

# XỬ LÝ TÍN HIỆU NGẪU NHIÊN

## Tập II: Nâng cao

NGUYỄN LINH TRUNG (chủ biên)  
HUỲNH HỮU TUỆ

Giáo trình: Xử lý tín hiệu ngẫu nhiên - Tập II: Nâng cao  
Nguyễn Linh Trung, Huỳnh Hữu Tuệ  
Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội, 2025

# MỤC LỤC

	<i>Danh sách bảng</i>	iv
	<i>Danh sách hình vẽ</i>	v
	<i>Danh sách thuật toán</i>	x
	<i>Danh sách chương trình MATLAB</i>	xi
	<i>Danh mục thuật ngữ và viết tắt</i>	xv
	<i>Quy ước ký hiệu và phong chữ</i>	xvii
	<i>Lời mở đầu</i>	xv
<b>1</b>	<b>Ước lượng phổ theo mô hình</b>	<b>1</b>
1.1	Phương pháp Burg sử dụng mô hình tự hồi quy	2
1.2	Phương pháp phân tích trị riêng Pisarenko	14
1.3	Thuật toán MUSIC	22
1.4	Thuật toán ESPRIT	25
1.5	Thực thi thuật toán Pisarenko và MUSIC	28
1.6	Tóm tắt chương	40
	<i>Phụ lục Chương 1</i>	41
	1.A Toán tử dự đoán bậc $N$	41
	1.B Thuật toán Levinson-Durbin	43
	1.C Thuật toán Burg	49
	<i>Bài tập Chương 1</i>	54
<b>2</b>	<b>Lọc tối ưu Kalman</b>	<b>57</b>
2.1	Bộ lọc Kalman	57
2.2	Lọc Kalman mở rộng	90

2.3	Bộ lọc Kalman mở rộng đa điểm xích-ma	101
2.4	Tóm tắt chương	108
	<i>Phụ lục Chương 2</i>	109
2.A	Ước lượng sai số bình phương trung bình tối thiểu	109
2.B	Mô hình chuyển động của vật thể	114
	<i>Bài tập Chương 2</i>	118
<b>3</b>	<b>Lọc tự thích nghi</b>	120
3.1	Lọc tự thích nghi LMS	121
3.2	Thuật toán LMS	128
3.3	Một số dạng cải biến của LMS	136
3.4	Lọc tự thích nghi RLS	141
3.5	Thuật toán RLS	144
3.6	Tóm tắt chương	162
	<i>Phụ lục Chương 3</i>	163
3.A	Định thức và nghịch đảo ma trận dạng khối	163
	<i>Bài tập Chương 3</i>	166
<b>4</b>	<b>Lọc thích nghi trong mạng neuron</b>	168
4.1	Lịch sử phát triển mạng neuron	169
4.2	Cấu trúc mạng neuron	171
4.3	Thuật toán tập huấn Widrow-Hoff	182
4.4	Phương pháp tập huấn lan truyền ngược	200
4.5	Tóm tắt chương	219
	<i>Bài tập Chương 4</i>	224
<b>5</b>	<b>Ước lượng hướng đến của tín hiệu</b>	227
5.1	Mô hình hệ thống và tín hiệu	228
5.2	Dạng ma trận của tập chòm dữ liệu	232
5.3	Hệ thống tạo búp bức xạ	234
5.4	Hệ thống ước lượng hướng đến của tín hiệu	236
5.5	Thực thi một số thuật toán ước lượng DOA	242

<i>Phụ lục Chương 5</i>	247
5.A Mô hình tín hiệu dải thông hẹp	247
<i>Bài tập Chương 5</i>	251
<i>Tài liệu tham khảo</i>	253

Giáo trình: Xử lý tín hiệu ngẫu nhiên - Tập II: Nâng cao  
Nguyễn Linh Trung, Huỳnh Hữu Tuệ  
Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội, 2025

# 1 Ước lượng phổ theo mô hình

Trong Chương 4 của Tập I giáo trình Xử lý tín hiệu ngẫu nhiên [18], chúng ta đã tìm hiểu các phương pháp ước lượng phổ như tuần hoàn đồ, phương pháp Bartlett-Welch và phương pháp Blackman-Tukey. Các phương pháp này sử dụng trực tiếp tập  $N$  dữ liệu  $\{x(0), \dots, x(N-1)\}$  quan sát được từ tín hiệu  $\{X(n)\}, n \in \mathbb{Z}$ , để tính trực tiếp ước lượng  $\hat{S}_{XX}(\omega)$  của  $S_{XX}(\omega)$ .

Trong rất nhiều tình huống, số dữ liệu quan sát được không lớn, nên những phương pháp trên không đạt được kết quả thích hợp. Một số nhà nghiên cứu đã gợi ý một hướng đi mới, đó là xây dựng mô hình một hệ thống thế nào để lúc kích thích bởi một nhiễu trắng, phổ công suất đầu ra tương ứng với ước lượng của phổ công suất của dữ liệu ta quan sát.

Những kết quả đầu tiên được công bố bởi Burg [6, 1967]; phương pháp này được gọi là thuật toán Burg hoặc **phương pháp entropy cực đại** (MEM: maximum entropy method). Thuật toán Burg để ước lượng phổ công suất một tín hiệu ngẫu nhiên dừng bậc hai có  $p$  điểm của hàm tự tương quan cho trước được xây dựng trên kết quả sau đây, thường được gọi là định lý Burg như dưới đây. Phổ công suất suy ra từ cấu trúc này được gọi là **phổ Burg**.

## Định lý Burg

Một tín hiệu ngẫu nhiên dừng bậc hai với  $p$  điểm của hàm tự tương quan cho trước có entropy cực đại là một tín hiệu Gauss có cấu trúc tự hồi quy bậc  $p$ .

Ngoài hướng xây dựng mô hình hệ thống mà Burg đã nêu ra, còn có một hướng khác là xây dựng trực tiếp mô hình của tín hiệu, rồi ước lượng tham số của mô hình tín hiệu, đã làm xuất hiện một số phương pháp đáng chú ý khác như Capon [8], MUSIC [1–3, 26], ESPRIT [19, 22]. Chương này sẽ trình bày một số phương pháp ước lượng phổ đã được đưa vào sử dụng trong thực tiễn từ nửa sau thế kỷ 20.

### 1.1 Phương pháp Burg sử dụng mô hình tự hồi quy

Trong thực tế, để ước lượng phổ công suất của một tín hiệu, trước tiên ta phải đo lường (quan sát) tín hiệu đó. Quá trình đo lường này chịu ảnh hưởng của hai yếu tố quan trọng hàng đầu: chiều dài tín hiệu quan sát được là hữu hạn và dữ liệu thu thập được (tín hiệu) luôn chứa sai số (nhiều quan sát).

Ta không thể loại bỏ nhiễu quan sát, vậy ta xét trường hợp tập dữ liệu hữu hạn. Khi ta có khá nhiều dữ liệu, kết quả ước lượng phổ công suất với phương pháp không dùng tham số (mô hình) có chất lượng hoặc cao, hoặc chấp nhận được; đây chính là các phương pháp cổ điển được trình bày trong phần trước của chương. Trong các phương pháp ước lượng phổ không sử dụng tham số, ta giả thiết một cách gián tiếp là các dữ liệu đều triệt tiêu ngoài khoảng ta quan sát được. Nếu chiều dài khoảng quan sát tương đối ngắn thì giả thiết này ảnh hưởng rất lớn đến chất lượng ước lượng với độ phân giải thấp, như ta đã thấy ở phần trước. Để giải quyết vấn đề này, ta tìm cách ngoại suy ước lượng hàm tự tương quan ở ngoài khoảng quan sát. Cách thực hiện hợp lý nhất là xây dựng một mô hình cho tín hiệu (dữ liệu) thế nào để trong khoảng ta quan sát, tín hiệu có tính chất thống kê tương đương với những gì ta quan sát được.

Mô hình tương đối phổ cập là hệ thống tự hồi quy trung bình trượt (ARMA: autoregressive - moving average) mà tín hiệu đầu vào và tín hiệu đầu ra được kết nối bởi hàm truyền

$H(z)$  sau đây:

$$\frac{X(z)}{W(z)} = H(z) = \frac{\sum_{k=0}^Q b_k z^{-k}}{\sum_{k=0}^P a_k z^{-k}}, \quad (1.1)$$

trong đó  $W(z)$  và  $X(z)$  là đầu vào và đầu ra của hệ thống. Hàm truyền  $H(z)$  cũng có thể viết dưới dạng

$$H(z) = \frac{\sum_{k=0}^Q b_k z^{-k}}{1 - \sum_{k=1}^P a_k z^{-k}}. \quad (1.2)$$

Dạng (1.2) được chuẩn hóa, đối dấu so với dạng (1.1) và là tương ứng với hệ phương trình sai phân sau đây:

$$x(n) = - \sum_{k=1}^P a_k x(n-k) + v(n), \quad (1.3a)$$

$$v(n) = \sum_{k=0}^Q b_k w(n-k). \quad (1.3b)$$

Dạng (1.3) có một ý nghĩa vật lý mà ta sẽ thảo luận sau.

Bây giờ, nếu đầu vào là một tín hiệu dừng mở rộng có phổ công suất là  $S_w(\omega)$  thì đầu ra có phổ công suất  $S_x(\omega)$  cho bởi

$$S_x(\omega) = S_w(\omega) |H(\omega)|^2. \quad (1.4)$$

Nếu  $w(n)$  là một nhiễu trắng có phương sai là đơn vị, tức là

$$S_w(\omega) = 1,$$

$S_x(\omega)$  trở thành

$$S_x(\omega) = |H(\omega)|^2. \quad (1.5)$$

Năm 1967, Burg đã gợi ý dùng mô hình tự hồi quy (AR: autoregressive), tương đương với  $Q = 0$  trong (1.3) khi biểu

diễn mô hình ARMA, vì ông ta đã tìm ra một kết quả rất đặc biệt và hữu ích mang tên là phương pháp entropy cực đại. Kết quả này đã được Burg mở rộng và trình bày chặt chẽ trong luận án tiến sĩ của mình mà ông đã bảo vệ thành công tại Đại học Stanford năm 1975. Người ta đặt tên cho phổ công suất ước lượng bởi phương pháp MEM là phổ Burg. Sau đây, ta sẽ tập trung vào phương pháp MEM của Burg.

### 1.1.1 Phổ Burg của một tín hiệu có quan sát hữu hạn

Cho một tín hiệu ngẫu nhiên  $\{X(n)\}$  có phổ công suất như sau:

$$S_X(\omega) = \sum_{n=-\infty}^{\infty} R_X(n) e^{-jn\omega},$$

trong đó  $R_X(n)$  là hàm tự tương quan của  $\{X(n)\}$ :

$$R_X(n) = \mathbb{E}\{X(k)X(k-n)\}.$$

Không mất tính tổng quát, giả sử rằng tín hiệu ngẫu nhiên  $\{X(n)\}$  được đặc trưng bởi chuỗi  $n$  biến ngẫu nhiên  $X_1, \dots, X_n$ . Đại lượng **entropy vi phân** (differential entropy) của chuỗi  $X_1, X_2, \dots, X_n$  được định nghĩa và ký hiệu như sau:

$$h(X_1, \dots, X_n) = - \int \cdots \int f(x_1, \dots, x_n) \log f(x_1, \dots, x_n) dx_1 \cdots dx_n, \quad (1.6)$$

trong đó  $f(x_1, \dots, x_n)$  là hàm mật độ xác suất của vector  $(X_1, \dots, X_n)$ . Nếu giới hạn

$$h = \lim_{n \rightarrow \infty} \frac{h(X_1, \dots, X_n)}{n} \quad (1.7)$$

hiện hữu, ta gọi  $h$  là **tốc độ biến thiên của entropy** (differential entropy rate) của tín hiệu ngẫu nhiên  $\{X(n)\}$ . Lúc  $\{X(n)\}$  là một quá trình Gauss, Kolmogorov chứng minh được rằng

$$h = \frac{1}{2} \log(2\pi e) + \frac{1}{2} \int_{-0.5}^{0.5} \log S_X(\nu) d\nu, \quad (1.8)$$



trong đó

$$S_X(\nu) = \sum_{n=-\infty}^{\infty} R_X(n) e^{-j2\pi n\nu}. \quad (1.9)$$

Từ kết quả (1.8) và (1.9), Burg đã nêu bài toán sau:

### Bài toán Burg

Giả sử tín hiệu  $\{X(n)\}$  là dừng bậc hai có các giá trị của hàm tự tương quan  $R_X(n)$  ứng với  $-p \leq n \leq p$  là cho trước, tính các thành phần  $R_X(n)$  còn lại ứng với  $|n| > p$  thế nào để  $h$  đạt cực đại.

Mục tiêu cực đại hóa  $h$  là để tìm phần còn lại của  $R_X(n)$  sao cho phổ được ước lượng là tương ứng với tín hiệu ngẫu nhiên nhất (khó dự đoán nhất), tức là đã biểu diễn tín hiệu tối ưu nhất theo nguyên lý cực đại entropy. Ta chú ý rằng cách nhìn này khác với cách nhìn của các phương pháp cổ điển đã trình bày ở trên; các phương pháp cổ điển giả sử  $R_X(n)$  triệt tiêu với  $|n| > p$ .

Đáp án của bài toán Burg có tên là **định lý entropy cực đại** của Burg, được nêu như sau:

### Định lý entropy cực đại

Quá trình ngẫu nhiên dừng mở rộng  $\{X(n)\}$  có entropy cực đại mà thỏa mãn điều kiện hàm tự tương quan

$$R_X(m) = r(m), \quad m = 0, \pm 1, \dots, \pm p, \quad (1.10)$$

là một quá trình ngẫu nhiên Gauss-Markov bậc  $p$  có dạng

$$X(n) = - \sum_{k=1}^p a_k X(n-k) + W(n), \quad (1.11)$$

trong đó  $\{W(n)\}$  là một quá trình Gauss trắng có công suất là  $\mathbb{E}\{W^2(n)\} = \sigma^2$  và các hệ số  $a_1, \dots, a_p, \sigma$  được xác định bởi các ràng buộc (1.10) cho trước.

Sau đây, ta có thể làm rõ hơn kết quả trên. Cho  $\{X(n)\}$  là một quá trình ngẫu nhiên Gauss dừng mở rộng. Xét vector ngẫu nhiên

$$\mathbf{X}_p = \{X(1), X(2), \dots, X(p)\}. \quad (1.12)$$

$\mathbf{X}_p$  là một vector ngẫu nhiên Gauss có vector trung bình  $\mathbf{m}_p = \mathbb{E}\{\mathbf{X}_p\}$  và ma trận hiệp phương sai

$$\begin{aligned} \Lambda_p &= [\lambda_p(i, j)], \\ \lambda_p(i, j) &= \mathbb{E}\{[X(i) - \mathbf{m}_p(i)][X(j) - \mathbf{m}_p(j)]\}. \end{aligned} \quad (1.13)$$

Trong lý thuyết thông tin (có thể xem giáo trình [10, Cover, 1991]), dùng phương pháp nhân tử Lagrange ta chứng minh được rằng dưới ràng buộc  $\Lambda_p$  cho trước, vector ngẫu nhiên  $\mathbf{X}_p$  đạt được entropy cực đại khi  $\mathbf{X}_p$  có phân phối Gauss. Như vậy, lúc  $p \rightarrow \infty$ , quá trình  $\{X(n)\}$  có hàm tự tương quan  $R_X(n) = \mathbb{E}\{X(m)X(m-n)\}$  cho trước có entropy cực đại khi  $\{X(n)\}$  là một quá trình Gauss.

Với quá trình Gauss dừng bậc hai có hàm tự tương quan  $R_X(n)$ ,

$$\mathbf{m}_p(i) = m_X, \text{ với mọi } i, \quad (1.14)$$

$$\Lambda_p = [\lambda_p(i, j)] = R_X(i - j) - m_X^2, \quad (1.15)$$

có thể tính được

$$h(X_1, \dots, X_p) = \frac{1}{2} \log(2\pi e)^p |\Lambda_p|. \quad (1.16)$$

Kolmogorov đã chứng minh được entropy của quá trình Gauss  $\{X(n)\}$  được cho bởi

$$h\{X(n)\} = \frac{1}{2} \log(2\pi e) + \frac{1}{2} \int_{-0.5}^{0.5} \log S_X(\nu) d\nu,$$

với  $S_X(\nu)$  là biến đổi Fourier của hàm tự hiệp phương sai  $\lambda_X(n)$

$$S_X(\nu) = \sum_{n=-\infty}^{\infty} \lambda_X(n) e^{-jn2\pi\nu}. \quad (1.17)$$

Khi trị trung bình của quá trình  $\{X(n)\}$  triệt tiêu, ta có

$$\lambda_X(n) = R_X(n), \quad (1.18)$$

do đó

$$S_X(\nu) = \sum_{n=-\infty}^{\infty} R_X(n) e^{-jn2\pi\nu}. \quad (1.19)$$

$S_X(\nu)$  chính là phổ công suất của tín hiệu Gauss dừng mở rộng có trị trung bình triệt tiêu và hàm tự tương quan  $R_X(n)$ .

Dựa trên kết quả này, bài toán Burg trở nên đơn giản hơn:

#### Bài toán Burg - phổ MEM

Cho trước các giá trị của  $R_X(n)$  với  $n = -p, \dots, p$ , tìm các giá trị của  $R_X(n)$  với  $|n| > p$  để tối đa hóa tích phân

$$\int_{-0,5}^{0,5} \log S_X(\nu) d\nu.$$

Dưới đây, ta sẽ cần tìm mô hình của phổ MEM của Burg, để có thể tính  $R_X(n)$  với  $|n| > p$ , từ đó tính các tham số mô hình theo các ràng buộc  $R_X(n) = r(n)$  với  $n = 0, \pm 1, \dots, \pm p$ .

#### Tìm mô hình cho phổ MEM của Burg

Đặt

$$H(S_X(\nu)) = \int_{-0,5}^{0,5} \log S_X(\nu) d\nu. \quad (1.20)$$

Lấy đạo hàm của  $H(S_X(\nu))$  theo  $R_X(n)$  khi  $|n| > p$

$$\frac{\partial H(S_X(\nu))}{\partial R_X(n)} = \int_{-0,5}^{0,5} \frac{\partial \log S_X(\nu)}{\partial R_X(n)} d\nu, \quad |n| > p. \quad (1.21)$$

$R_X(n)$  mà làm cho  $H(S_X(\nu))$  đạt tối đa là nghiệm của phương trình sau:

$$\int_{-0,5}^{0,5} \frac{\partial}{\partial R_X(n)} \log S_X(\nu) d\nu = 0, \quad |n| > p. \quad (1.22)$$

Ta biết

$$\begin{aligned}\frac{\partial}{\partial R_X(n)} \log S_X(\nu) &= \frac{1}{S_X(\nu)} \frac{\partial S_X(\nu)}{\partial R_X(n)} \\ &= \frac{1}{S_X(\nu)} e^{-jn2\pi\nu},\end{aligned}\quad (1.23)$$

nên điều kiện (1.22) trở thành

$$\int_{-0,5}^{0,5} \frac{1}{S_X(\nu)} e^{-jn2\pi\nu} d\nu = 0, \quad |n| > p. \quad (1.24)$$

Bởi vì  $S_X(\nu)$  là tuần hoàn có chu kỳ theo  $\nu$  bằng 1, cho nên  $1/S_X(\nu)$  cũng tuần hoàn theo  $\nu$  có cùng chu kỳ là 1. Từ đó, ta có thể khai triển  $1/S_X(\nu)$  theo chuỗi Fourier

$$\frac{1}{S_X(\nu)} = \sum_{n=-\infty}^{\infty} c(n) e^{-jn2\pi\nu}, \quad (1.25)$$

trong đó hệ số Fourier là

$$c(n) = \int_{-0,5}^{0,5} \frac{1}{S_X(\nu)} e^{jn2\pi\nu} d\nu.$$

Điều kiện (1.24) cho ta

$$c(n) = 0, \quad |n| > p. \quad (1.26)$$

Từ (1.25) và (1.26), ta suy ra ước lượng phổ Burg  $\hat{S}_X(\nu)$  của  $\{X(n)\}$  như sau:

$$\hat{S}_X(\nu) = \frac{1}{\sum_{n=-p}^p c(n) e^{-jn2\pi\nu}}. \quad (1.27)$$

$\hat{S}_X(\nu)$  là phổ công suất của tín hiệu  $\{X(n)\}$  mà làm tối đa hóa entropy  $H(S_X(\nu))$ . Bởi vì  $\hat{S}_X(\nu)$  dương và đối xứng qua gốc ( $\hat{S}_X(\nu)$  là một hàm chẵn theo  $\nu$ ), ta có thể phân tích hàm

$$\frac{1}{\hat{S}_X(z)} = \sum_{n=-p}^p c(n) z^{-n} \quad (1.28)$$

thành

$$\frac{1}{\hat{S}_X(z)} = \frac{1}{\sigma^2} A(z) A(z^{-1}), \quad (1.29)$$

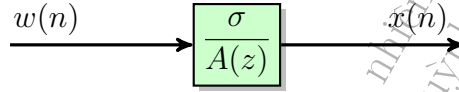
trong đó

$$A(z) = 1 + a_1 z^{-1} + \dots + a_p z^{-p}. \quad (1.30)$$

Như thế

$$\hat{S}_X(z) = \frac{\sigma^2}{A(z) A(z^{-1})}. \quad (1.31)$$

Kết quả (1.31) cho thấy phổ MEM của Burg chính là phổ công suất của một tín hiệu ngẫu nhiên có dạng AR. Mô hình minh họa trong Hình 1.1.



Hình 1.1 Mô hình tín hiệu AR cho phổ Burg.

#### Tính các hệ số của mô hình AR

Tiếp theo, ta cần tính các hệ số  $\sigma, a_1, a_2, \dots, a_p$  của mô hình AR trong Hình 1.1 từ các ràng buộc

$$R_X(n) = r(n), \quad n = 0, \pm 1, \dots, \pm p. \quad (1.32)$$

Giả sử đầu vào  $\{w(n)\}$  của hệ thống AR là một nhiễu trắng Gauss dừng mở rộng có phổ công suất là

$$S_w(\nu) = 1, \quad (1.33)$$

phổ công suất của đầu ra sẽ là

$$\begin{aligned} S_X(\nu) &= S_w(\nu) |H(\nu)|^2 \\ &= |H(\nu)|^2 \\ &= \frac{\sigma^2}{\left| 1 + \sum_{k=1}^p a_k e^{-jk2\pi\nu} \right|^2}. \end{aligned} \quad (1.34)$$

Bài toán đặt ra là tính hệ số  $\sigma$  và  $p$  hệ số  $a_k$ ,  $k = 1, \dots, p$ , nếu ta biết  $2p+1$  các trị  $r(n)$ ,  $n = 0, \pm 1, \dots, \pm p$ , của hàm tự tương quan  $R_X(n)$ . Nhắc lại rằng,  $R_X(n)$  có mối liên hệ với phổ công suất thông qua định lý Wiener-Khintchin

$$S_X(\nu) = \sum_{n=-\infty}^{\infty} R_X(n) e^{-jn2\pi\nu}$$

Bởi vì ta chỉ xét trường hợp  $X(n)$  thực, nên  $r(n) = r(-n)$ ,  $2p+1$  tham số  $r(n)$  thực chất chỉ gồm  $p+1$  tham số  $r(0), r(1), \dots, r(p)$ .

Mối liên hệ giữa  $x(n)$  và  $w(n)$  của hệ thống Hình 1.1 là

$$x(n) = -a_1 x(n-1) - \dots - a_p x(n-p) + \sigma w(n). \quad (1.35)$$

Trong lĩnh vực xử lý tín hiệu, phương trình (1.35), ngoài được sử dụng để mô tả một hệ thống AR, còn có một ý nghĩa đặc biệt, là bài toán dự đoán. Giả sử ở thời điểm  $n-1$  ta đã có  $p$  dữ liệu trong tay:  $x(n-1), x(n-2), \dots, x(n-p)$ . Ta muốn dự đoán dữ liệu sắp nhận được  $x(n)$  bằng một tổ hợp tuyến tính của  $x(n-1), \dots, x(n-p)$ :

$$\hat{x}(n) = \sum_{k=1}^p c_k x(n-k). \quad (1.36)$$

Sai số dự đoán là

$$e(n) = x(n) - \hat{x}(n). \quad (1.37)$$

Từ (1.35) và (1.37) ta có biểu thức

$$x(n) = \sum_{k=1}^p c_k x(n-k) + e(n). \quad (1.38)$$

Cần chọn các hệ số  $c_1, c_2, \dots, c_p$  thế nào để sai số bình phương trung bình  $\mathbb{E}\{e^2(n)\}$  đạt tối thiểu.

Ta thấy mô hình AR đặc tả bởi phương trình (1.35) và mô hình dự đoán (1.38) là đồng dạng. Biểu thức (1.36) được gọi là toán tử dự đoán tuyến tính 1 bước. Những tính chất quan trọng của toán tử dự đoán tuyến tính sẽ được trình bày trong Phụ lục 1.A ở cuối chương. Nhận xét là mô hình (1.35)

với dấu trừ trước các hệ số  $a_k$  là biểu diễn theo (1.1), còn dự đoán tuyến tính (1.36) với dấu cộng trước các hệ số  $c_k$  là biểu diễn theo (1.2).

Bây giờ ta sẽ biểu diễn các hệ số  $\sigma, a_1, \dots, a_p$  theo các tham số  $r(0), r(1), \dots, r(p)$ , trong đó

$$r(n) = R_X(n).$$

Viết phương trình (1.35) dưới dạng

$$x(n) + a_1x(n-1) + \dots + a_px(n-p) = \sigma w(n). \quad (1.39)$$

Nhiều  $w(n)$  chỉ tác động đến  $x(n)$  và không liên quan gì đến  $x(n-1), \dots, x(n-p)$ , nên ta có

$$\begin{aligned} \mathbb{E}\{w(n)x(n-m)\} &= \mathbb{E}\{w(n)\} \mathbb{E}\{x(n-m)\} \\ &= 0, \quad m = 1, \dots, p. \end{aligned}$$

Từ đó, nhân hai vế của (1.39) với  $w(n)$  rồi lấy trung bình, ta suy ra

$$\mathbb{E}\{w(n)x(n)\} = \sigma \mathbb{E}\{w^2(n)\} = \sigma.$$

Tiếp tục, nhân hai vế của (1.39) với  $x(n)$  rồi lấy trung bình, cũng như với  $x(n-m)$  rồi lấy trung bình, ta suy ra

$$\mathbb{E}\{x^2(n)\} + \sum_{k=1}^p a_k \mathbb{E}\{x(n-k)x(n)\} = \sigma^2, \quad (1.40a)$$

$$\begin{aligned} \mathbb{E}\{x(n)x(n-m)\} + \sum_{k=1}^p a_k \mathbb{E}\{x(n-k)x(n-m)\} \\ = 0, \quad m = 1, \dots, p. \end{aligned} \quad (1.40b)$$

Biểu diễn (1.40) theo tự tương quan, ta được

$$r(0) + \sum_{k=1}^p a_k r(k) = \sigma^2, \quad (1.41a)$$

$$r(m) + \sum_{k=1}^p a_k r(m-k) = 0, \quad m = 1, \dots, p. \quad (1.41b)$$

Do  $r(m-k) = r(k-m)$ , ta có thể viết (1.41) dưới dạng ma

trận như sau:

$$\begin{bmatrix} r(0) & r(1) & \cdots & r(p) \\ r(1) & r(0) & \cdots & r(p-1) \\ \vdots & \vdots & \ddots & \vdots \\ r(p) & r(p-1) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \sigma^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (1.42)$$

Kết quả (1.42) có tên là **hệ phương trình Yule-Walker**. Giải hệ phương trình này, ta sẽ suy ra được  $\sigma$  và  $a_1, \dots, a_p$ , tức là suy ra mô hình AR tương ứng với định lý Burg.

### 1.1.2 Thuật toán Burg

Người ta thường dùng thuật toán Levinson-Durbin, đệ quy theo bậc  $p$  của mô hình cho  $p$  từ 0 đến  $N$ , để giải hệ phương trình Yule-Walker được cho trong (1.42). Xem chi tiết thuật toán này trong Phụ lục 1.B. Thuật toán Levinson-Durbin khá thuận tiện và đã được sử dụng tương đối rộng rãi. Tuy vậy, có đôi lúc thuật toán trở thành bất ổn định.

Burg đã đề nghị một thuật toán luôn luôn bảo đảm tính ổn định để cho kết quả hoàn toàn thích hợp. Xem chi tiết thuật toán Burg trong Phụ lục 1.C. Các ý chính của thuật toán này như sau.

Dùng cấu trúc dàn chéo để luôn luôn có 2 sai số dự đoán dựa vào tập gồm  $p$  dữ liệu  $\{x(n-p+1), \dots, x(n)\}$  để ước lượng dữ liệu cách 1 bước trong tương lai  $\hat{x}(n+1)$  so với  $x(n)$  và dữ liệu cách 1 bước trong quá khứ  $\hat{x}(n-p)$  so với  $x(n-p+1)$ . Các sai số này tương ứng là

$$e_p^+(n) = x(n) - \hat{x}^+(n), \quad (1.43)$$

$$e_p^-(n) = x(n-p) - \hat{x}^-(n-p). \quad (1.44)$$

Sai số bình phương trung bình sẽ là

$$\varepsilon_p = \sum_{n=p}^N [(e_p^+(n))^2 + (e_p^-(n))^2].$$

Đồng thời, tiếp tục cách làm của Levinson-Durbin, nhưng thay thế trị trung bình thống kê bằng trị trung bình phương theo thời gian.



## 1.1.3 Tiêu chí xác định bậc của mô hình

Thuật toán Burg và một số thuật toán khác, là phương pháp xây dựng một mô hình tín hiệu loại AR bậc  $p$ , trước khi tính các tham số của mô hình tín hiệu dựa trên dữ liệu  $\{x(n)\}$ ,  $n = 1, \dots, N$ , ta chọn bậc  $p$  và cuối cùng ta tính sai số bình phương trung bình  $\varepsilon_p$ . Như thế, ta phải chọn bậc  $p$  của mô hình như thế nào? Nếu ta chọn  $p$  nhỏ thì thuật toán có khuynh hướng làm mượt phổ ta muốn. Ngược lại, nếu chọn  $p$  quá lớn thì thuật toán sẽ làm xuất hiện những đỉnh không có thật trong phổ.

Có một số tiêu chí được đề nghị và hầu hết đều có cơ sở lý thuyết khá phức tạp. Sau đây, ta tóm tắt bốn tiêu chí thường được sử dụng và cho kết quả tương đối thích hợp.

Cách làm tự nhiên nhất là tăng dần  $p$  cho đến lúc  $\varepsilon_p$  đạt được giá trị nhỏ nhất. Tuy nhiên, trong thực tiễn,  $\varepsilon_p$  giảm dần với  $p$ . Vì vậy, ta nên xây dựng một hàm chất lượng, hay hàm phí tổn, và chọn  $p$  để tối ưu hóa hàm này. Hàm phí tổn  $C(p)$  thường gặp gồm hai phần, một phần phụ thuộc vào  $\varepsilon_p$  và phần còn lại phụ thuộc vào bậc  $p$ . Do đó, ta có thể mô hình hóa  $C(p)$  theo dạng sau:

$$C(p) = N \log \varepsilon_p + f(N)p, \quad (1.45)$$

trong đó  $N$  là chiều dài của dữ liệu  $\{x(n)\}$ ,  $p$  là bậc của mô hình và  $f(N)$  là một hằng số phụ thuộc vào  $N$ .

Tiêu chí đầu tiên ta xét đến có tên là **sai số dự đoán cuối cùng** (FPE: final prediction error), do Akaike đề nghị:

$$\text{FPE}(p) = \varepsilon_p \frac{N + p + 1}{N - p - 1}. \quad (1.46)$$

Tiêu chí thứ hai là **tiêu chí hàm truyền tự hồi quy** (CAT: criterion autoregressive transfer function), do Parzen xây dựng:

$$\text{CAT}(p) = \left( \frac{1}{N} \sum_{k=1}^p \frac{N - k}{N \varepsilon_k} \right) - \frac{N - p}{N \varepsilon_p}. \quad (1.47)$$

Tiêu chí thứ ba là **tiêu chí thông tin Akaike** (AIC:

Akaike information criterion), do Rissanen xây dựng

$$\text{AIC}(p) = N \log \varepsilon_p + 2p. \quad (1.48)$$

Tiêu chí thứ tư là **độ dài mô tả tối thiểu** (MDL: minimum description length), cũng do Rissanen xây dựng:

$$\text{MDL}(p) = N \log \varepsilon_p + p \log N. \quad (1.49)$$

Hai tiêu chí FPE và CAT khá phổ cập. Hai tiêu chí AIC và MDL rất nổi tiếng. Lúc  $\{x(n)\}$  khá dài ( $N$  lớn), các tiêu chí này đều cho kết quả chấp nhận được. Tuy nhiên, lúc  $\{x(n)\}$  ngắn ( $N$  nhỏ), các tiêu chí này cho  $p$  nhỏ hơn trị số cần thiết. Vì vậy, ta chỉ nên sử dụng các tiêu chí này để chọn sơ bộ bậc  $p$  của mô hình. Đây cũng là điều hiển nhiên vì mô hình ta chọn để mô tả dữ liệu chưa chắc đã có độ chính xác cao và như thế  $p$  phải được chọn tùy theo tình huống và bản chất các vấn đề ta phải giải quyết. Các mô hình thường được chọn là AR, trung bình trượt (MA: moving average), ARMA, tín hiệu điều hòa.

## 1.2 Phương pháp phân tích trị riêng Pisarenko

Ta vừa xây dựng thuật toán Burg, là thuật toán đầu tiên dựa vào phương pháp xây dựng mô hình cho tín hiệu. Dựa vào định lý entropy cực đại, Burg đã gợi ý sử dụng mô hình AR và từ đó xây dựng thuật toán của mình. Về sau, cái nhìn của Burg đã được nhiều nhà nghiên cứu mở rộng nhằm sử dụng các mô hình MA và ARMA (xem [7, 9]).

Loại mô hình này rất hữu ích khi số dữ liệu đo được không đủ lớn để áp dụng các phương pháp tuần hoàn đồ của Bartlett hay của Blackman-Tukey. Nó cho ta một ước lượng phổ có độ phân giải cao, mặc dù số dữ liệu khá giới hạn.

Ngoài loại mô hình tuyến tính này, còn có một loại mô hình khác, rất thích hợp trong thực tiễn, đó là mô hình tín hiệu điều hòa có chứa nhiều. Mô hình này dẫn đến một họ các phương pháp gọi chung là các phương pháp phân tích trị riêng (eigen-analysis, subspace analysis), như các phương

pháp Pisarenko, MUSIC, ESPRIT. Sau đây ta sẽ trình bày các phương pháp phân tích trị riêng.

### 1.2.1 Phân tích trị riêng của ma trận tự tương quan

Cho một tín hiệu ngẫu nhiên dừng mở rộng có dạng

$$x(n) = s(n) + w(n), \quad (1.50)$$

trong đó tín hiệu ta muốn ước lượng phổ công suất là tổng của  $p$  tín hiệu điều hòa phức

$$s(n) = \sum_{k=1}^p a_k e^{j(n\omega_k + \varphi_k)}, \quad (1.51)$$

$a_k$  và  $\omega_k$  là các hằng số mà ta chưa biết,  $\varphi_k$  là các pha gốc ngẫu nhiên độc lập với nhau và có cùng phân phối đều từ 0 đến  $2\pi$ ,  $w(n)$  là nhiễu quan sát, được xem như một quá trình ngẫu nhiên dừng mở rộng có công suất là  $\sigma_w^2$  và độc lập với tín hiệu  $s(n)$ .

Chú ý là ta đang xét tín hiệu có dạng phức, nên hàm tự tương quan của  $\{x(n)\}$  được định nghĩa như sau:

$$R_x(n) = \mathbb{E}\{x(k+n)x^*(k)\}.$$

Cấu trúc của  $\{s(n)\}$  và  $\{w(n)\}$  cho ta

$$R_x(n) = R_s(n) + \sigma_w^2 \delta(n), \quad (1.52)$$

trong đó

$$R_s(n) = \sum_{k=1}^p P_k e^{jn\omega_k}, \quad (1.53)$$

với  $P_k$  là công suất thành phần điều hòa thứ  $k$  của tín hiệu, được cho bởi  $P_k = a_k^2$ .

Gọi  $\mathbf{x}_M(n)$  là vector cột gồm  $M$  dữ liệu  $x(n)$ , có cấu trúc như sau:

$$\mathbf{x}_M(n) = \begin{bmatrix} x(n) \\ \vdots \\ x(n+M-1) \end{bmatrix}.$$

Tương ứng, ta có vector cột của tín hiệu như sau:

$$\mathbf{s}_M(n) = [s(n) \ \cdots \ s(n+M-1)]^T.$$

Từ đó, ma trận tự tương quan  $\mathbf{R}_M^x$  của vector dữ liệu  $\mathbf{x}_M(n)$  được định nghĩa như sau:

$$\begin{aligned} \mathbf{R}_M^x &\triangleq \mathbb{E}\{\mathbf{x}_M(n)\mathbf{x}_M^H(n)\} \\ &= \begin{bmatrix} R_x(0) & R_x(-1) & \cdots & R_x(-M+1) \\ R_x(1) & R_x(0) & \cdots & R_x(-M+2) \\ \vdots & \vdots & \ddots & \vdots \\ R_x(M-1) & R_x(M-2) & \cdots & R_x(0) \end{bmatrix}. \end{aligned} \quad (1.54)$$

Thế (1.52) và (1.53) vào (1.54), ta có

$$\mathbf{R}_M^x = \mathbf{R}_M^s + \sigma_w^2 \mathbf{I}_M, \quad (1.55)$$

trong đó  $\mathbf{R}_M^s$  là ma trận tự tương quan của vector tín hiệu  $\mathbf{s}_M(n)$  và  $\mathbf{I}_M$  là ma trận đơn vị bậc  $M$ .

Sau đây, ta sẽ phân tích  $\mathbf{R}_M^x$  và  $\mathbf{R}_M^s$  dựa vào trị riêng (eigenvalue) và không gian riêng (eigen-subspace) của chúng. Để có thể trình bày súc tích, ta đặt

$$\mathbf{s}_k = [1 \ e^{j\omega_k} \ \cdots \ e^{j(M-1)\omega_k}]^T \quad (1.56)$$

là vector cột tương ứng với tham số  $\omega_k$ , dài  $M$ , rồi ghép các vector  $\mathbf{s}_1, \dots, \mathbf{s}_p$  để tạo ra ma trận

$$\mathbf{S} = [\mathbf{s}_1 \ \cdots \ \mathbf{s}_p]_{M \times p}. \quad (1.57)$$

Gọi  $\mathcal{S}$  là không gian được sinh (span) bởi  $\mathbf{s}_1, \dots, \mathbf{s}_p$ :

$$\mathcal{S} = \{\mathbf{s}_1, \dots, \mathbf{s}_p\}. \quad (1.58)$$

Không gian này có tên là không gian con chứa tín hiệu (signal subspace), hay gọn là **không gian tín hiệu**. Đặt

$$\mathbf{P} = \begin{bmatrix} P_1 & 0 & \cdots & 0 \\ 0 & P_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P_p \end{bmatrix}_{p \times p}, \quad (1.59)$$

trong đó, nhắc lại rằng,  $P_k$  là công suất của thành phần điều hòa thứ  $k$  của tín hiệu.

Với các ký hiệu này, ta có thể biểu diễn ma trận tự tương quan của tín hiệu,  $\mathbf{R}_M^s$ , như sau:

$$\mathbf{R}_M^s = \mathbf{S} \mathbf{P} \mathbf{S}^H. \quad (1.60)$$

Có thể viết lại ma trận  $\mathbf{S}$  là

$$\mathbf{S} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_p \\ \vdots & \vdots & \ddots & \vdots \\ z_1^{M-1} & z_2^{M-1} & \cdots & z_p^{M-1} \end{bmatrix}_{M \times p}, \quad z_p = e^{j\omega_p}. \quad (1.61)$$

Biểu diễn (1.61) cho thấy  $\mathbf{S}$  có hạng là  $p$ , với điều kiện là  $M \geq p$ , vì  $z_k \neq z_l$  với  $k \neq l$ . Theo đó, ma trận tự tương quan  $\mathbf{R}_M^s$ , là tích  $\mathbf{S} \mathbf{P} \mathbf{S}^H$ , cũng có hạng là  $p$ . Như thế, ma trận này có  $p$  trị riêng khác không và  $(M - p)$  trị riêng còn lại bằng không. Ngoài ra,  $\mathbf{R}_M^s$  là một ma trận tự tương quan bậc  $M$  bán xác định dương (positive semi-definite), tức là tất cả các trị riêng của nó không âm. Theo lý thuyết ma trận,  $\mathbf{R}_M^s$  là Hermite, tức là  $\mathbf{R}_M^s = (\mathbf{R}_M^s)^H$ . Với cấu trúc này, ta luôn luôn tìm được  $M$  vector riêng (eigenvector) trực chuẩn  $\mathbf{u}_k$  và  $M$  trị riêng  $\lambda_i$ , tức có thể biểu diễn

$$\mathbf{R}_M^s \mathbf{u}_i = \lambda_i \mathbf{u}_i, \quad i = 1, \dots, M, \quad (1.62)$$

với các trị riêng được sắp từ lớn đến nhỏ và thỏa mãn

$$\lambda_1 > \lambda_2 > \cdots > \lambda_p > \lambda_{p+1} = \cdots = \lambda_M = 0. \quad (1.63)$$

Bởi vì  $\mathbf{R}_M^s$  có  $(M - p)$  trị riêng triệt tiêu, ta có

$$\mathbf{R}_M^s = \sum_{i=1}^M \lambda_i \mathbf{u}_i (\mathbf{u}_i)^H = \sum_{k=1}^p \lambda_k \mathbf{u}_k (\mathbf{u}_k)^H. \quad (1.64)$$

Đặt  $\mathcal{S}_u$  là không gian được sinh bởi  $p$  vector  $\mathbf{u}_1, \dots, \mathbf{u}_p$ :

$$\mathcal{S}_u = \{\mathbf{u}_1, \dots, \mathbf{u}_p\}. \quad (1.65)$$

Ta có thể chứng minh rằng hai không gian  $\mathcal{S}_u$  trong (1.65) và không gian  $\mathcal{S}$  trong (1.58) đều chứa tất cả các tín hiệu  $\mathbf{s}_M(n)$ , nên  $\mathcal{S}_u$  cũng gọi là không gian tín hiệu.

Với ma trận tự tương quan của tín hiệu quan sát  $\mathbf{R}_M^x$  được cho bởi (1.54), ta thấy ngay vector riêng  $\mathbf{u}_k$  của  $\mathbf{R}_M^s$  trong (1.64) cũng là vector riêng của  $\mathbf{R}_M^x$ . Thật vậy,

$$\begin{aligned}\mathbf{R}_M^x \mathbf{u}_k &= \mathbf{R}_M^s \mathbf{u}_k + \sigma_w^2 \mathbf{I}_M \mathbf{u}_k \\ &= (\lambda_k + \sigma_w^2) \mathbf{u}_k, \quad k = 1, \dots, p.\end{aligned}\quad (1.66)$$

Kết quả (1.66) cho thấy  $\mathbf{u}_k$  cũng là vector riêng của  $\mathbf{R}_M^x$ , tương ứng với trị riêng  $\mu_k = \lambda_k + \sigma_w^2$ . Như thế,  $\mathbf{R}_M^x$  có  $p$  trị riêng đơn  $\lambda_k + \sigma_w^2$ ,  $k = 1, \dots, p$ , và  $(M - p)$  trị riêng bậc  $(M - p)$  bằng  $\sigma_w^2$ .

Các vector riêng  $\mathbf{u}_1, \dots, \mathbf{u}_p$  tương ứng với  $p$  trị riêng khác không được gọi là các **vector riêng chính** (principal eigen-vector) của ma trận tự tương quan  $\mathbf{R}_M^s$ , đồng thời cũng là  $p$  trong số  $M$  vector riêng của  $\mathbf{R}_M^x$ .

Kết quả (1.66) cũng cho thấy  $\mathbf{R}_M^x$  có  $M$  trị riêng khác không

$$\mu_1 = \lambda_1 + \sigma_w^2, \quad (1.67a)$$

$$\vdots = \vdots,$$

$$\mu_p = \lambda_p + \sigma_w^2, \quad (1.67b)$$

$$\mu_{p+1} = \dots = \mu_M = \sigma_w^2, \quad (1.67c)$$

tương ứng với các vector riêng trực chuẩn của  $\mathbf{R}_M^x$ , cũng là các vector riêng trực chuẩn của  $\mathbf{R}_M^x$ . Ta biểu diễn  $\mathbf{R}_M^x$  như sau:

$$\mathbf{R}_M^x = \sum_{k=1}^p (\lambda_k + \sigma_w^2) \mathbf{u}_k (\mathbf{u}_k)^H + \sum_{i=p+1}^M \sigma_w^2 \mathbf{u}_i (\mathbf{u}_i)^H. \quad (1.68)$$

Bên cạnh  $p$  vector riêng trực chuẩn  $\mathbf{u}_1, \dots, \mathbf{u}_p$  là vector trực chuẩn của  $\mathbf{R}_M^s$  sinh ra không gian tín hiệu  $\mathcal{S}_u$ ,  $M - p$  vector riêng trực chuẩn còn lại  $\mathbf{u}_{p+1}, \dots, \mathbf{u}_M$ , sinh ra không gian

$$\mathcal{N} = \{\mathbf{u}_{p+1}, \dots, \mathbf{u}_M\}, \quad (1.69)$$

được gọi là không gian con chứa nhiễu (noise subspace), hay thường gọn là **không gian nhiễu**.

Ta đã thấy không gian  $\mathcal{S}_u = \{\mathbf{u}_1, \dots, \mathbf{u}_p\}$  là không gian tín hiệu. Bởi vì  $\{\mathbf{u}_1, \dots, \mathbf{u}_M\}$  là trực chuẩn, ta thấy ngay

không gian nhiễu  $\mathcal{N}$  là trực giao với không gian tín hiệu  $\mathcal{S}_u$ . Đứng trên  $M$  vector  $\mathbf{u}_1, \dots, \mathbf{u}_M$  là các vector riêng của ma trận tín hiệu  $\mathbf{R}_M^s$  không chứa nhiễu; do đó khái niệm không gian tín hiệu và không gian nhiễu trở thành một công cụ hết sức hữu ích cho phân tích phổ tín hiệu.

### 1.2.2 Phương pháp phân tích điều hòa Pisarenko

Như vừa trình bày, với những giả thiết tương đối đơn giản, ta thấy không gian tín hiệu nhận được (gồm cả nhiễu) có thể phân tích thành hai không gian trực giao  $\mathcal{S}$  và  $\mathcal{N}$ , với  $\mathcal{S}$  là không gian tín hiệu (không bị nhiễu) và  $\mathcal{N}$  là không gian nhiễu.

Pisarenko [20] là người đầu tiên tìm thấy kết quả phân tích này cho trường hợp đặc biệt  $M = p + 1$ , tức là không gian tín hiệu  $\mathcal{S}$  có  $p$  chiều và không gian nhiễu  $\mathcal{N}$  chỉ có một chiều và không gian sinh bởi vector riêng  $\mathbf{u}_M$  tương ứng với trị riêng nhỏ nhất của  $\mathbf{R}_M^s$ .

Bởi vì  $\mathbf{u}_M$  trực giao với  $\mathcal{S}_u$ , ta có  $p = M - 1$  điều kiện trực giao sau:

$$(\mathbf{s}_k)^H \mathbf{u}_M = 0, \quad k = 1, \dots, M - 1. \quad (1.70)$$

Với vector  $\mathbf{u}_M$  được khai triển rõ

$$\mathbf{u}_M = [u_M(0) \quad \dots \quad u_M(M - 1)]^T$$

và vector  $\mathbf{s}_k$  đã được cho trong (1.56), các điều kiện trực giao (1.70) trở thành hệ  $M - 1$  phương trình

$$\sum_{l=0}^{M-1} u_M(l) e^{-jl\omega_k} = 0, \quad k = 1, \dots, M - 1. \quad (1.71)$$

Đặt đa thức theo biến  $z$

$$U_M(z) = \sum_{l=0}^{M-1} u_M(l) z^l, \quad z = e^{-j\omega}$$

thì phương trình đa thức

$$U_M(z) = 0 \quad (1.72)$$

sẽ có  $M - 1$  nghiệm  $z_k = e^{-j\omega_k}$ , với  $\omega_1, \dots, \omega_{M-1}$  chính là tần số góc của các thành phần điều hòa của tín hiệu.

Sau khi giải phương trình (1.72) và tìm ra  $\omega_1, \dots, \omega_{M-1}$  của các thành phần điều hòa của tín hiệu  $s(n)$ , ta cần tính công suất  $P_1, \dots, P_p$  tương ứng của các thành phần này, với  $P_k = a_k^2$ . Nhắc lại kết quả (1.54) và mô hình (1.60):

$$\mathbf{R}_M^x = \mathbf{R}_M^s + \sigma_w^2 \mathbf{I}_M, \quad (1.54)$$

$$\mathbf{R}_M^s = \mathbf{S} \mathbf{P} \mathbf{S}^H, \quad (1.60)$$

trong đó đường chéo của  $\mathbf{P}$  chứa các công suất  $P_k$ , theo (1.59). Ta suy ra

$$\mathbf{S} \mathbf{P} \mathbf{S}^H = \mathbf{R}_M^x - \sigma_w^2 \mathbf{I}_M.$$

Nhân hai phía của mỗi vế với  $\mathbf{S}^H$  và  $\mathbf{S}$ , ta được

$$\mathbf{S}^H \mathbf{S} \mathbf{P} \mathbf{S}^H \mathbf{S} = \mathbf{S}^H (\mathbf{R}_M^x - \sigma_w^2 \mathbf{I}_M) \mathbf{S}.$$

Cuối cùng, ta tính được ma trận công suất  $\mathbf{P}$ :

$$\mathbf{P} = (\mathbf{S}^H \mathbf{S})^{-1} \mathbf{S}^H (\mathbf{R}_M^x - \sigma_w^2 \mathbf{I}_M) \mathbf{S} (\mathbf{S}^H \mathbf{S})^{-1}. \quad (1.73)$$

Thay vì sử dụng (1.73) để tính công suất, ta có thể đưa ra cách tính đơn giản hơn. Dựa trên cấu trúc của  $\mathbf{R}_M^x$ , với  $p$  trị riêng lớn nhất  $\lambda_1 > \lambda_2 > \dots > \lambda_p$  tương ứng với  $p$  vector riêng  $\mathbf{u}_1, \dots, \mathbf{u}_p$  đầu của ma trận này, ta có

$$\begin{aligned} \mathbf{u}_m^H (\mathbf{R}_M^x \mathbf{u}_m) &= \mathbf{u}_m^H (\lambda_m \mathbf{u}_m) \\ &= \lambda_m \mathbf{u}_m^H \mathbf{u}_m \\ &= \lambda_m, \quad m = 1, \dots, p. \end{aligned}$$

Ta đã biết

$$\mathbf{R}_M^x = \sum_{k=1}^p P_k \mathbf{s}_k \mathbf{s}_k^H + \sigma_w^2 \mathbf{I}_M.$$

Kết hợp hai kết quả trên cho ta

$$\mathbf{u}_m^H \sum_{k=1}^p P_k \mathbf{s}_k \mathbf{s}_k^H \mathbf{u}_m + \mathbf{u}_m^H \sigma_w^2 \mathbf{I}_M \mathbf{u}_m = \lambda_m, \quad m = 1, \dots, p.$$



Cuối cùng, ta có được hệ phương trình

$$\sum_{k=1}^p P_k |\mathbf{s}_k^H \mathbf{u}_m|^2 = \lambda_m - \sigma_w^2, \quad m = 1, \dots, p \quad (1.74)$$

và dùng hệ phương trình này để giải cho  $P_1, \dots, P_p$ . Chú ý là  $\sigma_w^2$  là trị riêng thứ  $M$  của ma trận  $\mathbf{R}_M^x$ . Cố nhiên, trong thực tiễn, ta không có  $\mathbf{R}_M^x$  mà ta cần ước lượng nó dựa vào các dữ liệu nhận được.

### 1.2.3 Thuật toán Pisarenko

Cho một tập dữ liệu  $\{x(n)\}$ ,  $n = 0, \dots, L_x - 1$  và được biết trước (giả thiết) là  $\{x(n)\}$  chứa  $p$  thành phần điều hòa phức có dạng  $a_k e^{j\omega_k}$ ,  $k = 1, 2, \dots, p$ .

Thuật toán Pisarenko có ba bước như sau:

#### Bước 1

Dùng bất cứ phương pháp nào để ước lượng ma trận tự tương quan  $\mathbf{R}_M^x$  bậc  $M = p + 1$  của  $\{x(n)\}$ .

#### Bước 2

Tính  $M$  trị riêng  $\lambda_1 > \lambda_2 > \dots > \lambda_p > \lambda_{p+1} \equiv \lambda_M$  của  $\mathbf{R}_M^x$  và xác định công suất của nhiễu là  $\sigma_w^2 = \lambda_M$ , là trị riêng nhỏ nhất  $\lambda_{\min}$ . Tính vector riêng  $\mathbf{u}_M$  chuẩn hóa tương ứng với  $\lambda_M$ :

$$\mathbf{u}_M = [u_M(0) \ \dots \ u_M(M-1)]^T. \quad (1.75)$$

Góc của  $p = M - 1$  nghiệm của phương trình

$$\sum_{l=0}^{M-1} u_M(l) z^{-l} = 0, \quad z = e^{j\omega},$$

là giá trị của  $p$  tần số  $\omega_1, \dots, \omega_p$ . Từ đó, phương pháp **phân tích điều hòa Pisarenko** (Pisarenko harmonic decomposition) xây dựng hàm sau trong miền tần số:

$$\hat{p}_{\text{PHD}}(\omega) = \frac{1}{|\mathbf{s}^H(\omega) \mathbf{u}_M|^2}, \quad \omega \in [-\pi, \pi], \quad (1.76)$$

với

$$\mathbf{s}(\omega) = [1 \quad e^{j\omega} \quad \dots \quad e^{j(M-1)\omega}]^T.$$

Theo (1.70), tại  $\omega_k$ ,  $k = 1, \dots, p$ , ta có  $\mathbf{s}^H(\omega_k)\mathbf{u}_M = 0$ , mà trong thực tế thì  $\mathbf{s}^H(\omega_k)\mathbf{u}_M \approx 0$ , nên hàm  $\hat{p}_{\text{PHD}}(\omega)$  sẽ chứa  $p$  đỉnh tại các tần số này. Hàm này được gọi là **giả phổ** (pseudo-spectrum) của  $\{x(n)\}$  vì không phải là phổ thật của  $x(n)$  mà là phổ dạng khác nhưng chứa các đỉnh nằm tại đúng các tần số của  $x(n)$ .

### Bước 3

Giải hệ phương trình (1.74) để có được ước lượng công suất  $P_1, \dots, P_p$  của  $p$  thành phần điều hòa có các tần số  $\omega_1, \dots, \omega_p$  tương ứng.

Phương pháp Pisarenko rất hay về mặt toán học, nhưng không thích hợp trong thực tiễn vì hai lý do: (i) ta cần biết số thành phần  $p$  của tín hiệu điều hòa; (ii) khá nhạy cảm đối với nhiễu, tức là đối với độ chính xác của ước lượng  $\mathbf{R}_M^x$ . Ngoài ra, tính trị riêng và vector riêng đòi hỏi độ phức tạp cao, ta cần xây dựng một thuật toán như Levinson-Durbin để giải quyết vấn đề. Pisarenko công bố thuật toán của mình năm 1973 [20].

## 1.3 Thuật toán MUSIC

Ta đã thấy thuật toán Pisarenko là phương pháp phân tích điều hòa áp dụng vào ma trận bậc  $M$  (với  $M = p + 1$ ) có được bằng ước lượng dựa vào dữ liệu  $\{x(n)\}$ . Pisarenko đưa ra hai ràng buộc về mô hình của tín hiệu  $\{x(n)\}$ .

Thứ nhất là,  $x(n)$  là tín hiệu điều hòa chứa  $p$  tần số phức có dạng

$$\begin{aligned} x(n) &= s(n) + w(n) \\ &= \sum_{k=1}^p a_k e^{j(n\omega_k + \varphi_k)} + w(n), \end{aligned} \quad (1.77)$$

trong đó  $a_k$  và  $\omega_k$  là hằng số mà ta chưa biết (ẩn số),  $\varphi_k$  là pha gốc có phân phối đều  $\mathcal{U}(0, 2\pi)$  và độc lập với nhau.

Thứ hai là,  $w(n)$  là nhiễu trắng có công suất là  $\sigma_w^2$ , cũng là một ẩn số. Pisarenko đã chứng minh được không gian tín hiệu quan sát có thể tách thành hai không gian trực giao  $\mathcal{S}$  và  $\mathcal{N}$ , trong đó  $\mathcal{S}$  là không gian tín hiệu và  $\mathcal{N}$  là không gian nhiễu. Với  $M = p + 1$ ,  $\mathcal{N}$  chỉ có một chiều, xác định bởi vector riêng chuẩn hóa  $\mathbf{u}_M$  của  $\mathbf{R}_M^x$  mà tương ứng với trị riêng nhỏ nhất.

Đến năm 1979, hai nhóm nghiên cứu độc lập với nhau, là nhóm của Schmidt [26] và nhóm của Bienvenu [1–3], đã mở rộng thành công phương pháp Pisarenko. Phương pháp mở rộng này được Schmidt gọi là phương pháp **phân loại đa tín hiệu** (multiple signal classification), thường gọi tắt là phương pháp **MUSIC**. Nội dung chính của MUSIC đã được trình bày ngay từ phần mở đầu của Mục 1.2 mà ta tóm tắt lại như sau đây.

Cho tín hiệu ngẫu nhiên có chứa  $p$  thành phần điều hòa như trong công thức (1.77), cùng với các giả thiết liên quan. Với các giả thiết này, ta suy ra

$$\mathbf{R}_M^x = \mathbf{R}_M^s + \sigma_w^2 \mathbf{I}_M, \quad (1.78)$$

trong đó  $\mathbf{R}_M^x$  và  $\mathbf{R}_M^s$  là các ma trận tự tương quan của  $x(n)$  và  $s(n)$ .

Ta đã chứng minh  $\mathbf{R}_M^x$  có  $M$  giá trị riêng  $\lambda_1, \dots, \lambda_M$ , sắp theo thứ tự

$$\lambda_1 > \lambda_2 > \dots > \lambda_p > \lambda_{p+1} = \lambda_{p+2} = \dots = \lambda_M.$$

Mỗi trị riêng  $\lambda_i$  tương ứng một vector riêng  $\mathbf{u}_i$  được chọn là trực chuẩn. Cấu trúc đặc biệt của ma trận tự tương quan  $\mathbf{R}_M^x$  cho thấy không gian tín hiệu quan sát được có thể phân tích thành hai không gian, không gian tín hiệu  $\mathcal{S}$  và không gian nhiễu  $\mathcal{N}$ , được sinh ra như sau:

$$\mathcal{S} = \{\mathbf{u}_1, \dots, \mathbf{u}_p\}, \quad \mathcal{N} = \{\mathbf{u}_{p+1}, \dots, \mathbf{u}_M\}.$$

Tính chất quan trọng là  $\mathcal{S}$  và  $\mathcal{N}$  trực giao với nhau.

Ta có thể ước lượng công suất nhiễu  $\sigma_w^2$  bằng trung bình của các trị riêng  $\lambda_{p+1}, \dots, \lambda_M$ :

$$\hat{\sigma}_w^2 = \frac{1}{M - p} \sum_{i=p+1}^M \lambda_i. \quad (1.79)$$

Ước lượng tần số phức tạp hơn. Bởi vì  $\mathcal{S}$  và  $\mathcal{N}$  trực giao với nhau nên, với  $\mathbf{s}(\omega) \in \mathcal{S}$  và  $\mathbf{u}_i \in \mathcal{N}$ ,  $\mathbf{u}_i = [u_i(0) \cdots u_i(M-1)]^T$ , ta có  $(M-p)$  điều kiện trực giao sau:

$$\mathbf{s}^H(\omega)\mathbf{u}_i = 0, \quad i = p+1, \dots, M,$$

hay khai triển ra

$$\sum_{l=0}^{M-1} u_i(l)e^{-jl\omega} = 0, \quad i = p+1, \dots, M,$$

là  $(M-p)$  đa thức theo  $e^{-j\omega}$ . Đặt

$$U_i(z) = \sum_{l=0}^{M-1} u_i(l)z^l, \quad z = e^{j\omega}. \quad (1.80)$$

Đa thức  $U_i(z)$  có bậc  $(M-1)$  nên phương trình đa thức  $U_i(z) = 0$  sẽ có  $(M-1)$  nghiệm. Theo lý thuyết, các nghiệm này phải nằm trên vòng tròn đơn vị, vì  $z = e^{j\omega}$ . Nhưng trong thực tế  $\mathbf{R}_M^x$  chỉ là một ước lượng chứ không phải ma trận lý thuyết chính xác. Vì vậy, trong thực tế,  $U_i(e^{j\omega})$  sẽ có giá trị khá nhỏ lúc  $\omega$  nằm gần  $\omega_1, \dots, \omega_{M-1}$ . Như thế, một đại lượng  $V_i(e^{j\omega})$ , được gọi là **phổ riêng** (eigenspectrum), được định nghĩa bởi

$$V_i(e^{j\omega}) = \frac{1}{\left| \sum_{l=0}^{M-1} u_i(l)e^{-jl\omega} \right|^2}, \quad i = p+1, \dots, M, \quad (1.81)$$

sẽ chứa  $p$  đỉnh chung quanh các tần số  $\omega_1, \dots, \omega_p$ . Còn  $(M-p-1)$  nghiệm còn lại có thể nằm bất cứ ở đâu. Nếu ta không may mắn, thì có thể có một vài nghiệm này nằm gần vòng tròn đơn vị và  $V_i(e^{j\omega})$  lại có thêm những đỉnh phụ không thuộc phổ của tín hiệu  $s(n)$ . Như thế, nếu ta chỉ dùng một vector nhiều  $\mathbf{u}_i$  để ước lượng đỉnh phổ của  $s(n)$  thì sẽ gặp nhiều khó khăn vì sai số sẽ lớn. Phương pháp MUSIC đề nghị dùng trị trung bình của  $\mathbf{s}^H(e^{j\omega})\mathbf{u}_i$  để có giả phổ sau đây:

$$\hat{p}_{\text{MUSIC}}(\omega) = \frac{1}{\sum_{i=p+1}^M |\mathbf{s}^H(\omega)\mathbf{u}_i|^2}. \quad (1.82)$$

Từ đó, ta ước lượng được các tần số tương ứng với  $p$  đỉnh lớn nhất của  $\hat{p}_{\text{MUSIC}}(\omega)$ .

Sau khi đã ước lượng  $p$  tần số  $\omega_k$ ,  $k = 1, \dots, p$ , ta có thể suy ra  $p$  công suất  $P_1, \dots, P_p$  bằng cách giải hệ phương trình (1.74), đã được lý giải trong phương pháp Pisarenko và được nhắc lại ở đây như sau:

$$\sum_{k=1}^p P_k |\mathbf{s}_k^H \mathbf{u}_m|^2 = \lambda_m - \sigma_w^2, \quad m = 1, \dots, p. \quad (1.74)$$

## 1.4 Thuật toán ESPRIT

Thuật toán **ESPRIT** là gọi tắt của thuật toán **ước lượng tham số thông qua các kỹ thuật bất biến xoay** (estimation parameters via rotational invariance techniques). Thuật toán này, cũng giống như thuật toán MUSIC, giúp ta ước lượng  $p$  tần số có dạng phức của một tín hiệu  $\{x(n)\}$  có chứa nhiều cộng:

$$x(n) = s(n) + w(n),$$

trong đó  $s(n)$  có dạng

$$s(n) = \sum_{k=1}^p a_k e^{j(n\omega_k + \varphi_k)}$$

và nhiễu  $w(n)$  là trắng, có công suất là  $\sigma_w^2$ .

Thuật toán ESPRIT lợi dụng mối liên hệ giữa vector  $\mathbf{s}_N(n) = [s(n), \dots, s(n+N-1)]$  và vector  $\mathbf{s}_N(n+1)$  như sau:

$$\mathbf{s}_N(n) = \mathbf{S} \mathbf{a} \quad (1.83a)$$

$$\mathbf{s}_N(n+1) = \mathbf{S} \Phi \mathbf{a}, \quad (1.83b)$$

trong đó

$$\mathbf{S} = \begin{bmatrix} e^{jn\omega_1} & e^{jn\omega_2} & \dots & e^{jn\omega_p} \\ e^{j(n+1)\omega_1} & e^{j(n+1)\omega_2} & \dots & e^{j(n+1)\omega_p} \\ \vdots & \vdots & \ddots & \vdots \\ e^{j(n+N-1)\omega_1} & e^{j(n+N-1)\omega_2} & \dots & e^{j(n+N-1)\omega_p} \end{bmatrix}$$

$$\mathbf{a} = \begin{bmatrix} a_1 e^{j\varphi_1} \\ a_2 e^{j\varphi_2} \\ \vdots \\ a_p e^{j\varphi_p} \end{bmatrix}$$

$$\Phi = \begin{bmatrix} e^{j\omega_1} & 0 & \cdots & 0 \\ 0 & e^{j\omega_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{j\omega_p} \end{bmatrix}$$

Ma trận  $\Phi$  là ma trận unita (unitary matrix), tức thỏa mãn  $\Phi^{-1} = \Phi^H$ . Thông thường,  $\Phi$  được gọi là ma trận quay (rotation matrix) bởi vì  $\Phi$  nối kết  $\mathbf{s}_N(n)$  với  $\mathbf{s}_N(n+1)$  như trong (1.83).

Như ta đã thấy, ma trận tự tương quan của tín hiệu quan sát  $\{x(n)\}$  chứa nhiều công  $w(n)$  có dạng

$$\begin{aligned} \mathbf{R}_{\mathbf{x}_N(n)} &= \mathbb{E}\{\mathbf{x}_N(n)\mathbf{x}_N^H(n)\} \\ &= \mathbf{S}\mathbf{P}\mathbf{S}^H + \sigma_w^2 \mathbf{I}, \end{aligned}$$

trong đó  $\mathbf{P}$  là ma trận chéo chứa các công suất  $P_k = a_k^2$ :

$$\mathbf{P} = \begin{bmatrix} a_1^2 & 0 & \cdots & 0 \\ 0 & a_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_p^2 \end{bmatrix}.$$

Từ đó

$$\mathbf{R}_{\mathbf{x}_N(n), \mathbf{x}_N(n+1)} = \mathbf{S}\mathbf{P}\Phi^H \mathbf{S}^H + \mathbf{R}_{\mathbf{w}_N(n), \mathbf{w}_N(n+1)},$$

trong đó

$$\begin{aligned} \mathbf{R}_{\mathbf{x}_N(n), \mathbf{x}_N(n+1)} &= \\ & \begin{bmatrix} R_x(1) & R_x(2) & R_x(3) & \cdots & R_x(N) \\ R_x(0) & R_x(1) & R_x(2) & \cdots & R_x(N-1) \\ R_x(1) & R_x(0) & R_x(1) & \cdots & \cdots \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ R_x(N-2) & R_x(N-3) & R_x(N-4) & \cdots & R_x(1) \end{bmatrix}, \end{aligned}$$

$$\mathbf{R}_{\mathbf{w}_N(n), \mathbf{w}_N(n+1)} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ \sigma_w^2 & 0 & \cdots & 0 & 0 \\ 0 & \sigma_w^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_w^2 & 0 \end{bmatrix}.$$

Với các biểu thức này, ta suy ra

$$\begin{aligned} \mathbf{R}_{\mathbf{s}_N(n), \mathbf{s}_N(n+1)} &= \mathbf{R}_{\mathbf{x}_N(n), \mathbf{x}_N(n)} - \mathbf{R}_{\mathbf{w}_N(n), \mathbf{w}_N(n)} \\ &= \mathbf{S} \mathbf{P} \mathbf{\Phi}^H \mathbf{S}^H \end{aligned}$$

và từ đó

$$\mathbf{R}_{\mathbf{s}_N(n), \mathbf{s}_N(n)} - \lambda \mathbf{R}_{\mathbf{s}_N(n), \mathbf{s}_N(n+1)} = \mathbf{S} \mathbf{P} (\mathbf{I} - \lambda \mathbf{\Phi}^H) \mathbf{S}^H. \quad (1.84)$$

Từ (1.84), ta thấy  $\lambda$  mà làm triệt tiêu đại lượng  $|\mathbf{R}_{\mathbf{s}_N(n)} - \lambda \mathbf{R}_{\mathbf{s}_N(n), \mathbf{s}_N(n+1)}|$  chính là  $\lambda$  làm triệt tiêu  $|\mathbf{I} - \lambda \mathbf{\Phi}^H|$ , tức là

$$\lambda_k = e^{j\omega_k}, k = 1, \dots, p \quad (1.85)$$

Như vậy,  $\lambda_k$  là nghiệm của

$$|\mathbf{R}_{\mathbf{s}_N(n), \mathbf{s}_N(n)} - \lambda \mathbf{R}_{\mathbf{s}_N(n), \mathbf{s}_N(n+1)}| = 0. \quad (1.86)$$

Nghiệm của phương trình này được gọi là trị riêng mở rộng, được định nghĩa như sau:

$$\mathbf{A} \mathbf{u} = \lambda \mathbf{B} \mathbf{u}, \quad (1.87)$$

$$|\mathbf{A} - \lambda \mathbf{B}| = 0, \quad (1.88)$$

Từ đây ta suy ra thuật toán ESPRIT như sau:

*Bước 1*

Từ  $\{x(n)\}$ , ta ước lượng  $\hat{\mathbf{R}}_{\mathbf{x}_N}$  và  $\hat{\mathbf{R}}_{\mathbf{x}_N(n), \mathbf{x}_N(n+1)}$ .

*Bước 2*

Tìm trị riêng mở rộng của

$$|\mathbf{R}_{\mathbf{x}_N(n), \mathbf{x}_N(n)} - \lambda \mathbf{R}_{\mathbf{x}_N(n), \mathbf{x}_N(n+1)}| = 0.$$

*Bước 3*

Chọn  $p$  trị riêng  $\lambda_k$  mở rộng gần vòng tròn đơn vị nhất.

*Bước 4*

Ước lượng  $p$  tần số

$$\omega_k = \angle \lambda_k, \quad k = 1, \dots, p. \quad (1.89)$$

Giáo trình chỉ giới thiệu nhanh thuật toán ESPRIT như trên, mà không bàn đến các lợi điểm của ESPRIT so với MUSIC. Người đọc có thể tham khảo thêm trong công trình của Roy và cộng sự [22]. Giáo trình tập trung cho thực thi các thuật toán Pisarenko và MUSIC, được trình bày trong phần tiếp theo, để người đọc hiểu cận kề hai cách tiếp cận chính này.

## 1.5 Thực thi thuật toán Pisarenko và MUSIC

### 1.5.1 Chương trình lệnh so sánh hai thuật toán Pisarenko và MUSIC

Như ta vừa thấy, thuật toán Pisarenko là trường hợp đặc biệt của thuật toán MUSIC. Do đó, sau đây ta trình bày chương trình lệnh `pisarenko_music_thinghiem.m` cho phép ta thực hiện hoặc thuật toán Pisarenko hoặc thuật toán MUSIC.

**Chương trình 1.1** `pisarenko_music.m`

```
%% THI NGHIỆM VỚI HAI THUẬT TOÁN PIRASENKO VA MUSIC
%
% Thực hiện các thí nghiệm ước lượng p tần số của
% tín hiệu bằng các phương pháp Pisarenko và MUSIC.
% Nếu gia pho PsSp chưa ít hơn p đỉnh, thì chương
% trình này chỉ vẽ đồ thị của gia pho PsSp đã tính
% được mà không ước lượng các tần số.

% Giao trình: Xu lý tín hiệu ngẫu nhiên
% NXB Đại học Quốc gia Hà Nội, 2025
% Nguyễn Linh Trung, Huỳnh Hữu Tue
% =====
%%

%% Phần I: Nhập thông tin, tính toán thông số
% --- Chọn thông số cho tín hiệu điều hòa {x(n)}:
% fnu là vectơ tần số, A là vec-tơ biên độ
%fnu=.25; A=1;
```



```

%fnu=[.15 .3]; A=[1 1];
%fnu=[.15 .2]; A=[1 1];
fnu=[.15 .155]; A=[1 1];
%fnu = [.1 .12 .15 .2 .4]; A = [1 2 1 1 1];
%fnu = [.1 .11 .15 .2 .4]; A = [1 2 3 1 1];
%fnu = [.1 .12 .15 .2 .4]; A = [2 2 3 1 5];
%fnu = [.1 .11 .13 .15 .2 .3 .32 .4]; A = [2 4 ...
3 2 1 4 2 5];
%fnu = [.1 .15 .2 .3 .4]; A = [1 2 3 1 3];
%fnu = [.1 .12 .13 .18 .3]; A = [5 5 3 1 1];
%fnu = [.1 .11 .12 .13 .15]; A=[1 1 1 1 1];

% --- Chọn san 2 thông số có ảnh hưởng lớn nhất đến
% chất lượng thuật toán: tỷ lệ tín trên nhiễu SNRdB
% và chiều dài Lx của tín hiệu
SNRdB = 10; % 5, 10, 20, 30
Lx = 1000; % 100, 200, 500, 1000, 2000, 4000, ...

% --- Tính ra các thông số quan trọng của tín hiệu:
% số tần số p và công suất của nhiễu
p = length(fnu);
P = A.^2/2;
sigma2 = sum(P)*10^(-.1*SNRdB);
sigma = sqrt(sigma2);

%% Phần II: Thực hiện K thí nghiệm
% --- Chuẩn bị thí nghiệm:
% Chọn phương pháp phân tích: M = p+1 cho
% Pisarenko, M = max(5*p,30) cho MUSIC (Ta có thể
% tăng M nếu cần thiết) và một số chuẩn bị khác.
M = max(5*p,30); % MUSIC
%M = p+1; % Pisarenko
figure
hold
nu_hat = [];
% Giá phổ Pspp được tính tại các tần số chuẩn hóa
% nu(k), biến thiên từ 0 đến 0.5
nu = (0:.001:.5)';
% Chọn số lần thí nghiệm K
K = 50; % 1, 50
j = sqrt(-1);

% --- Bắt đầu thí nghiệm:
for kk = 1:K
    % Tạo tín hiệu {x(n)} cho thí nghiệm lần thứ k
    xsach = ...
        sum([diag(A)*exp(2*j*pi*(fnu'*(0:Lx-1) + ...
            rand(length(A),1)))]);
    x = xsach + sigma*(randn(1,Lx) + ...
        randn(1,Lx)*j)/sqrt(2);
    x = x(:);

    % Ước lượng ma trận tự tương quan RMx của tín
    % hiệu dữ liệu {x(n)} bằng phương pháp hiệp
    % phương sai

```

```

RMx = zeros(M,M);
for kr = M:Lx
    RMx = RMx+x(kr:-1:kr-M+1)*x(kr:-1:kr-M+1)';
end
RMx = RMx/Lx;

% Tính gia pho PsSp của {x(n)} bằng phương pháp
% Pisarenko-MUSIC và vẽ chồng lên nhau do thi
% của gia pho cho tung thi nghiệm. De tránh
% nhưng nham lan luc su dung FFT trong luc lap
% trình, o day ta tinh truc tiep bieu thuc giai
% tích cua gia pho PsSp(nu) tại các tần số
% nu(k) đã được chọn trước trong phân Chuan bi
[U,vD,V] = svd(RMx);
M = size(RMx,1);
S = exp(2*pi*j*nu*(0:M-1));
SU = S*U(:,p+1:M);
PsSp = 1./sum([abs(SU)'.^2;...
    zeros(1,length(nu))]);
PSdB = 10*log10(PsSp);

% Vẽ đồ thị của gia pho PSdB của thí nghiệm
% thu k
plot(nu,PSdB);

% Ước lượng p tần số của {x(n)}:
% Tìm đỉnh Pm của PsSp và vị trí k tần số tương
% ứng
[Pm,k] = findpeaks(PsSp(:));

% Tạo danh sách fPm các tần số tương ứng với
% các đỉnh Pm
fPm = nu(k);

% Sắp lại các đỉnh từ nhỏ đến lớn, và tính vị
% trí ik trước đây của các đỉnh Pm vừa được
% sắp đặt
[PSm ik] = sort(Pm);

% Tìm p vị trí nf trong danh sách ik tương ứng
% với p đỉnh cao nhất
nf = ik(length(ik)-p+1:end);
nu_hat = [nu_hat sort(fPm(nf))];
end % kết thúc K thí nghiệm

hold

%% Phần III. Kết thúc phần trình bày đồ thị
ylabel('Gia pho (dB)')
xlabel('Tần số chuẩn hóa \nu')
axis([0 .5 -20 60])
grid

if M ~= p+1
    if K ~= 1

```

```

        title([num2str(K), ' gia pho MUSIC của ...
              tin hieu dieu hoa co ', num2str(p), ' ...
              dinh'])
        ['M = ', num2str(M), '; Lx = ...
         ', num2str(Lx), '; SNRdB = ...
         ', num2str(SNRdB)]
        ['\nu = ', ' ', num2str(fnu), ' ']]})
    else
        title(['Gia pho MUSIC của tin hieu dieu ...
              hoa co ', num2str(p), ' dinh'])
        ['M = ', num2str(M), '; Lx = ...
         ', num2str(Lx), '; SNRdB = ...
         ', num2str(SNRdB)]
        ['\nu = ', ' ', num2str(fnu), ' ']]})
    end
else
    if K ≠ 1
        title([num2str(K), ' gia pho Pisarenko ...
              của tin hieu dieu hoa co ...
              ', num2str(p), ' dinh'])
        ['M = ', num2str(M), '; Lx = ...
         ', num2str(Lx), '; SNRdB = ...
         ', num2str(SNRdB)]
        ['\nu = ', ' ', num2str(fnu), ' ']]})
    else
        title(['Gia pho Pisarenko của tin hieu ...
              dieu hoa co ', num2str(p), ' dinh'])
        ['M = ', num2str(M), '; Lx = ...
         ', num2str(Lx), '; SNRdB = ...
         ', num2str(SNRdB)]
        ['\nu = ', ' ', num2str(fnu), ' ']]})
    end
end
if K ≠ 1
    nu_ave = [mean(nu_hat')]';
    std_nu = [std(nu_hat')]';
    [fnu', nu_ave + std_nu, nu_ave, nu_ave - std_nu]
else
    nu_ave=nu_hat;
    [fnu', nu_ave]
end
end

```

Các thông số đầu vào là vector **fnu** chứa các tần số của tín hiệu và vector **A** chứa các biên độ. Thông số **SNRdB** là tỷ lệ tín trên nhiễu tính theo đơn vị dB, thông số **Lx** là chiều dài của tín hiệu  $\{x(n)\}$ . Tỷ lệ tín trên nhiễu được cho bởi

$$\text{SNR} = \frac{P_s}{P_N}, \quad (1.90)$$

trong đó  $P_s$  là công suất của tín hiệu điều hòa,  $P_N$  là công suất của nhiễu quan sát  $w(n)$ . Ta có

$$P_s = \frac{1}{2} \sum_{k=1}^p |a_k|^2, \quad (1.91)$$

Nhiều là Gauss trắng có phương sai  $\sigma_w^2$  và trị trung bình triệt tiêu, nên ta có

$$P_N = \sigma_w^2. \quad (1.92)$$

Như thế

$$\sigma_w^2 = P_s \times 10^{-\text{SNR}_{\text{dB}}/10}. \quad (1.93)$$

Ngoài ra, còn một thông số đầu vào nữa là  $M$ , là số chiều  $M$  của không gian tín hiệu quan sát. Ta chọn  $M = p + 1$  nếu ta muốn dùng thuật toán Pisarenko, chọn  $M > p + 1$  cho thuật toán MUSIC.

Đầu ra của chương trình là giả phổ của  $\{x(n)\}$ , PsSp, và  $p$  ước lượng của tần số fnu. PsSp được trình bày dưới dạng đồ thị và ước lượng tần số dưới dạng ma trận gồm bốn cột: cột thứ nhất là trị số chính xác của vector tần số fnu, cột thứ hai, ba và bốn lần lượt là ước lượng fnu\_hat +  $\Delta$ , fnu\_hat và fnu\_hat -  $\Delta$ , trong đó fnu\_hat là ước lượng của tần số và  $\Delta$  là độ lệch chuẩn tính trên  $k$  mẫu của thí nghiệm. Một thí nghiệm là một lần mô phỏng bao gồm cả tính toán ước lượng.

Sau đây, ta sẽ thực hiện một số thí nghiệm để đánh giá chất lượng của hai thuật toán Pisarenko và MUSIC, nhất là để thấy được ảnh hưởng của các thông số SNR,  $L_x$  và  $M$ .

Đối với các thí nghiệm liên quan đến Pisarenko, ta chỉnh sửa chương trình lệnh ở trên để thuận lợi cho việc hiển thị kết quả; cụ thể là sử dụng chương trình lệnh sau đây:

#### Chương trình 1.2 pisarenko\_Lx.m

```
%% THI NGHIỆM THUAT TOAN PISARENKO
%
% Thuc hien cac thi nghiem uoc luong p tan so cua
% tin hieu bang cac phuong phap Pisarenko. Neu gia
% pho PsSp chua it hon p dinh, thi chuong trinh nay
% chi ve do thi cua gia pho PsSp da tinh duoc ma
% khong uoc luong cac tan so.

% Giao trinh: Xu ly tin hieu ngau nhien
% NXB: Dai hoc Quoc gia Ha Noi, 2025
% Nguyen Linh Trung, Huynh Huu Tue
% =====
```

```

%%

%% Phan I: Nhap thong tin, tinh toan thong so
% --- % Chon thong so cho tin hieu dieu hoa {x(n)}:
% fnu la vecto tan so, A la vec-to bien do
%fnu=.25; A=1;
%fnu=[.15 .3]; A=[1 1];
%fnu=[.15 .2]; A=[1 1];
fnu=[.15 .155]; A=[1 1];

% --- Chon ty le tin tren nhieu SNRdB va chieu dai
% Lx. Day la haithong so co anh huong nhut den
% chat luong thuat toan
SNRdB=5; % 5, 10, 20, 30
%L=[50 100 200 500];
%L=[100 200 500 1000];
%L=[500 1000 2000 4000];
L=[10000 20000 40000 80000];
LL=length(L);

% --- Suy ra cac thong so quan trong cua tin hieu
% la so tan so p va cong suat cua nhieu
P = A.^2/2;
sigma2 = sum(P)*10^(-.1*SNRdB);
sigma = sqrt(sigma2);
p = length(fnu);

%% Phan II: Thuc hien K thi nghiem
% --- Chuan bi:
% Chon phuong phap phan tich: M = p+1 cho
% Pisarenko, va mot so chuan bi khac
%M = max(5*p,50); %MUSIC
M = p+1; %Pisarenko
figure;

% Gia pho PsSp duoc tinh tai cac tan so chuan hoa
% nu(k), bien thien tu 0 den 0.5
nu=(0:.001:.5)';

% So lan lam thi nghiem
K=50; % 50, 1

% --- Thuc hien thi nghiem:
nu_hat=zeros(p,K,LL);

for m=1:LL
    Lx=L(m);
    subplot(2,2,m);
    hold
    j=sqrt(-1);

    % Thi nghiem bat dau
    for k=1:K
        % Tao tin hieu {x(n)} cho thi nghiem lan
        % thu k

```

```

xsach = ...
    sum([diag(A)*exp(2*j*pi*(fnu'*(0:Lx-1) ...
        + rand(length(A),1))]; zeros(1,Lx)]);
x = xsach + sigma*(randn(1,Lx) + ...
    randn(1,Lx)*j) / sqrt(2);

% Uoc luong ma tran tu tuong quan RMx cua
% tin hieu du lieu {x(n)} bang phuong phap
% hiep phuong sai
x = x(:); % chuyen vecto ve dang cot
RMx = zeros(M,M);
for n = M:Lx
    RMx = RMx+x(n:-1:n-M+1)*x(n:-1:n-M+1)';
end
RMx = RMx/Lx;

% Tinh gia pho PsSp cua {x(n)} bang phuong
% phap Pisarenko va ve chong len nhau do
% thi cua gia pho cho tung thi nghiem. De
% tranh nhung nham lan luc su dung FFT
% trong luc lap trinh, o day ta tinh truc
% tiep bieu thuc giai tích cua gia pho
% PsSp(nu) tai cac tan so nu(k) da duoc
% chon truoc trong phan chuan bi
[U,vD,V]=svd(RMx);
M = size(RMx,1);
S = exp(2*pi*j*nu*(0:M-1));
SU=S*U(:,p+1:M);
PsSp=1./sum([abs(SU)'.^2; ...
    zeros(1,length(nu))]);
PSdB=10*log10(PsSp);

% Ve do thi cua gia pho PSdB cua thi nghiem
% thu k
plot(nu,PSdB);
end % Ket thuc tat ca K thi nghiem
hold
axis([0 .5 -10 120]);
grid
title([num2str(K),' gia pho Pisarenko cua ...
    tin hieu dieu hoa co ',num2str(p),' dinh']
    ['SNRdB = ',num2str(SNRdB),''; \nu = ...
    ',num2str(fnu)]]);
legend(['Lx = ', num2str(L(m))]);
xlabel('Tan so chuan hoa, \nu');
ylabel('Gia pho (dB)');
end; % Ket thuc tat ca 4 chieu dai

```

### 1.5.2 Thực thi thuật toán Pisarenko

Chúng ta sẽ phân tích chất lượng của thuật toán Pisarenko thông qua một số tình huống.

*Trường hợp đơn giản*

Tín hiệu  $\{x(n)\}$  chỉ chứa một tần số  $\nu_0 = 0,25$  với  $\text{SNR} = 10$  dB, ta cần  $L_x = 100$  để Pisarenko cho kết quả chấp nhận được, như trong Hình 1.2(a). Nếu nhiễu nhỏ hơn, chẳng hạn  $\text{SNR} = 5$  dB, ta cần một chiều dài lớn hơn, chẳng hạn  $L_x = 500$ , để có kết quả chấp nhận được, như trong Hình 1.2(b).

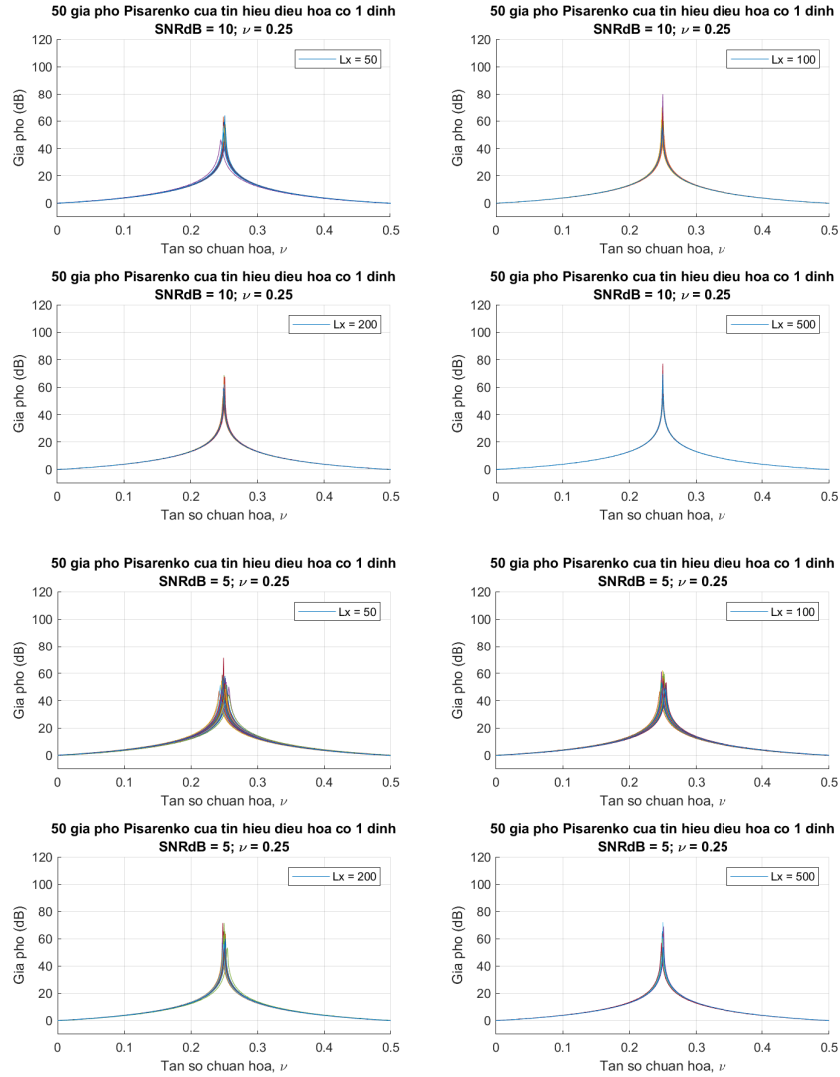
*Trường hợp nhiều tần số*

Lúc tín hiệu chứa nhiều hơn một tần số, vấn đề tách rời các đỉnh của giả phổ (tức là ước lượng các tần số) phụ thuộc vào độ phân giải phổ. Sơ bộ, ta có thể hình dung độ phân giải phổ là khoảng cách  $\Delta f$  nhỏ nhất giữa hai đỉnh kề nhau, và như thế độ phân giải phụ thuộc vào khả năng tách các đỉnh. Kết quả phụ thuộc chặt chẽ vào hai thông số là  $\text{SNR}$  và  $L_x$ . Chiều dài  $L_x$  càng lớn thì độ chính xác của ước lượng ma trận tự tương quan  $\mathbf{R}_M^x$  ( $\mathbf{R}\mathbf{M}\mathbf{x}$ ) càng lớn, giúp tăng độ phân giải phổ. Tuy nhiên, độ chính xác vẫn phụ thuộc vào  $\text{SNR}$ .

Hình 1.3 cho thấy với hai tần số 0,15 và 0,3 (tức độ phân giải là  $0,3 - 0,15 = 0,15$ ) với  $\text{SNR} = 5$  dB, ta cần  $L_x \geq 4.000$  để có kết quả chấp nhận được. Còn với  $\text{SNR} = 10$  dB, ta chỉ cần  $L_x \approx 1.000$ ; trong trường hợp này, độ phân giải tuy thô, nhưng ta vẫn cần có nhiều dữ liệu để đạt được kết quả đủ tốt.

Để thấy rõ ảnh hưởng của độ phân giải phổ, ta xét tình huống hai tần số gần nhau hơn là 0,15 và 0,2 (độ phân giải là 0,05). Trong thực tế, độ phân giải 0,05 chưa phải thật sự cao. Tuy nhiên, với  $\text{SNR} = 10$  dB, ta cần đến  $L_x \approx 40.000$  mới đạt được chất lượng chấp nhận được, như trong Hình 1.4(a). Có nhiên, nếu  $\text{SNR}$  tốt hơn, chẳng hạn  $\text{SNR} = 20$  dB, ta chỉ cần  $L_x \approx 1.000$  là tạm đủ, như trong Hình 1.4(b).

Trong trường hợp độ phân giải cao hơn nữa, chẳng hạn 0,15 và 0,155 ( $\Delta\nu = 0,005$ ), như trong Hình 1.5, ta thấy lúc  $\text{SNR} = 30$  dB, ta cần  $L_x \approx 80.000$ ; ta thấy ngay trong trường hợp này, thuật toán Pisarenko không thích hợp với nhu cầu thực tiễn.

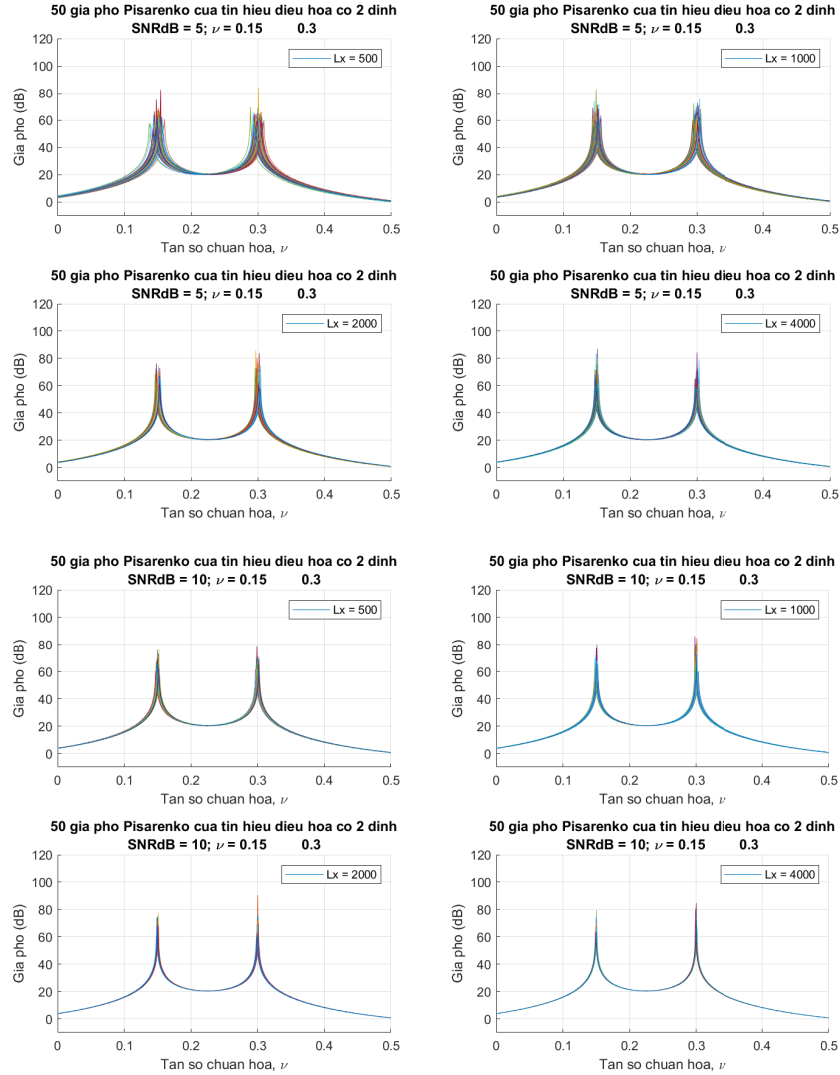


Hình 1.2 Thí nghiệm thuật toán Pisarenko cho tín hiệu điều hòa có 1 đỉnh.

### 1.5.3 Thực thi thuật toán MUSIC

Bây giờ, ta đánh giá thuật toán MUSIC thông qua một số thí nghiệm.

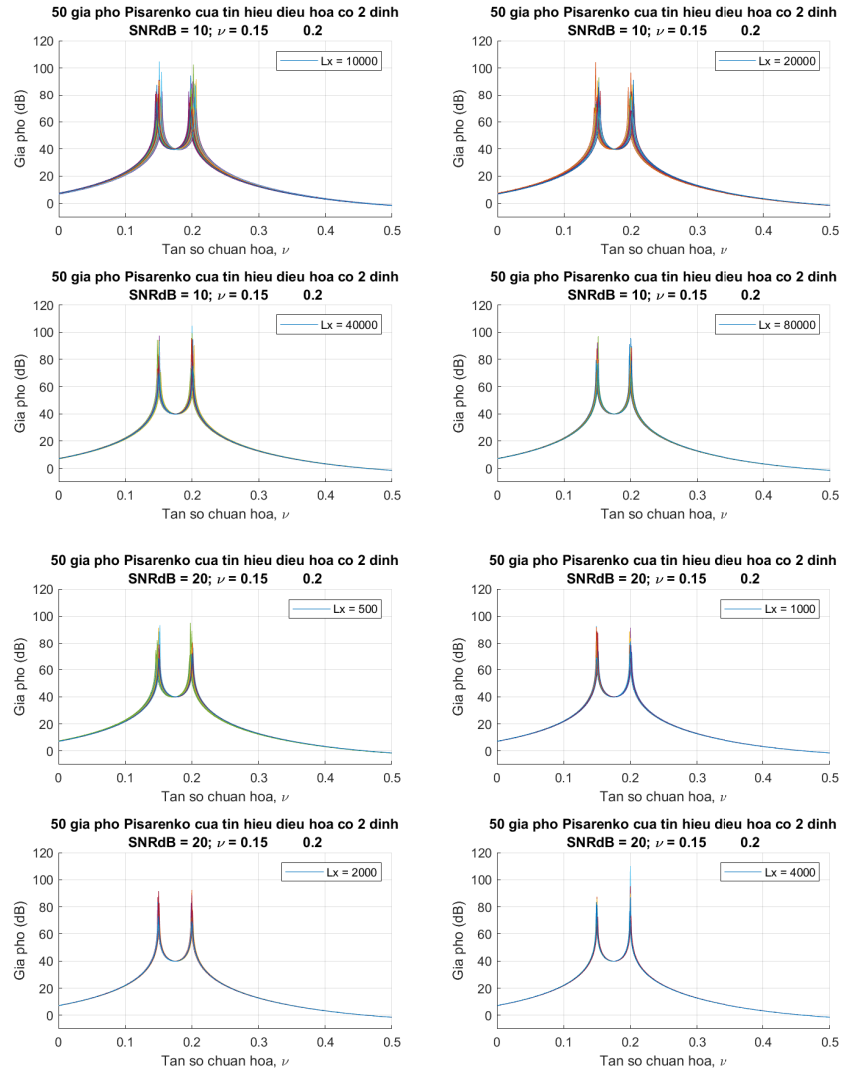




**Hình 1.3** Thí nghiệm thuật toán Pisarenko cho tín hiệu điều hòa có 2 đỉnh có độ phân giải thấp.

### Hai đỉnh với độ phân giải bình thường

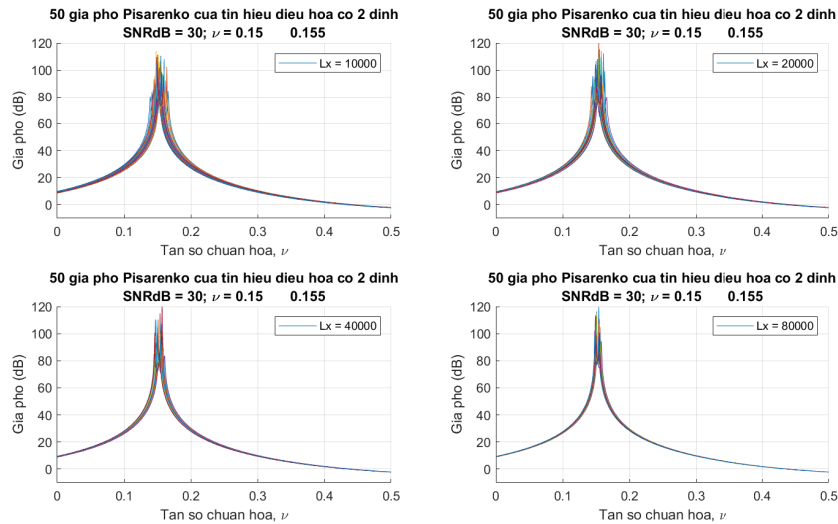
Xét trường hợp hai đỉnh có độ phân giải bình thường  $\nu_1 = 0,15$ ,  $\nu_2 = 0,2$  (độ phân giải là 0,05). Hình 1.6 cho thấy với  $L_x = 100$  và  $M = 20$ , ta chỉ cần SNR = 5 dB là đủ để có kết quả thích hợp.



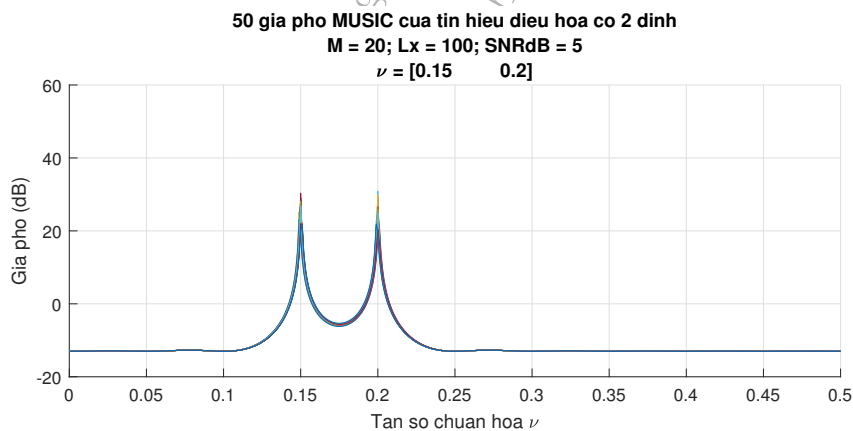
**Hình 1.4** Thí nghiệm thuật toán Pisarenko cho tín hiệu điều hòa có 2 đỉnh có độ phân giải bình thường.

#### Hai đỉnh với độ phân giải rất cao

Xét trường hợp với độ phân giải rất cao,  $\nu_1 = 0,15$  và  $\nu_2 = 0,155$  (độ phân giải là 0,005), ta hình dung MUSIC khó tách được hai đỉnh. Tuy nhiên, với SNR = 10 dB, ta chọn



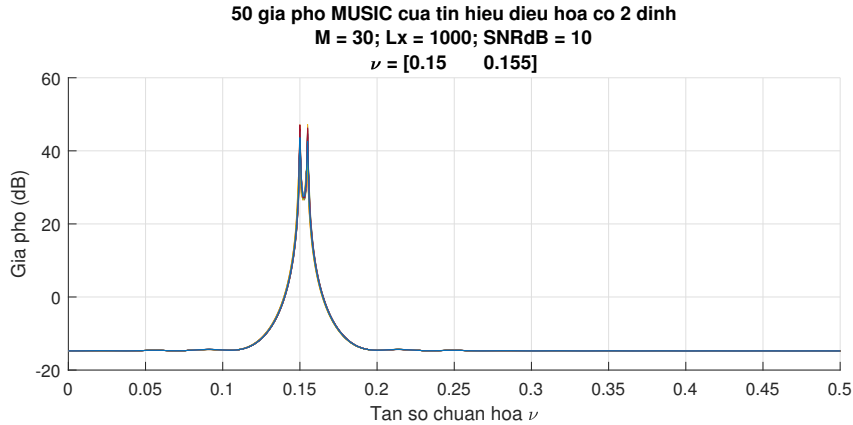
**Hình 1.5** Thí nghiệm thuật toán Pisarenko cho tín hiệu điều hòa có 2 đỉnh có độ phân giải cao.



**Hình 1.6** Thí nghiệm thuật toán MUSIC cho tín hiệu điều hòa có 2 đỉnh có độ phân giải bình thường.

$M = 30$  và  $L_x = 1000$ , thì chất lượng của giả phổ là hoàn toàn chấp nhận được (xem Hình 1.7).

Bởi vì ta có thêm thông số  $M$  cho thuật toán MUSIC, nên chất lượng của MUSIC dễ kiểm soát hơn và tốt hơn rất nhiều so với Pisarenko.



**Hình 1.7** Thí nghiệm thuật toán MUSIC cho tín hiệu điều hòa có 2 đỉnh có độ phân giải rất cao.

## 1.6 Tóm tắt chương

Ước lượng phổ công suất là một vấn đề quan trọng trong công nghiệp như trong lĩnh vực viễn thông, điều khiển, xử lý tiếng nói và ngay cả trong kinh tế tài chính nữa. Chương này trình bày một số phương pháp ước lượng phổ hiện đại, còn gọi là ước lượng tham số, hay ước lượng dựa trên mô hình.

Thuật toán chính là Burg, dựa vào mô hình AR. Ta cũng quan tâm đến các phương pháp ước lượng tần số của một tín hiệu điều hòa. Lý thuyết cơ bản được Pisarenko xây dựng năm 1973 và được mở rộng năm 1979 để có thuật toán MUSIC. Để ước lượng các tần số của một tín hiệu điều hòa, ta đã xây dựng khái niệm giả phổ hoặc phổ MUSIC, là một hàm trong miền tần số, có đỉnh nằm tại các tần số của tín hiệu. Ngoài ra, Roy và cộng sự đã xây dựng được một thuật toán có tên là ESPRIT, để ước lượng các tần số như đáp án của một phương trình phân tích trị riêng, mà không thông qua bài toán tìm kiếm đỉnh phổ.

Điều nên chú ý là ta cần nắm vững phương pháp tuần hoàn đồ và thuật toán Burg. Ngoài ra, hai thuật toán MUSIC và ESPRIT rất hữu ích cho vấn đề xác định hướng đến của tín hiệu, sẽ được xét đến trong Chương 5.

## Phụ lục Chương 1

### 1.A Toán tử dự đoán bậc $N$

Cho một tín hiệu ngẫu nhiên dừng mở rộng  $\{x(n)\}$ , ta có thể xây dựng một cấu trúc dự đoán tuyến tính, bậc  $N$  như sau.

Ở thời điểm  $n - 1$ , ta đã có  $x(n - 1), \dots, x(n - N)$  mẫu tín hiệu. Có thể dùng một tổ hợp tuyến tính của chúng để dự đoán  $x(n)$ :

$$\hat{x}(n) = \sum_{k=1}^N a_k x(n - k). \quad (1.94)$$

Sai số dự đoán là

$$\begin{aligned} e(n) &\triangleq x(n) - \hat{x}(n) \\ &= x(n) - \sum_{k=1}^N a_k x(n - k). \end{aligned} \quad (1.95)$$

Ta sẽ xác định cấu trúc của hệ thống dự đoán thế nào để tối thiểu sai số bình phương trung bình, tức có được

$$\varepsilon_N \triangleq \min_{\{a_k\}} \mathbb{E}\{e^2(n)\}. \quad (1.96)$$

Bởi vì  $\{x(n)\}$  là dừng mở rộng nên  $\varepsilon_N$  không phụ thuộc vào  $n$ , ta dùng  $\varepsilon_N$  để ký hiệu sai số bình phương trung bình tối thiểu (MMSE: minimum mean squared error) của hệ thống dự đoán tối ưu bậc  $N$ . Gọi  $\varepsilon_N$  là sai số dự đoán bình phương trung bình tối thiểu bậc  $N$ .

Để tìm  $a_1, \dots, a_N$  tối ưu, ta tính  $\mathbb{E}\{e^2(n)\}$  như sau. Trước hết, đặt

$$\mathbf{a} = [a_1, \dots, a_N]^T \quad (1.97)$$

$$\mathbf{x}_N(n) = [x(n-1), \dots, x(n-N)]^T. \quad (1.98)$$

Ta viết lại sai số

$$e(n) = x(n) - \mathbf{a}^T \mathbf{x}_N(n). \quad (1.99)$$

Từ đó khai triển  $\mathbb{E}\{e^2(n)\}$

$$\begin{aligned} \mathbb{E}\{e^2(n)\} &= \mathbb{E}\{[x(n) - \mathbf{a}^T \mathbf{x}_N(n)][x(n) - \mathbf{x}_N^T(n) \mathbf{a}]\} \\ &= \mathbb{E}\{x^2(n)\} + \mathbf{a}^T \mathbb{E}\{\mathbf{x}_N(n) \mathbf{x}_N^T(n)\} \mathbf{a} - \\ &\quad 2\mathbf{a}^T \mathbb{E}\{x(n) \mathbf{x}_N(n)\}. \end{aligned} \quad (1.100)$$

Tín hiệu  $x(n)$  là dừng mở rộng nên có tự tương quan  $R(n) = \mathbb{E}\{X(k)X(k-n)\}$  với mọi  $k$ . Ta đặt

$$\begin{aligned} \mathbf{R}_N &= \mathbb{E}\{\mathbf{x}_N(n) \mathbf{x}_N^T(n)\} \\ &= \begin{bmatrix} R(0) & R(1) & \dots & R(N-1) \\ R(1) & R(0) & \dots & R(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ R(N-1) & R(N-2) & \dots & R(0) \end{bmatrix}, \end{aligned} \quad (1.101)$$

$$\begin{aligned} \mathbf{r}_N &= \mathbb{E}\{x(n) \mathbf{x}_N(n)\} \\ &= [R(1), R(2), \dots, R(N)]^T. \end{aligned} \quad (1.102)$$

Ta suy ra

$$\mathbb{E}\{e^2(n)\} = R(0) + \mathbf{a}^T \mathbf{R}_N \mathbf{a} - 2\mathbf{a}^T \mathbf{r}_N. \quad (1.103)$$

Đáp án  $\mathbf{a}$  tối ưu là lời giải của hệ  $N$  phương trình

$$\frac{\partial \mathbb{E}\{e^2(n)\}}{\partial a_k} = 0, \quad k = 1, \dots, N,$$

hay dưới dạng ma trận

$$\mathbf{R}_N \mathbf{a} = \mathbf{r}_N. \quad (1.104)$$

Từ đó, ta suy ra giá trị sai số dự đoán bình phương trung bình tối thiểu là

$$\varepsilon_N = R(0) - \mathbf{a}^T \mathbf{r}_N. \quad (1.105)$$

Phương trình (1.104) gọi là **phương trình trực giao** (normal equation). Ta có thể giải phương trình trực giao bằng cách nghịch đảo ma trận  $\mathbf{R}_N$ . Tuy nhiên, có một phương pháp khác thích hợp hơn vì có cấu trúc đệ quy và đặc biệt cho phép xây dựng một cấu trúc gọi là **thang chéo** (trellis), có áp dụng rất quan trọng trong xử lý tiếng nói. Đó là **phương pháp Levinson-Durbin**. Phương pháp này đã được Levinson công bố năm 1947 nhằm để thiết kế bộ lọc tối ưu Wiener (xem Chương 5 của Tập I giáo trình Xử lý tín hiệu ngẫu nhiên [18]). Đến 1979, Durbin lại đề nghị một phương pháp hơi giống của Levinson nhưng chất lượng cao hơn. Đây là lý do tại sao phương pháp lại có tên là Levinson-Durbin.

## 1.B Thuật toán Levinson-Durbin

Bài toán được đặt ra như sau:

Cho một tín hiệu ngẫu nhiên dừng mở rộng  $\{X(n)\}$  mà ta dự đoán bởi một toán tử dự đoán tuyến tính tối ưu bậc  $(p - 1)$ , xác định mối liên hệ giữa toán tử dự đoán tuyến tính tối ưu bậc  $(p - 1)$  và bậc  $p$ .

Trước tiên, ta sẽ đặt một số ký hiệu. Vector cột  $\mathbf{a}_N$  ký hiệu bao gồm các hệ số của toán tử dự đoán tuyến tính tối ưu bậc  $N$ :

$$\mathbf{a}_N = [a_N(1), a_N(2), \dots, a_N(N)]^T. \quad (1.106)$$

Vector cột  $\mathbf{a}_N^{N-1}$  gồm  $(N - 1)$  thành phần đầu tiên của  $\mathbf{a}_N$ :

$$\mathbf{a}_N^{N-1} = [a_N(1), \dots, a_N(N - 1)]^T. \quad (1.107)$$

Vector cột  $\bar{\mathbf{a}}_N$  đảo ngược vị trí của các hệ số trong  $\mathbf{a}_N$ :

$$\bar{\mathbf{a}}_N = [a_N(N), a_N(N - 1), \dots, a_N(1)]^T. \quad (1.108)$$

Vector cột  $\bar{\mathbf{r}}_N$  đảo ngược vị trí các giá trị tương quan trong  $\mathbf{r}_N$ :

$$\bar{\mathbf{r}}_N = [R(N), \dots, R(0)]^T. \quad (1.109)$$

Ngoài ra, ta đặt

$$k_N = a_N(N), \quad (1.110)$$

$k_N$  có tên là **hệ số phản xạ** (reflection coefficient) của hệ thống mà ta sẽ thấy rõ ý nghĩa sau khi triển khai thuật toán.

Chọn  $N = p$ , phương trình trực giao

$$\mathbf{a}_p = \mathbf{R}_p^{-1} \mathbf{r}_p$$

được chi tiết hóa thành

$$\begin{bmatrix} R(0) & \cdots & R(p-2) & : & R(p-1) \\ \vdots & \ddots & \vdots & & \vdots \\ R(p-2) & \cdots & R(0) & & R(1) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ R(p-1) & \cdots & R(1) & : & R(0) \end{bmatrix} \begin{bmatrix} a_p(1) \\ \vdots \\ a_p(p-1) \\ \cdots \\ a_p(p) \end{bmatrix} = \begin{bmatrix} R(1) \\ \vdots \\ R(p-1) \\ \cdots \\ R(p) \end{bmatrix}.$$

Ta phân  $\mathbf{R}_p$  thành bốn khối như trên và sử dụng các ký hiệu ở trên để biểu diễn lại phương trình trực giao

$$\begin{bmatrix} \mathbf{R}_{p-1} & : & \bar{\mathbf{r}}_{p-1} \\ \cdots & \cdots & \cdots \\ \bar{\mathbf{r}}_{p-1}^T & : & R(0) \end{bmatrix} \begin{bmatrix} \mathbf{a}_p^{p-1} \\ \vdots \\ k_p \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{p-1} \\ \cdots \\ R(p) \end{bmatrix}. \quad (1.111)$$

Khai triển tích ma trận theo khối cho ta

$$\mathbf{R}_{p-1} \mathbf{a}_p^{p-1} + k_p \bar{\mathbf{r}}_{p-1} = \mathbf{r}_{p-1}, \quad (1.112)$$

$$\bar{\mathbf{r}}_{p-1}^T \mathbf{a}_p^{p-1} + R(0) k_p = R(p). \quad (1.113)$$

Biết rằng tại thời điểm  $(p-1)$  trong quá khứ ta đã có kết quả là phương trình trực giao

$$\mathbf{a}_{p-1} = \mathbf{R}_{p-1}^{-1} \mathbf{r}_{p-1},$$

nhân hai vế của (1.112) với  $\mathbf{R}_{p-1}^{-1}$  cho ta

$$\begin{aligned} \mathbf{a}_p^{p-1} &= \mathbf{a}_{p-1} - k_p \mathbf{R}_{p-1}^{-1} \bar{\mathbf{r}}_{p-1} \\ &= \mathbf{a}_{p-1} - k_p \bar{\mathbf{a}}_{p-1}. \end{aligned} \quad (1.114)$$

Như vậy, thông qua (1.114),  $(p-1)$  phân tử đầu tiên của vector tối ưu  $\mathbf{a}_p$  được xác định bởi vector tối ưu  $\mathbf{a}_{p-1}$  bậc  $(p-1)$  và hệ số phản xạ  $k_p$ .



Để thấy rõ ý nghĩa của  $k_p$ , ta thay (1.114) vào (1.113) để có

$$k_p[R(0) - \bar{\mathbf{r}}_{p-1}^T \bar{\mathbf{a}}_{p-1}] = R(p) - \bar{\mathbf{r}}_{p-1}^T \mathbf{a}_{p-1}.$$

Để dàng thấy

$$\bar{\mathbf{r}}_{p-1}^T \bar{\mathbf{a}}_{p-1} = \mathbf{r}_{p-1}^T \mathbf{a}_{p-1},$$

nên ta suy ra

$$k_p = \frac{R(p) - \bar{\mathbf{r}}_{p-1}^T \mathbf{a}_{p-1}}{R(0) - \mathbf{r}_{p-1}^T \mathbf{a}_{p-1}}. \quad (1.115)$$

Biết rằng sai số bình phương trung bình tối ưu tại  $(p-1)$  là

$$\varepsilon_{p-1} = R(0) - \mathbf{r}_{p-1}^T \mathbf{a}_{p-1}, \quad (1.116)$$

ta suy ra

$$k_p = \frac{1}{\varepsilon_{p-1}} [R(p) - \bar{\mathbf{r}}_{p-1}^T \mathbf{a}_{p-1}]. \quad (1.117)$$

Ta có thể tính  $\varepsilon_p$  trực tiếp, nhưng có một cách thuận tiện hơn để tính như sau. Biểu diễn  $\varepsilon_p$  tương tự như trong (1.116) nhưng cho chỉ số  $p$ :

$$\varepsilon_p = R(0) - \mathbf{a}_p^T \mathbf{r}_p \quad (1.118)$$

$$\begin{aligned} &= R(0) - [(\mathbf{a}_p^{p-1})^T \quad a_p(p)] \begin{bmatrix} \mathbf{r}_{p-1} \\ R(p) \end{bmatrix} \\ &= R(0) - (\mathbf{a}_p^{p-1})^T \mathbf{r}_{p-1} - k_p R(p) \quad (\text{với } k_p = a_p(p)) \\ &= R(0) - [\mathbf{a}_{p-1}^T - k_p \bar{\mathbf{a}}_{p-1}^T] \mathbf{r}_{p-1} - k_p R(p) \\ &\quad \quad \quad (\text{áp dụng (1.114)}) \\ &= \underbrace{[R(0) - \mathbf{a}_{p-1}^T \mathbf{r}_{p-1}]}_{\varepsilon_{p-1}} - \underbrace{k_p [R(p) - \bar{\mathbf{a}}_{p-1}^T \mathbf{r}_{p-1}]}_{k_p \varepsilon_{p-1}} \\ &= (1 - k_p^2) \varepsilon_{p-1}. \end{aligned} \quad (1.119)$$

So với (1.118), (1.119) chỉ cần một phép nhân và một phép cộng khi đã có trước  $\varepsilon_{p-1}$ , trong khi (1.116) cần đến  $p$  phép nhân và một phép cộng. Thuật toán Levinson-Durbin có cấu trúc như trong Thuật toán 1.1.

**Thuật toán 1.1** Thuật toán Levinson-Durbin**Khởi động:**Gán sai số tối ưu  $\varepsilon_0 = R(0)$ **for**  $p = 1 : N$  **do**

$$k_p = \frac{1}{\varepsilon_{p-1}} [R(p) - \sum_{k=1}^{p-1} a_{p-1}(k) R(p-k)], \text{ theo (1.117)}$$

Tính hệ số mô hình hiện tại và cập nhật các hệ số quá khứ

$$a_p(p) = k_p, \text{ theo (1.110)}$$

$$\mathbf{a}_p^{p-1} = \mathbf{a}_{p-1} - k_p \bar{\mathbf{a}}_{p-1}, \text{ theo (1.114)}$$

Tính sai số tối ưu

$$\varepsilon_p = (1 - k_p) \varepsilon_{p-1}, \text{ theo (1.119)}$$

**end for** $\mathbf{a}_N$ : bộ lọc dự đoán tối ưu bậc  $N$ 

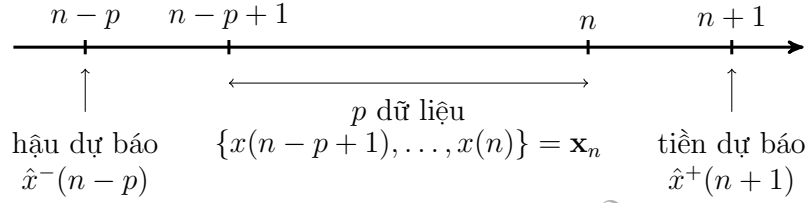
Cần chú ý rằng, nếu không cẩn thận, thuật toán này có khả năng bất ổn định và sai số bình phương trung bình tối ưu trở thành âm. Để giải quyết vấn đề này, Burg đã gợi ý dùng một cấu trúc hỗn hợp gọi là **bộ lọc thang chéo** (trellis) tức là xét cùng lúc bài toán **dự đoán một bước trong tương lai** (forward predictor) và **dự đoán một bước trong quá khứ** (backward predictor) mà ta sẽ gọi đơn giản là **tiền dự đoán**, ký hiệu là  $\hat{x}^+$ , và **hậu dự đoán**, ký hiệu là  $\hat{x}^-$ , như minh họa trong Hình 1.8, và cho dưới đây:

$$\hat{x}^+(n+1) = \sum_{k=1}^p a_p^+(k) x(n-k+1), \quad (1.120)$$

$$\hat{x}^-(n-p) = \sum_{k=1}^p a_p^-(k) x(n-k+p). \quad (1.121)$$

Nếu tín hiệu  $\{X(n)\}$  là dừng mở rộng thì thống kê sẽ không thay đổi nếu ta nhìn về hướng thời gian dương cũng như về hướng thời gian âm. Vậy dự đoán tối ưu theo hướng dương hay âm đều phải có cùng cấu trúc. Ta có thể kiểm chứng dễ dàng kết quả này bằng tính toán, nhưng đây là điều không cần thiết. Kết quả là

$$a_p^-(k) = a_p^+(p-k+1). \quad (1.122)$$



**Hình 1.8** Cải tiến thuật toán Levinson-Durbin, kết hợp tiên dự đoán với hậu dự đoán.

Từ đó, tại thời điểm  $n$ , với  $n = p+1, \dots, N$ , ta có sai số dự tiên dự đoán là

$$e_p^+(n) \triangleq x(n) - \hat{x}^+(n) \quad (1.123)$$

$$= x(n) - \sum_{k=1}^p a_p^+(k)x(n-k) \quad (1.124)$$

$$= x(n) - a_p^+(1)x(n-1) - a_p^+(2)x(n-2) - \dots - a_p^+(p)x(n-p)$$

và sai số hậu dự đoán là

$$e_p^-(n) \triangleq x(n-p) - \hat{x}^-(n-p) \quad (1.125)$$

$$= x(n-p) - \sum_{k=1}^p a_p^-(k)x(n-p+k) \quad (1.126)$$

$$= x(n-p) - \sum_{k=1}^p a_p^+(p-k+1)x(n-p+k) \quad (1.127)$$

$$= x(n-p) - a_p^+(1)x(n-p+1) - a_p^+(2)x(n-p+2) - \dots - a_p^+(p)x(n).$$

Đặt  $G_p^+(z)$  và  $G_p^-(z)$  là hàm truyền từ  $x(n)$  đến  $\hat{x}^+(n+1)$  và  $\hat{x}^-(n-p)$ , ta có

$$\mathbf{g}_p^+ = (1, -a_p(1), -a_p(2), \dots, -a_p(p)), \quad (1.128)$$

$$\mathbf{g}_p^- = (-a_p(p), -a_p(p-1), \dots, -a_p(1), 1) \quad (1.129)$$

và

$$G_p^+(z) = 1 - \sum_{k=1}^p a_p(k)z^{-k}, \quad (1.130)$$

$$G_p^-(z) = -a_p(p) - a_p(p-1)z^{-1} - \dots - a_p(1)z^{-(p-1)} + z^{-p}. \quad (1.131)$$

Cấu trúc này cho ta

$$E_p^+(z) = E_{p-1}^+(z) - k_p z^{-1} E_{p-1}^-(z),$$

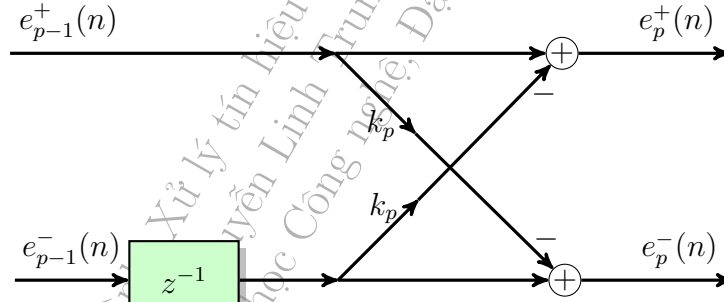
$$E_p^-(z) = z^{-1} E_{p-1}^-(z) - k_p E_{p-1}^+(z),$$

hoặc

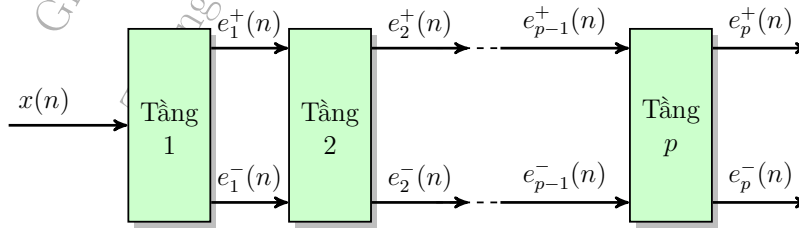
$$e_p^+(n) = e_{p-1}^+(n) - k_p e_{p-1}^-(n-1),$$

$$e_p^-(n) = e_{p-1}^-(n-1) - k_p e_{p-1}^+(n).$$

Từ đó ta có cấu trúc tầng  $p$  như trong Hình 1.9(a) và cấu trúc đầy đủ các tầng trong Hình 1.9(b).



(a) Cấu trúc tầng  $p$



(b) Bộ dự đoán  $p$  bước

**Hình 1.9** Cấu trúc thang chéo.

Cấu trúc thang chéo này rất hữu ích cho mã hóa tiếng nói (tương đương với nén tiếng nói). Ngoài ra, Burg đã sử dụng cả hai sai số dự đoán  $e_p^+(n)$  và  $e_p^-(n)$  để xây dựng thuật toán Burg bảo đảm tính ổn định và tính các hệ số của mô hình AR dựa vào ước lượng của hàm tự tương quan.

## 1.C Thuật toán Burg

Burg đã chứng minh rằng nếu ta áp đặt điều kiện  $R_X(0), \dots, R_X(N)$  như là  $(N+1)$  giá trị cho sẵn của hàm tự tương quan, thì tín hiệu  $\{X(n)\}$  thỏa điều kiện này và có entropy cực đại là một tín hiệu Gauss dừng mở rộng cơ cấu trúc AR bậc  $N$ . Vì kết quả này nên người ta thường sử dụng mô hình AR để ước lượng phổ công suất của một tín hiệu ngẫu nhiên mà ta đã ước lượng được  $(N+1)$  giá trị  $R_X(0), \dots, R_X(N)$  của hàm tự tương quan, hay  $(2N+1)$  giá trị của  $R_X(-N), \dots, R_X(0), \dots, R_X(N)$ . Từ tập các giá trị này, ta phải suy ra các hệ số  $a_N(1), \dots, a_N(N)$  của mô hình AR tương ứng:

$$x(n) = \sum_{k=1}^N a_N(k)x(n-k) + w(n). \quad (1.132)$$

Các hệ số  $\{a_N(k)\}$  được suy ra từ phương pháp tối ưu hóa sai số bình phương trung bình và dẫn tới hệ phương trình tuyến tính Yule-Walker, như đã cho trong (1.42). Người ta thường dùng thuật toán Levinson-Durbin, đệ quy theo bậc  $p$  của mô hình cho  $p$  từ 0 đến  $N$ , để giải hệ phương trình Yule-Walker. Xem thuật toán này trong Phụ lục 1.B.

Thuật toán Levinson-Durbin khá thuận tiện và đã được sử dụng tương đối rộng rãi. Tuy vậy, có đôi lúc thuật toán trở thành bất ổn định. Burg đã đề nghị một thuật toán luôn luôn bảo đảm tính ổn định để cho kết quả hoàn toàn thích hợp. Các ý chính của thuật toán này như sau.

Dùng cấu trúc dàn chéo để luôn luôn có hai sai số dự đoán dựa vào tập gồm  $p$  dữ liệu  $\{x(n-p+1), \dots, x(n)\}$  để ước lượng dữ liệu cách một bước trong tương lai  $\hat{x}(n+1)$  so với  $x(n)$  và dữ liệu cách một bước trong quá khứ  $\hat{x}(n-p)$  so

với  $x(n-p+1)$ . Các sai số này tương ứng là

$$e_p^+(n) = x(n) - \hat{x}^+(n) \quad (1.133a)$$

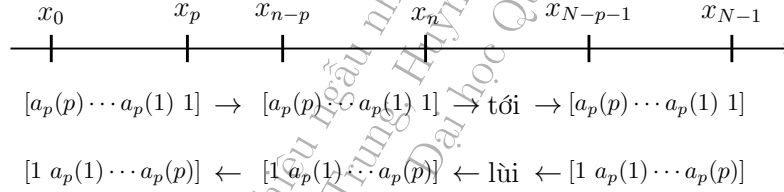
$$e_p^-(n) = x(n-p) - \hat{x}^-(n-p). \quad (1.133b)$$

Sai số bình phương trung bình sẽ là

$$\varepsilon_p = \sum_{n=p}^N ([e_p^+(n)]^2 + [e_p^-(n)]^2).$$

Đồng thời, tiếp tục cách làm của Levinson-Durbin, nhưng thay thế trị trung bình thống kê bằng trị bình phương trung bình theo thời gian.

Thuật toán được xây dựng cụ thể như sau. Hình 1.10 mô tả cụ thể tiền dự đoán và hậu dự đoán.



**Hình 1.10** Thuật toán Burg

Sai số tối thiểu là

$$\varepsilon_{\min} = \min_{\mathbf{a}_p} \left\{ \varepsilon = \sum_{n=p}^{N-1} ([e_p^+(n)]^2 + [e_p^-(n)]^2) \right\}. \quad (1.134)$$

Đáp án chưa chắc đã cho kết quả ổn định;  $[1 \ a_p(1), \dots, a_p(p)]$  có thể có zero ngoài, do đó AR trở thành bất ổn định.

Giả sử đã có  $\mathbf{a}_{p-1} = [a_{p-1}(1), \dots, a_{p-1}(p-1)]^T$ . Để tính  $\mathbf{a}_p = [a_p(1), \dots, a_p(p)]^T$ , ta cần tính hệ số phản xạ  $k_p$  để có thể sử dụng thuật toán đệ quy của Levinson-Durbin. Theo đó, sau khi có được  $k_p$ , ta cập nhật

$$\begin{bmatrix} a_p(1) \\ \vdots \\ a_p(p-1) \end{bmatrix} = \begin{bmatrix} a_{p-1}(1) \\ \vdots \\ a_{p-1}(p-1) \end{bmatrix} - k_p \begin{bmatrix} a_{p-1}(p-1) \\ \vdots \\ a_{p-1}(1) \end{bmatrix}, \quad (1.135)$$

hay gọn lại là

$$\mathbf{a}_p^{p-1} = \mathbf{a}_{p-1} - k_p \bar{\mathbf{a}}_{p-1}, \quad (1.136)$$

rồi đặt

$$a_p(p) = k_p \quad (1.137)$$

để cuối cùng có được

$$\mathbf{a}_p = \begin{bmatrix} \mathbf{a}_p^{p-1} \\ a_p(p) \end{bmatrix}.$$

Bây giờ, để tính  $k_p$ , ta giải phương trình

$$\frac{\partial \varepsilon}{\partial k_p} = 0 = 2 \sum_{n=p}^{N-1} \left( e_p^+(n) \frac{\partial e_p^+(n)}{\partial k_p} + e_p^-(n) \frac{\partial e_p^-(n)}{\partial k_p} \right). \quad (1.138)$$

Điều kiện (1.138) trở thành

$$\sum_{n=p}^{N-1} \left( e_p^+(n) \frac{\partial e_p^+(n)}{\partial k_p} + e_p^-(n) \frac{\partial e_p^-(n)}{\partial k_p} \right) = 0. \quad (1.139)$$

Theo cấu trúc thang chéo, ta có

$$e_p^+(n) = e_{p-1}^+(n) - k_p e_p^-(n-1), \quad (1.140a)$$

$$e_p^-(n) = e_{p-1}^-(n-1) - k_p e_{p-1}^+(n), \quad (1.140b)$$

$$p \leq n \leq N-1.$$

Thế (1.140) vào (1.139)

$$\sum_{n=p}^{N-1} ([e_{p-1}^+(n) - k_p e_{p-1}^-(n-1)] e_{p-1}^-(n-1) + [e_{p-1}^-(n-1) - k_p e_{p-1}^+(n)] e_{p-1}^+(n)) = 0$$

Ta suy ra

$$k_p = \frac{2 \sum_{n=p}^{N-1} e_{p-1}^+(n) e_{p-1}^-(n-1)}{\sum_{n=p}^{N-1} ([e_{p-1}^+(n)]^2 + [e_{p-1}^-(n-1)]^2)}. \quad (1.141)$$

Nếu ta lấy ký hiệu

$$\begin{aligned} \mathbf{e}_{p-1}^+ &= [e_{p-1}^+(N-1) \cdots e_{p-1}^+(p)]^2, \\ \mathbf{e}_{p-1}^- &= [e_{p-1}^-(N-2) \cdots e_{p-1}^-(p-1)]^2, \end{aligned}$$

ta có

$$k_p = \frac{2\langle \mathbf{e}_{p-1}^+, \mathbf{e}_{p-1}^- \rangle}{|\mathbf{e}_{p-1}^+|^2 + |\mathbf{e}_{p-1}^-|^2}.$$

Cho nên

$$|k_p| < 1. \quad (1.142)$$

Đây là điều kiện bảo đảm tính ổn định của mô hình.

Các biểu thức (1.136), (1.137), (1.140) và (1.141) cho ta thuật toán Burg, cũng như thuật toán Levinson-Durbin, là một thuật toán đệ quy của toán tử dự đoán, được tóm tắt trong Thuật toán 1.2.

### Thuật toán 1.2 Thuật toán Burg

**Khởi động:**  $p = 0$

$$e_0^+(n) = e_0^-(n) = x(n), \quad n = 0, 1, \dots, N-1$$

$$A_0(z) = 1$$

$$E_0 = \frac{1}{N} \sum_{n=0}^{N-1} x^2(n)$$

**for**  $p = 1, \dots, N-1$  **do**

Tại  $p$ , đã có  $e_{p-1}^+(n), e_{p-1}^-(n), n = 0, 1, \dots, N-1$   
và  $A_{p-1}(z), E_{p-1}$

Tính:

$$k_p = \frac{2 \sum_{n=p}^{N-1} e_{p-1}^+(n) e_{p-1}^-(n-1)}{\sum_{n=p}^{N-1} ([e_{p-1}^+(n)]^2 + [e_{p-1}^-(n-1)]^2)}$$

$$\mathbf{a}_p^{p-1} = \mathbf{a}_{p-1} - k_p \bar{\mathbf{a}}_{p-1}$$

$$a_p(p) = k_p$$



Suy ra

$\mathbf{a}_p$  và

$$A_p(z) = \sum_{k=0}^p a_p(k) z^{-k}, \quad a_p(0) = 1.$$

Tính:

$$e_p^+(n) = e_{p-1}^+(n) - k_p e_p^-(n-1)$$

$$e_p^-(n) = e_{p-1}^-(n-1) - k_p e_p^+(n), \quad p \leq n \leq N-1$$

Cập nhật  $E_p = (1 - k_p^2) E_{p-1}$

end for

Giáo trình: Xử lý tín hiệu ngẫu nhiên - Tập II.  
 Nguyễn Linh Trung, Huỳnh Hữu Tuệ  
 Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội

# Bài tập Chương 1

Nâng cao  
2025

## 1.1 Cho tín hiệu ngẫu nhiên

$$y(t) = 10 \sin(0,24\pi t + \psi_1) + 5 \sin(0,26\pi t + \psi_2) + e(t),$$

trong đó  $t = 1, 2, \dots, 64$ ,  $e(t)$  là nhiễu trắng Gauss với phương sai  $\sigma^2$ ,  $\psi_1$  và  $\psi_2$  là các biến ngẫu nhiên độc lập có phân phối đều trên đoạn  $[-\pi, \pi]$ . Chạy Monte-Carlo 50 lần tạo  $y(t)$  với phương sai đơn vị.

a) Tính PSD chính xác của  $y(t)$ . Ước lượng phổ của tín hiệu sử dụng các phương pháp Pisarenko và MUSIC với các bậc khác nhau. Với tham số nào, phương pháp nào có thể tìm ra hai tín hiệu dạng sin?

b) Thay đổi  $\sigma^2 = 0$  và làm lại yêu cầu trên.

17, 2, 7, 10

1.2 Cho  $\mathbf{R}$  là ma trận hiệp phương sai với  $m = n + 1$ ,  $\mathbf{g}$  là vector trị riêng của  $\mathbf{R}$  ứng với giá trị riêng nhỏ nhất. Phương pháp Pisarenko xác định thành phần tần số tín hiệu bằng cách khai thác tính chất

$$\mathbf{a}^*(\omega)\mathbf{g} = 0, \omega = \omega_k, k = 1, \dots, n.$$

Chỉ ra tính chất trên trực tiếp từ phương trình mô hình ARMA.

1.3 Cho ma trận tự tương quan  $3 \times 3$  của một quá trình điều

hòa

$$\mathbf{R} = \begin{bmatrix} 1 & j & -2 \\ -j & 1 & j \\ -2 & -j & 1 \end{bmatrix}.$$

- a) Sử dụng phương pháp phân tích điều hòa Pisarenko, hãy tìm các tần số mũ phức và phương sai của nhiễu trắng.
- b) Lập lại câu a) sử dụng phương pháp MUSIC.

**1.4** Trong thuật toán MUSIC, tìm các đỉnh của hàm giả phổ

$$\hat{p}_{\text{MUSIC}}(\omega) = \frac{1}{\sum_{k=p+1}^M |\mathbf{u}_k^H \mathbf{s}(\omega)|^2}$$

là tương đương với việc tìm cực tiểu của mẫu số. Hãy chứng minh rằng tìm cực tiểu của mẫu số tương đương với tìm cực đại của

$$\sum_{k=1}^p |\mathbf{u}_k^H \mathbf{s}(\omega)|^2.$$

**1.5** Bài tập này nhằm mở rộng phân tích điều hòa Pisarenko cho trường hợp nhiễu nhiều màu. Xét tín hiệu sau:

$$x(n) = \sum_{k=1}^p A_k e^{jn\omega_k} + w(n),$$

trong đó  $w(n)$  là nhiễu không tương quan với các thành phần mũ phức.

- a) Nếu  $x(n)$  đi qua bộ lọc FIR bậc  $p$  với

$$\mathbf{a} = [a(0), a(1), \dots, a(p)]^T,$$

năng lượng của quá trình đầu ra là

$$\xi = \mathbb{E}\{|y(n)|^2\} = \mathbf{a}^H \mathbf{R}_x \mathbf{a}.$$

Nếu ràng buộc  $\mathbf{a}$  có chuẩn đơn vị, tức là  $\mathbf{a}^H \mathbf{a} = 1$ , hãy chứng minh rằng bộ lọc làm tối thiểu hóa  $\xi$  có  $p$  điểm không nằm trên đường tròn đơn vị tại các tần số của các thành phần mũ phức,  $\omega_k$ , và giá trị tối thiểu này bằng  $\sigma_w^2$ .

- b) Giả sử  $w(n)$  có phổ năng lượng tùy ý. Nếu ma trận tự tương quan của  $w(n)$  là  $\sigma_w^2 \mathbf{Q}$ , thì ma trận tự tương quan của  $x(n)$  là

$$\mathbf{R}_x = \mathbf{E} \mathbf{P} \mathbf{E}^H + \sigma_w^2 \mathbf{Q}.$$

Nếu  $x(n)$  đi qua bộ lọc FIR bậc  $p$  với

$$\mathbf{a} = [a(0), a(1), \dots, a(p)]^T,$$

được chuẩn hóa sao cho

$$\mathbf{a}^H \mathbf{Q} \mathbf{a} = 1.$$

Hãy chứng minh rằng bộ lọc làm tối thiểu hóa năng lượng của quá trình lỗi ra có  $p$  điểm không nằm trên đường tròn đơn vị tại các tần số của các thành phần mũ phức,  $\omega_k$ , và giá trị tối thiểu này bằng  $\sigma_w^2$ .

- c) Hãy chứng minh rằng tối thiểu hóa  $\xi = \mathbf{a}^H \mathbf{R}_x \mathbf{a}$  với ràng buộc  $\mathbf{a}^H \mathbf{Q} \mathbf{a} = 1$  tương đương với việc giải bài toán giá trị riêng mở rộng

$$\mathbf{R}_x \mathbf{a} = \lambda \mathbf{Q} \mathbf{a}$$

để tìm giá trị riêng nhỏ nhất và vector riêng tương ứng. Do đó, các tần số của các thành phần mũ phức là các nghiệm của đa thức tạo bởi vector riêng nhỏ nhất

$$V_{\min}(z) = \sum_{k=0}^p v_{\min}(k) z^{-k},$$

và  $\sigma_w^2$  chính là giá trị riêng nhỏ nhất.

## Tài liệu tham khảo

- [1] G. Bienvenu, "Influence of the spatial coherence of the background noise on high resolution passive methods," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4. IEEE, 1979, pp. 306–309.
- [2] G. Bienvenu and L. Kopp, "Adaptivity to background noise spatial coherence for high resolution passive methods," in *ICASSP'80. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5. IEEE, 1980, pp. 307–310.
- [3] —, "Optimality of high-resolution array processing using the eigen-system approach," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, no. 5, pp. 1235–1248, 1983.
- [4] S. S. Blackman and R. Popoli, "Design and analysis of modern tracking systems," (*No Title*), 1999.
- [5] D. Brandwood, "A complex gradient operator and its application in adaptive array theory," in *IEE Proceedings H (Microwaves, Optics and Antennas)*, vol. 130, no. 1. IET Digital Library, 1983, pp. 11–16.
- [6] J. P. Burg, "Maximum entropy spectral analysis," in *37th Meeting of the Society of Exploration Geophysicists*, Oklahoma City, Oklahoma, 1967.
- [7] J. A. Cadzow, "Spectral estimation: An overdetermined rational model equation approach," *Proceedings of the IEEE*, vol. 70, no. 9, pp. 907–939, 1982.
- [8] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.
- [9] B. Choi, *ARMA model identification*. Springer Science & Business Media, 2012.
- [10] T. M. Cover, *Elements of Information Theory*. John Wiley & Sons, 1999.
- [11] R. F. Fischer, *Precoding and signal shaping for digital transmission*. John Wiley & Sons, 2005.
- [12] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural Network Design*. PWS Publishing Co., 1997.
- [13] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

- [14] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.
- [15] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME—Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.
- [16] H. Li and T. Adah, "Complex-valued adaptive signal processing using nonlinear functions," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, pp. 1–9, 2008.
- [17] S. Linnainmaa, "The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors," *Master's Thesis (in Finnish)*, Univ. Helsinki, 1970.
- [18] Nguyễn Linh Trung and Huỳnh Hữu Tuệ, *Xử lý tín hiệu ngẫu nhiên. Tập I: Cơ sở*. NXB Đại học Quốc gia Hà Nội, 2025.
- [19] A. Paulraj, R. Roy, and T. Kailath, "Estimation of signal parameters via rotational invariance techniques-esprit," in *Nineteenth Asilomar Conference on Circuits, Systems and Computers*, 1985. IEEE, 1985, pp. 83–89.
- [20] V. F. Pisarenko, "The retrieval of harmonics from a covariance function," *Geophysical Journal International*, vol. 33, no. 3, pp. 347–366, 1973.
- [21] R. Remmert, *Theory of complex functions*. Springer-Verlag New York, 1991, vol. 122.
- [22] R. Roy, A. Paulraj, and T. Kailath, "Estimation of signal parameters via rotational invariance techniques-esprit," in *IEEE Military Communications Conference (MILCOM)*, vol. 3, 1986, pp. 41–6.
- [23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.
- [24] —, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [25] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.
- [26] R. Schmidt, "Multiple emitter location and signal parameter estimation," in *Proceedings of the RADC Spectrum Estimation Workshop*, Rome, NY, 1979, pp. 243–258, (reprinted in IEEE Transactions on Antenna Propagation, vol. AP34, no. 3, pp. 276–280, 1986).
- [27] A. Van Den Bos, "Complex gradient and Hessian," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 141, no. 6, pp. 380–382, 1994.
- [28] P. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences," Ph.D. dissertation, Harvard University, Cambridge, MA, 1974.
- [29] B. Widrow and M. E. Hoff, "Adaptive switching circuits," Stanford Univ Ca Stanford Electronics Labs, Tech. Rep., 1960.