu nhỏ. Tín hiệu tham chiếu phải đồng bộ với tín hiệu thu được về mặt tần số và góc pha. Nó được tạo ra

bởi mạch khôi phục sóng mang (carrier recovery). Đối với tín hiệu thông dài, người ta không khuyến nghị sử dụng bộ lọc phôi hợp thay cho bộ tương quan bởi vì việc thiết kế một bộ lọc với đáp ứng xung  $h(t) = \cos(2\pi f_c(T-t))$  là rất khó khăn.

Giả thiết, tín hiệu thu được không bị tác động của nhiễu và tạp âm, đặt  $A = 1$ , đầu ra của bộ tương quan tại  $t = (k+1)T$  là [1]:

$$\begin{aligned} l &= \int_{kT}^{(k+1)T} r(t) \cos(2\pi f_c t) dt \\ &= \frac{T}{2} a_k + \frac{a_k}{8\pi f_c} [\sin(4\pi f_c(k+1)T) - \sin(4\pi f_c kT)]. \end{aligned} \quad (5.8)$$

Nếu  $f_c = mR_b$ , với  $m$  là một số nguyên dương và  $R_b$  là tốc độ bit, thì thành phần thứ 2 trong công thức (5.8) sẽ bằng 0, và do đó tín hiệu ban đầu  $a(t)$  được khôi phục hoàn toàn (trong điều kiện không có nhiễu và tạp âm). Nếu  $f_c \neq mR_b$ , thành phần thứ 2 sẽ khác 0. Tuy nhiên, nếu  $f_c$  thỏa mãn điều kiện  $f_c \gg R_b$  thì thành phần thứ 2 sẽ rất nhỏ hơn so với thành phần đầu, nên ta có thể bỏ qua.

Xác suất lỗi bit của tín hiệu BPSK trong kênh AWGN có thể được xác định dựa trên công thức tổng quát cho các tín hiệu nhị phân như sau [1]:

$$P_b = Q\left(\sqrt{\frac{E_1 + E_2 - 2\rho_{12}\sqrt{E_1 E_2}}{2N_0}}\right) \quad (5.9)$$

trong đó  $P_b$  là xác suất lỗi bit,  $E_1$  và  $E_2$  tương ứng là năng lượng của các tín hiệu nhị phân  $s_1(t)$  và  $s_2(t)$ ,  $\rho_{12}$  là hệ số tương quan giữa  $s_1(t)$  và  $s_2(t)$ ,  $N_0$  là mật độ phổ công suất của tạp âm, và  $Q(x)$  là hàm Q của Marcum (Marcum's Q-function) được cho bởi:

$$Q(x) = \frac{1}{2\pi} \int_x^{\infty} \exp\left(-\frac{u^2}{2}\right) du.$$

Đối với tín hiệu BPSK tách tín hiệu đồng bộ, ta có  $E_1 = E_2 =$

$E_s = E_b$ ,  $\rho_{12} = -1$ , do đó

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right). \quad (5.10)$$

Nếu tín hiệu BPSK được truyền đi trong kênh pha-định Rayleigh, thì xác suất lỗi bit được cho bởi biểu thức sau [1]:

$$P_b = \frac{1}{2} \left[ 1 - \frac{1}{\sqrt{1 + 1/(E_b/N_0)}} \right] \quad (5.11)$$

trong đó  $E_b$  là năng lượng bit.

### 5.3.2 Khóa dịch pha $M$ mức

Động lực chính thúc đẩy sự ra đời của khóa dịch pha  $M$  mức (MPSK) là nhằm tăng hiệu suất băng thông của các hệ thống điều chế PSK. Đối với hệ thống BPSK, một bit dữ liệu được biểu diễn bởi một ký hiệu. Trong khi đó, đối với các hệ thống MPSK, một ký hiệu sẽ biểu diễn cho  $n = \log_2 M$  bit dữ liệu, dẫn tới hiệu suất băng thông tăng lên  $n$  lần. Trong các hệ thống MPSK, QPSK (Quadrature PSK) là hệ thống được sử dụng rộng rãi nhất do hệ thống này làm tăng hiệu suất băng thông nhưng không làm ảnh hưởng tới tỷ số lỗi bit (BER). Các hệ thống MPSK khác đều làm tăng hiệu suất băng thông nhưng phải trả giá cho phẩm chất lỗi bit của hệ thống.

Tập các tín hiệu MPSK được cho bởi biểu thức:

$$s_i(t) = A \cos(2\pi f_c t + \theta_i), \quad 0 \leq t \leq T, i = 1, 2, \dots, M \quad (5.12)$$

trong đó

$$\theta_i = \frac{(2i - 1)\pi}{M}.$$

$M$  thường được chọn là một số lũy thừa với cơ số 2, tức là  $M = 2^n$ ,  $n = \log_2 M$ . Do đó, luồng dữ liệu nhị phân được chia thành các cụm  $n$  bit. Mỗi một cụm  $n$  bit được biểu diễn bởi một ký hiệu với một góc pha đầu xác định. Biểu thức (5.12) có thể được viết dưới

dạng

$$\begin{aligned}s_i(t) &= A \cos \theta_i \cos(2\pi f_{ct} t) - A \sin \theta_i \sin(2\pi f_{ct} t) \\ &= s_{i1}\phi_1(t) + s_{i2}\phi_2(t)\end{aligned}\quad (5.13)$$

trong đó  $\phi_1(t)$  và  $\phi_2(t)$  là các hàm cơ sở trực giao như định nghĩa trong (5.4) và (5.5), còn

$$\begin{aligned}s_{i1} &= \int_0^T s_i(t)\phi_1(t)dt = \sqrt{E_s} \cos \theta_i, \\ s_{i2} &= \int_0^T s_i(t)\phi_2(t)dt = \sqrt{E_s} \sin \theta_i,\end{aligned}$$

với

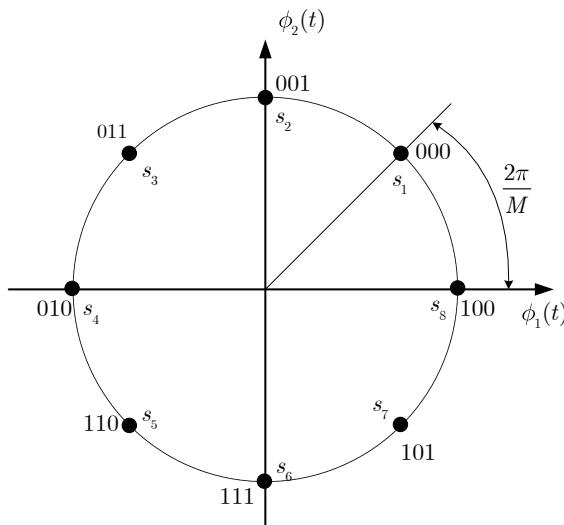
$$E_s = \frac{A^2 T}{2}$$

là năng lượng ký hiệu của tín hiệu MPSK. Góc pha đầu  $\theta_i$  có liên hệ với  $s_{i1}$  và  $s_{i2}$  theo công thức

$$\theta_i = \arctan \frac{s_{i2}}{s_{i1}}.$$

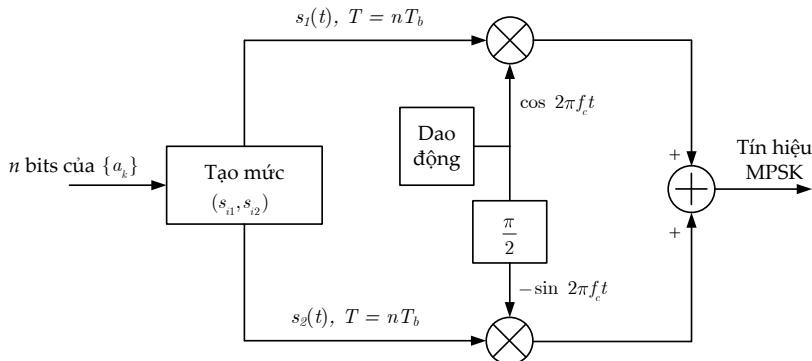
Như vậy, ta có thể thấy rằng chòm sao tín hiệu MPSK nằm trong một không gian hai chiều với hai trục tọa độ là  $\phi_1(t)$  và  $\phi_2(t)$  mà ở đó mỗi tín hiệu  $s_i(t)$  được biểu diễn bởi một điểm  $(s_{i1}, s_{i2})$ . Nếu xét trong hệ tọa độ cực thì mỗi một tín hiệu  $s_i(t)$  sẽ được biểu diễn bởi một điểm  $(\sqrt{E_s}, \theta_i)$ , tức là, độ lớn bằng  $\sqrt{E_s}$  và góc pha so với trục hoành bằng  $\theta_i$ . Các điểm tín hiệu trong chòm sao tín hiệu MPSK phân bố đều nhau trên một đường tròn có bán kính  $\sqrt{E_s}$  và có tâm tại gốc tọa độ. Các cụm  $n$  bit có thể được ánh xạ một cách bất kỳ với điều kiện phép ánh xạ đó là một-một. Tuy nhiên, trong thực tế, người ta thường sử dụng một phương pháp gọi là phương pháp mã hóa Gray để gán các cụm bit vào các tín hiệu MPSK. Trong phương pháp mã hóa Gray, các cụm  $n$  bit dữ liệu được gán cho các điểm tín hiệu trong chòm sao tín hiệu sao cho hai điểm tín hiệu nằm kề nhau chỉ khác nhau 1 bit. Khi một ký hiệu MPSK bị lỗi trong quá trình tách sóng, thì có một khả năng lớn là ký hiệu được tách sóng chính là ký hiệu nằm liền kề với ký hiệu được phát đi. Như vậy, nhờ

phương pháp mã hóa Gray, sẽ chỉ có 1 bit lỗi trong  $n$  bit được phát đi, dẫn tới phẩm chất lỗi bit của hệ thống được nâng lên. Minh họa trên Hình 5.7 là chòm sao tín hiệu 8PSK sử dụng phương pháp mã hóa Gray để ánh xạ các cụm 3 bit vào các ký hiệu.

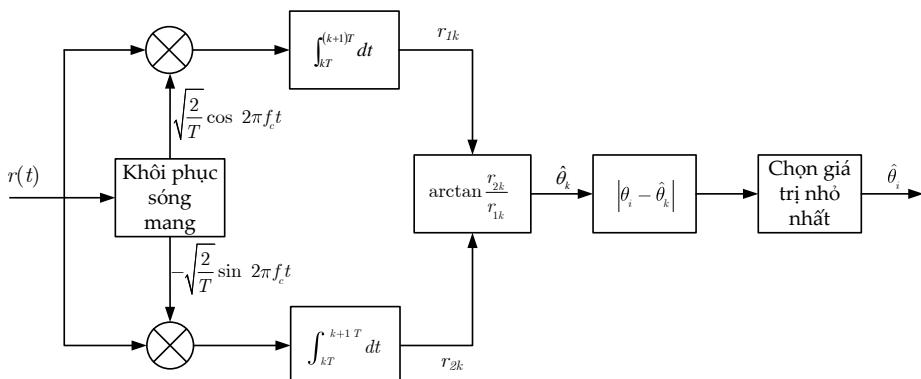


Hình 5.7: Chòm sao tín hiệu 8PSK sử dụng mã hóa Gray.

Khi  $M \geq 4$ , bộ điều chế MPSK có thể được thực hiện bởi một bộ điều chế cầu phương như minh họa trên Hình 5.8(a). Từ hình vẽ ta thấy rằng sự khác biệt duy nhất giữa các bộ điều chế MPSK chính là ở bộ tạo mức (level generator). Với mỗi giá trị  $M$ , bộ tạo mức được điều khiển bởi từng cụm  $n = \log_2 M$  bit dữ liệu đầu vào.



a) Bộ điều chế MPSK



b) Bộ giải điều chế MPSK đồng bộ

Hình 5.8: Sơ đồ điều chế và giải điều chế MPSK.

### Xác suất lỗi ký hiệu trong kênh AWGN

Đối với kỹ thuật điều chế MPSK, thông tin thống kê đủ để có thể khôi phục tín hiệu được cho bởi biểu thức:

$$\begin{aligned} l_i &= \int_0^T r(t)s_i(t)dt = \int_0^T r(t)[s_{1i}\phi_1(t) + s_{2i}\phi_2(t)]dt \\ &= \sqrt{E_s}[r_1 \cos \theta_i + r_2 \sin \theta_i] \end{aligned} \quad (5.14)$$

trong đó

$$\begin{aligned} r_1 &= \int_0^T r(t)\phi_1(t)dt = \int_0^T [s(t) + n(t)]\phi_1(t)dt = s_{i1} + z_1 \\ r_2 &= \int_0^T r(t)\phi_2(t)dt = \int_0^T [s(t) + n(t)]\phi_2(t)dt = s_{i2} + z_2 \end{aligned}$$

là các biến ngẫu nhiên Gauss độc lập với trị trung bình tương ứng là  $s_{i1}$  và  $s_{i2}$  và phương sai là  $N_0/2$ .

Sau một số phép biến đổi toán học, ta có thể viết:

$$l_i = \sqrt{E_s} \rho \cos(\theta_i - \hat{\theta}) \quad (5.15)$$

với

$$\rho = \sqrt{r_1^2 + r_2^2}$$

và

$$\hat{\theta} = \arctan \frac{r_2}{r_1}.$$

Khi hệ thống không bị tác động của nhiễu và tạp âm, ta có

$$\hat{\theta} = \arctan r_2/r_1 = \arctan s_{i2}/s_{i1} = \theta_i.$$

Dưới tác động của nhiễu và tạp âm,  $\hat{\theta}$  bị sai lệch đi so với  $\theta_i$ . Do  $\rho$  không phụ thuộc vào các tín hiệu MPSK nên việc lựa chọn giá trị  $l_i$  lớn nhất trong số  $M$  giá trị tương đương với lựa chọn giá trị  $|\theta_i - \hat{\theta}|$  nhỏ nhất. Hình 5.8(b) trình bày bộ giải điều chế đồng bộ MPSK dựa trên quy tắc quyết định này. Chỉ số  $k$  trên hình vẽ biểu thị chu kỳ ký hiệu thứ  $k$ . Chú ý, biên độ có của tín hiệu tham chiếu có thể có giá trị bất kỳ (trên hình vẽ là  $\sqrt{T/2}$ , do biên độ sẽ bị triệt tiêu khi tính toán  $\hat{\theta}_k$ ).

Xác suất lỗi ký hiệu (symbol error probability) của tín hiệu MPSK trong kênh truyền AWGN được tính toán dựa trên công thức sau [1]:

$$P_s = 1 - \int_{-\pi/M}^{\pi/M} p(\varphi) d\varphi \quad (5.16)$$

với

$$p(\varphi) = \frac{e^{(-E_s/N_o)}}{2\pi} \left\{ 1 + \sqrt{\frac{\pi E_s}{N_0}} (\cos \varphi) e^{((E_s/N_0) \cos^2 \varphi)} \left[ 1 + \operatorname{erf} \left( \sqrt{\frac{E_s}{N_0}} \cos \varphi \right) \right] \right\}$$

và

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

là hàm số lỗi (error function).

Đối với các trường hợp  $M > 4$ , xác suất lỗi ký hiệu trong (5.16) không thể tính toán được ở dạng biểu thức rút gọn mà chỉ có thể tính toán được dựa trên phương pháp số (numerical). Với trường hợp  $M = 4$  (QPSK), xác suất lỗi ký hiệu trong (5.16) được tính và rút gọn thành công thức sau:

$$P_s = 2Q\left(\sqrt{\frac{E_s}{N_0}}\right) - \left[Q\left(\sqrt{\frac{E_s}{N_0}}\right)\right]^2. \quad (5.17)$$

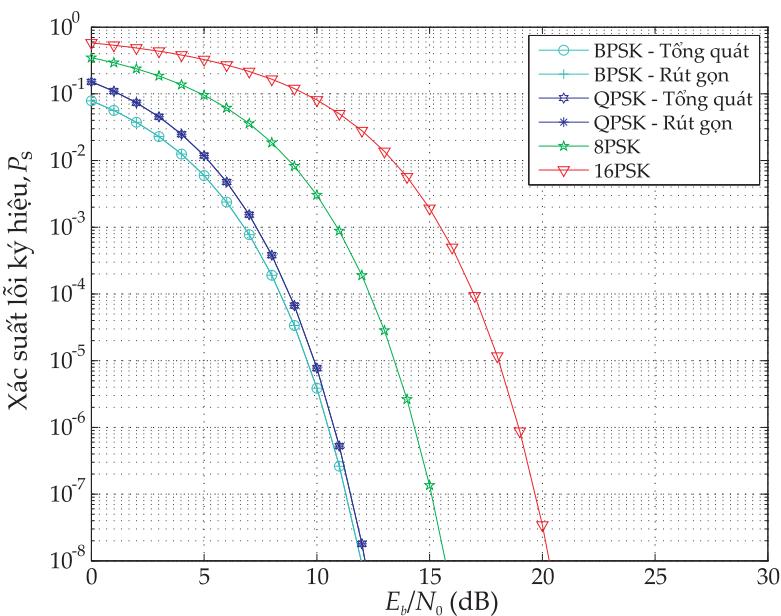
Biểu thức (5.16) còn có thể được viết dưới một dạng khác như sau:

$$\begin{aligned} P_s = & \frac{M-1}{M} - \frac{1}{2}\operatorname{erf}\left[\sqrt{\frac{E_s}{N_0}} \sin \frac{\pi}{M}\right] \\ & - \frac{1}{\pi} \int_0^{\sqrt{E_s/N_0} \sin \pi/M} e^{-y^2} \operatorname{erf}\left(y \cot \frac{\pi}{M}\right) dy. \end{aligned} \quad (5.18)$$

Minh họa trên Hình 5.9 là xác suất lỗi ký hiệu  $P_s$  của tín hiệu MPSK với  $M = 2, 4, 8$  và  $16$  được tính theo công thức tổng quát (general equation) (5.18). Để so sánh, Hình 5.9 cũng minh họa xác suất lỗi ký hiệu  $P_s$  của tín hiệu BPSK và QPSK được tính tương ứng theo các công thức rút gọn (closed form) (5.10) và (5.17). Từ hình vẽ, ta thấy rõ ràng rằng công thức (5.18) cho kết quả hoàn toàn chính xác với các công thức (5.10) và (5.17), tương ứng với  $M = 2$  và  $M = 4$ . Ta cũng thấy rằng, với các trường hợp  $M > 4$ , nếu chúng ta tăng gấp đôi số lượng góc pha (tương ứng với việc tăng 1 bit trong cụm  $n$  bit) thì tỷ lệ tín hiệu trên tạp âm (SNR) phải tăng lên rất đáng kể để có thể giữ nguyên xác suất lỗi ký hiệu. Ví dụ, tại giá trị  $P_s = 10^{-5}$ , sai lệch SNR giữa  $M = 4$  và  $M = 8$  là xấp xỉ  $4$  dB, và sai lệch SNR giữa  $M = 8$  và  $M = 16$  là khoảng  $5$  dB.

Khi hệ thống hoạt động trong dải tỷ số SNR đủ lớn, tức là  $E_s/N_0 \gg 1$ , xác suất lỗi ký hiệu  $P_s$  có thể được tính gần đúng dựa trên biểu thức sau:

$$P_s \approx \operatorname{erfc}\left(\sqrt{\frac{E_s}{N_0}} \sin \frac{\pi}{M}\right) = 2Q\left(\sqrt{\frac{2E_s}{N_0}} \sin \frac{\pi}{M}\right), \quad (5.19)$$



Hình 5.9: Xác suất lỗi ký hiệu của tín hiệu MPSK trong kênh AWGN.

trong đó

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) = 2Q\left(\sqrt{2}x\right)$$

là hàm lỗi bù (complementary error function).

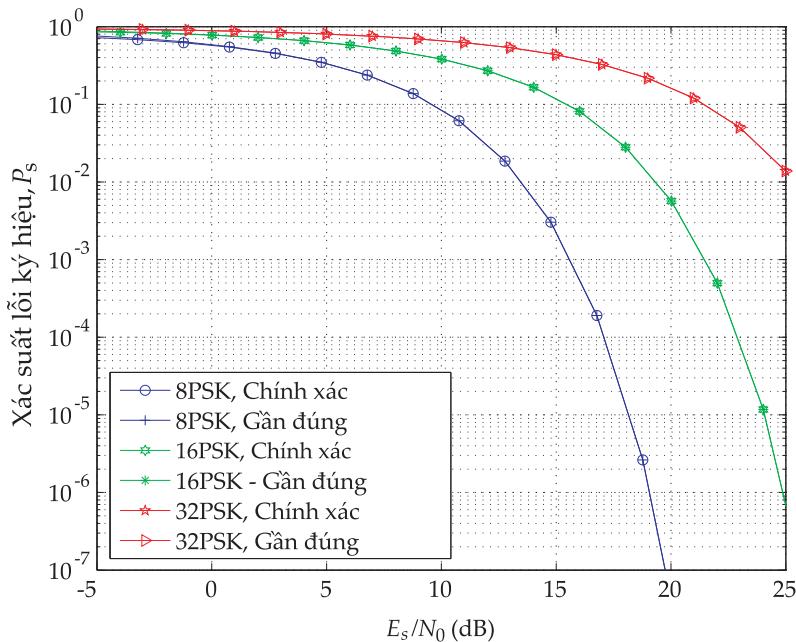
Như chúng ta thấy trên Hình 5.10, đối với tín hiệu 8PSK, công thức xấp xỉ (5.19) cho kết quả gần như chính xác tuyệt đối so với công thức tổng quát (5.18) khi tỷ số  $E_s/N_0 \geq 3$  dB (hay  $E_s/N_0 \geq 2$ ). Với các trường hợp  $M > 8$ , công thức (5.19) còn cho kết quả gần như chính xác ngay cả khi  $E_s/N_0 < 3$  dB (hay  $E_s/N_0 < 2$ ).

### Xác suất lỗi bit trong kênh AWGN

Đối với tín hiệu MPSK sử dụng mã hóa Gray, xác suất lỗi bit  $P_b$  có thể được tính gần đúng theo biểu thức:

$$P_b \approx P_{bu}^g = \frac{P_s}{\log_2 M}, \quad (5.20)$$

trong đó  $P_{bu}^g$  là đường bao cận trên (upper bound) của xác suất lỗi bit trong kênh AWGN, tức là  $P_b \leq P_{bu}^g$ .



**Hình 5.10:** Xác suất lỗi ký hiệu gần đúng của tín hiệu MPSK.

Tuy nhiên, đối với các trường hợp  $M = 4$  và  $M = 8$ , chúng ta có thể tính chính xác xác suất lỗi bit của tín hiệu MPSK. Đối với tín hiệu QPSK, xác suất lỗi bit được tính như sau [1]-[2]:

$$P_b = \frac{1}{2} \operatorname{erfc} \left( \sqrt{\frac{E_b}{N_0}} \right). \quad (5.21)$$

Đối với tín hiệu 8PSK, xác suất lỗi bit được tính như sau [3]:

$$P_b = \frac{1}{3} \operatorname{erfc} \left( \sqrt{\frac{3E_b}{N_0}} \sin \frac{\pi}{8} \right) + \frac{1}{3} \operatorname{erfc} \left( \sqrt{\frac{3E_b}{N_0}} \sin \frac{3\pi}{8} \right) \left[ 1 - \operatorname{erfc} \left( \sqrt{\frac{3E_b}{N_0}} \sin \frac{\pi}{8} \right) \right]. \quad (5.22)$$

Xác suất lỗi bit cho các cấu hình  $M > 8$  có thể được tính gần đúng theo công thức [3]:

$$P_b \approx P_{bl}^g = \frac{1}{\log_2 M} \left[ \operatorname{erfc} \left( \sqrt{\frac{E_b \log_2 M}{N_0}} \sin \frac{\pi}{M} \right) + \operatorname{erfc} \left( \sqrt{\frac{E_b \log_2 M}{N_0}} \sin \frac{3\pi}{M} \right) \right] \quad (5.23)$$

trong đó  $P_{bl}^g$  là đường bao cận dưới (lower bound) của xác suất lỗi bit, tức là  $P_b \geq P_{bl}^g$ .

### Xác suất lỗi ký hiệu trong kênh pha-định Rayleigh

Trong kênh pha-định Rayleigh, tỷ số SNR không còn là một giá trị xác định (như trong kênh AWGN) mà là một giá trị ngẫu nhiên có dạng  $\gamma_s = \zeta^2 E_s / N_0$ , với biến ngẫu nhiên  $\zeta$  có phân bố Rayleigh. Lúc này, tương ứng với một giá trị  $\gamma_s$ , ta có một xác suất lỗi ký hiệu tức thời  $P_s(\gamma_s)$ . Nói cách khác,  $P_s(\gamma_s)$  là xác suất lỗi ký hiệu có điều kiện  $\gamma_s$ . Khi  $\zeta$  có phân bố Rayleigh thì  $\gamma_s$  có phân bố  $\chi^2$  chi-square với hai bậc tự do. Vì vậy, xác suất lỗi ký hiệu trung bình trong kênh Rayleigh được tính theo công thức [1]:

$$P_s = \int_0^\infty P_s(\gamma_s) p(\gamma_s) d\gamma_s \quad (5.24)$$

trong đó,  $p(\gamma_s)$  là hàm mật độ xác suất của  $\gamma_s$  có dạng:

$$p(\gamma_s) = \frac{1}{\Gamma} \exp\left(-\frac{\gamma_s}{\Gamma}\right), \quad \Gamma \geq 0 \quad (5.25)$$

với

$$\Gamma = E\{\zeta^2\} \frac{E_s}{N_0} = E\{\zeta^2\} \frac{E_b \log_2 M}{N_0}$$

là tỷ số SNR trung bình,  $E_b$  là năng lượng bit.

Dựa vào công thức (5.16), ta có thể viết xác suất có điều kiện  $P_s(\gamma_s)$  đối với tín hiệu MPSK như sau:

$$P_s(\gamma_s) = 1 - \int_{-\pi/M}^{\pi/M} \frac{e^{-\gamma_s}}{2\pi} \left\{ 1 + \sqrt{\pi\gamma_s} (\cos \varphi) e^{\gamma_s \cos^2 \varphi} [1 + \operatorname{erf}(\sqrt{\gamma_s} \cos \varphi)] \right\} d\varphi. \quad (5.26)$$

Do đó, sau một số phép biến đổi, xác suất lỗi ký hiệu trung bình của tín hiệu MPSK trong kênh pha-định Rayleigh được tính

theo công thức:

$$\begin{aligned} P_s = & 1 - \frac{1}{M(1+\Gamma)} \\ & - \int_0^{\infty} \int_{-\pi/M}^{\pi/M} \frac{\sqrt{\gamma_s} e^{-\gamma_s(\sin^2 \varphi + 1/\Gamma)} \cos \varphi}{2\Gamma\sqrt{\pi}} [1 + \operatorname{erf}(\sqrt{\gamma_s} \cos \varphi)] d\varphi d\gamma_s. \end{aligned} \quad (5.27)$$

Đối với các trường hợp  $M \geq 4$ , xác suất lỗi bit trung bình trong (5.27) không thể rút gọn mà chỉ có thể tính được bằng phương pháp số. Ta có thể sử dụng hàm dblquad() của Matlab để tính toán thành phần tích phân kép rồi từ đó tính ra  $P_s$ .

Một phương pháp đơn giản hơn để xác định  $P_s$  là sử dụng công thức tính gần đúng (5.19) cho vùng giá trị SNR trung bình đủ lớn, tức là  $\Gamma \gg 1$ . Theo công thức này, xác suất có điều kiện  $P_s(\gamma_s)$  được cho bởi:

$$P_s(\gamma_s) \approx 2Q\left(\sqrt{2\gamma_s} \sin \frac{\pi}{M}\right). \quad (5.28)$$

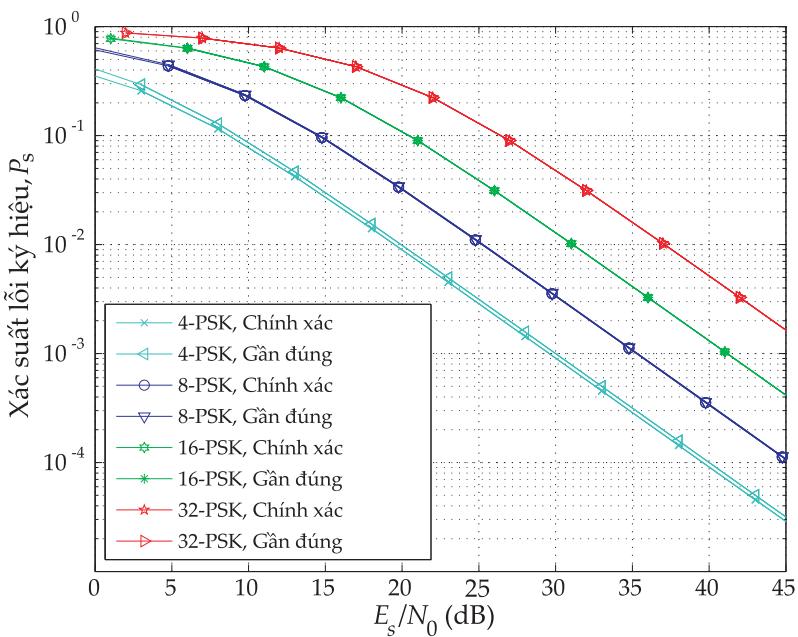
Do đó, xác suất lỗi ký hiệu trung bình có thể được tính xấp xỉ bằng:

$$P_s \approx \int_0^{\infty} 2Q\left(\sqrt{2\gamma_s} \sin \frac{\pi}{M}\right) \frac{e^{-\gamma_s/\Gamma}}{\Gamma} d\gamma_s. \quad (5.29)$$

Sau một số bước tính toán, ta có thể viết:

$$P_s \approx 1 - \frac{1}{\sqrt{1 + \frac{1}{\Gamma \sin^2(\pi/M)}}}, \quad M \geq 4. \quad (5.30)$$

Minh họa trên Hình 5.11 là xác suất lỗi ký hiệu của các tín hiệu MPSK với  $M = 4, 8, 16$ , và 32 sử dụng công thức tổng quát (5.27) và công thức gần đúng (5.30). Từ hình vẽ ta thấy rằng công thức gần đúng (5.30) cho kết quả gần như chính xác tuyệt đối so với công thức tổng quát (5.27) khi  $M > 4$ . Đối với trường hợp  $M = 4$ , có một sự sai lệch rất nhỏ giữa kết quả gần đúng và kết quả chính xác. Sự sai lệch này là do chúng ta đã bỏ qua thành phần thứ 2 trong (5.17) khi tính toán gần đúng xác suất lỗi ký hiệu của tín hiệu QPSK trong kênh Rayleigh.



Hình 5.11: Xác suất lỗi ký hiệu của tín hiệu MPSK trong kênh pha-định Rayleigh.

### Xác suất lỗi bit trong kênh pha-định Rayleigh

Tương tự như trong kênh AWGN, xác suất lỗi bit của tín hiệu MPSK có thể được tính gần đúng dựa trên công thức:

$$P_b \approx P_{bu}^r = \frac{P_s}{\log_2 M} \quad (5.31)$$

trong đó  $P_{bu}^r$  là đường bao cận trên của xác suất lỗi bit trong kênh Rayleigh.

Tuy nhiên, đối với tín hiệu QPSK, xác suất lỗi bit trong kênh Rayleigh có thể được tính chính xác theo công thức [1]-[2]:

$$P_b = \frac{1}{2} \left[ 1 - \frac{1}{\sqrt{1 + \left( \frac{E_b}{N_0} \right)^{-1}}} \right]. \quad (5.32)$$

Đối với các tín hiệu MPSK còn lại (tức là,  $M \geq 8$ ), ta có thể tính gần

đúng xác suất lỗi bit như sau:

$$P_b \approx P_{ul}^r = \frac{1}{\log_2 M} \left[ 2 - \frac{1}{\sqrt{1 + \left( \frac{E_b \log_2 M}{N_0} \sin^2 \frac{\pi}{M} \right)^{-1}}} - \frac{1}{\sqrt{1 + \left( \frac{E_b \log_2 M}{N_0} \sin^2 \frac{3\pi}{M} \right)^{-1}}} \right] \quad (5.33)$$

trong đó  $P_{ul}^r$  là đường bao cận dưới (lower bound) của xác suất lỗi bit của các hệ thống MPSK trong kênh pha-định Rayleigh.

## 5.4 Kỹ thuật điều chế biên độ cầu phương QAM

Kỹ thuật điều chế PSK mà chúng ta nghiên cứu ở trên có một đặc điểm quan trọng là biên độ của tín hiệu PSK không thay đổi dưới tác động của dữ liệu điều chế (đường bao sóng mang là hằng số). Đặc tính này của tín hiệu PSK (cũng như của một số tín hiệu khác như FSK, ...) rất quan trọng đối với các hệ thống mà ở đó các bộ khuếch đại công suất phải hoạt động trong vùng phi tuyến của đường đặc tuyến vào - ra nhằm đạt hiệu suất khuếch đại lớn nhất, chẳng hạn như ở các bộ phát đáp vệ tinh. Tuy nhiên, trong một số hệ thống thông tin, đặc tính đường bao không đổi có thể không phải là một yêu cầu quan trọng mà hiệu suất băng thông mới cần thiết. Đối với những hệ thống như vậy, chúng ta có thể sử dụng kỹ thuật điều chế biên độ cầu phương QAM.

QAM thuộc tập hợp các kỹ thuật điều chế có đường bao sóng mang thay đổi và có ưu điểm là đạt hiệu suất băng thông cao hơn kỹ thuật điều chế MPSK ứng với cùng một công suất tín hiệu trung bình. QAM được sử dụng rộng rãi trong các modem được thiết kế để truyền tín hiệu qua các kênh thoại.

### Mô tả tín hiệu QAM

Tín hiệu QAM là tín hiệu được điều chế cả về biên độ và góc pha và được biểu diễn bởi biểu thức sau

$$s_i(t) = A_i \cos(2\pi f_c t + \theta_i), \quad i = 1, 2, \dots, M \quad (5.34)$$

trong đó  $A_i$  là biên độ và  $\theta_i$  là góc pha của tín hiệu thứ  $i$  trong tập  $M$  tín hiệu QAM. Để cải thiện phổ tần của tín hiệu QAM, qua đó giúp kiểm soát hiện tượng giao thoa giữa các ký hiệu (ISI), người ta thường sử dụng kỹ thuật định dạng xung (pulse shaping). Khi xung định dạng  $p(t)$ ,  $t \leq 0 \leq T$  được sử dụng, tín hiệu QAM được viết như sau

$$s_i(t) = A_i p(t) \cos(2\pi f_c t + \theta_i), \quad i = 1, 2, \dots, M. \quad (5.35)$$

Tương tự tín hiệu MPSK, tín hiệu QAM cũng có thể được biểu diễn dưới dạng tổng của hai hàm số trực giao chuẩn hóa (orthonormal). Tức là, ta có thể biểu diễn (5.35) dưới dạng

$$s_i(t) = s_{i1}\phi_1(t) + s_{i2}\phi_2(t), \quad (5.36)$$

trong đó

$$\phi_1(t) = \sqrt{\frac{2}{E_p}} p(t) \cos 2\pi f_c t, \quad t \leq 0 \leq T \quad (5.37)$$

$$\phi_2(t) = -\sqrt{\frac{2}{E_p}} p(t) \sin 2\pi f_c t, \quad t \leq 0 \leq T \quad (5.38)$$

và

$$s_{i1} = \sqrt{\frac{E_p}{2}} A_{i1} = \sqrt{\frac{E_p}{2}} A_i \cos \theta_i \quad (5.39)$$

$$s_{i2} = \sqrt{\frac{E_p}{2}} A_{i2} = \sqrt{\frac{E_p}{2}} A_i \sin \theta_i \quad (5.40)$$

với  $E_p = \int_0^T p^2(t) dt$  là năng lượng của  $p(t)$ . Trong trường hợp không có định dạng xung,  $p(t) = 1$ ,  $t \leq 0 \leq T$ , và do đó  $E_p = T$ .

Nếu  $f_c \gg 1/T$ , ta có thể dễ dàng chứng minh được rằng  $\phi_1(t)$  và  $\phi_2(t)$  là các hàm trực giao chuẩn hóa, tức là:

$$\int_0^T \phi_i^2(t) dt \cong 1, \quad i = 1, 2$$

và

$$\int_0^T \phi_1(t) \phi_2(t) dt \cong 0.$$

Từ công thức (5.36), năng lượng của tín hiệu thứ  $i$  bằng:

$$E_i = \int_0^T s_i^2(t) dt \cong \frac{1}{2} A_i^2 E_p \quad (5.41)$$

và năng lượng trung bình của tín hiệu QAM bằng:

$$E_{avg} = \frac{1}{2} E_p \text{E}\{A_i^2\}, \quad (5.42)$$

trong đó  $\text{E}\{a\}$  là trung bình thống kê của biến ngẫu nhiên  $a$ .

Công suất trung bình và biên độ trung bình lần lượt bằng:

$$P_{avg} = \frac{E_{avg}}{T}, \quad (5.43)$$

$$A_{avg} = \sqrt{2P_{avg}}. \quad (5.44)$$

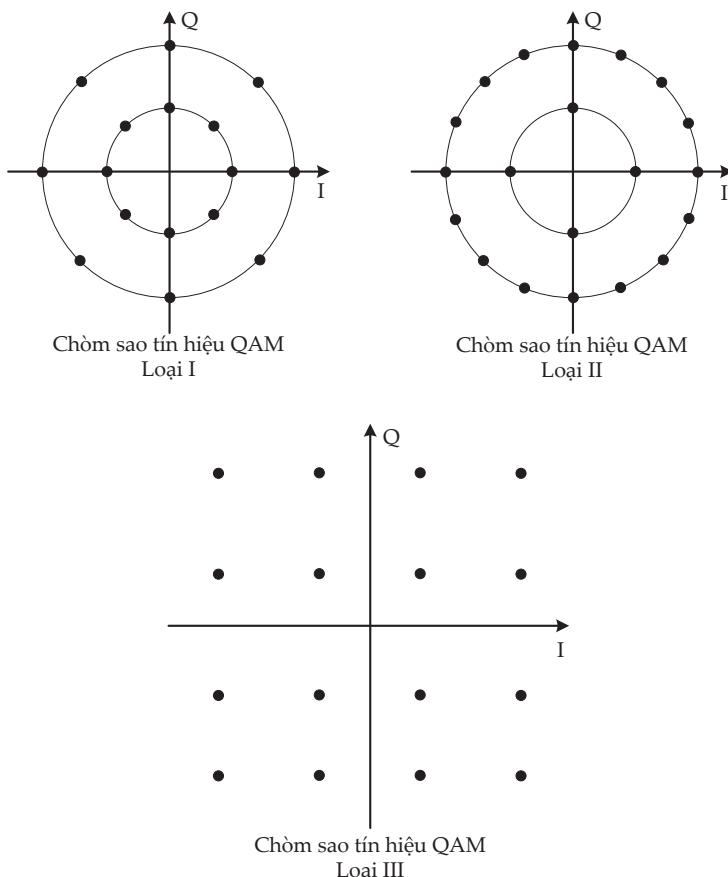
Tương tự như tín hiệu MPSK, tín hiệu QAM cũng có thể được biểu diễn bằng các chòm sao tín hiệu với trực hoành  $\phi_1(t)$  tương ứng với trực I và trực tung  $\phi_2(t)$  tương ứng với trực Q. Trong chòm sao tín hiệu QAM, mỗi một tín hiệu QAM được biểu diễn bằng một điểm với tọa độ  $(s_{i1}, s_{i2})$ . Minh họa trên Hình 5.12 ba dạng chòm sao tín hiệu QAM. Trong số ba dạng này, chúng ta sẽ tập trung nghiên cứu chòm sao tín hiệu QAM vuông (dạng III).

### **QAM vuông**

Đối với tín hiệu QAM vuông  $M$  điểm, công thức (5.36) có thể được viết lại dưới dạng sau:

$$s_i(t) = I_i \sqrt{\frac{E_0}{2}} \phi_1(t) + Q_i \sqrt{\frac{E_0}{2}} \phi_2(t) \quad (5.45)$$

trong đó,  $E_0$  là năng lượng của tín hiệu có biên độ nhỏ nhất, và  $(I_i, Q_i)$  là một cặp số nguyên độc lập xác định vị trí của điểm tín hiệu trong chòm sao tín hiệu. Giá trị nhỏ nhất của cặp  $(I_i, Q_i)$  là  $(\pm 1, \pm 1)$ . Cặp số  $(I_i, Q_i)$  là một phần tử của một ma trận kích thước



**Hình 5.12:** Một số dạng chòm sao tín hiệu QAM.

$L \times L$  như sau:

$$[I_i, Q_i] = \begin{bmatrix} (-L+1, L-1) & (-L+3, L-1) & \cdots & (L-1, L-1) \\ (-L+1, L-3) & (-L+3, L-3) & \cdots & (L-1, L-3) \\ \vdots & \vdots & \ddots & \vdots \\ (-L+1, -L+1) & (-L+3, -L+1) & \cdots & (L-1, -L+1) \end{bmatrix} \quad (5.46)$$

trong đó  $L = \sqrt{M} \cdot M = 4^n, n = 1, 2, 3, \dots$

Ví dụ, với 16-QAM,  $L = 4$ , và ta có ma trận:

$$[I_i, Q_i] = \begin{bmatrix} (-3, 3) & (-1, 3) & (1, 3) & (3, 3) \\ (-3, 1) & (-1, 1) & (1, 1) & (3, 1) \\ (-3, -1) & (-1, -1) & (1, -1) & (3, -1) \\ (-3, -3) & (-1, -3) & (1, -3) & (3, -3) \end{bmatrix}. \quad (5.47)$$

Khi sử dụng cặp số  $(I_i, Q_i)$ , tọa độ của các điểm tín hiệu trong chòm sao tín hiệu QAM được biểu diễn dưới dạng phasor theo công thức:

$$\mathbf{s}_i = \left( I_i \sqrt{\frac{E_0}{2}}, Q_i \sqrt{\frac{E_0}{2}} \right), \quad i = 1, 2, \dots, M. \quad (5.48)$$

Biên độ và góc pha của phasor bằng:

$$\|\mathbf{s}_i\| = \sqrt{\frac{E_0}{2}(I_i^2 + Q_i^2)} \quad (5.49)$$

$$\theta_i = \arctan \frac{Q_i}{I_i}. \quad (5.50)$$

Năng lượng trung bình của tín hiệu QAM bằng:

$$E_{avg} = E \left\{ \frac{E_0}{2}(I_i^2 + Q_i^2) \right\} = E_0 E \{ I_i^2 \}. \quad (5.51)$$

Công suất trung bình của tín hiệu QAM bằng;

$$P_{avg} = \frac{E_0}{T} E \{ I_i^2 \}. \quad (5.52)$$

Đối với tín hiệu QAM vuông, ta dễ dàng tính được

$$\begin{aligned} E \{ I_i^2 \} &= \frac{1}{L} [(-(L-1))^2 + (-(L-3))^2 + \dots + (L-3)^2 + (L-1)^2] \\ &= \frac{1}{3}(L^2 - 1) = \frac{1}{3}(M-1). \end{aligned} \quad (5.53)$$

Do đó

$$P_{avg} = \frac{E_0}{3T} (M-1) = \frac{P_0}{3} (M-1). \quad (5.54)$$

### Bộ điều chế QAM

Bộ điều chế QAM có cấu trúc gần giống với cấu trúc của bộ điều chế PSK do chúng đều hoạt động dựa trên hai sóng mang trực giao. Để xây dựng bộ điều chế QAM, trước hết ta biểu diễn tín hiệu QAM dưới dạng:

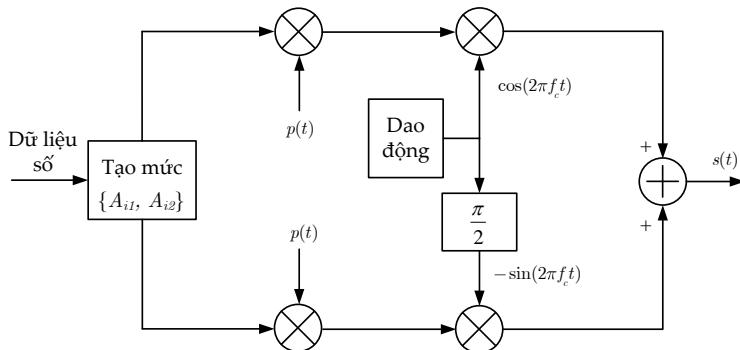
$$s(t) = s_1(t) \cos 2\pi f_c t - s_2(t) \sin 2\pi f_c t \quad (5.55)$$

với

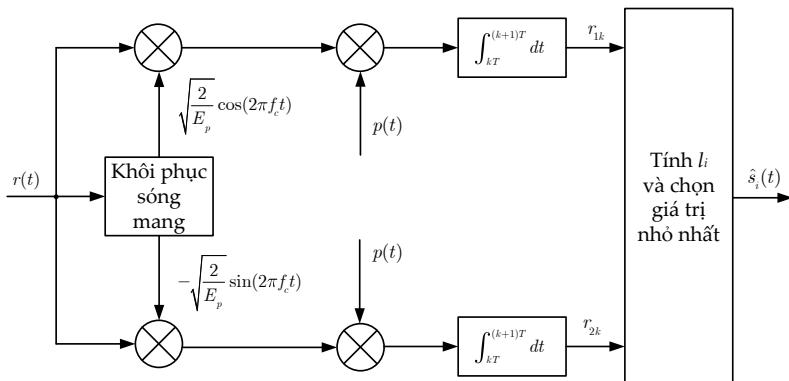
$$s_1(t) = \sum_{-\infty}^{\infty} A_{k1} p(t - kT) \quad (5.56)$$

$$s_2(t) = \sum_{-\infty}^{\infty} A_{k2} p(t - kT). \quad (5.57)$$

Dựa vào biểu thức (5.55), ta có thể xây dựng bộ điều chế QAM như



a) Điều chế QAM

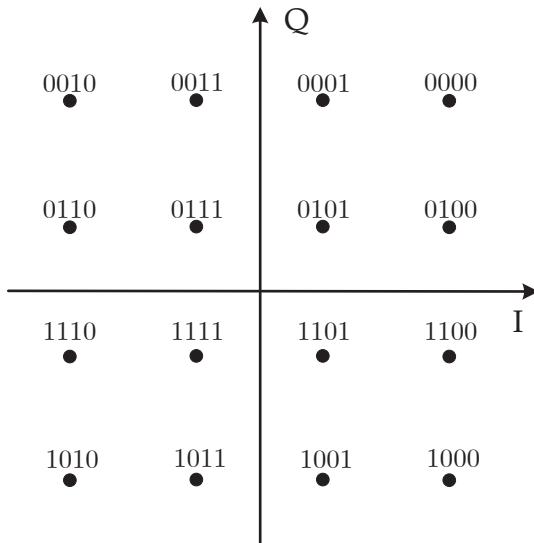


b) Giải điều chế QAM đồng bộ

**Hình 5.13:** Sơ đồ bộ điều chế và giải điều chế QAM.

trên Hình 5.13(a). Tín hiệu nhị phân tại đầu vào bộ điều chế QAM được chia thành các cụm  $n$  bit. Có tất cả  $M = 2^n$  cụm  $n$  bit khác nhau. Các cụm  $n$  bit được sử dụng để điều khiển bộ tạo mức (Level Generator) nhằm tạo ra các tọa độ (hay các giá trị kênh I và kênh Q) của các điểm tín hiệu QAM. Phép ánh xạ từ các cụm  $n$  bit vào các điểm tín hiệu QAM thường được thực hiện theo phương pháp mã

hóa Gray nhằm giảm thiểu số bit lỗi. Đối với tín hiệu QAM vuông, ta có thể thực hiện mã hóa Gray một cách hoàn chỉnh (xem minh họa trên Hình 5.14).



Hình 5.14: Chòm sao tín hiệu 16-QAM với mã hóa Gray.

Trong kênh truyền AWGN, tín hiệu tại máy thu là:

$$r(t) = s_i(t) + z(t). \quad (5.58)$$

Đối với tín hiệu QAM, thông tin thống kê đủ để có thể khôi phục tín hiệu là khoảng cách bình phương:

$$l_i = (r_1 - s_{i1})^2 + (r_2 - s_{i2})^2 \quad (5.59)$$

trong đó

$$r_1 \triangleq \int_0^T r(t) \phi_1(t) dt \approx s_{i1} + z_1$$

$$r_2 \triangleq \int_0^T r(t) \phi_2(t) dt \approx s_{i2} + z_2$$

là các biến ngẫu nhiên Gauss độc lập với trị trung bình lần lượt là  $s_{i1}$  và  $s_{i2}$ , phương sai là  $N_0/2$ . Cặp giá trị  $(r_1, r_2)$  xác định một điểm trong mặt phẳng chòm sao tín hiệu. Nó biểu thị các điểm tín hiệu

thu được dưới tác động của tạp âm. Bộ tách sóng sẽ so sánh khoảng các từ điểm  $(r_1, r_2)$  tới tất cả các điểm  $(s_{i1}, s_{i2})$  trong chòm sao tín hiệu rồi từ đó chọn ra điểm  $(s_{i1}, s_{i2})$  nào gần với  $(r_1, r_2)$  nhất (hay  $l_i$  nhỏ nhất).

Hình 5.13(b) trình bày bộ giải điều chế đồng bộ QAM dựa trên nguyên lý tách sóng trình bày ở trên, trong đó chỉ số  $k$  biểu thị chu kỳ tín hiệu thứ  $k$ . Chú ý rằng, đối với tín hiệu QAM vuông, kênh I và kênh Q có thể được tách sóng hoàn toàn độc lập với nhau. Tức là, ta sử dụng  $r_{1k}$  để khôi phục thành phần  $s_{i1}$ , sử dụng  $r_{2k}$  để khôi phục thành phần  $s_{i2}$  rồi kết hợp để xác định  $s_i(t)$ .

### Xác suất lỗi ký hiệu trong kênh truyền AWGN

Trong kênh truyền AWGN, xác suất lỗi ký hiệu của tín hiệu QAM vuông được cho bởi công thức [1]:

$$P_s = 2P_{\sqrt{M}} - P_{\sqrt{M}}^2 \quad (5.60)$$

trong đó,

$$P_{\sqrt{M}} = \frac{2(\sqrt{M} - 1)}{\sqrt{M}} Q\left(\sqrt{\frac{3E_{avg}}{(M-1)N_0}}\right) \quad (5.61)$$

và  $E_{avg}/N_0 = \log_2 M \frac{E_b}{N_0}$  là tỷ số SNR trung bình của mỗi ký hiệu QAM,  $\frac{E_b}{N_0}$  là tỷ số SNR trung bình của bit.

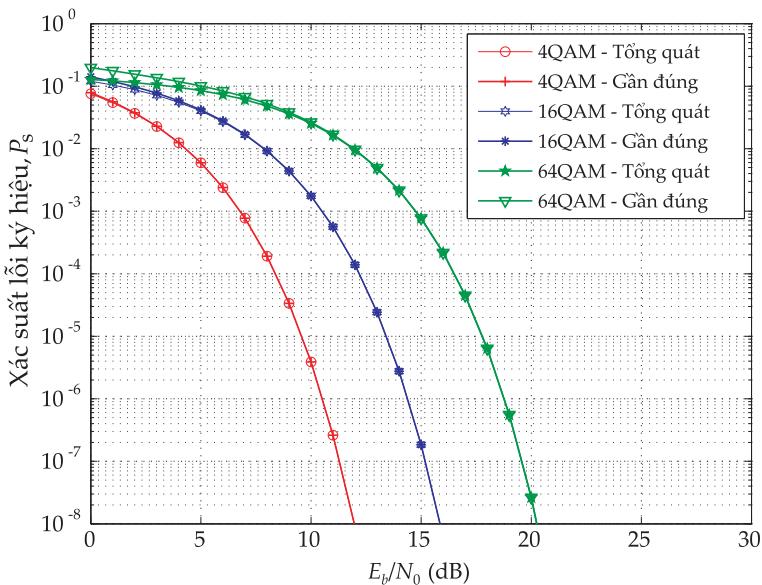
Trong vùng SNR đủ lớn, xác suất lỗi ký hiệu được tính theo công thức gần đúng sau:

$$P_s \approx 2P_{\sqrt{M}} = \frac{4(\sqrt{M} - 1)}{\sqrt{M}} Q\left(\sqrt{\frac{3E_{avg}}{(M-1)N_0}}\right). \quad (5.62)$$

Hình 5.15 trình bày các đường cong lỗi ký hiệu tính theo công thức tổng quát (5.60) và công thức xấp xỉ (5.62) cho các giá trị  $M = 4, 16$  và  $64$ .

### Xác suất lỗi bit trong kênh truyền AWGN

Tương tự như đối với các hệ thống MPSK, xác suất lỗi bit  $P_b$  của các hệ thống QAM có thể được tính gần đúng dựa trên đường bao



**Hình 5.15:** Xác suất lỗi kí hiệu của tín hiệu QAM trong kênh AWGN.

cận trên như trong (5.20). Tuy nhiên, dựa vào phương pháp đệ quy đề xuất bởi Yang và Hanzo [4], xác suất lỗi bit của tín hiệu QAM có thể được tính gần đúng bằng các công dưới đây. **Đối với hệ thống 4QAM, xác suất lỗi bit được tính chính xác bằng:**

$$P_b = \frac{1}{2} \operatorname{erfc} \left( \sqrt{\frac{E_b}{N_0}} \right). \quad (5.63)$$

Đối với hệ thống 16QAM, xác suất lỗi bit được tính gần đúng bằng [4]:

$$P_b \approx \frac{3}{8} \operatorname{erfc} \left( \sqrt{\frac{2}{5} \frac{E_b}{N_0}} \right) + \frac{1}{4} \operatorname{erfc} \left( \sqrt{\frac{18}{5} \frac{E_b}{N_0}} \right) \quad (5.64)$$

Đối với hệ thống 64QAM, xác suất lỗi bit được tính gần đúng bằng [4]:

$$P_b \approx \frac{7}{24} \operatorname{erfc} \left( \sqrt{\frac{1}{7} \frac{E_b}{N_0}} \right) + \frac{11}{48} \operatorname{erfc} \left( \sqrt{\frac{9}{7} \frac{E_b}{N_0}} \right). \quad (5.65)$$

Bạn đọc có thể tham khảo [4] để xây dựng các công thức tính xác suất lỗi bit cho các trường hợp  $M > 64$ .

### Xác suất lỗi ký hiệu trong kênh truyền pha-định Rayleigh

Áp dụng các biểu thức (5.24), (5.25) và (5.60), (5.61), ta có thể tính toán xác suất lỗi ký hiệu của tín hiệu QAM trong kênh pha-định Rayleigh như sau:

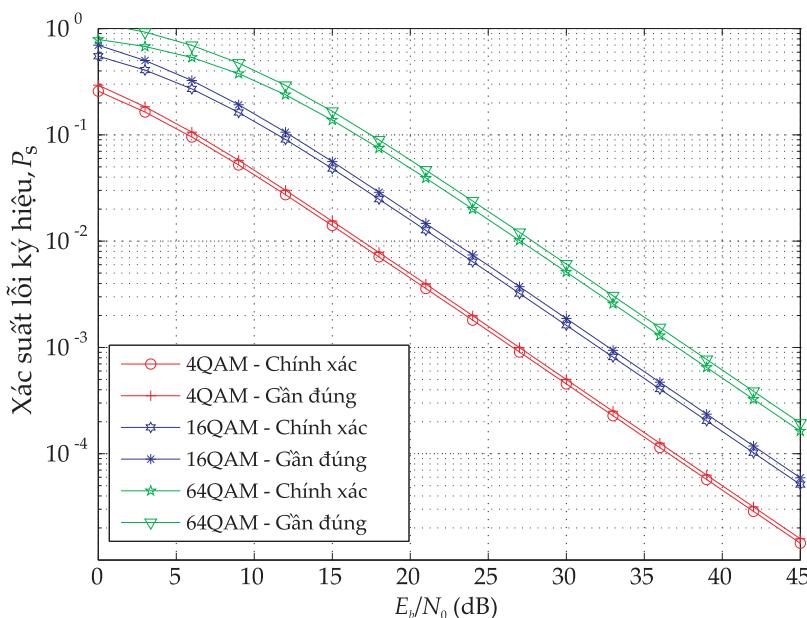
$$P_s = \int_0^{\infty} \left[ 2P_{\sqrt{M}}(\gamma_s) - P_{\sqrt{M}}^2(\gamma_s) \right] p(\gamma_s) d\gamma_s \quad (5.66)$$

trong đó

$$P_{\sqrt{M}}(\gamma_s) = \frac{2(\sqrt{M}-1)}{\sqrt{M}} Q\left(\sqrt{\frac{3\gamma_s}{(M-1)}}\right) \quad (5.67)$$

$$\gamma_s = \zeta^2 \frac{E_{avg}}{N_0} = \zeta^2 \frac{E_b \log_2 M}{N_0}$$

và  $p(\gamma_s)$  được định nghĩa như trong (5.25).



**Hình 5.16:** Xác suất lỗi ký hiệu của tín hiệu QAM trong kênh pha-định Rayleigh.

Do mức độ phức tạp của các hàm số dưới dấu tích phân, chúng ta không tính được biểu thức rút gọn của (5.66). Tuy nhiên, ta có thể sử dụng công thức gần đúng (5.62) để tính xấp xỉ  $P_s$  trong (5.66)

theo công thức sau cho các giá trị SNR đủ lớn:

$$P_s = \frac{2(\sqrt{M} - 1)}{\sqrt{M}} \left[ 1 - \frac{1}{\sqrt{1 + \frac{2(M-1)}{3\Gamma}}} \right]. \quad (5.68)$$

Sự sai lệch giữa công thức chính xác (5.66) và công thức gần đúng (5.68) được minh họa trên Hình 5.16 cho các trường hợp  $M = 4, 16$  và  $64$ . Chúng ta thấy rằng khi  $M$  càng tăng thì độ sai lệch càng lớn, đặc biệt ở vùng giá trị SNR thấp.

### Xác suất lỗi bit trong kênh truyền pha-định Rayleigh

Tương tự như trong kênh AWGN, xác suất lỗi bit của tín hiệu QAM có thể được tính gần đúng dựa trên công thức:

$$P_b \approx P_{bu}^r = \frac{P_s}{\log_2 M} \quad (5.69)$$

trong đó  $P_{bu}^r$  là đường bao cận trên của xác suất lỗi bit trong kênh Rayleigh.

Tuy nhiên, đối với tín hiệu 4QAM, xác suất lỗi bit trong kênh Rayleigh có thể được tính chính xác theo công thức:

$$P_b = \frac{1}{2} \left[ 1 - \frac{1}{\sqrt{1 + \left( \frac{E_b}{N_0} \right)^{-1}}} \right]. \quad (5.70)$$

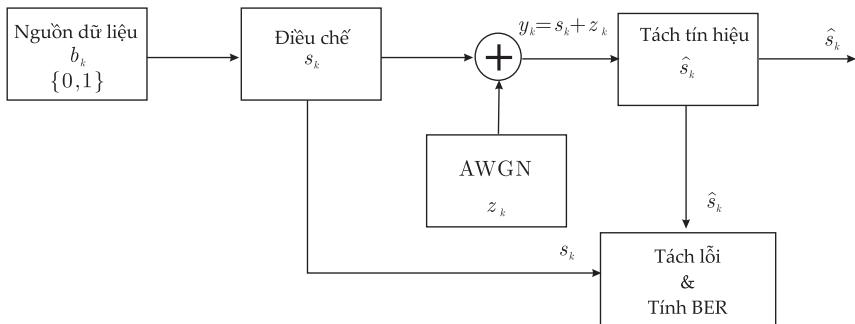
Một cách tương tự, ta cũng có thể dễ dàng đưa ra được các công thức tính gần đúng xác suất lỗi bit cho các trường hợp  $M = 16$  và  $M = 64$  tương ứng như sau:

$$P_b \approx \frac{3}{8} \left[ 1 - \frac{1}{\sqrt{1 + \left( \frac{2}{5} \frac{E_b}{N_0} \right)^{-1}}} \right] + \frac{1}{4} \left[ 1 - \frac{1}{\sqrt{1 + \left( \frac{18}{5} \frac{E_b}{N_0} \right)^{-1}}} \right] \quad (5.71)$$

$$P_b \approx \frac{7}{24} \left[ 1 - \frac{1}{\sqrt{1 + \left( \frac{1}{7} \frac{E_b}{N_0} \right)^{-1}}} \right] + \frac{11}{48} \left[ 1 - \frac{1}{\sqrt{1 + \left( \frac{9}{7} \frac{E_b}{N_0} \right)^{-1}}} \right]. \quad (5.72)$$

## 5.5 Mô phỏng truyền dẫn BPSK qua kênh AWGN

Hình vẽ 5.17 mô tả một sơ đồ mô phỏng đơn giản trên kênh AWGN sử dụng điều chế BPSK.



**Hình 5.17:** Sơ đồ mô phỏng truyền dẫn BPSK trên kênh AWGN.

Tại máy phát dữ liệu phát  $b_k \in \{0, 1\}$  được tạo ra từ một nguồn gián đoạn không nhớ (DMS:Discrete Memoryless Source). Trong Matlab chuỗi dữ liệu phát  $b_k$  có thể được tạo nhờ sử dụng hàm có sẵn `rand` hoặc `randn` như sau

$$\text{bk} = \text{rand}(1, N) > 0.5$$

hoặc

$$\text{bk} = \text{randn}(1, N) > 0$$

trong đó  $N$  là số mẫu cần tạo.

Chuỗi dữ liệu  $b_k$  sau đó được điều chế BPSK. Phép điều chế BPSK ở đây có thể coi tương đương với phép ánh xạ

$$s_k = \begin{cases} \sqrt{E_s} & \text{nếu } b_k = 0 \\ -\sqrt{E_s} & \text{nếu } b_k = 1 \end{cases} \quad (5.73)$$

tạo nên chuỗi ký hiệu phát  $s_k \in \{+\sqrt{E_s}, -\sqrt{E_s}\}$ . Trong trường hợp điều chế BPSK,  $E_s = E_b = 1$  nên  $s_k \in \{+1, -1\}$ . Do đó phép ánh xạ  $b_k \rightarrow s_k$  trong điều chế BPSK có được thực hiện bằng Matlab như

sau

$$sk = 1 - 2 * bk$$

Các dấu phát  $s_k$  truyền qua kênh truyền và chịu ảnh hưởng của tạp âm AWGN. Do ảnh hưởng của AWGN, tín hiệu thu  $y_k$  là xếp chồng (cộng) của các dấu phát  $s_k$  và các dấu tạp âm  $n_k$ , tức là

$$y_k = s_k + z_k \quad (5.74)$$

trong đó  $z_k$  là các dấu tạp âm AWGN phức có dạng

$$z_k = z_{I,k} + j z_{Q,k} \quad (5.75)$$

trong đó  $z_{I,k}$  và  $z_{Q,k}$  tương ứng là thành phần đồng pha và vuông pha của tạp âm. Do phương sai của các thành phần  $\sigma_{z_I}^2 = \sigma_{z_Q}^2 = \sigma_z^2 = N_0/2$ , trong đó  $N_0/2$  là mật độ phổ công suất tạp âm. Như vậy, phương sai của tạp âm  $n_k$  trở thành  $2\sigma_z^2 = N_0$ . Để tạo được tạp âm  $z_k$  với phương sai  $2\sigma_z^2$  chúng ta có thể sử dụng hàm `randn` có sẵn trong Matlab để tạo ra chuỗi các dấu tạp âm có phân bố chuẩn chính tắc, rồi nhân với độ lệch chuẩn của tạp âm  $\sigma_n$  như sau

$$zk = sigma * (randn(1, N) + j * randn(1, N_s)) \quad (5.76)$$

Để tạo ra tạp âm có năng lượng thỏa mãn tỉ số  $(E_b/N_0)_{req}$  cho trước chúng ta đặt độ lệch chuẩn

$$\sigma_z = \sqrt{(\sigma_z^2)} = \sqrt{\frac{E_b}{2(E_b/N_0)_{req}}}. \quad (5.77)$$

Tức là,

$$sigma = sqrt(Eb/(2 * EbNo))$$

$$zk = sigma * (randn(1, N) + j * randn(1, N_s))$$

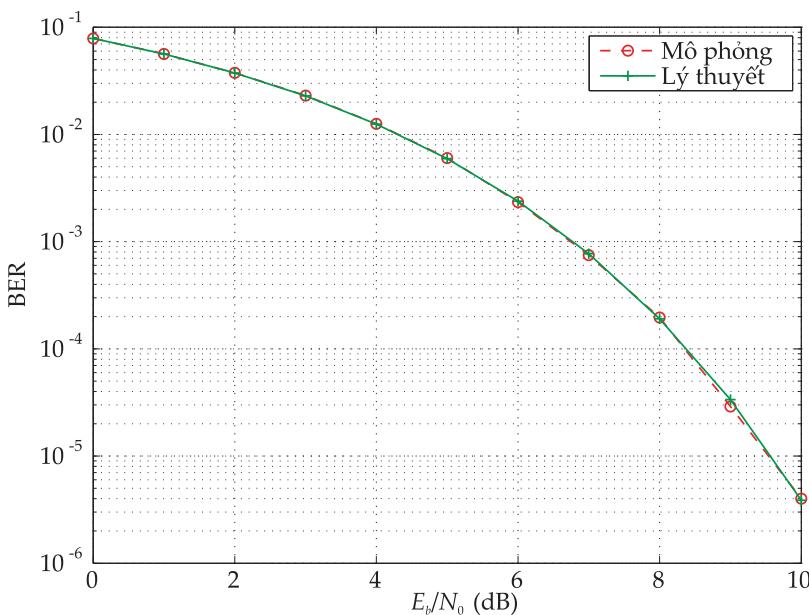
Tại máy thu, do tín hiệu điều chế BPSK chỉ chứa thành phần đồng pha (phần thực), nên để tách tín hiệu phát  $s_k$  từ tín hiệu thu được  $y_k$ , máy thu thực hiện tách lấy phần thực trước, sau đó thực hiện tách tín hiệu sử dụng phương pháp tách sóng hợp lõi tối ưu

(MLD: Maximum Likelihood Detection). Cụ thể, máy thu thực hiện phép so sánh và quyết định sau:

$$\text{if } \Re\{y_k\} \geq 0 \rightarrow \hat{s}_k = +1 \quad (5.78)$$

$$\text{elseif } \Re\{y_k\} < 0 \rightarrow \hat{s}_k = -1 \quad (5.79)$$

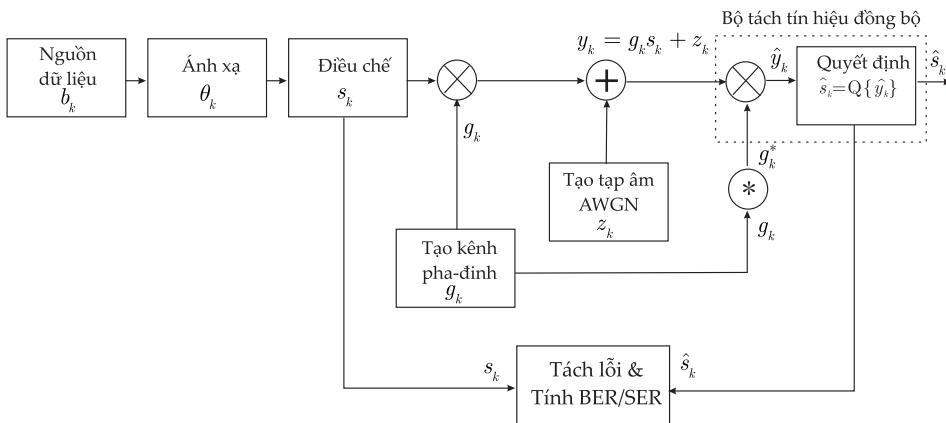
trong đó  $\Re\{\cdot\}$  biểu diễn phép toán lấy phần thực số phức. Phép so sánh này được thực hiện trong Matlab nhờ sử dụng hàm `sign(real(yk))`. Tín hiệu tách được  $\hat{s}_k$  sau đó được so sánh với tín hiệu phát  $s_k$  để tính toán phẩm chất lỗi bít BER của hệ thống.



Hình 5.18: Phẩm chất BPSK trên kênh AWGN.

Một ví dụ mẫu chương trình mô phỏng truyền BPSK qua kênh AWGN được trình bày ở chương trình MATLAB `BPSKAWGN.m` ở Matlab Program 5.1. Chương trình thực hiện mô phỏng với  $N = 10^5$  dấu BPSK  $\{+1, -1\}$ . Kết quả BER thu được từ chương trình mô phỏng `BPSKAWGN.m` được so sánh với giá trị lý thuyết trong (5.10) để xác định tính chính xác của kết quả mô phỏng (xem Hình vẽ 5.18).

## 5.6 Mô phỏng hệ thống truyền dẫn MPSK



**Hình 5.19:** Mô phỏng truyền MPSK qua kênh pha-định Rayleigh.

Sơ đồ mô phỏng Monte-Carlo của hệ thống truyền dẫn MPSK qua kênh pha-định Rayleigh được biểu diễn ở Hình 5.19. Trong trường hợp chúng ta muốn mô phỏng hệ thống trong kênh AWGN, ta chỉ việc gán hệ số kênh truyền  $g_k = 1$ .

Dựa kiến thức lý thuyết mà chúng ta đã nghiên cứu ở trên cũng như sơ đồ mô phỏng trên Hình 5.19, chúng ta có thể xây dựng thuật toán mô phỏng hệ thống truyền dẫn MPSK qua kênh pha-định Rayleigh như sau:

1. *Tạo dữ liệu và Điều chế:* thực hiện điều chế MPSK bằng cách nhóm  $\kappa = \log_2 M$  bit dữ liệu nhị phân và ánh xạ thành một điểm tín hiệu trong chòm sao tín hiệu. Phép ánh xạ có thể theo phương pháp mã hóa Gray như trên Hình 5.7 hoặc không. Dựa trên phương pháp gán nhãn các điểm tín hiệu từ 0 đến  $M - 1$  như ở hình vẽ, chúng ta thấy rằng điểm tín hiệu thứ  $i$  có thể được biểu diễn bởi

$$s_i = A \exp \left( j \left[ \frac{2\pi}{M} i + \frac{\pi}{M} \right] \right). \quad (5.80)$$

Trong đó  $A = \sqrt{E_s}$  là biên độ tín hiệu và  $\frac{\pi}{M}$  là pha ban đầu của sơ đồ tín hiệu. Như vậy, để tạo ra chuỗi các dấu điều chế

MPSK  $s_k$ , chúng ta có thể tạo ra các số nguyên ngẫu nhiên  $b_k \in \{0, 1, 2, \dots, M - 1\}$ , rồi thay  $i = b_k$  ở công thức (5.80). Như vậy, toàn quá trình tạo dữ liệu, ánh xạ, và điều chế có thể được thực hiện bằng Matlab như sau

```
b = randint(1, sigLen, [0 : M - 1]);
Const = pskmod([0 : M - 1], M, pi/M);
Sig = Const(b + 1);
```

Trên đây, chúng ta sử dụng hàm `pskmod()` của Matlab để tạo ra chòm sao tín hiệu MPSK, sau đó chúng ta ánh xạ dữ liệu trong `b` vào chòm sao tín hiệu `Const` để tạo ra tín hiệu phát `Sig`. Ta có thể sử dụng hàm `PSKGrayCoded()` (như minh họa trong chương trình mô phỏng dưới đây) để tạo ra chòm sao tín hiệu đã được mã hóa Gray `ConstGray`, sau đó thực hiện ánh xạ dữ liệu trong `b` vào chòm sao tín hiệu `ConstGray` để tạo ra tín hiệu phát `SigGray`.

2. *Tạo kênh pha-định:* kênh pha-định  $g_k$  được tạo bởi thuật toán tạo pha-định Rayleigh trình bày ở chương trước. Tuy nhiên, đối với kênh pha-định Rayleigh phẳng, biến đổi chậm, chúng ta có thể tạo ra pha-định Rayleigh trong Matlab như sau

```
g = 1/sqrt(2) * (randn(1, sigLen) + j * randn(1, sigLen));
```

3. *Tạo tạp âm AWGN  $z_k$ :* tạp âm  $z_k$  được tạo ra sử dụng phương pháp mô tả ở Mục 5.1 với  $\sigma_z^2 = \frac{E_s}{2E_s/N_0}$ . Với tín hiệu MPSK một dấu  $s_k$  chứa  $\kappa = \log_2 M$  bit dữ liệu nhị phân, vì vậy, quan hệ năng lượng bit và dấu được biểu diễn bởi  $E_s = \kappa E_b = E_b \log_2 M$
4. *Tách tín hiệu đồng bộ (coherent detection):* nguyên lý tách coherent detection là sử dụng liên hợp phức của ước lượng kênh truyền  $\hat{g}_k$  (ước lượng được nhờ các bộ ước lượng kênh) nhân với tín hiệu thu  $y_k$  để quay pha tín hiệu, bị dịch đi do pha-định, về vị trí ban đầu, tức là,  $\hat{y}_k = \hat{g}_k^* y_k$ . Hoặc ta có thể sử

dụng phương pháp ZF equalization để quay pha tín hiệu, tức là  $\hat{y}_k = y_k / \hat{g}_k$ . Để thuận tiện cho mô phỏng chúng ta có thể coi  $\hat{g}_k$  được ước lượng một cách chính xác, tức là,  $\hat{g}_k = g_k$ , và sử dụng ngay  $g_k$  cho tách tín hiệu đồng bộ. Mặc dù bằng cách này ảnh hưởng quay pha do pha-định đã được khắc phục, nhưng góc pha tín hiệu thu vẫn không trùng với góc pha của tín hiệu phát do còn chịu ảnh hưởng của tạp âm. Sử dụng phương pháp tách tín hiệu hợp lẽ tối đa (MLD), bộ tách tín hiệu thực hiện quyết định dựa trên góc pha của  $\hat{y}_k$ . Tức là,

$$\hat{\theta}_k = \angle \hat{y}_k \quad (5.81)$$

trong đó  $\angle$  biểu diễn phép tính lấy góc pha, và  $\hat{\theta}_k \in [0, 2\pi]$ . Trong Matlab phép tính lấy góc này có thể thực hiện được nhờ hàm có sẵn `angle`. Tuy nhiên, do hàm `angle` của Matlab cho ta các giá trị  $\hat{\theta}_k \in [0, \pm\pi]$ . Vì vậy, để biểu diễn góc  $\hat{\theta}_k \in [0, 2\pi)$  bằng hàm `angle`, chúng ta cần thực hiện chuyển đổi sử dụng hàm `mod` như sau

$$\text{thetak\_hat} = \text{mod}((\text{angle}(\text{yk\_hat}) + 2 * \pi), 2 * \pi) \quad (5.82)$$

Từ góc pha  $\hat{\theta}_k$  này chuỗi dữ liệu phát ban đầu  $b_k$  bằng thao tác ánh xạ ngược

$$\hat{b}_k = \left\lfloor \frac{M}{2\pi} \hat{\theta}_k \right\rfloor \quad (5.83)$$

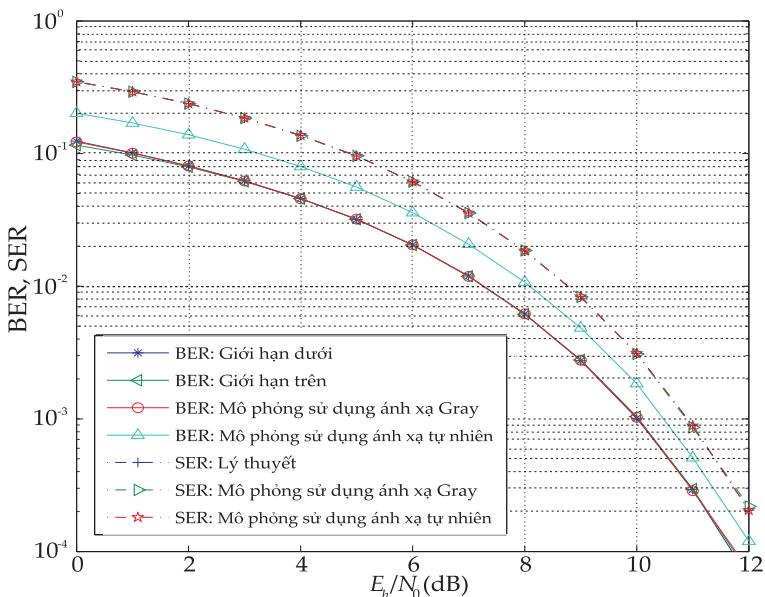
trong đó  $\lfloor \cdot \rfloor$  biểu diễn phép tính làm tròn về số nguyên gần nhất về phía 0, tức là phép tính lấy `floor`. Trong Matlab phép tính  $\lfloor \cdot \rfloor$  được thực hiện bằng hàm có sẵn `floor`.

5. *Tính toán tỉ số lỗi ký hiệu SER và tỉ số lỗi bit BER:* Để tính toán các tỷ số lỗi bit và lỗi ký hiệu chúng ta sử dụng các hàm `biterr.m` và `symerr.m` của Matlab như sau:

$$[\text{ErrNum}, \text{BER}] = \text{biterr}(\text{bk}, \text{bk\_hat}, \kappa)$$

$$[\text{ErrNum}, \text{SER}] = \text{symerr}(\text{bk}, \text{bk\_hat})$$

với  $\kappa = \log_2 M$ .

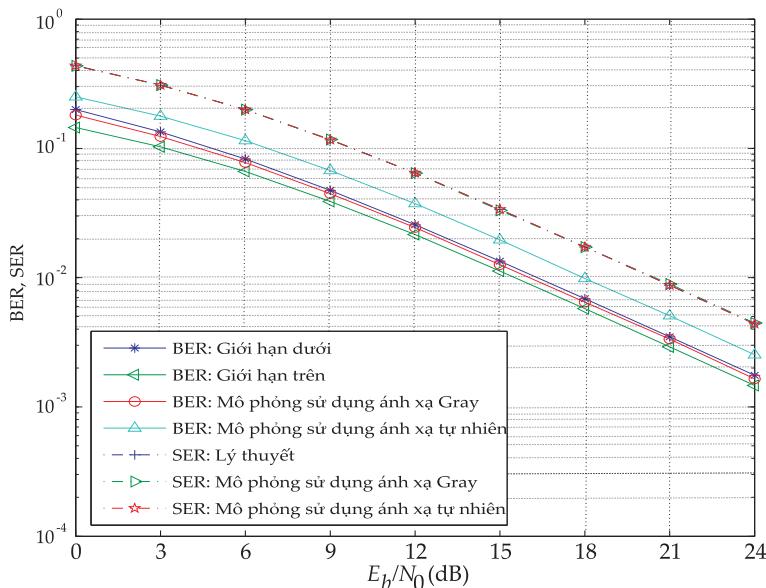


Hình 5.20: Phẩm chất lỗi của hệ thống 8PSK trên kênh AWGN.

Chương trình mô phỏng mẫu truyền tín hiệu MPSK qua kênh AWGN và pha-định Rayleigh được trình bày ở chương trình MATLAB pskSimulation.m ở Matlab Program 5.2. Kết quả BER và SER thu được từ chương trình mô phỏng pskSimulation.m được so sánh với các giá trị SER và BER lý thuyết trong (5.18) và (5.23) cho kênh AWGN, và trong (5.27) và (5.33) cho kênh pha-định Rayleigh để xác định tính chính xác của kết quả mô phỏng (xem các Hình vẽ 5.20 và 5.21). Từ các hình vẽ này chúng ta có thể thấy được sự chính xác giữa kết quả lý thuyết và kết quả mô phỏng. Ta cũng thấy rõ ràng phép ánh xạ Gray cho phép nâng cao phẩm chất lỗi bit của hệ thống.

### 5.7 Mô phỏng truyền dẫn QAM

Trong mục trước chúng ta đã nghiên cứu hệ thống truyền dẫn tín hiệu MPSK qua kênh pha-định. Chúng ta cũng đã nghiên cứu phương pháp điều chế và giải điều chế tín hiệu MPSK. Trong phần này, chúng ta mở rộng thuật toán mô phỏng truyền dẫn MPSK cho trường hợp điều chế QAM. Trong chương trình mô phỏng, chúng ta sẽ tận dụng hàm điều chế qammod.m và giải điều chế qamdemod.m có sẵn trong Matlab. Thuật toán mô phỏng QAM được trình bày dưới đây.



**Hình 5.21:** Phẩm chất lỗi của hệ thống 8PSK trên kênh pha-định Rayleigh.

1. *Tạo dữ liệu và điều chế:* Tương tự như trường hợp điều chế MPSK, chúng ta thực hiện điều chế QAM bằng cách nhóm  $\kappa = \log_2 M$  bit dữ liệu nhị phân và ánh xạ thành một điểm tín hiệu trong chòm sao tín hiệu QAM. Phép ánh xạ có thể theo phương pháp mã hóa Gray như trên Hình 5.14 hoặc không. Dữ liệu phát đi có thể được tạo ra từ một nguồn các số nguyên ngẫu nhiên nằm trong giải từ 0 đến  $M - 1$ , trong đó  $M$  là bậc điều chế. Trong Matlab, nguồn số nguyên này được tạo ra bằng hàm có sẵn `randint.m`. Cụ thể

$$\text{bk} = \text{randint}(1, N, [0, M - 1])$$

Để điều chế, chúng ta có thể sử dụng trực tiếp hàm điều chế QAM có sẵn trong Communications Toolbox của Matlab. Ví dụ

$$\text{sk} = \text{qammod}(\text{bk}, M).$$

Sau đó, chúng ta có thể thực hiện tạo ra một chòm sao tín hiệu

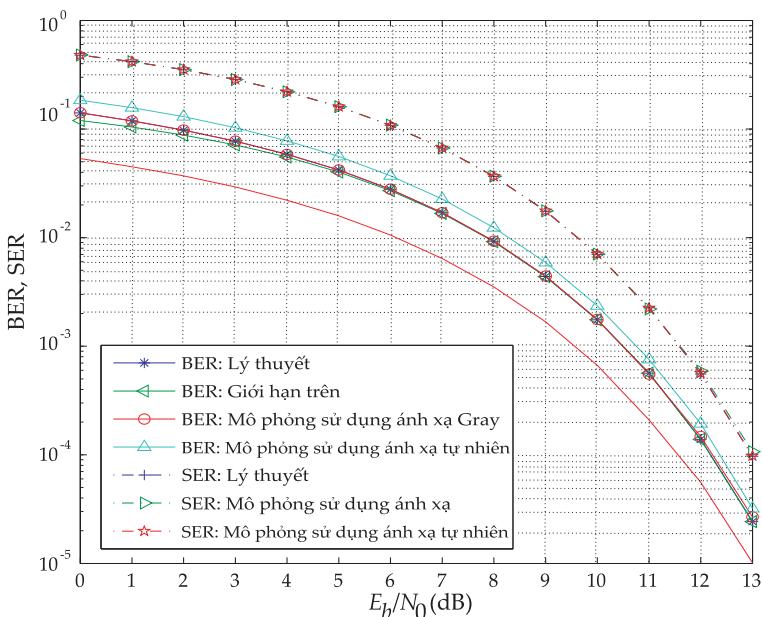
mã hóa Gray bằng cách sử dụng chương trình qamGrayCoded.m:

$$[\text{ConstOut}, \text{Pos}] = \text{qamGrayCoded}(\text{Const}, M)$$

Chú ý, khi  $M = 4$ , hàm qammod.m của Matlab sẽ tạo ra chòm sao tín hiệu đã được mã hóa Gray. Trong trường hợp này, hàm qamGrayCoded.m đóng vai trò tạo ra chòm sao tín hiệu không được mã hóa Gray. Vì vậy để đánh giá đúng BER của các hệ thống 4QAM có sử dụng và không sử dụng mã hóa Gray, ở cuối chương trình mô phỏng chúng ta thực hiện tráo đổi các giá trị BER như sau:

```
if M == 4
    temp = NBER; NBER = GBER; GBER = temp;
end
```

2. *Tạo kênh pha-định:* Kênh pha-định được tạo ra tương tự như trong chương trình mô phỏng hệ thống truyền dẫn MPSK trình bày ở trên.
3. *Tạo tạp âm AWGN  $z_k$ :* tạp âm  $z_k$  được tạo ra sử dụng phương pháp mô tả ở các chương trình mô phỏng BPSK và MPSK với  $\sigma_z^2 = \frac{E_s}{2E_s/N_0} = \frac{E_s}{2\kappa E_b/N_0}$ . Đối với tín hiệu QAM xét ở trên, tín hiệu có biên độ nhỏ nhất sẽ là các điểm tín hiệu với các giá trị  $(I_i, Q_i) = (\pm 1, \pm 1)$ . Từ đó, ta suy ra năng lượng của tín hiệu có biên độ nhỏ nhất là  $E_0 = 2$ , dẫn tới  $E_s = E_{avg} = \frac{2}{3}(M - 1)$ .
4. *Tách tín hiệu đồng bộ (coherent detection):* Tín hiệu QAM là tín hiệu được điều chế cả biên độ và góc pha. Để khôi phục lại tín hiệu ban đầu, chúng ta cần loại bỏ ảnh hưởng của pha-định tới cả biên độ và góc pha của tín hiệu thu được. Vì vậy, nguyên lý tách tín hiệu QAM đồng bộ bao gồm 2 bước: (1) sử dụng liên hợp phức của ước lượng kênh truyền  $\hat{g}_k$  (ước lượng được nhờ các bộ ước lượng kênh) nhân với tín hiệu thu  $y_k$  để quay pha tín hiệu, bị dịch đi do pha-định, về vị trí ban đầu, tức là,



**Hình 5.22:** Phẩm chất lỗi của hệ thống 16QAM trên kênh AWGN.

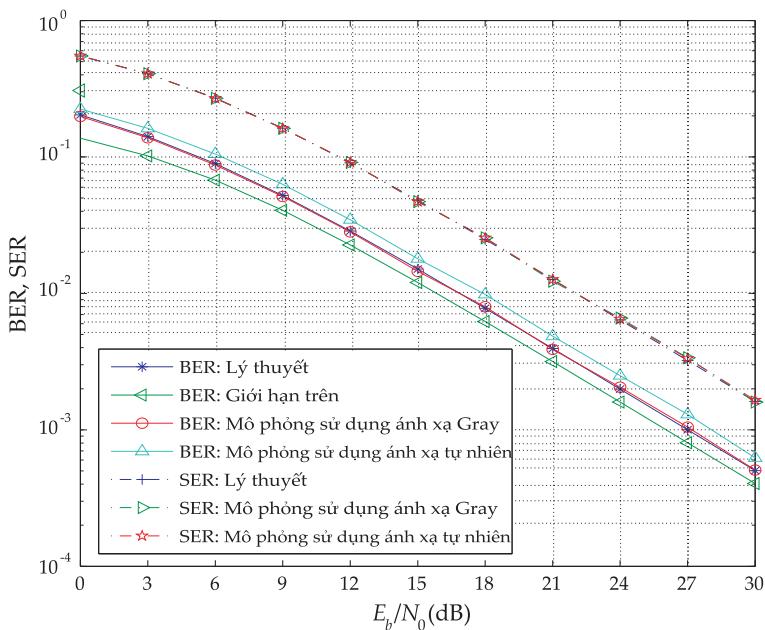
$\hat{g}_k^* y_k$ ; (2) chia kết quả thu được sau khi quay pha ở bước (1) cho  $|\hat{g}_k|^2$  để loại trừ ảnh hưởng của pha-định tới biên độ tín hiệu QAM. Kết quả thu được sau hai bước:  $\hat{y}_k = \hat{g}_k^* y_k / |\hat{g}_k|^2$ . Để thuận tiện cho mô phỏng chúng ta có thể coi  $\hat{g}_k$  được ước lượng một cách chính xác, tức là,  $\hat{g}_k = g_k$ , và sử dụng ngay  $g_k$  cho tách tín hiệu đồng bộ. Cuối cùng, ta đưa  $\hat{y}_k$  vào hàm qamdemod.m để khôi phục lại tín hiệu phát.

5. *Tính toán tỉ số lỗi ký hiệu SER và tỉ số lỗi bit BER:* Để tính toán các tỷ số lỗi bit và lỗi ký hiệu chúng ta sử dụng các hàm biterr.m và symerr.m của Matlab như sau:

$$\begin{aligned} [\text{ErrNum}, \text{BER}] &= \text{biterr}(\text{bk}, \text{bk\_hat}, \text{kappa}) \\ [\text{ErrNum}, \text{SER}] &= \text{symerr}(\text{bk}, \text{bk\_hat}) \end{aligned}$$

với  $\text{kappa} = \kappa = \log_2 M$ .

Chương trình mô phỏng mẫu truyền tín hiệu QAM qua kênh AWGN và pha-định Rayleigh được trình bày ở chương trình MATLAB qamSimulation.m ở Matlab Program 5.3. Kết quả BER và SER



**Hình 5.23:** Phẩm chất lỗi của hệ thống 16QAM trên kênh pha-định Rayleigh.

thu được từ chương trình mô phỏng `qamSimulation.m` được so sánh với các giá trị SER và BER lý thuyết trong (5.66) và (5.63)–(5.65) cho kênh AWGN, và trong (5.66) và (5.70)–(5.72) cho kênh pha-định Rayleigh để xác định tính chính xác của kết quả mô phỏng (xem các Hình vẽ 5.22 và 5.23). Từ các hình vẽ này chúng ta có thể thấy được sự chính xác giữa kết quả lý thuyết và kết quả mô phỏng. Ta cũng thấy rõ rằng phép ánh xạ Gray cho phép nâng cao phẩm chất lỗi bit của hệ thống.

## 5.8 Tóm tắt

Trong chương này tác giả đã giới thiệu các kỹ thuật điều chế số sử dụng trong các hệ thống thông tin số. Nội dung chương tập trung vào nguyên lý hoạt động, phương pháp tính tỉ số lỗi BER và SER. Phương pháp mô phỏng và các chương trình mẫu minh họa cũng được trình bày chi tiết làm cơ sở cho người đọc nắm bắt tốt hơn.

## Tài liệu tham khảo

- [1] F. Xiong, *Digital Modulation Techniques*. Artech House, 2000.
- [2] H. Harada and R. Prasad, *Simulation and Software Radio for Mobile Communications*. Artech House, 2002.
- [3] P. J. Lee, "Bit error rate of coherent M-ary PSK," *TDA Progress Report 42-81*, pp. 31–37, Jan.– Mar. 1985.
- [4] L. L. Yang and L. Hanzo, "A recursive algorithm for the error probability evaluation of M-QAM," *IEEE Trans. Commun. Letter*, vol. 4, pp. 304–306, Oct. 2000.

## Phụ lục: Chương trình MATLAB mẫu

---

### MATLAB Program 5.1 BPSKAWGN.m

```
% Chuong trinh mo phong truyen dan BPSK qua kenh AWGN
% Dinh nghia tham so
EbNodB=0:10;
EbNo=10.^{(EbNodB./10)};
sigLen=10^6;
% Tao tin hieu BPSK {+1,-1}
s=1-2*(randn(1,sigLen)>=0.5);
% Tinh toan nang luong bit tin hieu Eb
Eb=norm(s)^2/sigLen;
% Mat do pho AWGN
No=Eb./EbNo;
% Vong lap tinh toan BER theo Eb/No
for k=1:length(EbNo)
    % Tao AWGN
    n=sqrt(No(k)./2).* (randn(1,sigLen)+j*randn(1,sigLen));
    % Tin hieu thu
    y=s+n;
    % Tach tin hieu
    shat=sign(real(y));
    error=s-shat;
```

```

noError=length(find(error~=0));
BER(k)=noError/sigLen;
end

% BER ly thuyet cua truyen dan BPSK qua kenh AWGN
BERtheory=1/2*erfc(sqrt(EbNo));
% Ve do thi
semilogy(EbNodB,BER,'*',EbNodB,BERtheory)
 xlabel('Eb/No')
 ylabel('BER')
 legend('By simulation','By theory')
 title('BER cua BPSK qua kenh AWGN')
grid

```

### **■ MATLAB Program 5.2 pskSimulation.m**

```

% Chuong trinh mo phong Monte-Carlo truyen dan MPSK qua kenh AWGN
% va pha dinh Rayleigh
% Written by: Le Minh Tuan, UVISION JSC.
% Tran Xuan Nam, Le Qui Don Technical University

% Lats updated 24/06/2011
%-----

clc;
clear all

M=input('Nhap vao cap dieu che PSK: ')
while ((M~=4)&&(M~=8)&&(M~=16))
    M=input('Nhap vao cap dieu che PSK (M=4, 8, 16): ')
end

Ctype=input('Nhap vao loai kenh truyen: ','s')
while ((Ctype~= 'G')&&(Ctype~= 'g')&&(Ctype~= 'R')&&(Ctype~= 'r'))
    Ctype=input('Chon loai kenh truyen (G: AWGN, R: Rayleigh): ','s')
end

```

```
switch M
    case 4
        if(Ctype=='G' || Ctype=='g')
            EbNodB=0:1:10;
        else
            EbNodB=0:3:24;
        end
    case 8
        if(Ctype=='G' || Ctype=='g')
            EbNodB=0:1:12;
        else
            EbNodB=0:3:24;
        end
    case 16
        if(Ctype=='G' || Ctype=='g')
            EbNodB=0:2:16;
        else
            EbNodB=0:3:30;
        end
    otherwise
        break;
    end
% Dinh nghia tham so
m=log2(M);      % So bit/ky hieu
EbNo=10.^ (EbNodB./10);
Es=1;           % Nang luong ky hieu

if(Ctype=='G' || Ctype=='g')
    sigLen=10^6;    % So ky hieu gui di
    g=1;
    [BER] = pskBERCalAWGN(M, EbNo);%BER theo ly thuyet
    [SER] = pskSERCalAWGN(M, EbNo);%SER theo ly thuyet
else
    sigLen=2*10^5;    % So ky hieu gui di
    % Tao pha dinh Rayleigh
```

```
g=1/sqrt(2)*(randn(1,sigLen)+j*randn(1,sigLen));
[BER] = pskBERCalRay(M, EbNo);%BER theo ly thuyet
[SER] = pskSERCalRay(M, EbNo);%SER theo ly thuyet
end

%PSK Constellation
Const=pskmod([0:M-1],M,pi/M);

%Gray-Coded PSK Constellations
[ConstGray, Pos]=pskGrayCoded(Const,M);

% Tao chuoi bk chua cac so integer tu 0 den M-1 ngau nhien va
% anh xa vao chom sao tin hieu
b=randint(1,sigLen,[0 M-1]);      % Random intergers from 0:M-1

%Tin hieu khong duoc ma hoa Gray
Sig=Const(b+1);

%Tin hieu duoc ma hoa Gray
SigGray=ConstGray(b+1);

% Mat do pho AWGN
sigma=sqrt(Es./(2*m.*EbNo));

% Vong lap tinh toan BER theo Eb/No
for k=1:length(EbNo)
    % Tao AWGN voi variance sigma^2
    n=sigma(k)*(randn(1,sigLen)+j*randn(1,sigLen));

    % Tin hieu thu khong ma hoa Gray
    yk=Sig.*g+n;

    % Tin hieu thu co ma hoa Gray
    ykGray=SigGray.*g+n;
```

```
%Tach song tin hieu khong ma hoa Gray
[bHat]=pskDecision(yk,g,M);

%Tach song tin hieu khong ma hoa Gray
[bkHat]=pskDecision(ykGray,g,M);
bHatGray=Pos(bkHat+1);

% Tinh toan SER va BER
[noBitErr,NBER(k)]=biterr(bHat,b,m);    % BER
[noSymErr,NSER(k)]=symerr(bHat,b);    % SER
[noBitErr,GBER(k)]=biterr(bHatGray,b,m);% Gray-mapping BER
[noSymErr,GSER(k)]=symerr(bHatGray,b); %Gray-mapping SER

end

% Ve do thi
semilogy(EbNodB,BER,'-*',EbNodB,SER./m,'-<', EbNodB,GBER,'-o',...
EbNodB,NBER,'-^')

hold on
semilogy(EbNodB,SER,'-.+',EbNodB,GSER,'-.>',EbNodB,NSER,'-.p')
xlabel('E_b/N_o (dB)')
ylabel('BER, SER')

txt=['BER: Lower Bound          ';
      'BER: Upper Bound          ';
      'BER: Simulation with Gray mapping';
      'BER: Simulation w/o Gray mapping ';
      'SER: Theory                ';
      'SER: Simulation with Gray mapping';
      'SER: Simulation w/o Gray mapping '];
legend(txt)


```

### Function 5.1 pskBERCalAWGN.m

```
% Ham tinh BER cua he thong MPSK trong kenh AWGN
% Written by: Le Minh Tuan, UVISION JSC.
%-----
```

```
% Doi voi QPSK, cong thuc tinh BER la chinh xac
% Doi voi MPSK, cong thuc tinh BER la duong bao can duoi

% Lats updated 24/06/2011
```

```
function [BER] = pskBERCalAWGN(M, EbNo)
%M: Cap do dieu che
%EbNo: Ty so SNR
%BER: Xac suat loi bit

switch M
    case 4
        BER = erfc(sqrt(EbNo))./2;
    otherwise
        k=log2(M);
        a=pi/M;
        d=sqrt(k*EbNo);
        sa=sin(a);
        s3a=sin(3*a);
        BER=1/k*(erfc(d*sa)+erfc(d*s3a));
    end
end
```

### Function 5.2 pskSERCalAWGN.m

```
% Ham tinh SER cua he thong MPSK trong kenh AWGN
% Written by: Le Minh Tuan, UVISION JSC.
%-----
% Cong thuc tinh SER la chinh xac cho cac gia tri M
```

```
% Lats updated 24/06/2011
```

```
function [SER] = pskSERCalAWGN(M, EbNo)
%M: Cap do dieu che
%EbNo: Ty so SNR
%SER: Xac suat loi ky hieu

switch M
```

```

case 4
    temp=erfc(sqrt(EbNo));
    SER=temp-temp.^2./4;
otherwise
    EsNo_eq = sqrt(log2(M)*EbNo)*sin(pi/M);
    F=@(y)exp(-y.*y).*erf(y*cot(pi/M));
    for k=1:length(EbNo)
        Ps(k) = quad(F,0,EsNo_eq(k),1.0e-12);
    end
    SER=(M-1)/M - erf(EsNo_eq)/2-(1/sqrt(pi))*Ps;
end
end

```

---

### **Function 5.3 pskBERCalRay.m**

```

% Ham tinh BER cua he thong MPSK trong kenh pha dinh Rayleigh
% Written by: Le Minh Tuan, UVISION JSC.
%-----
% Doi voi QPSK, cong thuc tinh BER la chinh xac
% Doi voi MPSK, cong thuc tinh BER la duong bao can duoi (lower bound)
% Last updated 24/06/2011

```

```

function [BER] = pskBERCalRay(M, EbNo)
%M: Cap do dieu che
%EbNo: Ty so SNR
%BER: Xac suat loi bit
switch M
case 4
    BER=(1-1./sqrt(1+1./EbNo))./2;
otherwise
    k=log2(M);
    a=pi/M;
    sa=sin(a);
    sa=sa*sa;
    s3a=sin(3*a);
    s3a=s3a*s3a;

```

```

EbNoEqv1=EbNo.*k.*sa;
EbNoEqv2=EbNo.*k.*s3a;
BER = [2-1./sqrt(1+1./EbNoEqv1)-1./sqrt(1+1./EbNoEqv2)]./k;
end
end

```

---

**■ Function 5.4 pskSERCalRay.m**

```

% Ham tinh SER cua he thong MPSK trong kenh pha dinh Rayleigh
% Written by: Le Minh Tuan, UVISION JSC.
%-----
% Cong thuc tinh SER la chinh xac cho cac gia tri M

% Lats updated 24/06/2011

```

```

function [SER] = pskSERCalRay(M, EbNo)
%M: Cap do dieu che
%EbNo: Ty so SNR
%SER: Xac suat loi ky hieu

n=log2(M);
pM=pi/M;
sM=sin(pM)*sin(pM);
cy=@(y)cos(y);
sx=@(x)sqrt(x);
for k=1:length(EbNo)
    Gam=n*EbNo(k);
    F=@(x,y)(sx(x).*exp(-x.*(1-cy(y).*cy(y)+1/Gam)).*cy(y)...
        .*(1+erf(sx(x).*cy(y))))./(2*Gam*sqrt(pi));
    Ps_ray(k) = dblquad(F,0,1.0e+2,-pM,pM,1.0e-10)...
        +dblquad(F,1.0e+2,1.0e+6,-pM,pM,1.0e-10)...
        +dblquad(F,1.0e+6,1.0e+8,-pM,pM,1.0e-10);
    Ps_ray(k)=1-1/(M*(1+Gam))-Ps_ray(k);
end
SER = Ps_ray;

```

---

**Function 5.5 pskGrayCoded.m**

```
% Ham anh xa bit du lieu vao cac ky hieu MPSK su dung ma hoa Gray
% Written by: Le Minh Tuan, UVISION JSC.
%-----
```

```
% Lats updated 24/06/2011
```

```
function [ConstOut, Pos]=pskGrayCoded(Const,M)
%Const: Chom sao tin hieu khong duoc ma hoa Gray
%M: Cap do dieu che PSK
%ConstOut: Chom sao tin hieu da duoc ma hoa Gray
%Pos: Du lieu anh xa theo ma hoa Gray

switch M
    case 4
        Pos=[0 1 3 2];
        [a b]=sort(Pos);
        ConstOut=Const(b);
    case 8
        Pos=[0 1 3 2 6 7 5 4];
        [a b]=sort(Pos);
        ConstOut=Const(b);
    case 16
        Pos=[0 1 3 2 6 7 5 4 12 13 15 14 10 11 9 8];
        [a b]=sort(Pos);
        ConstOut=Const(b);
    otherwise
        Pos=[];
        ConstelOut=[];
    end
end
```

**Function 5.6 pskDecision.m**

```
% Ham tach song ML tin hieu MPSK
% Written by: Le Minh Tuan, UVISION JSC.
```

```
%-----
```

```
% Lats updated 24/06/2011
```

```
function [Out]=pskDecision(y,g,M)
%y: tin hieu thu duoc
%g: channel gain
%M: Cap do dieu che PSK
%Out: Du lieu sau tach song

%Coherent detection with ZF
ykHat=y./g;

% Decision
thetak=angle(ykHat);
thetak=mod(thetak+2*pi,2*pi); % chuyen tu [-pi,pi] --> [0,2*pi)
Out=floor(M/(2*pi)*thetak);

end
```

### MATLAB Program 5.3 qamSimulation.m

```
% Chuong trinh mo phong Monte-Carlo truyen dan QAM qua kenh AWGN
% va pha dinh Rayleigh
% Written by: Le Minh Tuan, UVISION JSC.
% Tran Xuan Nam, Le Qui Don Technical University
```

```
% Lats updated 24/06/2011
```

```
%-----
```

```
clc;
clear all
M=input('Nhap vao cap dieu che QAM: ')
while ((M~=4)&&(M~=16)&&(M~=64))
    M=input('Nhap vao cap dieu che PSK (M=4, 16, 64): ')
end
```

```
Ctype=input('Nhập vào loại kênh truyền: ','s')
while ((Ctype=='G')||(Ctype=='g')||(Ctype=='R')||(Ctype=='r'))
    Ctype=input('Nhập vào loại kênh truyền (G: AWGN, R: Rayleigh): ','s')
end

switch M
    case 4
        if(Ctype=='G' || Ctype=='g')
            EbNodB=0:1:10;
            sigLen=10^6;      % Số ký hiệu gửi đi
        else
            EbNodB=0:3:24;
            sigLen=2*10^5;    % Số ký hiệu gửi đi
        end
    case 16
        if(Ctype=='G' || Ctype=='g')
            EbNodB=0:1:13;
            sigLen=10^6;      % Số ký hiệu gửi đi
        else
            EbNodB=0:3:30;
            sigLen=10^5;      % Số ký hiệu gửi đi
        end
    case 64
        if(Ctype=='G' || Ctype=='g')
            EbNodB=0:2:18;
            sigLen=2*10^5;    % Số ký hiệu gửi đi
        else
            sigLen=2*10^5;    % Số ký hiệu gửi đi
            EbNodB=0:3:36;
        end
    otherwise
        break;
end

% Định nghĩa tham số
```

```

m=log2(M);           % So bit/ky hieu
EbNo=10.^EbNodB./10;
Es = 2*(M-1)/3; %Nang luong ky hieu

if(Ctype=='G' || Ctype=='g')
    g=1;
    [BER] = qamBERCalAWGN(M, EbNo);%BER theo ly thuyet
    [SER] = qamSERCalAWGN(M, EbNo);%SER theo ly thuyet
    EqCoef=1;
else
    % Tao pha dinh Rayleigh
    g=1/sqrt(2)*(randn(1,sigLen)+j*randn(1,sigLen));
    EqCoef=conj(g)./abs(g).^2; %One tap equalization
    [BER] = qamBERCalRay(M, EbNo);%BER theo ly thuyet
    [SER] = qamSERCalRay(M, EbNo);%SER theo ly thuyet
end

%QAM Constellation
Const=qammod([0:M-1],M);

%Gray-Coded QAM Constellations
[ConstGray, Pos]=qamGrayCoded(Const,M);

% Tao chuoi bk chua cac so integer tu 0 den M-1 ngau nhien va
% anh xa vao chom sao tin hieu
b=randint(1,sigLen,[0 M-1]);      % Random intergers from 0:M-1

%Tin hieu khong duoc ma hoa Gray
Sig=Const(b+1);

%Tin hieu duoc ma hoa Gray
SigGray=ConstGray(b+1);

% Mat do pho AWGN
sigma=sqrt(Es./(2*m.*EbNo));

```

```
% Vong lap tinh toan BER theo Eb/No
for k=1:length(EbNo)
    % Tao AWGN voi variance sigma^2
    n=sigma(k)*(randn(1,sigLen)+j*randn(1,sigLen));

    % Tin hieu thu khong ma hoa Gray
    yk=Sig.*g+n;

    % Tin hieu thu co ma hoa Gray
    ykGray=SigGray.*g+n;

    %Tach song tin hieu khong ma hoa Gray
    [bHat]=qamdemod(yk.*EqCoef,M);

    %Tach song tin hieu khong ma hoa Gray
    [bkHat]=qamdemod(ykGray.*EqCoef,M);
    bHatGray(1,:)=Pos(bkHat+1);

    % Tinh toan SER va BER
    [noBitErr,NBER(k)]=biterr(bHat,b,m);    % BER
    [noSymErr,NSER(k)]=symerr(bHat,b);    % SER
    [noBitErr,GBER(k)]=biterr(bHatGray,b,m);% Gray-mapping BER
    [noSymErr,GSER(k)]=symerr(bHatGray,b); %Gray-mapping SER
end

%Voi M=4, ham "qammod" cho ta chom sao tin hieu da ma hoa Gray.
%Do do ta phai hoan doi cac gia tri BER va SER thu duoc o tren.

if M==4
    temp=NBER;  NBER=GBER;  GBER=temp;
end

% Ve do thi
semilogy(EbNodB,BER,'-*',EbNodB,SER./m,'-<', EbNodB,GBER,'-o',...
```

```

EbNodB,NBER,'-^')
hold on
semilogy(EbNodB,SER,'-.+',EbNodB,GSER,'->',EbNodB,NSER,'-.p')
xlabel('E_b/N_o (dB)')
ylabel('BER, SER')

txt=['BER: Theory           ';
      'BER: Upper Bound       ';
      'BER: Simulation with Gray mapping';
      'BER: Simulation w/o Gray mapping ';
      'SER: Theory           ';
      'SER: Simulation with Gray mapping';
      'SER: Simulation w/o Gray mapping '];
legend(txt)

```

### **Function 5.7 qamBERCalAWGN.m**

```

% Ham tinh BER cua he thong QAM trong kenh AWGN
% Written by: Le Minh Tuan, UVISION JSC.
%-----
% Doi voi 4QAM, cong thuc tinh BER la chinh xac
% Doi voi MQAM, cong thuc tinh BER la gan dung

% Lats updated 24/06/2011

function [BER] = qamBERCalAWGN(M, EbNo)
%M: Cap do dieu che
%EbNo: Ty so SNR
%BER: Xac suat loi bit

switch M
    case 4
        BER = erfc(sqrt(EbNo))./2;
    case 16
        BER = erfc(sqrt(EbNo*2/5))*(3/8)+erfc(sqrt(EbNo*18/5))/4;
    case 64
        BER = erfc(sqrt(EbNo/7))*(7/24)+erfc(sqrt(EbNo*9/7))*11/48;
end

```

```

otherwise
BER= [] ;
end
end

```

### ■ Function 5.8 qamSERCalAWGN.m

```

% Ham tinh SER cua he thong QAM trong kenh AWGN
% Written by: Le Minh Tuan, UVISION JSC.
%-----
% Cong thuc tinh SER la chinh xac cho cac gia tri M

% Lats updated 24/06/2011

function [SER] = qamSERCalAWGN(M, EbNo)
%M: Cap do dieu che
%EbNo: Ty so SNR
%SER: Xac suat loi ky hieu
EsNo=EbNo*log2(M);
EsNo_eq = sqrt(3*EsNo/(M-1)/2);
Pm=(2*(sqrt(M)-1)/sqrt(M))*erfc(EsNo_eq)/2;
SER = (2.*Pm-Pm.*Pm);
%end

```

### ■ Function 5.9 qamBERCalRay.m

```

% Ham tinh BER cua he thong QAM trong kenh pha dinh Rayleigh
% Written by: Le Minh Tuan, UVISION JSC.
%-----
% Doi voi 4QAM, cong thuc tinh BER la chinh xac
% Doi voi MQAM, cong thuc tinh BER la gan dung

% Lats updated 24/06/2011

```

```

function [BER] = qamBERCalRay(M, EbNo)
%M: Cap do dieu che

```

```
%EbNo: Ty so SNR
%BER: Xac suat loi bit

switch M
    case 4
        BER = (1-1./sqrt(1+1./EbNo))./2;
    case 16
        BER = (1-1./sqrt(1+1./(2*EbNo/5)))*3/8+...
                (1-1./sqrt(1+1./(18*EbNo/5)))/4;
    case 64
        BER = (1-1./sqrt(1+1./(EbNo/7)))*7/24+...
                (1-1./sqrt(1+1./(9*EbNo/7)))*11/48;
    otherwise
        BER=[];
    end
end
```

#### **Function 5.10 qamSERCalRay.m**

```
% Ham tinh SER cua he thong QAM trong kenh pha dinh Rayleigh
% Written by: Le Minh Tuan, UVISION JSC.
%-----
% Cong thuc tinh SER la chinh xac cho cac gia tri M

% Lats updated 24/06/2011

function [SER] = qamSERCalRay(M, EbNo)
%M: Cap do dieu che
%EbNo: Ty so SNR
%SER: Xac suat loi ky hieu

Gam=log2(M)*EbNo;

sM=sqrt(M);
Psg=@(y)((sM-1)/sM).*erfc(sqrt(3*y/(M-1)/2));

for k=1:length(EbNo)
```

```

F=@(y)(exp(-y./Gam(k))./Gam(k)).*(2.*Psg(y)-Psg(y).*Psg(y));
Ps(k)=quad(F,0,1.0e+3,1.0e-10)+quad(F,1.0e+3,1.0e+8,1.0e-10);
end
SER = Ps;
end

```

---

### ■ Function 5.11 qamGrayCoded.m

```

% Ham anh xa bit du lieu vao cac ky hieu QAM su dung ma hoa Gray
% Written by: Le Minh Tuan, UVISION JSC.
%-----
%
```

% Lats updated 24/06/2011

```

function [ConstOut, Pos]=qamGrayCoded(Const,M)
%Const: Chom sao tin hieu khong duoc ma hoa Gray
%M: Cap do dieu che QAM
%ConstOut: Chom sao tin hieu da duoc ma hoa Gray
%Pos: Du lieu anh xa theo ma hoa Gray
switch M
    case 4
        Pos = [0 1 3 2];
        [a b]=sort(Pos);
        ConstOut=Const(b);
    case 16
        p=[0 1 3 2];
        p1=repmat(p,1,4)';
        p2=repmat(p,4,1);
        p2=reshape(p2,16,1);
        Pos = bitshift(p2,2)+p1;
        [a b]=sort(Pos);
        ConstOut=Const(b);
    case 64
        p=[0 1 3 2 6 7 5 4];
        p1=repmat(p,1,8)';
        p2=repmat(p,8,1);

```

```
p2=reshape(p2,64,1);  
Pos = bitshift(p2,3)+p1;  
[a b]=sort(Pos);  
ConstOut=Const(b);  
otherwise  
    Pos=[];  
    Constel0ut=[];  
end  
end
```

---



## **Phần III**

# **Các hệ thống phân tập và ăng-ten mảng thích nghi**



## Chương 6

# Kỹ thuật thu phát phân tập không gian-thời gian

### 6.1 Các phương pháp phân tập

Trong thông tin vô tuyến, các phương pháp phân tập được sử dụng để hạn chế ảnh hưởng của pha-định và nâng cao chất lượng truyền thông tin. Kỹ thuật kết hợp phân tập (diversity combining) được thử nghiệm trong thông tin vô tuyến lần đầu tiên vào năm 1927 và nghiên cứu rộng rãi vào những năm cuối thập niên 60, đầu 70 [1]. Phương pháp phân tập đòi hỏi sự tồn tại của một số đường truyền có các tham số thống kê độc lập, nhưng truyền tải cùng một thông tin giống nhau. Bản chất của phương pháp phân tập là tín hiệu được truyền trên các đường truyền độc lập sẽ chịu ảnh hưởng của hiệu ứng pha-định khác nhau. Tức là, trong số các tín hiệu thu được sẽ có tín hiệu thu được với chất lượng tốt và có tín hiệu thu được với chất lượng xấu. Do đó, nếu kết hợp các tín hiệu này một cách thích hợp, chúng ta có thể thu được một tín hiệu tổng hợp chịu ảnh hưởng của pha-định ít hơn. Kết quả này đồng nghĩa với việc tín hiệu được truyền đi với độ tin cậy cao hơn.

Tùy theo miền (domain) ứng dụng, các phương pháp phân tập sử dụng trong thông tin vô tuyến có thể được phân loại thành: *phân tập thời gian*, *phân tập tần số*, *phân tập phân cực* và *phân tập không gian* [1],[2].

Trong các phần sau đây, chúng ta sẽ tìm hiểu kỹ về các phương pháp này.

### 6.1.1 Phân tập thời gian

Do tính chất ngẫu nhiên của pha-định, biên độ của một tín hiệu chịu ảnh hưởng pha-định ngẫu nhiên tại các thời điểm lấy mẫu cách xa nhau đủ lớn về thời gian sẽ không tương quan với nhau. Vì vậy, truyền một tín hiệu tại các thời điểm cách biệt đủ lớn tương đương với việc truyền một tín hiệu trên nhiều đường truyền độc lập, tạo nên sự phân tập về thời gian. Khoảng thời gian cần thiết để đảm bảo thu được các tín hiệu pha-định không tương quan tại máy thu tối thiểu là thời gian đồng bộ<sup>5</sup> (coherence time) của kênh truyền [1]. Đối với thông tin di động khoảng thời gian đồng bộ này là  $T_c = c/(2vf_c)$ , trong đó  $c = 3 \times 10^8$  m/s là tốc độ ánh sáng,  $v$  là tốc độ di chuyển của máy di động, và  $f_c$  là tần số sóng mang [1]. Với các máy di động làm việc ở tần số 800 Mhz và di chuyển với tốc độ 50km/h, thời gian đồng bộ  $T_c = 13,5$  ms. Để tạo ra  $M_d$  nhánh phân tập, tín hiệu cần được truyền đi tại  $M_d$  khe thời gian. Vì vậy, khoảng thời gian giữ chậm cần thiết để truyền tín hiệu trên  $M_d$  nhánh phân tập là  $M_d \cdot c/2vf_c$ . Đối với truyền dẫn tín hiệu thoại, tốc độ lấy mẫu cần thiết ít nhất là 8 kHz [1]. Đồng thời, để đảm bảo độ rộng xung truyền nằm trong băng tần truyền dẫn, chúng ta chỉ có thể sử dụng tối đa  $M_d = 50$  nhánh phân tập [1]. Do thời gian cách biệt tỷ lệ nghịch với tốc độ di chuyển nên, khác với các phương pháp phân tập khác, phương pháp phân tập thời gian không có ý nghĩa trong trường hợp máy di động đứng yên.

Gần đây, trong các hệ thống thông tin di động hiện đại, mã sửa lỗi được sử dụng kết hợp với phương pháp xen kẽ tín hiệu (inter-leaving) để tạo nên một phương pháp phân tập thời gian mới. Do thời gian xen kẽ dài sẽ gây nên độ giữ chậm giải mã lớn, nên phương

<sup>5</sup>Thời gian đồng bộ là khoảng thời gian trung bình trong đó đáp ứng của kênh truyền về cơ bản là bằng phẳng [3].

pháp này chỉ thích hợp đối với các kênh pha-định biến động nhanh.

Nhược điểm chính của phương pháp phân tập thời gian là làm suy giảm hiệu suất bằng tần do có sự dư thừa trong miền thời gian.

### 6.1.2 Phân tập tần số

Tương tự như phương pháp phân tập thời gian, chúng ta có thể sử dụng một tập hợp các số tần số để truyền đi cùng một tín hiệu, tạo nên sự phân tập tần số. Khoảng cách giữa các tần số cần phải đủ lớn, vào khoảng vài lần băng tần đồng bộ (coherence bandwidth)<sup>6</sup>, để đảm bảo pha-định ứng với các tần số sử dụng không tương quan với nhau. Đối với thông tin di động, băng tần đồng bộ đo được vào khoảng 500 kHz, vì vậy khoảng cách cần thiết giữa các nhánh phân tập tần số ít nhất là 1–2 MHz [1].

Trong thông tin di động hiện đại, phân tập tần số còn có thể nhận được thông qua việc sử dụng các kỹ thuật điều chế đa sóng mang (multicarrier modulation) hay sử dụng phương pháp nhảy tần (frequency hopping).

Nhược điểm của phương pháp phân tập tần số là sự tiêu tốn phổ tần số. Ngoài ra, do các nhánh phân tập có tần số khác nhau nên mỗi nhánh cần sử dụng một máy thu phát cao tần riêng.

### 6.1.3 Phân tập phân cực

Nghiên cứu cho thấy tín hiệu truyền đi trên 2 phân cực trực giao trong môi trường thông tin di động có các tham số thống kê độc lập [1]. Vì vậy, hai phân cực này có thể được coi là cơ sở của hai nhánh phân tập phân cực. Do chỉ tồn tại hai phân cực sóng trực giao nên số lượng tối đa các nhánh phân tập có thể tạo được chỉ là hai. Ngoài ra, do sự hạn chế của công suất máy phát nên công suất tín hiệu

---

<sup>6</sup>Băng tần đồng bộ là giải tần số trong đó kênh truyền được coi là băng phẳng, tức là, các thành phần phổ tần được cho qua với biên độ xấp xỉ như nhau và pha tuyến tính. Nói cách khác là biên bộ của các thành phần tần số có khả năng tương quan cao [4].

phát cần phải chia đều cho hai nhánh, và vì vậy chất lượng tín hiệu thu cũng bị suy giảm đi 2 lần hay 3 dB.

#### 6.1.4 Phân tập không gian

Phân tập không gian là phương pháp đã được sử dụng rộng rãi trong thông tin vô tuyến. Phương pháp này sử dụng nhiều ăng-ten ở máy thu, máy phát hoặc cả ở phía máy thu và máy phát để tạo nên các nhánh phân tập không gian khác nhau. Khoảng cách cần thiết giữa các ăng-ten tối thiểu là một nửa bước sóng ( $\lambda/2$ ). Khi sử dụng nhiều ăng-ten ở máy phát, chúng ta có hệ thống *phân tập không gian phát*, và tương tự chúng ta có *phân tập không gian thu* nếu sử dụng nhiều ăng-ten thu. Trường hợp sử dụng nhiều ăng-ten tại cả máy phát và máy thu chúng ta có một tập hợp kênh truyền với nhiều đầu vào và nhiều đầu ra. Các hệ thống phân tập thu phát không gian kiểu này thường được gọi là các hệ thống đa đầu vào - đa đầu ra (MIMO: Multiple Input–Multiple Output).

Cũng giống như phương pháp phân tập phân cực, so hạn chế về công suất phát, nếu  $M_d$  nhánh phân tập phát được sử dụng thì chất lượng tín hiệu bị suy giảm đi  $1/M_d$  lần so với phân tập thu không gian cùng bậc. Tuy nhiên, sử dụng phân tập phát rất có ý nghĩa do hai lý do sau: (i) tiết kiệm, do chỉ cần thực hiện ở trạm gốc chứ không phải ở tất cả các máy di động, (ii) dễ thực hiện, do yêu cầu về sự nhỏ gọn của máy di động nên rất khó có thể đặt được hai hay nhiều ăng-ten thu cách xa nhau hơn nửa bước sóng (khoảng 19 cm ở tần 800 MHz và 15cm ở tần số 2 GHz). Vì vậy, sử dụng phân tập phát tại trạm gốc sẽ thay thế cho phân tập tại máy thu.

Ưu điểm của phương pháp phân tập không gian là không làm suy giảm hiệu suất băng tần, không tiêu tốn phổ tần số, dễ sử dụng, và trên lý thuyết không có sự hạn chế về số lượng các nhánh phân tập. Do có các ưu điểm nói trên, phương pháp phân tập không gian đã được nghiên cứu rộng rãi từ năm 1927 đến tận ngày nay. Các nghiên cứu về phân tập không gian tập trung chủ yếu vào các kỹ

thuật kết hợp tín hiệu phân tập. Trong các phần tiếp sau đây, chúng ta sẽ tìm hiểu kỹ hơn về các biện pháp kết hợp phân tập thu và phát không gian.

## 6.2 Kỹ thuật kết hợp phân tập không gian thu

### 6.2.1 Mô hình tín hiệu

Giả sử tín hiệu  $s(t)$  được truyền qua môi trường pha-định Rayleigh tới máy thu sử dụng phân tập không gian với  $M$  nhánh phân tập. Sau khi tách sóng cao tần, tín hiệu thu được tại nhánh phân tập (ăng-ten) thứ  $m$ ,  $m \in \{1, 2, \dots, M\}$ , được biểu diễn như sau:

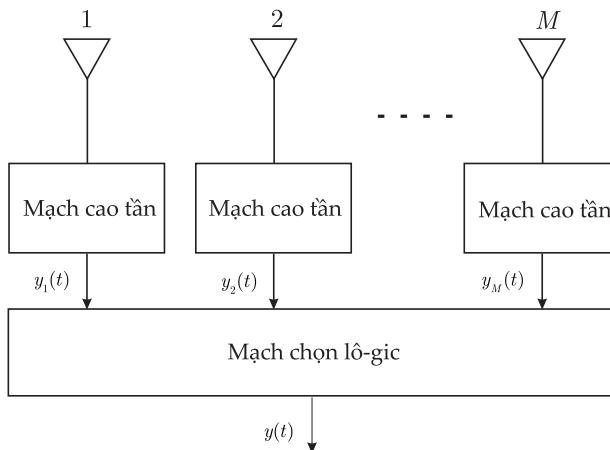
$$y_m(t) = h_m(t)s(t) + z_m(t), \quad (6.1)$$

trong đó  $h_m(t)$ , là một số phức Gauss có giá trị trung bình bằng không, biểu diễn đường truyền pha-định từ ăng-ten phát tới ăng-ten thu thứ  $m$ ;  $z_m(t)$  là tạp âm ở nhánh phân tập  $m$ .

Từ  $M$  tín hiệu nhánh trên, chúng ta mong muốn sử dụng một phương pháp kết hợp thích hợp sao cho tín hiệu ở đầu ra bộ kết hợp có chất lượng tốt hơn. Trong các phần tiếp theo, chúng ta sẽ tìm hiểu ba phương pháp kết hợp phân tập không gian được sử dụng phổ biến ở máy thu. Cụ thể là các phương pháp kết hợp chọn lọc (SC: Selection Combining), kết hợp số cực đại (MRC: Maximal-Ratio Combining) và kết hợp đồng độ lợi (EGC: Equal-Gain Combining).

### 6.2.2 Kết hợp chọn lọc (Selection Combining)

Cấu hình của bộ kết hợp chọn lọc được minh họa ở Hình 6.1. Tại một thời điểm  $t$ , mạch chọn lọc logic thực hiện việc đo lường và tính toán tỷ số tín hiệu trên tạp âm SNR (Signal to Noise Ratio) của từng nhánh phân tập và chọn ra tín hiệu ở nhánh có tỷ số SNR lớn nhất. Trong thực tế, việc đo lường tỉ số SNR rất khó thực hiện và, vì vậy, tín hiệu trên nhánh phân tập có tổng công suất tín hiệu và tạp âm lớn nhất sẽ được chọn.

**Hình 6.1:** Mô hình phương pháp kết hợp chọn lọc.

**Tỉ số công suất tín hiệu trên tạp âm (SNR).** Giả sử tín hiệu thu được tại các nhánh phân tập là không tương quan và có phân phối Rayleigh với công suất trung bình  $\zeta^2 = \zeta_m^2 = \frac{1}{2} E\{|y_m|^2\}$ , trong đó ký hiệu  $E\{\cdot\}$  biểu thị phép toán lấy giá trị kỳ vọng. Hàm mật độ của đường bao tín hiệu trên nhánh  $m$  là

$$p(y_m) = \frac{y_m}{\zeta^2} \exp\left(-\frac{y_m^2}{2\zeta^2}\right), \quad (6.2)$$

trong đó  $y_m$  là đường bao tín hiệu trên nhánh  $m$ . Giả sử công suất trung bình của tạp âm trên các nhánh,  $z_m(t)$ , là như nhau và bằng

$$\sigma^2 = \frac{1}{2} E\{|z_m(t)|^2\}. \quad (6.3)$$

Tỷ số SNR tức thời ở nhánh  $m$  được định nghĩa như sau

$$\gamma_m \triangleq \frac{1}{2} \frac{|y_m|^2}{\sigma^2}. \quad (6.4)$$

Tiếp theo, định nghĩa tỷ số SNR trung bình của  $\gamma_m$

$$\Gamma_m \triangleq E\{\gamma_m\} = \frac{1}{2} \frac{E\{|y_m|^2\}}{\sigma^2} \quad (6.5)$$

ta có hàm mật độ xác suất của  $\gamma_m$  được cho bởi

$$p(\gamma_m) = \frac{1}{\Gamma_m} \exp\left(-\frac{\gamma_m}{\Gamma_m}\right). \quad (6.6)$$

Xác suất SNR của một nhánh phân tập nhỏ hơn một giá trị  $x$  nào đó là

$$F(x) \triangleq \Pr\{\gamma_m < x\} = 1 - \exp\left(-\frac{x}{\Gamma_m}\right). \quad (6.7)$$

Do pha-định trên các nhánh phân tập được giả định là độc lập, xác xuất  $\gamma < x$  chính là tích số của các xác suất để từng  $\gamma_m < x$ , tức là

$$\Pr\{\gamma < x\} = \prod_{m=1}^M \Pr\{\gamma_m < x\} = \prod_{m=1}^M \left[1 - \exp\left(-\frac{x}{\Gamma_m}\right)\right]. \quad (6.8)$$

Như vậy, hàm mật độ xác suất tương ứng là [5]

$$p(\gamma) = \frac{d}{dx} \left\{ \prod_{m=1}^M \left[1 - \exp\left(-\frac{x}{\Gamma_m}\right)\right] \right\}_{x=\gamma} = \frac{d}{d\gamma} \left\{ \prod_{m=1}^M \left[1 - \exp\left(-\frac{\gamma}{\Gamma_m}\right)\right] \right\}. \quad (6.9)$$

Xét trường hợp đặc biệt trong đó tất cả các nhánh phân tập đều có cùng một giá trị SNR trung bình, tức là  $\Gamma_m = \Gamma$ . Khi đó ta có [5]

$$\Pr\{\gamma < x\} = \left[1 - \exp\left(-\frac{x}{\Gamma}\right)\right]^M \quad (6.10)$$

và

$$p(\gamma) = \frac{M}{\Gamma} \exp\left(-\frac{\gamma}{\Gamma}\right) \left[1 - \exp\left(-\frac{\gamma}{\Gamma}\right)\right]^{M-1}. \quad (6.11)$$

Nhờ đó SNR trung bình của một nhánh được chọn là [1]

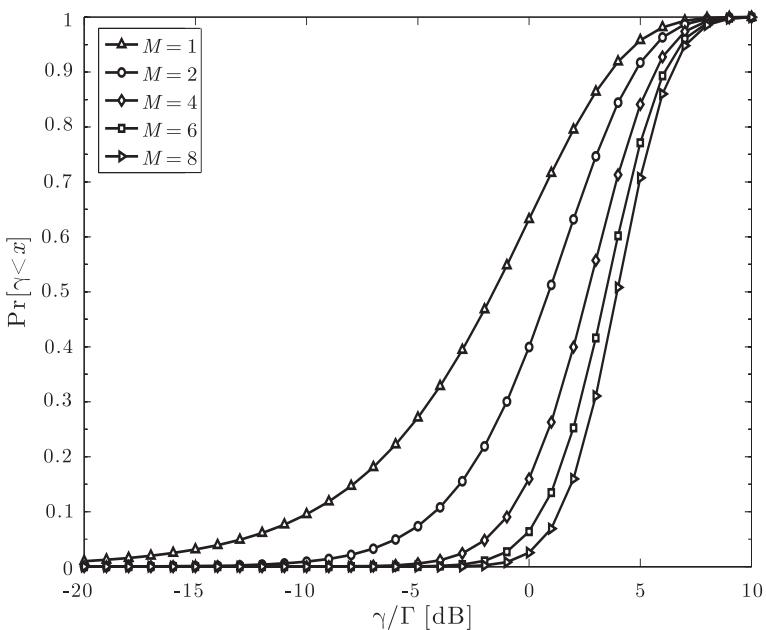
$$\bar{\Gamma} = \mathbb{E}\{\gamma\} = \int_0^\infty \gamma p(\gamma) d\gamma. \quad (6.12)$$

Thay công thức (6.11) vào (6.12) ta thu được [1]

$$\bar{\Gamma}_{sc} \triangleq \bar{\Gamma} = \Gamma \sum_{m=1}^M \frac{1}{m} \quad (6.13)$$

Hình 6.2 biểu diễn phân phối xác suất của SNR cho phương pháp kết hợp phân tập lựa chọn với  $M$  nhánh phân tập sử dụng công thức (6.10). Thông qua hình vẽ ta có thể thấy được hiệu quả của việc tăng số lượng nhánh phân tập  $M$ . Cụ thể là việc tăng số lượng ăng-ten cho phép giảm nhỏ xác suất SNR thấp hơn một giá trị cho trước<sup>7</sup>, hay nói cách khác, làm tăng xác suất SNR lớn hơn một

<sup>7</sup>Tức là giảm outage probability.

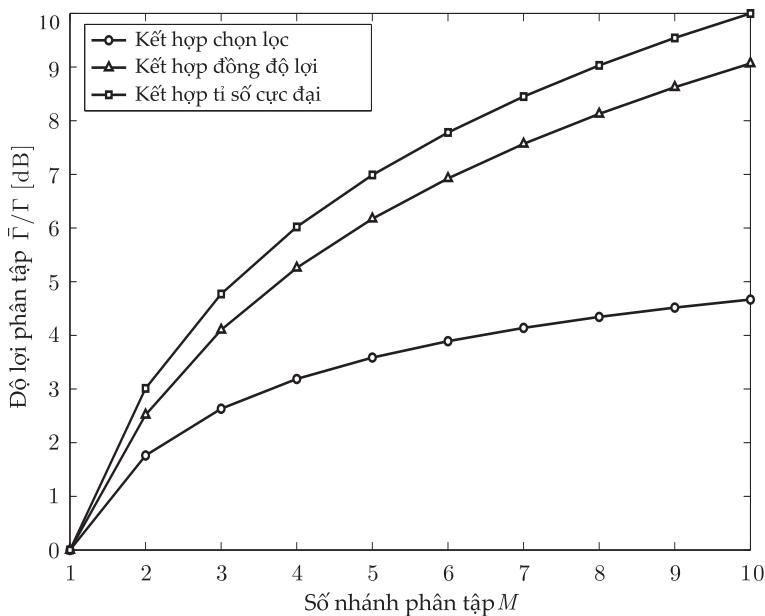


**Hình 6.2:** Phân phối xác xuất (CDF) của SNR cho phương pháp kết hợp phân tập lựa chọn. (Vẽ bằng chương trình `combSNRcdf.m`).

giá trị cho trước. Ngoài ra, từ công thức (6.3) chúng ta cũng nhận thấy rằng, tăng số lượng nhánh phân tập giúp tăng độ lợi phân tập một cách đáng kể. Mức độ độ lợi thu được của phương pháp kết hợp chọn lọc được so sánh với các phương pháp khác ở Hình 6.3. So với các phương pháp khác, phương pháp kết hợp lựa chọn có độ lợi phân tập thấp nhất.

**Xác suất lỗi bít.<sup>8</sup>** Để tính toán xác suất lỗi bít trung bình, chúng ta cần tìm xác suất lỗi có điều kiện gắn liền với các bít đầu ra sau đó lấy trung bình theo các tham số thống kê của SNR tại đầu ra bộ kết hợp phân tập. Với giả thiết pha-định Rayleigh không chọn lọc theo tần số và biến đổi chậm, từ các công thức (9-5-19) và (9-5-20) của [5], chúng ta có xác suất lỗi bít có điều kiện cho các trường hợp tách tín hiệu đồng bộ và không đồng bộ (vi sai) sử dụng điều chế khóa dịch pha (PSK: Phase Shift Keying) và khóa dịch tần số (FSK: Frequency

<sup>8</sup>Xác suất lỗi bít đồng nghĩa với tỉ số lỗi bít BER.



**Hình 6.3:** Độ lợi phân tập của các phương pháp kết hợp phân tập. (Vẽ bằng chương trình combSNRcdf.m)

Shift Keying) như sau:

$$P^{(\text{dif})}(E|\gamma) = \frac{1}{2} \exp(-\alpha\gamma) \quad (6.14)$$

$$P^{(\text{ch})}(E|\gamma) = \frac{1}{2} \operatorname{erfc}(\sqrt{\alpha\gamma}), \quad (6.15)$$

trong đó  $\alpha$  là hằng số điều chế;  $\alpha = 1/2$  đối với trường hợp FSK và  $\alpha = 1$  với PSK.

Sử dụng hàm phân phối xác xuất của SNR đầu ra bộ kết hợp,  $p(\gamma)$ ,  $\gamma > 0$ , ở công thức (6.9), xác suất lỗi trung bình của phương pháp kết hợp chọn lọc đối với *trường hợp tách tín hiệu không đồng bộ* là

$$P_{\text{sc}}^{(\text{dif})} = \int_0^{\infty} P^{(\text{dif})}(E|\gamma)p(\gamma)d\gamma \quad (6.16)$$

$$= \frac{1}{2} \int_0^{\infty} \exp(-\alpha\gamma) \left[ 1 - \exp\left(-\frac{\gamma_{\text{sc}}}{\Gamma}\right) \right]^M d\gamma. \quad (6.17)$$

Giải công thức trên ta thu được[5]

$$P_{sc}^{(dif)} = \frac{1}{2} \frac{M}{\prod_{m=1}^M (m + \alpha\Gamma)}. \quad (6.18)$$

Cuối cùng, thay  $\alpha = 1$  và  $\alpha = 1/2$  vào công thức trên ta có xác suất lỗi bít trung bình tương ứng cho các trường hợp DPSK và FSK không đồng bộ. Trong trường hợp tỉ số SNR đầu vào thấp, thỏa mãn điều kiện  $\alpha\Gamma \gg M$ , giá trị gần đúng của xác suất lỗi bít là

$$P_{sc}^{(dif)} \approx \frac{1}{2} \frac{M}{(\alpha\Gamma)^M}. \quad (6.19)$$

Đối với *tách tín hiệu đồng bộ*, sử dụng phương pháp tính toán tương tự như trường hợp tách tín hiệu không đồng bộ, chúng ta có [5]

$$P_{sc}^{(ch)} = \int_0^\infty P^{(ch)}(E|\gamma) p(\gamma) d\gamma \quad (6.20)$$

$$= \frac{1}{2} \int_0^\infty \operatorname{erfc}(\sqrt{\alpha\gamma}) \left[ 1 - \exp\left(-\frac{\gamma_{sc}}{\Gamma}\right) \right]^M d\gamma. \quad (6.21)$$

### 6.2.3 Kết hợp tỉ số cực đại (Maximal Ratio Combining)

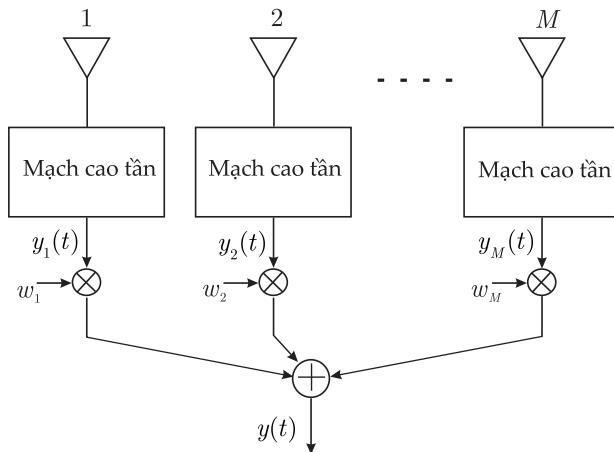
Phương pháp kết hợp tỉ số cực đại<sup>9</sup> được Kahn đề xuất năm 1954 [6]. Sử dụng phương pháp này, tín hiệu của  $M$  nhánh phân tập được nhân trọng số (weighted) cân xứng theo tỉ số SNR của các nhánh, sau đó được điều chỉnh đồng pha rồi kết hợp (cộng) với nhau. Sơ đồ cấu hình một bộ kết hợp tỉ số cực đại được trình bày ở Hình 6.4.

*Tỉ số công suất tín hiệu trên tần âm (SNR).* Tín hiệu thu được tại một nhánh phân tập, sau khi được chuyển về băng tần gốc (base-band) được biểu diễn như sau:

$$y_m(t) = h_m(t)s(t) + z_m(t), \quad (6.22)$$

trong đó  $h_m(t)$ , là một biến Gauss phức có giá trị trung bình bằng không, biểu diễn đường truyền pha-định từ ăng-ten phát tới ăng-

<sup>9</sup>Tên gọi Tỉ số cực đại xuất phát từ ý nghĩa đây là phương pháp kết hợp cho phép thu được tỉ số SNR ở đầu ra cực đại.



**Hình 6.4:** Mô hình phương pháp kết hợp tỷ số cực đại.

ten thu thứ  $m$ ;  $z_m(t)$  là tạp âm ở nhánh phân tập  $m$ . Sau khi được đồng pha và nhân trọng số, đường bao của tín hiệu kết hợp là

$$y(t) = \sum_{m=1}^M w_m y_m(t) \quad (6.23)$$

với  $w_m$  là một số phức, thường được gọi là trọng số kết hợp (combining weight). Thay (6.22) vào (6.23) ta có

$$y(t) = \sum_{m=1}^M w_m [h_m(t)s(t) + z_m(t)] \quad (6.24)$$

$$= s(t) \sum_{m=1}^M w_m h_m(t) + \sum_{m=1}^M w_m z_m(t). \quad (6.25)$$

Từ công thức (6.25) ta có công suất tạp âm tức thời ở đầu ra bộ kết hợp là

$$\sigma_{\text{mrc}}^2 = \frac{1}{2} \text{E}\{|z(t)|^2\} = \sum_{m=1}^M |w_m|^2 \sigma_m^2. \quad (6.26)$$

Như vậy, tỉ số SNR tức thời ở đầu ra là

$$\gamma_{\text{mrc}} = \frac{1}{2} \frac{\left| \sum_{m=1}^M w_m h_m \right|^2}{\sum_{m=1}^M |w_m|^2 \sigma_m^2}. \quad (6.27)$$

Để tìm giá trị lớn nhất của SNR đầu ra, sử dụng bất đẳng thức sau của Schwarz [5]:

$$\left| \sum_{m=1}^M a_m^* b_m \right|^2 \leq \left( \sum_{m=1}^M |a_m|^2 \right) \left( \sum_{m=1}^M |b_m|^2 \right). \quad (6.28)$$

Điều kiện để thu được dấu bằng là

$$b_m = C a_m, \quad \forall m \quad (6.29)$$

với  $C$  là một hằng số bất kỳ. Đặt

$$a_m \triangleq \frac{h_m^*}{\sigma_m} \quad (6.30)$$

$$b_m \triangleq w_m \sigma_m \quad (6.31)$$

ta có

$$\left| \sum_{m=1}^M w_m h_m \right|^2 \leq \left( \sum_{m=1}^M \frac{|h_m|^2}{\sigma_m^2} \right) \left( \sum_{m=1}^M |w_m|^2 \sigma_m^2 \right). \quad (6.32)$$

với dấu hoa thị phía trên \* biểu diễn phép toán lấy liên hợp phức. So sánh (6.32) với (6.36) ta thu được

$$\gamma_{\text{mrc}} \leq \frac{1}{2} \sum_{m=1}^M \frac{|h_m|^2}{\sigma_m^2} \quad (6.33)$$

với dấu bằng xảy ra khi và chỉ khi

$$w_m = C \cdot \frac{h_m^*}{\sigma_m^2}. \quad (6.34)$$

Đây chính là giá trị của các trọng số của phương pháp kết hợp tỉ số cực đại. Trong trường hợp công suất tạp âm trên các nhánh phân tập là như nhau, tức là,  $\sigma_m^2 = \sigma_z^2$ , ta nhận thấy  $\frac{C}{\sigma_m^2}$  là một hằng số chung cho tất cả các nhánh phân tập. Do đó, ta có thể loại bỏ  $\frac{C}{\sigma_m^2}$  khỏi  $w_m$  mà không làm ảnh hưởng đến SNR đầu ra. Như vậy,  $w_m$  có thể được biểu diễn rút gọn thành

$$w_m = h_m^*. \quad (6.35)$$

Quan sát (6.32) chúng ta thấy về phải của bất đẳng thức chính là tổng giá trị của SNR trên tất cả các nhánh phân tập. Hay nói cách

khác, giá trị tối đa của SNR tức thời ở đầu ra bộ kết hợp tỉ số cực đại là

$$\gamma_{\text{mrc}} = \frac{1}{2} \sum_{m=1}^M \frac{|h_m|^2}{\sigma_m^2} = \sum_{m=1}^M \gamma_m. \quad (6.36)$$

Theo [1],  $\gamma_m$  có thể phân tích thành  $2M$  biến ngẫu nhiên Gauss có cùng giá trị trung bình bằng không và cùng variance  $\zeta^2$ . Như vậy,  $\gamma_{\text{mrc}}$  chính là phân bố  $\chi^2$  (chi-square) của  $2M$  biến ngẫu nhiên Gauss với phương sai  $\zeta^2/2\sigma^2 = \Gamma/2$ . Vì vậy, hàm mật độ xác suất của  $\gamma$  trong trường hợp  $\Gamma_m = \Gamma$  có thể viết được như sau [1]:

$$p(\gamma) = \frac{1}{(M-1)!} \frac{\gamma^{M-1}}{\Gamma^M} \exp\left(-\frac{\gamma}{\Gamma}\right), \quad \gamma \geq 0. \quad (6.37)$$

Từ đây, chúng ta có thể tính được hàm phân bố xác suất của  $\gamma$  bằng cách lấy tích phân của hàm mật độ xác suất

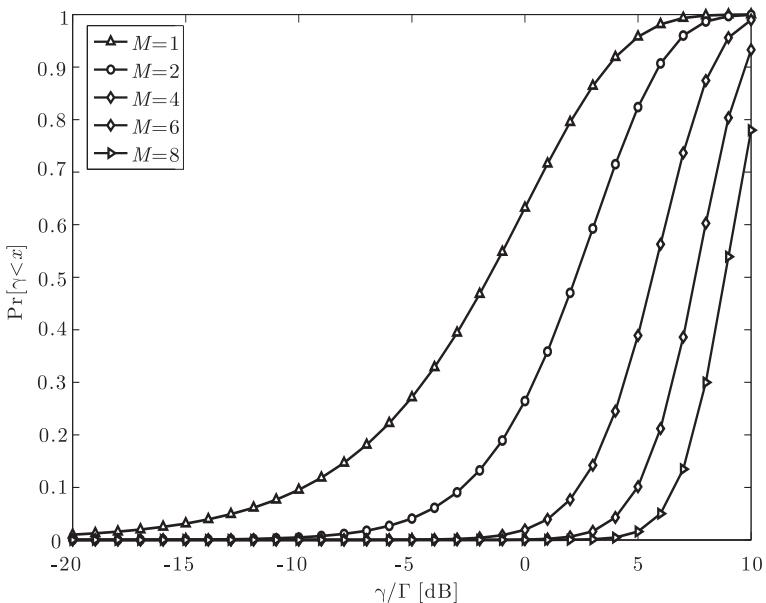
$$\begin{aligned} \Pr\{\gamma < x\} &= \int_0^x p(\gamma) d\gamma = \frac{1}{\Gamma^M (M-1)!} \int_0^x \gamma^{M-1} \exp\left(-\frac{\gamma}{\Gamma}\right) d\gamma \\ &= 1 - \exp\left(-\frac{x}{\Gamma}\right) \sum_{m=1}^M \frac{\left(\frac{x}{\Gamma}\right)^{m-1}}{(m-1)!} \end{aligned} \quad (6.38)$$

Hàm phân phối xác suất  $\Pr\{\gamma < x\}$  của phương pháp kết hợp tỉ số cực đại được mô tả ở Hình 6.5. So sánh với phương pháp kết hợp chọn lọc (Hình 6.2), phương pháp kết hợp tỉ số cực đại có tỉ lệ quá tải (outage rate)<sup>10</sup> tốt hơn. Thực tế, phương pháp kết hợp tỉ số cực đại là phương pháp kết hợp cho độ lợi lớn nhất. Phương pháp kết hợp này còn được gọi là phương pháp *kết hợp tối ưu* (optimum combining).

#### 6.2.4 Kết hợp đồng độ lợi (Equal Gain Combining)

Như đã chỉ ra ở phần trước, MRC là phương pháp kết hợp tối ưu cho độ lợi phân tập lớn nhất trong tất cả các phương pháp kết hợp phân tập thu. Tuy nhiên, phương pháp MRC yêu cầu phải biết chính xác được các trọng số kết hợp  $w_m$ , do đó tương đối phức tạp. Hơn nữa, độ lợi thu được của phương pháp MRC không lớn hơn nhiều

<sup>10</sup>Cần hiểu chính xác tỉ lệ quá tải là tỉ lệ ngắt kết nối của một liên kết do SNR nhỏ hơn ngưỡng cho trước.



**Hình 6.5:** Phân phối xác xuất (CDF) của SNR cho phương pháp kết hợp tỉ số cực đại. (Vẽ bằng chương trình combSNRcdf.m)

so với phương pháp kết hợp chọn lọc. Điều này có nghĩa là phần lớn độ lợi phân tập thu được từ nhánh phân tập có công suất lớn nhất và nếu một phương pháp kết hợp có thể thu được độ lợi từ nhánh phân tập đó thì tổng độ lợi thu được hầu như không thay đổi. Quan sát này dẫn đến một phương pháp phân tập mới, kỹ thuật kết hợp phân tập đồng độ lợi (EGC: Equal Gain Combiner), đơn giản hơn phương pháp MRC. Sử dụng phương pháp kết hợp EGC, tín hiệu tại các nhánh được đồng pha (co-phasing) giống như trường hợp MRC, nhưng sau đó được nhân với các trọng số có cùng độ lớn, rồi kết hợp với nhau. Trường hợp đơn giản nhất là đặt độ lợi của các trọng số bằng hằng số đơn vị. Như vậy, phương pháp kết hợp EGC chỉ là một trường hợp đặc biệt của phương pháp MRC. Sử dụng

công thức (6.36), ta có tỉ số SNR ở đầu ra bộ kết hợp EGC là

$$\gamma_{\text{egc}} = \frac{1}{2} \frac{\left[ \sum_{m=1}^M |h_m| \right]^2}{\sum_{m=1}^M \sigma_m^2}. \quad (6.39)$$

Giả sử tỉ số SNR  $\Gamma_m$  trên các nhánh phân tập như nhau, hàm mật độ xác suất của SNR tại đầu ra bộ kết hợp với các giá trị  $\gamma$  nhỏ là [5]

$$p(\gamma) = \frac{2^{M-1}}{(2M-1)!} \frac{1}{\prod_{m=1}^M \frac{\sigma_m^2}{\sigma_{\Sigma}^2}} \frac{\gamma^{M-1}}{\prod_{m=1}^M \Gamma_m} \quad (6.40)$$

trong đó

$$\sigma_{\Sigma}^2 = \sum_{m=1}^M \sigma_m^2. \quad (6.41)$$

Để cực đại hóa độ lợi kết hợp của phương pháp EGC, ta cần cực thiểu hóa  $p(\gamma)$  ở (6.40). Điều này tương đương với việc cực đại hóa thành phần mẫu số. Tuy nhiên, từ (6.41) ta có

$$\sum_{m=1}^M \frac{\sigma_m^2}{\sigma_{\Sigma}^2} = 1. \quad (6.42)$$

Như vậy, ta cần phải tìm một tập hợp của  $M$  sao cho tích của chúng là tối đa với điều kiện tổng của chúng là đơn vị. Sử dụng phương pháp bội số Lagrange (Lagrange multipliers), thì giá trị của từng hệ số là  $1/M$ . Do đó ta thu được

$$\prod_{m=1}^M \frac{\sigma_m^2}{\sigma_{\Sigma}^2} = \left( \frac{1}{M} \right)^M. \quad (6.43)$$

Từ đây,  $p(\gamma)$  ở (6.40) trở thành

$$p(\gamma) = \frac{2^{M-1} M^M}{(2M-1)!} \frac{\gamma^{M-1}}{\prod_{m=1}^M \Gamma_m}. \quad (6.44)$$

Hàm phân phối xác suất tương ứng của SNR ở đầu ra bộ kết hợp là [5]

$$\Pr\{\gamma < x\} = \int_0^x p(\gamma) d\gamma = \frac{(2M)^M}{(2M)!} \frac{x^M}{\prod_{m=1}^M \Gamma_m}. \quad (6.45)$$

Dựa vào (6.38) và (6.45), Schwartz và cộng sự [5] đã tính được tỉ số độ lợi của phương pháp MRC so với phương pháp EGC là

$$\eta = \frac{M}{2} \frac{1}{\sqrt[M]{(M - \frac{1}{2})! / \sqrt{\pi}}}. \quad (6.46)$$

Với  $M \gg 1$  thì  $\eta \approx 1.36$  hay 1.34 dB. Điều này có thể thấy rõ ở Hình 6.3 khi so sánh độ lợi phân tập  $\bar{\Gamma}/\Gamma$ .

### 6.2.5 Kết hợp phân tập thu và tách tín hiệu MLD

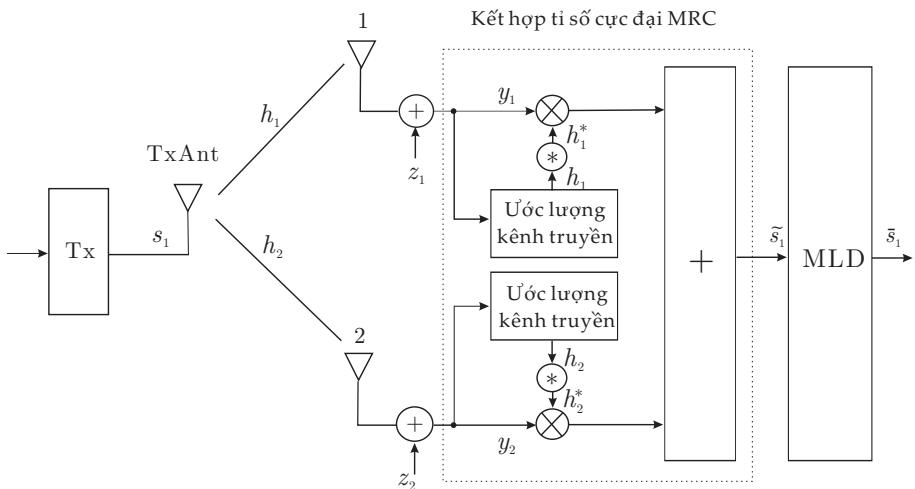
Trong phần trước, chúng ta đã biết phương pháp kết hợp phân tập MRC là phương pháp kết hợp tối ưu. Vì vậy, phương pháp này được sử dụng phổ biến nhất trong thực tế. Trong phần này chúng ta sẽ nghiên cứu sơ đồ thực hiện phương pháp MRC kết hợp với phương pháp tách tín hiệu hợp lệ cực đại (MLD: Maximum Likelihood Detection). Mô hình hệ thống của một hệ thống vô tuyến trong đó máy thu sử dụng 2 nhánh phân tập MRC được chỉ ra ở Hình 6.6.

**Tín hiệu đầu ra bộ kết hợp MRC:** Giả sử tại một thời điểm  $k$ , một tín hiệu  $s_k$  được truyền đi từ máy phát. Độ lợi kênh truyền giữa ăng-ten máy phát tới các ăng-ten máy thu thứ nhất và thứ hai tại thời điểm  $k$  được ký hiệu là  $h_{1,k}$  và  $h_{2,k}$ . Giả sử kênh truyền chịu ảnh hưởng của pha-định biến đổi chậm. Nhờ đó ta có thể bỏ qua chỉ số thời gian  $k$ , tức là,  $h_{1,k} = h_1$  và  $h_{2,k} = h_2$ . Các kênh này được biểu diễn như sau:

$$h_1 = \alpha_1 e^{j\phi_1} \quad (6.47)$$

$$h_2 = \alpha_2 e^{j\phi_2} \quad (6.48)$$

trong đó  $\alpha_m$  và  $\phi_m$  tương ứng là biên độ phức và pha của kênh.



**Hình 6.6:** Sơ đồ máy thu với 2 nhánh phân tập MRC và một bộ tách tín hiệu tối ưu.

Các tín hiệu thu tại hai nhánh phân tập là

$$y_{1,k} = h_1 s_k + z_{1,k} \quad (6.49)$$

$$y_{2,k} = h_2 s_k + z_{2,k} \quad (6.50)$$

trong đó \$z\_{m,k}\$ là các mẫu (samples) tạp âm Gauss.

Như đã chỉ ra ở phần trước, để tối đa hóa SNR đầu ra, các tín hiệu thu trên hai nhánh phân tập được nhân với các hệ số trọng số \$w\_m\$ rồi kết hợp lại với nhau. Theo công thức (6.35) ta có \$w\_m = h\_m^\*\$, tức là

$$w_1 = h_1^* \quad (6.51)$$

$$w_2 = h_2^*. \quad (6.52)$$

Vì vậy, sử dụng các bộ ước lượng kênh truyền (channel estimator, xem Hình 6.6) ở máy thu, ta có thể biết được các kênh truyền \$h\_1\$ và \$h\_2\$, và nhờ đó tính được các trọng số kết hợp \$w\_1\$ và \$w\_2\$. Giả sử các kênh truyền ước lượng được \$\tilde{h}\_1\$ và \$\tilde{h}\_2\$ là chính xác, tức là, \$\tilde{h}\_1 = h\_1\$ và \$\tilde{h}\_2 = h\_2\$. Sử dụng phương pháp MRC, tín hiệu đầu ra bộ kết hợp được cho bởi

$$\tilde{s}_k = y_{1,k} w_1 + y_{2,k} w_2. \quad (6.53)$$

Thay các công thức (6.51) và (6.52) vào (6.53), sau đó sử dụng công thức (6.49) và (6.50), ta có

$$\tilde{s}_k = y_{1,k}h_1^* + y_{2,k}h_2^* \quad (6.54)$$

$$= (h_1 s_k + z_{1,k})h_1^* + (h_2 s_k + z_{2,k})h_2^* \quad (6.55)$$

$$= |h_1|^2 s_k + z_{1,k}h_1^* + |h_2|^2 s_k + z_{2,k}h_2^* \quad (6.56)$$

được biểu diễn rút gọn thành

$$\tilde{s}_k = (|h_1|^2 + |h_2|^2) s_k + z_{1,k}h_1^* + z_{2,k}h_2^*. \quad (6.57)$$

**Tách tín hiệu hợp lệ cực đại:** Giả sử rằng tín hiệu phát là  $s_k \in \chi_c$ , trong đó  $\chi_c$  là sơ đồ chòm sao tín hiệu (signal constellation). Do các mẫu tập âm  $z_{m,k}$  có phân phối Gauss, luật quyết định hợp lệ cực đại (maximum likelihood decision rule) tại máy thu là

$$\begin{aligned} \bar{s}_k &= \arg \min_{s_k \in \chi_c} \left\{ \sum_{m=1}^{M=2} |y_{m,k} - h_m s_k|^2 \right\} \\ &= \arg \min_{s_k \in \chi_c} \left\{ \sum_{m=1}^{M=2} d^2(y_{m,k}, h_m s_k) \right\} \end{aligned} \quad (6.58)$$

trong đó

$$d^2(y_{m,k}, h_m s_k) \triangleq |y_{m,k} - h_m s_k|^2 = (y_{m,k} - h_m s_k)(y_{m,k} - h_m s_k)^*$$

là bình phương khoảng cách Euclid giữa các tín hiệu thu được  $y_{m,k}$  và các tín hiệu không chứa tạp âm  $h_m s_k$ .

Khai triển đối số của công thức (6.58) ta có

$$\begin{aligned} A_{MRC} &= |y_{1,k} - h_1 s_k|^2 + |y_{2,k} - h_2 s_k|^2 \\ &= (y_{1,k} - h_1 s_k)(y_{1,k} - h_1 s_k)^* + (y_{2,k} - h_2 s_k)(y_{2,k} - h_2 s_k)^* \\ &= y_{1,k} y_{1,k}^* - h_1 s_k y_{1,k}^* - y_{1,k} h_1^* s_k^* + h_1 s_k (h_1 s_k)^* \\ &\quad + y_{2,k} y_{2,k}^* - h_2 s_k y_{2,k}^* - y_{2,k} h_2^* s_k^* + h_2 s_k (h_2 s_k)^*. \end{aligned} \quad (6.59)$$

Do  $|y_{1,k}|^2 = y_{1,k} y_{1,k}^*$  và  $|y_{2,k}|^2 = y_{2,k} y_{2,k}^*$  là các thành phần chung và không ảnh hưởng tới quyết định ở (6.58), chúng ta có thể bỏ qua các

thành phần này và thu được

$$\begin{aligned}
 A_{\text{MRC}} &= -h_1 s_k y_{1,k}^* - y_{1,k} h_1^* s_k^* + h_1 s_k (h_1 s_k)^* \\
 &\quad - h_2 s_k y_{2,k}^* - r_{2,k} h_2^* s_k^* + h_2 s_k (h_2 s_k)^* \\
 &= -h_1 y_{1,k}^* s_k - y_{1,k} h_1^* s_k^* + |h_1|^2 |s_k|^2 - h_2 y_{2,k}^* s_k - y_{2,k} h_2^* s_k^* + |h_2|^2 |s_k|^2 \\
 &= |h_1|^2 |s_k|^2 + |h_2|^2 |s_k|^2 - \underbrace{(h_1 y_{1,k}^* + h_2 y_{2,k}^*) s_k}_{\tilde{s}_k^*} - \underbrace{(y_{1,k} h_1^* + y_{2,k} h_2^*) s_k}_{\tilde{s}_k} \\
 &= (|h_1|^2 + |h_2|^2) |s_k|^2 - \tilde{s}_k^* s_k - \tilde{s}_k s_k^* \tag{6.60}
 \end{aligned}$$

Để ý rằng

$$|h_m|^2 = h_m h_m^* = \alpha_m e^{j\phi_m} \alpha_m e^{-j\phi_m} = \alpha_m^2$$

ta có

$$A_{\text{MRC}} = (\alpha_1^2 + \alpha_2^2) |s_k|^2 - \tilde{s}_k^* s_k - \tilde{s}_k s_k^* \tag{6.61}$$

$$= (\alpha_1^2 + \alpha_2^2) |s_k|^2 + |\tilde{s}_k - s_k|^2 - |\tilde{s}_k|^2 - |s_k|^2. \tag{6.62}$$

Để ý tiếp rằng  $|\tilde{s}_k|^2$  là thành phần chung và thực tế không ảnh hưởng tới luật quyết định, ta có thể bỏ qua thành phần này và thu được

$$A_{\text{MRC}} = (\alpha_1^2 + \alpha_2^2) |s_k|^2 - |s_k|^2 + |\tilde{s}_k - s_k|^2 \tag{6.63}$$

$$= (\alpha_1^2 + \alpha_2^2 - 1) |s_k|^2 + |\tilde{s}_k - s_k|^2. \tag{6.64}$$

Vì vậy, luật quyết định ML ở (6.58) cho phương pháp MRC lược giản thành

$$\bar{s}_k = \arg \min_{s_k \in \chi_c} \{(\alpha_1^2 + \alpha_2^2 - 1) |s_k|^2 + |\tilde{s}_k - s_k|^2\}. \tag{6.65}$$

Đối với trường hợp tín hiệu PSK (PSK: phase shift keying), năng lượng tín hiệu,  $E_s = |s_k|^2$ , là như nhau cho tất cả các điểm phân phối tín hiệu. Do đó, luật quyết định ML tiếp tục được lược giản thành

$$\bar{s}_k = \arg \min_{s_k \in \chi_c} \{|\tilde{s}_k - s_k|^2\}. \tag{6.66}$$

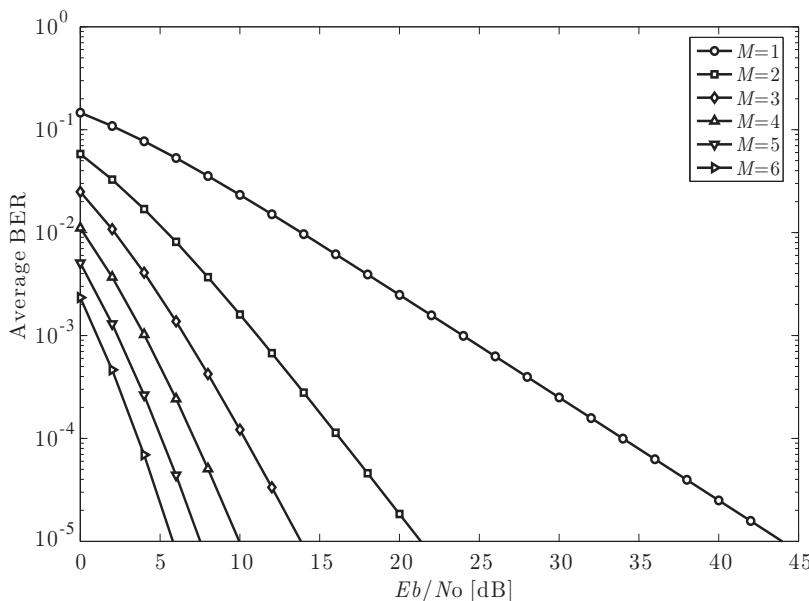
**Phẩm chất BER của phương pháp MRC:** Định nghĩa tỉ số SNR trung bình cho một bít tín hiệu tại từng nhánh phân tập là

$$\bar{\gamma}_m \triangleq E\{\gamma_m\} = \frac{E\{\alpha_m^2\}}{\sigma^2} \tag{6.67}$$

Giả sử rằng tất cả các nhánh phân tập đều có chung một tỉ số SNR trung bình, tức là,  $\bar{\gamma} \equiv \bar{\gamma}_m$ . Tỉ số lỗi bít trung bình BER của tín hiệu BPSK là [7]:

$$P_{\text{MRC}} = \left[ \frac{1}{2} \left( 1 - \sqrt{\frac{\bar{\gamma}}{1 + \bar{\gamma}}} \right) \right]^M \sum_{m=0}^{M-1} \binom{M-1+m}{m} \left[ \frac{1}{2} \left( 1 + \sqrt{\frac{\bar{\gamma}}{1 + \bar{\gamma}}} \right) \right]^m. \quad (6.68)$$

Hình 6.7 mô tả BER trung bình của tín hiệu BPSK cho các trường hợp số nhánh phân tập bằng  $M = \{1, 2, \dots, 6\}$ .

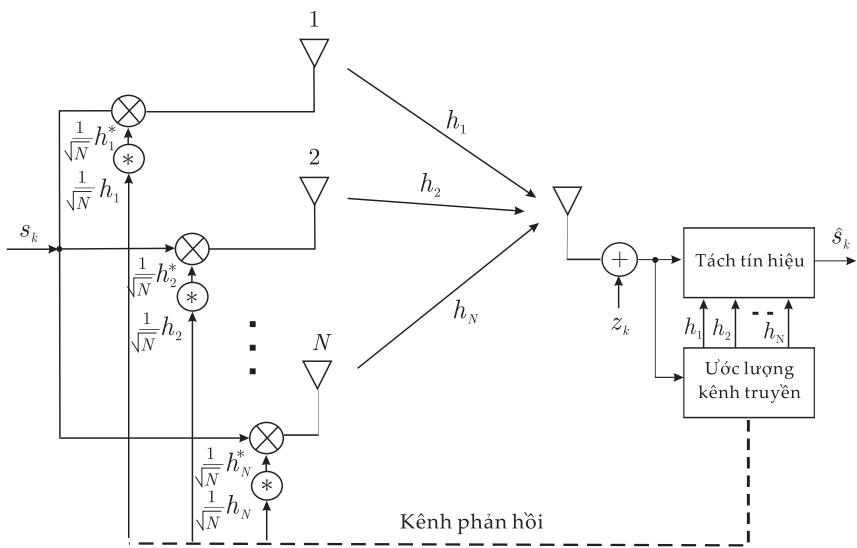


Hình 6.7: Phẩm chất BER trung bình của máy thu MRC với  $M$  nhánh phân tập sử dụng điều chế BPSK.

### 6.3 Kỹ thuật kết hợp phân tập không gian phát

Phân tập phát được tạo nên bởi việc sử dụng nhiều ăng-ten phát kết hợp với một phương pháp xử lý tín hiệu thích hợp. Một số phương pháp phân tập phát điển hình được đề xuất gần đây là:

1. Phân tập phát tỉ số cực đại (MRT: Maximal-Ratio Transmit) [8].



**Hình 6.8:** Sơ đồ phân tập MRT có  $N$  nhánh phân tập với các đường phản hồi.

2. Phân tập phát giữ chậm [9].
3. Phân tập phát không gian-thời gian [10].

Trong phần dưới đây chúng ta sẽ tìm hiểu thêm về các phương pháp này.

### 6.3.1 Phân tập phát tỉ số cực đại (MRT)

Tương tự như phương pháp kết hợp phân tập thu tỉ số cực đại (MRC) trình bày ở phần trước, phương pháp phân tập phát tỉ số cực đại MRT nhân các tín hiệu trên các nhánh phân tập,  $s_k$ , với các hệ số trọng số phát tương ứng,  $w_n C \cdot h_n^*$ ;  $n \in \{1, 2, \dots, N\}$ , sau đó truyền các tín hiệu được nhân trọng số này thông qua  $N$  ăng-ten phát. Hệ số chuẩn hóa công suất  $C$  được chọn sao cho tổng công suất phát đi từ  $N$  ăng-ten phát bằng một giá trị cho trước, thông thường đặt bằng  $C = \frac{1}{\sqrt{N}}$  cho đơn giản. Chú ý rằng để tìm được các hệ số trọng số  $w_n$ , ta cần phải biết các kênh truyền  $h_n$ . Việc này có thể thực hiện được bằng các cách sau:

- (i) Máy thu ước lượng  $h_n$  và gửi thông tin về  $h_n$  về cho máy phát.

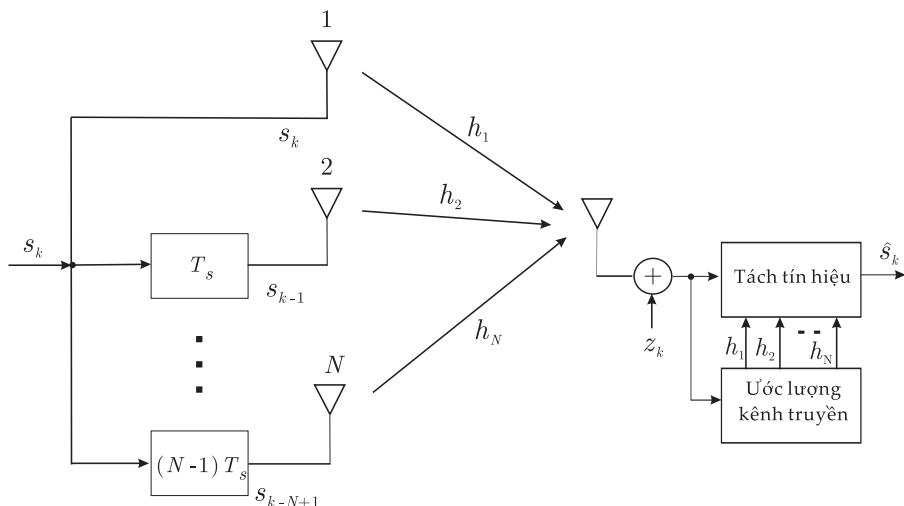
Phương pháp này đòi hỏi phải có kênh phản hồi từ máy thu về máy phát.

- (ii) Sử dụng tính chất thuận nghịch của các kênh truyền lên (up-link) và kênh truyền xuống (down-link). Thực tế là ở các hệ thống phân chia theo tần số (FDD: Frequency-Division Duplex) có khoảng cách tần số thu và phát nhỏ (nhỏ hơn độ rộng băng tần đồng bộ kênh truyền), thì các hệ số kênh truyền lên và kênh truyền xuống rất tương quan với nhau [8]. Do đó, chúng ta có thể sử dụng các kênh truyền  $h_n$  ước lượng được ở kênh truyền lên như là các hệ số trọng số phát  $w_n$ .

Sơ đồ cấu hình của một bộ phân tập MRT có  $N$  nhánh phân tập với các đường phản hồi được mô tả ở Hình 6.8.

Bộ phân tập MRT cho chúng ta bậc phân tập (diversity order) giống như của một bộ phân tập thu MRC. Tuy nhiên, do tổng công suất phát được chuẩn hóa thành đơn vị, nên bị thiệt  $\eta_{\text{loss}} = 10 \log_{10} N$  [dB] về phẩm chất ỗi bít (BER).

### 6.3.2 Phân tập phát giữ chậm



Hình 6.9: Sơ đồ cấu hình phân tập phát giữ chậm với  $N$  nhánh phân tập.

Cấu hình phân tập phát giữ chậm được mô tả ở Hình 6.9. Các bản sao của tín hiệu  $s_k$  được truyền tới máy thu tại các thời điểm khác nhau và thông qua các ăng-ten phát khác nhau. Các tín hiệu giữ chậm được máy thu coi như các tín hiệu đa đường. Vì vậy, để tách được các tín hiệu phát, máy thu sử dụng một bộ san bằng ước lượng chuỗi hợp lệ cực đại (MLSE: Maximum Likelihood Sequence Estimator) hay một bộ san bằng sai số bình phương trung bình cực tiểu (MMSE: Minimum Mean Square Error) để đạt được độ lợi phân tập  $N$ . Ưu điểm của phương pháp phân tập phát giữ chậm là nó có thể cho độ lợi phân tập bậc  $N$  mà không yêu cầu phải mở rộng băng tần, cũng như cần phản hồi từ máy thu. Một ưu điểm khác của phương pháp này là nó có thể áp dụng trực tiếp cho các kênh đa đường để thu được thêm độ lợi phân tập đa đường (path diversity). Phương pháp phân tập phát giữ chậm được coi là trường hợp đơn giản của mã xoắn không gian-thời gian [11] được đề xuất gần đây.

### 6.3.3 Phân tập phát không gian-thời gian

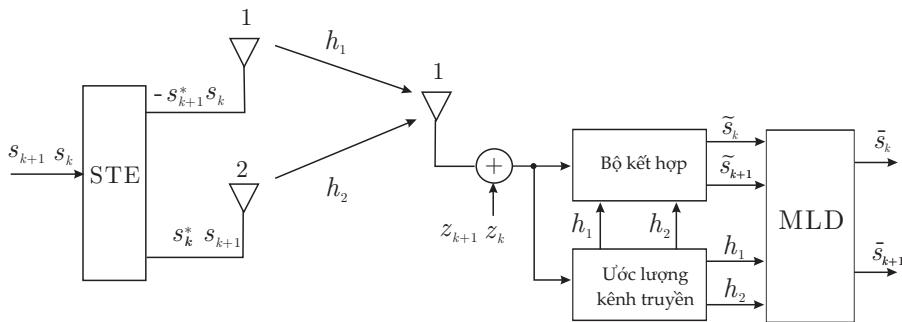
#### Trường hợp kênh MISO $2 \times 1$

Hình 6.10 mô tả sơ đồ phân tập phát không gian-thời gian do Alamouti đề xuất năm 1998 sử dụng hai ăng-ten phát và một ăng-ten thu [10]. Phương pháp phân tập này còn được gọi là mã khối không gian-thời gian (STBC: Space-Time Block Code) Alamouti, mặc dù về mặt bản chất nó không hề đem lại độ lợi mã hóa.

**Bảng 6.1:** Quy luật mã hóa không gian-thời gian Alamouti

	Thời gian $k$	Thời gian $k + 1$
Ăng-ten phát 1	$s_k$	$-s_{k+1}^*$
Ăng-ten phát 2	$s_{k+1}$	$s_k^*$

**Phương pháp mã hóa:** Tại một chu kỳ symbol tín hiệu (symbol period) cho trước hai symbol (dấu)  $s_k$  và  $s_{k+1}$  được mã hóa cả về



**Hình 6.10:** Sơ đồ máy phát mã khồi STBC Alamouti với 2 ăng-ten phát và 1 ăng-ten thu.

không gian và thời gian theo như mô tả ở Bảng 6.1. Tại khe thời gian  $k$  ăng-ten phát thứ nhất phát đi  $s_k$  trong khi ăng-ten phát thứ hai phát đi  $s_{k+1}$ . Tại khe thời gian tiếp theo,  $k + 1$ , ăng-ten phát thứ nhất phát đi  $-s_{k+1}^*$  trong khi đó ăng-ten thứ hai phát đi  $s_k^*$ . Bằng cách này việc mã hóa được thực hiện theo cả 2 miền không gian và thời gian.

Tương tự như trường hợp kết hợp thu phân tập MRC, giả sử kênh pha-định biến đổi chậm, tức là, pha-định không thay đổi trong khoảng thời gian giữa hai symbol tín hiệu. Nhờ đó, chúng ta có thể bỏ qua chỉ số thời gian ở trong biểu diễn độ lợi kênh truyền. Đặt các độ lợi kênh truyền từ ăng-ten phát thứ nhất và thứ hai tới ăng-ten máy thu tương ứng là  $h_1 = \alpha_1 e^{j\phi_1}$  và  $h_2 = \alpha_2 e^{j\phi_2}$ . Các tín hiệu thu được tại các khe thời gian  $k$  và  $k + 1$  là

$$y_k = h_1 s_k + h_2 s_{k+1} + z_k \quad (6.69)$$

$$y_{k+1} = -h_1 s_{k+1}^* + h_2 s_k^* + z_{k+1}. \quad (6.70)$$

**Phương pháp kết hợp tín hiệu:** Biến đổi (6.69) và (6.70) lại thành

$$y_k = h_1 s_k + h_2 s_{k+1} + z_k \quad (6.71)$$

$$y_{k+1} = h_2 s_k^* - h_1 s_{k+1}^* + z_{k+1}. \quad (6.72)$$

Để ý rằng để ước lượng được các symbol tín hiệu  $s_k$  và  $s_{k+1}$  một cách tối ưu, chúng ta cần phải tách được thông tin về các symbol

này chứa trong các tín hiệu thu  $y_k$  và  $y_{k+1}$ . Việc này có thể thực hiện được nhờ sử dụng luật kết hợp sau đây [10]

$$\tilde{s}_k = h_1^* y_k + h_2 y_{k+1}^* \quad (6.73)$$

$$\tilde{s}_{k+1} = h_2^* y_k - h_1 y_{k+1}^*. \quad (6.74)$$

Thay (6.71) và (6.72) vào (6.73) và (6.74), chúng ta thu được

$$\tilde{s}_k = (\alpha_1^2 + \alpha_2^2) s_k + h_1^* z_k + h_2 z_{k+1}^* \quad (6.75)$$

$$\tilde{s}_{k+1} = (\alpha_1^2 + \alpha_2^2) s_{k+1} - h_1 z_{k+1}^* + h_2^* z_k. \quad (6.76)$$

So sánh (6.75) và (6.76) với (6.57), chúng ta có thể thấy rằng sự khác biệt duy nhất ở các symbol ước lượng được nhờ sử dụng phương pháp MRC và phương pháp STBC của Alamouti là sự quay pha của các thành phần tạp âm. Tuy nhiên, sự quay pha tạp âm này không làm suy giảm tỉ số SNR hiệu dụng đầu ra. Vì vậy, chúng ta có thể kết luận rằng *phương pháp phân tập phát không gian-thời gian sử dụng hai ăng-ten phát và một ăng-ten thu của Alamouti tương đương với phương pháp phân tập thu MRC sử dụng một ăng-ten phát và hai ăng-ten thu*. Tuy nhiên, quan sát này chỉ đúng cho trường hợp tổng công suất phát không bị chuẩn hóa sao cho công suất phát của cả hai phương pháp phân tập như nhau. Nếu tổng công suất của hai ăng-ten phát trong trường hợp phân tập phát của Alamouti được chuẩn hóa bằng công suất phát từ một ăng-ten trong trường hợp thu MRC thì các thành phần tín hiệu  $\tilde{s}_k$  và  $\tilde{s}_{k+1}$  chỉ bằng một nửa. Kết quả là, tỉ số SNR và phẩm chất BER đầu ra của hệ thống Alamouti STBC sẽ bị suy giảm đi 3 dB so với phẩm chất tương ứng của hệ thống phân tập thu MRC. Kết luận lại chúng ta có thể nói rằng *Alamouti STBC đạt được cấp độ phân tập giống như của phương pháp MRC nhưng bị suy giảm về phẩm chất 3dB*.

**Tách tín hiệu tối ưu:** luật quyết định ML được định nghĩa đồng thời cho cả  $s_k$  và  $s_{k+1}$  như sau

$$\begin{aligned} \{s_k, s_{k+1}\} = \arg \min_{s_k, s_{k+1} \in \chi_c} & \left\{ |y_k - (h_1 s_k + h_2 s_{k+1})|^2 \right. \\ & \left. + |y_{k+1} - (h_2 s_k^* - h_1 s_{k+1}^*)|^2 \right\}. \quad (6.77) \end{aligned}$$

Khai triển thành phần đối số của (6.77) ta có

$$\begin{aligned}
 A_{\text{STBC}} &= |y_k - (h_1 s_k + h_2 s_{k+1})|^2 + |y_{k+1} - (h_2 s_k^* - h_1 s_{k+1}^*)|^2 \\
 &= [y_k - (h_1 s_k + h_2 s_{k+1})][y_k - (h_1 s_k + h_2 s_{k+1})]^* \\
 &\quad + [y_{k+1} - (h_2 s_k^* - h_1 s_{k+1}^*)][y_{k+1} - (h_2 s_k^* - h_1 s_{k+1}^*)]^* \\
 &= |y_k|^2 - y_k h_1^* s_k^* - y_k h_2^* s_{k+1}^* - h_1 y_k^* s_k + |h_1|^2 |s_k|^2 + h_1 h_2^* s_k s_{k+1}^* \\
 &\quad - h_2 y_k^* s_{k+1} + h_2 s_{k+1} h_1^* s_k^* + |h_2|^2 |s_{k+1}|^2 + |y_{k+1}|^2 - y_{k+1} h_2^* s_k \\
 &\quad + y_{k+1} h_1^* s_{k+1} - h_2 s_k^* y_{k+1}^* + |h_2|^2 |s_k|^2 - h_2 s_k^* h_1^* s_{k+1} + h_1 s_{k+1}^* y_{k+1}^* \\
 &\quad - h_1 s_{k+1}^* h_2^* s_k + |h_1|^2 |s_{k+1}|^2. \tag{6.78}
 \end{aligned}$$

Để ý rằng  $|y_k|^2$  và  $|y_{k+1}|^2$  là các thành phần chung và không làm thay đổi luật quyết định, vì vậy chúng ta có thể bỏ qua chúng cho đơn giản. Ngoài ra, các thành phần  $h_1 h_2^* s_k s_{k+1}^*$  và  $h_2 h_1^* s_k^* s_{k+1}$  sẽ tự triệt tiêu lẫn nhau, do đó, đối số  $A_{\text{STBC}}$  trở thành

$$\begin{aligned}
 A_{\text{STBC}} &= (\alpha_1^2 + \alpha_2^2) |s_k|^2 + (\alpha_1^2 + \alpha_2^2) |s_{k+1}|^2 \\
 &\quad - y_k h_1^* s_k^* - y_k h_2^* s_{k+1}^* - h_1 y_k^* s_k - h_2 y_k^* s_{k+1} \\
 &\quad - y_{k+1} h_2^* s_k + y_{k+1} h_1^* s_{k+1} - h_2 y_{k+1}^* s_k^* + h_1 y_{k+1}^* s_{k+1}^* \\
 &= (\alpha_1^2 + \alpha_2^2) |s_k|^2 + (\alpha_1^2 + \alpha_2^2) |s_{k+1}|^2 \\
 &\quad - \underbrace{(h_1^* y_k + h_2 y_{k+1}^*)}_{\tilde{s}_k} s_k^* - \underbrace{(h_1 y_k^* + h_2^* y_{k+1})}_{\tilde{s}_k^*} s_k \\
 &\quad - \underbrace{(h_2 y_k^* - h_1^* y_{k+1})}_{\tilde{s}_{k+1}^*} s_{k+1} - \underbrace{(y_k h_2^* - h_1 y_{k+1}^*)}_{\tilde{s}_{k+1}} s_{k+1}^* \tag{6.79}
 \end{aligned}$$

và vì vậy

$$\begin{aligned}
 A_{\text{STBC}} &= (\alpha_1^2 + \alpha_2^2) |s_k|^2 + (\alpha_1^2 + \alpha_2^2) |s_{k+1}|^2 \\
 &\quad - \tilde{s}_k s_k^* - \tilde{s}_k^* s_k - \tilde{s}_{k+1} s_{k+1}^* - \tilde{s}_{k+1} s_{k+1}^* \\
 &= (\alpha_1^2 + \alpha_2^2) |s_k|^2 + |\tilde{s}_k - s_k|^2 - |\tilde{s}_k|^2 - |s_k|^2 \\
 &\quad + (\alpha_1^2 + \alpha_2^2) |s_{k+1}|^2 + |\tilde{s}_{k+1} - s_{k+1}|^2 - |\tilde{s}_{k+1}|^2 - |s_{k+1}|^2. \tag{6.80}
 \end{aligned}$$

Một điểm thú vị có thể nhận thấy rằng luật quyết định ML đồng thời cho cả  $s_k$  và  $s_{k+1}$  đã được chia thành các luật quyết định độc lập

cho  $s_k$  và  $s_{k+1}$ . Do các luật quyết định cho  $s_k$  và  $s_{k+1}$  là như nhau, sau khi bỏ qua thành phần chung  $|\tilde{s}_k|^2$  chúng ta thu được luật quyết định chung cho Alamouti STBC

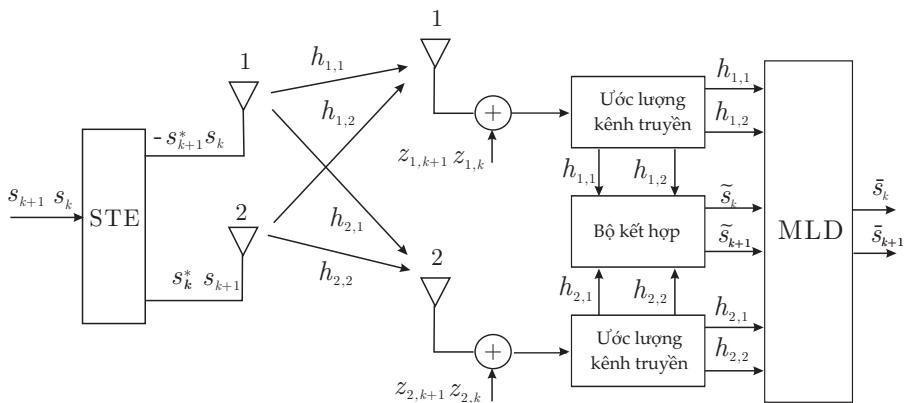
$$\bar{s}_k = \arg \min_{s_k \in \chi_c} \left\{ (\alpha_1^2 + \alpha_2^2 - 1) |s_k|^2 + |s_k - \tilde{s}_k|^2 \right\}, \quad (6.81)$$

giống như cho MRC ở công thức (6.65). Đổi với tín hiệu PSK, do năng lượng tín hiệu,  $E_s = |s_k|^2$ , như nhau đối với tất cả các thành phần tín hiệu  $s_k$ , luật quyết định ML lược giản thành giống như ở (6.66), tức là

$$\bar{s}_k = \arg \min_{s_k \in \chi_c} \left\{ |s_k - \tilde{s}_k|^2 \right\}. \quad (6.82)$$

### Trường hợp kênh MIMO $2 \times 2$

Trong phần này, chúng ta sẽ mở rộng nguyên lý của Alamouti STBC đối với trường hợp kênh MISO  $2 \times 1$  trong ở mục trước cho trường hợp kênh 2 ăng-ten phát và 2 ăng-ten thu, tức là, kênh MIMO  $2 \times 2$ . Sơ đồ cấu hình của Alamouti STBC cho trường hợp kênh MIMO  $2 \times 2$  được mô tả ở Hình 6.11.



Hình 6.11: Sơ đồ Alamouti STBC với 2 ăng-ten phát và 2 ăng-ten thu.

**Phương pháp kết hợp:** Các tín hiệu thu được tại các ăng-ten thu

thứ nhất và thứ hai tại các thời điểm  $k$  và  $k + 1$  là

$$y_{1,1} = h_{1,1}s_k + h_{1,2}s_{k+1} + z_{1,1} \quad (6.83)$$

$$y_{1,2} = -h_{1,1}s_{k+1}^* + h_{1,2}s_k^* + z_{1,2} \quad (6.84)$$

$$y_{2,1} = h_{2,1}s_k + h_{2,2}s_{k+1} + z_{2,1} \quad (6.85)$$

$$y_{2,2} = -h_{2,1}s_{k+1}^* + h_{2,2}s_k^* + z_{2,2} \quad (6.86)$$

trong đó  $z_{m,k}$  là các mẫu tạp âm ở tại ăng-ten thu  $m$  và khe thời gian  $k$ . Để ước lượng tối ưu được các dấu tín hiệu phát đi  $s_k$  và  $s_{k+1}$ , chúng ta cần phải tách được thông tin của chúng chứa trong  $y_{1,1}$ ,  $y_{1,2}$ ,  $y_{2,1}$  và  $y_{2,2}$ . Việc này có thể thực hiện được nhờ sử dụng phương pháp kết hợp sau đây [10]

$$\tilde{s}_k = h_{1,1}^*y_{1,1} + h_{1,2}y_{1,2}^* + h_{2,1}^*y_{2,1} + h_{2,2}y_{2,2}^* \quad (6.87)$$

$$\tilde{s}_{k+1} = h_{1,2}^*y_{1,1} - h_{1,1}y_{1,2}^* + h_{2,2}^*y_{2,1} - h_{2,1}y_{2,2}^*. \quad (6.88)$$

Thay các công thức (6.83)–(6.86) vào (6.87) và (6.88), chúng ta thu được

$$\begin{aligned} \tilde{s}_k &= (\alpha_{1,1}^2 + \alpha_{1,2}^2 + \alpha_{2,1}^2 + \alpha_{2,2}^2)s_k \\ &\quad + h_{1,1}^*z_{1,1} + h_{1,2}z_{1,2}^* + h_{2,1}^*z_{2,1} + h_{2,2}z_{2,2}^* \end{aligned} \quad (6.89)$$

$$\begin{aligned} \tilde{s}_{k+1} &= (\alpha_{1,1}^2 + \alpha_{1,2}^2 + \alpha_{2,1}^2 + \alpha_{2,2}^2)s_{k+1} \\ &\quad + h_{1,2}^*z_{1,1} - h_{1,1}z_{1,2}^* + h_{2,2}^*z_{2,1} - h_{2,1}z_{2,2}. \end{aligned} \quad (6.90)$$

Để ý rằng các tín hiệu kết hợp  $\tilde{s}_k$  và  $\tilde{s}_{k+1}$  ở các công thức trên thực tế là tổng của các tín hiệu kết hợp từ từng ăng-ten thu ở các công thức (6.75) và (6.76). Điều này có nghĩa là để có được tín hiệu tổng hợp cho trường hợp kênh MIMO  $2 \times M$ , chúng ta cần phải tìm được tín hiệu kết hợp cho từng ăng-ten thu, sau đó chỉ cần cộng chúng lại với nhau. Một quan sát khác cũng được nhận thấy là việc sử dụng 2 ăng-ten thu cho phép tăng gấp đôi bậc phân tập so với hệ thống sử dụng một ăng-ten thu. Một cách tổng quát, chúng ta có thể kết luận rằng bậc phân tập của một hệ thống MIMO STBC  $2 \times M$  gấp  $M$  lần bậc phân tập của một hệ thống MISO STBC  $2 \times 1$ . Điều này cũng có nghĩa rằng *bậc phân tập của một hệ thống MIMO STBC  $2 \times M$*

bằng bậc phân tập của một hệ thống SIMO MRC  $1 \times 2M$ . Tuy nhiên, cần chú ý rằng nếu như tổng công suất phát được chuẩn hóa thành đơn vị sao cho tổng công suất phát của cả hai hệ thống như nhau thì phẩm chất của các hệ thống STBC bị suy giảm 3 dB so với các hệ thống MRC. Trong các hệ thống thông tin vô tuyến việc chuẩn hóa công suất nhằm đảm bảo tổng công suất phát xạ từ một máy phát bị ràng buộc để đảm bảo không gây nhiễu đến các hệ thống khác hoặc không gây ảnh hưởng đến môi trường. Trong trường hợp máy phát sử dụng hai ăng-ten phát thì cần có 2 tuyền phát cao tần và công suất phát xạ bằng tổng công suất phát ra từ hai nhánh ăng-ten. Để đảm bảo điều kiện chuẩn hóa công suất, mỗi nhánh cao tần chỉ được phép phát  $1/2$  công suất cho phép.

**Tách tín hiệu tối ưu:** luật quyết định ML đồng thời cho  $s_k$  và  $s_{k+1}$  được định nghĩa như sau

$$\begin{aligned} \{\hat{s}_k, \hat{s}_{k+1}\} = \arg \min_{s_k, s_{k+1} \in \chi_c} & \left\{ |y_{1,k} - (h_{1,1}s_k + h_{1,2}s_{k+1})|^2 \right. \\ & + |y_{1,k+1} - (h_{1,1}s_k^* - h_{1,2}s_{k+1}^*)|^2 - |y_{2,k} - (h_{2,1}s_k + h_{2,2}s_{k+1})|^2 \\ & \left. + |y_{2,k+1} - (h_{2,1}s_k^* - h_{2,2}s_{k+1}^*)|^2 \right\}. \end{aligned} \quad (6.91)$$

Sử dụng phương pháp tương tự như đã sử dụng cho trường hợp kênh MISO  $2 \times 1$  trình bày ở mục trước chúng ta có thể thu được luật quyết định ML tổng quát như sau

$$\bar{s}_k = \arg \min_{s_k \in \chi_c} \left\{ (\alpha_{1,1}^2 + \alpha_{1,2}^2 + \alpha_{2,1}^2 + \alpha_{2,2}^2 - 1) |s_k|^2 + |s_k - \tilde{s}_k|^2 \right\}, \quad (6.92)$$

được lược giản thành

$$\bar{s}_k = \arg \min_{s_k \in \chi_c} \left\{ |s_k - \tilde{s}_k|^2 \right\}, \quad (6.93)$$

cho tín hiệu PSK.

## Chương trình mô phỏng

Chương trình mô phỏng mẫu các hệ thống Alamouti và SIMO MRC truyền tín hiệu BPSK qua kênh pha định Rayleigh phẳng, biến đổi chậm được trình bày ở chương trình MATLAB bpskAlamoutiMRC.m ở Matlab Program 6.2. Trong hệ thống Alamouti, do tổng công suất phát phải chia đều cho các ăng ten phát nên phương sai của các thành phần I và Q của nhiễu cộng được xác định theo công thức  $\sigma_n^2 = \frac{E_b}{2(Eb/N_0)/N} = \frac{NE_b}{2(Eb/N_0)}$ , với  $N$  là số lượng ăng ten phát. Như vậy, trong chương trình Matlab, nhiễu cộng của hệ thống Alamouti được tạo ra như sau:

```
sigma = sqrt(N/(2 * EbNo));
n = sigma * (randn(M, T, L) + j * randn(M, T, L));
```

Nếu  $b_1$  và  $b_2$  là 2 bit dữ liệu được phát đi trong mỗi khôi mã STBC của hệ thống Alamouti thì ta có thể tạo các khôi mã như sau:

$$S = b_1 A + b_2 B$$

với

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Để khôi phục được tín hiệu phát trong hệ thống Alamouti, ta cần xác định các giá trị  $\tilde{s}_k$  và  $\tilde{s}_{k+1}$  cho bởi (6.73)–(6.74) đối với hệ thống  $2 \times 1$  và (6.87)–(6.88) đối với hệ thống  $2 \times 2$ . Trong chương trình mô phỏng, để đẩy nhanh tốc độ tính toán, ta coi máy thu có khả năng dự đoán chính xác kênh truyền, rồi từ đó tính trước các ma trận kênh tương đương như sau:

```
for k2 = 1 : L
    gnHat(:,:,k2,1) = [gn(1,1,k2)' gn(1,2,k2); gn(1,2,k2)' - gn(1,1,k2)];
    gnHat(:,:,k2,2) = [gn(2,1,k2)' gn(2,2,k2); gn(2,2,k2)' - gn(2,1,k2)];
end
```

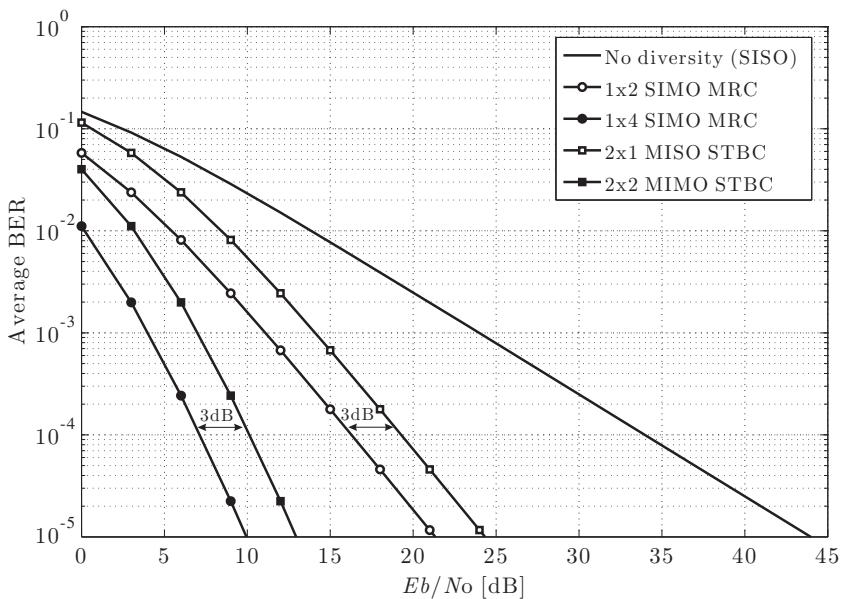
Và trong vòng lặp chính của chương trình Matlab, chúng ta tính  $\tilde{s}_k$  và  $\tilde{s}_{k+1}$  như sau:

```
S(:,:,1,k2) = g(:,:,1) * y(1,:).';
S(:,:,2,k2) = g(:,:,2) * y(2,:).';
⋮
S(:,:,2,:) = sum(S,2);
```

Kết quả BER của các hệ thống Alamouti  $2 \times 1$  và  $2 \times 2$  được so sánh với kết quả BER của các hệ thống SIMO MRC  $1 \times 2$  và  $1 \times 4$  như trình bày trên Hình 6.12. Theo hình vẽ, do tổng công suất phát ở các hệ thống STBC được chia đều cho hai ăng ten phát ( $N = 2$ ), các đường cong BER của các hệ thống STBC có cùng độ dốc các đường cong BER của các hệ thống SIMO MRC tương ứng nhưng dịch sang bên trái 3 dB. Điều này chứng tỏ rằng các hệ thống STBC và MRC có cùng cấp độ phân tập<sup>11</sup>.

---

<sup>11</sup>Một hệ thống MIMO với độ phân tách  $N \times M$  có BER suy giảm tỉ lệ nghịch theo bậc  $(N \times M)$  của tỉ số SNR trung bình.



**Hình 6.12:** Phẩm chất BER của các hệ thống Alamouti STBC so sánh với các hệ thống MRC.

## 6.4 Tóm tắt

Trong chương này, chúng ta đã tìm hiểu tổng quan về các phương pháp phân tập trong thông tin vô tuyến. Chúng ta đã tìm hiểu chi tiết về nguyên lý của hai phương pháp phân tập thu và phát điển hình, cụ thể là phương pháp thu phân tập MRC và mã không gian thời gian của Alamouti. Từ đó, chúng ta xây dựng các chương trình mô phỏng cho các hệ thống này. Thông qua kết quả mô phỏng, chúng ta đã thấy rằng một hệ thống MIMO STBC  $2 \times M$  có cùng độ phân tập với một hệ thống SIMO MRC  $1 \times 2M$ . Tuy nhiên, các hệ thống STBC bị suy giảm phẩm chất 3 dB về BER do điều kiện chuẩn hóa công suất phát.

Do phẩm chất hấp dẫn của STBC, máy thu trạm gốc có thể sử dụng 2 ăng-ten để thu được độ lợi phân tập cho cả kênh truyền lên và kênh truyền xuống. Vì vậy, mã khồi không gian-thời gian Alamouti đã được khuyến nghị sử dụng ở trong các hệ thống thông tin di động vô tuyến thế hệ thứ ba, LTE, Wi-Fi và WiMAX.

## Tài liệu tham khảo

- [1] W. C. Jakes, *Microwave Mobile Communications*. IEEE Press, 1974.
- [2] B. Vucetic and J. Yuan, *Space-Time Coding*. Wiley, 2003.
- [3] B. Sklar, "Rayleigh fading channels in mobile digital communication systems, Part II: Mitigation," *IEEE Communication Magazines*, vol. 35, no. 7, pp. 102–109, July 1997.
- [4] ——, "Rayleigh fading channels in mobile digital communication systems, Part I: Characterization," *IEEE Communication Magazine*, vol. 35, no. 7, pp. 90–100, July 1997.
- [5] M. Schwartz, W. R. Bennett, and S. Stein, *Communication Systems and Techniques*. McGraw-Hill, Inc., 1966.
- [6] L. R. Kahn, "Ratio squarer," *Proc. IRE*, vol. 47, p. 1704, November 1954.
- [7] J. Proakis, *Digital Communications*. McGraw-Hill, Inc., 2nd ed., 2001.
- [8] J. K. Carvers, "Single-user and multiuser adaptive maximal ratio transmission for Rayleigh channels," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 6, pp. 2043–2050, November 2000.
- [9] A. Wittneben, "A new bandwidth efficient transmit antenna modulation diversity scheme for linear digital modulation," in *IEEE International Conference on Communications (ICC)*, vol. 3, Geneva, Switzerland, May 1993, pp. 1630–1634.
- [10] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, 1998.
- [11] V. Tarokh, N. Seshadri, and A. Calderbank, "Space-time codes for high data rate wireless communication: performance criteria and code construction," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 744–765, March 1998.

## Phụ lục: Chương trình MATLAB mẫu

### ■ MATLAB Program 6.1 combSNRcdf.m

```
% Chuong trinh ve CDF (outage probability) cua SNR cho cac phuong
% phap ket hop phan tap thu khac nha
% Written by Tran Xuan Nam, The University of Electro-Communications.
% Date: 18/8/2005

clear all
close all
% Define parameters
M = [1, 2, 4, 6, 8]; % Number of diversity branches (Rx antennas)
gOverGdB = -20:10;    % Ratio of gamma/Gamma in dB
gOverG = 10.^ (gOverGdB./10);
G = 1;
g = 10.^ (gOverGdB./10);

% CDF of SNR for selection combining
for ii=1:length(M)
    Psc(ii,:)=(1-exp(-gOverG)).^M(ii);
end

figure(1)
plot(gOverGdB,Psc(1,:),'-^',gOverGdB,Psc(2,:),'-o',gOverGdB, ...
      Psc(3,:),'-d',gOverGdB,Psc(4,:),'-s',gOverGdB,Psc(5,:),'->')
axis([-20 10 0 1])
xlabel('\gamma/\Gamma [dB]')
ylabel('Pr[\gamma<x]')
legend('M=1','M=2','M=4','M=6','M=8','Location','NorthWest')
title('CDF of SNR for selection combining')

% CDF of maximal ratio combining
for ii=1:length(M)
    for m=1:M(ii)
        SumM(m,:)=gOverG.^ (m-1)/factorial(m-1);
```

```

    end
    Pmrc(ii,:)=1-exp(-gOverG).*sum(SumM,1);
end
figure(2)
plot(gOverGdB,Pmrc(1,:),'-^',gOverGdB,Pmrc(2,:),'-o',gOverGdB, ...
      Pmrc(3,:),'-d',gOverGdB,Pmrc(4,:),'-s',gOverGdB,Pmrc(5,:),'->')
axis([-20 10 0 1])
xlabel('gamma/\Gamma [dB]')
ylabel('Pr[gamma<x]')
legend('M=1','M=2','M=4','M=6','M=8','Location','NorthWest')
title('CDF of SNR for maximal-ratio combining')

% Diversity gain comparison
M=1:10;
for i=1:length(M)
    sel(i)=10*log10(sum(1./M(1:i))); % Diversity gain of SC
    Geg(i)=10*log10(1+(M(i)-1)*pi/4);
    Gmrc(i)=10*log10(M(i));
end
figure(4)
plot(M,Gsel,'-s',M,Geg,'-^', M,Gmrc,'-o')
xlabel('Diversity Branches M','interpreter','latex')
ylabel('Diversity gain $\bar{\Gamma}/\Gamma$ [dB]',...
      'interpreter','latex')
legend('Selection Combining', 'Equal Gain Combining',...
      'Maximum Ratio Combining',...
      'Location','NorthWest')

```

### MATLAB Program 6.2 bpskAlamoutiMRC.m

```

% Chuong trinh mo phong Monte-Carlo cac he thong Alamouti
% va MRC trong kenh pha dinh Rayleigh bien doi cham
% Written by: Tran Xuan Nam, Le Qui Don Technical University
%             Le Minh Tuan, UVISION JSC.

% Lats updated 06/07/2011

```

```
clc;
clear all
% Dinh nghia tham so
EbNodB=0:3:21;
EbNo=10.^{EbNodB./10};
sigLen=[1e4 3e4 10e4 80e4 100e4 250e4 250e4 250e4]./10;
Loop=[1 1 1 1 6 12 12 12];
N=2; %So luong anten phat cua he thong Alamouti
M=2; %So luong anten thu cua he thong Alamouti
T=2; %Do dai cua mot khoi ma Alamouti
%So luong cac ma tran kenh va nhieu duoc tao san
L=100000;

%Cac ma tran de tao khoi ma Alamouti
A=[1 0;0 1];
B=[0 -1;1 0];
% Mat do pho AWGN
sigma=sqrt(N./(2.*EbNo));
%Cac ma tran he so pha dinh
gn=1/sqrt(2)*(randn(M,N,L)+j*randn(M,N,L));
%Tao ma tran kenh tuong duong cho qua trinh
%khoi phuc tin hieu thu duoc
gnHat=zeros(2,2,L,2);
for k2=1:L
    gnHat(:,:,:,k2,1)=[gn(1,1,k2)' gn(1,2,k2); gn(1,2,k2)' -gn(1,1,k2)];
    gnHat(:,:,:,k2,2)=[gn(2,1,k2)' gn(2,2,k2); gn(2,2,k2)' -gn(2,1,k2)];
end

% Vong lap tinh toan BER theo Eb/No
for k=1:length(EbNo)
    for k3=1:Loop(k)
        %Tao du lieu phat
        data=randn(T,sigLen(k))>0;
        b=data.*2-1;
```

```

% Tao AWGN voi variance sigma^2
n=sigma(k)*(randn(M,T,L)+j*randn(M,T,L));

S=zeros(2,2,sigLen(k));
Sig=zeros(N,T);
y=zeros(M,T);
for k2=1:sigLen(k)
    %Cac khoi ma Alamouti phat di
    Sig=b(1,k2).*A+b(2,k2).*B;
    ktem=mod(k2,L)+1;
    % Tin hieu thu cho M anten
    y=gn(:,:,ktem)*Sig+n(:,:,ktem);
    y(:,2)=conj(y(:,2));
    g=gnHat(:,:,ktem,:);
    %Ket hop tin hieu thu cho anten thu 1
    S(:,:,k2)=g(:,:,1)*y(1,:).';
    %Ket hop tin hieu thu cho anten thu 2
    S(:,:,k2)=g(:,:,2)*y(2,:).';
end
S(:,:,2)=sum(S,2);
bHat=real(S)>0;
% Tinh toan so luong bit loi va BER
temp=repmat(reshape(data, 2,1,sigLen(k)),1,2);
noBitErr(:,:,k3)=sum(sum(abs(bHat-temp)),3)';
end
ABER(:,:,k)=sum(noBitErr,2)/sigLen(k)/Loop(k)/T;
end

%BER cua he thong BPSK trong kenh Rayleigh
EbNodB1=0:3:45;
EbNo1=10.^-(EbNodB1./10);
[BER1]=bpskFuncMRC(EbNo1, 1);
%BER cua he thong BPSK with MRC, M=2
[BER2]=bpskFuncMRC(EbNo, 2);
%BER cua he thong BPSK with MRC, M=4

```

```
[BER3]=bpskFuncMRC(EbNo, 4);

% Ve do thi
semilogy(EbNodB1,BER1(1,:),'-o');
hold on
semilogy(EbNodB,BER2,'r-*',EbNodB,BER3,'r-p')
semilogy(EbNodB,ABER(1,:),'g^-',EbNodB,ABER(2,:),'g-+')

 xlabel('E_b/N_o (dB)')
 ylabel('BER')
 axis([0 45 1e-5 1])
 grid on
 legend('No diversity SISO', '1x2 SIMO MRC',...
 '1x4 SIMO MRC', '2x1 MISO STBC', '2x2 MISO STBC')
```

### **Function 6.1** bpskFuncMRC.m

```
function [BER]=bpskFuncMRC(EbNo, M)
% Ham mo phong Monte-Carlo truyen dan BPSK voi MRC
% qua kenh pha dinh Rayleigh
% Written by: Tran Xuan Nam, Le Qui Don Technical University
% Le Minh Tuan, UVISION JSC.
% Last updated 24/06/2011

% Dinh nghia tham so
sigLen=10e3;
Loop=M*400;
% Mat do pho AWGN
sigma=sqrt(1./(2.*EbNo));

% Vong lap tinh toan BER theo Eb/No
for k=1:length(EbNo)
    gr=randn(M,sigLen);
    nr=randn(M,sigLen);
    nerror=zeros(Loop,1);
    %
```

```
for k2=1:Loop
    % Tao chuoi bit du lieu phat
    b=randn(1,sigLen)>0;
    %Tin hieu phat
    Sig=b*2-1;
    Sig=Sig(ones(M,1),:);
    % Tao AWGN voi variance sigma^2
    n=sigma(k)*(nr+j*randn(M,sigLen));
    %He so pha dinh
    g=1/sqrt(2)*(gr+j*randn(M,sigLen));
    % Tin hieu thu cho M anten
    yk=Sig.*g+n;
    %Thuc hien MRC va khai phuc tin hieu
    ykHat=real(yk.*conj(g));
    bHat=sum(ykHat,1)>0;
    nerror(k2) = sum(abs(bHat-b));
end
%Tinh toan so bit loi
nBitErr(k)=sum(nerror);
end


---


```



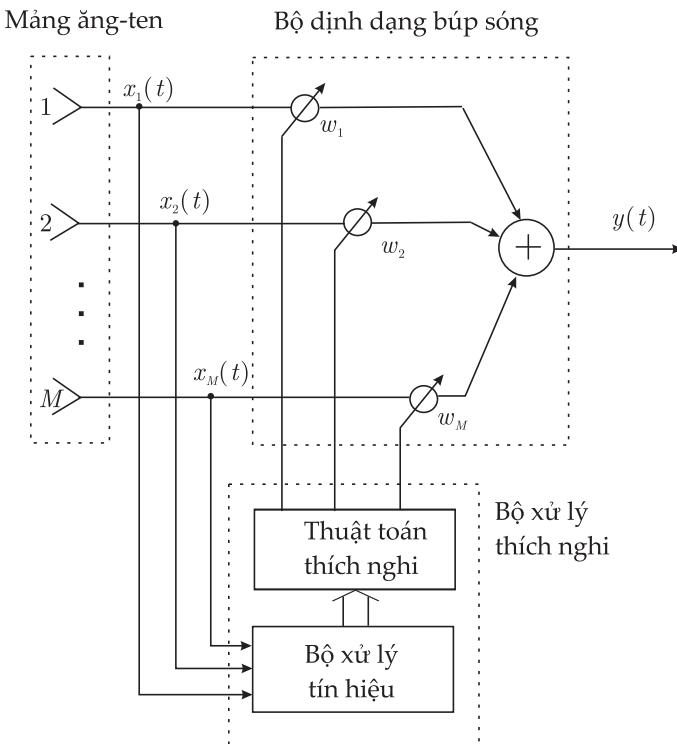
## Chương 7

# Ăng-ten thích nghi

### 7.1 Các khái niệm cơ bản

Ăng-ten *thích nghi* là một hệ thống bao gồm một mảng (dàn) các phần tử ăng-ten và một bộ xử lý tín hiệu thích nghi theo thời gian thực có nhiệm vụ điều khiển khôi tạo búp sóng (beamforming) tự động điều chỉnh các trọng số điều khiển của nó nhằm tối ưu hóa một tiêu chuẩn nhất định tuân theo một thuật toán chọn trước [1]–[2]. Đôi khi, ăng-ten thích nghi cũng còn được gọi là *ăng-ten mảng* (array antenna) hay *ăng-ten thông minh* (smart antenna). Cấu hình điển hình của một ăng-ten thích nghi được mô tả ở Hình 7.1.

Các phần tử ăng-ten trong mảng có thể sắp xếp theo nhiều cấu hình khác nhau, trong đó phổ biến nhất là cấu hình đường thẳng (tuyến tính), tròn, hay mặt phẳng như biểu diễn trên Hình 7.2. Một *mảng tuyến tính* (linear array) gồm các phần tử mảng có trung tâm đặt dọc theo một đường thẳng. Nếu khoảng cách giữa các phần tử kề nhau của mảng bằng nhau, thì ăng-ten mảng đó được gọi là *mảng tuyến tính cách đều* (USLA: Uniformly Spaced Linear Array). Tương tự, một *mảng hình tròn* (circular array) chứa các phần tử mảng có trung tâm đặt trên một đường tròn. Cuối cùng, một *mảng hình phẳng* (planar array) chứa các phần tử của mảng có tâm nằm trên cùng một mặt phẳng. Trong khi cả hai loại mảng tuyến tính và mảng hình



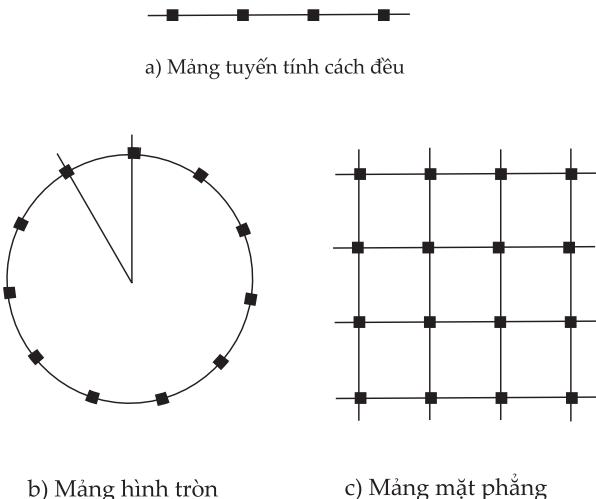
**Hình 7.1:** Cấu hình một ăng-ten thích nghi với  $M$  phần tử.

tròn chỉ có thể thực hiện tạo búp sóng một chiều trên mặt phẳng nằm ngang, thì mảng hình phẳng có thể sử dụng để tạo búp sóng trên cả hai chiều thẳng đứng và nằm ngang.

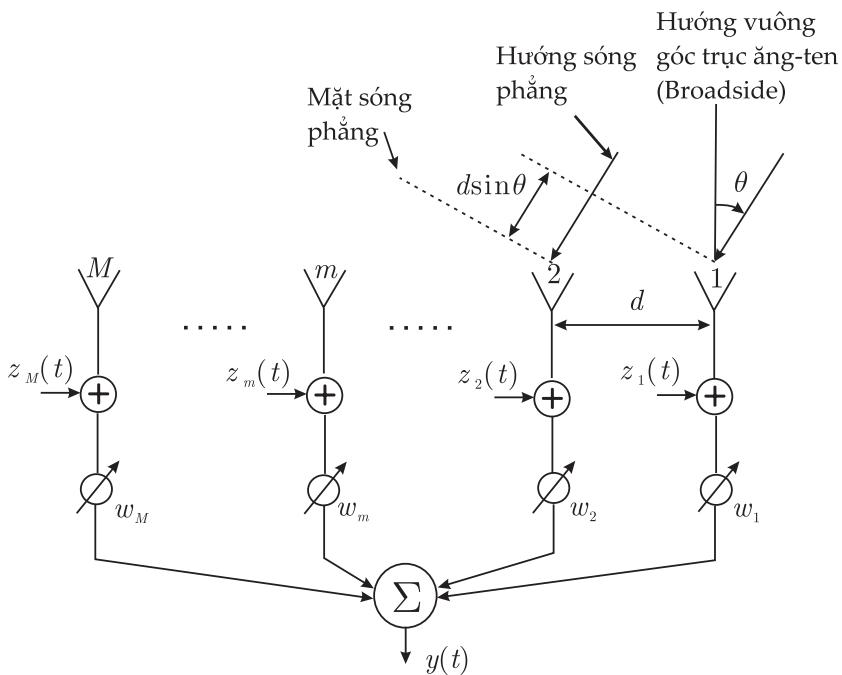
Mặc dù, cấu hình hình học có thể khác nhau, nhưng nguyên lý hoạt động của các ăng-ten mảng đều tương tự nhau. Vì vậy, để đơn giản hóa cho các phân tích toán học, mảng tuyến tính cách đều thường được sử dụng để mô tả nguyên lý hoạt động. Các mô hình toán học sử dụng cho mảng tuyến tính trình bày sau đây hoàn toàn có thể mở rộng dễ dàng cho các cấu hình khác.

## 7.2 Mô hình tín hiệu mảng ăng-ten

Xét một ăng-ten USLA có  $M$  phần tử như mô tả ở Hình 7.3, trong đó  $d$  là khoảng cách giữa các phần tử kề nhau.



**Hình 7.2:** Các cấu hình mảng khác nhau của ăng-ten thích nghi.



**Hình 7.3:** Mô hình tín hiệu của ăng-ten thích nghi.

Giả thiết mặt phẳng sóng tới mảng từ một hướng với góc tới  $\theta$  so với hướng bức xạ của mảng (array broadside). Góc  $\theta$ , đo theo chiều quay của kim đồng hồ từ hướng bức xạ của mảng được gọi

là *hướng tới* (DOA: Direction of Arrival) hoặc *góc tới* (AOA: Angle of Arrival) của tín hiệu thu. Nếu tín hiệu tới  $s(t)$  được phát đi từ một nguồn trường xa (far-field source) và đã lan truyền qua một môi trường không phân tán (non-dispersive) chỉ gây nên trễ truyền sóng, thì mặt sóng (wavefront) tại phần thứ  $(m+1)$  bị trễ hơn so với tại phần tử thứ  $m$  kề trước nó một khoảng  $d \sin \theta$ . Nếu chọn phần tử thứ nhất làm phần tử tham chiếu và ký hiệu tín hiệu tại phần tử tham chiếu là  $s(t)$ , thì độ trễ pha của tín hiệu tại phần tử thứ  $m$  so với phần tử thứ nhất là  $(m-1)kd \sin \theta$ , trong đó  $k = \frac{2\pi}{\lambda}$  là hằng số sóng (wave number) và  $\lambda$  là bước sóng. Kết quả là tín hiệu thu được tại phần tử thứ  $m$  ký hiệu là  $x_m(t)$  được cho bởi

$$x_m(t) = s(t)e^{-j\frac{2\pi}{\lambda}(m-1)d \sin \theta} \quad (7.1)$$

trong đó  $j = \sqrt{-1}$  ký hiệu số ảo và  $m = 1, 2, \dots, M$ .

Sắp xếp các tín hiệu  $x_m(t)$  chúng ta có véc-tơ thu sau

$$\mathbf{x}(t) = [x_1(t) \quad x_2(t) \quad \dots \quad x_M(t)]^T. \quad (7.2)$$

Ký hiệu

$$\mathbf{a}(\theta) = [1 \quad e^{-j\frac{2\pi}{\lambda}d \sin \theta} \quad \dots \quad e^{-j\frac{2\pi}{\lambda}(M-1)d \sin \theta}]^T \quad (7.3)$$

ta có biểu thức (7.2) được biểu diễn ở dạng sau

$$\mathbf{x}(t) = s(t)\mathbf{a}(\theta). \quad (7.4)$$

Véc-tơ  $\mathbf{x}(t)$  được gọi là véc-tơ dữ liệu đầu vào và  $\mathbf{a}(\theta)$  được gọi là véc-tơ đáp ứng của mảng (array response) hoặc là véc-tơ lái (steering véc-tơ). Véc-tơ đáp ứng mảng trong trường hợp này chỉ phụ thuộc vào góc tới  $\theta$ . Trong trường hợp tổng quát véc-tơ đáp ứng của mảng cũng có thể phụ thuộc vào các đáp ứng của từng phần tử, cấu hình hình học của mảng, và tần số tín hiệu. Tập hợp các véc-tơ đáp ứng của mảng theo tất cả các hướng và tần số thường được gọi là đa tạp của mảng (array manifold). Đối với các mảng đơn giản như mảng tuyến tính cách đều đang xét ở đây thì đa tạp mảng có thể tính toán

bằng giải tích. Tuy nhiên, trong thực tế nó có thể được đo từ các đáp ứng nguồn điểm (point source responses) trên tất cả các hướng tới và tại tất cả các tần số. Quá trình tìm đa tạp của mảng được gọi là hiệu chuẩn mảng (array calibration).

Xét trường hợp có ảnh hưởng của tạp âm tại các phần tử, véc-tơ dữ liệu đầu vào trở thành

$$\mathbf{x}(t) = s(t)\mathbf{a}(\theta) + \mathbf{z}(t), \quad (7.5)$$

trong đó véc-tơ tạp âm  $\mathbf{z}(t)$  được định nghĩa như sau

$$\mathbf{z}(t) = [z_1(t) \ z_2(t) \ \dots \ z_M(t)]^T. \quad (7.6)$$

Công thức (7.1) đúng cho các tín hiệu có băng thông nhỏ hơn rất nhiều so với nghịch đảo của thời gian truyền sóng qua mảng. Tín hiệu thỏa mãn điều kiện này được gọi là tín hiệu băng hẹp, trường hợp ngược lại gọi là băng rộng.

Xét mô hình tín hiệu mảng cho trường hợp tổng quát chịu ảnh hưởng của hiệu ứng truyền sóng đa đường và ảnh hưởng của nhiều đồng kênh. Ký hiệu  $U$  là tổng số người dùng có tín hiệu đến mảng và giả thiết rằng tín hiệu tới của người dùng thứ  $i$ , là  $s_i(t)$ , chứa  $P_i$  tia đa đường với biên độ phức  $\alpha_{i,p}$ , góc tới  $\theta_{i,p}$  và trễ lớn nhất  $\tau_{i,p}$ , trong đó  $p$  là chỉ số của tia đa đường và  $p = 1, 2, \dots, P_i$ . véc-tơ tín hiệu thu tương ứng với người dùng thứ  $i$  có thể biểu diễn như sau

$$\mathbf{x}_i(t) = \sum_{p=1}^{P_i} \alpha_{i,p} \mathbf{a}(\theta_{i,p}) s_{i,p}(t - \tau_{i,p}). \quad (7.7)$$

Xét ảnh hưởng của toàn bộ  $U$  người dùng và tạp âm, véc-tơ dữ liệu đầu vào có thể được tổng quát hóa thành

$$\mathbf{x}(t) = \sum_{i=1}^U \sum_{p=1}^{P_i} \alpha_{i,p} \mathbf{a}(\theta_{i,p}) s_{i,p}(t - \tau_{i,p}) + \mathbf{z}(t). \quad (7.8)$$

Trong các công thức (7.7) và (7.8), thành phần  $\sum_{p=1}^{P_i} \alpha_{i,p} \mathbf{a}(\theta_{i,p})$  được gọi là *véc-tơ chữ ký không gian* (spatial signature véc-tơ) của người dùng thứ  $i$ .

### 7.3 Nguyên lý tạo búp sóng thích nghi

Tạo búp sóng (beamforming) là một kiểu xử lý tín hiệu được sử dụng để tạo búp sóng của mảng về hướng nguồn tín hiệu mong muốn trong khi đồng thời lại tạo nên các điểm không (nulls) về phía các nguồn nhiễu. Quá trình phân tách người dùng mong muốn ra khỏi các nguồn nhiễu này dựa trên các đặc tính không gian được gọi là *lọc không gian* (spatial filtering). Trên đường lên (từ máy di động đến trạm gốc), mục đích của tạo búp sóng là cực đại hóa tỉ số tín hiệu trên nhiễu cộng tạp âm (SINR: Signal to Interference plus Noise Ratio) của tín hiệu mong muốn. Tương tự, tạo búp sóng được sử dụng ở đường xuống (downlink) (từ trạm gốc tới máy di động) để cực đại hóa công suất phát của trạm gốc tới một máy di động mong muốn, nhờ vậy cực đại hóa SINR của đường xuống. Khi tạo búp sóng được điều khiển sử dụng xử lý tín hiệu thích nghi, nó được gọi là *tạo búp sóng thích nghi* (adaptive beamforming). Trong một số trường hợp, chúng ta mong muốn quay các điểm không (nulls) của mảng về một hướng cho trước để triệt nhiễu. Quá trình tạo búp sóng trong các trường hợp đó được gọi là *tạo điểm không* (null-forming).

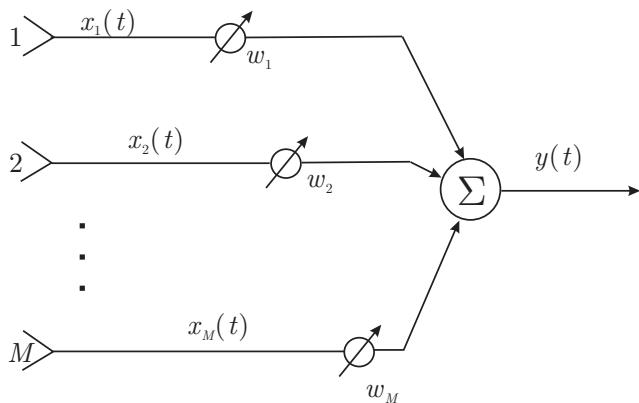
Một bộ tạo búp sóng (beamformer) là một bộ xử lý sử dụng kết hợp với một ăng-ten mảng để thực hiện nhiều dạng lọc không gian khác nhau [3]. Có hai loại bộ tạo búp sóng là *bộ tạo búp sóng băng hẹp* (narrowband beamformer) và *bộ tạo búp sóng băng rộng* (broadband beamformer).

Một bộ tạo búp sóng băng hẹp lấy mẫu tín hiệu đầu vào trên miền không gian và thường sử dụng để xử lý các tín hiệu băng hẹp. Cấu hình của một bộ tạo búp sóng băng hẹp được mô tả ở Hình 7.4.

Đầu ra của bộ tạo búp sóng băng hẹp là kết hợp tuyến tính có trọng số của các tín hiệu thu tại từng phần tử mảng, và được cho bởi biểu thức sau

$$y(t) = \mathbf{w}^H \mathbf{x}(t), \quad (7.9)$$

trong đó  $\mathbf{x}(t)$  là véc-tơ dữ liệu đầu vào và véc-tơ trọng số phức  $\mathbf{w}$



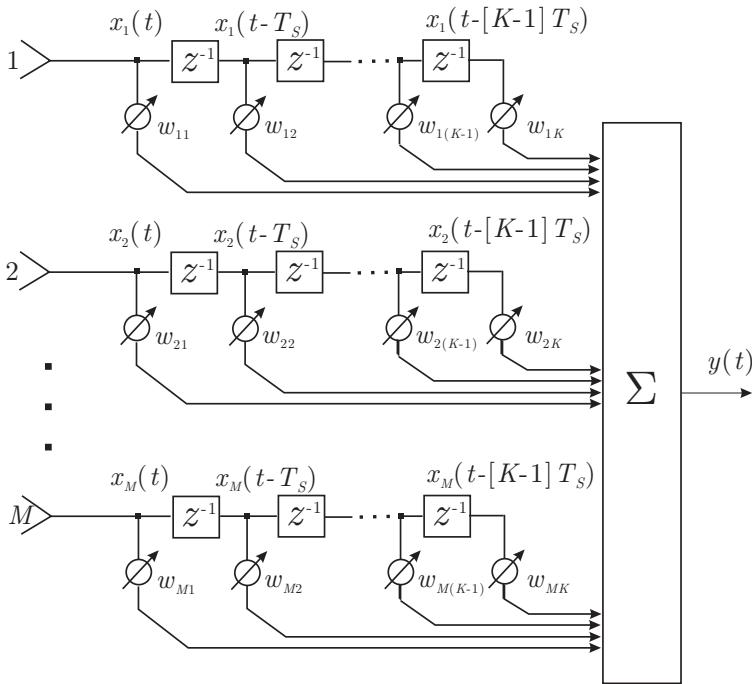
**Hình 7.4:** Cấu hình một bộ tạo búp sóng băng hẹp.

được định nghĩa như sau

$$\mathbf{w} = [w_1 \quad w_2 \quad \dots \quad w_M]^T. \quad (7.10)$$

Khác so với bộ tệp búp sóng băng hẹp, một bộ tạo búp sóng băng rộng lấy mẫu tín hiệu đầu vào trên cả hai miền không gian và thời gian và được sử dụng để xử lý tín hiệu băng rộng. Một bộ tạo búp sóng băng rộng còn được gọi là *bộ xử lý không gian thời gian* (*spatio-temporal processor*) hay *bộ san bằng không gian thời gian* (*spatio-temporal equalizer*). Cấu trúc của một bộ tạo búp sóng băng rộng thường gồm các dây giữ chậm (TDL: Tapped Delay Lines) ở các phần tử mảng. Nếu khoảng cách giữa các khâu giữ chậm (tập)  $T_s$  đủ dài và số khâu đủ lớn, thì TDL có thể coi như một bộ lọc lý tưởng cho phép điều khiển chính xác độ lợi và pha tại mỗi tần số trong băng tần quan tâm [4]. Dây giữ chậm TDL không chỉ có ý nghĩa đối với việc điều chỉnh độ lợi và pha trên băng tần mong muốn đối với tín hiệu băng rộng mà còn thích hợp cho các mục đích khác như hạn chế ảnh hưởng của pha-định đa đường, bù ảnh hưởng của trễ truyền sóng và mất phôi hợp giữa các kênh [4]. Một bộ tạo búp sóng băng rộng điển hình sử dụng TDL được mô tả ở Hình 7.5.

Để mô hình hóa bộ tạo búp sóng băng rộng, thực hiện sắp xếp



**Hình 7.5:** Cấu hình bộ tạo búp sóng sử dụng các dây giữ chậm.

các tín hiệu và trọng số phức tại  $K$  khâu TDL của ăng-ten  $m$  như sau

$$\dot{\mathbf{x}}_m(t) = \begin{bmatrix} x_m(t) & x_m(t - T_s) & \dots & x_m(t - [K-1]T_s) \end{bmatrix}^T, \quad (7.11)$$

$$\dot{\mathbf{w}}_m = \begin{bmatrix} w_{m1} & w_{m2} & \dots & w_{mK} \end{bmatrix}^T, \quad (7.12)$$

và định nghĩa

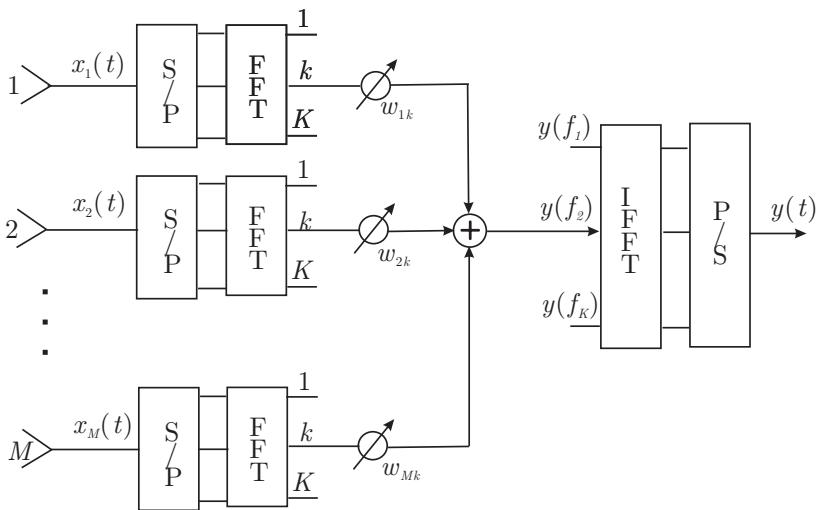
$$\mathbf{x}(t) = \begin{bmatrix} \dot{\mathbf{x}}_1^T(t) & \dot{\mathbf{x}}_2^T(t) & \dots & \dot{\mathbf{x}}_M^T(t) \end{bmatrix}^T, \quad (7.13)$$

$$\mathbf{w} = \begin{bmatrix} \dot{\mathbf{w}}_1^T & \dot{\mathbf{w}}_2^T & \dots & \dot{\mathbf{w}}_M^T \end{bmatrix}^T. \quad (7.14)$$

Nhờ vậy, đầu ra của bộ tạo búp sóng băng rộng có thể biểu diễn ở dạng giống như bộ tạo búp sóng băng hẹp ở (7.9), tức là

$$y(t) = \mathbf{w}^H \mathbf{x}(t). \quad (7.15)$$

Bộ tạo búp sóng băng rộng sử dụng TDL xét ở trên là một bộ xử lý tín hiệu trên miền không gian và thời gian cổ điển. Gần đây, biến



S/P: serial-to-parallel conversion

P/S: parallel-to-serial conversion

**Hình 7.6:** Bộ tạo búp sóng trên miền tần số sử dụng FFT. Ví dụ mô tả cho trường hợp xử lý tín hiệu tại một băng tần con  $k$ .

đổi Fourier nhanh (FFT: Fast Fourier Transform) cũng đã được sử dụng để thay thế cho các dây giũ chậm TDL ở các bộ tạo búp sóng băng rộng, tạo nên một bộ tạo búp sóng băng rộng tương đương trên hai miền không gian và miền tần số như ở Hình 7.6 [3][5].

Ưu điểm của giải pháp tạo búp sóng trên miền tần số là tăng tốc độ hội tụ và giảm độ phức tạp tính toán. Do các trọng số điều khiển được tính toán một cách độc lập cho từng băng con (subband), quá trình lựa chọn các trọng số có thể được thực hiện song song, giúp cho quá trình cập nhật trọng số diễn ra nhanh hơn [5]. Ngoài ra, khi các thuật toán thích nghi như bình phương trung bình tối thiểu (LMS: Least Mean Squares) được sử dụng, có thể áp dụng các bước nhảy (step size) khác nhau cho từng băng con, vì vậy, giúp cho quá trình hội tụ của véc-tơ trọng số diễn ra nhanh hơn [5]. Các bộ tạo búp sóng băng rộng trên miền tần số còn được gọi là mảng thích nghi băng con (SBAA: Subband Adaptive Array). Chi tiết thêm về nguyên lý hoạt động và các cấu hình của SBAA có thể tham khảo

thêm ở [6].

## 7.4 Đặc tính búp sóng của ăng-ten thích nghi

Đặc tính búp sóng của ăng-ten thích nghi thường được mô tả bởi hệ số mảng (array factor). Hệ số mảng biểu diễn mẫu phát xạ trường xa của các phần tử phát xạ đẳng hướng. Hệ số mảng cho biết đặc tính phát xạ của mảng theo cả góc tà  $\phi$  và góc phương vị  $\theta$ , tức là  $F(\phi, \theta)$ .

Đối với mảng tuyến tính cách đều, hệ số mảng được định nghĩa là đầu ra của mảng

$$F(\theta) = \mathbf{w}^T \mathbf{a} = w_1 + w_2 e^{-j \frac{2\pi}{\lambda} d \sin \theta} + \dots + w_M e^{-j \frac{2\pi}{\lambda} (M-1) d \sin \theta}. \quad (7.16)$$

Nếu trọng số  $w_k$  được lựa chọn sao cho

$$w_m = A_k e^{-j(m-1)\alpha}, \quad (7.17)$$

trong đó  $\alpha$  là sự lệch pha giữa phần tử thứ  $m$  và  $m-1$ , thì hệ số mảng trở thành

$$F(\theta) = \sum_{m=1}^M w_m e^{-j[\frac{2\pi}{\lambda}(m-1)d \sin \theta + (m-1)\alpha]}. \quad (7.18)$$

Nếu góc lệch pha được chọn thỏa mãn  $\alpha = -\frac{2\pi}{\lambda} d \sin \theta_0$ , tức là

$$\mathbf{w} = \mathbf{a}^*(\theta_0), \quad (7.19)$$

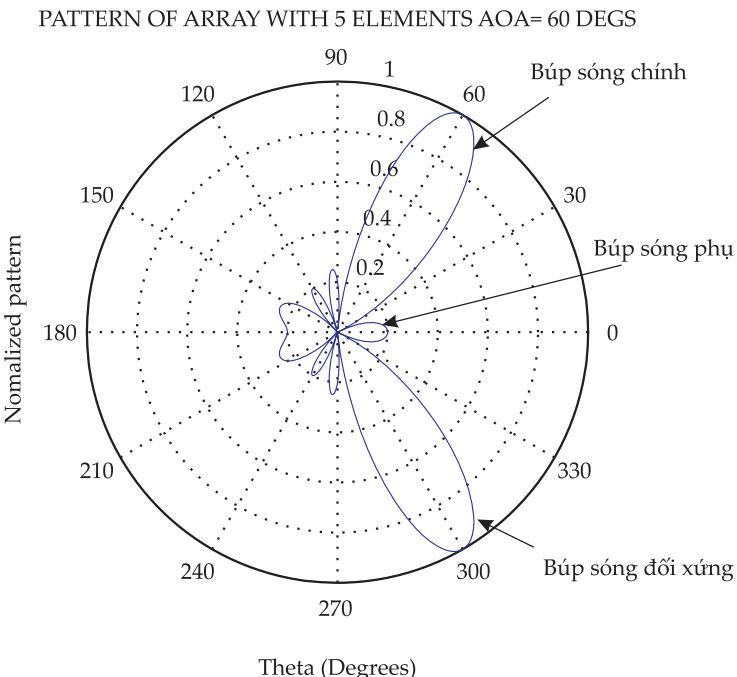
thì hệ số mảng sẽ đạt được cực đại theo hướng  $\theta_0$ . Tức là búp sóng của mảng đã được hướng về phía nguồn phát xạ  $\theta_0$ . Trường hợp này hệ số

$$\rho(\theta) = \frac{\mathbf{w}^H \mathbf{a}}{\|\mathbf{w}\| \|\mathbf{a}\|} = \frac{\mathbf{a}^H(\theta_0) \mathbf{a}(\theta)}{M} \quad (7.20)$$

được gọi là búp sóng chuẩn hóa với biên độ cực đại bằng 1.

Hình 7.7 minh họa búp sóng của một dàn tuyến tính cách đều nửa bước sóng với  $M = 5$  phần tử và góc phát xạ của nguồn từ hướng  $60^\circ$ . Có thể nhận thấy từ hình vẽ là ngoài búp sóng chính và

búp sóng phụ, búp sóng của mảng còn chứa một búp sóng đối xứng (grating lobe) qua trục của mảng (hướng  $0^\circ$ ). Nguyên nhân của búp sóng đối xứng là do các phần tử đặt cách xa nhau.



**Hình 7.7:** Búp sóng chuẩn hóa của dàn tuyền tính cách đều với  $M = 5$  phần tử và hướng phát xạ  $\theta = 60^\circ$ .

Chương trình `simArrayPattern.m` mô phỏng tính chất thích nghi của mảng ăng-ten thích nghi khi hướng của nguồn phát xạ di chuyển theo các hướng khác nhau.

## 7.5 Tiêu chuẩn tối ưu phẩm chất

Như đã đề cập trên đây, bộ xử lý thích nghi điều khiển bộ tạo búp sóng để tối ưu hóa các trọng số tạo búp sóng tương ứng với một tiêu chuẩn nhất định. Có bốn (4) tiêu chuẩn thường được sử dụng để tìm trọng số tối ưu cho ăng-ten mảng thích nghi là Sai số Bình phương Trung bình Cực tiểu (MMSE: Minimum Mean Square Error), Tỉ số Tín hiệu trên Nhiều cộng Tạp âm Cực đại (MSINR: Max-

imum Signal to Interference plus Noise ratio), Phương sai Cực tiểu (MV: Minimum Variance) và Hợp lệ Cực đại (ML: Maximum Likelihood). Các tiêu chuẩn tối ưu này sẽ được trình bày tóm tắt dưới đây.

### 7.5.1 Tiêu chuẩn MMSE

Tiêu chuẩn MMSE được Widrow và cộng sự đề xuất đầu tiên ở [1]. Tiêu chuẩn này tối thiểu hóa sai số bình phương trung bình giữa tín hiệu đầu ra của mảng  $y(t)$  và tín hiệu mong muốn  $s(t)$ . Trong thực tế, tín hiệu mong muốn  $s(t)$  không biết được trước ở máy thu. Tuy nhiên, sử dụng một số kỹ thuật như huấn luyện (training) hay ước lượng dựa trên các đặc tính của tín hiệu mong muốn có thể tạo ra một *tín hiệu tham chiếu* (*reference signal*)  $r(t)$  gần giống với tín hiệu mong muốn. Xét véc-tơ tín hiệu thu sau đây

$$\begin{aligned} \mathbf{x}(t) &= s(t)\mathbf{a}(\theta) + \mathbf{u}(t) \\ &= \mathbf{s}(t) + \mathbf{u}(t), \end{aligned} \tag{7.21}$$

trong đó  $\mathbf{a}(\theta)$  là đáp ứng của mảng và  $\mathbf{u}(t)$  là véc-tơ chứa tạp âm và nhiễu không tương quan. Đối với mảng thích nghi bằng hép, tín hiệu đầu ra có dạng giống như trình bày ở công thức (7.9)

$$y(t) = \mathbf{w}^H \mathbf{x}(t). \tag{7.22}$$

Tín hiệu sai số được định nghĩa như sau

$$\begin{aligned} \epsilon(t) &= r(t) - y(t) \\ &= r(t) - \mathbf{w}^H \mathbf{x}(t). \end{aligned} \tag{7.23}$$

Các trọng số được chọn để tối giản sai số bình phương trung bình (MSE: Mean Square Error) của tín hiệu lỗi

$$\text{MSE} \triangleq \text{E}\{\|\epsilon(t)\|^2\} = \text{E}\{|r(t) - \mathbf{w}^H \mathbf{x}(t)|^2\}, \tag{7.24}$$

trong đó  $E\{\cdot\}$  ký hiệu toán tử kỳ vọng. Khai triển (7.24) chúng ta có

$$\begin{aligned} E\{|\epsilon(t)|^2\} &= E\left\{\left[r(t) - \mathbf{w}^H \mathbf{x}(t)\right]\left[r(t) - \mathbf{w}^H \mathbf{x}(t)\right]^H\right\} \\ &= E\{|r(t)|^2\} - \mathbf{w}^H E\{\mathbf{x}(t)r^*(t)\} - E\{r^*(t)\mathbf{x}^H(t)\}\mathbf{w} \\ &\quad + \mathbf{w}^H E\{\mathbf{x}(t)\mathbf{x}^H(t)\}\mathbf{w} \\ &= E\{|r(t)|^2\} - \mathbf{w}^H \mathbf{r}_{xx} - \mathbf{r}_{xx}^H \mathbf{w} + \mathbf{w}^H \mathbf{R}_{xx} \mathbf{w}, \end{aligned} \quad (7.25)$$

trong đó  $\mathbf{r}_{xx} = E\{\mathbf{x}(t)r^*(t)\}$  và  $\mathbf{R}_{xx} = E\{\mathbf{x}(t)\mathbf{x}^H(t)\}$  được gọi tương ứng là *véc-tơ tương quan* (correlation vector) và *ma trận hiệp phương sai* (covariance matrix).

Véc-tơ trọng số tối ưu có thể tìm được bằng cách lấy gradiên (đạo hàm theo các biến) của (7.25) theo  $\mathbf{w}$ , rồi đặt bằng không và giải phương trình, tức là

$$\nabla_{\mathbf{w}} \text{MSE} \triangleq \frac{\partial E\{|\epsilon(t)|^2\}}{\partial \mathbf{w}} = 0. \quad (7.26)$$

Sử dụng tính chất  $\frac{\partial \mathbf{w} \mathbf{a}^H}{\partial \mathbf{w}} = \mathbf{a}^H$ ,  $\frac{\partial \mathbf{w}^H \mathbf{a}}{\partial \mathbf{w}} = \mathbf{0}$  chúng ta có thể tính được

$$\frac{\partial E\{|\epsilon(t)|^2\}}{\partial \mathbf{w}} = -\mathbf{r}_{xx}^H + \mathbf{w}^H \mathbf{R}_{xx} = 0. \quad (7.27)$$

Kết quả cho ta véc-tơ trọng số tối ưu theo tiêu chuẩn MMSE

$$\mathbf{w}_{\text{MMSE}} = \mathbf{R}_{xx}^{-1} \mathbf{r}_{xx}. \quad (7.28)$$

Công thức (7.28) thường được gọi là nghiệm Wiener-Hopf tối ưu [7].

### 7.5.2 Tiêu chuẩn MSINR cực đại

Nhắc lại (7.9) và sử dụng (7.21), ta có đầu ra của mảng có thể biểu diễn như sau

$$\begin{aligned} y(t) &= \mathbf{w}^H \mathbf{x}(t) = \mathbf{w}^H \mathbf{s}(t) + \mathbf{w}^H \mathbf{u}(t) \\ &= y_s(t) + y_u(t). \end{aligned} \quad (7.29)$$

Dựa vào kết quả này, chúng ta có thể định nghĩa tỉ số công suất tín hiệu trên công suất tạp âm cộng nhiễu (SINR) trung bình đầu ra

nhus sau

$$\text{SINR} = \frac{\mathbb{E}\{|y_s(t)|^2\}}{\mathbb{E}\{|y_u(t)|^2\}} = \frac{\mathbf{w}^H \mathbb{E}\{\mathbf{s}(t)\mathbf{s}^H(t)\}\mathbf{w}}{\mathbf{w}^H \mathbb{E}\{\mathbf{u}(t)\mathbf{u}^H(t)\}\mathbf{w}} = \frac{\mathbf{w}^H \mathbf{R}_{ss} \mathbf{w}}{\mathbf{w}^H \mathbf{R}_{uu} \mathbf{w}}, \quad (7.30)$$

trong đó  $\mathbf{R}_{ss} = \mathbb{E}\{\mathbf{s}(t)\mathbf{s}^H(t)\}$  và  $\mathbf{R}_{uu} = \mathbb{E}\{\mathbf{u}(t)\mathbf{u}^H(t)\}$ . Lấy gradien của (7.30) theo  $\mathbf{w}$  cho ta

$$\begin{aligned} \nabla_{\mathbf{w}} \text{SINR} &= \frac{\nabla_{\mathbf{w}}(\mathbf{w}^H \mathbf{R}_{ss} \mathbf{w})(\mathbf{w}^H \mathbf{R}_{uu} \mathbf{w}) - (\mathbf{w}^H \mathbf{R}_{ss} \mathbf{w}) \nabla_{\mathbf{w}}(\mathbf{w}^H \mathbf{R}_{uu} \mathbf{w})}{(\mathbf{w}^H \mathbf{R}_{uu} \mathbf{w})^2} \\ &= \frac{2\mathbf{R}_{ss} \mathbf{w}(\mathbf{w}^H \mathbf{R}_{uu} \mathbf{w}) - 2\mathbf{R}_{uu} \mathbf{w}(\mathbf{w}^H \mathbf{R}_{ss} \mathbf{w})}{(\mathbf{w}^H \mathbf{R}_{uu} \mathbf{w})^2}. \end{aligned} \quad (7.31)$$

Véc-tơ trọng số tối ưu  $\mathbf{w}_{opt}$  có thể tìm được bằng cách đặt  $\nabla_{\mathbf{w}} \text{SINR} = 0$ , và thu được

$$\mathbf{R}_{ss} \mathbf{w} = \frac{\mathbf{w}^H \mathbf{R}_{ss} \mathbf{w}}{\mathbf{w}^H \mathbf{R}_{uu} \mathbf{w}} \mathbf{R}_{uu} \mathbf{w} = \text{SINR} \mathbf{R}_{uu} \mathbf{w}. \quad (7.32)$$

Nếu  $\mathbf{R}_{uu}$  khả đảo, thì (7.32) có thể thu được

$$\mathbf{R}_{uu}^{-1} \mathbf{R}_{ss} \mathbf{w} = \text{SINR} \mathbf{w}. \quad (7.33)$$

Đây là bài toán tổng quát tìm giá trị riêng (eigen problem) trong đó giá trị bên phải của (7.31) được bao bởi giá trị riêng lớn nhất và nhỏ nhất của  $\mathbf{R}_{uu}^{-1} \mathbf{R}_{ss}$ . Giá trị riêng lớn nhất  $\lambda_{\max}$  thỏa mãn điều kiện sau đây

$$\mathbf{R}_{uu}^{-1} \mathbf{R}_{ss} \mathbf{w} = \lambda_{\max} \mathbf{w}. \quad (7.34)$$

So sánh (7.33) với (7.34) cho thấy  $\lambda_{\max}$  là giá trị SINR tối ưu. Tương ứng với giá trị  $\lambda_{\max}$  này chỉ có một véc-tơ riêng (eigenvector) cho bởi  $\mathbf{w}_{opt}$

$$\begin{aligned} \mathbf{w}_{opt} &= \frac{\mathbf{R}_{uu}^{-1} \mathbf{R}_{ss} \mathbf{w}_{opt}}{\text{SINR}} = \frac{\mathbf{R}_{uu}^{-1} \mathbb{E}\{\mathbf{s}(t)\mathbf{a}(\theta)\mathbf{s}(t)\mathbf{a}^H(\theta)\}\mathbf{w}_{opt}}{\text{SINR}} \\ &= \frac{\mathbf{R}_{uu}^{-1} \mathbf{a}(\theta)\mathbf{a}^H(\theta)\mathbf{w}_{opt} \mathbb{E}\{|s(t)|^2\}}{\text{SINR}}. \end{aligned} \quad (7.35)$$

Định nghĩa

$$\beta = \frac{\mathbf{a}^H(\theta)\mathbf{w}_{opt} \mathbb{E}\{|s(t)|^2\}}{\text{SINR}}, \quad (7.36)$$

chúng ta có thể biểu diễn véc-tơ trọng số tối ưu ở dạng tương tự biểu thức Wiener-Hopf như sau

$$\mathbf{w}_{\text{SINR}} = \mathbf{w}_{\text{opt}} = \beta \mathbf{R}_{uu}^{-1} \mathbf{a}(\theta). \quad (7.37)$$

### 7.5.3 Tiêu chuẩn hợp lệ cực đại (ML)

Nhắc lại véc-tơ tín hiệu đầu vào (7.21)

$$\begin{aligned} \mathbf{x}(t) &= s(t)\mathbf{a}(\theta) + \mathbf{u}(t) \\ &= \mathbf{s}(t) + \mathbf{u}(t) \end{aligned} \quad (7.38)$$

và định nghĩa hàm mật độ xác suất (pdf) của  $s(t)$  cho trước  $\mathbf{x}(t)$  như sau  $p_{\mathbf{x}(t)|s(t)}\{\mathbf{x}(t)\}$ . Giả thiết cho trước  $\mathbf{x}(t)$ , chúng ta mong muốn cực đại xác suất  $p_{\mathbf{x}(t)|s(t)}\{\mathbf{s}(t)\}$ . Do logarithm tự nhiên là một hàm đơn điệu, tăng  $p_{\mathbf{x}(t)|s(t)}\{\mathbf{x}(t)\}$  tương đương với tăng  $\ln[p_{\mathbf{x}(t)|s(t)}\{\mathbf{x}(t)\}]$ . Vì vậy, hàm hợp lệ (likelihood function) của  $\mathbf{x}(t)$  có thể định nghĩa được như sau

$$\mathcal{L}[\mathbf{x}(t)] = -\ln[p_{\mathbf{x}(t)|s(t)}\{\mathbf{x}(t)\}]. \quad (7.39)$$

Giả thiết rằng  $\mathbf{u}(t)$  là một véc-tơ chứa các phần tử có phân bố Gauss dừng với trung bình không và ma trận hiệp phương sai  $\mathbf{R}_{uu}$ , và giả thiết  $\mathbf{x}(t)$  là một véc-tơ Gauss ngẫu nhiên có trung bình  $s(t)\mathbf{a}(\theta)$ . Hàm hợp lệ có thể biểu diễn như sau [4]

$$\mathcal{L}[\mathbf{x}(t)] = C[\mathbf{x}(t) - \mathbf{a}(\theta)s(t)]^H \mathbf{R}_{uu}^{-1} [\mathbf{x}(t) - \mathbf{a}(\theta)s(t)], \quad (7.40)$$

trong đó  $C$  là một hằng số không phụ thuộc vào  $\mathbf{x}(t)$  và  $s(t)$ .

Mục đích của chúng ta là tìm ước lượng  $\hat{s}(t)$  của  $s(t)$  cho phép cực tiểu hóa (7.40). Đặt đạo hàm từng phần (partial derivative) của  $\mathcal{L}[\mathbf{x}(t)]$  theo  $s(t)$  bằng không [4]

$$\frac{\partial \mathcal{L}[\mathbf{x}(t)]}{\partial s(t)} = -2\mathbf{a}^H(\theta) \mathbf{R}_{uu}^{-1} \mathbf{x}(t) + 2\hat{s}(t) \mathbf{a}^H(\theta) \mathbf{R}_{uu}^{-1} \mathbf{a}(\theta) = 0 \quad (7.41)$$

và để ý rằng  $\mathbf{a}^H(\theta) \mathbf{R}_{uu}^{-1} \mathbf{a}(\theta)$  là một số vô hướng (scalar), cho ta

$$\hat{s}(t) = \frac{\mathbf{a}^H(\theta) \mathbf{R}_{uu}^{-1}}{\mathbf{a}^H(\theta) \mathbf{R}_{uu}^{-1} \mathbf{a}(\theta)} \mathbf{x}(t). \quad (7.42)$$

So sánh (7.42) với (7.9), có thể dễ dàng nhận thấy rằng véc-tơ trọng số tối ưu  $\mathbf{w}_{\text{opt}}$  sử dụng tiêu chuẩn ML là

$$\mathbf{w}_{\text{ML}} = \mathbf{w}_{\text{opt}} = \frac{\mathbf{R}_{uu}^{-1}\mathbf{a}(\theta)}{\mathbf{a}^H(\theta)\mathbf{R}_{uu}^{-1}\mathbf{a}(\theta)}. \quad (7.43)$$

**Định nghĩa**

$$\beta = \frac{1}{\mathbf{a}^H(\theta)\mathbf{R}_{uu}^{-1}\mathbf{a}(\theta)} \quad (7.44)$$

thì chúng ta có thể biểu diễn véc-tơ trọng số tối ưu sử dụng tiêu chuẩn ML ở dạng tương tự biểu thức Wiener-Hopf như sau

$$\mathbf{w}_{\text{ML}} = \beta \mathbf{R}_{uu}^{-1}\mathbf{a}(\theta). \quad (7.45)$$

#### 7.5.4 Tiêu chuẩn MV

Phương sai cực tiểu (MV), còn được gọi là phương sai cực tiểu ràng buộc tuyến tính (LCMV: Linear Constrained Minimum Variance), được sử dụng khi tín hiệu mong muốn và hướng tới của nó đều được biết trước. Nhắc lại đầu ra của bộ tạo búp sóng ở (7.9)

$$\begin{aligned} y(t) &= \mathbf{w}^H \mathbf{x}(t) = \mathbf{w}^H \mathbf{s}(t) + \mathbf{w}^H \mathbf{u}(t) \\ &= \mathbf{w}^H \mathbf{a}(\theta) \mathbf{s}(t) + \mathbf{w}^H \mathbf{u}(t). \end{aligned} \quad (7.46)$$

Để thu được tín hiệu mong muốn với một độ lợi xác định theo một hướng cho trước, chúng ta có thể sử dụng ràng buộc [7]

$$\mathbf{w}^H \mathbf{a}(\theta) = g. \quad (7.47)$$

Thay (7.47) vào (7.46), chúng ta thu được đầu ra của mảng tuân theo ràng buộc [8]

$$y(t) = gs(t) + \mathbf{w}^H \mathbf{u}(t). \quad (7.48)$$

Do  $\mathbf{u}(t)$  được giả thiết là một véc-tơ có chứa các phần tử ngẫu nhiên Gauss không tương quan, chúng ta có  $E\{y(t)\} = gs(t)$ . Do đó

phương sai của đầu ra mảng được cho bởi

$$\begin{aligned}\text{var}\{y(t)\} &= \text{E}\{\left[y(t) - gs(t)\right]\left[y(t) - gs(t)\right]^*\} \\ &= \text{E}\left\{\boldsymbol{w}^H \boldsymbol{u}(t) [\boldsymbol{w}^H \boldsymbol{u}(t)]^H\right\} \\ &= \boldsymbol{w}^H \boldsymbol{R}_{uu} \boldsymbol{w}.\end{aligned}\quad (7.49)$$

Sử dụng phương pháp Lagrange, chúng ta có

$$\nabla_{\boldsymbol{w}} \left\{ \boldsymbol{w}^H \boldsymbol{R}_{uu} \boldsymbol{w} - \beta [g - \boldsymbol{w}^H \boldsymbol{a}(\theta)] \right\} = 0 \quad (7.50)$$

tương đương với

$$\boldsymbol{R}_{uu} \boldsymbol{w} - \beta \boldsymbol{a}(\theta) = 0. \quad (7.51)$$

Nếu  $\boldsymbol{R}_{uu}$  khả đảo véc-tơ trọng số tối ưu sử dụng tiêu chuẩn MV có thể được biểu diễn như sau

$$\boldsymbol{w}_{\text{MV}} = \beta \boldsymbol{R}_{uu}^{-1} \boldsymbol{a}(\theta), \quad (7.52)$$

trong đó [7, 8]

$$\beta = \frac{g}{\boldsymbol{a}^H(\theta) \boldsymbol{R}_{uu}^{-1} \boldsymbol{a}(\theta)}. \quad (7.53)$$

Trường hợp  $g = 1$ , bộ tạo búp sóng MV thường được gọi là bộ tạo búp sóng đáp ứng không méo phương sai cực tiểu (MVDR: Minimum Variance Distortionless Response), hay bộ tạo búp sóng Capon [7].

## 7.6 Các thuật toán thích nghi

### 7.6.1 Thuật toán bình phương trung bình nhỏ nhất

Thuật toán bình phương trung bình nhỏ nhất (LMS: Least Mean Square) là thuật toán thích nghi phổ biến nhất trong bài toán thích nghi liên tục [7]. Thuật toán dựa trên phương pháp sườn dốc (steepest descent) [9]. Phương pháp sườn dốc chọn véc-tơ trọng số để cực tiểu hóa trung bình thống kê của bình phương lỗi về phía trung bình lỗi

nhỏ nhất (MSE). Sử dụng phương pháp sườn dốc, véc-tơ trọng số được cập nhật tại thời điểm  $(n + 1)$  được cho bởi [7]

$$\mathbf{w}(n + 1) = \mathbf{w}(n) - \frac{\mu}{2} \nabla \text{E}\{\epsilon^2(n)\}, \quad (7.54)$$

trong đó  $\mu$  là kích thước bước nhảy (step size) điều khiển đặc tính hội tụ  $\mathbf{w}(n)$

$$0 < \mu < \frac{1}{\lambda_{max}}. \quad (7.55)$$

Ở công thức trên  $\lambda_{max}$  là giá trị riêng (eigenvalue) của ma trận tương quan (covariance matrix)  $\mathbf{R}_{xx}$ . Từ công thức (7.26) chúng ta có

$$\nabla \text{E}\{\epsilon^2(n)\} = -2\mathbf{r}_{xr} + 2\mathbf{R}_{xx}\mathbf{w}(n). \quad (7.56)$$

Thay công thức (7.56) vào công thức (7.54), chúng ta có

$$\mathbf{w}(n + 1) = \mathbf{w}(n) + \mu[\mathbf{r}_{xr} - \mathbf{R}_{xx}\mathbf{w}(n)]. \quad (7.57)$$

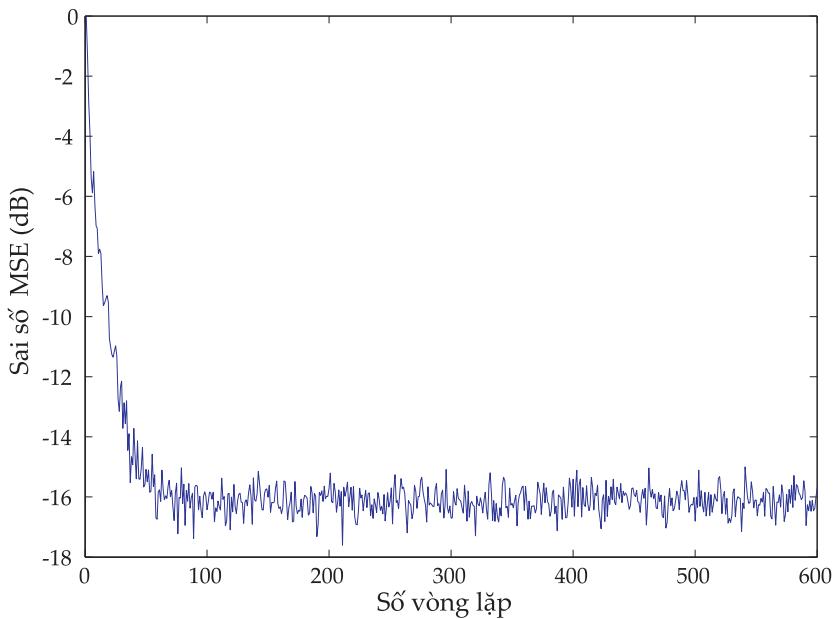
Để cập nhật trọng số tối ưu sử dụng công thức (7.57) cần phải biết trước cả  $\mathbf{R}_{xx}$  và  $\mathbf{r}_{xr}$ , và sử dụng các giá trị tức thời của chúng như sau

$$\begin{aligned} \mathbf{R}_{xx}(n) &= \mathbf{x}(n)\mathbf{x}^H(n) \\ \mathbf{r}_{xr}(n) &= \mathbf{x}(n)r^*(n). \end{aligned} \quad (7.58)$$

Nhờ vậy, (7.57) trở thành

$$\begin{aligned} \mathbf{w}(n + 1) &= \mathbf{w}(n) + \mu\mathbf{x}(n)[r^*(n) - \mathbf{x}^H(n)\mathbf{w}(n)] \\ &= \mathbf{w}(n) + \mu\mathbf{x}(n)[r^*(n) - y^*(n)] \\ &= \mathbf{w}(n) + \mu\mathbf{x}(n)\epsilon^*(n). \end{aligned} \quad (7.59)$$

Để ý rằng tốc độ hội tụ của thuật toán LMS phụ thuộc vào kích thước bước nhảy (step size)  $\mu$  và phụ thuộc vào mức độ trãi giá trị riêng của ma trận tương quan (covariance matrix)  $\mathbf{R}_{xx}$ . Hình 7.8 mô tả một ví dụ về đặc tính hội tụ (còn gọi là *learning curve*) của thuật toán LMS.



**Hình 7.8:** Ví dụ về đường huấn luyện của thuật toán LMS. Ăng-ten mảng tuyến tính với  $d = \lambda/2$ ,  $M = 4$ ,  $\mu = 0.05$ , và  $\text{SNR}_{\text{in}} = 0\text{dB}$

### 7.6.2 Thuật toán đảo ma trận mẫu

Thuật toán đảo ma trận mẫu (SMI: Sample Matrix Inversion) được sử dụng trong trường hợp biết trước cả tín hiệu mong muốn và tín hiệu tham chiếu. Trong trường hợp này thì các trọng số tối ưu có thể tính được sử dụng phép đảo trực tiếp ma trận tương quan  $\mathbf{R}_{xx}$  tương tự như ở công thức (7.28). Do cả hai tín hiệu mong muốn và tín hiệu tham chiếu đều biết trước nên có thể sử dụng ước lượng của chúng từ véc-tơ dữ liệu đầu vào như sau [10]

$$\begin{aligned}\mathbf{R}_{xx}(n) &= \frac{1}{n} \sum_{i=1}^n \mathbf{x}(i)\mathbf{x}^H(i) \\ \mathbf{r}_{xr}(n) &= \frac{1}{n} \sum_{i=1}^n \mathbf{x}(i)r^*(i).\end{aligned}\tag{7.60}$$

Từ công thức Wiener-Hopf ở (7.28) có thể thấy rằng véc-tơ trọng số ước lượng được sử dụng thuật toán SMI được cho bởi công thức sau

$$\mathbf{w}(n) = \mathbf{R}_{xx}^{-1}(n)\mathbf{r}_{xr}(n). \quad (7.61)$$

Có thể nhận thấy rằng thuật toán SMI là thuật toán thích nghi theo khồi. Thuật toán này đã được chứng minh là thuật toán ước lượng véc-tơ trọng số tối ưu nhanh nhất [11]. Tuy nhiên, thuật toán này có nhược điểm là yêu cầu độ phức tạp tính toán lớn và thường bất ổn định do phải thực hiện phép đảo ma trận có kích thước lớn [7].

### 7.6.3 Thuật toán bình phương nhỏ nhất đệ quy

Thuật toán bình phương nhỏ nhất đệ quy (RLS: Recursive Least Squares) sử dụng tiêu chí bình phương nhỏ nhất để ước lượng véc-tơ trọng số. Tức là, thuật toán lựa chọn véc-tơ trọng số sao cho tối thiểu hóa hàm chi phí của tổng các bình phương sai số trong khoảng một cửa sổ thời gian

$$Q(n) = \sum_{i=1}^n \gamma^{n-i} |\epsilon(i)|^2 \quad (7.62)$$

trong đó hàm sai số  $\epsilon(i)$  được định nghĩa ở công thức (7.23) và  $0 < \gamma < 1$  là hệ số quên (forgetting factor). Sử dụng phương pháp LS (least squares), ma trận hiệp phương sai và véc-tơ tương quan được cho bởi công thức sau [7]

$$\mathbf{R}_{xx}(n) = \sum_{i=1}^n \gamma^{n-i} \mathbf{x}(i) \mathbf{x}^H(i) \quad (7.63)$$

$$\mathbf{r}_{xr}(n) = \sum_{i=1}^n \gamma^{n-i} \mathbf{x}(i) r^*(i). \quad (7.64)$$

Tách thành phần tương ứng với  $i = n$  sẽ thu được (7.63) và (7.64) như sau

$$\begin{aligned}\mathbf{R}_{xx}(n) &= \sum_{i=1}^{n-1} \gamma^{(n-1)-i} \gamma \mathbf{x}(i) \mathbf{x}^H(i) + \mathbf{x}(n) \mathbf{x}^H(n) \\ &= \gamma \mathbf{R}_{xx}(n-1) + \mathbf{x}(i) \mathbf{x}^H(i)\end{aligned}\quad (7.65)$$

$$\begin{aligned}\mathbf{r}_{xr}(n) &= \sum_{i=1}^{n-1} \gamma^{(n-1)-i} \gamma \mathbf{x}(i) r^*(i) + \mathbf{x}(i) r^*(i) \\ &= \gamma \mathbf{r}(n-1) + \mathbf{x}(n) r^*(n).\end{aligned}\quad (7.66)$$

Áp dụng đẳng thức Woodbury, chúng ta có thể thu được nghịch đảo của ma trận hiệp phương sai như sau [7]

$$\mathbf{R}_{xx}^{-1}(n) = \gamma^{-1} \left[ \mathbf{R}_{xx}^{-1}(n-1) - \mathbf{q}(n) \mathbf{x}^H(n) \mathbf{R}_{xx}^{-1}(n-1) \right] \quad (7.67)$$

trong đó,

$$\mathbf{q}(n) = \frac{\gamma^{-1} \mathbf{R}_{xx}^{-1}(n-1) \mathbf{x}(n)}{1 + \gamma^{-1} \mathbf{x}^H(n) \mathbf{R}_{xx}^{-1}(n-1) \mathbf{x}(n)}. \quad (7.68)$$

Véc-tơ trọng số ước lượng được có thể được cập nhật sử dụng công thức (7.28)

$$\begin{aligned}\mathbf{w}(n) &= \mathbf{R}_{xx}^{-1}(n) \mathbf{r}_{xr}(n) \\ &= \gamma^{-1} \left[ \mathbf{R}_{xx}^{-1}(n-1) - \mathbf{q}(n) \mathbf{x}^H(n) \mathbf{R}_{xx}^{-1}(n-1) \right] \left[ \gamma \mathbf{r}(n-1) + \mathbf{x}(n) r^*(n) \right],\end{aligned}\quad (7.69)$$

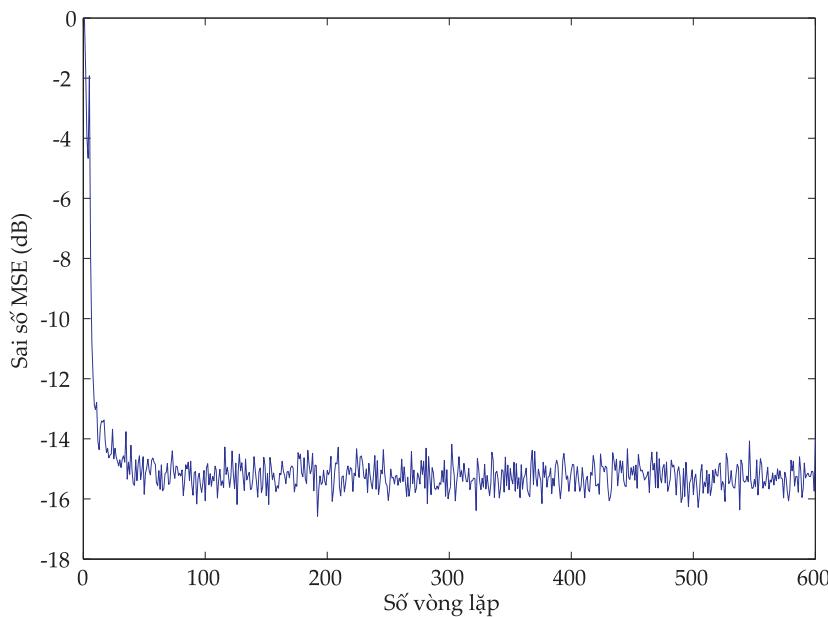
cuối cùng chúng ta thu được [7]

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{q}(n) \epsilon^*(n), \quad (7.70)$$

trong đó

$$\epsilon(n) = \mathbf{r}(n) - \mathbf{w}^H(n-1) \mathbf{x}(n). \quad (7.71)$$

Do thuật toán RLS sử dụng thông tin từ mẫu khởi tạo để ước lượng các trọng số nên tốc độ hội tụ nhanh hơn thuật toán LMS vài lần [12]. Tuy nhiên, cải thiện về tốc độ hội tụ đạt được với chi phí về gia tăng độ phức tạp tính toán. Một ví dụ về đặc tính hội tụ của thuật toán RLS được biểu diễn ở Hình 7.9. So sánh đường huấn luyện của thuật toán RLS với LMS trên Hình 7.8 có thể thấy thuật toán RLS có tốc độ hội tụ nhanh hơn



**Hình 7.9:** Một ví dụ về đường huấn luyện của thuật toán RLS. Ăng-ten mảng thích nghi với  $d = \lambda/2$ ,  $M = 4$ ,  $\gamma = 0.999$ , và  $\text{SNR}_{\text{in}} = 0\text{dB}$

## 7.7 Ưu điểm của ăng-ten thích nghi

Việc sử dụng ăng-ten mảng thích nghi mang lại nhiều lợi ích cho thông tin vô tuyến và đã được trình bày nhiều trong các tài liệu tham khảo [7, 13, 14]. Một số lợi điểm chính được trình bày tóm tắt ở phần sau đây.

### 7.7.1 Cải thiện chất lượng tín hiệu

Nhờ sử dụng nhiều phần tử ăng-ten nên ăng-ten thích nghi có thể mang lại độ lợi mảng theo số lượng các phần tử mảng được sử dụng. Độ lợi mảng này sẽ đem lại cải thiện về tỉ số SINR. Nếu định nghĩa SNR đầu vào là  $\text{SNR}_{\text{in}}$  và nếu số nguồn nhiễu nhỏ hơn độ tự do của dàn (DoF: Degree of Freedom) cho bởi  $\text{DoF} = M - 1$ , thì SINR đầu ra trong môi trường truyền sóng đơn tia (không chịu ảnh hưởng của các tia đa đường) có thể biểu diễn như sau

$$\text{SINR}_{\text{out}} = M \cdot \text{SNR}_{\text{in}}, \quad (7.72)$$

hay

$$\text{SINR}_{\text{out}}[\text{dB}] = 10 \log_{10} M + \text{SNR}_{\text{in}}[\text{dB}], \quad (7.73)$$

trong đó  $M$  là số phần tử của mảng.

Trong môi trường pha-định đa đường, nếu xử lý tín hiệu được thực hiện trên cả hai miền không gian và thời gian như trường hợp sử dụng bộ tạo búp sóng băng rộng thì có thể thu thêm độ lợi phân tập đa đường tùy theo số khâu (tap) sử dụng trong đường dây giữ chậm phân khâu (TDL) và đặc tính truyền sóng. Ví dụ xét trường hợp mô hình hai tia sau đây. Giả thiết có hai tia không tương quan với một tia tới từ góc  $0^\circ$  và một tia giữ chậm từ góc  $30^\circ$ . Lúc này, SINR đầu ra được ước lượng như sau

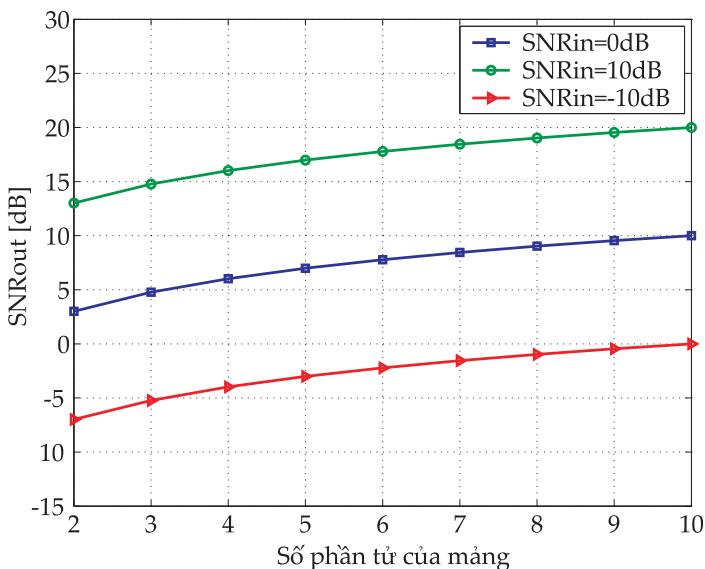
$$\text{SINR}_{\text{out}}[\text{dB}] = 10 \log_{10} M + 10 \log_{10}(2) + \text{SNR}_{\text{in}}[\text{dB}]. \quad (7.74)$$

Điều này có nghĩa là dàn ăng-ten đã thu được thêm 3dB độ lợi phân tập trong môi trường đa đường. Với môi trường pha-định càng giàu tần xạ thì độ lợi phân tập thu được càng lớn. Hình 7.10 biểu diễn SNR đầu ra theo số phần tử ăng-ten sử dụng.

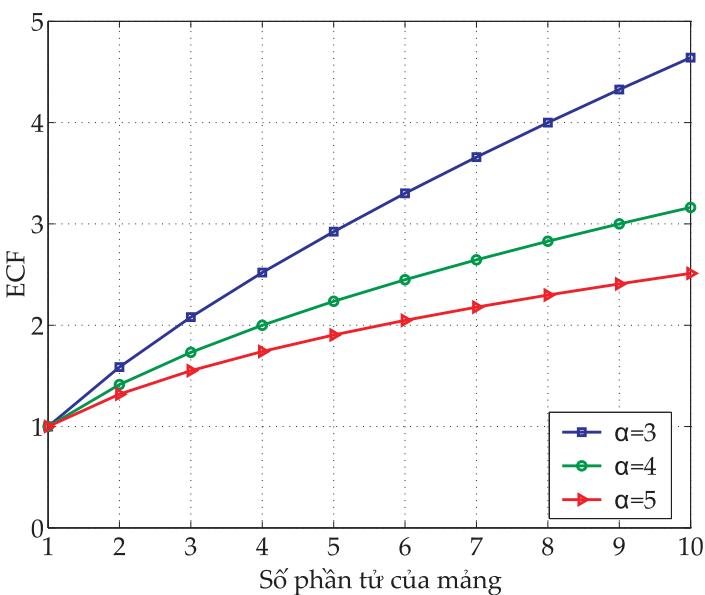
### 7.7.2 Mở rộng vùng phủ

Từ công thức (7.73), có thể nhận thấy rằng độ lợi mảng thu được bởi dàn ăng-ten là

$$G = 10 \log_{10} M. \quad (7.75)$$



Hình 7.10: Tỉ số SNR đầu ra theo số phần tử ăng-ten.



Hình 7.11: Cải thiện vùng phủ nhờ sử dụng ăng-ten thích nghi.

Độ lợi thêm này cho phép mở rộng vùng phủ của trạm gốc. Khi trai góc (angular spread) là nhỏ và suy hao đường truyền được mô hình hóa với hệ số mũ  $\alpha$ , thì hệ số mở rộng vùng phủ (REF: Range

Extension Factor) được cho bởi [13] công thức sau

$$\text{REF} = \frac{r_{\text{array}}}{r_{\text{conv}}} = M^{\frac{1}{\alpha}}, \quad (7.76)$$

trong đó  $r_{\text{conv}}$  và  $r_{\text{array}}$  tương ứng là cự ly vùng phủ bởi ăng-ten thông thường và ăng-ten mảng. Hệ số mở rộng vùng phủ (ECF: Extended Coverage Factor) là [13]

$$\text{ECF} = \left( \frac{r_{\text{array}}}{r_{\text{conv}}} \right)^2 = \text{REF}^2. \quad (7.77)$$

Hình 7.11 biểu diễn kết quả cho một dàn ăng-ten 6 phần tử 6. Có thể thấy vùng phủ tăng gấp đôi so với trường hợp đơn ăng-ten cho trường hợp  $\alpha = 5$ . Do nghịch đảo của ECF biểu thị hệ số tiết kiệm số lượng các trạm gốc cần thiết để bao phủ cùng một vùng sử dụng đơn ăng-ten [13], nên rõ ràng là sử dụng ăng-ten thích nghi có thể giảm được một số lượng lớn các trạm gốc. Ví dụ, với trường hợp sử dụng  $\alpha = 5$  như ở trên, số lượng các trạm gốc có thể tiết kiệm có thể giảm xuống chỉ còn một nửa so với số lượng ban đầu.

### 7.7.3 Giảm công suất phát

Trong phần 7.7.1 chúng ta đã thấy việc sử dụng ăng-ten thích nghi có thể mang lại một độ lợi mảng lớn. Độ lợi này sẽ dẫn đến hiệu quả tiết kiệm công suất phát của trạm gốc. Giả thiết là độ nhạy thu được giữ nguyên, thì yêu cầu về công suất của trạm gốc sử dụng dàn ăng-ten  $M$  phần tử giảm xuống  $M^{-1}$  và công suất đầu ra yêu cầu của bộ khuếch đại công suất ở trạm gốc có thể giảm xuống  $M^{-2}$  [13]. Hiệu quả của giảm công suất phát có lợi cho sức khỏe của người sử dụng và tiết kiệm chi phí chế tạo các bộ khuếch đại công suất lớn.

### 7.7.4 Phẩm chất BER của các hệ thống ăng-ten thích nghi

Việc phân tích phẩm chất xác suất lỗi bit  $P_E$  của các hệ thống truyền dẫn sử dụng ăng-ten thích nghi là khá phức tạp do ăng-ten thích

nghi yêu cầu tín hiệu trên các phần tử ăng-ten tương quan với nhau ( $d < \frac{\lambda}{2}$ ). Trong trường hợp pha-định ảnh hưởng tới các ăng-ten là độc lập, xác suất lỗi bít cho trường hợp không có ảnh hưởng của nhiều giống xác suất lỗi bit của hệ thống thu phân tập MRC. Trong trường hợp hệ thống chịu ảnh hưởng của một nguồn nhiễu mạnh, xác suất lỗi bít phụ thuộc và tỉ số công suất nhiễu trên tạp âm (INR: Interference to Noise Ratio)  $\Gamma_i$  theo quy luật sau:

$$P_E \propto \frac{1}{(M\Gamma_i)^{M-1}}. \quad (7.78)$$

## 7.8 Tóm tắt

Chương này đã trình bày tổng quan về ăng-ten thích nghi và các nguyên lý kỹ thuật xử lý tín hiệu cơ bản sử dụng trong các hệ thống ăng-ten thích nghi. Các mô hình tín hiệu cũng được trình bày cho cả hai trường hợp kênh truyền băng hẹp và kênh truyền băng rộng. Các tiêu chuẩn tối ưu theo MMSE, MSINR, MV và ML đã được trình bày cùng với các thuật toán thích nghi SMI, RLS, LMS. Cuối cùng chương cũng đã phân tích các ưu điểm của ăng-ten thích nghi trong các hệ thống tin di động. Các chương trình mẫu ở cuối chương mô tả mô phỏng hoạt động chi tiết của ăng-ten thích nghi.

## Tài liệu tham khảo

- [1] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive antenna systems," *Proceedings of IEEE*, vol. 55, pp. 2143–2159, December 1967.
- [2] R. T. Compton, *Adaptive Antennas: Concepts and Performance*. Prentice Hall, 1988.
- [3] B. D. Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE Signal Processing Magazine*, pp. 4–24, April 1988.
- [4] R. A. Monzingo and T. W. Miller, *Introduction to Adaptive Arrays*. John Wiley & Sons, 1980.

- [5] L. C. Godara, "Application of antenna arrays to mobile communications, part 2: Beam-forming and direction-of-arrival considerations," *Proceedings of the IEEE*, vol. 85, no. 8, pp. 1193–1245, August 1997.
- [6] X. N. Tran, "Subband adaptive arrays for mobile communications with applications to CDMA systems," PhD dissertation, The University of Electro-Communications, 2003.
- [7] J. Litva and T. K.-Y. Lo, *Digital Beamforming in Wireless Communications*. Artech House, 1996.
- [8] R. B. Ertel, "Antenna array systems: Propagation and performance," PhD Thesis, Virginia Polytechnic Institute and State University, 1999.
- [9] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Prentice Hall, 1985.
- [10] N. Kikuma, *Adaptive Signal Processing with Array Antenna*. Science and Technology Publishing Co., Ltd., Japan, 1999.
- [11] J. E. Hudson, *Adaptive Array Principle*. Peter Peregrinus Ltd., 1981.
- [12] S. Haykin, *Adaptive Filter Theory*. 4th edition., Prentice Hall, 2002.
- [13] G. V. Tsoulos, "Smart antennas for mobile communication systems," *Electronics and Communication Engineering Journal*, vol. 11, no. 2, pp. 84–94, April 1999.
- [14] J. H. Winters, "Smart antennas for wireless systems," *IEEE Personal Communications*, vol. 5, no. 1, pp. 23–27, February 1988.

## Phụ lục: Chương trình MATLAB mẫu

---

### Function 7.1 steervect.m

```

function a = steervect(theta,d,lambda,M)
% function a = steervect(theta,d,lambda,M)
% Array steering vector of an array with m elements
% d: distance b/w array elements,
% theta: angle of arrival (in radians),
% lambda: wave length
% Written by Tran Xuan Nam, The University of Electro-Communications
% Date: 13/11/2000
% Ref: - J.E Hudson, "Adaptive Array Principles", IEE Press, 1991 %

```

```
% Last modified on 06/11/2000

if nargin<3
    disp('Incorrect input arguments')
help sourcevect;
return;
elseif nargin==3
    d=lambda/2;
end

% Setup source space vector containing m elements.

m = 1:M;
a = exp(-j*(m-1)*2*pi*d*sin(theta)/lambda);
% note: take +sin for correct plot (in the past -)
%!
```

---

**Function 7.2 arraypat.m**

```
function bpat = arraypat(a_theta0,a_theta);
% Function to calculate the array antenna beam pattern;
% a_alpha is the setup space vector to plot.
% a_theta is input space source vector

% Written by Tran Xuan Nam, The University of Electro-Communications
% Date: 01/11/2000;
% Ref: J.E Hudson, "Adaptive Array Principles"

if nargin <2
    disp('Not enough input arguments');
    help arraybeam;
    return
end

[M,N]=size(a_theta0);
```

```
bpat=zeros(1,N);
bpat=a_theta0'*a_theta/M;
%!
```

---

**■ Function 7.3 snrcal.m**

```
function sndb=snrcal(sig,ref)

% function sndb=snrcal(sig,ref)
% To calculate the signal to noise ratio using correlation method
% SNDb is the signal-to-noise ratio in dB,
% sig: the signal vector plus additive noise,
% ref: the reference signal
% Written by Tran Xuan Nam (The University of Electro-Communications)
% Date: 07/11/2000
% Modified on 09/11/2000

if nargin<2
    disp('Not enough input argument');
    help snrcal
    return
end
rho=mean(sig.*conj(ref))/sqrt(mean(abs(sig).^2).*mean((abs(ref)).^2));
sn=(abs(rho).^2)/(1-(abs(rho)).^2);
sndb=10*log10(sn);
%!
```

---

**■ MATLAB Program 7.1 simArrayPattern.m**

```
% Program to demonstrate beam rotating operation of array antenna
%     steervect.m to calculate signal vectors to elements
%     wtsmi.m to calculate weight matrix using SMI
%     arraypat.m to calculate array pattern
% Written by Tran Xuan Nam, The University of Electro-Communications
% Ref: - J.E Hudson, "Adaptive Array Principles", IEE Press, 1991
% Date: 16/11/2000
```

```
% Last updated:

%% Start program
clear all;
close all;
randn('seed',0);
rand('seed',0);

%% Input simulation parameters
M = input('Enter the number of elements M = ')
lambda = 1; % Wave length
d = lambda/2; % Distance b/w antenna elements
resnum = 1; % Resolution number to simulate
%theta0 = linspace(0,pi,resnum); % Angle of Arrival
theta0 = 0;% pi/3;
theta0_rot = pi/2 - theta0; % Rotated AoA coordinate
%% Calculate array steering vector to the source
s = zeros(M,resnum);
for i=1:resnum
a_theta0(:,i) = steervect(theta0_rot(i),d,lambda,M);
end
%% Calculate array pattern
numplot = 360; % Number of points to plot,
theta = linspace(0,2*pi,numplot); % resolution at 1 degree
theta_deg = theta*180/pi;
a_theta = zeros(M,numplot);
for i = 1 : numplot
a_theta(:,i) = steervect(theta(i),d,lambda,M);
end
beam_pattern = zeros(resnum,numplot);
for i = 1 : resnum
beam_pattern(i,:) = arraypat(a_theta0(:,i),a_theta);
% Plot pattern in the polar coordinates, pi/2 is to rotate the
% coordinates
polar(pi/2-theta,abs(beam_pattern(i,:)))
```

```

xlabel('Theta(Degrees)');
ylabel('Nomalized pattern');
title(['PATTERN OF ARRAY WITH ' num2str(M) ' ELEMENTS AOA= '...
        num2str(theta0(i)*180/pi) ' DEGS' ]);
pause(2); % Pause for 2 seconds
figure(2)
plot(theta_deg-108,abs(beam_pattern(i,:)))
end
%

```

#### Function 7.4 wtsmi.m

```

function w = wtsmi(x,ref);
% Returns weight vector using SMI algorithm
% x: the signal vector
% ref: reference signal.
% x and ref must be in same length
% Written by Tran Xuan Nam (The University of Electro-Communications)
% Date: 15/10/2000

if nargin~=2
    disp('Incorrect input arguments')
    help wtLMS;
    return;
end

if length(x)~=length(ref)
    disp('X and Y must be in same length');
    help wtLMS
    return;
end

phi_x=x*x'; % Signal covariance matrix
Sx=x*ref'; % Reference correlation vector
wr=inv(phi_x)*Sx; % Weight calculation in rows
w=wr; % Weight coefficients in columns

```

%!

---

**Function 7.5 wtlms.m**

```
function w = wtsmi(x,ref);
% Returns weight vector using SMI algorithm
% x: the signal vector
% ref: reference signal.
% x and ref must be in same length
% Written by Tran Xuan Nam (The University of Electro-Communications)
% Date: 15/10/2000

if nargin~=2
    disp('Incorrect input arguments')
    help wtlms;
    return;
end

if length(x)~=length(ref)
    disp('X and Y must be in same length');
    help wtlms
    return;
end

phi_x=x*x'; % Signal covariance matrix
Sx=x*ref'; % Reference correlation vector
wr=inv(phi_x)*Sx; % Weight calculation in rows
w=wr; % Weight coefficients in columns
%!
```

---

**Function 7.6 plotbeampat.m**

```
function rho = plotbeampat(w,numplot,type)
% function rho = plotbeampat(w,numplot,type)
% Program to calculate array antenna beam pattern;
% w: weight matrix.
```

```
% numplot: number of points to plot
% type: coordinate to plot; type=1: decade, type=2: polar.
% Written by Tran Xuan Nam (University of Electro-Communications)
% Date: 23/1/2001;
% Ref: J.E Hudson, "Adaptive Array Principles"

if nargin <3
    disp('Not enough input arguments');
    help antpat;
    return
end

M = length(w);
lambda = 1;
d = lambda/2;
theta = linspace(-pi,pi,numplot); % resolution is 1 degree
theta_deg = theta*360/pi;
a = zeros(M,numplot);
for n = 1 : numplot
    a(:,n) = steervect(theta(n),d,lambda,M);
end
rho = a'*w/M;
if type == 1
    rho = 10*log10(rho/max(rho));
    plot(theta_deg,rho)
elseif type == 2
    polar(theta,rho')
    view([90 -90])
end
title(['BEAM PATTERN OF ADAPTIVE ARRAY WITH ' num2str(M) 'elements']);
xlabel('Angle (deg)');
ylabel('Normalized Gain');
grid
%!
```

```
% Program to simulate operation of Adaptive Array Antenna
% Related functions:
%   sourcevect.m: array source vector
%   wtsmi.m to calculate weight matrix using SMI algorithm
%   arraypat.m to calculate array patern
%   snrcal.m to calculate SNR
% Written by Tran Xuan Nam (The University of Electro-Communications)
% Ref: - J.E Hudson, "Adaptive Array Principles", IEE Press, 1991
% Date: 02/11/2000
% Last updated:12/12/2000

% Start program
clear all;
clc
close all;

%% Input simulation parameters
lambda=1; % Wave length
d=lambda/2; % Distance b/w antenna elements
M = input('Enter the number of array elements M: ');
disp('DESIRED SOURCE INFORMATION')
AoA = input('Enter angle of arrival (AoA) in radians (e.g.: pi/3): ');
SNRdB = input('Enter the SNR in dB: ');

% Select adaptive algorithm
algsel = input('Select adaptive algorithm (SMI/LMS/RLS): ','s');

%% Generate transmit signal
fr_len = 1000; % Length of transmit frame
ref_len = 200; % Length of reference symbols
data_len = fr_len - ref_len;
ref = randint(1,ref_len); % Random binary data stream
ref_sym = pskmod(ref,2); % Reference BPSK symbols
data = randint(1,data_len);
data_sym = pskmod(data,2); % Data BPSK symbols
fr_sym = [ref_sym data_sym];% Transmit frame
s = fr_sym;
```

```
% Rx signal vectors at array
theta=AoA;           % Rotated coordinate aoa
a = steervect(theta,d,lambda,M);
xm = a.*s; % Received signal at M elements
% Generate noise
%SNRdB = 0;      % Input SNR in dB
snr = 10.^(SNRdB/10); %
sigma = 1./sqrt(2*snr); % Noise variance
% Simulation loop
num_fr = 100; % 1000 frames
for n = 1:length(snr)
    for it = 1:num_fr
        z = sigma(n)*(randn(M,fr_len)+j*randn(M,fr_len)); % AWGN
        x = xm + z;
        % Array weight estimation
        switch algsel
            case 'SMI'
                w = wtsmi(x(:,1:ref_len),ref_sym); % SMI algorithm
            case 'LMS'
                mu = 0.05;
                [err(it,:),w] = wtlms(x(:,1:ref_len),ref_sym,mu);
            case 'RLS'
                gamma = 0.99;
                [err(it,:),w]= wtrls(x(:,1:ref_len),ref_sym,gamma);
            otherwise
                disp('Restart and Select algorithm again!')
        end
        % Array output
        y = w'*x(:,ref_len+1 : end);
        data_hat = pskdemod(y,2);
        %% Calculate output SNR
        snr_in(n,it)=snrcal(x(1,ref_len+1:end),data_sym);
        snr_out(n,it)=snrcal(y,data_sym);
        snr_gain(n,it)=snr_out(n,it) - snr_in(n,it);
        BER(n,it) = biterr(data,data_hat);
```

```
end
end
%% Plot learning curve
switch algsel
case 'LMS' | 'RLS'
    mse = sum((abs(err)).^2,1)/num_fr;
    msedB = 10*log10(mse/max(mse));
    plot(msedB)
    xlabel('Number of iterations')
    ylabel('Average MSE (dB)')
end
%% Plot beampattern
numplot = 180;
type = 2;
rho = plotbeampat(w,numplot,type);
%% Display array output SNR and array gain
snr_in = sum(snr_in,2)/num_fr;
snr_out = sum(snr_out,2)/num_fr;
snr_gain = sum(snr_gain,2)/num_fr;
disp(['Array Input SNR: ' num2str(snr_in) 'dB']);
disp(['Array Output SNR: ' num2str(snr_out) 'dB']);
disp(['SNR Gain: ' num2str(snr_gain) ' dB']);
%!
```

---

## **Phần IV**

# **Mã không gian-thời gian và các hệ thống MIMO**

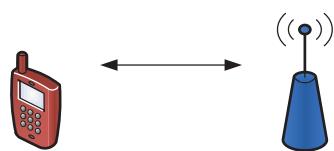


## Chương 8

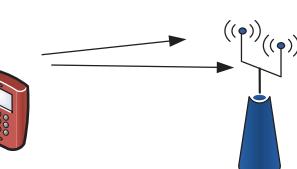
# Các hệ thống MIMO

### 8.1 Mô hình kênh MIMO

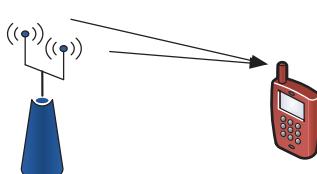
Các hệ thống phân tập không gian có thể được phân loại thành các hệ thống như minh họa trên Hình 8.1:



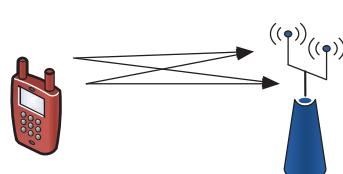
(a) SISO: Single Input Single Output



(b) SIMO: Single Input Multiple Output  
(Phân tập thu)



(c) MISO: Multiple Input Single Output  
(Phân tập phát)



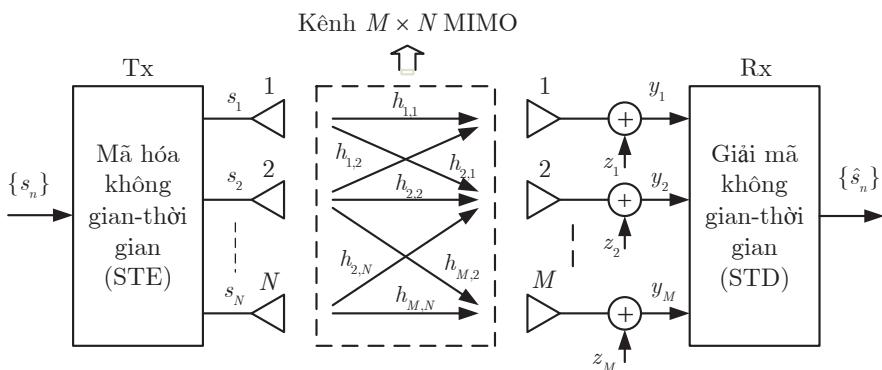
(d) MIMO: Multiple Input Multiple Output  
(Phân tập kết hợp phát/thu)

Hình 8.1: Các mô hình phân tập không gian.

- Hệ thống đơn ăng-ten phát, đơn ăng-ten thu (SISO: Single-Input Single-Output). Đây là trường hợp hệ thống truyền dẫn

thông thường, không sử dụng phân tập không gian đã được trình bày ở Phần II.

- Hệ thống đơn ăng-ten phát, đa ăng-ten thu (SIMO: Single-Input Multiple-Output). Hệ thống SIMO tương ứng với các hệ thống phân tập thu và ăng-ten mảng thích nghi đã được trình bày ở Phần III.
- Hệ thống đa ăng-ten phát, đơn ăng-ten thu (MISO: Multiple-Input Single-Output). Đây chính là hệ thống phân tập phát trình bày ở Chương 6 của Phần III.
- Hệ thống đa ăng-ten phát, đa ăng-ten thu (MIMO: Multiple-Input Multiple Output). Nội dung chương này tập trung chính vào hệ thống MIMO này.



Hình 8.2: Mô hình kênh MIMO vô tuyến.

Xét một hệ thống truyền dẫn vô tuyến sử dụng cả phân tập phát và thu với  $N$  ăng-ten phát và  $M$  ăng-ten thu như ở Hình vẽ 8.2. Kênh truyền giữa các ăng-ten máy phát (Tx) và ăng-ten máy thu (Rx) như mô tả ở hình vẽ được gọi là một kênh đa đầu vào-đa đầu ra MIMO (MIMO: Multiple Input-Multiple Output). Một hệ thống truyền dẫn trên kênh MIMO được gọi là hệ thống truyền dẫn MIMO. Trong các trường hợp đặc biệt khi  $N = 1$  và  $M = 1$ , tương ứng chúng ta có các hệ thống phân tập thu SIMO (Single-Input Multiple Output)

và phát MISO (Multiple-Input Single-Output) như đã trình bày ở Chương 6.

Kênh truyền đơn giữa ăng-ten máy thu  $m \in \{1, 2, \dots, M\}$  và ăng-ten máy phát  $n \in \{1, 2, \dots, N\}$  được ký hiệu là  $h_{mn}$ . Tương tự như các hệ thống phân tập phát hoặc thu ở Chương 6, để tránh ảnh hưởng giữa các ăng-ten phát hoặc các ăng-ten thu với nhau khoảng cách yêu cầu tối thiểu giữa các phần tử ăng-ten ở các mảng ăng-ten phát hoặc thu là  $\lambda/2$ . Thực tế khoảng cách này cần phải lựa chọn vào khoảng vài bước sóng. Kênh MIMO trong trường hợp này được gọi là kênh MIMO không tương quan (uncorrelated MIMO channel). Trong trường hợp pha-định Rayleigh bằng phẳng (flat fading) không có tương quan,  $h_{mn}$  được mô hình hóa bằng một biến số Gauss phức có giá trị trung bình 0 và phương sai 1. Một kênh MIMO gồm  $N$  ăng-ten phát và  $M$  ăng-ten thu thường được biểu diễn bởi một ma trận số phức gồm  $M$  hàng và  $N$  cột như sau

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1N} \\ h_{21} & h_{22} & \cdots & h_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ h_{M1} & h_{M2} & \cdots & h_{MN} \end{bmatrix}. \quad (8.1)$$

Định nghĩa các véc-tơ phát, thu và tệp âm tương ứng là:

$$\mathbf{s} = [s_1, s_2, \dots, s_N]^T \quad (8.2)$$

$$\mathbf{y} = [y_1, y_2, \dots, y_M]^T \quad (8.3)$$

$$\mathbf{z} = [z_1, z_2, \dots, z_M]^T \quad (8.4)$$

chúng ta có mối quan hệ giữa tín hiệu thu và phát biểu diễn qua phương trình hệ thống sau

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{z}. \quad (8.5)$$

Ở đây, tổng công suất phát trung bình trong một chu kỳ tín hiệu được giới hạn bằng  $P_T$  cho bất kỳ số lượng ăng ten phát  $N$  nào. Điều đó có nghĩa là, ma trận tương quan (covariance matrix) của  $s$ ,

$\mathbf{R}_{ss} = \text{E}\{\mathbf{s}\mathbf{s}^H\}$ , thỏa mãn điều kiện

$$\text{trace}(\mathbf{R}_{ss}) = P_T, \quad (8.6)$$

trong đó  $\text{trace}(\mathbf{A})$  là "vết" của ma trận  $\mathbf{A}$  được tính bằng tổng tất cả các phần tử nằm trên đường chéo chính của  $\mathbf{A}$ ;  $z$  là véc-tơ tạp âm với các phần tử  $z_m$  được mô phỏng bởi các biến số phức Gauss độc lập có phân bố như nhau và có cùng công suất trung bình  $\sigma^2$ , tức là,  $\text{E}\{zz^H\} = \sigma_z^2 \mathbf{I}_M$ , trong đó  $\mathbf{I}_M$  biểu diễn một ma trận đơn vị với  $M$  hàng và  $M$  cột.

## 8.2 Dung lượng kênh truyền MIMO

Tuy trong thực tế kênh truyền  $\mathbf{H}$  là một đại lượng ngẫu nhiên, nhưng trước hết chúng ta sẽ khảo sát dung lượng của một kênh truyền cố định (deterministic). Trong phạm vi cuốn sách này, chúng ta giả thiết rằng máy thu biết hoặc ước lượng được chính xác thông tin về kênh truyền được biểu diễn mới ma trận kênh  $\mathbf{H}$ .

### 8.2.1 Dung lượng kênh truyền cố định

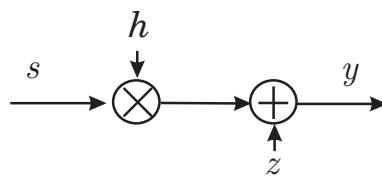
Dung lượng kênh truyền (channel capacity) được định nghĩa là tốc độ có thể truyền dẫn tối đa với một xác suất lỗi tương đối nhỏ nào đó. Dung lượng của một kênh truyền chịu ảnh hưởng của tạp âm nhiễu công trắng Gauss do Shannon tìm ra vào năm 1948 được biểu diễn như sau [1]

$$C = W \log_2(1 + \rho) \quad [\text{bits/s}], \quad (8.7)$$

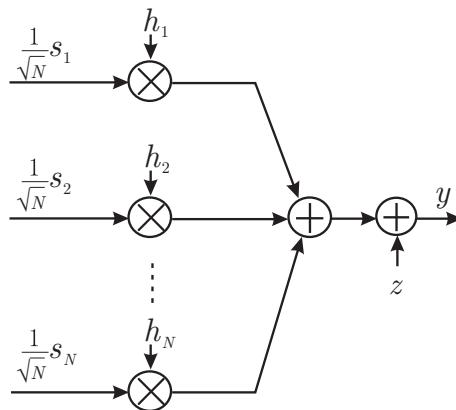
trong đó  $W$  là băng thông của kênh truyền tính bằng đơn vị Hz,  $\rho$  là tỉ số công suất tín hiệu trên tạp âm (SNR).

**Kênh SISO:** Trong trường hợp truyền tín hiệu qua một kênh truyền cố định SISO có hệ số kênh truyền  $h$  như ở Hình vẽ 8.3 chúng ta có tỉ số SNR tại đầu vào máy thu như sau

$$\rho_{\text{SISO}} = \frac{P_R}{P_N} = \frac{P_T |h|^2}{\sigma_z^2} = \rho |h|^2, \quad (8.8)$$



Hình 8.3: Mô hình tương đương của kênh truyền SISO.



Hình 8.4: Mô hình tương đương của kênh truyền MISO.

trong đó  $P_T = E\{|s|^2\}$  là công suất tín hiệu tại máy phát còn  $P_R = E\{|hs|^2\}$  là công suất tín hiệu tại máy thu. Dung lượng kênh truyền trong trường hợp này có thể tính được bằng cách thay tỉ số SNR  $\rho_{SISO}$  vào công thức Shannon:

$$C_{SISO} = W \log_2(1 + \rho|h|^2) \quad [\text{bits/s}]. \quad (8.9)$$

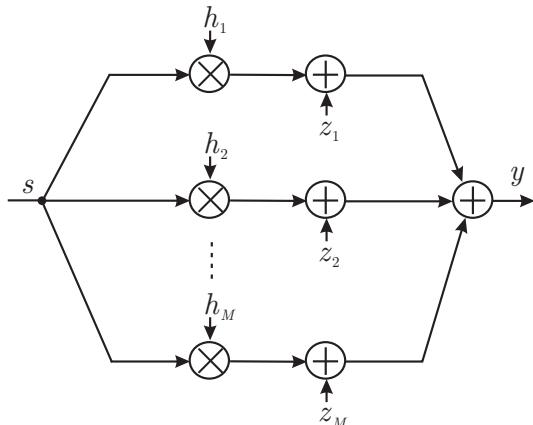
**Kênh MISO:** tương tự như kênh truyền SISO, đối với các trường hợp kênh truyền phân tập phát MISO ở Hình vẽ 8.4 chúng ta có thể tính được tỉ số SNR  $\rho_{MISO}$  và dung lượng kênh truyền  $C_{MISO}$  như sau

$$\rho_{MISO} = \frac{\frac{1}{N} \sum_{n=1}^N |h_n|^2 P_T}{P_N} = \frac{\rho}{N} \sum_{n=1}^N |h_n|^2 \quad (8.10)$$

$$C_{MISO} = W \log_2 \left( 1 + \frac{\rho}{N} \sum_{n=1}^N |h_n|^2 \right) \quad [\text{bits/s}], \quad (8.11)$$

trong đó hệ số  $\frac{1}{\sqrt{N}}$  được sử dụng để chuẩn hóa công suất phát. Điều

kiện chuẩn hóa đảm bảo sao cho tổng công suất phát ra từ máy phát với  $N$  ăng-ten bằng với công suất phát ra trên một ăng-ten trong hệ thống SISO do các điều kiện giới hạn về công suất phát đặt ra.



**Hình 8.5:** Mô hình tương đương của kênh truyền SIMO.

**Kênh SIMO:** đối với kênh truyền SIMO ở Hình vẽ 8.5, tỉ số SNR trên một nhánh phân tập là

$$\rho_m = \frac{P_T |h_m|^2}{\sigma_m^2}. \quad (8.12)$$

Giả sử rằng công suất tạp âm trên  $M$  nhánh phân tập thu đều như nhau, tức là  $\sigma_m^2 \triangleq \sigma_z^2$ , thì dung lượng kênh truyền  $C_{\text{SIMO}}$  tính được như sau:

$$C_{\text{SIMO}} = W \log_2 \left( 1 + \rho \sum_{m=1}^M |h_m|^2 \right) \quad [\text{bits/s}]. \quad (8.13)$$

Để ý rằng dung lượng của các kênh truyền phân tập thu ở công thức (8.11) hoặc phát ở công thức (8.13) tăng theo qui luật logarithm theo số lượng ăng-ten phân tập. Điều này dẫn đến khả năng cho phép tăng dung lượng kênh khi tăng số ăng-ten, tuy nhiên, việc gia tăng sẽ sớm bị bão hòa khi số lượng ăng-ten lớn.

**Kênh MIMO:** đối với trường hợp kênh MIMO như mô tả ở Hình vẽ 8.2, chúng ta có mỗi quan hệ thu phát được biểu diễn bằng

phương trình hệ thống sau:

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{z}. \quad (8.14)$$

Sử dụng phương pháp SVD (Singular Value Decomposition) chúng ta có thể phân tích ma trận  $\mathbf{H}$  thành [2]:

$$\mathbf{H} = \mathbf{U}\mathbf{D}\mathbf{V}^H, \quad (8.15)$$

trong đó  $\mathbf{U}$  và  $\mathbf{V}$  là các ma trận đơn nhất (unitary)<sup>12</sup>, tức là  $\mathbf{U}\mathbf{U}^H = \mathbf{U}^H\mathbf{U} = \mathbf{I}_M$  và  $\mathbf{V}\mathbf{V}^H = \mathbf{V}^H\mathbf{V} = \mathbf{I}_N$ .  $\mathbf{D}$  là một ma trận đường chéo với các giá trị không âm ( $\geq 0$ ) trong đó các phần tử trên đường chéo là các giá trị căn bậc hai của các trị riêng (eigenvalue) của ma trận

$$\Phi = \begin{cases} \mathbf{H}\mathbf{H}^H & \text{nếu } M < N \\ \mathbf{H}^H\mathbf{H} & \text{nếu } M \geq N \end{cases}. \quad (8.16)$$

Thay  $\mathbf{H}$  ở (8.15) vào (8.14) chúng ta có

$$\mathbf{y} = \mathbf{U}\mathbf{D}\mathbf{V}^H\mathbf{s} + \mathbf{z}. \quad (8.17)$$

Nhân hai vế của phương trình trên với  $\mathbf{U}^H$  chúng ta thu được phương trình tương đương

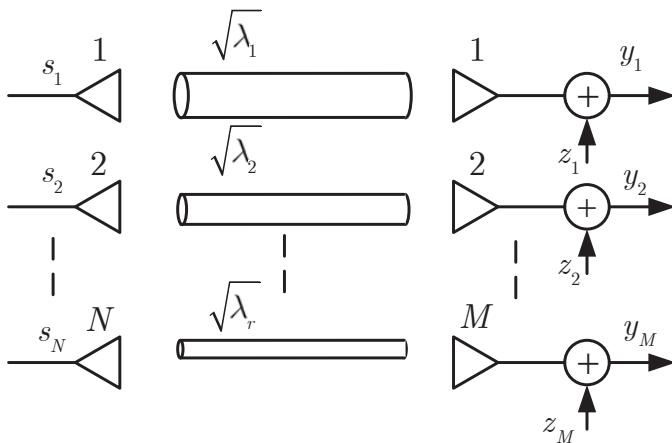
$$\mathbf{y}' = \mathbf{D}\mathbf{s}' + \mathbf{z}', \quad (8.18)$$

trong đó  $\mathbf{y}' \triangleq \mathbf{U}^H\mathbf{y}$ ,  $\mathbf{s}' \triangleq \mathbf{V}^H\mathbf{s}$ , và  $\mathbf{z}' \triangleq \mathbf{U}^H\mathbf{z}$ . Để ý rằng do  $\mathbf{D}$  là một ma trận đường chéo với  $r = \min(M, N)$  phần tử đầu tiên khác không, nên thông qua phép biến đổi SVD kênh MIMO đã được phân tích thành  $r$  kênh truyền song song hữu ích.  $N - r$  kênh còn lại không đóng vai trò hữu ích. Hình 8.6 minh họa mô hình kênh MIMO tương đương với các giá trị riêng tương ứng với biên độ của độ lợi của các kênh song song. Ký hiệu các trị riêng khác không của ma trận  $\Phi$  là  $\lambda_i$ , ta có thể biểu diễn công thức (8.18) ở dạng  $r$  kênh song song như sau:

$$y'_i = \sqrt{\lambda_i} s'_i + z'_i, \quad i = 1, 2, \dots, r. \quad (8.19)$$

---

<sup>12</sup>Từ điển Toán học 1976 dịch là ma trận unita



Hình 8.6: Mô hình kênh MIMO tương đương.

trong đó  $\sqrt{\lambda_i}$  biểu diễn biên độ của độ lợi (tăng ích) kênh truyền tương đương thứ  $i$ . Dung lượng kênh truyền MIMO, vì vậy, là tổng dung lượng của  $r$  kênh song song.

Giả sử sóng suất phát trên các ăng-ten phát là như nhau và được chuẩn hóa thành  $\frac{P_T}{N}$ . Do  $s'_i = \mathbf{V}^H(i,:)s = \mathbf{V}(:,i)s$  với  $\mathbf{A}(i,:)$  và  $\mathbf{A}(:,i)$  tương ứng biểu diễn một véc-tơ xây dựng từ hàng  $i$  hay cột  $i$  của ma trận  $\mathbf{A}$ . Do đó công suất thu tại kênh thứ  $i$ ,  $i \leq r$ , có thể tính được như sau:

$$\begin{aligned} P_{R_i} &= E\left\{|\sqrt{\lambda_i} \mathbf{V}(:,i)s|^2\right\} = \lambda_i \mathbf{V}^H(:,i) \underbrace{E\{\mathbf{s}^H \mathbf{s}\}}_{\frac{P_T}{N}r} \mathbf{V}(:,i) \\ &= \frac{\lambda_i P_T r}{N} \mathbf{V}(:,i) \mathbf{V}(:,i)^H = \frac{\lambda_i P_T r}{N}, \end{aligned} \quad (8.20)$$

trong đó  $\mathbf{V}(:,i)^H \mathbf{V}(:,i) = 1$  dựa trên tính chất của ma trận unitary.

Tương tự, công suất tạp âm ở kênh  $i$  được tính như sau:

$$\begin{aligned} P_{N_i} &= E\left\{|\mathbf{U}^H(i,:)z|^2\right\} = \mathbf{U}(i,:) \underbrace{E\{\mathbf{z}^H \mathbf{z}\}}_{r N_0} \mathbf{U}^H(i,:) \\ &= r \sigma_z^2 \mathbf{U}(i,:) \mathbf{U}^H(i,:) = r \sigma_z^2. \end{aligned} \quad (8.21)$$

Từ các công thức (??) và (8.21) chúng ta có thể tính được tỉ số

SNR ở kênh  $i$

$$\rho_i = \frac{\lambda_i P_T}{N\sigma_z^2}. \quad (8.22)$$

Thay  $\rho_i$  vào công thức Shannon chúng ta có

$$C_{\text{MIMO}} = W \sum_{i=1}^r \log_2 \left( 1 + \frac{\lambda_i P_T}{N\sigma_z^2} \right). \quad (8.23)$$

Công thức (8.23) chứng tỏ rằng dung lượng của một kênh MIMO bằng tổng dung lượng của  $r$  kênh SISO mà mỗi một kênh có độ tăng ích công suất là  $\lambda_i$  và công suất phát  $\frac{P_T}{N}$ . Điều này chứng tỏ rằng công nghệ MIMO cho phép một tuyến thông tin vô tuyến tạo ra được nhiều đường truyền tín hiệu khác nhau trong không gian nối liền giữa máy phát và máy thu. Do đó, kênh MIMO có hiệu suất phổ tần lớn hơn rất nhiều so với kênh SISO.

Sử dụng ma trận  $\Phi$ , công thức dung lượng kênh truyền MIMO có thể biểu diễn một cách tổng quát như sau [2, 3]:

$$C_{\text{MIMO}} = \begin{cases} W \log_2 \det \left( \mathbf{I}_r + \frac{\rho}{N} \mathbf{H} \mathbf{H}^H \right) [\text{bits/s}] & \text{nếu } M < N \\ W \log_2 \det \left( \mathbf{I}_r + \frac{\rho}{N} \mathbf{H}^H \mathbf{H} \right) [\text{bits/s}] & \text{nếu } M \geq N \end{cases} \quad (8.24)$$

Dựa vào công thức (8.24) các công trình nghiên cứu của Foschini và Gan năm 1988 [3] và Telatar năm 1999 [2] đã chứng minh được rằng dung lượng kênh truyền tăng tuyến tính theo  $r$ , tức là số ăng-ten tối thiểu sử dụng ở phía thu hay phát. Chứng minh này sẽ được mô tả kỹ ở phần tính toán dung lượng kênh truyền cho kênh Rayleigh ở phần sau.

### 8.2.2 Dung lượng kênh truyền pha-định Rayleigh

Trong phần trước, chúng ta đã tìm hiểu về dung lượng kênh truyền MIMO trong trường hợp kênh truyền cố định. Trong thực tế do tác động của pha-định, kênh truyền biến động theo thời gian và thường được mô hình hóa bằng các biến số ngẫu nhiên tuân theo phân bố

Rayleigh. Ma trận kênh truyền  $\mathbf{H}$  trong trường hợp này là một ma trận chứa các biến số ngẫu nhiên Gauss phức độc lập với giá trị trung bình 0 và phương sai 1, tức là,  $E\{|h_{mn}|^2\} = 1$ . Do kênh truyền  $\mathbf{H}$  là ngẫu nhiên, dung lượng của một kênh MIMO cũng là một đại lượng ngẫu nhiên. Vì vậy người ta thường sử dụng hai đại lượng thống kê thông dụng là *dung lượng ergodic* và *dung lượng outage* để đánh giá dung lượng của của một kênh MIMO. Trước khi đi vào khảo sát hai đại lượng thống kê này, chúng ta sẽ nghiên cứu dung lượng của các kênh MIMO sử dụng một số lượng đủ lớn các ăng ten phát và ăng ten thu.

### Dung lượng của một kênh MIMO với số lượng ăng-ten lớn

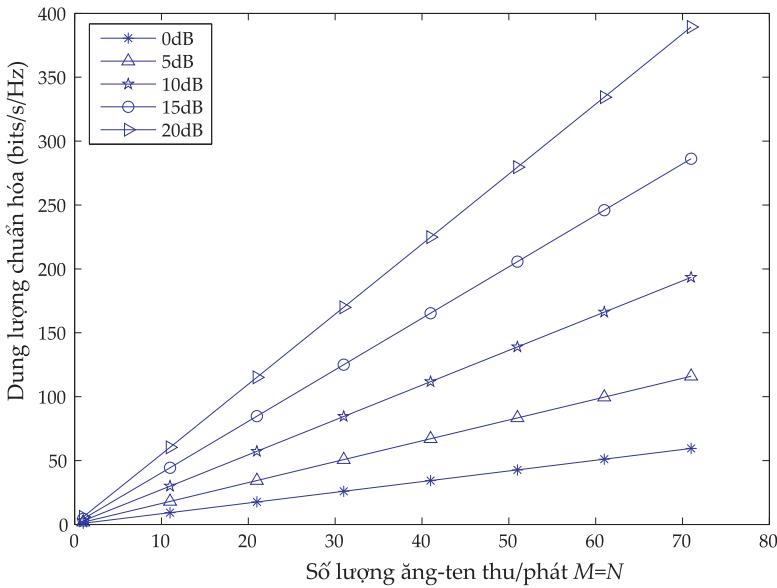
Để ý rằng do các phần tử của  $\mathbf{H}$  là các biến số ngẫu nhiên, nên nếu áp dụng qui luật số lớn chúng ta có [2]

$$\begin{aligned}\lim_{N \rightarrow \infty} \mathbf{H} \mathbf{H}^H &= N \mathbf{I}_M \\ \lim_{M \rightarrow \infty} \mathbf{H}^H \mathbf{H} &= M \mathbf{I}_N.\end{aligned}\tag{8.25}$$

Do đó, dung lượng kênh MIMO khi số lượng ăng ten lớn được cho bởi

$$\begin{aligned}C_{\text{MIMO}, N \rightarrow \infty} &= MW \log_2 (1 + \rho) \\ C_{\text{MIMO}, M \rightarrow \infty} &= NW \log_2 \left(1 + \rho \frac{M}{N}\right).\end{aligned}\tag{8.26}$$

Giả sử  $M = N$ , chúng ta thấy rõ ngay rằng dung lượng kênh truyền MIMO tăng tuyến tính theo số lượng ăng-ten tối thiểu sử dụng ở máy phát hay máy thu. Điều này được mô tả rõ thông qua kết mô phỏng máy tính ở Hình 8.7. Một nhận xét tiếp theo có thể thấy là nếu so sánh (8.26) với (8.9) chúng ta có thể thấy rằng dung lượng kênh truyền MIMO pha-định Rayleigh có thể đạt đến gấp  $r = \min(M, N)$  lần dung lượng một kênh truyền SISO cố định. Hai nhận xét này cho chúng ta thấy rõ tầm quan trọng của việc sử dụng kênh truyền MIMO trong thông tin vô tuyến.



**Hình 8.7:** Dung lượng Ergodic ứng với các cấu hình MIMO khác nhau.

### Dung lượng ergodic của một kênh MIMO

Dung lượng ergodic  $\bar{C}$  của một kênh MIMO là trung bình thống kê của dung lượng kênh trên tất cả các thực thể (realization) của  $\mathbf{H}$ . Dung lượng ergodic có ý nghĩa thực tiễn khi kênh truyền  $\mathbf{H}$  được rút ra từ một quá trình ngẫu nhiên ergodic và khi thời gian truyền tin đủ dài.

Khi máy thu biết hay ước lượng chính xác được ma trận kênh truyền  $\mathbf{H}$ , dung lượng ergodic được cho bởi

$$\bar{C}_{\text{MIMO}} = \begin{cases} E_{\mathbf{H}} \left\{ W \log_2 \det \left( \mathbf{I}_r + \frac{\rho}{N} \mathbf{H} \mathbf{H}^H \right) \right\} [\text{bit/s}] & \text{nếu } M < N \\ E_{\mathbf{H}} \left\{ W \log_2 \det \left( \mathbf{I}_r + \frac{\rho}{N} \mathbf{H}^H \mathbf{H} \right) \right\} [\text{bit/s}] & \text{nếu } M \geq N. \end{cases} \quad (8.27)$$

Cần chú ý rằng để đạt được dung lượng kênh MIMO nói trên, các phần tử  $h_{mn}$  của ma trận kênh  $\mathbf{H}$  cần là các biến Gauss phức và độc lập lẫn nhau. Điều này xảy ra trong trường hợp môi trường

truyền dẫn giữa máy phát và máy thu là một môi trường pha-định Rayleigh giàu tán xạ (uncorrelated rich scattering environment).

Dung lượng ergodic trong (8.27) có thể được xác định bằng phương pháp mô phỏng thống kê. Tuy nhiên phương pháp này đòi hỏi những phép tính về ma trận tồn tại nhiều thời gian, đặc biệt là khi số lượng ăng-ten lớn. Vì vậy Shin và các cộng sự [4] đã xây dựng các biểu thức toán học ngắn gọn để tính toán dung lượng ergodic cho kênh MIMO khi công suất phát được phân bố đồng đều trên các ăng ten. Như đã trình bày trong [4], dung lượng ergodic chuẩn hóa có thể được tính gần đúng bằng các biểu thức sau:

- Với  $N = M$ :

$$\begin{aligned} \overline{C}_{\text{MIMO}}/W &\approx e^{1/\rho} \log_2(e) E_1\left(\frac{1}{\rho}\right) + (N-1) \left\{ 2 \log_2(\sqrt{4\rho+1} + 1) \right. \\ &\quad \left. - \frac{\log_2(e)}{4\rho} (\sqrt{4\rho+1} - 1)^2 - 2 \right\} [\text{bit/s/Hz}] \end{aligned} \quad (8.28)$$

- Với  $N < M$

$$\begin{aligned} \overline{C}_{\text{MIMO}}/W &\approx e^{1/\rho} \log_2(e) \sum_{k=0}^{\lceil M/N \rceil - 1} E_{k+1}\left(\frac{1}{\rho}\right) \\ &\quad + (N-1) \left[ \frac{M}{N} \log_2 \left\{ 1 + \rho - \mathcal{J}\left(\rho, \frac{M}{N}\right) \right\} \right. \\ &\quad + \log_2 \left\{ 1 + \frac{\rho M}{N} - \mathcal{J}\left(\rho, \frac{M}{N}\right) \right\} \\ &\quad \left. - \frac{\log_2(e)}{\rho} \mathcal{J}\left(\rho, \frac{M}{N}\right) \right] [\text{bit/s/Hz}] \end{aligned} \quad (8.29)$$

- Với  $N > M$

$$\begin{aligned} \overline{C}_{\text{MIMO}}/W &\approx e^{\lceil N/M \rceil / \rho} \log_2(e) \sum_{k=0}^{\lceil N/M \rceil - 1} E_{k+1}\left(\frac{\lceil N/M \rceil}{\rho}\right) \\ &\quad + (M-1) \left[ \frac{N}{M} \log_2 \left\{ 1 + \frac{\rho M}{N} - \mathcal{J}\left(\frac{\rho M}{N}, \frac{N}{M}\right) \right\} \right. \\ &\quad + \log_2 \left\{ 1 + \rho - \mathcal{J}\left(\frac{\rho M}{N}, \frac{N}{M}\right) \right\} \\ &\quad \left. - \frac{N \log_2(e)}{M \rho} \mathcal{J}\left(\frac{\rho M}{N}, \frac{N}{M}\right) \right] [\text{bit/s/Hz}] \end{aligned} \quad (8.30)$$

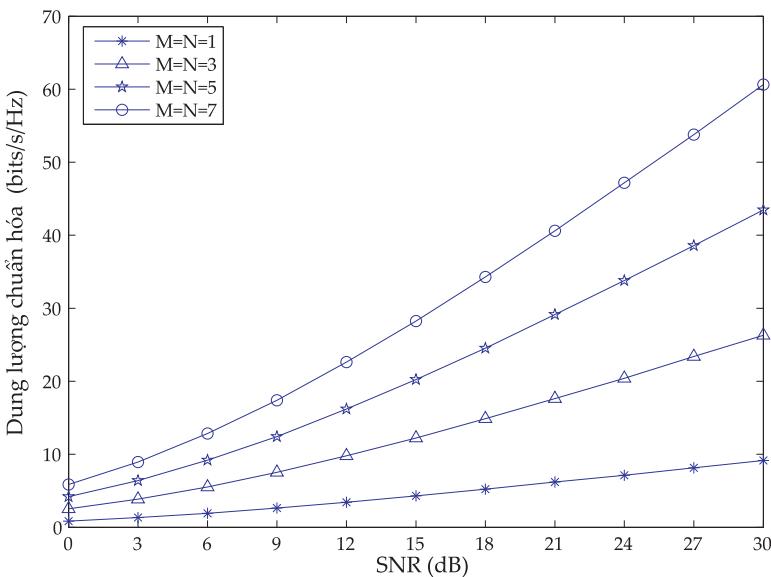
trong đó  $[a]$  biểu thị số nguyên nhỏ nhất lớn hơn hoặc bằng  $a$ ,

$$\mathcal{J}(u, v) = \frac{1}{4} \left\{ \sqrt{u(\sqrt{v} + 1)^2 + 1} - \sqrt{u(\sqrt{v} - 1)^2 + 1} \right\}^2 \quad (8.31)$$

và  $E_k(x)$  là tích phân hàm mũ bậc  $k$  được cho bởi [5]:

$$E_k(x) = \int_1^\infty \frac{e^{-xt}}{t^k} dt, \quad x > 0, k = 0, 1, \dots \quad (8.32)$$

Hình 8.8 trình bày mối quan hệ giữa dung lượng ergodic chuẩn hóa với SNR cho các cấu hình MIMO khác nhau. Chúng ta thấy rằng dung lượng ergodic tăng tỷ lệ với cả tỷ số tín hiệu trên tạp âm SNR và số lượng các ăng ten  $N = M$ . Số liệu cũng cho thấy dung lượng ergodic của một kênh MIMO lớn hơn rất nhiều so với một kênh SISO.



Hình 8.8: Dung lượng ergodic ứng với các giá trị SNR khác nhau.

Chú ý rằng từ công thức (8.27) hoặc các công thức (8.29) và (8.30) chúng ta dễ dàng tính được dung lượng của các kênh SIMO và MISO cho trường hợp pha-định Rayleigh bằng cách đặt tương ứng  $N = 1$  hay  $M = 1$ .

### Dung lượng outage của một kênh MIMO

Xét một kênh MIMO có ma trận  $H$  được chọn ngẫu nhiên theo phân bố Rayleigh khi bắt đầu truyền tin và không thay đổi cho mỗi lần sử dụng kênh truyền. Trong trường hợp này, dung lượng ergodic bằng 0 do kênh truyền được tạo ra từ một quá trình ngẫu nhiên phi ergodic. Đôi với một kênh truyền như vậy, dung lượng kênh là một biến ngẫu nhiên được mô tả bởi dung lượng outage.

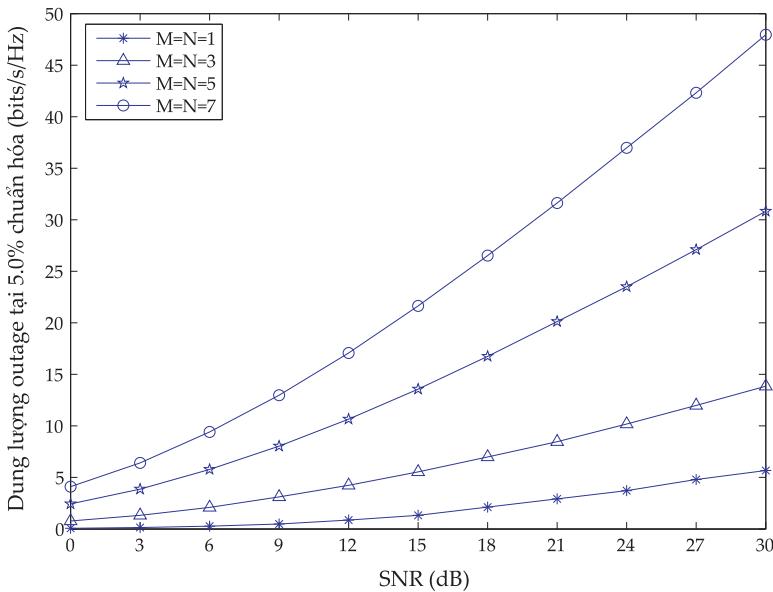
Theo định nghĩa, *dung lượng outage*  $q\%$ , viết tắt là  $C_{out,q}$ , của một kênh MIMO pha-định là tốc độ truyền tin được bảo đảm trong  $(100 - q)\%$  thực thể của kênh truyền. Nói cách khác, xác suất mà dung lượng kênh  $C$  nhỏ hơn hoặc bằng dung lượng outage  $C_{out,q}$  là  $q\%$ .

Dung lượng outage hàm ý rằng trong một kênh truyền phi ergodic, xác suất tồn tại một thực thể (phiên bản), mà thực thể này không thể đáp ứng được một tỷ số lỗi thấp bất kỳ, là khác 0 cho dù ta có sử dụng kỹ thuật mã hóa nào đi chăng nữa. Vì vậy khi kênh truyền là phi ergodic, ta cần phải có sự dung hòa giữa tốc độ truyền tin với xác suất outage hay xác suất lỗi bit.

Để xác định được dung lượng outage cho các cấu hình MIMO ta phải sử dụng phương pháp mô phỏng thống kê vì đến nay chưa có công thức toán học ngắn gọn để tính toán dung lượng outage. Minh họa trên Hình 8.9 là dung lượng outage 5% cho một số cấu hình MIMO khác nhau. Giống như đối với dung lượng ergodic, ở đây ta quan sát thấy rằng dung lượng outage tăng khi SNR tăng hoặc khi  $N = M$  tăng. Bên cạnh đó, kết quả mô phỏng cũng khẳng định rằng kênh truyền MIMO cải thiện đáng kể dung lượng outage so với kênh truyền SISO.

## 8.3 Các phương pháp truyền dẫn trên kênh truyền MIMO

Kết quả phân tích dung lượng kênh truyền MIMO đã thúc đẩy một làn sóng nghiên cứu các kỹ thuật truyền dẫn hiệu quả trên kênh



Hình 8.9: Dung lượng outage  $C_{out,5}/W$  ứng với các giá trị SNR khác nhau.

truyền MIMO. Tiếp theo công trình chung với Gan [3], Foschini đã đề xuất một hệ thống truyền dẫn theo từng lớp (layer) kết hợp với mã hóa nhằm đạt được dung lượng kênh truyền mong muốn như kết quả phân tích [6]. Tuy nhiên, kết quả nghiên cứu của Foschini [6] cho thấy dung lượng kênh truyền MIMO ở công thức (8.27) thực tế là một đường giới hạn trên (upper bound) có thể đạt được nhờ kết hợp các phương pháp mã hóa và thuật toán có độ phức tạp hay độ trễ không có giới hạn. Vì vậy, để có được các hệ thống truyền dẫn MIMO hiệu quả có thể ứng dụng trong thực tế, các công trình nghiên cứu về MIMO đã tập trung vào việc đề xuất các phương pháp truyền dẫn thỏa mãn được sự cân bằng giữa lợi ích thu được từ kênh MIMO và độ phức tạp cần thiết cho quá trình khôi phục tín hiệu ở phía thu. Các phương pháp truyền dẫn này có thể phân loại thành hai nhóm sau:

1. *Ghép kênh phân chia theo không gian (SDM: spatial division multiplexing):* phương pháp này tập trung vào việc tăng tốc độ

truyền dẫn bằng cách truyền đồng thời một loạt các luồng tín hiệu độc lập qua các ăng-ten phát và sử dụng các máy thu có độ phức tạp thấp để duy trì tỉ số lỗi bít cho phép. Phương pháp này cho phép thu được độ tăng ích ghép kênh (multiplexing gain) lớn.

2. *Mã không gian-thời gian (STC: space-time codes):* khác với phương pháp ghép kênh theo không gian, mã không gian-thời gian kết hợp việc mã hóa giữa các luồng tín hiệu để tối đa hóa độ tăng ích phân tập (diversity gain) nhằm giảm thiểu tỉ số lỗi bít (BER).

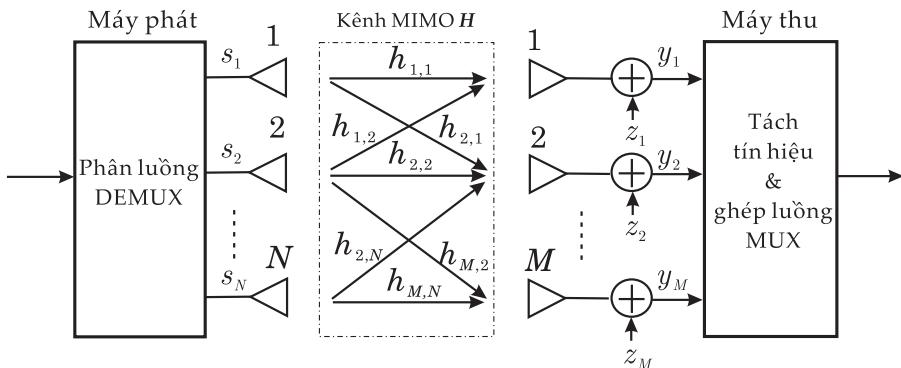
Trong phần còn lại của chương này chúng ta sẽ nghiên cứu phương pháp ghép kênh theo không gian, tập trung vào các phương pháp tách sóng được sử dụng ở máy thu. Phương pháp mã không gian-thời gian sẽ được trình bày cụ thể ở Chương 9.

## 8.4 Ghép kênh theo không gian

Nguyên lý của phương pháp phân kênh theo không gian rất đơn giản: ở máy phát (Tx) luồng tín hiệu phát được chia nhỏ (DEMUX: Demultiplexing) thành  $N$  luồng nhỏ  $s_n(t)$  và truyền đồng qua  $N$  ăng-ten phát. Tại máy thu các luồng tín hiệu sẽ được tách riêng ra rồi ghép lại (MUX) với nhau. Phương pháp phân kênh theo không gian này được mô tả ở Hình 8.10.

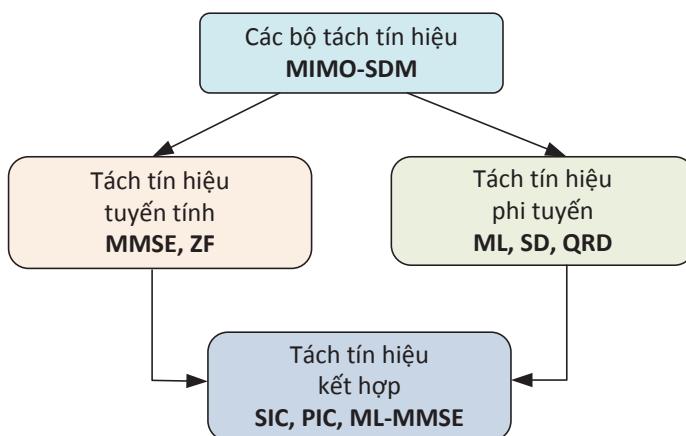
Do tín hiệu phát từ các ăng-ten phát khác nhau nên việc tách tín hiệu của mỗi luồng phát ở máy thu sẽ chịu ảnh hưởng nhiều đồng kênh từ các luồng còn lại. Vì vậy, máy thu cần sử dụng một bộ tách tín hiệu tốt có khả năng cung cấp tỉ số lỗi bít (BER) thấp, đồng thời lại không yêu cầu quá cao về độ phức tạp tính toán. Do máy phát sử dụng ở phương pháp phân kênh theo không gian này chỉ đơn thuần là một bộ phân kênh, các nghiên cứu về MIMO-SDM đều tập trung vào việc thiết kế bộ tách tín hiệu ở máy thu.

Dựa theo tính chất tuyến tính của phương pháp tách tín hiệu,



**Hình 8.10:** Phương pháp phân kênh theo không gian.

các bộ tách tín hiệu MIMO-SDM được phân loại thành hai nhóm lớn là các bộ **tách tín hiệu tuyến tính** và các bộ **tách tín hiệu phi tuyến** như ở Hình vẽ 8.11.



**Hình 8.11:** Phân loại các bộ tách tín hiệu MIMO-SDM.

Các bộ tách tín hiệu tuyến tính tiêu biểu bao gồm: bộ tách tín hiệu ZF (Zero-Forcing: *cưỡng bức bằng không*) và bộ tách tín hiệu MMSE (Minimum Mean-Square Error: *sai số bình phương trung bình tối thiểu*). Ưu điểm của các bộ tách tín hiệu tuyến tính là có độ phức tạp tính toán thấp và dễ thực hiện nhờ các thuật toán thích nghi phổ biến như LMS (Least Mean Square: *bình phương trung bình nhỏ nhất*), RLS (Recursive Least Square: *bình phương nhỏ nhất qui hồi*),....

Nhược điểm của các bộ tách tín hiệu tuyến tính là phẩm chất tách tín hiệu (tỉ số lỗi bít) đạt được tương đối thấp, đặc biệt là khi sử dụng số lượng ăng-ten phát lớn. Gần đây, nhờ việc áp dụng kết hợp với thuật toán lattice-reduction các bộ tách tín hiệu tuyến tính ZF và MMSE có thể đạt được tỉ số lỗi bít (BER) gần tối ưu, trong khi độ phức tạp tính toán hầu như không thay đổi. Xét một cách tổng quát thì vào thời điểm mà yêu cầu về độ tính toán thấp vẫn là quan trọng như hiện nay thì các bộ tách tín hiệu tuyến tính có ưu điểm hơn và, vì vậy, thường được áp dụng trong thực tế nhiều hơn.

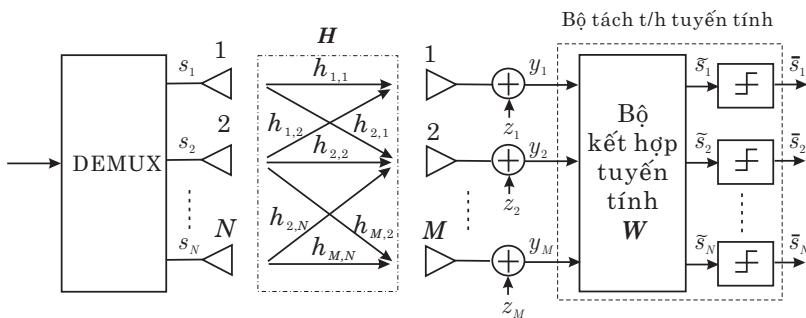
Ngược lại so với các bộ tách tín hiệu tuyến tính, các bộ tách tín hiệu phi tuyến có ưu điểm là có phẩm chất BER tốt hơn, nhưng lại chịu phải nhược điểm về độ phức tạp tính toán lớn. Trong các bộ tách tín hiệu phi tuyến, bộ tách tín hiệu ML (Maximum Likelihood: *hợp lệ cực đại*) là bộ tách tín hiệu tối ưu, tức là có phẩm chất BER tốt nhất. Tuy nhiên, độ phức tạp tính toán của bộ tách tín hiệu ML lại lớn nhất (biến thiên theo hàm mũ), vì vậy, bộ tách tín hiệu này rất ít được sử dụng trong thực tế. Gần đây, các nhà nghiên cứu đã đề xuất áp dụng thuật toán giải mã cầu (sphere decoding) vào bộ tách tín hiệu ML nhằm giảm độ phức tạp tính toán của nó đến mức độ có thể áp dụng được trong thực tế. Các bộ tách tín hiệu sử dụng thuật toán giải mã cầu, gọi tắt là các bộ tách tín hiệu SD, hiện tại đang là các bộ tách tín hiệu được đánh giá có triển vọng nhất do chúng có phẩm chất như một bộ tách tín hiệu tối ưu ML trong khi lại có độ phức tạp tính toán biến thiên theo đa thức. Ngoài bộ tách tín hiệu ML, các bộ tách tín hiệu phi tuyến khác như SIC (Successive Interference Cancellation: *triệt nhiễu nối tiếp*) hay PIC (Parallel Interference Cancellation: *triệt nhiễu song song*) đều sử dụng phương pháp kết hợp một bộ tách sóng tuyến tính với các phương pháp triệt nhiễu song song hoặc nối tiếp nhằm cải thiện phẩm chất BER trong khi vẫn tận dụng được độ tính toán thấp của các bộ tách tín hiệu tuyến tính.

Trong các phần tiếp theo, chúng ta sẽ nghiên cứu cụ thể các bộ

tách tín nói trên.

## 8.5 Các bộ tách tín hiệu tuyến tính

Sơ đồ cấu hình một bộ tách tín hiệu tuyến tính cho MIMO-SDM được mô tả ở Hình vẽ 8.12.



Hình 8.12: Sơ đồ bộ tách tín hiệu tuyến tính cho MIMO-SDM.

Phần cốt lõi của một tách tín hiệu tuyến tính là bộ kết hợp tuyến tính biểu diễn bởi ma trận trọng số (weight matrix)  $W$ . Dựa trên ma trận trọng số này, véc-tơ tín hiệu ước lượng được  $\hat{s}$  là kết quả của phép kết hợp (nhân) tuyến tính giữa véc-tơ tín hiệu thu  $y$  và ma trận trọng số  $W$

$$\hat{s} = W^H y. \quad (8.33)$$

Các giá trị ước lượng được  $\hat{s}$  này sau đó sẽ được đưa qua một bộ quyết định để lựa chọn đầu ra bộ tách tín hiệu

$$\bar{s} = Q\{\hat{s}\} = Q\{W^H y\} \quad (8.34)$$

trong đó  $Q\{\bullet\}$  biểu diễn toán tử quyết định hay lượng tử hóa. Trong trường hợp tín hiệu phát được điều chế bằng phương pháp BPSK thì toán tử quyết định tương đương với phép lấy dấu phần thực của  $\hat{s}$ , tức là

$$\bar{s} = \text{sign}\{\Re\{\hat{s}\}\} \quad (8.35)$$

trong đó  $\text{sign}\{\bullet\}$  và  $\Re\{\bullet\}$  biểu diễn tương ứng các toán tử lấy dấu và lấy phần thực của một số phức. Tùy thuộc vào phương pháp tìm ma trận trọng số  $\mathbf{W}$  chúng ta có các bộ tách tín hiệu tương ứng là ZF hay MMSE.

### 8.5.1 Bộ tách tín hiệu ZF

Bộ tách tín hiệu ZF còn có tên gọi là bộ tách tín hiệu LS (Least Square: *bình phương nhỏ nhất*). Bản chất của bộ tách tín hiệu LS là giả sử tạp âm bằng không rồi sử dụng phương pháp bình phương nhỏ nhất để tìm các tín hiệu phát  $s_n$ . Việc này tương đương với giải một hệ  $M$  phương trình với  $N$  ẩn số.

**Ma trận trọng số:** Hàm chi phí (cost function) để tìm  $s$  được định nghĩa như sau

$$\hat{s} = \arg \min_{\hat{s}} \left\{ \|\mathbf{y} - \mathbf{H}\hat{s}\|_2^2 \right\} \quad (8.36)$$

trong đó  $\|\cdot\|_2$  biểu diễn phép toán lấy chuẩn (norm) của véc-tơ hay ma trận. Tức là chúng ta cần tìm  $\hat{s}$  sao cho tối giản hóa giá trị bình phương sai số sau

$$\|\Delta_y\|_2^2 = \|\mathbf{y} - \mathbf{H}\hat{s}\|_2^2. \quad (8.37)$$

Khai triển  $\|\Delta_y\|_2^2$  chúng ta có

$$\|\Delta_y\|_2^2 = [\mathbf{y} - \mathbf{H}\hat{s}]^H [\mathbf{y} - \mathbf{H}\hat{s}] \quad (8.38a)$$

$$= \mathbf{y}^H \mathbf{y} - \hat{s}^H \mathbf{H}^H \mathbf{y} - \mathbf{y}^H \mathbf{H} \hat{s} + \hat{s}^H \mathbf{H}^H \mathbf{H} \hat{s}. \quad (8.38b)$$

Lấy đạo hàm của (8.38b) theo  $\hat{s}$  chúng ta có<sup>13</sup>

$$\frac{\partial \|\Delta_y\|_2^2}{\partial \hat{s}} = -\mathbf{y}^H \mathbf{H} + \hat{s}^H \mathbf{H}^H \mathbf{H}. \quad (8.39)$$

Đặt giá trị đạo hàm này bằng không, tức là  $\frac{\partial \|\Delta_y\|_2^2}{\partial \hat{s}} = 0$ , chúng ta tìm được

$$\hat{s} = (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H \mathbf{y}. \quad (8.40)$$

<sup>13</sup>Ở đây chúng ta sử dụng tính chất đạo hàm sau đây:  $\frac{\partial \mathbf{A}\mathbf{B}}{\partial \mathbf{A}} = \mathbf{B}^H$  và  $\frac{\partial \mathbf{A}^H \mathbf{B}}{\partial \mathbf{A}} = \mathbf{0}$

trong đó

$$\mathbf{H}^{\dagger} \triangleq (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H \quad (8.41)$$

được gọi là *ma trận giả đảo bên trái* (left pseudo-inverse) của  $\mathbf{H}$  do  $\mathbf{H}^{\dagger} \mathbf{H} = \mathbf{I}_N$ . Để ý rằng điều kiện đủ để chúng ta có thể tính được ma trận giả đảo bên trái là  $\text{rank}(\mathbf{H}) = N$  [7, Ch.17]. Hay nói cách khác  $N$  cột của ma trận  $\mathbf{H}$  cần phải độc lập tuyến tính với nhau. Điều kiện đủ là số hàng  $M$  của ma trận  $\mathbf{H}$  phải lớn hơn số cột  $N$ , tức là  $M \geq N$  [7, Ch.17]. Trong trường hợp đặc biệt khi  $M = N$ , phép đảo ma trận giả bên trái trùng với phép đảo ma trận thông thường. Điều này có nghĩa là bộ tách tín hiệu ZF chỉ có thể áp dụng được cho các hệ thống MIMO-SDM trong đó số ăng-ten thu nhiều hơn số ăng-ten phát.

Bỏ qua thành phần tạp âm  $z$  chúng ta có thể biểu diễn lại (8.40) như sau

$$\hat{s} = (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H \mathbf{H} s. \quad (8.42)$$

Do  $(\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H \mathbf{H} = \mathbf{I}_N$  là một ma trận đơn vị với  $N$  hàng  $N$  cột nên chúng ta thấy ngay rằng bộ tách tín hiệu ZF đã tách riêng ra từng tín hiệu phát  $s_n$  và loại bỏ hoàn toàn can nhiễu của tín hiệu từ các ăng-ten khác. Hay nói cách khác, can nhiễu từ các ăng-ten bên cạnh đã bị cưỡng bức bằng không (triệt tiêu). Vì vậy, ngoài tên gọi LS bộ tách sóng này còn có tên gọi ZF, hay *cưỡng bức bằng không*.

Kết hợp (8.33) và (8.40) chúng ta tìm được ma trận trọng số cho bộ tách tín hiệu ZF như sau

$$\mathbf{W} = \mathbf{H}^{\dagger H} = \mathbf{H}^H (\mathbf{H}^H \mathbf{H})^{-1}, \quad M \geq N \quad (8.43)$$

Do giá trị  $\mathbf{W}$  chỉ phụ thuộc vào ma trận kênh truyền  $\mathbf{H}$  nên máy thu chỉ cần ước lượng ma trận kênh truyền  $\mathbf{H}$  và sử dụng nó để tách các tín hiệu phát  $s_n$  ở phía thu.

Mặc dù bộ tách tín hiệu ZF chỉ áp dụng được cho các kênh truyền có số hàng  $M$  lớn hơn số cột  $N$ , trong một số trường hợp chúng ta vẫn mong muốn sử dụng một bộ tách tín hiệu tương tự

cho kênh truyền có  $N > M$ . Trong trường hợp đó chúng ta gấp phải bài toán giải một hệ phương trình có số phương trình ít hơn số ẩn số. Kết quả (8.40) không áp dụng được do ma trận  $\mathbf{H}^H \mathbf{H}$  trở nên gần đơn điệu (singular) và vì vậy không lấy nghịch đảo được. Tuy nhiên, sử dụng phương pháp SVD kết hợp với số nhân Lagrange [8] chúng ta thu có thể tìm được  $\hat{s}$  ở dạng tương tự

$$\hat{s} = \mathbf{H}^H (\mathbf{H} \mathbf{H}^H)^{-1} \mathbf{y}, \quad (8.44)$$

trong đó

$$\mathbf{H}^\ddagger \triangleq \mathbf{H}^H (\mathbf{H} \mathbf{H}^H)^{-1} \quad (8.45)$$

được gọi là *ma trận giả đảo bên phải* (right pseudo-inverse) của  $\mathbf{H}$ . Kết hợp (8.33) và (8.44), chúng ta tìm được ma trận trọng số cho trường hợp này là

$$\mathbf{W} = \mathbf{H}^{\ddagger H} = (\mathbf{H} \mathbf{H}^H)^{-1} \mathbf{H}, \quad M < N. \quad (8.46)$$

Trong MATLAB hàm `pinv` có thể áp dụng cho cả hai phép đảo ma trận giả bên phải và bên trái.

**Sai số bình phương trung bình (MSE):** Phương pháp thông thường để tính MSE là tính ma trận tương quan sai số (error covariance matrix) để tìm ra các giá trị  $MSE_n$  gắn với việc tách các ký hiệu phát  $s_n$  trên đường chéo chính. Ma trận tương quan sai số của bộ tách tín hiệu ZF được cho bởi [7, Ch.17]:

$$\begin{aligned} \mathbf{R}_{\Delta_{\hat{s}}} &= E\{\|\mathbf{s} - \hat{\mathbf{s}}\|^2\} = E\{[\mathbf{s} - \hat{\mathbf{s}}][\mathbf{s} - \hat{\mathbf{s}}]^H\} \\ &= \sigma_z^2 (\mathbf{H}^H \mathbf{H})^{-1}. \end{aligned} \quad (8.47)$$

Giá trị MSE gắn với tách dấu  $s_n$  là phần tử thứ  $n$  trên đường chéo của  $\mathbf{R}_{\Delta_{\hat{s}}}$  và bằng:

$$\begin{aligned} MSE_n &= \sigma_z^2 \mathbf{w}_n^H \mathbf{w}_n \\ &= \sigma_z^2 (\mathbf{h}_n^H \mathbf{h}_n)^{-1}, \end{aligned} \quad (8.48)$$

trong đó  $\mathbf{w}_n$  và  $\mathbf{h}_n$  biểu thị các véc-tơ cột thứ  $n$  của ma trận tương ứng  $\mathbf{W}$  và  $\mathbf{H}$ . Như vậy, giá trị MSE trung bình của phương pháp ZF

là

$$\begin{aligned}\overline{\text{MSE}} &= \frac{1}{N} \text{trace}\{\mathbf{R}_{\Delta_s}\} \\ &= \frac{1}{N} \sigma_z^2 \text{trace}\left\{\left(\mathbf{H}^H \mathbf{H}\right)^{-1}\right\}.\end{aligned}\quad (8.49)$$

Để ý rằng phần lớn độ phức tạp tính toán của bộ tách tín hiệu tập trung vào phép lấy nghịch đảo ma trận  $(\mathbf{H}^H \mathbf{H})^{-1}$  hoặc  $(\mathbf{H} \mathbf{H}^H)^{-1}$ . Vì vậy, độ phức tạp tính toán của bộ tách tín hiệu ZF tỉ lệ với hàm bậc ba của  $\min(M, N)$ , tức là,  $C_{\text{ZF}} \sim O(\min[M^3, N^3])$ .

Ưu điểm nổi bật của bộ tách tín hiệu ZF hay LS là đơn giản và có yêu cầu độ phức tạp tính toán thấp. Tuy nhiên, theo công thức (8.47) ta thấy rằng sai số giữa  $s$  và  $\hat{s}$  phụ thuộc vào công suất tạp âm và sẽ có giá trị lớn nếu các trị riêng của ma trận  $\mathbf{H}^H \mathbf{H}$  có giá trị nhỏ. Điều này chứng tỏ rằng bộ tách tín hiệu ZF chịu ảnh hưởng của hiệu ứng khuếch đại tạp âm (noise amplification). Vì vậy, nó thường thích hợp với các kênh truyền có tỉ số SNR cao.

### 8.5.2 Bộ tách tín hiệu MMSE

Khác với bộ tách tín hiệu ZF, ngoài đặc tính thống kê của tín hiệu từ các ăng-ten phát, bộ tách tín hiệu MMSE (Minimum Mean Square Error: sai số bình phương trung bình tối thiểu) còn xem xét đến cả đặc tính tạp âm tại các nhánh ăng-ten thu. Do đó bộ tách tín hiệu MMSE khắc phục được nhược điểm khuếch đại tạp âm của bộ tách tín hiệu ZF.

**Ma trận trọng số:** Hàm chi phí để tìm ma trận trọng số của bộ tách tín hiệu MMSE được định nghĩa như sau:

$$\mathbf{W} = \arg \min_{\mathbf{W}} E\left\{\|s - \mathbf{W}^H \mathbf{y}\|^2\right\} \quad (8.50)$$

tức là chúng ta cần tìm ma trận trọng số  $\mathbf{W}$  để tối giản hóa giá trị trung bình sai số bình phương giữa véc-tơ phát và véc-tơ ước lượng được

$$E\{\|\Delta_s\|^2\} = E\left\{\|s - \mathbf{W}^H \mathbf{y}\|^2\right\}. \quad (8.51)$$

Để tìm  $\mathbf{W}$  một cách dễ dàng, để ý rằng

$$\mathrm{E}\{\|\Delta_s\|^2\} = \mathrm{E}\{\mathrm{trace}(\mathbf{R}_{\Delta_s})\} \quad (8.52)$$

nên chúng ta sẽ tìm ma trận tương quan  $\mathbf{R}_{\Delta_s}$  của  $\Delta_s$  trước, sau đó tính  $\mathrm{E}\{\|\Delta_s\|^2\}$  sau. Từ định nghĩa ma trận tương quan chúng ta có

$$\begin{aligned} \mathbf{R}_{\Delta_s} &= [s - \mathbf{W}^H \mathbf{y}] [s - \mathbf{W}^H \mathbf{y}]^H \\ &= ss^H - \mathbf{W}^H \mathbf{y} s^H - s \mathbf{y}^H \mathbf{W} + \mathbf{W}^H \mathbf{y} \mathbf{y}^H \mathbf{W}. \end{aligned} \quad (8.53)$$

Để ý rằng

$$\mathrm{E}\{ss^H\} = \Lambda \quad (8.54)$$

$$\mathrm{E}\{\mathbf{y}s^H\} = \mathbf{H}\Lambda \quad (8.55)$$

$$\mathrm{E}\{\mathbf{y}\mathbf{y}^H\} = \mathbf{H}\Lambda\mathbf{H}^H + \sigma_z^2 \mathbf{I}_M \quad (8.56)$$

trong đó ma trận công suất  $\Lambda$  phát là một ma trận đường chéo với các phần tử trên đường chéo tương ứng với công suất phát từ các ăng-ten phát. Trong trường hợp MIMO-SDM thì do công suất phát trên các nhánh ăng-ten phát đều bằng nhau và bằng  $\zeta^2 = \frac{P_T}{N}$  nên chúng ta có  $\Lambda = \zeta^2 \mathbf{I}_M$ . Do đó chúng ta dễ dàng tính được

$$\begin{aligned} \mathrm{E}\{\|\Delta_s\|^2\} &= \mathrm{trace}(\mathrm{E}\{ss^H\} - \mathbf{W}^H \mathrm{E}\{\mathbf{y}s^H\} - \mathrm{E}\{s\mathbf{y}^H\}\mathbf{W} + \mathbf{W}^H \mathrm{E}\{\mathbf{y}\mathbf{y}^H\}\mathbf{W}) \\ &= \mathrm{trace}(\Lambda - \mathbf{W}^H \mathbf{H}\Lambda - (\mathbf{H}\Lambda)^H \mathbf{W} + \mathbf{W}^H (\mathbf{H}\Lambda\mathbf{H}^H + \sigma_z^2 \mathbf{I}_M) \mathbf{W}) \end{aligned}$$

Để tìm được  $\mathbf{W}$ , tương tự như trường hợp ZF, chúng ta lấy đạo hàm của  $\mathrm{E}\{\|\Delta_s\|^2\}$  theo  $\mathbf{W}$  rồi đặt giá trị đạo hàm đó bằng không. Sử dụng tính chất đạo hàm trace chúng ta có [9]

$$\frac{\partial \mathrm{E}\{\|\Delta_s\|^2\}}{\partial \mathbf{W}} = -\mathbf{H}\Lambda + (\mathbf{H}\Lambda\mathbf{H}^H + \sigma_z^2 \mathbf{I}_M) \mathbf{W}. \quad (8.57)$$

Đặt giá trị đạo hàm này bằng không chúng thu được

$$\begin{aligned} \mathbf{W} &= (\mathbf{H}\Lambda\mathbf{H}^H + \sigma_z^2 \mathbf{I}_M)^{-1} \mathbf{H}\Lambda \\ &= (\zeta^2 \mathbf{H}\mathbf{H}^H + \sigma_z^2 \mathbf{I}_M)^{-1} \mathbf{H}\zeta^2 \\ &= \left( \mathbf{H}\mathbf{H}^H + \frac{N}{\rho} \mathbf{I}_M \right)^{-1} \mathbf{H}. \end{aligned} \quad (8.58)$$

trong đó  $\rho = \frac{P_T}{\sigma_z^2}$  là SNR trên mỗi ăng ten thu.

**Sai số bình phương trung bình (MSE):** Thay (8.58) vào (8.53) chúng ta thu được ma trận tương quan lỗi

$$\begin{aligned} \mathbf{R}_{\Delta_s} &= \Lambda - (\mathbf{H}\Lambda)^H \mathbf{W} \\ &= \Lambda(\mathbf{I} - \mathbf{H}^H \mathbf{W}). \end{aligned} \quad (8.59)$$

Như vậy, giá trị MSE tối thiểu gắn với tách dấu  $s_n$  sử dụng phương pháp MMSE là

$$\text{MSE}_n = \zeta^2(1 - \mathbf{h}_n^H \mathbf{w}_n), \quad (8.60)$$

và giá trị trung bình tối thiểu của phương pháp MMSE là:

$$\overline{\text{MSE}} = \frac{1}{N} \text{trace} \left\{ \Lambda(\mathbf{I} - \mathbf{H}^H \mathbf{W}) \right\}. \quad (8.61)$$

**Biểu diễn phương pháp MMSE đồng dạng với ZF:** Phương pháp biểu diễn MMSE đồng dạng với MMSE được Hassibi đề xuất ở [10]. Sử dụng phép các phép biến đổi toán học ma trận trọng số  $\mathbf{W}$  ở (8.58) có thể biểu diễn lại thành [10]

$$\mathbf{W} = \mathbf{H} \left( \mathbf{H}^H \mathbf{H} + \frac{N}{\rho} \mathbf{I}_N \right)^{-1}. \quad (8.62)$$

Định nghĩa

$$\bar{\mathbf{H}} = \begin{bmatrix} \mathbf{H} \\ \sqrt{\frac{N}{\rho}} \mathbf{I}_N \end{bmatrix} \quad \text{và} \quad \bar{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} \quad (8.63)$$

thì phương pháp tách tín hiệu MMSE có thể được biểu diễn ở dạng tương tự ZF như sau

$$\hat{s} = \mathbf{W}^H \mathbf{y} = \bar{\mathbf{H}}^\dagger \bar{\mathbf{y}}. \quad (8.64)$$

Cách biểu diễn này cho phép chúng ta áp dụng qua lại hai phương pháp MMSE và ZF khi cần thiết.

Bộ tách tín hiệu MMSE có ưu điểm là đơn giản và dễ triển khai trong thực tế nhờ các thuật toán thích nghi như LMS, RLS,... Như đã đề cập ở phần trên, do bộ tách tín hiệu MMSE có tính đến đặc

tính của tạp âm nên khắc phục được nhược điểm khuếch đại tạp âm của bộ tách tín hiệu ZF. Vì vậy, phẩm chất BER hay SINR của bộ tách tín hiệu MMSE thường tốt hơn của bộ tách tín hiệu ZF. Ngoài ra, cũng giống như bộ tách tín hiệu ZF, bộ tách tín hiệu MMSE có độ phức tạp tính toán thấp. Do phần lớn độ tính toán đều tập trung vào phép tính đảo ma trận ở công thức (8.58) nên bậc phức tạp của bộ tách tín hiệu MMSE là  $O(M^3)$ .

### 8.5.3 Các tham số phẩm chất bộ tách tín hiệu tuyến tính

Từ công thức (8.33) chúng ta có véc-tơ tín hiệu ước lượng của véc-tơ tín hiệu phát  $s$  sử dụng các phương pháp tách tín hiệu tuyến tính là

$$\hat{s} = \mathbf{W}^H \mathbf{y}. \quad (8.65)$$

Do đó tín hiệu ước lượng được của ký hiệu phát đi từ ăng-ten phát  $n$  là

$$\hat{s}_n = \mathbf{w}_n^H \mathbf{y}. \quad (8.66)$$

Thay  $\mathbf{y}$  từ (8.14) vào (8.66) chúng ta có thể biểu diễn  $s_n$  như sau:

$$\begin{aligned} \hat{s}_n &= \mathbf{w}_n^H (\mathbf{H} s + \mathbf{z}) \\ &= \underbrace{\mathbf{w}_n^H \mathbf{h}_n s_n}_{\hat{s}_S} + \underbrace{\mathbf{w}_n^H \sum_{i=1, i \neq n}^N \mathbf{h}_i s_i}_{\hat{s}_I} + \underbrace{\mathbf{w}_n^H \mathbf{z}}_{\hat{s}_Z} \end{aligned} \quad (8.67)$$

Trong đó thành phần đầu tiên biểu diễn tín hiệu mong muốn, thành phần thứ hai biểu diễn nhiều đồng kênh CCI từ các ăng-ten còn lại, và thành phần cuối biểu diễn tạp âm, tại đầu ra bộ tách tín hiệu. Như vậy, phương sai (hay chính là công suất) của các thành phần tín hiệu tại đầu ra bộ tách tín hiệu tính được như sau

$$\sigma_S^2 = E\{\hat{s}_S^* \hat{s}_S\} = \mathbf{w}_n^H \mathbf{R}_{ss} \mathbf{w}_n \quad (8.68)$$

$$\sigma_I^2 = E\{\hat{s}_I^* \hat{s}_I\} = \mathbf{w}_n^H \mathbf{R}_{ii} \mathbf{w}_n \quad (8.69)$$

$$\sigma_Z^2 = E\{\hat{s}_Z^* \hat{s}_Z\} = \mathbf{w}_n^H \mathbf{R}_{zz} \mathbf{w}_n \quad (8.70)$$

trong đó

$$\mathbf{R}_{ss} = \zeta_n^2 \mathbf{h}_n \mathbf{h}_n^H \quad (8.71)$$

$$\mathbf{R}_{ii} = \sum_{i=1, i \neq n}^N \zeta_i^2 \mathbf{h}_i \mathbf{h}_i^H \quad (8.72)$$

$$\mathbf{R}_{zz} = \sigma_z^2 \mathbf{I} \quad (8.73)$$

tương ứng là các ma trận tương quan của tín hiệu mong muốn, nhiễu và tạp âm. Dựa vào kết quả tính toán này chúng ta có thể tính được các tỉ số tín hiệu trên tạp âm (SNR), tỉ số tín hiệu trên nhiễu (SIR: Signal to Interference Ratio), tỉ số tín hiệu trên tạp âm cộng với nhiễu (SINR: Signal to Interference plus Noise Ratio) như sau:

$$\text{SNR}_n = \frac{\sigma_S^2}{\sigma_Z^2} = \frac{\mathbf{w}_n^H \mathbf{R}_{ss} \mathbf{w}_n}{\mathbf{w}_n^H \mathbf{R}_{zz} \mathbf{w}_n} \quad (8.74)$$

$$\text{SIR}_n = \frac{\sigma_S^2}{\sigma_I^2} = \frac{\mathbf{w}_n^H \mathbf{R}_{ss} \mathbf{w}_n}{\mathbf{w}_n^H \mathbf{R}_{ii} \mathbf{w}_n} \quad (8.75)$$

$$\text{SINR}_n = \frac{\sigma_Z^2}{\sigma_I^2 + \sigma_Z^2} = \frac{\mathbf{w}_n^H \mathbf{R}_{ss} \mathbf{w}_n}{\mathbf{w}_n^H (\mathbf{R}_{ii} + \mathbf{R}_{zz}) \mathbf{w}_n}. \quad (8.76)$$

## 8.6 Các bộ tách tín hiệu phi tuyến

### 8.6.1 Bộ tách tín hiệu QRD

Bộ tách tín hiệu QR dựa trên phương pháp phân tích thừa số QR (QR factorization) của ma trận kênh truyền  $\mathbf{H}$ . Theo phương pháp phân tích thừa số QR thì bất kỳ một ma trận kênh truyền  $\mathbf{H} \in \mathbb{C}^{M \times N}$  nào với  $M \geq N$ <sup>14</sup> đều có thể phân tích thành

$$\mathbf{H} = \mathbf{Q} \mathbf{R} \quad (8.77)$$

---

<sup>14</sup>  $\mathbb{C}$  là trường số phức

trong đó  $\mathbf{R} \in \mathbb{C}^{N \times N}$  là một ma trận tam giác trên (upper triangular matrix) có dạng như sau

$$\mathbf{R} = \begin{bmatrix} R_{1,1} & R_{1,2} & \cdots & R_{1,N} \\ 0 & R_{2,2} & \cdots & R_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{N,N} \end{bmatrix} \quad (8.78)$$

còn  $\mathbf{Q} \in \mathbb{C}^{M \times N}$  là một ma trận đơn nhất (unitary matrix) có tính chất  $\mathbf{Q}^H \mathbf{Q} = \mathbf{Q}^{-1} \mathbf{Q} = \mathbf{I}$ . Từ phương trình hệ thống (8.14)

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{z} \quad (8.79)$$

sử dụng phương pháp QR và tính chất ma trận đơn nhất, nhân 2 vế của phương trình với  $\mathbf{Q}^H$  chúng ta có

$$\mathbf{Q}^H \mathbf{y} = \mathbf{R}\mathbf{s} + \mathbf{Q}^H \mathbf{z}. \quad (8.80)$$

Đặt  $\mathbf{y}' \triangleq \mathbf{Q}^H \mathbf{y}$  và  $\mathbf{z}' \triangleq \mathbf{Q}^H \mathbf{z}$  chúng ta có phương trình hệ thống mới

$$\mathbf{y}' = \mathbf{R}\mathbf{s} + \mathbf{z}'. \quad (8.81)$$

$$= \begin{bmatrix} R_{1,1} & R_{1,2} & \cdots & R_{1,N} \\ 0 & R_{2,2} & \cdots & R_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{N,N} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{bmatrix} + \begin{bmatrix} z'_1 \\ z'_2 \\ \vdots \\ z'_N \end{bmatrix} \quad (8.82)$$

trong đó  $\mathbf{z}'$  là véc-tơ chứa các thành phần tạp âm Gauss độc lập. Để ý rằng do  $\mathbf{R}$  là một ma trận tam giác trên nên phần tử thứ  $i$  của  $\mathbf{y}'$ , tức là  $y'_i$ , chỉ phụ thuộc vào các ký hiệu phát phía dưới, tức là  $s_j, j > i$ . Một cách tổng quát chúng ta có thể biểu diễn phần tử  $y'_i$  như sau [11]

$$y'_i = R_{i,i}s_i + \sum_{j=i+1}^N R_{i,j}s_i + z'_i. \quad (8.83)$$

trong đó phần tử thứ nhất là tín hiệu mong muốn, phần tử thứ hai bao gồm nhiều từ các ăng-ten khác, còn phần tử cuối cùng biểu diễn

tập âm. Do thành phần tín hiệu ở ăng-ten (lớp)<sup>15</sup> cuối cùng  $N$ ,  $y'_N$  không chịu ảnh hưởng của nhiều từ các ăng-ten khác nên được tách trước tiên, tiếp theo đến tín hiệu của các ăng-ten (lớp) phía trên. Khi tín hiệu ở các lớp dưới được khôi phục, ta giả thiết chúng chính là các ký hiệu đã được phát đi và sử dụng chúng để triệt tiêu khỏi tín hiệu thu ở các lớp trên. Sử dụng phương pháp quyết định cứng (hard decision), bỏ qua thành phần tạp âm chúng ta có thể biểu diễn ước lượng của ký hiệu phát  $s_i$  như sau [11]:

$$\bar{s}_i = \mathcal{Q} \left\{ \frac{y_i - \sum_{j=i+1}^N R_{i,j} s_j}{R_{i,i}} \right\}, i = 1, 2, \dots, N. \quad (8.84)$$

Toàn bộ nguyên lý bộ tách tín hiệu QR có thể được tóm tắt lại bằng thuật toán Bảng 8.1. Chương trình MATLAB MIMOQRDetect.m mô phỏng bộ tách tín hiệu QR được trình bày ở Program 8.3 của phần phụ lục chương.

**Bảng 8.1:** Thuật toán tách tín hiệu QR

- 
- ```

(1) BEGIN: Nhập  $\mathbf{y}$ ,  $\mathbf{H}$ ,  $N$ 
(2)  $[\mathbf{Q}, \mathbf{R}] = \text{QR}\{\mathbf{H}\}$ 
(3)  $\mathbf{y}' = \mathbf{Q}^H \mathbf{y}$ 
(4) For  $i = N$  to 1
    (5) If  $i \equiv N$  then  $\bar{s}_i = \mathcal{Q} \left\{ \frac{y'_i}{R_{i,i}} \right\}$ 
    (7) else  $\bar{s}_i = \mathcal{Q} \left\{ \frac{y'_i - \sum_{j=i+1}^N R_{i,j} \hat{s}_j}{R_{i,i}} \right\}$ 
    (8) End
    (9) End
(10) END: Xuất  $\hat{s}$ 

```
- 

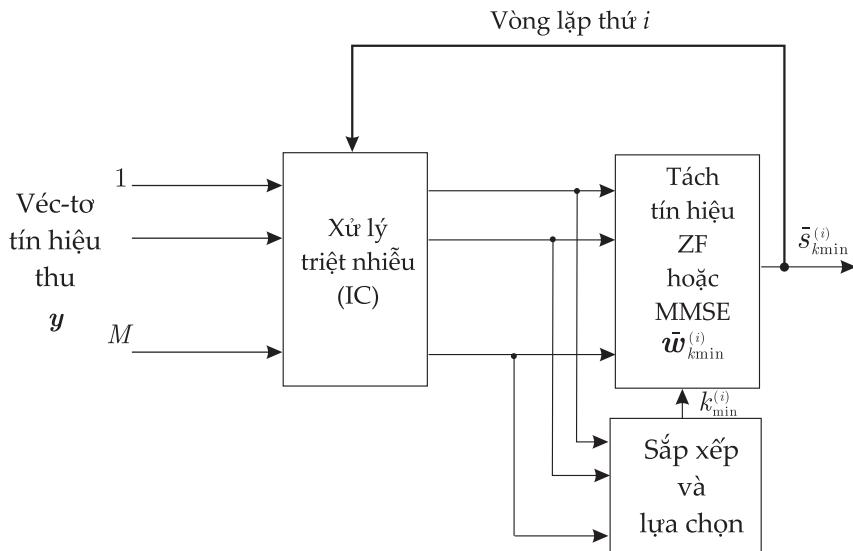
Phẩm chất BER của bộ tách tín hiệu QR cho hệ thống  $4 \times 4$  MIMO-SDM được so sánh với các bộ tách tín hiệu khác ở Hình

<sup>15</sup>Các ăng-ten phát thứ tự từ 1 đến  $N$  còn được gọi là các lớp  $1, 2, \dots, N$ , với ăng-ten 1 là lớp trên cùng, ăng-ten  $N$  là lớp cuối cùng.

vẽ 8.14. So với các bộ tách tín hiệu khác bộ tách tín hiệu QR có phẩm chất tương đối kém, chỉ hơn được bộ tách tín hiệu ZF. Tuy nhiên, bộ tách tín hiệu QR có ưu điểm đơn giản và không yêu cầu độ tính toán cao.

### 8.6.2 Bộ tách tín hiệu V-BLAST

Bộ tách tín hiệu V-BLAST (Vertical Bell-Labs Layered Space-Time) được phòng thí nghiệm Bell-Labs đề xuất cho các hệ thống MIMO-SDM năm 1998 tại [12] và [13]. Về bản chất bộ tách tín hiệu V-BLAST tương tự như các bộ tách tín hiệu sử dụng phương pháp triệt nhiễu nối tiếp (SIC:Successive Interference Cancellation) ở các hệ thống CDMA (Code Division Multiple Access). Nguyên lý của bộ tách tín hiệu V-BLAST cũng tương tự như bộ tách tín hiệu QR. Tức là, sử dụng phương pháp tách tín hiệu vòng lặp, trong đó tại mỗi vòng lặp sẽ chỉ có tín hiệu của một lớp được tách ra. Tín hiệu tách được của một lớp sau đó được phản hồi lại để triệt tiêu khỏi tín hiệu thu nhằm loại bỏ ảnh hưởng của nó đến quá trình tách tín hiệu của các lớp còn lại. Tuy nhiên, khác với phương pháp QR, bộ tách tín hiệu V-BLAST không sử dụng phương pháp phân tích QR nên tín hiệu của lớp được tách trước bị ảnh hưởng của tất cả các lớp còn lại nhiều hơn so với các lớp tách sau. Giả sử, tín hiệu tách được ở các lớp là chính xác thì sau khi phản hồi lại, thành phần của chúng sẽ được triệt tiêu hoàn toàn khỏi tín hiệu thu và như vậy, tín hiệu của lớp tách sau cùng sẽ không bị ảnh hưởng nhiều từ các lớp khác. Giải thích theo kỹ thuật xử lý tín hiệu không gian thì độ tự do được tăng dần theo vòng lặp, và các lớp tín hiệu được tách sau sẽ có lợi thế về độ tự do của mảng ăng-ten thu. Kết quả là chúng sẽ có phẩm chất lỗi bit tốt hơn. Về mặt trực giác, chúng ta có thể thấy rằng các lớp tín hiệu “khoẻ” nên được tách trước, các lớp “yếu” nên được tách sau. Trong thông tin, điều này tương đương với việc chọn ra các lớp có tỉ số SINR cao để tách trước. Do tỉ số SINR tỉ lệ nghịch với sai



**Hình 8.13:** Sơ đồ nguyên lý hoạt động của bộ tách tín hiệu V-BLAST.

số bình phương trung bình (MSE)<sup>16</sup>, nên thay cho việc tìm ra lớp có  $\text{SINR}_{\max}$  các tác giả của V-BLAST đã đề xuất sử dụng phương pháp tìm  $\text{MSE}_{\min}$  từ ma trận kênh truyền có thể tính toán được một cách dễ dàng [12],[13]. Như vậy, tại mỗi vòng lặp bộ tách tín hiệu V-BLAST tính toán sai số bình phương trung bình (MSE) gắn với việc tách tín hiệu của từng lớp rồi chọn ra lớp tương ứng với  $\text{MSE}_{\min}$  để tách. Sau khi loại bỏ thành phần của tín hiệu lớp đã được tách ra, quá trình này lại được lặp lại cho đến khi tín hiệu của tất cả các lớp đã được tách xong. Để giảm bớt độ phức tạp tính toán, việc tách tín hiệu của từng lớp được tiến hành dựa trên các phương pháp tuyến tính MMSE hay ZF. Vì vậy, bộ tách tín hiệu V-BLAST còn được gọi là bộ tách tín hiệu kết hợp MMSE-SIC hay MMSE-ZF. Nguyên lý hoạt động của bộ tách tín hiệu V-BLAST được trình bày tóm tắt dưới đây.

Biểu diễn  $\{k_1, k_2, \dots, k_N\}$  là một tập trình tự sắp xếp lại của tập các chỉ số ăng-ten phát  $1, 2, \dots, N$  xác định thứ tự tách các dấu phát  $s_i$  thuộc  $s$ . Tại vòng lặp thứ  $i$ , bộ tách tín hiệu V-BLAST tính toán sai

<sup>16</sup>Cụ thể là:  $\text{MSE}_n = \frac{\zeta_n^2}{1+\text{SINR}_n}$

**Bảng 8.2:** Thuật toán tách tín hiệu V-BLAST

- (1) BEGIN: Nhập  $\mathbf{y}, \mathbf{H}, N, \rho$
- (2) For  $i = 1$  to  $N$
- (3)      $\mathbf{G} = \mathbf{H}^\dagger$
- (4)      $\{\text{MSE}_j\} = \text{diag}\{\mathbf{G}\mathbf{G}^H\}$
- (5)      $k_{\min} = \arg \min_j \{\text{MSE}_j\}$
- (6)     If ‘ZF’ then  

$$\mathbf{w}_{k_{\min}} = \mathbf{g}_{k_{\min}}^H$$
- (7)     Elseif ‘MMSE’ then  

$$\mathbf{w}_{k_{\min}} = \left( \mathbf{H}\mathbf{H}^H + \frac{N}{\rho} \mathbf{I} \right)^{-1} \mathbf{h}_{k_{\min}}$$
- (8)     End
- (6)      $\bar{s}_{k_{\max}} = \mathcal{Q}\{\mathbf{w}_{\max}^H \mathbf{y}\}$
- (9)      $\mathbf{y} = \mathbf{y} - \zeta \mathbf{h}_{k_{\min}} \hat{s}_{\min}$
- (10)     $\mathbf{h}_{k_{\min}} = [ ]$  (Xóa cột thứ  $k_{\min}$  khỏi  $\mathbf{H}$ )
- (11)     $s_{k_{\min}} = [ ]$  (Xóa phần tử thứ  $k_{\min}$  khỏi  $\mathbf{s}$ )
- (12)    End
- (13) END: Xuất  $\bar{s}$

số MSE tương ứng với tất cả các tín hiệu phát  $s_{k_i}$  như sau

$$\{\text{MSE}_j\} = \text{diag}\left\{\mathbf{H}^\dagger (\mathbf{H}^\dagger)^H\right\}, j = 1, 2, \dots, N - 1 + i \quad (8.85)$$

và chọn ra lớp có MSE nhỏ nhất, tức là

$$k_i = \arg \min_j \{\text{MSE}_j, j = 1, 2, \dots, N - 1 + i\}. \quad (8.86)$$

Ở đây  $\text{diag}\{\bullet\}$  biểu diễn phép toán lấy các phần tử đường chéo của ma trận bên trong dấu ngoặc;  $\text{MSE}_j$  được tính toán theo công thức (8.48) nếu phương pháp ZF được dùng và (8.60) nếu MMSE được sử dụng. Sau khi đã xác định được lớp  $k_i$  với  $\text{MSE}_{\min}$ , bộ tách tín hiệu sử dụng phương pháp tách tín hiệu tuyến tính để tách ra tín

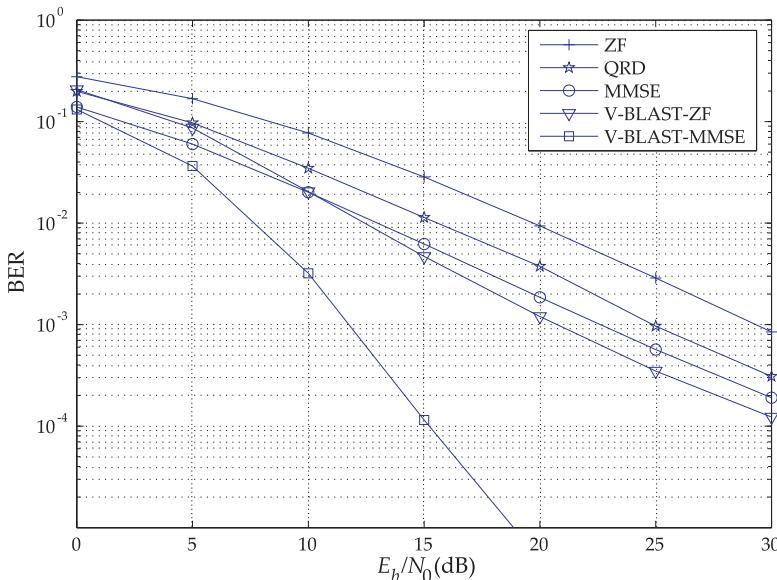
hiệu tương ứng với lớp đã xác định được

$$\bar{s}_{k_i} = \mathcal{Q}\left\{\mathbf{w}_{k_i}^H \mathbf{y}^{(i)}\right\} \quad (8.87)$$

trong đó  $\mathbf{w}_{k_i}$  là cột thứ  $k_i$  của ma trận trọng số  $\mathbf{W}$  được tính toán bằng phương pháp ZF ở công thức (8.43) hay bằng phương pháp MMSE ở công thức (8.58). Trong kỹ thuật xử lý ăng-ten thích nghi, thao tác này còn được gọi là “nulling”. Tiếp theo, tín hiệu tách được  $\bar{s}_{k_i}$  được phản hồi lại bộ tách tín hiệu tuyến tính để triệt tiêu (cancellation) khỏi véc-tơ tín hiệu thu ở vòng lặp tách tín hiệu tiếp theo:

$$\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} - \bar{s}_{k_i} \mathbf{h}_{k_i} \quad (8.88)$$

trong đó  $\mathbf{h}_{k_i}$  là véc-tơ được xây dựng từ cột  $k_i$  của ma trận  $\mathbf{H}$ . Các thao tác “nulling and cancellation” này được lặp lại cho đến khi tín hiệu của tất cả các ăng-ten phát (lớp) đã được tách xong. Thuật toán tách tín hiệu V-BLAST được tóm tắt ở Bảng 8.2 và chương trình mẫu MATLAB VBLASTDetect.m mô phỏng BER của bộ tách tín hiệu V-BLAST được trình bày ở Program 8.4.



Hình 8.14: Phẩm chất của các bộ tách tín hiệu cho hệ thống  $4 \times 4$  MIMO-SDM.

So với các bộ tách tín hiệu tuyến tính MMSE hay ZF, bộ tách tín hiệu V-BLAST yêu cầu khối lượng tính toán lớn hơn. Tuy nhiên, lượng tính toán gia tăng này có thể chấp nhận được. Ngoài ra, nếu xét về bậc tính toán thì bộ tách tín hiệu V-BLAST cũng có cùng bậc tính toán giống như của các bộ tách tín hiệu tuyến tính, tức là  $O(M^3)$ . Đổi lại cho phần gia tăng tính toán, phẩm chất BER của bộ tách tín hiệu V-BLAST được cải thiện hơn rất nhiều so với các bộ tách tín hiệu tuyến tính (Xem Hình vẽ 8.14). Đặc biệt chúng ta có thể nhận thấy rằng bộ tách tín hiệu V-BLAST sử dụng phương pháp MMSE cho phẩm chất BER vượt trội rất nhiều so với các bộ tách tín hiệu khác. Ưu điểm về phẩm chất BER tốt kết hợp với độ phức tạp tính toán thấp làm cho bộ tách tín hiệu V-BLAST trở thành một ứng cử viên sáng giá để tách tín hiệu trong các hệ thống MIMO-SDM.

### 8.6.3 Bộ tách tín hiệu có trợ giúp của phương pháp rút gọn cơ sở dàn

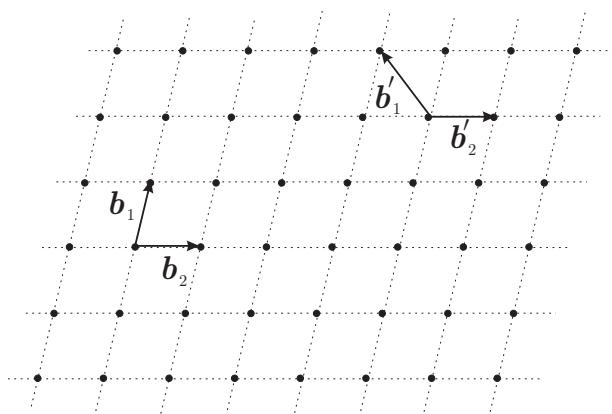
#### 8.6.3.a. Cơ sở phương pháp rút gọn cơ sở dàn

**Định nghĩa [14][15]:** Giả sử  $M$  và  $N$  là hai số nguyên dương với  $N \leq M$ . Một tập hợp con  $\mathcal{L}$  của tập hợp số thực  $\mathbb{R}^M$  được gọi là một dàn<sup>17</sup> có kích thước  $M$  nếu tồn tại  $N$  véc-tơ với độ dài  $M$  độc lập tuyến tính với nhau sao cho

$$\mathcal{L} = \mathcal{L}\{\mathbf{B}\} = \left\{ \sum_{i=1}^N \lambda_i \mathbf{b}_i, \lambda_i \in \mathbb{Z} \right\} \quad (8.89)$$

trong đó  $\mathbf{B}$  là một ma trận có kích thước  $M \times N$ . Tập hợp các véc-tơ cột  $\{\mathbf{b}_i, i = 1, 2, \dots, N\}$  và ma trận  $\mathbf{B}$  tương ứng được gọi là cơ sở và ma trận sinh (generator matrix) của  $\mathcal{L}$ .

Do  $\lambda_i$  là các số nguyên nên một dàn (lattice) chính là một tổ hợp tuyến tính nguyên của các véc-tơ cơ sở [14]. Nói chung, một dàn có thể có nhiều cơ sở nhưng chúng đều có chung định thức. Hình vẽ 8.15 biểu diễn một ví dụ dàn 2 chiều gồm các điểm giao của dàn. Chúng ta có thể thấy rằng cả 2 tập hợp véc-tơ  $\{\mathbf{b}_1, \mathbf{b}_2\}$  và  $\{\mathbf{b}'_1, \mathbf{b}'_2\}$  đều có thể coi là các cơ sở của dàn.



Hình 8.15: Biểu diễn một dàn 2 chiều.

<sup>17</sup>Nguyên gốc tiếng Anh: lattice. Dịch theo “Từ điển Toán học Anh-Việt,” NXB Khoa học-Kỹ Thuật, 1976.

Mục đích của phương pháp rút gọn cơ sở dàn (lattice basis reduction) hay còn gọi ngắn gọn là phương pháp rút gọn dàn (LR: Lattice Reduction) là biến đổi một cơ sở cho trước  $\mathbf{B}$  thành một cơ sở mới  $\mathbf{B}'$  có các cột gần trực giao với nhau. Điều này cũng tương đương với việc tạo nên cơ sở  $\mathbf{B}'$  mới có các véc-tơ cơ sở ngắn nhất. Có nhiều phương pháp rút gọn dàn khác nhau đã được nghiên cứu, trong đó phương pháp phổ biến nhất là phương pháp LLL do Lenstra, Lenstra and Lovász đề xuất năm 1982 [16]. Phương pháp LLL cho phép tạo nên một cơ sở dàn rút gọn từ một cơ sở dàn bất kỳ với độ phức tạp tính toán ở bậc đa thức (polynomial complexity). Nền tảng của phương pháp rút gọn cơ sở dàn LLL dựa trên thuật toán trực giao hoá của Gram-Schmidt [17].

Theo phương pháp Gram-Schmidt thì chúng ta có thể thu được cơ sở trực giao  $\mathbf{B}'$  từ một cơ sở cho trước  $\mathbf{B}$  thông qua vòng lặp sau

$$\mathbf{b}'_i = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{ij} \mathbf{b}'_j, \quad (8.90)$$

trong đó

$$\mu_{ij} = \frac{\mathbf{b}'_j^H \mathbf{b}_i}{\|\mathbf{b}'_j\|_2^2} \quad (8.91)$$

biểu diễn độ dài của hình chiếu của  $\mathbf{b}_i$  lên  $\mathbf{b}'_j$ . Thao tác ở công thức (8.90) có nghĩa là loại bỏ các thành phần  $\mathbf{b}'_1, \dots, \mathbf{b}'_{i-1}$  từ  $\mathbf{b}_i$  làm cho  $\mathbf{b}'_i$  trực giao với các véc-tơ cơ sở khác. Cơ sở  $\mathbf{B}' = [\mathbf{b}'_1, \mathbf{b}'_2, \dots, \mathbf{b}'_N]$  được gọi là rút gọn LLL nếu phân tích Gram-Schmidt của nó thoả mãn điều kiện

$$|\mu_{ij}| \leq \frac{1}{2} \quad \text{for } 1 \leq j < i \leq N, \quad (8.92)$$

và

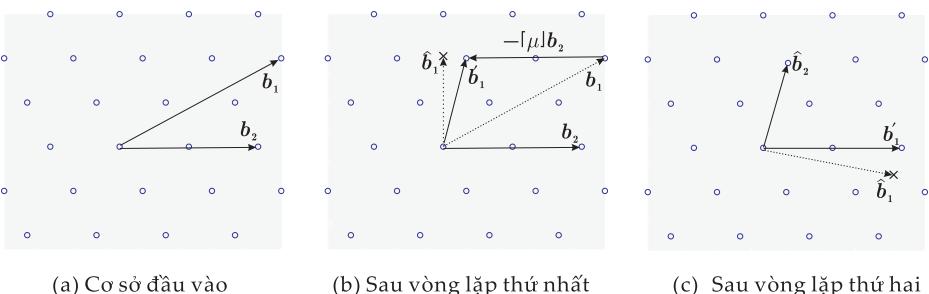
$$\delta \|\mathbf{b}'_{i-1}\|_2^2 \leq \|\mathbf{b}'_i + \mu_{i,j-1} \mathbf{b}'_{i-1}\|_2^2 \quad (8.93)$$

với bất kỳ  $i \in \{2, \dots, N\}$  nào và  $\frac{1}{4} < \delta \leq 1$ . Thông thường  $\delta = 3/4$  hay được sử dụng ở trong thuật toán LLL [16].

Do thuật toán trực giao hoá Gram-Schmidt làm thay đổi dàn nên chúng ta không thể áp dụng trực tiếp cho phép toán rút gọn cơ sở dàn được mà cần phải có sửa đổi. Cách đề xuất của thuật toán LLL là thay thế hệ số Gram-Schmidt  $\mu$  bằng một số nguyên gần nhất, tức là,  $[\mu_{ij}]$ , trong đó  $[.]$  biểu diễn phép toán làm tròn riêng biệt cho phần thực và phần ảo. Phương pháp này tạo nên phép rút gọn cơ sở “yếu” (weak basis reduction) mà không hề làm thay đổi dàn. Sử dụng phép rút gọn cơ sở “yếu” thuật toán trực giao hoá Gram-Schmidt ở công thức (8.90) trở thành

$$\mathbf{b}'_i = \mathbf{b}_i - \sum_{j=1}^{i-1} [\mu_{ij}] \mathbf{b}_j, \quad (8.94)$$

và thoả mãn điều kiện (8.92). Tuy nhiên, chỉ riêng phép rút gọn cơ sở yếu ở công thức (8.94) chưa đảm bảo được các véc-tơ cơ sở có được độ dài ngắn nhất do các véc-tơ ở phía trước ở trong dãy véc-tơ  $\{\mathbf{b}'_1, \mathbf{b}'_2, \dots, \mathbf{b}'_N\}$  có thể dài hơn các véc-tơ ở phía sau. Để tránh trường hợp này, thuật toán LLL thực hiện việc kiểm tra điều kiện (8.93) cho từng  $i$ . Nếu tồn tại một giá trị  $i$  vi phạm điều kiện kiểm tra ở (8.93) thì thuật toán thực hiện phép hoán vị  $\mathbf{b}'_i$  và  $\mathbf{b}'_j$ . Sau khi hoán vị, phép rút gọn yếu lại được lặp lại để đảm bảo hai véc-tơ có được độ dài ngắn nhất so với nhau. Thuật toán lặp lại phép kiểm tra và rút gọn yếu này cho đến khi không thể thực hiện rút gọn hơn nữa.



**Hình 8.16:** Ví dụ biểu diễn thao tác của thuật toán LLL trên một lưới 2 chiều.

Hình vẽ 8.16 trình bày một ví dụ về các thao tác của thuật toán LLL cho một cơ sở 2 chiều  $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2]$ . Hai véc-tơ có sẵn ban đầu  $\mathbf{b}_1$ ,

**Bảng 8.3:** Thuật toán rút gọn cơ sở dàn LLL

```

(1) BEGIN: Nhập  $\mathbf{B}$ , đặt  $i := 2$ ,  $\beta = 3/4$ , và  $\mathbf{T} = \mathbf{I}_M$ 
(2)      while  $i \leq N$ 
(3)          for  $j := i - 1, \dots, 1$ 
(4)              tính  $\mu_{pq}$  sử dụng (8.91)
(5)               $\mathbf{b}_i := \mathbf{b}_i - \lceil \mu_{ij} \rceil \mathbf{b}_j$ 
(6)               $\mathbf{T}_i := \mathbf{T}_i - \lceil \mu_{ij} \rceil \mathbf{T}_j$ 
(7)          end
(8)          Cập nhật  $\mathbf{b}'_i, \mu_{i,1}, \dots, \mu_{i,j-1}$  sử dụng (8.90), (8.91)
(9)          if  $\delta \|\mathbf{b}'_{i-1}\|^2 \leq \|\mathbf{b}'_i + \mu_{i,i-1} \mathbf{b}'_{i-1}\|^2$ 
(10)             Hoán đổi các cột  $i-1$  và  $i$  ở  $\mathbf{B}$  và  $\mathbf{T}$ .
(11)              $i := \max\{i-1, 2\}$ 
(12)         else
(13)              $i := i + 1$ 
(14)         end
(15)     end
(16) END: Xuất  $\mathbf{B}' = \mathbf{B}$  và  $\mathbf{T}$ .

```

---

$\mathbf{b}_2$  được biểu ở Hình 8.16(a). Ở vòng lặp đầu tiên, thuật toán Gram-Schmid tìm được véc-tơ  $\hat{\mathbf{b}}_1$  trực giao với  $\mathbf{b}_2$ . Tuy nhiên, chúng ta có thể thấy rõ rằng véc-tơ  $\hat{\mathbf{b}}_1$  nằm ngoài dàn. Phép rút gọn yếu sử dụng  $\lceil \mu_{ij} \rceil$  để kéo  $\hat{\mathbf{b}}_1$  về điểm dàn gần nhất tạo nên véc-tơ  $\mathbf{b}'_1$ . Ở vòng lặp tiếp theo thuật toán LLL thực hiện phép hoán đổi 2 véc-tơ  $\mathbf{b}_1, \mathbf{b}_2$  với nhau rồi thực hiện tiếp tục phép rút gọn yếu để tìm ra véc-tơ  $\mathbf{b}'_1$  gần trực giao với véc-tơ  $\mathbf{b}_2$ . Do hai véc-tơ tìm được  $\mathbf{b}_1, \mathbf{b}_2$  sau vòng lặp thứ hai không thể rút gọn hơn được nữa nên thuật toán LLL dừng tại vòng lặp này.

Thực tế thì toàn bộ thuật toán rút gọn cơ sở dàn có thể được mô tả bằng một phép biến đổi tuyến tính sau  $\mathbf{B}' = \mathbf{BT}$ , trong đó ma

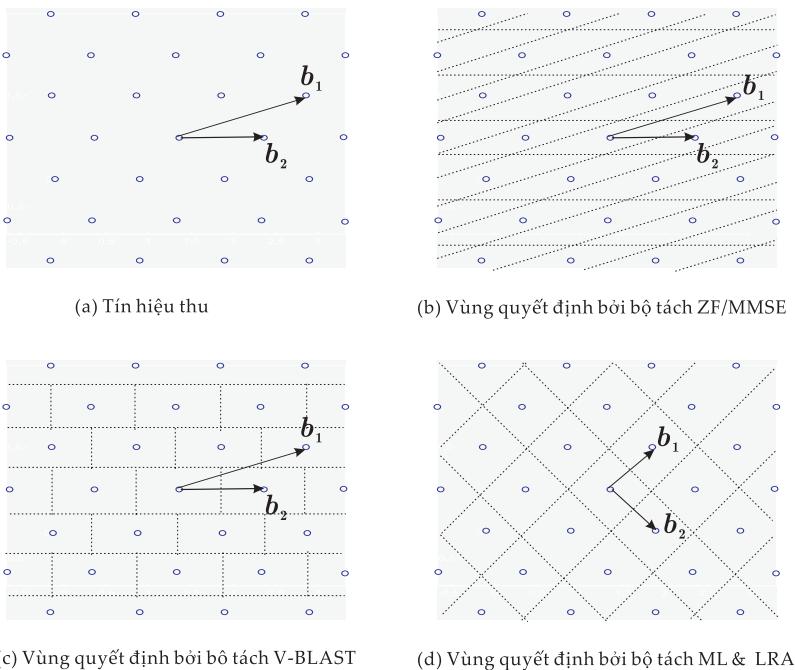
trận chuyển đổi  $T$  là một ma trận đơn (unimodular) có định thức  $\det\{T\} = \pm 1$ . Ở trong các bộ tách tín hiệu có trợ giúp rút gọn dàn (LRA: Lattice Reduction Aided), ma trận chuyển đổi  $T$  được dùng để lượng tử hoá các ước lượng tín hiệu về chùm sao tín hiệu.

Thuật toán LLL có sửa đổi cho phép nhận được cả ma trận chuyển đổi  $T$  được trình bày tóm tắt ở Bảng 8.3.

### *3.6.3.b. Các bộ tách tín hiệu có trợ giúp rút gọn cơ sở dàn*

Động cơ áp dụng phương pháp rút gọn cơ sở dàn (LR) vào các bộ tách tín hiệu tuyến tính (bao gồm cả bộ tách tín hiệu V-BLAST) là để nhằm cải thiện vùng quyết định (decision region) của các bộ tách tín hiệu. Quan sát Hình vẽ 8.17 chúng ta có thể thấy rằng sử dụng trợ giúp của phương pháp rút gọn cơ sở dàn chúng ta thu được hai véc-tơ trực giao với nhau và vì vậy vùng quyết định được cải thiện hơn so với các bộ tách tín hiệu tuyến tính ZF/MMSE hay V-BLAST. Ở trường hợp lý tưởng chúng ta có thể thu được các véc-tơ hoàn toàn trực giao với nhau như mô tả ở Hình vẽ 8.17(d), vùng quyết định của các bộ tách tín hiệu có trợ giúp phương pháp rút gọn cơ sở dàn (LRA) trùng với vùng quyết định của bộ tách tín hiệu MLD (Maximum Likelihood Detection: *tách tín hiệu hợp lệ cực đại*). Vì vậy, một cách tổng quát hoá chúng ta có thể nói rằng các bộ tách tín hiệu LRA có thể đạt được phẩm chất lỗi bít gần với phẩm chất của bộ tách tín hiệu MLD. Thực tế thì theo kết quả của [18],[19], và [20] thì các bộ tách tín hiệu V-BLAST có trợ giúp của LR có thể đạt được phẩm chất BER gần với đường phẩm chất tối ưu của bộ tách tín hiệu MLD.

Ý tưởng áp dụng phương pháp rút gọn cơ sở dàn vào các bộ tách tín hiệu tuyến tính được Yao và Wornell đề xuất lần đầu tiên ở [18], sau đó tiếp tục phát triển thêm tại các công trình nghiên cứu ở [19] và [20]. Cơ sở của các bộ tách tín hiệu có sự trợ giúp của LR là coi thành phần tín hiệu không có tạp âm  $Hs$  là một dàn (lattice) trong đó ma trận kênh truyền  $H$  là cơ sở dàn và các cột của ma trận



**Hình 8.17:** Miền (vùng) quyết định của các bộ tách tín hiệu [18].

$\mathbf{h}_i$  là các véc-tơ cơ sở. Như vậy, dàn do thành phần tín hiệu không có tạp âm  $\mathbf{Hs}$  tạo nên có thể biểu diễn như sau:

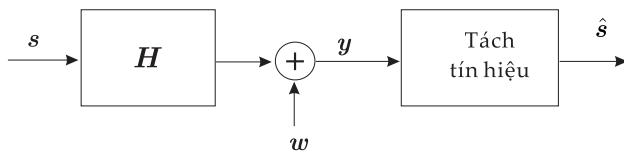
$$\mathcal{L}(\mathbf{H}) = s_1 \mathbf{h}_1 + s_2 \mathbf{h}_2 + \dots + s_N \mathbf{h}_N = \mathbf{Hs}, \quad (8.95)$$

trong đó các dấu phát  $s_i$  được lấy từ một tập hợp số nguyên cho trước tuỳ theo từng phương pháp điều chế. Ví dụ,  $s_i \in \{-1, 1\}$  cho BPSK và QPSK, hay  $s_i \in \{-3, -1, 1, 3\}$  cho 16-QAM.

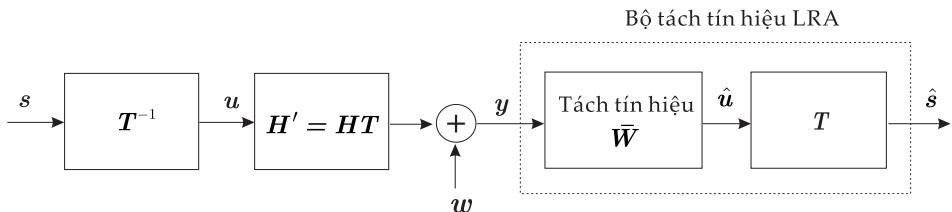
Sử dụng phương pháp rút gọn cơ sở dàn như trình bày ở phần trước chúng ta có thể viết lại phương trình hệ thống như sau

$$\mathbf{y} = \mathbf{Hs} + \mathbf{z} = (\mathbf{HT})(\mathbf{T}^{-1}\mathbf{s}) + \mathbf{z} = \mathbf{H}'\mathbf{u} + \mathbf{z}, \quad (8.96)$$

trong đó  $\mathbf{H}' = \mathbf{HT}$  và  $\mathbf{u} = \mathbf{T}^{-1}\mathbf{s}$  tương ứng là các ma trận kênh truyền và véc-tơ tín hiệu phát mới. Sử dụng mô hình mới ở (8.96), các bộ tách tín hiệu tuyến tính có trợ giúp của LR ước lượng  $\mathbf{u}$  trước, sau đó dùng  $\mathbf{T}$  để tính lại  $\mathbf{s}$ . Hình vẽ 8.18 trình bày mô hình tương đương của bộ tách tín hiệu có trợ giúp rút gọn cơ sở dàn. Trong mô



(a) Mô hình tách tín hiệu truyền thống



(b) Mô hình tách tín hiệu LRA

**Hình 8.18:** Mô hình tương đương của bộ tách tín hiệu có trợ giúp rút gọn cơ sở dàn.

hình mới này, bộ tách tín hiệu thứ nhất là các bộ tách tín hiệu tuyến tính ZF/MMSE hay cũng có thể là bộ tách tín hiệu kết hợp tuyến tính và triệt nhiễu nối tiếp V-BLAST. Ma trận trọng số của các bộ tách tín hiệu này có thể viết lại như sau

$$\mathbf{W}' = \begin{cases} \mathbf{H}'^H (\mathbf{H}'^H \mathbf{H}')^{-1} & \text{cho ZF} \\ (\zeta^2 \mathbf{H}' \mathbf{H}'^H + \sigma_z^2 \mathbf{I}_M)^{-1} \mathbf{H}' \zeta^2 & \text{cho MMSE} \end{cases} \quad (8.97)$$

trong đó  $\mathbf{H}' = \mathbf{HT}$  là cơ sở dàn đã rút gọn sử dụng thuật toán ở Bảng 8.3.

Do ma trận chuyển đổi  $T^{-1}$  chỉ chứa các số nguyên, nên nếu  $s$  được chọn từ một tập hợp số nguyên phức thì  $u$  cũng là một véc-tơ nguyên phức. Vì vậy, thao tác lượng tử hoá ước lượng  $\hat{u}_n$  từ đầu ra bộ tách tín hiệu tuyến tính tương đương với phép làm tròn riêng biệt phần thực và phần ảo, tức là

$$\bar{u}_n = [\Re\{\hat{u}_n\}] + j[\Im\{\hat{u}_n\}], \quad (8.98)$$

trong đó  $j = \sqrt{-1}$ . Cuối cùng véc-tơ tín hiệu phát gốc được khôi phục

lại nhờ ma trận chuyển đổi  $\mathbf{T}$  như sau

$$\hat{\mathbf{s}} = \mathbf{T}\bar{\mathbf{u}}. \quad (8.99)$$

Ước lượng  $\hat{\mathbf{s}}$  sau đó được tiếp tục được lượng tử hoá về chòm sao tín hiệu như sau

$$\bar{\mathbf{s}} = \mathcal{Q}\{\hat{\mathbf{s}}\}. \quad (8.100)$$

Trong trường hợp  $s$  thuộc chòm sao của tín hiệu QAM, chúng ta cần thực hiện phép dịch chuyển và lấy tỉ lệ (shift and scaling) trước sao cho đơn giản hoá bước làm tròn tiếp theo [19]. Ví dụ, với điều chế 4-QAM, trong đó tín hiệu phát thuộc chùm sao  $\mathcal{A} \in \{\pm \frac{1}{\sqrt{2}}, \pm j\frac{1}{\sqrt{2}}\}$ , thì  $s$  biểu diễn một véc-tơ nguyên phức  $\check{s} \in \mathbb{Z} + j\mathbb{Z}$  với các phần tử được dịch chuyển và lấy tỉ lệ như sau [19, 20]:

$$\mathbf{s} = \sqrt{2} \left( \check{\mathbf{s}} - \left[ \frac{1+j}{2}, \dots, \frac{1+j}{2} \right]^T \right). \quad (8.101)$$

Vì vậy, véc-tơ tín hiệu phát mới trở thành

$$\begin{aligned} \mathbf{u} &= \mathbf{T}^{-1}\mathbf{s} = \sqrt{2}\mathbf{T}^{-1}\check{\mathbf{s}} - \underbrace{\mathbf{T}^{-1}\left[\frac{1+j}{\sqrt{2}}, \dots, \frac{1+j}{\sqrt{2}}\right]^T}_{\tilde{\mathbf{s}}} \\ &= \sqrt{2}\underbrace{\mathbf{T}^{-1}\check{\mathbf{s}}}_{\check{\mathbf{u}}} - \tilde{\mathbf{s}} \\ &= \sqrt{2}\check{\mathbf{u}} - \tilde{\mathbf{s}}. \end{aligned} \quad (8.102)$$

Từ công thức (8.101) và (8.102) chúng ta có thể biểu diễn véc-tơ nguyên phức  $\check{\mathbf{u}}$  thông qua  $\mathbf{u}$  như sau

$$\check{\mathbf{u}} = \frac{1}{\sqrt{2}}(\mathbf{u} + \tilde{\mathbf{s}}). \quad (8.103)$$

Do đó, thao tác lượng tử hoá các ước lượng  $\hat{u}_n$  có thể được tiến hành sử dụng (8.103)

$$\hat{u}_n = \left\lfloor \frac{1}{\sqrt{2}}(\hat{u}_n + \tilde{s}_n) \right\rfloor. \quad (8.104)$$

và, cuối cùng, nhờ (8.102) như sau

$$\bar{u}_n = \sqrt{2}\hat{u}_n - \tilde{s}_n. \quad (8.105)$$

### 8.6.4 Bộ tách tín hiệu MLD

Giả sử tín hiệu phát được điều chế bởi phương pháp  $M_c$ -PSK hay  $M_c$ -QAM, chúng ta thấy véc-tơ tín hiệu phát  $s$  thuộc tập hợp  $\{s_\ell, \ell = 1, 2, \dots, M_c^N\}$ . Bộ tách tín hiệu MLD (Maximum Likelihood Detection: *tách tín hiệu hợp lệ cực đại*) [21],[22],[23] sử dụng tiêu chuẩn xác suất hậu nghiệm tối đa (MAP: Maximum a Posteriori Probability) để cực đại hóa xác suất ước lượng đúng véc-tơ tín hiệu phát  $s$  với điều kiện biết được tín hiệu thu  $y$  và ma trận kênh truyền  $H$ , tức là xác suất hậu nghiệm (posterior probability)[24, Ch.5]

$$\Pr\{s_\ell|y, H\}. \quad (8.106)$$

Sử dụng định luật Bayes chúng ta có thể biểu diễn xác suất hậu nghiệm này như sau

$$\Pr\{s_\ell|y, H\} = \frac{p(y|s_\ell, H)\Pr\{s_\ell\}}{p(y|H)}, \quad (8.107)$$

trong đó  $p(y|s_\ell, H)$  là hàm mật độ xác suất (PDF) có điều kiện của véc-tơ tín hiệu thu  $y$  khi cho trước  $s_\ell$  và  $H$ . Giả sử rằng tập tín hiệu phát  $\{s_\ell\}$  có xác suất như nhau ta có  $\Pr\{s_\ell\} = 1/M_c^N$ . Hơn nữa, để ý rằng  $p(y|H)$  không phụ thuộc vào  $s$  nên luật tách tín hiệu dựa vào cực đại hóa  $\Pr\{s_\ell|y, H\}$  ở (8.106) tương đương với tối đa hóa hàm mật độ xác suất có điều kiện  $p(y|s_\ell, H)$ , tức là chúng ta cần tìm véc-tơ

$$\bar{s} = \arg \max_{s_\ell \in \mathcal{A}} \{p(y|s_\ell, H)\}, \quad (8.108)$$

trong đó  $\mathcal{A}$  biểu diễn chòm sao tín hiệu (signal constellation). Hàm PDF có điều kiện  $p(y|s_\ell, H)$  thường được gọi là *hàm hợp lệ* (likelihood function) và, vì vậy, phương pháp tách tín hiệu này được gọi là phương pháp tách tín hiệu hợp lệ tối đa. Do  $H$  biết trước và các thành phần của  $z$  là các biến ngẫu nhiên Gauss với giá trị trung bình 0 và phương sai  $\sigma_z^2$  nên véc-tơ thu  $y$  là một véc-tơ Gauss phức có giá trị trung bình  $E\{y\} = Hs_\ell$  và ma trận tương quan

$$R_{yy} = E\{[y - Hs_\ell][y - Hs_\ell]^H\} = R_{zz} = \sigma_z^2 I.$$

Như vậy, chúng ta có thể biểu diễn  $p(\mathbf{y}|\mathbf{s}_\ell, \mathbf{H})$  như sau:

$$\begin{aligned} p(\mathbf{y}|\mathbf{s}_\ell, \mathbf{H}) &= \frac{1}{(\sqrt{\pi})^{2N} \det(\mathbf{R}_{zz})} \exp(-[\mathbf{y} - \mathbf{H}\mathbf{s}_\ell]^H \mathbf{R}_{zz}^{-1} [\mathbf{y} - \mathbf{H}\mathbf{s}_\ell]) \\ &= \frac{1}{\pi^N \sigma_z^{2P}} \exp\left(-\frac{\|\mathbf{y} - \mathbf{H}\mathbf{s}_\ell\|_2^2}{\sigma_z^2}\right). \end{aligned} \quad (8.109)$$

Do đó, tách tín hiệu  $\mathbf{s}_\ell$  sử dụng phương pháp MLD thông qua tối đa hoá hàm PDF có điều kiện (hàm hợp lệ) ở (8.109) tương đương với việc tối thiểu hoá bình phương khoảng cách giữa 2 véc-tơ  $\mathbf{y}$  và  $\mathbf{H}\mathbf{s}_\ell$ , tức là

$$\bar{\mathbf{s}} = \arg \min_{\mathbf{s}_\ell \in \mathcal{A}} \{\|\mathbf{y} - \mathbf{H}\mathbf{s}_\ell\|_2^2\}. \quad (8.110)$$

Nói cách khác bộ tách tín hiệu MLD cần thực hiện phép thử với  $M_c^N$  tổ hợp của  $\mathbf{s}_\ell$  và chọn véc-tơ cho khoảng cách nhỏ nhất làm véc-tơ tín hiệu tách được.

Phương pháp tách tín hiệu MLD cho kết quả BER tối ưu và vì vậy bộ tách tín hiệu MLD còn được gọi là bộ tách tín hiệu tối ưu (optimum detector)[24, Sect.5.1.3]. Tuy nhiên, bộ tách tín hiệu MLD có nhược điểm là khối lượng tính toán lớn. Do phải thực hiện  $M_c^N$  phép thử nên bộ tách tín hiệu MLD có bậc phức tạp  $O(M_c^N)$ . Với các hệ thống MIMO sử dụng nhiều ăng-ten và có bậc điều chế cao, độ phức tạp tính toán này quá cao và, vì vậy, bộ tách tín hiệu MLD ít được sử dụng trong thực tế. Giải pháp thay thế cho phép chúng ta thu được BER gần với của bộ tách tín hiệu MLD là sử dụng các phương pháp kết hợp MMSE với triết nối tiếp (SIC)[12],[13], sử dụng phương pháp kết hợp tách tín hiệu MMSE với triết nối song song [7, Sect.17.3.2], hay sử dụng phương pháp kết hợp MMSE và MLD [25],[26]. Ngoài ra, gần đây thuật toán giải mã cầu (sphere decoding) cũng đã được đề xuất áp dụng vào các bộ tách tín hiệu MLD cho phép giảm đáng kể độ phức tạp tính toán. Một xu hướng khác cũng đang được quan tâm là sử dụng kết hợp phương pháp LR (Lattice Reduction: rút gọn dàn) kết hợp với các bộ tách tín hiệu tuyến tính cho phép thu được đường BER tiệm cận với đường BER

của phương pháp MLD trong khi không đòi hỏi thêm đáng kể độ tính toán.

### 8.6.5 Bộ tách tín hiệu hình cầu (sphere detector)

#### Khái quát về bộ giải mã cầu

Như đã trình bày ở phần trước các bộ tách tín hiệu MLD mặc dù cho phẩm chất BER tối ưu nhưng đòi hỏi độ phức tạp tính toán lớn. Để tìm được  $\bar{s}$  bộ tách tín hiệu MLD cần phải thực hiện phép thử đối với toàn bộ tập véc-tơ tín hiệu có thể và tìm ra véc-tơ tương ứng thỏa mãn

$$\bar{s} = \arg \min_{s \in \mathcal{A}} \{\|y - Hs\|_2^2\}. \quad (8.111)$$

Nói một cách khác bộ tách tín hiệu MLD cần phải tín toán khoảng cách bình phương từ điểm tín hiệu thu đến các điểm dàn (thuộc chòm sao tín hiệu)  $d_\ell^2 = \|y - Hs_\ell\|^2, \ell = 1, 2, \dots, M_c^N$  để tìm ra véc-tơ cho khoảng cách nhỏ nhất. Với các phương pháp điều chế có độ điều chế cao như 64-QAM hay các hệ thống MIMO có số ăng-ten phát lớn thì số lượng phép thử khoảng cách này trở nên quá lớn làm cho các bộ tách tín hiệu MLD không sử dụng được.

Mặc dù các thuật toán khác như MMSE hay V-BLAST có thể được sử dụng để khôi phục tín hiệu ở phía máy thu nhằm giảm khối lượng tính toán, nhưng phẩm chất lỗi bit của các thuật toán này là không tối ưu và thực tế là còn một khoảng cách rất xa so với phẩm chất lỗi bit tối ưu. Chính vì vậy, ta cần phát triển các phương pháp khôi phục tín hiệu cho phép đạt được phẩm chất lỗi bit tối ưu nhưng có độ phức tạp trong tính toán giảm. Một trong những phương pháp như vậy là phương pháp tách tín hiệu (giải mã) hình cầu (SD: Sphere Detector) hay ta còn gọi là phương pháp giải mã cầu. Phương pháp này cho phép máy thu đạt được phẩm chất lỗi bit tối ưu với khối lượng tính toán có bậc đa thức (polynomial complexity). Phương pháp giải mã cầu được đề xuất lần đầu tiên bởi Fincke và Posh [27, 28] nhằm tìm kiếm các véc-tơ có chiều dài ngắn

nhất trong một dàn cho trước. Sau đó nó được ứng dụng để giải quyết bài toán *bình phương tối thiểu của tập số nguyên* (integer least-square)

$$\arg \min_{\mathbf{u} \in \mathbb{Z}^n} \|\mathbf{x} - \mathbf{M}\mathbf{u}\|^2 \quad (8.112)$$

trong đó  $\mathbf{x}$  là một véc-tơ thực có kích thước  $m \times 1$  ( $\mathbf{x} \in \mathbb{R}^m$ ),  $\mathbf{M}$  là một ma trận thực có kích thước  $m \times n$  được gọi là ma trận sinh của dàn ( $\mathbf{M} \in \mathbb{R}^{m \times n}$ ),  $\mathbb{Z}^n$  là một dàn số nguyên  $n$  chiều, và  $\mathbf{u}$  là một véc-tơ có kích thước  $n \times 1$  với các phân tử là số nguyên. Chú ý rằng do  $\mathbf{u}$  là một véc-tơ các số nguyên, ta không thể áp dụng các thuật toán thích nghi như LMS hay RLS để giải quyết bài toán trên mà phải áp dụng thuật toán giải mã cầu để tìm ra nghiệm tối ưu.

Đối với một hệ thống MIMO sử dụng phương pháp điều chế QAM có mối quan hệ giữa tín hiệu phát và tín hiệu thu như ở công thức (8.5), ta có thể biểu diễn (8.5) một cách tương đương như sau:

$$\mathbf{x} = \mathbf{M}\mathbf{u} + \mathbf{n} \quad (8.113)$$

trong đó

$$\mathbf{x} = \begin{bmatrix} \Re\{\mathbf{y}\} \\ \Im\{\mathbf{y}\} \end{bmatrix} \in \mathbb{R}^m, \quad (8.114)$$

$$\mathbf{u} = \begin{bmatrix} \Re\{\mathbf{s}\} \\ \Im\{\mathbf{s}\} \end{bmatrix} \in \mathbb{Z}^n, \quad (8.115)$$

$$\mathbf{n} = \begin{bmatrix} \Re\{\mathbf{z}\} \\ \Im\{\mathbf{z}\} \end{bmatrix} \in \mathbb{R}^m, \quad (8.116)$$

$$\mathbf{M} = \begin{bmatrix} \Re\{\mathbf{H}\} & -\Im\{\mathbf{H}\} \\ \Im\{\mathbf{H}\} & \Re\{\mathbf{H}\} \end{bmatrix} \in \mathbb{R}^{m \times n}, \quad (8.117)$$

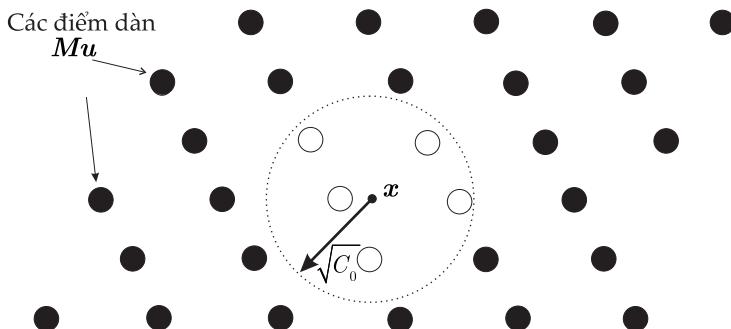
$m = 2M$ , và  $n = 2N$ .

Lúc này véc-tơ tín hiệu phát  $\mathbf{u}$  được khôi phục dựa trên biểu thức sau:

$$\bar{\mathbf{u}} = \arg \min_{\mathbf{u} \in \mathcal{Z}} \|\mathbf{x} - \mathbf{M}\mathbf{u}\|^2 \quad (8.118)$$

trong đó  $Z$  là một tập hữu hạn các số nguyên mà từ đó máy phát xây dựng nên các ký hiệu QAM. Ví dụ, các ký hiệu 16-QAM được xây dựng từ tập hợp các số nguyên  $Z = \{-3, -1, 1, 3\}$ .

So sánh (8.118) với (8.112) ta thấy rằng bài toán khôi phục tín hiệu tối ưu (8.118) hoàn toàn tương đương với bài toán bình phương tối thiểu của tập số nguyên (8.112). Điều này chứng tỏ rằng, thuật toán giải mã cầu có thể được sử dụng để giải bài toán khôi phục tín hiệu tối ưu trong các hệ thống MIMO.



**Hình 8.19:** Minh họa nguyên lý hoạt động của một bộ giải mã cầu.

Ý tưởng chủ đạo trong thuật toán giải mã cầu là chỉ tìm kiếm nghiệm tối ưu trên các điểm dàn nằm trong một hình cầu với một bán kính  $\sqrt{C_0}$  định trước có tâm là điểm tín hiệu thu  $x$ , như được minh họa trên Hình 8.19. Mỗi lần tìm được một điểm dàn hợp lệ nằm trong hình cầu, bộ tách tín hiệu SD rút gọn bán kính xuống bằng khoảng cách giữa điểm tín hiệu thu và điểm dàn tìm được. Bộ tách tín hiệu SD được tiếp tục cho đến khi còn lại một điểm dàn duy nhất còn lại trong hình cầu. Điểm dàn cuối cùng được chọn là nghiệm hợp lệ tối ưu của véc-tơ (điểm) tín hiệu phát đi với các thành phần là toạ độ của điểm dàn. Cốt lõi của phương pháp tách tín hiệu SD là thuật toán tìm kiếm các điểm dàn hợp lệ nằm trong hình cầu. Thuật toán này được đề xuất đầu tiên ở [27, 28]. Ứng dụng của thuật

toán vào bộ giải mã cho kênh pha-định được đề xuất ở [29], và cho các hệ thống MIMO ở [30], [31], và [32]. Dưới đây chúng ta lần lượt xem xét một số thuật toán SD cơ bản nhất.

### Các thuật toán giải mã cầu trong không gian thực

Các thuật toán giải mã cầu trong không gian thực (gọi tắt là giải mã cầu thực) được đề xuất để giải bài toán khôi phục tín hiệu tối ưu trong (8.118). Theo các thuật toán này một điểm dàn  $M\mathbf{u}$  nằm trong một hình cầu có tâm tại  $\mathbf{x}$  và có bán kính  $\sqrt{C_0}$  nếu và chỉ nếu:

$$\|\mathbf{x} - M\mathbf{u}\|^2 \leq C_0. \quad (8.119)$$

Thực hiện phân tích thừa số QR đối với ma trận  $M$ , ta được:

$$M = Q \begin{bmatrix} R \\ 0_{(m-n),n} \end{bmatrix} \quad (8.120)$$

trong đó<sup>18</sup>  $R$  là một ma trận tam giác trên có kích thước  $n \times n$  với các phần tử là số thực,  $Q = [Q_1 \ Q_2]$  là một ma trận thực, đơn nhất có kích thước  $m \times m$ ,  $Q_1$  and  $Q_2$  lần lượt chứa  $n$  cột đầu tiên và  $(m-n)$  cột cuối của  $Q$ .

Khi đó, điều kiện (8.119) có thể được viết như sau:

$$\|Q_1^H \mathbf{x} - R\mathbf{u}\|^2 \leq C_0 - \|Q_2^H \mathbf{x}\|^2. \quad (8.121)$$

Đặt  $\mathbf{y} = Q_1^H \mathbf{x}$  và  $C = C_0 - \|Q_2^H \mathbf{x}\|^2$ , công thức ((8.121)) có thể được biểu diễn lại như sau:

$$\sum_{i=1}^n \left( y_i - \sum_{j=i}^n r_{i,j} u_j \right)^2 \leq C \quad (8.122)$$

trong đó  $r_{i,j}$  là phần tử nằm ở hàng  $i$  và cột  $j$  của  $R$ .

Triển khai về trái của (8.122) ta được

$$(y_n - r_{n,n} u_n)^2 + (y_{n-1} - r_{n-1,n} u_n - r_{n-1,n-1} u_{n-1})^2 + \dots \leq C \quad (8.123)$$

<sup>18</sup>Chú ý font chữ khác nhau biểu diễn các véc-tơ khác nhau: ví dụ cần phân biệt  $R$  với  $\mathbf{R}$ .

trong đó số hạng đầu tiên là một hàm số với biến số duy nhất  $u_n$ , số hạng thứ hai là một hàm số có hai biến số là  $(u_n, u_{n-1})$ ,....

Từ (8.122), điều kiện cần để  $Mu$  nằm bên trong hình cầu là

$$(y_n - r_{n,n}u_n)^2 \leq C. \quad (8.124)$$

Điều kiện này có nghĩa là  $u_n$  nằm trong khoảng

$$LB(u_n) = \left\lceil \frac{-\sqrt{C} + y_n}{r_{n,n}} \right\rceil \leq u_n \leq \left\lfloor \frac{\sqrt{C} + y_n}{r_{n,n}} \right\rfloor = UB(u_n) \quad (8.125)$$

trong đó  $\lceil . \rceil$  và  $\lfloor . \rfloor$  tương ứng biểu thị các phép làm tròn số tới các giá trị lớn hơn gần nhất và nhỏ hơn gần nhất thuộc tập  $\mathbb{Z}$ ,  $LB(u_n)$  và  $UB(u_n)$  tương ứng biểu thị cận dưới và cận trên cho  $u_n$ .

Với mỗi một giá trị của  $u_n$  thỏa mãn điều kiện (8.125), ta đặt  $C_{n-1} = C - (y_n - r_{n,n}u_n)^2$ . Khi đó, một điều kiện cần khác, mạnh hơn điều kiện (8.124), để  $Mu$  nằm bên trong hình cầu là

$$(y_{n-1} - r_{n-1,n}u_n - r_{n-1,n-1}u_{n-1})^2 \leq C_{n-1} \quad (8.126)$$

Sau một số bước biến đổi toán học, ta có thể dễ dàng chứng minh được  $u_{n-1}$  thỏa mãn điều kiện:

$$LB(u_{n-1}) \leq u_{n-1} \leq UB(u_{n-1}) \quad (8.127)$$

trong đó  $LB(u_{n-1}) = \left\lceil \frac{-\sqrt{C_{n-1}} + y_{n-1}^{(u_n)}}{r_{n-1,n-1}} \right\rceil$ ,  $UB(u_{n-1}) = \left\lfloor \frac{\sqrt{C_{n-1}} + y_{n-1}^{(u_n)}}{r_{n-1,n-1}} \right\rfloor$ , và  $y_{n-1}^{(u_n)} = y_{n-1} - r_{n-1,n}u_n$ .

Bằng các bước biến đổi tương tự, chúng ta tìm được các khoảng giá trị cho  $u_{n-2}, u_{n-3}, \dots, u_1$ , và do đó ta tìm được tất cả các điểm dàn thỏa mãn điều kiện (8.119).

Một cách tổng quát ta thấy rằng  $u_i, i = n, n-1, \dots, 1$ , được giới hạn bởi  $LB(u_i) \leq u_i \leq UB(u_i)$ . Phương pháp xác định các giá trị của  $u_i$  sẽ tác động tới mức độ phức tạp (biểu thị bằng khối lượng tính toán) của một thuật toán giải mã cầu. Trong hầu hết các bộ giải mã cầu, các giá trị của  $u_i$  được xác định bằng phương pháp liệt kê Pohst (Pohst enumeration) hoặc bằng phương pháp liệt kê Schnorr-Euchner (Schnorr-Euchner enumeration).

- Thuật toán giải mã cầu với phương pháp liệt kê Pohst

Phương pháp liệt kê Pohst được đề xuất bởi Pohst [27]. Phương pháp này dựa trên việc mở rộng một cách tự nhiên các khoảng  $[LB(u_i), UB(u_i)]$  tại mỗi mức  $i$ . Theo phương pháp liệt kê Pohst  $u_i$  sẽ lấy các giá trị như sau:  $LB(u_i), LB(u_i) + 2, \dots, UB(u_i)$ . Phương pháp Pohst được ứng dụng lần đầu tiên vào lĩnh vực truyền thông số bởi Viterbo và các cộng sự [29]. Thuật toán giải mã cầu với phương pháp liệt kê Pohst (viết tắt là PE-SD: Pohst enumeration Sphere Decoder) được tóm tắt như sau.

### Thuật toán PE-SD

---

BEGIN: Nhập  $C, y, \mathbf{R}$ ,

(1) **Bước 1.** (Khởi tạo) Đặt  $i := n$ ,  $T_i := 0$ ,  $\xi_i := 0$  và  $d_c := C$  (bán kính cầu hiện tại).

(2) **Bước 2.** If  $d_c < T_i$ : chuyển tới Bước 4.

Else: tính  $LB(u_i), UB(u_i)$ , và đặt  $u_i := LB(u_i) - 2$ .

(3) **Bước 3.** (*Pohst enumeration*) Đặt  $u_i := u_i + 2$ . If:  $u_i \leq UB(u_i)$  chuyển tới Bước 5. Else: chuyển tới Bước 4.

(4) **Bước 4.** If  $i = n$ : ngắt. Else: đặt  $i := i + 1$ , chuyển tới Bước 3.

(5) **Bước 5.** If  $i > 1$ : đặt  $\xi_{i-1} := \sum_{j=i}^n r_{i-1,j} u_j$ ,  $T_{i-1} := T_i + |y_i - \xi_i - r_{i,i} u_i|^2$ , đặt  $i := i - 1$  và chuyển tới Bước 2.

(6) **Bước 6.** (*Tìm được một điểm hợp lệ*) Đặt  $d := T_1 + |y_1 - \xi_1 - r_{1,1} u_1|^2$ . Nếu  $d < d_c$ : đặt  $d_c := d$ , lưu  $\bar{\mathbf{u}} := \mathbf{u}$ , và cập nhật các biên trên  $UB(u_i)$ ,  $i = 1, \dots, n$ . Chuyển tới Bước 3.

---

END: Xuất  $\bar{\mathbf{u}}$

- Thuật toán giải mã cầu với phương pháp liệt kê Schnorr-Euchner

Phương pháp liệt kê Schnorr-Euchner, được đề xuất trong [33], là một biến thể của phương pháp liệt kê Pohst. Theo phương pháp này, các giá trị của  $u_i$  được xác định theo thứ tự đích dắc, khởi đầu từ điểm giữa

$$\tilde{u}_i = \left\lfloor \frac{1}{r_{i,i}} \left( y_i - \sum_{j=i+1}^n r_{i,j} u_j \right) \right\rfloor \quad (8.128)$$

trong đó  $\lfloor . \rfloor$  ký hiệu phép làm tròn tới số nguyên gần nhất nằm trong tập  $\mathbb{Z}$ . Một cách cụ thể hơn, tại mỗi mức  $i$ , phương pháp liệt kê Schnorr-Euchner sẽ tạo một dãy các giá trị của  $u_i$  như sau

- Nếu  $y_i - \sum_{j=i+1}^n r_{i,j} u_j - r_{i,i} \tilde{u}_i \geq 0$ , thì

$$u_i \in \{\tilde{u}_i, \tilde{u}_i + 2, \tilde{u}_i - 2, \tilde{u}_i + 4, \tilde{u}_i - 4, \dots\} \cap [LB(u_i), UB(u_i)].$$

- Nếu  $y_i - \sum_{j=i+1}^n r_{i,j} u_j - r_{i,i} \tilde{u}_i < 0$ , thì

$$u_i \in \{\tilde{u}_i, \tilde{u}_i - 2, \tilde{u}_i + 2, \tilde{u}_i - 4, \tilde{u}_i + 4, \dots\} \cap [LB(u_i), UB(u_i)].$$

Người ta đã chứng minh rằng phẩm chất và mức độ phức tạp của thuật toán PE-SD rất dễ bị tác động bởi sự lựa chọn giá trị bán kính ban đầu hình cầu,  $\sqrt{C_0}$ , so với thuật toán giải mã cầu với phương pháp liệt kê Schnorr-Euchner (viết tắt là SE-SD: Schnorr-Euchner enumeration Sphere Decoder) [31]. Đối với thuật toán PE-SD, nếu bán kính  $\sqrt{C_0}$  quá lớn, ta sẽ phải kiểm tra rất nhiều véc-tơ tín hiệu, dẫn tới khối lượng tính toán rất lớn. Ngược lại, nếu bán kính  $\sqrt{C_0}$  quá nhỏ, ta có thể không thu được véc-tơ tín hiệu nào nằm trong hình cầu, tức là ta không khôi phục được véc-tơ tín hiệu phát. Nếu trường hợp này xảy ra, thuật toán có thể tăng bán kính lên và thực hiện lại quá trình khôi phục tín hiệu.

Trái lại, nhờ có sự trợ giúp của phương pháp liệt kê Schnorr-Euchner, sự lựa chọn giá trị bán kính ban đầu của hình cầu  $\sqrt{C_0}$  ít tác động tới mức độ phức tạp của thuật toán SE-SD. Do đó, đối với thuật toán SE-SD, ta có thể chọn một giá trị bán kính lớn ngay từ đầu. Khi đó xác suất xuất hiện sự kiện một hình cầu rỗng (không có

véc-tơ tín hiệu nào nằm trong hình cầu) bằng 0. Bên cạnh ưu điểm này, thuật toán SE-SD còn được chứng minh là có mức độ phức tạp nhỏ hơn nhiều so với thuật toán PE-SD khi cả hai thuật toán có cùng giá trị bán kính ban đầu, đặc biệt là khi số lượng ký hiệu phát lớn.

Thuật toán SE-SD đề xuất bởi Damen và các cộng sự [31] được tóm tắt như sau:

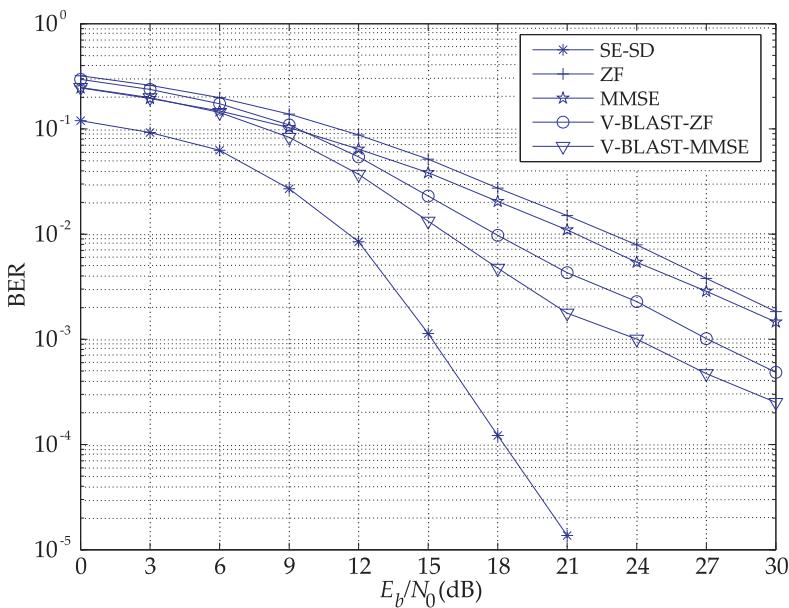
### Thuật toán SE-SD

BEGIN: Input  $C, \mathbf{y}, \mathbf{R}$ .

- (1) **Bước 1.** (Khởi tạo) Đặt  $i := n, T_i := 0, \xi_i = 0$  and  $d_c := C$  (bán kính hình cầu hiện tại).
- (2) **Bước 2.** Đặt  $u_i := \lfloor (y_i - \xi_i)/r_{i,i} \rfloor$  và  $\Delta_i := 2\text{sign}(y_i - \xi_i - r_{i,i}u_i)$ .
- (3) **Bước 3.** (Bước chính) If  $d_c < T_i + |y_i - \xi_i - r_{i,i}u_i|^2$  (tức là đang ở ngoài hình cầu) chuyển tới Bước 4. Elseif  $u_i \in \mathcal{Z}$  (tức là, ở trong hình cầu nhưng ngoài tập tín hiệu), chuyển tới Bước 6. Elseif  $i > 1$  (tức là, đang ở trong hình cầu và trong tập tín hiệu  $\mathcal{Z}$ ), đặt  $\xi_{i-1} := \sum_{j=i}^n r_{i-1,j}u_j$ ,  $T_{i-1} := T_i + |y_i - \xi_i - r_{i,i}u_i|^2$ ,  $i := i - 1$ , và chuyển tới Bước 2. Elseif  $i = 1$ , chuyển tới Bước 5.
- (4) **Bước 4.** If  $i = n$ , ngắt, Else đặt  $i := i + 1$  và chuyển tới Bước 6.
- (5) **Bước 5.** (tìm được một điểm hợp lệ) Đặt  $d_c := T_1 + |y_1 - \xi_1 - r_{1,1}u_1|^2$ , lưu  $\bar{\mathbf{u}} := \mathbf{u}$ . Sau đó đặt  $i := i + 1$  và chuyển tới Bước 6.
- (6) **Bước 6.** (Liệt kê Schnorr-Euchner cho mức  $i$ ) Đặt  $u_i := u_i + \Delta_i$ ,  $\Delta_i := -\Delta_i - 2\text{sign}(\Delta_i)$ , và chuyển tới Bước 3.

END: Output  $\bar{\mathbf{u}}$ .

Hình 8.20 so sánh phẩm chất lỗi bit của hệ thống  $4 \times 4$  MIMO-SDM sử dụng phương pháp điều chế 16QAM được tách sóng bằng bộ giải mã cầu SE-SD và một số bộ giải mã khác. Từ hình vẽ, chúng ta thấy bộ giải mã cầu SE-SD giúp hệ thống đạt được phẩm chất



Hình 8.20: BER của hệ thống  $4 \times 4$  MIMO-SDM; Điều chế 16QAM.

lỗi bit tối ưu, tốt hơn rất nhiều so với phẩm chất lỗi bit của các bộ giải mã khác mà chúng ta đã xét được ở phần trên. Ví dụ, bộ giải mã cầu SE-SD giúp hệ thống đạt được  $\text{BER} = 10^{-3}$  ở  $E_b/N_0$  khoảng 16 dB, trong khi bộ giải mã V-BLAST-MMSE đạt được  $\text{BER} = 10^{-3}$  ở  $E_b/N_0$  cao hơn rất nhiều (khoảng 24 dB). Các bộ giải mã khác như V-BLAST-ZF hay giải mã tuyến tính còn yêu cầu tỷ số tín hiệu trên tạp âm cao hơn nữa (trên 27 dB) để có thể đạt được  $\text{BER} = 10^{-3}$ .

Các thuật toán giải mã cầu trong không gian số thực đòi hỏi một hệ thống MIMO phức phải được phân chia thành các phần thực và phần ảo nhằm tạo ra một hệ thống tương đương trong không gian thực. Vì vậy, các thuật toán này sẽ hiệu quả nhất khi được sử dụng để khôi phục tín hiệu được điều chế dựa trên các dàn số nguyên như QAM hay PAM. Đối với các kỹ thuật điều chế khác như PSK, việc sử dụng các thuật toán SD thực sẽ không hiệu quả do sự tồn tại các những điểm tín hiệu không hợp lệ (invalid), dẫn tới sự suy giảm phẩm chất lỗi bit của hệ thống. Để giải quyết bài toán này,

ta có thể sử dụng phương pháp loại bỏ điểm tín hiệu không hợp lệ đề xuất bởi Cui và các cộng sự [34] hoặc sử dụng các thuật toán giải mã cầu trong không gian phức (complex SD).

### Các thuật toán giải mã cầu trong không gian phức

Như đã đề cập, khi các thuật toán giải mã cầu trong không gian thực được sử dụng để khôi phục tín hiệu PSK trong các hệ thống MIMO sẽ dẫn tới sự suy giảm về phẩm chất lỗi bit do các thuật toán này có thể lựa chọn các điểm tín hiệu thỏa mãn tiêu chí hợp lệ tối đa (maximum likelihood). Tuy nhiên, các điểm tín hiệu này lại không thuộc chòm sao tín hiệu mà phía phát sử dụng. Những điểm tín hiệu như vậy được gọi là các điểm tín hiệu không hợp lệ. Vấn đề này được giải quyết bởi Hochwald và các cộng sự [35]. Tiếp theo đó, các bộ giải mã cầu có mức độ tính toán nhỏ hơn đã được đề xuất [36], [37], và [38].

Trước hết, chúng ta xem xét thuật toán giải mã cầu đề xuất bởi Hochwald (gọi tắt là Hochwald SD). Thuật toán này tuân thủ tất cả các bước của một thuật toán giải mã cầu trong không gian thực trình bày ở trên. Điểm khác biệt chủ yếu là việc thực hiện phân tích thừa số QR được áp dụng cho ma trận  $\mathbf{H}$  trong hệ thống phức (8.5) để thu được hệ thống phức tương đương như sau:

$$\underbrace{\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix}}_{\mathbf{v}} = \underbrace{\begin{bmatrix} R_{1,1} & R_{1,2} & \dots & R_{1,N} \\ 0 & R_{2,2} & \dots & R_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & R_{N,N} \end{bmatrix}}_{\mathbf{R}} \underbrace{\begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{bmatrix}}_{\mathbf{s}} + \underbrace{\begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_N \end{bmatrix}}_{\mathbf{n}} \quad (8.129)$$

trong đó  $\mathbf{v} = \mathbf{Q}^H \mathbf{y}$  và  $\mathbf{n} = \mathbf{Q}^H \mathbf{z}$  lần lượt là véc-tơ tín hiệu thu được và véc-tơ tạp âm sau khi thực hiện phân tích thừa số QR.

Tương tự như (8.123), điều kiện để một điểm tín hiệu nằm

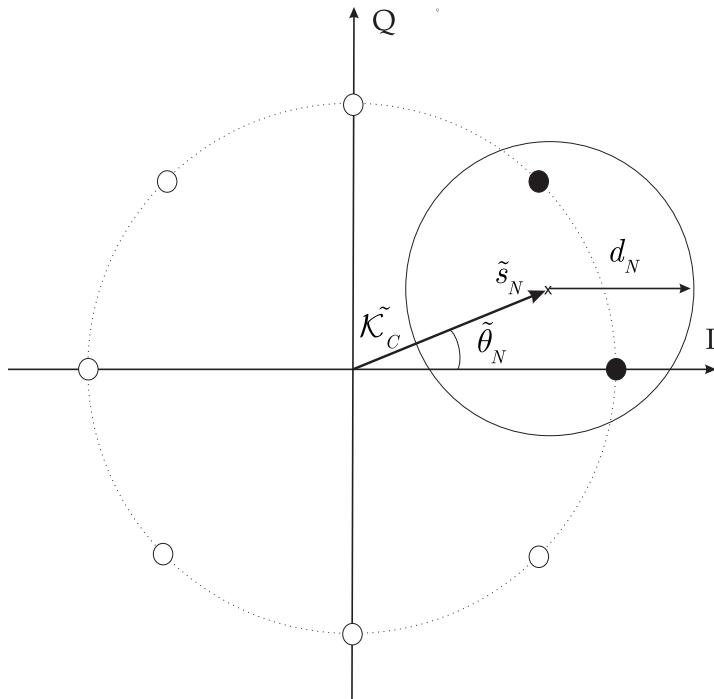
trong hình cầu là:

$$|v_N - R_{N,N}s_N|^2 + |v_{N-1} - R_{N-1,N}s_N - R_{N-1,N-1}s_{N-1}|^2 + \dots \leq C. \quad (8.130)$$

Do đó, điều kiện cần  $|v_N - R_{N,N}s_N|^2 \leq C$  có thể được biểu diễn tương đương như sau:

$$|s_N - v_N/R_{N,N}|^2 \leq C/R_{N,N}^2. \quad (8.131)$$

Bất đẳng thức này chỉ ra rằng ta chỉ nên tìm nghiệm tối ưu thông qua việc thử tất cả các điểm tín hiệu giới hạn trong một vòng tròn có tâm tại  $\tilde{s}_N = v_N/R_{N,N}$  và có bán kính  $d_N = \sqrt{C}/R_{N,N}$ , như minh họa trên Hình 8.21.



**Hình 8.21:** Nguyên lý tìm các điểm tín hiệu của thuật toán giải mã cầu trong không gian phức.

Đặt  $s_N = \kappa_c e^{j\theta_N}$ , trong đó  $\kappa_c$  là một hằng số dương và  $\theta_N \in \{0, 2\pi/\mathcal{M}, 4\pi/\mathcal{M}, \dots, 2\pi(\mathcal{M}-1)/\mathcal{M}\}$ , điều này có nghĩa là,  $s_N$  là một ký hiệu  $\mathcal{M}$ -PSK. Tương tự như vậy, đặt  $\tilde{s}_N = \tilde{\kappa}_c e^{j\tilde{\theta}_N}$ , trong đó  $\tilde{\kappa}_c > 0$ .

Ta có thể viết (8.131) như sau [35]:

$$\kappa_c^2 + \tilde{\kappa}_c^2 - 2\kappa_c\tilde{\kappa}_c \cos(\theta_N - \tilde{\theta}_N) \leq d_N^2. \quad (8.132)$$

Từ biểu thức này, ta rút ra

$$\cos(\theta_N - \tilde{\theta}_N) \geq \frac{1}{2\kappa_c\tilde{\kappa}_c} (\kappa_c^2 + \tilde{\kappa}_c^2 - d_N^2) = \eta. \quad (8.133)$$

Biểu thức (8.133) cho ta thấy rằng:

- Nếu  $\eta > 1$ , vòng tròn không chứa bất cứ điểm nào của chòm sao tín hiệu PSK.
- Nếu  $\eta < -1$ , vòng tròn chứa toàn bộ chòm sao tín hiệu PSK.
- Nếu  $-1 \leq \eta \leq 1$ , góc pha có thể có của ký hiệu  $s_N$  nằm trong khoảng

$$\left[ \frac{\mathcal{M}}{2\pi} (\tilde{\theta}_N - \cos^{-1} \eta) \right] \leq \frac{\mathcal{M}}{2\pi} \theta_N \leq \left[ \frac{\mathcal{M}}{2\pi} (\tilde{\theta}_N + \cos^{-1} \eta) \right]. \quad (8.134)$$

Bây giờ, ta có thể chọn  $s_N$  là một điểm trong chòm sao tín hiệu PSK bằng cách chọn  $\theta_N$  là một giá trị thỏa mãn biểu thức (8.134). Phần còn lại của thuật toán giải mã cầu trong không gian phức diễn tiến tương tự như trường hợp trong không gian thực.

Một phiên bản khác của thuật toán Hochwald SD có độ phức tạp thấp hơn, được gọi là thuật toán Pham SD, đã được đề xuất trong [36]. Chìa khóa của thuật toán Pham SD chính là việc sử dụng thuật toán DF-BB (DF-BB: Depth-First Branch and Bound algorithm). Về bản chất, thuật toán DF-BB tương tự như là phương pháp liệt kê Schnorr-Euchner. Trong thuật toán Pham SD, các điểm tín hiệu tiềm năng tại mỗi mức được xác định dựa trên ý tưởng của thuật toán Hochwald SD. Nhưng sau đó, Pham SD sử dụng một thủ tục sắp xếp các điểm tín hiệu tiềm năng nhằm làm cho số lượng phép thử tìm điểm tín hiệu tối ưu giảm tới mức tối đa. Tuy nhiên, một nhược điểm của cả thuật toán Hochwald SD và thuật toán Pham SD là chúng phải xử lý hàm  $\cos^{-1}$ . Điều này dẫn tới sự

không hiệu quả trong tính toán và làm chậm quá trình khôi phục tín hiệu.

Trong [38], Tuan và các cộng sự trình bày một thuật toán tối ưu PMLD dùng để khôi phục các tín hiệu PSK. Dựa trên thuật toán này ta dễ dàng xây dựng được một thuật toán giải mã cầu trong không gian phức cho tín hiệu PSK. Thuật toán này, còn được gọi là thuật toán Tuan SD, khôi phục véc-tơ tín hiệu phát  $s$  dựa vào biểu thức

$$\bar{s} = \arg \min_{s \in \Omega} \|v - Rs\|^2 \quad (8.135)$$

$$= \arg \min_{s \in \Omega} \sum_{k=1}^N |\xi_k - R_{k,k}s_k|^2 \quad (8.136)$$

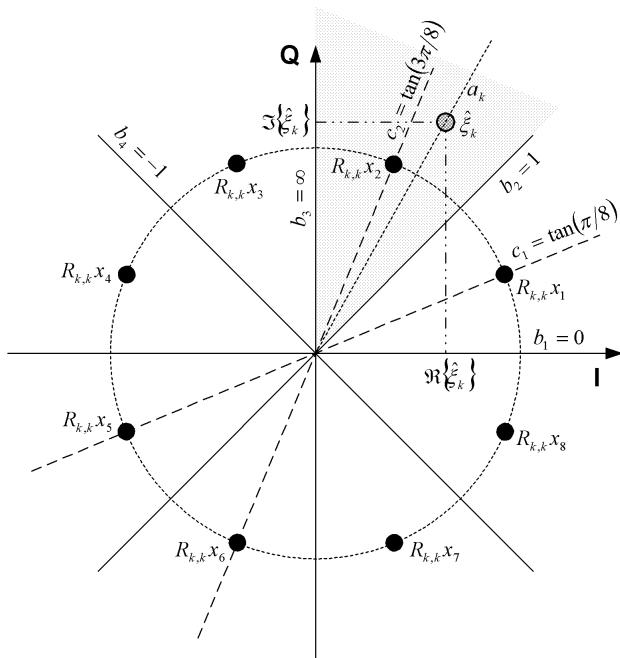
trong đó  $\xi_k = v_k - \sum_{i=k+1}^N R_{k,i}s_i$ ,  $\Omega$  là chòm sao tín hiệu phát.

Xét một chòm sao tín hiệu  $\mathcal{M}$ -PSK được cho bởi

$$\Omega = \{\rho e^{j \frac{(2n+1)\pi}{\mathcal{M}}} : n = 0, 1, \dots, \mathcal{M}-1\}. \quad (8.137)$$

Đặt  $x = (x_1, \dots, x_{\mathcal{M}})$  là một véc-tơ có kích thước  $\mathcal{M} \times 1$  chứa  $\mathcal{M}$  ký hiệu  $\epsilon \Omega$  theo thứ tự  $n = 0, 1, \dots, \mathcal{M}-1$ . Khi đó từ biểu thức (8.136) ta thấy rằng bài toán xác định thứ tự thử tối ưu (tức là xác định thứ tự các điểm tín hiệu sẽ được thử để cho số lượng phép thử giảm tối đa) ở lớp  $k$  trở thành bài toán xác định giá trị nào trong số  $\mathcal{M}$  giá trị  $R_{k,k}x_i$ ,  $i = 1, \dots, \mathcal{M}$ , là gần  $\hat{\xi}_k$  nhất, gần  $\hat{\xi}_k$  thứ hai, ..., vân vân. Chú ý rằng  $\hat{\xi}_k$  chính là  $\xi_k$  được tính với các giá trị  $s_i = \bar{s}_i$ , tức là  $\hat{\xi}_k = \xi_k|_{s_i=\bar{s}_i} = v_k - \sum_{i=k+1}^N R_{k,i}\bar{s}_i$ , với  $\bar{s}_i$  là các giá trị quyết định thăm dò (tentative decision) của  $s_i$ . Giá trị quyết định thăm dò  $\bar{s}_i$  được hiểu một cách đơn giản là một điểm tín hiệu nào đó trong  $\Omega$  mà ta giả thiết nó chính là điểm tín hiệu được phát đi  $s_i$ .

Để minh họa, ta xét trường hợp  $\mathcal{M} = 8$ , tức là, 8PSK. Trình bày trên Hình 8.22 là các giá trị  $R_{k,k}x_i$ ,  $i = 1, \dots, 8$ , và  $\hat{\xi}_k$  trong một mặt phẳng phức. Do  $R_{k,k}$  là một số thực dương, nên góc pha của  $R_{k,k}x_i$  cũng chính là góc pha của  $x_i$  với mọi  $i$ . Hình 8.22 gợi mở cho chúng ta một phương pháp đơn giản để xác định thứ tự thử tối ưu cho tất cả các điểm tín hiệu tại lớp  $k$  bằng cách so sánh độ dốc của các đường thẳng đi qua gốc tọa độ [38].



Hình 8.22: Xác định thứ tự thử tối ưu tại lớp  $k$  cho các ký hiệu 8-PSK trong  $x$ .

Trước hết, ta tính độ dốc (hay còn gọi là hệ số góc) của đường thẳng đi qua gốc tọa độ và đi qua điểm  $\hat{\xi}_k$  như sau:

$$a_k = \frac{\Im\{\hat{\xi}_k\}}{\Re\{\hat{\xi}_k\}} \quad (8.138)$$

trong đó  $\Re\{a\}$  and  $\Im\{a\}$  tương ứng là phần thực và phần ảo của số phức  $a$ . Khi đó, bằng cách so sánh  $a_k$  trong (8.138) với các độ dốc  $b_1, b_2, b_3$ , và  $b_4$  của các đường thẳng liên tục (được gọi là các đường bao  $b$  cho ngắn gọn), chúng ta có thể xác định được khu vực  $\hat{\xi}_k$  nằm. Từ đó, ta dễ dàng xác định được điểm tín hiệu đầu tiên cần thử,  $x_l$ , và độ dốc  $c_l$  của đường đứt quãng đi qua gốc tọa độ và đi qua điểm  $R_{k,k}x_l$ . Để ngắn gọn, những đường đứt quãng đó được gọi là các đường bao  $c$ . Chú ý rằng mỗi một chòm sao tín hiệu PSK có các đường bao  $b$  và  $c$  khác nhau. Cuối cùng, bằng cách so sánh  $a_k$  với  $c_l$ , thứ tự thử tối ưu nhất của các điểm tín hiệu còn lại có thể được xác định một cách dễ dàng.

Ví dụ, quan sát trên Hình 8.22 ta thấy rằng  $b_2 < a_k < b_3$ , do đó  $x_2$  sẽ là điểm đầu tiên thuật toán Tuan SD thử. Thêm vào đó, do  $a_k < c_2$ , thứ tự thử tối ưu của 8 điểm tín hiệu là  $(x_2, x_1, x_3, x_8, x_4, x_7, x_5, x_6)$ . Trong trường hợp  $a_k > c_2$ , thứ tự thử trở thành  $(x_2, x_3, x_1, x_4, x_8, x_5, x_7, x_6)$ .

Trong thuật toán Tuan SD, thứ tự thử tối ưu ở lớp  $k$  được lưu trong các bảng tra cứu  $i_k = (i_1^{(k)}, \dots, i_M^{(k)})$  dưới dạng các chỉ số của các điểm tín hiệu trong  $\mathbf{x}$ . Ví dụ,  $i_1^{(k)} = 6$  chỉ ra rằng tại lớp  $k$ ,  $x_6$  sẽ là điểm tín hiệu đầu tiên trong  $\Omega$  cần phải thử. Rõ ràng là thay vì phải tính toán với hàm  $\cos^{-1}$  và sắp xếp các metric khoảng cách như thuật toán Pham SD, thuật toán Tuan SD đơn giản hơn nhiều do nó xác định thứ tự thử tối ưu của các điểm tín hiệu tại mức  $k$  chỉ với một phép chia số thực.

Thuật toán Tuan SD được tóm tắt như sau.

### Thuật toán Tuan SD

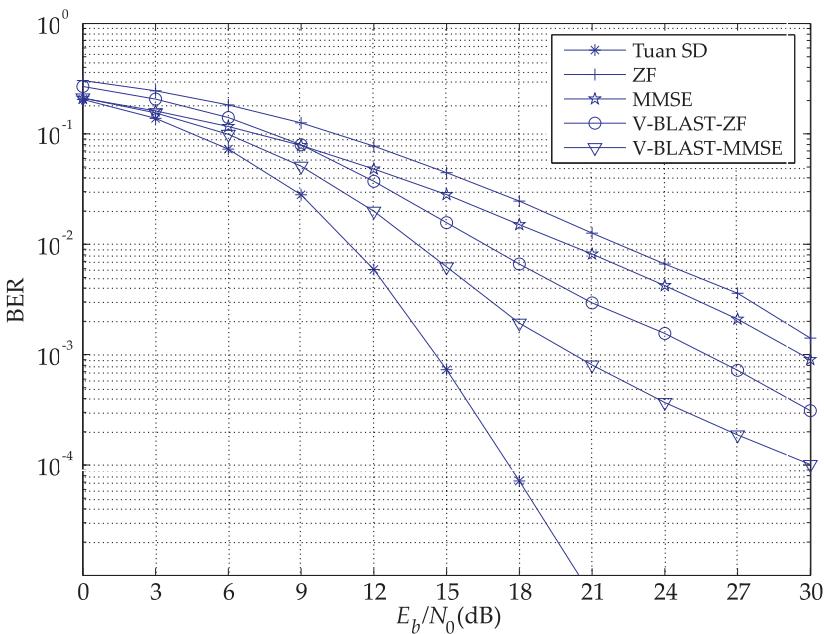
---

BEGIN: Input  $v, R, x, M, C$ .

- (1) **Bước 1.** (Khởi tạo) Đặt  $k := N$ ,  $T_k := 0$ ,  $\xi_k := v_k$ ,  $D_{min} := C$ .
- (2) **Bước 2.** (Tìm kiếm) Sử dụng  $\xi_k$  để tạo bảng tra (lookup table)  $i_k$  cho mức  $k$ , đặt  $l_k := 1$ .
- (3) **Bước 3.** If ( $l_k \leq M$ ), nhận ký hiệu phức  $s_k$  bằng cách đặt  $s_k := x_{i_{l_k}^{(k)}}$  và đặt  $D := |\xi_k - R_{k,k}s_k|^2 + T_k$ .
- (4) **Bước 4.** If ( $D \geq D_{min}$ ) hoặc ( $l_k > M$ ), thì if ( $k = N$ ), ngắt, else đặt  $k := k+1$ ,  $l_k := l_k+1$ , và chuyển tới Bước 3.
- (5) **Bước 5.** If ( $k > 1$ ), đặt  $\xi_{k-1} := v_{k-1} - \sum_{i=k}^N R_{k,i}s_i$ ,  $T_{k-1} := D$ ,  $k := k-1$ , và chuyển tới Bước 2, else đặt  $D_{min} := D$ , lưu kết quả mới  $\bar{s} := s$ , let  $k := 2$ ,  $l_k := l_k + 1$ , và chuyển tới Bước 3.

END: Output  $\bar{s} = (\bar{s}_1, \bar{s}_2, \dots, \bar{s}_N)$ .

---



Hình 8.23: BER của hệ thống  $4 \times 4$  MIMO-SDM; Điều chế 8PSK.

Hình 8.23 so sánh phẩm chất lỗi bit của Tuan SD với các bộ tách tín hiệu VBLAST cũng như tuyến tính khi chúng được áp dụng trong hệ thống MIMO  $4 \times 4$  sử dụng kỹ thuật điều chế 8PSK. Chúng ta thấy rằng, Tuan SD cho phép hệ thống đạt được phẩm chất lỗi bit tối ưu và tốt hơn rất nhiều so với phẩm chất lỗi bit của các bộ tách tín hiệu khác.

Các thuật toán SD phát triển gần đây được chứng minh là có độ phức tạp ở bậc đa thức (thường là đa thức bậc 3 [31]) mà không phụ thuộc vào phương thức điều chế. Do đó, chúng rất phù hợp với các ứng dụng tách tín hiệu hợp lệ tối ưu (MLD) trong các hệ thống MIMO có cấp độ điều chế cao (chòm sao tín hiệu có số điểm tín hiệu lớn).

## 8.7 Tóm tắt

Trong chương này chúng ta đã nghiên cứu về hệ thống MIMO, mô hình toán học cũng như dung lượng của kênh truyền MIMO. Các

kết quả cho thấy rằng, kỹ thuật MIMO cho phép các hệ thống thông tin vô tuyến đạt được dung lượng lớn hơn nhiều lần so với dung lượng của kênh SISO truyền thống.

Bên cạnh đó, bạn đọc cũng có thêm được các kiến thức về kỹ thuật ghép kênh theo không gian và phần quan trọng nhất trong kỹ thuật này là các phương pháp khôi phục tín hiệu tại máy thu. Thông qua chương này, bạn đọc có thể nắm rõ các thuật toán khôi phục tín hiệu tuyến tính, các thuật toán khôi phục tín hiệu có sự trợ giúp rút gọn cơ sở dàn, và các thuật toán giải mã cầu. Đây là những thuật toán cơ bản nhất đã được phát triển cho kỹ thuật ghép kênh theo không gian.

## Tài liệu tham khảo

- [1] C. E. Shannon, *The Mathematical Theory of Communication*. University of Illinois Press, 1949.
- [2] I. E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Transactions on Telecommunications*, vol. 10, no. 6, pp. 585–595, November/December 1999.
- [3] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Communications*, vol. 6, pp. 311–335, 1998.
- [4] H. Shin and J. H. Lee, "On the capacity of MIMO wireless channels," *IEICE Transactions on Communications*, vol. E87-B, no. 3, pp. 671–677, March 2004.
- [5] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C++*, *The art of scientific computing*. Cambridge University Press, 1992.
- [6] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multielement antennas," *Bell Labs Technical Journal*, pp. 41–59, Autumn, 1996.
- [7] L. Hanzo, M. Münster, B. J. Choi, and T. Keller, *OFDM and MC-CDMA for Broadband Multi-user Communications, WLANs and Broadcasting*. Wiley, 2003.
- [8] P. N. Sabes, *Linear Algebraic Equations, SVD, and the Pseudo-Inverse*. Lecture Note, University of California San Francisco, 2001.

- [9] M. Brookes, *The Matrix Reference Manual*. [on-line in internet at] <http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/intro.html>, 2005.
- [10] B. Hassibi, "A fast square-root implementation for BLAST," in *Acoustics, Speech, and Signal Processing (ICASSP'00)*, vol. 2, June 2000, pp. II737 – II740.
- [11] B. Vucetic and J. Yuan, *Space-Time Coding*. Wiley, 2003.
- [12] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel," in *1998 URSI International Symposium on Signals, Systems, and Electronics (ISSSE'98)*, Pisa, Italy, September 1998, pp. 295–300.
- [13] G. D. Golden, G. J. Foschini, R. A. Valenzuela, and P. W. Wolniansky, "Detection algorithm and initial laboratory results using the V-BLAST space-time communication architecture," *Electronics Letters*, vol. 35, no. 1, pp. 14–15, 1999.
- [14] W. H. Mow, "Universal lattice decoding: Principle abd recent advances," *Wireless Communications and Mobile Computing*, vol. 3, no. 5, pp. 553–569, 2003.
- [15] F. Oggier and E. Viterbo, *Algebraic Number Theory and Code Design for Rayleigh Fading Channels*. Now Publisher, 2004, 1966.
- [16] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*.
- [17] D. C. Lay, *Linear Algebra and Its Applications*. Addison Wesley, 2003.
- [18] H. Yao and G. W. Wornell, "Lattice-reduction-aided detectors for MIMO communication systems," in *IEEE Global Communications Conference (GLOBECOM'02)*, vol. 1, November 2002, pp. 424–428.
- [19] C. Windpassinger and R. F. H. Fischer.
- [20] D. Wübben, R. Bohnke, V. Kuhn, and K. D. Kammeyer, "Near-maximum-likelihood detection of MIMO systems using MMSE-based lattice reductio," in *2004 IEEE International Conference on Communications (ICC'04)*, vol. 2, Paris, June 2004, pp. 798–802.
- [21] S. J. Grant and J. K. Cavers, "Performance enhancement through joint detection of cochannel signals using diversity arrays," *IEEE Transactions on Communications*, vol. 46, no. 8, pp. 1038–1049, 1998.

- [22] R. van Nee, A. van Zelst, and G. Awater, "Maximum likelihood decoding in a space division multiplexing system," in *IEEE Vehicular Technology Conference (VTC'00)*, vol. 1, Tokyo, May 2000, pp. 6–10.
- [23] X. Zhu and R. D. Murch, "Performance analysis of maximum likelihood detection in a MIMO antenna system," *IEEE Transactions on Communications*, vol. 50, no. 2, pp. 187–191, 2002.
- [24] J. G. Proakis, *Digital Communications*. McGraw-Hill, 2001.
- [25] X. N. Tran, A. T. Le, and T. Fujino, "Combined MMSE-ML detection for wireless MIMO-SDM communications," in *2005 International Symposium on Antenna and Propagation (ISAP'05)*, Seoul, Korea, August 2005.
- [26] ——, "Performance comparison of MMSE-SIC and MMSE-ML multiuser detectors in a STBC-OFDM system," in *16th IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC'05)*, Berlin Germany, September 2005.
- [27] M. Pohst, "On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications," *SIGSAM Bull.*, vol. 15, no. 1, pp. 37–44, 1981.
- [28] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis."
- [29] E. Viterbo and J. Bouros, "A universal lattice code decoder for fading channels," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1639–1642, July 1999.
- [30] A. M. Chan and I. Lee, "A new reduced-complexity sphere decoder for multiple antenna systems," in *IEEE International Conference on Communications*, vol. 1, April/May, 2002, pp. 460 – 464.
- [31] M. Damen, H. Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2389 – 2402, October 2003.
- [32] B. M. Hochwald and S. Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 389 – 399, March 2003.
- [33] C. P. Schnorr and M. Euchner, "Lattice basis reduction: improved practical algorithms and solving subset sum problems," *Mathematical Programming*, vol. 66, pp. 181 – 191, 1994.

- [34] T. Cui and C. Tellambura, "Joint channel estimation and data detection for ofdm systems via sphere decoding," *In IEEE Proceeding Globecom'04*, vol. 6, pp. 3656 – 3660, December 2004.
- [35] B. Hochwald and S. Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Transactions on Communications*, vol. 51, no. 3, March 2003.
- [36] D. Pham, K. R. Pattipati, P. K. Willett, and J. Luo, "An improved complex sphere decoder for V-BLAST systems," *IEEE Signal Processing Letter*, vol. 11, no. 9, pp. 748 – 751, September 2004.
- [37] L. Lampe, R. Schober, V. Pauli, and C. Windpassinger, "Multiple-symbol differential sphere decoding," *IEEE Transactions on Communications*, vol. 53, no. 12, December 2005.
- [38] L. M. Minh-Tuan Le, Van-Su Pham and G. Yoon, "Rate-one full-diversity quasi-orthogonal STBCs with low decoding complexity," *IEICE Transactions on Communications*, vol. E89-B, no. 12, pp. 3376 – 3385, December 2006.

## Phụ lục: Chương trình MATLAB mẫu

---

### ■ MATLAB Program 8.1 MIMOCapac.m

```
% Chuong trinh ve dung luong ergodic va dung luong outage
% cua kenh MIMO pha dinh Rayleigh
% Written by: Tran Xuan Nam, Le Qui Don Technical University
%           Le Minh Tuan, UVISION JSC.

% Lats updated 24/06/2011

clear all
M=[1:2:7]; % Number of Rx antennas
N=M; % Number of Tx antennas
rhodB=0:5:30; % SNR in dB
rho=10.^((rhodB/10));
Pout=0.05; %Outage probability
NoChReal=1/Pout/0.001; % Number of channel realizations

Ci=zeros(NoChReal,1);
```

```
for n=1:length(rho)
    for k=1:length(M)
        % Generate channels and compute capacity
        for i=1>NoChReal
            H=(randn(M(k),N(k))+j*randn(M(k),N(k)))/sqrt(2);
            Ci(i)=log2(det(diag(ones(1,M(k)))+rho(n)*(H'*H)/N(k)));
        end
        temp=sort(real(Ci));
        %Find Outage Capacity
        Cout(n,k)=temp(Pout*NoChReal+1);
        % Everage C over all channel realizations
        C(n,k)=real(sum(Ci))/NoChReal;
    end
end
txt=['b-*';'b^-';'b-p';'b-o';'b->';'b-<';'b-v'];
hold on
for k=1:min(length(rho),length(txt))
    plot(M,C(k,:),txt(k,:))
end
% title('Ergodic Capacity of MIMO Rayleigh Fading Channel')
box on
xlabel('Number of antennas M=N')
ylabel('Normalized Capacity (bits/s/Hz)')
legend('0dB', '5dB', '10dB', '15dB', '20dB', 'Location', 'NorthWest')
figure
hold on
for k=1:length(M)
    plot(rhodB,C(:,k),txt(k,:))
end
% title('Ergodic Capacity of MIMO Rayleigh Fading Channel')
box on
xlabel('SNR (dB)')
ylabel('Normalized Capacity (bits/s/Hz)')
legend('M=N=1', 'M=N=3', 'M=N=5', 'M=N=7', 'Location', 'NorthWest')
```

```
figure
hold on
for k=1:length(M)
    plot(rhodB,Cout(:,k),txt(k,:))
end
% title('Outage Capacity of MIMO Rayleigh Fading Channel')
box on
xlabel('SNR (dB)')
ylabel(['Normalized ' num2str(Pout*100, '%2.1f')...
        '% Outage Capacity (bits/s/Hz)'])
legend('M=N=1', 'M=N=3', 'M=N=5', 'M=N=7', 'Location', 'NorthWest')
%
%
```

---

**■ MATLAB Program 8.2 MIMOLinDetect.m**

```
% Program to simulate performance of linear MMSE detector for MIMO-SDM
% (C)2005 Tran Xuan Nam, The Univ. of Electro-Communications, Tokyo, Japan

% Paramaters
clear all
N=4;                                % Number of TxAnt
M=4;                                % Number of RxAnt (M>=N)
Am=1/sqrt(N);                         % Signal amplitude, power normalized to 1
EbNodB=0:5:30;
EbNo=10.^{(EbNodB/10)};               % Transmit data block size
BlkSize=100;                           % Number of simulation iterations

% Select ZF/MMSE detectors
reply = input('Please select detectors ZF/MMSE [Default: ZF]: ', 's');
if isempty(reply)
    reply = 'ZF';
elseif ~strcmp(reply,'ZF') && ~strcmp(reply,'MMSE')
    display('Invalid input! Start and try again.');
    return;
end
```

```

end

% Simulation loop
for i=1:IterNo
    data=sign(rand(1,N*BlkSize)-.5);           % BPSK data
    s=reshape(data,N,BlkSize);
    st=Am*s;
    H=(randn(M,N)+j*randn(M,N))/sqrt(2);      % Channel matrix
    for n=1:length(EbNo)
        sigma=1/sqrt(2*EbNo(n));                % Noise deviation
        z=sigma*(randn(M,BlkSize)+j*randn(M,BlkSize));
        y=H*st+z;                                % Rx signal
        switch reply
            case 'ZF'
                W=pinv(H)';
            case 'MMSE'
                R=inv(Am^2*H*H'+sigma^2*diag(ones(1,M)));
                W=R*H*Am^2;
        end
        s_hat=sign(real(W'*y));                  % BPSK demodulation
        ErrChk=s_hat-s;
        ErrCount(i,n)=length(find(ErrChk~=0)); % Counting errors
    end
end
BER=sum(ErrCount)/(N*BlkSize*IterNo);
semilogy(EbNodB,BER,'-o');
grid;
xlabel('Eb/No [dB]')
ylabel('Average BER')
title(['BER of ',num2str(M),'x',num2str(N), ...
        ' MIMO using linear ',reply,])
%

```

### MATLAB Program 8.3 MIMOQRDetect.m

```
% Program to simulate performance of QR detectors for MIMO-SDM
```

```
% (C)2005 Tran Xuan Nam, The Univ. of Electro-Communications, Tokyo,  
% Japan  
  
% Paramaters  
clear all  
N=4; % No of TxAnt  
M=4; % No of RxAnt (M>=N)  
Am=sqrt(1/N); % Signal amplitude, power normalized to 1  
EbNodB=0:5:30;  
EbNo=10.^{EbNodB/10};  
BlkSize=100; % No of data block, sending data in blocks  
IterNo=10000; % No of simulation iterations  
  
% Simulation iteration  
for i=1:IterNo  
    for n=1:length(EbNo)  
        data=sign(rand(1,N*BlkSize)-.5); % BPSK data  
        s=reshape(data,N,BlkSize);  
        st=Am*s;  
        H=(randn(M,N)+j*randn(M,N))/sqrt(2); % MIMO channel H~Nc(0,1)  
        sigma=1/sqrt(2*EbNo(n)); % sqrt of noise variance  
        z=sigma*(randn(M,BlkSize)+j*randn(M,BlkSize)); % AWGN  
        y=H*st+z;  
        [Q,R]=qr(H); % QR factorization  
        r=Q'*y;  
        s_hat=zeros(N,BlkSize);  
        for k=N:-1:1  
            if k==N  
                s_hat(k,:)=sign(real(r(k,:)/R(k,k)));  
            else  
                s_hat(k,:)=sign(real((r(k,:)-Am*R(k,k+1:end)*...  
                    s_hat(k+1:end,:))/R(k,k)));  
            end  
            ErrChk=s_hat(k,:)-s(k,:);  
            Error(k)=length(find(ErrChk~=0)); % Counting errors
```

```

    end
    ErrCount(i,n)=sum(Error); % Summing accross rows
end
BER=sum(ErrCount)/(N*BlkSize*IterNo)
semilogy(EbNodB,BER,'-p');
grid;
xlabel('Eb/No [dB]')
ylabel('Average BER')
title(['BER of ',num2str(M),'x',num2str(N), ' MIMO using QR detector'])
%!

```

#### MATLAB Program 8.4 VBLASTDetect.m

```

% Program to simulate performance of V-BLAST detectors for MIMO-SDM
% (C)2005 Tran Xuan Nam, The Univ. of Electro-Communications, Tokyo,
% Japan
% Date: 15/03/2005

% Paramaters
clear all
N=4; % No of TxAnt
M=4; % No of RxAnt
Am=sqrt(1/N); % Signal amplitude, power normalized to 1
EbNodB=0:5:25;
EbNo=10.^{(EbNodB/10)};
BlkSize=500; % No of data block
IterNo=30000; % No of simulation iterations

% Select ZF/MMSE detectors
reply = input('Select detectors ZF/MMSE [Default: ZF]:','s');
if isempty(reply)
    reply = 'ZF';
elseif ~strcmp(reply,'ZF') && ~strcmp(reply,'MMSE')
    display('Invalid input! Start and try again.');
    return;

```

```
end

% Simulation iteration
ErrCount=zeros(IterNo,length(EbNo));
for i=1:IterNo
    for n=1:length(EbNo)
        %data=bpskgen(N*BlkSize);
        data=1-2*(randn(1,N*BlkSize)>0);
        s=reshape(data,N,BlkSize);
        st=Am*s;
        H=(randn(M,N)+j*randn(M,N))/sqrt(2); % Channel H~N(0,1)
        sigma=1/sqrt(2*EbNo(n)); % noise deviation
        z=sigma*(randn(M,BlkSize)+j*randn(M,BlkSize));% AWGN
        r=H*st+z;
        for k=1:N
            G=pinv(H);
            P=diag(G*G');
            k_min=find(P==min(P)); % k= arg min ||(G_1)_j||^2
            switch reply
                case 'ZF'
                    w=G(k_min,:);
                case 'MMSE'
                    R=inv(Am^2*H*H'+(1/EbNo(n))*diag(ones(1,M)));
                    w=R*H(:,k_min)*Am^2;
            end
            s_hat=sign(real(w'*r)); % BPSK demodulation
            ErrChk=s_hat-s(k_min,:);
            r=r-Am*H(:,k_min)*s_hat; % Cancellation
            H(:,k_min)=[];
            % Deleting column k_min of H
            s(k_min,:)=[];
            Error(k)=length(find(ErrChk~=0));% Counting errors
        end
        ErrCount(i,n)=sum(Error); % Summing error accross rows
    end
end
```

```

BER=sum(ErrCount)/(N*BlkSize*IterNo)
semilogy(EbNodB,BER,'-v');
grid;
axis([0 30 1e-5 1])
xlabel('Eb/No [dB]')
ylabel('Average BER')
title(['BER of ',num2str(M),'x',num2str(N), ...
        ' MIMO using V-BLAST-',reply])
%

```

### MATLAB Program 8.5 MIMOMLDetect.m

```

% Program to simulate performance of ML detector for MIMO-SDM
% (C)2005 Tran Xuan Nam, The Univ. of Electro-Communications

% Paramaters
clear all
N=4; % Number of TxAnt
M=4; % Number of RxAnt (M>=N)
Am=1/sqrt(N); % Signal amplitude(zeta), power normalized to 1
EbNodB=0:5:15;
EbNo=10.^{(EbNodB/10)};
IterNo=100000; % Number of simulation iterations

% Generate signal look-up table for BPSK
TxData=0:2^N-1; % Tx data in decimal
TxBin=dec2bin(TxData)'; % convert Tx data to binary
Tx2Bin=zeros(N,2^N-1);
for i=1:2^N
    for k=1:N
        Tx2Bin(k,i)=str2num(TxBin(k,i));
    end
end
stSet=(2*Tx2Bin-1); % All sets of possible Tx signals
stSetTx=Am*stSet;

```

```
% Simulation loop
for i=1:IterNo
    data=rand(1,N)-.5; % BPSK data
    s=reshape(data,N,1);
    st=Am*s;
    H=(randn(M,N)+j*randn(M,N))/sqrt(2); % Channel matrix
    for n=1:length(EbNo)
        sigma=1/sqrt(2*EbNo(n)); % Noise deviation
        z=sigma*(randn(M,1)+j*randn(M,1));% AWGN
        y=H*st+z; % Rx signal
        for k=1:2^N
            d(k)=norm(y-H*stSetTx(:,k))^2;% Compute ML metric
        end
        k_min=find(d==min(d)); % Find a set with smallest metric
        s_hat=stSet(:,k_min); % BPSK demodulation
        ErrChk=s_hat-s;
        ErrCount(i,n)=length(find(ErrChk~=0)); % Counting errors
    end
    BER=sum(ErrCount)/(N*IterNo);
    semilogy(EbNodB,BER);
    grid;
    xlabel('Eb/No [dB]')
    ylabel('Average BER')
    title(['BER of ',num2str(M),'x',num2str(N), ' MIMO using ML Detector'])
%!
```

### **■ MATLAB Program 8.6 latreduc.m**

```
function [T,H_red] = latreduc(H,delta)
% Function to perform lattice reduction based on LLL algorithm
% function [T,H_red] = latreduc(H)
%
%     H      : original basis,
%     H_red: reduced lattice basis
%     T      : transform matrix, i.e., H_red = H*T
%     delta: reduction factor (default: delta=3/4)
```

% (C)2005 by Tran Xuan Nam and Nguyen Tien Duc  
% The Univ. of Electro-Communications, Tokyo, Japan

```

% Check function inputs
if nargin==1
    delta=3/4;
elseif nargin>2
    disp('Incorrect input arguments')
    help latreduc;
    return;
end

[M,N]=size(H);           % returns number of RxAnt M and TxAnt N
i=2;                      % Initial value of i;
T= eye(N);
mu=eye(N,N);
while i <= N
    % Iteration for weak reduction
    H_hat(:,1)=H(:,1);
    for j=i-1:-1:1;
        mu(i,j)=round(H_hat(:,j)'*H(:,i)/(H_hat(:,j)'*H_hat(:,j)));
        H(:,i)=H(:,i)-mu(i,j)*H(:,j);
        T(:,i)=T(:,i)-mu(i,j)*T(:,j);
    end
    % Reduction iteration for updating basis vectors
    H_hat(:,i)=H(:,i);
    for j=i-1:-1:1;
        mu(i,j)=H_hat(:,j)'*H_hat(:,i)/(H_hat(:,j)'*H_hat(:,j));
        H_hat(:,i)=H_hat(:,i)-mu(i,j)*H_hat(:,j);% from Lenstra
    end
    if (delta*norm(H_hat(:,i-1))^2>norm(H_hat(:,i))...
        + mu(i,i-1)*H_hat(:,i-1))^2)
        H(:,[i-1,i])=H(:,[i,i-1]); % Swap columns i and i-1
        T(:,[i-1,i])=T(:,[i,i-1]);
    end
end

```

```

    i=max(i-1,2);

else
    i=i + 1;
end

end
H_red=H;
%!

```

### ■ MATLAB Program 8.7 MIMOLRADetect.m

```

% Program to simulate performance of MMSE detector for MIMO-SDM
% (C)2005 Nguyen Tien Duc & Tran Xuan Nam
% The Univ. of Electro-Communications, Tokyo, Japan

```

```

clear all;

% Select ZF/MMSE detectors
reply = input('Select detectors ZF/MMSE [Default: ZF]:' , 's');
if isempty(reply)
    reply = 'ZF';
elseif ~strcmp(reply,'ZF') && ~strcmp(reply,'MMSE')
    display('Invalid input! Start and try again.');
    return;
end

% Input Parameters
M=4;           % Number of RxAnt
N=4;           % Number of TxAnt
% Define 4-QAM signal
Mc=4;          % Modulation order (fixed for 4-QAM)
sigAlpha=0:Mc-1; % Signal alphabet
Es=2;          % Average energy per symbol of 4-QAM
Esn=Es/N;      % Normalized Es
Eb=Esn*log2(Mc); % Average bit energy
blkSize=50;     % Block size (number of symbols)
EbNodB=0:5:20;

```

```

EbNo=10.^ (EbNodB/10) ;
sigma=sqrt(N*Eb./EbNo); % sqrt of noise variance
noIter=10e3; % Number of simulation iterations
% Simulation loop
for i=1:noIter
    % Generate Tx signal
    txSymb=randsrc(N,blkSize,sigAlph);% Random Tx integer symbols
    s=qammod(txSymb,Mc); % 4QAM modulation
    st=Esn*s; % Tx signal
    H=(randn(M,N)+j*randn(M,N))/sqrt(2);% Channel matrix
    switch reply
        case 'ZF'
            [T,Hred]=latreduc(H);% Return H reduced & T matrices
        case 'MMSE'
            for n=1:length(EbNo)
                % Extended matrix
                H_bar=[ H ; sqrt(EbNo(n)^-1)*eye(N)];
                [T_mmse(:,:,n),Hred(:,:,n)]=latreduc(H_bar);
            end
        end
        for n=1:length(EbNo)
            z=sigma(n)*(randn(M,blkSize)+j*randn(M,blkSize));% AWGN
            y=H*st+z; % Rx signal
            switch reply
                case 'ZF'
                    W=pinv(Hred)';
                case 'MMSE'
                    W=pinv(Hred(:,:,n))';
                    W=W(1:N,1:N);
                    T=T_mmse(:,:,n);
                end
            % Estimates of u
            u_hat=round(W'*y-inv(T)*(ones(N,blkSize)*(1+j)/2));
            s_hat=T*u_hat;
            s_bar=(real(s_hat)>-0.5)+(imag(s_hat)>-0.5)*j - ...
        end
    end
end

```

```

    1/2*(1+j)*ones(N,blkSize);
errChk=round(s_bar-st);
errCount(i,n)=sum(sum(real(errChk)~=0))...
+sum(sum(imag(errChk)~=0));
end
end
BER=sum(errCount)/(N*blkSize*noIter*log2(Mc));
semilogy(EbNodB,BER,'-o');
hold on
grid;
xlabel('Eb/No [dB]')
ylabel('Average BER')
title(['BER of ',num2str(M),'x',num2str(N), ' MIMO using LRA- ', reply,])
%!

```

### MATLAB Program 8.8 SESD16qamDetect.m

```

% Program to simulate performance of SE-SD decoder for MIMO-SDM
% 16QAM Modulation.
% (C)2011 Le Minh Tuan, UVISION JSC.
% Date: 16/07/2011

clear all;

N=4; % No of TxAnt
M=4; % No of RxAnt
ml=4; % ml:number of bit per symbol
EbNodB=0:3:21;
EbNo=10.^{(EbNodB/10)};
BlkSize=[10 10 10 20 30 40 40 40 50 100] ;
IterNo=[400 400 400 400 400 400 1000 2000 3000 6000];
Es = 10; %16QAM Average Symbol Energy
%Integer values for I and Q channels
iv=[-3 -1 3 1]';
sigma=sqrt(N*Es./(2*ml*EbNo)); % noise deviation
% Simulation iteration

```

```

for n=1:length(EbNo)
    for i=1:IterNo(n)
        %16QAM modulation
        x=randint(N*BlkSize(n),2,ml);
        s=iv(x(:,1)+1)+j*iv(x(:,2)+1);
        st=reshape(s,N,BlkSize(n));
        %MIMO channel
        Hi=randn(M,N)/sqrt(2);
        Hq=randn(M,N)/sqrt(2);
        %Noise
        z=sigma(n)*(randn(M,BlkSize(n))+j*randn(M,BlkSize(n)));
        %Received Signal Vector
        r=(Hi+j*Hq)*st+z;
        Hr=[Hi -Hq;Hq Hi];
        %QR Decomposition
        [Q R]=QRD(Hr);
        y=Q'*[real(r);imag(r)];
        %Detection using sphere decoder
        for k=1:BlkSize(n)
            [dout(:,k)]=SESD(y(:,k),R,1000,4);
        end
        %Calculate BER
        dataout=zeros(2*N,BlkSize(n));
        for kk=0:length(iv)-1
            dataout=dataout+kk*(dout==iv(kk+1));
        end
        data=[reshape(x(:,1),N,BlkSize(n));...
               reshape(x(:,2),N,BlkSize(n))];
        ne(i)=biterr(data,dataout,2);
    end
    ber(n)=sum(ne)/N/BlkSize(n)/ml/IterNo(n)/2;
end
semilogy(EbNodB,ber,'-*')
xlabel('E_b/N_0 (dB)')
ylabel('BER')

```

```

hold on
title(['BER of ',num2str(M), 'x', num2str(N), ...
        ' MIMO using SE-SD Decoder'])
%!

```

### ■ Function 8.1 QRD.m

```

function [Q,R]=QRD(M)
% Program to perform QR factorization
% (C)2011 Le Minh Tuan, UVISION JSC.
% Date: 16/07/2011

%M: Input matrix to be factorized
%Q: Orthogonal matrix
%R: Upper triangular matrix

[m,n]=size(M);
R=zeros(n,n);
M1=M;
for k=1:n
    R(k,k)=norm(M1(:,k));
    Q(:,k)=M1(:,k)/R(k,k);
    for kk=k+1:n
        R(k,kk)=Q(:,k)'*M1(:,kk);
        M1(:,kk)=M1(:,kk)-Q(:,k)*R(k,kk);
    end
end
%!

```

### ■ Function 8.2 SESD.m

```

function [Sout]=SESD(y,R,C,MLen)
% SE-SD decoder for MIMO-SDM with QAM modulation.
% (C)2011 Le Minh Tuan, UVISION JSC.
% Date: 16/07/2011

```

```
%y: input signal  
%R: upper triangular matrix  
%C: initial sphere radius  
%MLen: maximum tested points per layer  
%Sout: Recovered I and Q Signals
```

```

start=1;
NN=length(y);
T=zeros(NN,1);
Shi=zeros(NN,1);
Len=zeros(NN,1);
Delta=zeros(NN,1);

while(start==1)
    k=NN;
    T(k)=0;
    Shi(k)=0;
    Dc=C;
    validpoint=0;
    Enable1=1;
    start1=1;
    while(start1==1)
        if(Enable1==1)
            u=(y(k)-Shi(k))/R(k,k);
            %Rounding to nearest point, 16QAM
            if u>0
                if u<2
                    ui(k,1)=1;
                else
                    ui(k,1)=3;
                end
            else
                if u>-2
                    ui(k,1)=-1;
                else

```

```
ui(k,1)=-3;
end
end
%Determine DELTA
temp=y(k)-Shi(k)-R(k,k)*ui(k,1);
Delta(k)=2*sign(temp+1e-20);
Len(k)=1;
end
if Len(k)<=MLen
    temp3=T(k)+temp*temp;
end
Enable2=1;
if((temp3>=Dc) || (Len(k)>MLen))
    if(k==NN)
        start1=0;
        break;
    else
        k=k+1;
        Len(k)=Len(k)+1;
        Enable1=0;
        Enable2=0;
        Enable3=1;
    end
end

if(Enable2==1)
    if(k>1)
        kk=k;
        k=k-1;
        Shi(k)=R(k,kk:NN)*ui(kk:NN,1);
        T(k)=temp3;
        Enable1=1;
        Enable3=0;
    else
        if(Dc>temp3)
```

```
Dc=temp3;
Sout=ui;
validpoint=1;
end
k=k+1;
Len(k)=Len(k)+1;
Enable1=0;
Enable3=1;
end
if(Enable3==1)
    %Calculate new point in the lattice
    ui(k)=ui(k)+Delta(k);
    Delta(k)=-Delta(k)-2*sign(Delta(k)+1e-18);
    if((abs(ui(k))>3)&&(Len(k)<=MLen))
        ui(k)=ui(k)+Delta(k);
        Delta(k)=-Delta(k)-2*sign(Delta(k)+1e-18);
        Enable3=1;
    end
    temp=y(k)-Shi(k)-R(k,k)*ui(k);
    Enable1=0;
    Enable2=0;
end
if(validpoint~=1)
    C=C*1.2;
else
    start=0;
    break;
end
end
%!
```

**■ MATLAB Program 8.9 TuanSD8pskDetect.m**

```
% Program to simulate performance of Tuan-SD decoder for MIMO-SDM  
% 8PSK Modulation.  
% (C)2011 Le Minh Tuan, UVISION JSC.  
% Date: 16/07/2011  
  
clear all;  
  
N=4; % No of TxAnt  
M=4; % No of RxAnt  
ml=3; % ml:number of bit per symbol  
EbNodB=0:3:21;  
EbNo=10.^{(EbNodB/10)};  
BlkSize=[10 10 10 20 30 40 50 100] ;  
IterNo=[400 400 600 600 1000 2000 4000 10000];  
Es = 1; %8PSK Average Symbol Energy  
  
%8PSK Constellation  
CN=2^ml;  
Const=pskmod([0:CN-1],CN,pi/CN);  
%Gray Mapping  
Pos=[0 1 3 2 6 7 5 4];  
[a indx]=sort(Pos);  
ConstGray=Const(indx).';  
  
%Look-up table for 8PSK  
LTab=[2 3 1 4 8 5 7 6; 2 1 3 8 4 7 5 6;...  
     1 2 8 3 7 4 6 5; 1 8 2 7 3 6 4 5;...  
     8 1 7 2 6 3 5 4; 8 7 1 6 2 5 3 4;...  
     7 8 6 1 5 2 4 3; 7 6 8 5 1 4 2 3;...  
     4 5 3 6 2 7 1 8; 4 3 5 2 6 1 7 8;...  
     3 4 2 5 1 6 8 7; 3 2 4 1 5 8 6 7;...  
     6 7 5 8 4 1 3 2; 6 5 7 4 8 3 1 2;...  
     5 6 4 7 3 8 2 1; 5 4 6 3 7 2 8 1];
```

```

sigma=sqrt(N*Es./(2*ml*EbNo)); % noise deviation

% Simulation iteration
for n=1:length(EbNo)
    for i=1:IterNo(n)
        % 8PSK modulation
        data=randint(N,BlkSize(n),CN);
        st=ConstGray(data+1);
        %MIMO channel
        H=(randn(M,N)+j*randn(M,N))/sqrt(2);
        %Noise
        z=sigma(n)*(randn(M,BlkSize(n))+j*randn(M,BlkSize(n)));
        %Received Signal Vector
        r=H*st+z;
        %QR Decomposition
        [Q R]=QRD(H);
        y=Q'*r;
        for k=1:BlkSize(n)
            [Sout(:,k) dout(:,k)]=TuanSD(y(:,k),R,1000,Const.',LTab);
        end
        %Gray Demapping
        dataout=Pos(dout);
        ne(i)=biterr(data,dataout,ml);
    end
    ber(n)=sum(ne)/N/BlkSize(n)/ml/IterNo(n);
end

semilogy(EbNodB,ber,'-*')
xlabel('E_b/N_0 (dB)')
ylabel('BER')
hold on
title(['BER of ',num2str(M),'x',num2str(N),...
        ' MIMO using Tuan SD Decoder'])
%!

```

**Function 8.3 TuanSD.m**

```
function [Sout,dout]=TuanSD(y,R,C,Const,LTab)
% Tuan-SD decoder for MIMO-SDM with PSK modulation
% (C)2011 Le Minh Tuan, UVISION JSC.
% Date: 16/07/2011

%y: input signal
%R: upper triangular matrix
%C: initial sphere radius
%Const: PSK Constellation
%LTab: Lookup Table
%Sout: Recovered Signal
%dout: Recovered integer data

start=1;
NN=length(y);
CS=length(Const);
T=zeros(NN,1);
Shi=zeros(NN,1);
Len=zeros(NN,1);

while(start==1)
    k=NN;
    T(k)=0;
    Shi(k)=y(k);
    Dc=C;
    validpoint=0;
    Enable1=1;
    start1=1;
    while(start1==1)
        if(Enable1==1)
            Trow=TableLookUp(Shi(k));
            Indx(k,:)=LTab(Trow,:);
            Len(k)=1;
        end
    end
end
```

```

if Len(k)<=CS
    temp=Shi(k)-R(k,k)*Const(Indx(k,Len(k)));
    D=T(k)+temp*conj(temp);
end
Enable2=1;
if((D>=Dc)|| (Len(k)>CS))
    if(k==NN)
        start1=0;
        break;
    else
        k=k+1;
        Len(k)=Len(k)+1;
        Enable1=0;
        Enable2=0;
    end
end

if(Enable2==1)
    if(k>1)
        kk=k;
        k=k-1;
        Cindx=[kk:NN] +(Len(kk:NN)-1)*NN;
        Shi(k)=y(k)-R(k,kk:NN)*Const(Indx(Cindx));
        T(k)=D;
        Enable1=1;
    else
        Dc=D;
        Cindx2=[1:NN] +(Len(1:NN)-1)*NN;
        Sout=Const(Indx(Cindx2));
        dout=Indx(Cindx2);
        validpoint=1;
        k=k+1;
        Len(k)=Len(k)+1;
        Enable1=0;
    end

```

```
    end
  end
  if(validpoint~=1)
    C=C*1.2;
  else
    start=0;
  break;
end
end
%!
```

---

**Function 8.4 TableLookUp.m**

```
function [Trow]=TableLookUp(Shi)
% Function to determine the Table Row for 8PSK modulation
% (C)2011 Le Minh Tuan, UVISION JSC.
% Date: 16/07/2011
%Shi: input for calculating the slopes
%Trow: row of the Lookup Table

%b and c boundaries for 8PSK
bc=[0.41421356237; 1.0; 2.41421356237];

iShi=imag(Shi);
rShi=real(Shi);
temp=abs(iShi/rShi);

t1=[rShi>=0; iShi>=0];
t2=temp>=bc;
t=[16 8]*t1+ [4 2 1]*t2;
Value=[31 30 28 24 16 20 23 22 8 12 15 14 7 6 4 0];
Trow=find(t==Value);

end
%!
```





## Chương 9

# Mã không gian – thời gian

### 9.1 Giới thiệu

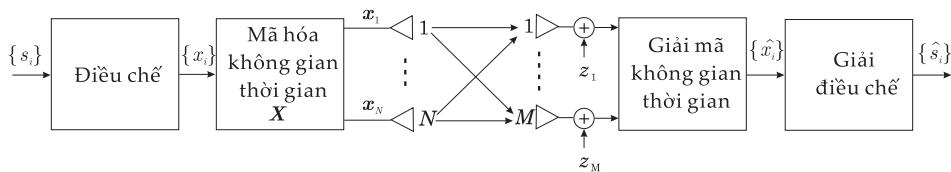
Mã hóa không gian – thời gian (STC: Space-Time Coding) [1],[2],[3],[4] là kỹ thuật mã hóa được sử dụng cho trường hợp máy phát được trang bị nhiều ăng-ten phát. Sự mã hóa được thực hiện trong cả miền không gian và miền thời gian nhằm tạo ra sự tương quan giữa các tín hiệu được phát đi từ các ăng-ten khác nhau và tại những thời điểm khác nhau. Nhờ đó ta có thể cực đại hóa hệ số tăng ích phân tập (diversity gain) và hệ số tăng ích mã hóa (coding gain). Nói cách khác, kỹ thuật mã hóa không gian thời gian là kỹ thuật cho phép chúng ta thực hiện phân tập không gian phát. Kỹ thuật phân tập không gian-thời gian của Alamouti hay kỹ thuật phân tập phát giữ chậm trình bày ở Chương 6 có thể coi là các bộ mã không gian-thời gian đơn giản.

Nền móng của kỹ thuật mã hóa không gian-thời gian (gọi tắt là kỹ thuật STC) bắt nguồn từ sự ra đời của *mã lưới không gian-thời gian* (hay STTC) [2]. Một ưu điểm của mã STTC là chúng có thể cung cấp cả hệ số tăng ích phân tập và hệ số tăng ích mã hóa. Tuy nhiên, để giải mã ta phải sử dụng thuật toán giải mã Viterbi, dẫn tới mức độ phức tạp quá trình khôi phục tín hiệu lớn (biến thiên theo hàm mũ). Kỹ thuật STC trở nên ngày càng thông dụng hơn với sự ra đời của

kỹ thuật mã hóa khôi không gian-thời gian (STBC: Space-Time Block Coding).

Trong các phần sau đây, chúng ta sẽ nghiên cứu chi tiết thêm về các hai loại mã này. Đôi với loại mã STBC phần trình bày sẽ chỉ giới hạn vào loại mã STBC tuyến tính trực giao do loại mã này cho phép đơn giản hóa việc giải mã ở máy thu thông qua các phép xử lý tín hiệu tuyến tính đơn giản.

## 9.2 Mã khôi không gian – thời gian



**Hình 9.1:** Sơ đồ khái niệm của một hệ thống mã hóa STBC.

Sơ đồ một hệ thống MIMO sử dụng STBC được mô tả Hình vẽ 9.1. Một cách tổng quát, bộ mã hóa (STE: Space-Time Encoder) của STBC có thể có như một phép ánh xạ của một tập  $N_s$  ký hiệu phát phức sang một ma trận phát  $\mathbf{X}$  với  $N$  hàng và  $N_t$  cột. Nhìn chung, phép ánh xạ

$$\{s_1, s_2, \dots, s_{N_s}\} \rightarrow \mathbf{X} \quad (9.1)$$

có thể ở bất kỳ dạng nào như tuyến tính hoặc phi tuyến. Tuy nhiên, do STBC phi tuyến ít được quan tâm, trong phần sau đây chúng ta sẽ giới hạn trình bày vào loại mã STBC tuyến tính. Với mã STBC tuyến tính thì phép ánh xạ (9.1) có thể được biểu diễn ngắn gọn như sau [5]:

$$\mathbf{X} = \sum_{k=1}^{N_s} (\tilde{s}_k \mathbf{A}_k + j\check{s}_k \mathbf{B}_k) \quad (9.2)$$

trong đó  $\tilde{s}_k = \Re\{s_k\}$  và  $\check{s}_k = \Im\{s_k\}$  tương ứng biểu diễn phần thực và ảo của  $s_k$ , còn  $\mathbf{A}_k$  và  $\mathbf{B}_k$  là các ma trận cố định cho trước được gọi là

các *ma trận phân tán* (dispersion matrix). Ma trận  $\mathbf{X}$  này được truyền tới máy thu thông qua  $N$  ăng-ten và  $N_t$  khe thời gian. véc-tơ tín hiệu phát từ ăng-ten  $n$  (thông qua  $N_t$  khe thời gian) được biểu diễn bởi hàng  $n$  của  $\mathbf{X}$  và véc-tơ tín hiệu phát đồng thời qua  $N$  ăng-ten tại một khe thời gian  $t$  được biểu diễn bằng cột thứ  $t$  của  $\mathbf{X}$ . Như vậy, một phần tử  $x_{n,t}$  biểu diễn một ký hiệu phát mã hoá được phát đi từ ăng-ten  $n$  tại khe thời gian  $t$ .

Do  $N_s$  ký hiệu được truyền đi trong khoảng thời gian  $N_t$  khe thời gian nên tốc độ truyền của mã STBC là

$$R = \frac{N_s}{N_t}. \quad (9.3)$$

Giả sử bộ điều chế được sử dụng có bậc  $\mathcal{M}$  thì số bít được truyền đi trong mỗi ký hiệu phát là  $m_b = \log_2 \mathcal{M}$ . Như vậy, hiệu suất phổ tần của mã STBC có thể tính được như sau [6, Ch.3]

$$\eta = \frac{r_b}{B} = \frac{r_s m_b R}{r_s} = \frac{m_b N_s}{N_t} \quad [\text{bits/s/Hz}] \quad (9.4)$$

trong đó  $r_b$  và  $r_s$  tương ứng là các tốc độ truyền bit và truyền ký hiệu, còn  $B$  là băng tần tín hiệu.

Một ví dụ điển hình của mã STBC là bộ mã do Alamouti [1] đề xuất với ma trận phát

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{bmatrix} = \begin{bmatrix} s_1 & s_2^* \\ s_2 & -s_1^* \end{bmatrix}. \quad (9.5)$$

Các ma trận ánh xạ  $\mathbf{A}_k$  và  $\mathbf{B}_k$  của mã Alamouti STBC có thể biểu diễn như sau [5]

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (9.6)$$

$$\mathbf{B}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{B}_2 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \quad (9.7)$$

Do  $N_s = N_t = 2$  nên tốc độ truyền dẫn của mã Alamouti STBC là  $R = 1$ .

Để thu được bậc phân tập toàn phần (full diversity)  $N$  do  $N$  ăng-ten phát cung cấp, ma trận mã truyền  $\mathbf{X}$  được thiết kế trên cơ sở trực giao sao cho [4]

$$\mathbf{X} \mathbf{X}^H = C \sum_{k=1}^{N_s} |s_k|^2 \mathbf{I}_N \quad (9.8)$$

trong đó  $C$  là một hằng số nào đó. Loại mã STBC được xây dựng trên cơ sở trực giao này được gọi là mã STBC trực giao (O-STBC: orthogonal STBC). Với mã STBC trực giao, các véc-tơ phát từ các ăng-ten trực giao với nhau. Tức là nếu chúng ta định nghĩa  $\mathbf{x}_n$  là các véc-tơ (hàng) phát từ ăng-ten  $n$ , tương ứng với hàng thứ  $n$  của  $\mathbf{X}$ , thì

$$\mathbf{x}_i \mathbf{x}_j^H = \sum_{t=1}^{N_t} x_{i,t} x_{j,t}^* = 0, \quad i, j \in \{1, 2, \dots, N\}. \quad (9.9)$$

Tính chất trực giao của mã STBC này cho phép đạt được bậc phân tập toàn phần, đồng thời cho phép máy thu tách riêng các thành phần tín hiệu phát từ các ăng-ten phát, và vì vậy đơn giản hóa bộ tách tín hiệu thông qua tách sóng hợp lẽ tối ưu dựa trên các phép xử lý tuyến tính trên tín hiệu thu.

Tốc độ truyền dẫn tối đa của các mã STBC trực giao có bậc phân tập toàn phần là 1, tức là,  $R \leq 1$  [4], trong đó mã Alamouti là bộ mã cho tín hiệu phức duy nhất cho phép đạt được đồng thời cả bậc phân tập toàn phần và tốc độ toàn phần (full rate).

Tùy theo tập tín hiệu (signal constellation), chúng ta có thể thiết kế được các bộ mã STBC trực giao khác nhau. Trong phần sau đây chúng ta sẽ tìm hiểu về các bộ mã dành cho tập tín hiệu thực và tập tín hiệu phức.

### 9.2.1 Mã hóa khối không gian thời gian trực giao

#### A. Mã O-STBC cho tập tín hiệu thực

Một ví dụ điển hình của tín hiệu thực là tập tín hiệu sử dụng phương pháp điều chế biên độ xung (PAM: Pulse Amplitude Modulation),

trong đó biên độ của tín hiệu sau điều chế thay đổi theo tín hiệu điều chế.

Để đạt được độ phân tách toàn phần, các bộ mã STBC cho các hệ thống  $N$  ăng-ten phát sử dụng tập tín hiệu thực cũng cần phải thỏa mãn điều kiện trực giao (9.8)

$$\mathbf{X}_N \mathbf{X}_N^T = C \sum_{k=1}^{N_s} |s_k|^2 \mathbf{I}_N. \quad (9.10)$$

Dựa định lý Hurwitz-Radon về thiết kế trực giao, Tarokh, Jafarkhani và Calderbank [4] đã chứng minh được rằng các bộ mã vuông ( $N = N_t$ ) có thể đạt được đồng thời cả bậc phân tách toàn phần và tốc độ toàn phần cho tập tín hiệu thực chỉ tồn tại với số ăng-ten phát  $N = 2, 4, 8$ . Ma trận truyền dẫn của các bộ mã này là [4]

$$\mathbf{X}_2 = \begin{bmatrix} s_1 & -s_2 \\ s_2 & s_1 \end{bmatrix}, \quad (9.11)$$

$$\mathbf{X}_4 = \begin{bmatrix} s_1 & -s_2 & -s_3 & -s_4 \\ s_2 & s_1 & s_4 & -s_3 \\ s_3 & -s_4 & s_1 & s_2 \\ s_4 & s_3 & -s_2 & s_1 \end{bmatrix}, \quad (9.12)$$

$$\mathbf{X}_8 = \begin{bmatrix} s_1 & -s_2 & -s_3 & -s_4 & -s_5 & -s_6 & -s_7 & -s_8 \\ s_2 & s_1 & -s_4 & s_3 & -s_6 & s_5 & s_8 & -s_7 \\ s_3 & s_4 & s_1 & -s_2 & -s_7 & -s_8 & s_5 & s_6 \\ s_4 & -s_3 & s_2 & s_1 & -s_8 & s_7 & -s_6 & s_5 \\ s_5 & s_6 & s_7 & s_8 & s_1 & -s_2 & -s_3 & -s_4 \\ s_6 & -s_5 & s_8 & -s_7 & s_2 & s_1 & s_4 & -s_3 \\ s_7 & -s_8 & -s_5 & s_6 & s_3 & -s_4 & s_1 & s_2 \\ s_8 & s_7 & -s_6 & -s_5 & s_4 & s_3 & -s_2 & s_1 \end{bmatrix}. \quad (9.13)$$

Để ý rằng các bộ mã  $\mathbf{X}_2, \mathbf{X}_4, \mathbf{X}_8$  sử dụng  $N_t$  khe thời gian để truyền đi  $N_s = N_t$  ký hiệu phát nên có tốc độ truyền  $R = 1$ . Các bộ mã này

cũng thoả mãn tính chất trực giao do

$$\mathbf{x}_i \mathbf{x}_i^H = \sum_{k=1}^{N_s} |s_k|^2 \quad (9.14)$$

$$\mathbf{x}_i \mathbf{x}_j^H = \mathbf{0}, \quad i \neq j. \quad (9.15)$$

Tổng quát hoá định lý Hurwitz-Radon cho trường hợp ma trận không vuông, Tarokh, Jafarkhani và Calderbank [4] đã tìm được các ma trận phát  $\mathbf{X}_N$  cho bất kỳ số ăng-ten phát  $N$  nào. Các ví dụ ma trận phát cho  $N \leq 8$  được trình bày dưới đây [4]

$$\mathbf{X}_3 = \begin{bmatrix} s_1 & -s_2 & -s_3 & -s_4 \\ s_2 & s_1 & s_4 & -s_3 \\ s_3 & -s_4 & s_1 & s_2 \end{bmatrix}, \quad (9.16)$$

$$\mathbf{X}_5 = \begin{bmatrix} s_1 & -s_2 & -s_3 & -s_4 & -s_5 & -s_6 & -s_7 & -s_8 \\ s_2 & s_1 & -s_4 & s_3 & -s_6 & s_5 & s_8 & -s_7 \\ s_3 & s_4 & s_1 & -s_2 & -s_7 & -s_8 & s_5 & s_6 \\ s_4 & -s_3 & s_2 & s_1 & -s_8 & s_7 & -s_6 & s_5 \\ s_5 & s_6 & s_7 & s_8 & s_1 & -s_2 & -s_3 & -s_4 \end{bmatrix}, \quad (9.17)$$

$$\mathbf{X}_6 = \begin{bmatrix} s_1 & -s_2 & -s_3 & -s_4 & -s_5 & -s_6 & -s_7 & -s_8 \\ s_2 & s_1 & -s_4 & s_3 & -s_6 & s_5 & s_8 & -s_7 \\ s_3 & s_4 & s_1 & -s_2 & -s_7 & -s_8 & s_5 & s_6 \\ s_4 & -s_3 & s_2 & s_1 & -s_8 & s_7 & -s_6 & s_5 \\ s_5 & s_6 & s_7 & s_8 & s_1 & -s_2 & -s_3 & -s_4 \\ s_6 & -s_5 & s_8 & -s_7 & s_2 & s_1 & s_4 & -s_3 \end{bmatrix}, \quad (9.18)$$

$$\mathbf{X}_7 = \begin{bmatrix} s_1 & -s_2 & -s_3 & -s_4 & -s_5 & -s_6 & -s_7 & -s_8 \\ s_2 & s_1 & -s_4 & s_3 & -s_6 & s_5 & s_8 & -s_7 \\ s_3 & s_4 & s_1 & -s_2 & -s_7 & -s_8 & s_5 & s_6 \\ s_4 & -s_3 & s_2 & s_1 & -s_8 & s_7 & -s_6 & s_5 \\ s_5 & s_6 & s_7 & s_8 & s_1 & -s_2 & -s_3 & -s_4 \\ s_6 & -s_5 & s_8 & -s_7 & s_2 & s_1 & s_4 & -s_3 \\ s_7 & -s_8 & -s_5 & s_6 & s_3 & -s_4 & s_1 & s_2 \end{bmatrix}. \quad (9.19)$$

So sánh (9.16) với (9.12), (9.17)–(9.19) với (9.13) chúng ta có thể nhận thấy rằng bộ mã  $\mathbf{X}_3$  được xây dựng trên cơ sở bộ mã vuông  $\mathbf{X}_4$  bằng cách lấy 3 hàng đầu tiên. Tương tự, các bộ mã  $\mathbf{X}_5$ ,  $\mathbf{X}_6$ ,  $\mathbf{X}_7$  tương ứng được xây dựng từ 5, 6, và 7 hàng đầu tiên của bộ mã vuông  $\mathbf{X}_8$ . Thực tế thì chúng ta có thể xây dựng được các bộ mã cho các hệ thống với  $N'$  ăng-ten phát từ bộ mã vuông  $\mathbf{X}_N$  bằng cách lấy  $N'$  hàng bất kỳ từ  $\mathbf{X}_N$  [5]. Cũng giống như các bộ mã STBC trực giao vuông các bộ mã không vuông này sử dụng  $N_t$  khe thời gian để truyền đi  $N_s$  ký hiệu phát. Vì vậy, các bộ mã này cũng có tốc độ toàn phần, tức là  $R = 1$ , đồng thời cũng cho phép thu được độ phân tập toàn phần  $N$ .

## B. Mã O-STBC cho tập tín hiệu phức

Các tập tín hiệu phức được sử dụng rất phổ biến trong thông tin vô tuyến số như  $M$ -QAM hay  $M$ -PSK. Để đạt được độ phân tập toàn phần, các bộ mã O-STBC cho các hệ thống  $N$  ăng-ten phát sử dụng tập tín hiệu phức cũng cần phải thoả mãn điều kiện trực giao (9.8)

$$\mathbf{X}_N \mathbf{X}_N^H = C \sum_{k=1}^{N_s} |s_k|^2 \mathbf{I}_N. \quad (9.20)$$

Bộ mã Alamouti STBC với ma trận phát (9.5)

$$\mathbf{X} = \begin{bmatrix} s_1 & s_2^* \\ s_2 & -s_1^* \end{bmatrix} \quad (9.21)$$

là một ví dụ điển hình về bộ mã phức cho 2 ăng-ten phát. Đây cũng là bộ mã O-STBC phức duy nhất có thể đạt được cả tốc độ và độ phân tập toàn phần [4]. Với các hệ thống có nhiều hơn 2 ăng-ten phát thì một cách tổng quát chúng ta có thể xây dựng được các bộ mã phức cho bất kỳ số ăng-ten  $N$  nào với tốc độ  $R = 1/2$  [4]. Một số

ví dụ về các bộ mã phức tốc độ 1/2 cho  $N = 3, 4$  là

$$\mathbf{X}_3 = \begin{bmatrix} s_1 & -s_2 & -s_3 & -s_4 & s_1^* & -s_2^* & -s_3^* & -s_4^* \\ s_2 & s_1 & s_4 & -s_3 & s_2^* & s_1^* & s_4^* & -s_3^* \\ s_3 & -s_4 & s_1 & s_2 & s_3^* & -s_4^* & s_1^* & s_2^* \end{bmatrix}, \quad (9.22)$$

$$\mathbf{X}_4 = \begin{bmatrix} s_1 & -s_2 & -s_3 & -s_4 & s_1^* & -s_2^* & -s_3^* & -s_4^* \\ s_2 & s_1 & s_4 & -s_3 & s_2^* & s_1^* & s_4^* & -s_3^* \\ s_3 & -s_4 & s_1 & s_2 & s_3^* & -s_4^* & s_1^* & s_2^* \\ s_4 & s_3 & -s_2 & s_1 & s_4^* & s_3^* & -s_2^* & s_1^* \end{bmatrix}. \quad (9.23)$$

Một cách tổng quát, chúng ta có thể xây dựng các bộ mã O-STBC phức tốc độ  $R = 1/2$  bằng cách xây dựng một bộ mã O-STBC phức có cấu trúc tương tự như một bộ mã O-STBC thực có tốc độ toàn phần rồi nối với ma trận liên hợp phức của nó.

Với tốc độ lớn hơn 1/2 có một số mã STBC trực giao phức với tốc độ  $R = 3/4$  cho các hệ thống 3 và 4 ăng-ten phát được xây dựng trên cơ sở lý thuyết thiết kế “thoả hiệp” (admissible design) như sau [4]:

$$\mathbf{X}_3^{3/4} = \begin{bmatrix} s_1 & -s_2^* & \frac{s_3^*}{\sqrt{2}} & \frac{s_3^*}{\sqrt{2}} \\ s_2 & s_1^* & \frac{s_3^*}{\sqrt{2}} & \frac{-s_3^*}{\sqrt{2}} \\ \frac{s_3}{\sqrt{2}} & \frac{s_3}{\sqrt{2}} & \frac{-s_1 - s_1^* + s_2 - s_2^*}{2} & \frac{s_1 - s_1^* + s_2 + s_2^*}{2} \end{bmatrix}, \quad (9.24)$$

$$\mathbf{X}_4^{3/4} = \begin{bmatrix} s_1 & -s_2^* & \frac{s_3^*}{\sqrt{2}} & \frac{s_3^*}{\sqrt{2}} \\ s_2 & s_1^* & \frac{s_3^*}{\sqrt{2}} & \frac{-s_3^*}{\sqrt{2}} \\ \frac{s_3}{\sqrt{2}} & \frac{s_3}{\sqrt{2}} & \frac{-s_1 - s_1^* + s_2 - s_2^*}{2} & \frac{s_1 - s_1^* + s_2 + s_2^*}{2} \\ \frac{s_3}{\sqrt{2}} & \frac{-s_3}{\sqrt{2}} & \frac{s_1 - s_1^* - s_2 - s_2^*}{2} & \frac{-s_1 - s_1^* - s_2 + s_2^*}{2} \end{bmatrix}. \quad (9.25)$$

Một số mã có tốc độ 3/4 khác dùng cho các hệ thống 3 và 4

ăng-ten phát được cho bởi các biểu thức sau [7]-[5]:

$$\mathbf{X}_3 = \begin{bmatrix} s_1 & 0 & -s_2^* & s_3^* \\ 0 & s_1 & -s_3 & -s_2 \\ s_2 & s_3^* & s_1^* & 0 \end{bmatrix}, \quad (9.26)$$

$$\mathbf{X}_4 = \begin{bmatrix} s_1 & 0 & -s_2^* & s_3^* \\ 0 & s_1 & -s_3 & -s_2 \\ s_2 & s_3^* & s_1^* & 0 \\ -s_3 & s_2^* & 0 & s_1^* \end{bmatrix}. \quad (9.27)$$

Ta nhận thấy rằng các mã này đơn giản hơn các mã trong (9.24) và (9.25) ở chỗ nó không đòi hỏi phải thực hiện các phép cộng và nhân.

### 9.2.2 Các thuật toán giải mã tối ưu cho mã O-STBC

Trong mục này trước hết chúng ta sẽ nghiên cứu các thuật toán giải mã tối ưu cho mã O-STBC trong trường hợp máy thu biết chính xác kênh truyền (perfect channel state information). Tiếp đến chúng ta xem xét một kỹ thuật ước lượng kênh có thể được áp dụng trong thực tế khi kênh truyền biến đổi chậm (slow fading), đó là kỹ thuật sai số bình phương tối thiểu (LSE: Least Square Error). Cuối cùng là các kỹ thuật giải mã kết hợp với ước lượng kênh (cyclic ML) trong trường hợp kênh truyền biến đổi nhanh (fast fading).

#### A. Trường hợp máy thu biết chính xác kênh truyền

Các thuật toán giải mã tối ưu cho mã Alamouti và các mã O-STBC thực và phức khác khi máy thu biết chính xác kênh truyền đã được trình bày trong [1], [3], và [6]. Tuy nhiên, các thuật toán này chỉ được trình bày cho từng mã O-STBC riêng biệt. Không có công thức tổng quát cho tất cả các mã. Bạn đọc có thể tham khảo ở phụ lục cuối chương để hiểu thêm về các thuật toán giải mã tối ưu này. Ở đây chúng tôi trình bày dạng tổng quát cho thuật toán giải mã tối ưu tất cả các mã O-STBC được đề xuất trong [5] và [8].

*Giải mã O-STBC cho tập tín hiệu thực:* Khi tín hiệu  $s_k$  là tín hiệu thực, các ma trận mã O-STBC có thể được biểu diễn như sau [8]:

$$\mathbf{X} = \sum_{k=1}^{N_s} s_k \mathbf{A}_k, \quad (9.28)$$

trong đó  $\mathbf{A}_k$  có kích thước  $N \times N_t$  là các ma trận phân tán thỏa mãn các điều kiện sau:

$$\mathbf{A}_k \mathbf{A}_k^T = \mathbf{I}_N, \quad (9.29)$$

$$\mathbf{A}_k \mathbf{A}_n^T = -\mathbf{A}_n \mathbf{A}_k^T, \quad k \neq n \quad (9.30)$$

với  $k = 1, 2, \dots, N_s$ ,  $n = 1, 2, \dots, N_s$ .

Giả thiết kênh pha-định chậm, tức là các hệ số kênh truyền  $h_{m,n}$  không đổi trong khoảng  $N_t$  khe thời gian phát (một ma trận mã), ta có

$$h_{m,n}(t) = h_{m,n}, \quad t = 1, 2, \dots, N_t \quad (9.31)$$

và khối tín hiệu thu được tương ứng với từ mã phát đi được cho bởi biểu thức:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z}, \quad (9.32)$$

trong đó  $\mathbf{H} \in \mathbb{C}^{M \times N}$  là kênh truyền MIMO,  $\mathbf{Y} \in \mathbb{C}^{M \times N_s}$  là ma trận tín hiệu thu;  $\mathbf{Z} \in \mathbb{C}^{M \times N_s}$  là tạp âm cộng Gauss được giả thiết là tạp âm trắng cả trong miền không gian và miền thời gian với giá trị trung bình bằng 0 và phương sai  $\sigma^2$ .

Do  $\mathbf{X}$  chỉ bao gồm các giá trị thực, ta có thể viết lại (9.32) như sau:

$$\tilde{\mathbf{Y}} = \tilde{\mathbf{H}}\mathbf{X} + \tilde{\mathbf{Z}} \quad (9.33)$$

với  $\tilde{\mathbf{Y}} = \begin{bmatrix} \Re\{\mathbf{Y}\} \\ \Im\{\mathbf{Y}\} \end{bmatrix}$ ,  $\tilde{\mathbf{H}} = \begin{bmatrix} \Re\{\mathbf{H}\} \\ \Im\{\mathbf{H}\} \end{bmatrix}$ , và  $\tilde{\mathbf{Z}} = \begin{bmatrix} \Re\{\mathbf{Z}\} \\ \Im\{\mathbf{Z}\} \end{bmatrix}$ .

Khi máy thu biết chính xác kênh truyền  $\mathbf{H}$  hay  $\tilde{\mathbf{H}}$ , nó thực hiện giải mã tối ưu véc-tơ thông tin  $s$  như sau:

$$\bar{s} = \arg \min_{s \in \mathcal{A}} \|\tilde{\mathbf{Y}} - \tilde{\mathbf{H}}\mathbf{X}\|_F^2 \quad (9.34)$$

trong đó  $\|A\|_F$  ký hiệu độ lớn Frobenius (Frobenius norm) của ma trận  $A$ .

Dựa vào tính chất trực giao của ma trận  $X$ , ta có thể biến đổi đổi số trong vế phải của (9.34) như sau:

$$\begin{aligned}
 \|\tilde{Y} - \tilde{H}X\|_F^2 &= \|\tilde{Y}\|_F^2 - 2\{\text{trace}(\tilde{Y}^T \tilde{H}X)\} + \|\tilde{H}X\|_F^2 \\
 &= \|\tilde{Y}\|_F^2 - 2 \sum_{k=1}^{N_s} \text{trace}(\tilde{Y}^T \tilde{H}A_k)s_k + \text{trace}(\|s\|_2^2 \tilde{H}I_N \tilde{H}^T) \\
 &= \sum_{k=1}^{N_s} [-2\text{trace}(\tilde{Y}^T \tilde{H}A_k)s_k + |s_k|^2 \|\tilde{H}\|_F^2] + \text{const.} \\
 &= \|\tilde{H}\|_F^2 \sum_{k=1}^{N_s} (s_k - \tilde{s}_k)^2 + \text{const.}
 \end{aligned} \tag{9.35}$$

trong đó

$$\tilde{s}_k = \frac{\text{trace}(\tilde{Y}^T \tilde{H}A_k)}{\|\tilde{H}\|_F^2} \tag{9.36}$$

là biến thông kê quyết định (decision statistic) tương ứng với ký hiệu  $s_k$ ,  $\text{trace}(A)$  ký hiệu vết của ma trận  $A$  được tính bằng tổng tất cả các phần tử nằm trên đường chéo chính của  $A$ .

Từ công thức (9.35), ta có thể viết lại luật quyết định tối ưu (9.34) như sau:

$$\bar{s} = \arg \min_{s \in \mathcal{A}} \sum_{k=1}^{N_s} (s_k - \tilde{s}_k)^2. \tag{9.37}$$

Biểu thức (9.37) chứng tỏ rằng metric tối ưu được tách thành tổng của  $N_s$  thành phần, trong đó mỗi thành phần phụ thuộc duy nhất vào một ký hiệu thực  $s_k$ . Do đó, ký hiệu  $s_k$  được giải mã độc lập với ký hiệu  $s_n$ ,  $k \neq n$ , theo luật sau:

$$\bar{s}_k = \arg \min_{s_k \in \mathcal{A}} (s_k - \tilde{s}_k)^2. \tag{9.38}$$

*Giải mã O-STBC cho tập tín hiệu phức:* Khi tín hiệu  $s_k$  là tín hiệu phức, các ma trận mã O-STBC có thể được biểu diễn như trong công thức (9.2), với các ma trận phân tán thỏa mãn các điều kiện sau

[5]:

$$\mathbf{A}_k^H \mathbf{A}_k = C \mathbf{I}_N, \quad \mathbf{B}_k^H \mathbf{B}_k = C \mathbf{I}_N, \text{ với } C \text{ là hằng số bất kỳ} \quad (9.39)$$

$$\mathbf{A}_k^H \mathbf{A}_n = -\mathbf{A}_n^H \mathbf{A}_k, \quad \mathbf{B}_k^H \mathbf{B}_n = -\mathbf{B}_n^H \mathbf{B}_k, \quad k \neq n \quad (9.40)$$

$$\mathbf{A}_n^H \mathbf{B}_k = \mathbf{B}_k^H \mathbf{A}_n \quad (9.41)$$

trong đó  $k = 1, 2, \dots, K$ ,  $n = 1, 2, \dots, K$ .

Tương tự như trong trường hợp tín hiệu thực, nếu chúng ta giả thiết kênh truyền  $\mathbf{H}$  không thay đổi trong thời gian truyền một từ mã, ta có thể viết ma trận tín hiệu thu được như sau trong (9.32).

Khi máy thu biết chính xác kênh truyền  $\mathbf{H}$ , nó thực hiện giải mã tối ưu véc-tơ thông tin  $s$  như sau:

$$\bar{s} = \arg \min_{s \in \mathcal{A}} \|\mathbf{Y} - \mathbf{H}\mathbf{X}\|_F^2. \quad (9.42)$$

Khi các ma trận phân tán  $\mathbf{A}_k$  and  $\mathbf{B}_k$  thỏa mãn các điều kiện (9.39)-(9.41), thì ma trận matrix  $\mathbf{X}$  thỏa mãn điều kiện (9.8). Do đó, ta có thể biến đổi đổi số trong vế phải của (9.42) như trong biểu thức sau:

$$\begin{aligned} \|\mathbf{Y} - \mathbf{H}\mathbf{X}\|_F^2 &= \|\mathbf{Y}\|_F^2 - 2\Re\{\text{trace}(\mathbf{Y}^H \mathbf{H}\mathbf{X})\} + \|\mathbf{H}\mathbf{X}\|_F^2 \\ &= \|\mathbf{Y}\|_F^2 - 2 \sum_{k=1}^{N_s} \Re\{\text{trace}(\mathbf{Y}^H \mathbf{H}\mathbf{A}_k)\} \Re\{s_k\} \\ &\quad + 2 \sum_{k=1}^{N_s} \Im\{\text{trace}(\mathbf{Y}^H \mathbf{H}\mathbf{B}_k)\} \Im\{s_k\} + \text{trace}(C \|s\|_2^2 \mathbf{H} \mathbf{I}_N \mathbf{H}^H) \\ &= \sum_{k=1}^{N_s} \left[ -2\Re\{\text{trace}(\mathbf{Y}^H \mathbf{H}\mathbf{A}_k)\} \Re\{s_k\} \right. \\ &\quad \left. + 2\Im\{\text{trace}(\mathbf{Y}^H \mathbf{H}\mathbf{B}_k)\} \Im\{s_k\} + C|s_k|^2 \|\mathbf{H}\|_F^2 \right] + \text{const.} \\ &= C \|\mathbf{H}\|_F^2 \sum_{k=1}^{N_s} |s_k - \tilde{s}_k|^2 + \text{const.} \end{aligned} \quad (9.43)$$

trong đó

$$\tilde{s}_k = \frac{\Re\{\text{trace}(\mathbf{Y}^H \mathbf{H}\mathbf{A}_k)\} - j\Im\{\text{trace}(\mathbf{Y}^H \mathbf{H}\mathbf{B}_k)\}}{C \|\mathbf{H}\|_F^2} \quad (9.44)$$

là biến ngẫu nhiên quyết định (decision statistic) tương ứng với ký hiệu  $s_k$ .

Từ công thức (9.43), ta có thể viết lại luật quyết định tối ưu (9.42) như sau:

$$\bar{s} = \arg \min_{s \in \mathcal{A}} \sum_{k=1}^{N_s} |s_k - \tilde{s}_k|^2. \quad (9.45)$$

Một lần nữa, từ biểu thức (9.45) ta thấy rằng chỉ số đo (metric) tối ưu được tách thành tổng của  $N_s$  thành phần, trong đó mỗi thành phần phụ thuộc duy nhất vào một ký hiệu phức  $s_k$ . Do đó, ký hiệu  $s_k$  được giải mã độc lập với ký hiệu  $s_n$ ,  $k \neq n$ , theo luật sau:

$$\bar{s}_k = \arg \min_{s_k \in \mathcal{A}} |s_k - \tilde{s}_k|^2. \quad (9.46)$$

Chú ý, đối với tín hiệu QAM, chúng ta có thể thực hiện tách một cách độc lập phần thực và phần ảo của mỗi ký hiệu phức  $s_k$  như sau:

$$\bar{s}_k^I = \arg \min_{s_k^I \in \mathcal{N}} |s_k^I - \tilde{s}_k^I|^2 \quad (9.47)$$

$$\bar{s}_k^Q = \arg \min_{s_k^Q \in \mathcal{N}} |s_k^Q - \tilde{s}_k^Q|^2 \quad (9.48)$$

trong đó,  $\mathcal{N}$  là tập các số nguyên để xây dựng nên tín hiệu  $\mathcal{M}$ -QAM,

$$\tilde{s}_k^I = \frac{\Re\{\text{trace}(\mathbf{Y}^H \mathbf{H} \mathbf{A}_k)\}}{C \|\mathbf{H}\|_F^2}, \quad (9.49)$$

$$\tilde{s}_k^Q = \frac{-\Im\{\text{trace}(\mathbf{Y}^H \mathbf{H} \mathbf{B}_k)\}}{C \|\mathbf{H}\|_F^2}. \quad (9.50)$$

Công thức (9.46) có thể được sử dụng để giải mã cho các mã O-STBC với các ký hiệu được điều chế PSK. Tuy nhiên, do các tín hiệu PSK là các tín hiệu có biên độ không đổi (constant modulus), nên để khôi phục lại tín hiệu  $s_k$  ta chỉ cần thành phần pha của  $\tilde{s}_k$ . Biên độ của  $\tilde{s}_k$  không tác động tới sự khôi phục  $s_k$ . Vì vậy, thành phần mấu số trong (9.44) có thể được bỏ qua, và ký hiệu PSK,  $s_k$ , được giải mã tối ưu dựa trên công thức:

$$\bar{s}_k = \arg \min_{s_k \in \mathcal{A}} |s_k - \tilde{s}_k^p|^2 \quad (9.51)$$

với

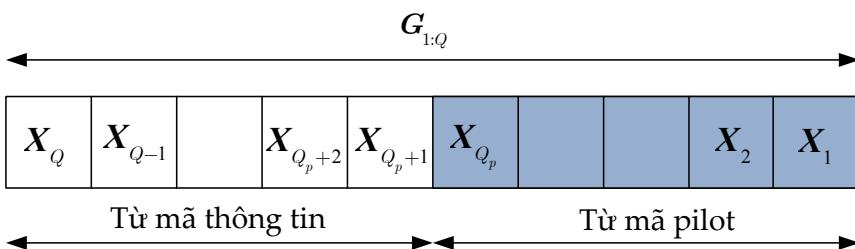
$$\tilde{s}_k^p = \Re\{\text{trace}(\mathbf{Y}^H \mathbf{H} \mathbf{A}_k^T)\} - j\Im\{\text{trace}(\mathbf{Y}^H \mathbf{H} \mathbf{B}_k^T)\} \quad (9.52)$$

là biến thông kê quyết định tương ứng với ký hiệu  $s_k$ .

Rõ ràng là tính chất trực giao của các mã O-STBC đã biến bài toán giải mã tối ưu phức tạp trong (9.34) và (9.42) thành các bài toán giải mã tối ưu cho từng ký hiệu riêng biệt như biểu thị trong các công thức (9.38), (9.46) và (9.51), do đó đơn giản hóa một cách đáng kể quá trình giải mã.

## B. Ước lượng kênh truyền bằng thuật toán LSE

Giả thiết máy thu biết chính xác kênh truyền là một giả thiết lý tưởng. Trong thực tế để có được thông tin về kênh truyền, máy thu sẽ phải thực hiện việc ước lượng kênh (channel estimation). Tuy nhiên, do tác động của tạp âm, kênh thu được do ước lượng không phải là kênh truyền chính xác, dẫn tới sự suy giảm phẩm chất BER của hệ thống.



**Hình 9.2:** Cấu trúc của một khung thông tin gồm các từ mã O-STBC.

Với giả thiết kênh truyền là cận tĩnh, tức là nó không thay đổi trên toàn bộ một khung tín hiệu  $G_{1:Q}$  bao gồm  $Q$  ma trận mã liên tiếp, để thực hiện việc ước lượng kênh, ta sử dụng  $Q_p$  ma trận mã đầu tiên khung.  $Q_p$  ma trận mã này được cả máy phát và máy thu biết rõ và được gọi là các ma trận mã pilot (xem Hình 9.2). Với các từ mã pilot, ta có thể ước lượng được kênh truyền một cách khá dễ dàng nhờ các thuật toán sai số bình phương tối thiểu (LSE) hoặc

sai số bình phương trung bình cực tiểu (MMSE: Minimum Mean Square Error). Tuy phương pháp MMSE cho kết quả ước lượng chính xác hơn nhưng chúng đòi hỏi ta phải biết được công suất của tạp âm tại máy thu, do đó phương pháp này có độ phức tạp lớn hơn phương pháp LSE. Dưới đây, chúng tôi sẽ trình bày phương pháp LSE một cách cụ thể cho trường hợp tín hiệu phát đi là các tín hiệu phức.

Dựa vào khung tín hiệu như trên Hình 9.2, ta có thể viết tín hiệu thu được ứng với toàn bộ khung tín hiệu  $\mathbf{G}_{1:Q}$  như sau:

$$\mathbf{Y} = \mathbf{H}\mathbf{G}_{1:Q} + \mathbf{Z} \quad (9.53)$$

trong đó  $\mathbf{G}_{1:Q} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_Q]$  là khung tín hiệu phát đi;  $\mathbf{X}_n$ ,  $n = 1, \dots, Q$ , là từ mã thứ  $n$ ;  $\mathbf{H}$  là ma trận kênh MIMO với kích thước  $M \times N$ ;  $\mathbf{Y} = [\mathbf{Y}_p, \mathbf{Y}_d] = [\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_Q]$  là khung tín hiệu thu được có kích thước  $M \times N_t Q$ ;  $\mathbf{Y}_p$  là khung tín hiệu con ứng với toàn bộ  $Q_p$  từ mã pilot còn  $\mathbf{Y}_d$  là khung tín hiệu con ứng với  $(Q - Q_p)$  từ mã thông tin; và  $\mathbf{Z} = [\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_Q]$  là ma trận có kích thước  $M \times N_t Q$  chứa các mẫu tạp âm tại các ăng-ten thu.

Theo phương pháp LSE, kênh truyền  $\mathbf{H}$  được ước lượng dựa trên biểu thức sau:

$$\hat{\mathbf{H}}_{LSE} = \arg \min_{\mathbf{H} \in \mathbb{C}^{M \times N}} \|\mathbf{Y}_p - \mathbf{H}\mathbf{G}_{1:Q_p}\|_F^2. \quad (9.54)$$

Nghiệm của phương trình trên chính là ước lượng LSE của kênh truyền  $\mathbf{H}$  được cho bởi biểu thức sau [9, 10]:

$$\begin{aligned} \hat{\mathbf{H}}_{LSE} &= \mathbf{Y}_p \mathbf{G}_{1:Q_p}^H (\mathbf{G}_{1:Q_p} \mathbf{G}_{1:Q_p}^H)^{-1} \\ &= \sum_{n=1}^{Q_p} \mathbf{Y}_n \mathbf{X}_n^H \left[ \sum_{n=1}^{Q_p} \mathbf{X}_n \mathbf{X}_n^H \right]^{-1}. \end{aligned} \quad (9.55)$$

Do các ma trận  $\mathbf{X}_n$  thỏa mãn điều kiện (9.8), ta có:

$$\begin{aligned} \hat{\mathbf{H}}_{LSE} &= \sum_{n=1}^{Q_p} \mathbf{Y}_n \mathbf{X}_n^H \left[ \sum_{n=1}^{Q_p} C \|\mathbf{s}_n\|_2^2 \mathbf{I}_N \right]^{-1} \\ &= \frac{\sum_{n=1}^{Q_p} \mathbf{Y}_n \mathbf{X}_n^H}{C \sum_{n=1}^{Q_p} \|\mathbf{s}_n\|_2^2}. \end{aligned} \quad (9.56)$$

Nếu các ký hiệu phát đi được chọn từ biểu đồ phân phối điểm tín hiệu (chòm sao tín hiệu)  $M$ -PSK, tức là  $\mathcal{A} = \{e^{jm\frac{2\pi}{M}}\}$  với  $m = 0, 1, \dots, M-1$ , thì  $\|\mathbf{s}_1\|_2^2 = \|\mathbf{s}_2\|_2^2 = \dots = \|\mathbf{s}_{Q_p}\|_2^2 = N_s$ . Lúc này, biểu thức (9.56) được rút gọn thành:

$$\hat{\mathbf{H}}_{LSE} = \frac{1}{CN_s Q_p} \sum_{n=1}^{Q_p} \mathbf{Y}_n \mathbf{X}_n^H. \quad (9.57)$$

Nếu ta coi  $\hat{\mathbf{H}}_{LSE}$  là chính xác, ta có thể áp dụng chúng vào các biểu thức (9.46) và (9.51) để khôi phục lại các tín hiệu đã được phát đi.

### Thuật toán giải mã luân phiên

Như đã trình bày ở trên, để giải mã được tín hiệu phát đi, máy thu phải thực hiện ước lượng kênh thông qua các từ mã pilot. Rõ ràng là việc phát đi các từ mã pilot làm giảm tốc độ truyền tin. Do đó, số lượng các từ mã pilot phải càng nhỏ càng tốt, đặc biệt là trong các kênh pha-định nhanh, nhằm tránh cho tốc độ truyền tin giảm ở một mức quá lớn. Ví dụ, trong kênh pha-định chậm, ta có thể tạo một khung có độ dài  $Q = 100$  với  $Q_p = 10$  từ mã pilot. Như vậy tốc độ thông tin của hệ thống sẽ bị giảm đi  $1/10$  so với khi không truyền pilot. Trong trường hợp kênh bị pha-định nhanh và ta chỉ có thể tạo được một khung có độ dài tối đa là  $Q = 30$ . Nếu ta vẫn giữ nguyên  $Q_p = 10$  thì tốc độ truyền tin bị giảm tới  $1/3$  so với khi không phát tín hiệu pilot.

Tuy nhiên, khi số lượng các từ mã pilot nhỏ, sai số trong việc ước lượng kênh lớn, dẫn tới sự suy giảm chất lượng truyền tin (tăng BER). Để khắc phục nhược điểm này, ta có thể áp dụng thuật toán giải mã luân phiên (cyclic decoder) [8]. Thuật toán này được mô tả ngắn gọn như sau. Trước hết ta sử dụng  $Q_p$  từ mã pilot để ước lượng kênh theo thuật toán LSE. Rồi sử dụng kênh ước lượng  $\hat{\mathbf{H}}_{LSE}$  để giải mã toàn bộ  $(Q - Q_p)$  từ mã thông tin còn lại. Sau đó ta coi tất cả  $(Q - Q_p)$  từ mã thông tin được giải mã đúng rồi sử dụng chúng cùng với  $Q_p$  từ mã pilot để ước lượng lại kênh truyền. Sau đó ta

sử dụng kênh truyền vừa được ước lượng để giải mã lại các từ mã thông tin. Cứ như thế ta sẽ luân phiên ước lượng kênh truyền và giải mã các từ mã thông tin cho đến khi đạt được độ chính xác cần thiết hoặc cho đến khi đạt đến một số lần giải mã cho trước. Thuật toán này được tóm tắt như sau:

### *Thuật toán giải mã luân phiên*

- **Khởi tạo:** Tính giá trị ước lượng ban đầu của  $\hat{\mathbf{H}}$  theo công thức (9.56) hoặc (9.57)

- **Lặp:**

1. Sử dụng ước lượng kênh  $\hat{\mathbf{H}}_{LSE}$  mới nhất để giải mã các từ mã phát đi theo công thức (9.46) hoặc (9.51). Sau bước này ta thu được các véc-tơ tín hiệu  $\bar{s}_n$  hay các từ mã  $\bar{X}_n$ ,  $n = Q_p + 1, \dots, Q$ .
2. Ước lượng lại kênh truyền theo biểu thức sau:

$$\hat{\mathbf{H}}_{LSE} = \frac{\sum_{n=1}^{Q_p} \mathbf{Y}_n \mathbf{X}_n^H + \sum_{n=Q_p+1}^Q \mathbf{Y}_n \bar{\mathbf{X}}_n^H}{C \sum_{n=1}^{Q_p} \|\mathbf{s}_n\|_2^2 + C \sum_{n=Q_p+1}^Q \|\bar{\mathbf{s}}_n\|_2^2}$$

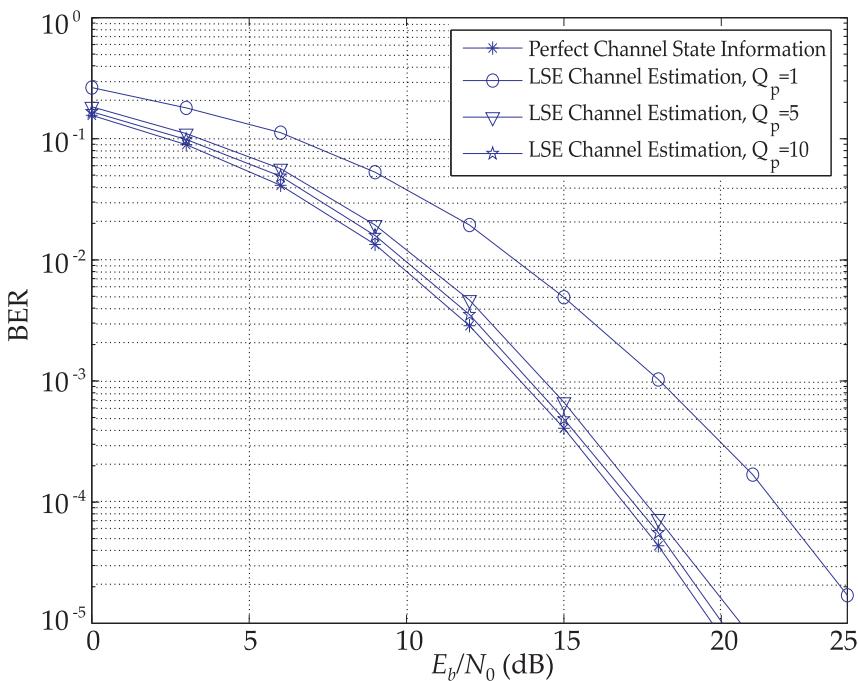
hoặc

$$\hat{\mathbf{H}}_{LSE} = \frac{1}{cN_s Q} \left( \sum_{n=1}^{Q_p} \mathbf{Y}_n \mathbf{X}_n^H + \sum_{n=Q_p+1}^Q \mathbf{Y}_n \bar{\mathbf{X}}_n^H \right) \text{cho PSK}$$

3. Lặp lại các bước 1 và 2 cho đến khi thuật toán hội tụ hay đến khi thuật toán đã tiến hành được một số vòng lặp cho trước.

### **Phẩm chất BER của mã O-STBC**

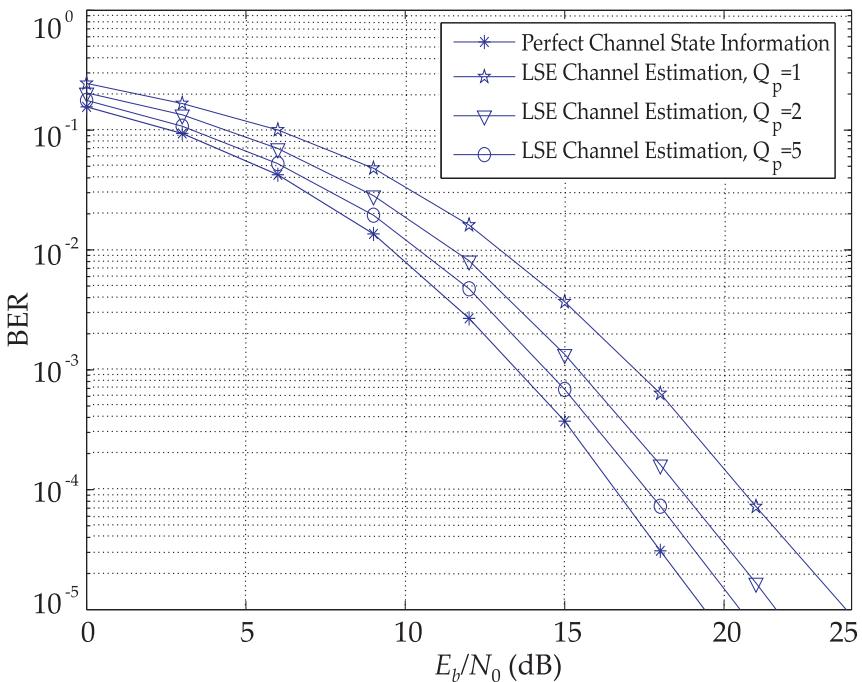
Hình 9.3 trình bày phẩm chất lỗi bit của mã OSTBC thực, với ma trận mã  $\mathbf{X}_4$  trong biểu thức (9.12), trong hệ thống phân tập phát với 4 ăng-ten phát, 1 ăng-ten thu sử dụng kỹ thuật điều chế 4PAM. Hình vẽ minh họa các đường BER khi kênh truyền được biết chính



Hình 9.3: Phẩm chất BER của mã OSTBC thực,  $N = 4, M = 1$ ; Điều chế 4PAM.

xác và khi kênh truyền được ước lượng dựa vào kỹ thuật LSE như trình bày ở phần trên cho các số lượng từ mã pilot  $Q_p = 1, 5$  và  $10$ . Trong chương trình mô phỏng, ta giả định kênh truyền là không đổi trong khoảng thời gian  $(400 + 4Q_p)$  chu kỳ tín hiệu (tương đương với  $100 + Q_p$  ma trận mã). Từ hình vẽ ta thấy rõ ràng rằng khi số lượng từ mã pilot tăng lên thì kênh truyền được ước lượng càng chính xác, và do đó đường BER của hệ thống càng gần với đường BER lý tưởng (tức là khi máy thu biết chính xác kênh truyền). Cụ thể, trong vùng giá trị  $\text{BER} < 10^{-3}$ , thì sai lệch giữa đường BER lý tưởng và đường BER của  $Q_p = 1$  là khoảng  $4\text{dB}$ ; sai lệch này giảm xuống dưới  $0,5 \text{ dB}$  khi số lượng từ mã pilot tăng lên  $Q_p = 10$ . Hiện nhiên, việc phẩm chất BER của hệ thống được cải thiện phải đánh đổi bằng sự suy giảm tốc độ truyền tin (từ  $100/101$  giảm xuống còn  $100/110$ ).

Tương tự như vậy, Hình 9.4 trình bày phẩm chất lỗi bit của mã OSTBC phức, với ma trận mã  $X_4$  trong biểu thức (9.27), trong hệ thống phân tập phát với 4 ăng-ten phát, 1 ăng-ten thu sử dụng kỹ



**Hình 9.4:** Phẩm chất BER của mã OSTBC phức,  $N = 4, M = 1$ ; Điều chế 16QAM.

thuật điều chế 16QAM. Từ hình vẽ, chúng ta cũng thấy rõ được sự cải thiện về BER khi  $Q_p$  tăng từ 1 lên đến 5.

Các chương trình mô phỏng cho hai hệ thống trên được trình bày ở chương trình COSTBC4Tx.m và ROSTBC4Tx.m trong MATLAB Program 9.2 và 9.1.

### 9.3 Tóm tắt

Trong chương này chúng ta đã nghiên cứu về kỹ thuật mã hóa khôi trực giao không gian-thời gian (OSTBC), các phương pháp khôi phục tín hiệu ở phía thu, cũng như phương pháp ước lượng kênh truyền LSE. Các chương trình mô phỏng mẫu trong chương này cho phép bạn đọc có thể đánh giá được phẩm chất lỗi bit của các hệ thống phân tập phát khi sử dụng mã OSTBC và sự đơn giản của hệ thống trong quá trình giải mã khôi phục tín hiệu.

## Tài liệu tham khảo

- [1] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, 1998.
- [2] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time codes for high data rate wireless communications: performance criterion and code construction," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 744–765, 1998.
- [3] ——, "Space-time block coding for wireless communications: performance results," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 3, pp. 451–460, 1999.
- [4] ——, "Space-time block codes from orthogonal designs," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1456–1467, 1999.
- [5] E. G. Larsson and P. Stoica, *Space-Time Block Coding for Wireless Communications*. Cambridge University Press, 2003.
- [6] B. Vucetic and J. Yuan, *Space-Time Coding*. Wiley & Sons Ltd., 2003.
- [7] G. Ganesan and P. Stoica, "Space-time block codes: A maximum snr approach," *IEEE Transactions on Information Theory*, vol. 47, no. 4, pp. 1650–1656, May 2001.
- [8] P. Stoica and G. Ganesan, "Space-time block codes: trained, blind, and semi-blind detection," *Digital Signal Processing*, vol. 13, pp. 93–135, 2003.
- [9] E. Larsson, P. Stoica, and J. Li, "Orthogonal space-time block codes: maximum likelihood detection for unknown channels and unstructured interferences," *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 362–372, February 2003.
- [10] P. Garg, R. K. Mallic, and H. M. Gupta, "Exact error performance of square orthogonal space-time block coding with channel estimation," *IEEE Transactions on Communications*, vol. 54, no. 3, pp. 430–437, March 2006.

## Phụ lục A: Các thuật toán giải mã tối ưu khác dùng cho mã O-STBC

### Giải mã STBC cho tập tín hiệu thực

Giả thiết kênh pha-định chậm, tức là các hệ số kênh truyền  $h_{m,n}$  không đổi trong khoảng  $N_t$  khe thời gian phát, ta có

$$h_{m,n}(t) = h_{m,n}, \quad t = 1, 2, \dots, N_t. \quad (9.58)$$

Như vậy, tín hiệu thu tại thời điểm  $t$  và ăng-ten thu  $m$  có thể được biểu diễn như sau

$$y_{m,t} = \sum_{n=1}^N h_{m,n} s_{n,t} + z_{m,t} \quad (9.59)$$

Đối với loại STBC có ma trận truyền dẫn vuông như  $X_2$ ,  $X_4$  và  $X_8$ , chúng ta để ý rằng các cột từ thứ 2 đến thứ  $N$  là các hoán vị của cột đầu tiên  $[s_1, s_2, \dots, s_N]^T$ . Đặt  $\wp_t$  là phép hoán vị của các ký hiệu phát (symbol) từ cột thứ nhất đến cột thứ  $t$ . Vị trí hàng của  $s_n$  ở cột  $t$  được biểu diễn bởi  $\wp_t(s_n)$  và dấu của  $s_i$  ở cột  $t$  được ký hiệu là  $\text{sgn}_t(s_n)$ .

Với giả thiết là hệ số kênh truyền  $h_{m,n}$  được biết trước thì máy thu sử dụng phương pháp kết hợp tuyến tính tương tự như phương pháp Alamouti ở các công thức (6.87) và (6.88) để thiết lập thống kê quyết định (decision stastistics) cho tín hiệu  $s_n$  như sau [4]

$$\tilde{s}_n = \sum_{t=1}^N \sum_{m=1}^M \text{sgn}_t(s_n) y_{m,t} h_{m,\wp_t(s_n)}^*. \quad (9.60)$$

Do tính chất đặc biệt của các hàng trong ma trận truyền dẫn, tương tự như phương pháp Alamouti (xem các công thức (6.91) (6.91)), luật tách sóng hợp lẽ tối ưu (ML)

$$\hat{s}_n = \arg \min_{x_{n,t} \in \chi} \left\{ \sum_{t=1}^{N_t} \sum_{m=1}^M \left| y_{m,t} - \sum_{n=1}^N h_{m,n} x_{n,t} \right|^2 \right\}. \quad (9.61)$$

tương đương với

$$\hat{s}_n = \arg \min_{s_n \in \chi} \left\{ \sum_{n=1}^N \left[ |\tilde{s}_n - s_n|^2 + \left( \sum_{n=1}^N \sum_{m=1}^M |h_{m,n}|^2 - 1 \right) |s_n|^2 \right] \right\}. \quad (9.62)$$

Do  $\tilde{s}_n$  chỉ phụ thuộc vào  $s_n$ , metric quyết định kết hợp (joint decision metric) (9.62) được giản ước thành metric quyết định độc lập cho từng ký hiệu phát như sau:

$$\hat{s}_n = \arg \min_{s_n \in \chi} \left\{ |\tilde{s}_n - s_n|^2 + \left( \sum_{n=1}^N \sum_{m=1}^M |h_{m,n}|^2 - 1 \right) |s_n|^2 \right\}. \quad (9.63)$$

Điều này có nghĩa là do tính chất trực giao, thông kê quyết định cho tín hiệu phát mong muốn  $s_n$  hoàn toàn độc lập với các tín hiệu phát  $s_j, \{j = 1, 2, \dots, N, j \neq n\}$  khác.

Đối với mã STBC có ma trận truyền dẫn không vuông ( $N \neq N_t$ ) như  $\mathbf{X}_3, \mathbf{X}_5, \mathbf{X}_6$  và  $\mathbf{X}_7$  thì thông kê quyết định tại máy thu là

$$\tilde{s}_i = \sum_{t \in \psi(i)} \sum_{j=1}^M \text{sgn}_t(s_i) y_{m,t} h_{m,\wp_t(s_i)}^*, \quad (9.64)$$

trong đó  $i = 1, 2, \dots, N_t$ , còn  $\psi(i)$  là tập hợp của các cột của ma trận truyền dẫn có xuất hiện  $s_i$ . Ví dụ, với ma trận truyền dẫn  $\mathbf{X}_3$  như ở công thức (9.16) thì

$$\psi(1) = \{1, 2, 3\} \quad \psi(2) = \{1, 2, 4\} \quad (9.65)$$

$$\psi(3) = \{1, 3, 4\} \quad \psi(4) = \{2, 3, 4\}. \quad (9.66)$$

Tương tự nhu trường hợp với ma trận phát vuông thì metric quyết định cho các từng tín hiệu  $s_i$  là

$$\hat{s}_i = \arg \min_{s_i \in \chi} \left\{ |\tilde{s}_i - s_i|^2 + \left( \sum_{n=1}^N \sum_{m=1}^M |h_{m,n}|^2 - 1 \right) |s_i|^2 \right\}. \quad (9.67)$$

### Giải mã STBC cho tập tín hiệu phức

Để ý rằng bộ mã STBC cho tập tín hiệu phức có ma trận truyền dẫn vuông chỉ tồn tại với mã  $\mathbf{X}_2$  của Alamouti. Phương pháp giải mã cho bộ mã này tương tự như phương pháp giải mã cho các bộ mã

STBC cho tập tín hiệu trực có ma trận truyền dẫn vuông, và đã được trình bày cụ thể ở Chương 6.

Đối với các bộ mã có ma trận truyền dẫn không vuông  $\mathbf{X}_3, \mathbf{X}_4$  có tốc độ  $1/2$  thì cũng tương tự như phương pháp giải mã cho tập tín hiệu thực chúng ta có thống kê quyết định như sau

$$\tilde{s}_i = \sum_{t \in \psi(i)} \sum_{j=1}^M \text{sgn}_t(s_i) y'_{m,t}(i) h'^*_{m,\wp_t(s_i)}, \quad (9.68)$$

trong đó

$$y'_{m,t}(i) = \begin{cases} y_{m,t} & \text{nếu } s_i \text{ nằm ở cột } t \text{ của } \mathbf{X}_N \\ y^*_{m,t} & \text{nếu } s_i^* \text{ nằm ở cột } t \text{ của } \mathbf{X}_N \end{cases} \quad (9.69)$$

còn

$$h'_{m,\wp_t(s_i)} = \begin{cases} h^*_{m,\wp_t(s_i)} & \text{nếu } s_i \text{ nằm ở cột } t \text{ của } \mathbf{X}_N \\ h_{m,\wp_t(s_i)} & \text{nếu } s_i^* \text{ nằm ở cột } t \text{ của } \mathbf{X}_N \end{cases}. \quad (9.70)$$

Ví dụ, đối với bộ mã STBC  $\mathbf{X}_3$  có tốc độ  $1/2$  thì thống kê quyết định

cho các ký hiệu phát được biểu diễn cụ thể như sau [6]

$$\begin{aligned}\tilde{s}_1 &= \sum_{m=1}^M (y_{m,1}h_{m,1}^* + y_{m,2}h_{m,2}^* + y_{m,3}h_{m,3}^* + y_{m,5}^*h_{m,1} + y_{m,6}^*h_{m,2} + y_{m,7}^*h_{m,3}) \\ &= \rho_3 s_1 + \sum_{m=1}^M (z_{m,1}h_{m,1}^* + z_{m,2}h_{m,2}^* + z_{m,3}h_{m,3}^* + z_{m,5}^*h_{m,1} + z_{m,6}^*h_{m,2} + z_{m,7}^*h_{m,3})\end{aligned}\quad (9.71)$$

$$\begin{aligned}\tilde{s}_2 &= \sum_{m=1}^M (y_{m,1}h_{m,2}^* - y_{m,2}h_{m,1}^* + y_{m,4}h_{m,3}^* + y_{m,5}^*h_{m,2} - y_{m,6}^*h_{m,1} + y_{m,8}^*h_{m,3}) \\ &= \rho_3 s_2 + \sum_{m=1}^M (z_{m,1}h_{m,2}^* - z_{m,2}h_{m,1}^* + z_{m,4}h_{m,3}^* + z_{m,5}^*h_{m,2} - z_{m,6}^*h_{m,1} + z_{m,8}^*h_{m,3})\end{aligned}\quad (9.72)$$

$$\begin{aligned}\tilde{s}_3 &= \sum_{m=1}^M (y_{m,1}h_{m,3}^* - y_{m,3}h_{m,1}^* - y_{m,4}h_{m,2}^* + y_{m,5}^*h_{m,3} - y_{m,7}^*h_{m,1} - y_{m,8}^*h_{m,2}) \\ &= \rho_3 s_3 + \sum_{m=1}^M (z_{m,1}h_{m,3}^* - z_{m,3}h_{m,1}^* - z_{m,4}h_{m,2}^* + z_{m,5}^*h_{m,3} - z_{m,7}^*h_{m,1} - z_{m,8}^*h_{m,2})\end{aligned}\quad (9.73)$$

$$\begin{aligned}\tilde{s}_4 &= \sum_{m=1}^M (-y_{m,2}h_{m,3}^* + y_{m,3}h_{m,2}^* - y_{m,4}h_{m,1}^* - y_{m,6}^*h_{m,3} + y_{m,7}^*h_{m,2} - y_{m,8}^*h_{m,1}) \\ &= \rho_3 s_4 + \sum_{m=1}^M (-z_{m,2}h_{m,3}^* + z_{m,3}h_{m,2}^* - z_{m,4}h_{m,1}^* - z_{m,6}^*h_{m,3} + z_{m,7}^*h_{m,2} - z_{m,8}^*h_{m,1})\end{aligned}\quad (9.74)$$

trong đó

$$\rho_3 = 2 \sum_{n=1}^3 \sum_{m=1}^M |h_{m,n}|^2. \quad (9.75)$$

Đối với bộ mã  $\mathbf{X}_3$  có tốc độ  $3/4$ , thống kê quyết định cho các ký hiệu phát là [6]

$$\tilde{s}_1 = \sum_{m=1}^M \left( y_{m,1}h_{m,1}^* + y_{m,2}h_{m,2}^* + \frac{(y_{m,4} - y_{m,3})h_{m,3}^*}{2} - \frac{(y_{m,4} - y_{m,3})^*h_{m,3}}{2} \right) \quad (9.76)$$

$$\tilde{s}_2 = \sum_{m=1}^M \left( y_{m,1}h_{m,2}^* - y_{m,2}h_{m,1}^* + \frac{(y_{m,4} + y_{m,3})h_{m,3}^*}{2} + \frac{(y_{m,4} - y_{m,3})^*h_{m,3}}{2} \right) \quad (9.77)$$

$$\tilde{s}_3 = \sum_{m=1}^M \left( \frac{(y_{m,1} + y_{m,2})h_{m,3}^*}{\sqrt{2}} + \frac{y_{m,3}^*(h_{m,1} + h_{m,2})}{\sqrt{2}} + \frac{y_{m,4}^*(h_{m,1} - h_{m,2})}{\sqrt{2}} \right) \quad (9.78)$$

Tương tự, thống kê quyết định để giải mã bộ mã STBC  $\mathbf{X}_4$  có tốc độ

3/4 là

$$\tilde{s}_1 = \sum_{m=1}^M \left( y_{m,1} h_{m,1}^* + y_{m,2}^* h_{m,2} + \frac{(y_{m,4} - y_{m,3})(h_{m,3}^* - h_{m,4}^*)}{2} - \frac{(y_{m,3} + y_{m,4})^*(h_{m,3} + h_{m,4})}{2} \right) \quad (9.79)$$

$$\tilde{s}_2 = \sum_{m=1}^M \left( y_{m,1} h_{m,2}^* - y_{m,2}^* h_{m,1} + \frac{(y_{m,4} + y_{m,3})(h_{m,3}^* - h_{m,4}^*)}{2} + \frac{(y_{m,4} - y_{m,3})^*(h_{m,3} + h_{m,4})}{2} \right) \quad (9.80)$$

$$\tilde{s}_3 = \sum_{m=1}^M \left( \frac{(y_{m,1} + y_{m,2})h_{m,3}^*}{\sqrt{2}} + \frac{(y_{m,1} - y_{m,2})h_{m,4}^*}{\sqrt{2}} + \frac{y_{m,3}^*(h_{m,1} + h_{m,2})}{\sqrt{2}} + \frac{y_{m,4}^*(h_{m,1} - h_{m,2})}{\sqrt{2}} \right) \quad (9.81)$$

## Phụ lục B: Chương trình MATLAB mẫu

---

### ■ MATLAB Program 9.1 ROSTBC4Tx.m

```
% Program to simulate Real OSTBC in (4,M) MIMO system
% 4PAM Modulation.
% (C)2011 Le Minh Tuan, UVISION JSC.
% Date: 09/08/2011
```

```
clear all;
clc;

N=4; % No of TxAnt
M=1; % No of RxAnt
T=4; % Transmission period per code block
nd=4; % Number of symbols per code block
ml=2; % ml:number of bit per symbol
C=2^ml; % Number of signal points
EbNodB=0:3:21;
EbNo=10.^ (EbNodB/10);
```

```
Qp=10; %Number of pilot codewords
BlkSize=Qp+100*ones(length(EbNo),1);
IterNo=[500 500 1000 3000 6000 10000 20000 40000 30000];
Es = 5; %4PAM Average Symbol Energy

%Integer values for I channel
iv=[-3 -1 3 1]';

sigma=sqrt(N*Es./(2*ml*EbNo)); % noise deviation

%Dispersion matrices
A(:,:,1)=[1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];
A(:,:,2)=[0 -1 0 0; 1 0 0 0; 0 0 0 1; 0 0 -1 0];
A(:,:,3)=[0 0 -1 0; 0 0 0 -1; 1 0 0 0; 0 1 0 0];
A(:,:,4)=[0 0 0 -1; 0 0 1 0; 0 -1 0 0; 1 0 0 0];

%Select channel estimation
ChanEst=1;

% Simulation iteration
for n=1:length(EbNo)
    for i=1:IterNo(n)
        %16QAM modulation
        data=randint(nd,BlkSize(n),C);
        s=iv(data+1);
        XX=zeros(N,T*BlkSize(n),nd);
        for k=1:nd
            XX(:, :, k)=kron(s(k, :), A(:, :, k));
        end
        X=sum(XX, 3);
        %MIMO channel
        H=(randn(M,N)+j*randn(M,N))/sqrt(2);
        %Noise
        z=sigma(n)*(randn(M,T*BlkSize(n))+j*randn(M,T*BlkSize(n)));
        %Received Signal Vector
```

```

r=H*X+z;

%Channel Estimation

if ChanEst==1

    X_p=X(:,1:T*Qp);

    Heq=[real(r(:,1:T*Qp));imag(r(:,1:T*Qp))]*pinv(X_p);

else

    Heq=[real(H);imag(H)];

end

Hnorm=trace(Heq*Heq');

%Signal dection

y=reshape(r,M,T,BlkSize(n));

Y=[real(y);imag(y)]./Hnorm;

d_hat=zeros(nd,BlkSize(n));

s_hat=zeros(1,BlkSize(n));

for k=1:nd

    HA=Heq*A(:,:,k);

    for k2=Qp+1:BlkSize(n)

        s_hat(k2)=sum(diag(Y(:,:,k2)'*HA));

    end

    st=iv(:,ones(1,length(s_hat)))-s_hat(ones(4,1),:);

    [a d_hat(k,:)]=min(st.*st);

end

dataout=d_hat-1;

ne(i)=length(find(dec2bin(dataout(:,Qp+1:end))~=...
dec2bin(data(:,Qp+1:end))));

end

ber(n)=sum(ne)/nd/(BlkSize(n)-Qp)/ml/IterNo(n);

semilogy(EbNodB,ber,'-p')
xlabel('E_b/N_0 (dB)')
ylabel('BER')
hold on

```

```
title(['BER of ',num2str(M),'x',num2str(N), 'Real OSTBC'])  
%!
```

---

**■ MATLAB Program 9.2 COSTBC4Tx.m**

```
% Program to simulate Complex OSTBC in (4,M) MIMO system  
% 16QAM Modulation.  
% (C)2011 Le Minh Tuan, UVISION JSC.  
% Date: 09/08/2011
```

```
clear all;  
clc;  
  
N=4; % No of TxAnt  
M=1; % No of RxAnt  
T=4; % Transmission period per code block  
nd=3; % Number of symbols per code block  
ml=4; % ml:number of bit per symbol  
  
EbNodB=0:3:21;  
EbNo=10.^{(EbNodB/10)};  
Qp=5; % Number of pilot codewords  
BlkSize=Qp+100*ones(length(EbNo),1);  
IterNo=[1500 1500 1500 2000 4000 6000 15000 30000 35000];  
Es = 10; %16QAM Average Symbol Energy  
  
%Integer values for I or Q channel  
iv=[-3 -1 3 1]';  
  
sigma=sqrt(N*Es./(2*ml*EbNo)); % noise deviation  
  
%Dispersion matrices  
A(:,:,1)=[1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];  
A(:,:,2)=[0 0 -1 0; 0 0 0 -1; 1 0 0 0 ; 0 1 0 0];  
A(:,:,3)=[0 0 0 1; 0 0 -1 0; 0 1 0 0; -1 0 0 0];
```

```
B(:,:,1)=[1 0 0 0; 0 1 0 0; 0 0 -1 0; 0 0 0 -1];
B(:,:,2)=[0 0 1 0; 0 0 0 -1; 1 0 0 0 ; 0 -1 0 0];
B(:,:,3)=[0 0 0 -1; 0 0 -1 0; 0 -1 0 0; -1 0 0 0];

%Select channel estimation
ChanEst=1;

% Simulation iteration
for n=1:length(EbNo)
    for i=1:IterNo(n)
        %16QAM modulation
        idata=randint(nd,BlkSize(n),m1);
        qdata=randint(nd,BlkSize(n),m1);
        is=iv(idata+1);
        qs=iv(qdata+1);
        XX=zeros(N,T*BlkSize(n),nd);
        for k=1:nd
            XX(:,:,k)=kron(is(k,:),A(:,:,k))+j*kron(qs(k,:),B(:,:,k));
        end
        X=sum(XX,3);
        %MIMO channel
        H=(randn(M,N)+j*randn(M,N))/sqrt(2);
        %Noise
        z=sigma(n)*(randn(M,T*BlkSize(n))+j*randn(M,T*BlkSize(n)));
        %Received Signal Vector
        r=H*X+z;
        %Channel Estimation
        if ChanEst==1
            X_p=X(:,1:T*Qp);
            H_est=r(:,1:T*Qp)*pinv(X_p);
        else
            H_est=H;
        end
        Hnorm=trace(H_est*H_est');
        %Signal dectection
```

```

y=reshape(r,M,T,BlkSize(n));
y=y./Hnorm;

id_hat=zeros(nd,BlkSize(n));
qd_hat=zeros(nd,BlkSize(n));
is_hat=zeros(1,BlkSize(n));
qs_hat=zeros(1,BlkSize(n));
for k=1:nd
    HA=H_est*A(:,:,k);
    HB=H_est*B(:,:,k);
    for k2=Qp+1:BlkSize(n)
        is_hat(k2)=real(sum(diag(y(:,:,k2)'*HA)));
        qs_hat(k2)=-imag(sum(diag(y(:,:,k2)'*HB)));
    end
    st=iv(:,ones(1,BlkSize(n)))-is_hat(ones(4,1),:);
    [a id_hat(k,:)]=min(st.*st);
    st=iv(:,ones(1,BlkSize(n)))-qs_hat(ones(4,1),:);
    [a qd_hat(k,:)]=min(st.*st);
end
idataout=id_hat-1;
qdataout=qd_hat-1;
ne(i,1)=length(find(dec2bin(idataout(:,Qp+1:end))~=...
    dec2bin(idata(:,Qp+1:end)))); 
ne(i,2)=length(find(dec2bin(qdataout(:,Qp+1:end))~=...
    dec2bin(qdata(:,Qp+1:end)))); 
end
ber(n)=sum(sum(ne))/nd/(BlkSize(n)-Qp)/ml/IterNo(n);
end

semilogy(EbNodB,ber,'-v')
xlabel('E_b/N_0 (dB)')
ylabel('BER')
hold on
title(['BER of ',num2str(M),'x',num2str(N), 'Complex OSTBC'])
%!

```





# Giới thiệu tác giả

**PGS.TS Trần Xuân Nam** sinh ngày 8/9/1971 tại Hà Giang, Hà Trung, Thanh Hóa. Tốt nghiệp Kỹ sư Vô tuyến Điện tử, chuyên ngành Thông tin, tại Học viện Kỹ thuật Quân sự năm 1993, Thạc sỹ Kỹ thuật Viễn thông tại Trường Đại học Kỹ thuật Sydney (UTS) của Australia năm 1998, Tiến sỹ Kỹ thuật Điện tử tại Trường Đại học Điện-Thông tin (UEC), Nhật Bản năm 2003. Từ 10/2003 đến tháng 3/2006 là giáo viên tại Trường Đại học Điện-Thông tin (UEC), Nhật Bản. PGS.TS Trần Xuân Nam được trao danh hiệu *Thủ khoa chuyên ngành Thông tin* của Học viện KTQS năm 1993, giải thưởng *Nhà khoa học Trẻ của Hiệp hội IEEE-APS, Nhật Bản* năm 2003, và đồng giải thưởng *Bài báo xuất sắc nhất Hội nghị Quốc tế về các Công nghệ tiên tiến trong Truyền thông* 2012 (ATC2012). Hiện là Phó chủ nhiệm Khoa Vô tuyến Điện tử, Học viện Kỹ thuật Quân sự. Hướng nghiên cứu chính liên quan đến kỹ thuật xử lý tín hiệu cho thông tin vô tuyến bao gồm: kỹ thuật truyền dẫn, tách tín hiệu đa người dùng, kỹ thuật triệt nhiễu, ăng-ten thích nghi, mã không gian-thời gian và các hệ thống MIMO.

**TS Lê Minh Tuấn** sinh ngày 27/10/1976 tại Hoàng Hợp, Hoằng Hóa, Thanh Hóa. Tốt nghiệp Trường Đại học Bách khoa Hà Nội năm 1999, chuyên ngành Điện tử-Viễn thông, tốt nghiệp Thạc sỹ và Tiến sỹ tại Trường Đại học Thông tin Liên lạc (ICU), nay là Phân viện tại Viện Khoa học và Công nghệ Tiên tiến Hàn Quốc (KAIST) vào các năm 2003 và 2007. TS Lê Minh Tuấn là Giảng viên Khoa Kỹ thuật Điện tử I, Học viện Công nghệ Bưu chính Viễn thông từ năm 1999 đến 2008. Hiện đang làm việc Trung tâm Nghiên cứu, Chuyển giao Công nghệ và Phân tích, Sở Khoa học và Công nghệ Hà Nội. TS Tuấn nhận đồng giải thưởng *Bài báo xuất sắc nhất Hội nghị Quốc tế về các Công nghệ tiên tiến trong Truyền thông* 2012 (ATC2012). Lĩnh vực nghiên cứu chính bao gồm thông tin vô tuyến, mã không gian - thời gian, và các hệ thống MIMO.