國立交通大學

電機與控制工程學系

碩士論文

使用正交投影近似子空間追蹤技術之快速適應性 ESPRIT 方法

Fast Adaptive ESPRIT Algorithm Using Orthonormal Projection Approximation Subspace Tracking Technique

研究生：蘇衍禎
指導教授：鄭木火 博士

中華民國九十六年七月

使用正交投影近似子空間追蹤技術之快速適應性 ESPRIT 方法

Fast Adaptive ESPRIT Algorithm Using Orthonormal Projection Approximation Subspace Tracking Technique

研究生：	蘇衍禎	Student	: Yan-Jhen Su
指導教授：	鄭木火 博士	Advisor	: Dr. Mu-Huo Cheng

國立交通大學
電機與控制工程學系
碩士論文

A Thesis
Submitted to Department of Electrical and Control Engineering
College of Electrical and Computer Engineering
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of Master
in
Electrical and Control Engineering
July 2007
Hsinchu, Taiwan, Republic of China

中華民國九十六年七月

使用正交投影近似子空間追蹤技術之快速適應性 ESPRIT 方法

研究生: 蘇衍禎　　　　　　指導教授: 鄭木火 博士

國立交通大學電機與控制工程學系

摘要

本論文提出一種新的使用正交投影近似子空間追蹤技術之快速適應性ESPRIT 方法。ESPRIT是一項以子空間法爲基礎的演算法, 其主要的功用在於進行信號參數的估測, 尤其是應用在利用由N對感應器所組成之陣列的輸出資料來進行r個信號源的方位估測(N必須大於r)。這套演算法在最初設計時, 即是以批次式信號處理爲基礎, 因此需要繁雜的計算量來處理特徵值分解的問題。目前適應性 ESPRIT 演算法的實現, 即是結合子空間追蹤技術以及傳統 ESPRIT 方法去降低運算複雜度。在本論文中, 我們先針對古典的 ESPRIT 演算法以及適用於信號源方位估測的資料模型作一簡單的描述, 然後介紹由 Peter Strobach 所提出的兩種適應性 ESPRIT 方法; Peter Strobach 利用 QR 化簡、時序正交疊代、Given plane rotation 的概念提出 LORAF2、LORAF3兩種子空間追蹤技術, 再進一步發展出兩套適應性 ESPRIT 演算法。我們在論文中提出一種使用正交投影近似子空間追蹤技術的快速適應性 ESPRIT 方法, 本方法相當簡單且直觀, 不需引入任何複雜的數值分析概念, 且在每一次的更新處理中, 只需要約$11Nr + 10N + \mathcal{O}(r^3)$的計算量, 和 Peter Strobach 的兩種方法相比較, 本技術的確有效降低了計算量和記憶量的需求成本。我們同時也藉由電腦模擬證實, 本論文所提出的技術在信號源方位估測的應用上, 的確擁有與 Peter Strobach 的方法相同的效能。

關鍵詞: 適應性信號處理, 正交投影近似子空間追蹤技術, 方位估測, 子空間

Fast Adaptive ESPRIT Algorithm Using Orthonormal Projection Approximation Subspace Tracking Technique

Student: Yan-Jhen Su Advisor: Dr. Mu-Huo Cheng

Institute of Electrical and Control Engineering

National Chiao-Tung University

Abstract

This thesis proposes a new and fast adaptive ESPRIT algorithm using orthonormal projection approximation subspace tracking (OPAST) technique. The estimation of signal parameters via rotational invariance techniques (ESPRIT) is an attractive subspace-based algorithm for estimating signal parameters, particularly the directions of arrival (DOA) of a set of r narrowband signal sources collected by an array composed of N sensor doublets, where $N > r$. The ESPRIT algorithm, originally designed in a batch signal processing, requires large amounts of computations to implement eigenvalue decomposition. Recently, the adaptive ESPRIT algorithm is realized normally by combining an adaptive subspace tracker with classical ESPRIT to reduce the arithmetic operation complexity. In this thesis, we describe the classical ESPRIT algorithm and the data model for DOA estimation first. Then, we present some simple introductions for two adaptive ESPRIT algorithms proposed by Peter Strobach. Peter Strobach uses the concepts of QR-reduction, sequential orthogonal iteration, and Givens plane rotation to develop two subspace trackers, called LORAF2 and LORAF3, then further proposes two adaptive ESPRIT algorithms. Further we propose a fast adaptive ESPRIT technique utilizing the OPAST method to implement works for real-time processing. This technique is very simple and intuitive in no need of many complex concepts in numerical analysis, and requires only about $11Nr+10N+\mathcal{O}(r^3)$ computational complexity every update. Compare with the adaptive ESPRIT algorithms proposed by Peter Strobach, our method indeed effectively saves the costs of computations and storage sizes. By computer simulations of DOA estimations, we also demonstrate that it has the good performance identical to the adaptive ESPRIT algorithms proposed by Peter Strobach.

Keywords:ESPRIT, Adaptive signal processing, OPAST, DOA estimation, Subspace

誌謝

首先，我要特別眞誠地感謝我的指導教授鄭木火教授，由於老師的鼓勵指導與諄諄教誨，使得此論文能順利完成。在短短兩年的研究生涯中，在學習與研究的路上循循善誘，一次又一次地幫助我解決難題，並且讓我學習分析問題並加以解決的能力，得以有所成長，此外，老師待人接物的誠懇眞摯與治學態度的嚴謹細心，均使我在生活與學識上獲益良多。因此在本論文付梓之際，對於辛勤傳道並耐心授業的老師致上最誠摯的謝意。

在口試期間，承蒙口試委員林清安教授、鄭嘉慶教授以及蔡尚澕教授撥冗指導並提供許多寶貴的意見，使此論文能更臻於完善，在此也誠摯地感謝您們的辛勞。

同時，我要感謝在電控所914實驗室裡所提供的完備研究資源。感謝已畢業學姊佩樺、學長嘉富、浩緯、俊維、啓峰、信良的提攜與照顧，幫助我在研究與生活上都能順利克服困難，向前邁進。而實驗室裡一起打拼的夥伴，嘉明、宏揚、佳華與志全，在課業上的砥礪互勉，以及生活中的鼓勵和幫忙，都成爲我在忙碌的求學生涯中最好的前進助力。也感謝實驗室學弟立中、皓淵以及英哲，因爲你們的幫助和打氣，讓我更能專心於研究之上。

此外，感謝好友政傑、振銘、心樺、偉珊、迦、宗恩以及其他多年來的摯友，由於你們的耐心陪伴，鼓勵打氣以及互勉祝福，讓我能在辛苦的研究生活中，依然可以心情愉悅，展露笑顏，因爲你們的友誼，不但是我心靈上堅強的寄託，更增添我人生的色彩。

我要特別感謝我的家人，溫暖合諧的家庭讓我可以無後顧之憂得汲取知識，致力於研究和課業之上，因爲你們長久以來的支持鼓勵和包容體諒，使我能順利完成學業，並做好面對下一個人生階段挑戰的準備。最後，我再一次誠懇地感謝所有幫助過我的人，謝謝大家。

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Background

In many practical signal processing applications, the super-resolution algorithms of signal parameters estimation are very significant. Such applications include the temporal problem of estimating the frequencies of complex sinusoids in additive receiver noise, and the spatial problem of estimating the directions-of-arrival (DOA) of incident plane waves corrupted by additive sensor noise. The simplest and statistically optimal solution for frequency or DOA estimations is the classical Fourier-based method. Another satisfactory approach is the subspace-based super-resolution technique. The better technique has become attractive method for frequency or DOA estimation in the signal processing since it gives us the precise algebraic structures of signals and noises. The signal subspace is the space spanned by the eigenvectors corresponding to the larger components of eigenvalues of the input data autocorrelation matrix, and the noise subspace spanned by the eigenvectors corresponding to the smaller ones. The signal and noise subspaces usually represent the statistics of the signal and the additive noise respectively, and they are always mutually orthogonal.

Most of these approaches require the extraction of one of the two subspaces, so it would not be necessary to calculate the full eigenvalue decomposition (ED). The multiple signal classification (MUSIC) algorithm [10]-[11],[16] and the minimum-norm method [12],[16] utilize the noise subspace, while the estimation of signal parameters via rotational invariance techniques (ESPRIT) algorithm [1],[8]-[9] utilizes the signal subspace. From the computational viewpoint, the significant advantage of ESPRIT is that it produces signal parameter estimation directly in terms of ED without implementing the search procedure inherent in other methods like as MUSIC and the minimum-norm method. Although ESPRIT attractively reduces the computation and storage costs, it also has been based on batch ED of the signal correlation matrix or on singular value decomposition (SVD) of the data matrix in implemention. This method is inadvisable for real-time processing because it requires repeated ED or SVD on every updating which is very time consuming. Thus, the development of the adaptive ESPRIT algorithm is necessary for real-time frequency or DOA estimations.

1.2 Motive and Literatures Review

ESPRIT algorithm is a class of subspace-based super-resolution techniques, thus ED of the signal correlation matrix or SVD of the data matrix plays an important role to split a signal into one desired signal subspace and the other unwanted noise subspace. The ESPRIT is first introduced on block and off-line processing, and the required decomposition is computationally expensive. In order to overcome computational complexity and be suitable for on-line processing, many adaptive algorithms for subspace tracking have been researched in resent years. A class of high efficient adaptive subspace tracker based on sequential orthogonal iteration [13] is presented by Peter Strobach [5]; LORAF1 is based on QR decomposition [15] and requires $\mathcal{O}(Nr^2)$ complex arithmetic operations at each time step, where the number of sensors N is much larger than the number of signal sources r. LORAF2 uses the operation of Givens plane rotations [15] to replace the QR decomposition and also needs $\mathcal{O}(Nr^2)$ computation complexity. Only the $\mathcal{O}(Nr)$ computational complexity is necessary for LORAF3, which utilizes some approximation and requires only $(2r-1)$ operations of a Givens plane rotation. Furthermore, Peter Strobach also presents Bi-SVD1 $\mathcal{O}(Nr^2)$ and Bi-SVD2 $\mathcal{O}(Nr)$ [6] based on bi-iteration SVD concept. In reference [2], Bin Yang proposes PAST subspace tracker with $\mathcal{O}(Nr)$ computation complexity. It relies on a different interpretation of the signal subspace as the resolution of an unconstrained minimization problem, and uses the recursive least squares (RLS) methods [16] to track the signal subspace. Orthonormal PAST (OPAST) $\mathcal{O}(Nr)$ algorithm [3] consists of PAST algorithm plus an orthonormalization step proposed by K. Abed-Meraim. It promises that the columns of the signal subspace are exact orthonormal per iteration. Note that the necessary operations for each formula are calculated in terms of multiply-accumulate 'Mac' operations.

The adaptive ESPRIT algorithm is realized normally by combining an adaptive subspace tracker with classical ESPRIT to reduce the arithmetic operation complexity. Peter Strobach presents two adaptive ESPRIT algorithms [4] which used a special QR-reduction [15]. One adaptive ESPRIT needs $\mathcal{O}(Nr^2) + \mathcal{O}(r^3)$ computation complexity for realization; this algorithm exploits the subspace trackers of $\mathcal{O}(Nr^2)$ category such as LORAF2 or Bi-SVD1. The other adaptive ESPRIT needs $\mathcal{O}(Nr) + \mathcal{O}(r^3)$; it uses the 'triangular plus rank one' category as LORAF3 with $\mathcal{O}(Nr)$ computational complexity for subspace tracking.

In this thesis, we develop a fast adaptive ESPRIT algorithm based on OPAST subspace tracking. This technique can reduce the total computation complexity to $\mathcal{O}(Nr) + \mathcal{O}(r^3)$, and represents the advantage of saving computations and storage sizes. The simulation performance of this approach is also the same as the adaptive ESPRIT algorithms with LORAF2 or Bi-SVD1 subspace tracker presented by Peter Strobach.

1.3 Organization of the Thesis

The remainder of this thesis is divided into five chapters including conclusions. Chapter 2 reviews the major data model for DOA estimation and the classical ESPRIT algorithm. Chapter 3 introduces four kinds of subspace trackers, LORAF2, LORAF3, PAST, and OPAST. Chapter 4 exhibits an introduction of the adaptive ESPRIT algorithms using LORAF2 and LORAF3 subspace trackers and proposes the fast adaptive ESPRIT algorithm utilizing OPAST subspace tracking. Chapter 5 demonstrates the computer simulations and illustrates comparisons with other adaptive ESPRIT algorithms. The final chapter is the conclusions.

Chapter 2

REVIEW OF ESPRIT ALGORITHM

In order to develop an adaptive ESPRIT algorithm for real-time DOA estimation, the fundamental understanding of the basic ESPRIT algorithm [1] is necessary. A simple introduction of ESPRIT background is described first. Then we present the data model for the DOA estimation based on the spatial sensors array. Subsequently, the basic principle of ESPRIT approach is elaborated in Section 3. Finally, there is a simple summary for a class of ESPRIT algorithm, total least squares ESPRIT(TLS-ESPRIT).

2.1 Background of ESPRIT Algorithm

The DOA estimation is one of the array processing problems which relied on the spatial properties of the signals impinging on the array of sensors. Until the mid-1970's, the direction finding (DF) methods required the knowledge of the array directional sensitivity pattern in investigative form, and the antenna designer would have to construct an array of antennae according to a prespecified sensitivity pattern. It was a difficult task to analyze the array directional sensitivity pattern, because it was usually an intricately nonlinear problem. Schmidt presented the MUSIC algorithm [10]-[11] based on subspace technique by taking a geometric view of the signal parameters estimation problem in 1977, then developed it continually. One of the significant breakthroughs contributed by the MUSIC algorithm was the ability to treat the problems with arbitrary arrays of sensors. Schmidt's research mainly helped the designer out of such constraints by reducing the analytical complexity that could be realized by calibrating the array. Thus, the highly nonlinear problem of analyzing the array response to a signal from a given direction could be reduced to that of measuring and storing the response. The major disadvantage of MUSIC is that it was achieved at a considerable cost in computation for searching over parameter space and storage of array calibration information. However, although MUSIC did not reduce the computational complexity of solution to the DF problem, it did extend the applicability of super-resolution DOA estimation to arbitrary arrays of antennae.

MUSIC was the first of the super-resolution algorithms to precisely exploit the underlying data model of narrow-band signals corrupted by additive sensor noise, but the algorithm has several restrictions including the fact that complete information of the array manifold is necessary, and that the search over parameter space is computationally very expensive. ESPRIT [1] is a good subspace-based super-resolution approach to the signal parameters estimation problem that uses the invariance properties of a sensor array, and it is similar to MUSIC in that it precisely exploits the underlying data model. MUSIC exploits the orthogonal properties of the noise subspace, but ESPRIT utilizes the rotational invariance structure of the signal subspace. The ESPRIT super-resolution method also exhibits significant virtues over MUSIC, because it does not need to search the peaks over parameter space and dramatically reduces these computational complexity and storage costs. In the topic of DOA estimation, the reductions are achieved by requiring that the sensor array possess a displacement invariance, i.e., sensors occur in matched pairs with identical displacement. Thus, ESPRIT just needs to know the constant displacement between two antennae of each antenna pairs, and does not require the complete information of the array manifold. For simplification, we will only represent the description of the basic ideas behind ESPRIT for the problem of multiple sources DOA estimation from data collected by an array of sensors. In order to simplify the discussions, we deal only with single dimensional parameter spaces, e.g., azimuth-only direction finding (DF) of far-field point sources, since the basic concepts are most easily understood in such spaces. Narrow-band signals of known center frequency will be assumed for the DOA/DF estimation problem. It is worth to note that a DOA/DF estimation problem is classified as narrow-band if the signal bandwidth is small compared to the inverse of the transit time of a wavefront across the array, and the array response is not a function of frequency over the signal bandwidth.

2.2 Data Model

ESPRIT algorithm can process the spatial problem of estimating the DOA of incident plane waves corrupted by additive sensor noise, and the temporal problem of estimating the frequencies of complex sinusoids in additive receiver noise. The main difference between the two problems by utilizing ESPRIT algorithm is that their data models have some different sets. The DOA estimation is based on the spatial antennae array and the data are collected by an array of sensors. But the frequency estimation utilizes the temporal sampling concepts and the data are collected by only one antenna with different sample snapshots. To simplify the description in this thesis, all the ensuing discussion is focused on the problem of DOA estimation, and the data model of DOA estimation is presented on the following context.

A basic assumption in the ESPRIT technique is that there is a planar array of arbitrary geometry comprised of N matched sensor doublets, so there are $2N$ sensor components in this array. The sensors in each doublet are translationally separated by a known constant displacement δ and have the same sensitivity patterns. The sensor characteristics such as phase, gain, and polarization sensitivity of the elements in the doublet are arbitrary as long as the sensors are pairwise-identical, and there is a manifest requirement that each element has nonzero sensitivity in all interesting directions.

Further, assume that there are a number of $r \leq N$ independent narrow-band signals located sufficiently far from the array such that the wavefronts impinging on the array are planar in homogeneous isotropic transmission media. These sources have known central frequency ω_0 and may be concerned to be stationary zero-mean random processes or deterministic signals. Additive independent white noise is assumed to be a stationary zero-mean random process and present at all $2N$ sensors. There is an illustration shown in Figure 2.1.

In order to describe the effect of the translational invariance of the sensor array by

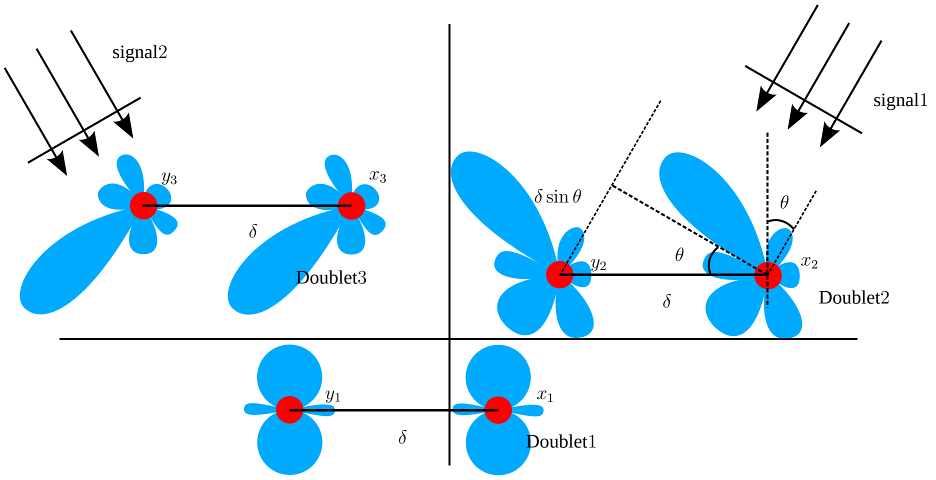

Figure 2.1: Sensor array for DOA estimation utilizing ESPRIT

mathematics, it is convenient to represent the array as being composed of two subarrays, Z_X and Z_Y, the same in every consideration even though physically displaced (not rotated) from each other by a known displacement. Further, we can present the received signals at the ith doublet as:

$$x_i(t) = \sum_{k=1}^{r} s_k(t)a_i(\theta_k) + n_{x_i}(t) \tag{2.1}$$

$$y_i(t) = \sum_{k=1}^{r} s_k(t)e^{j\omega_0\delta\sin\theta_k/c}a_i(\theta_k) + n_{y_i}(t) \tag{2.2}$$

where $s_k(t)$ is the kth wavefront source signal, θ_k is the direction-of-arrival of the kth source, $a_i(\theta_k)$ is the ith sensor of either subarray complex response for the kth wavefront source impinging on the array from direction θ_k, c is the propagative speed, $n_{x_i}(t)$ and $n_{y_i}(t)$ are the additive white noise in the ith doublet for subarrays Z_X and Z_Y, respectively.

In order to present the procedure of ESPRIT conventionally, we will combine the outputs of each of the sensors in the two subarrays and rewrite the received data in matrix notation as follows:

$$\boldsymbol{x}(t) = A\boldsymbol{s}(t) + \boldsymbol{n_x}(t) \tag{2.3}$$

$$\boldsymbol{y}(t) = A\Phi\boldsymbol{s}(t) + \boldsymbol{n_y}(t) \tag{2.4}$$

Where $\boldsymbol{x}(t)$ is an $N \times 1$ vector as:

$$\boldsymbol{x}(t) = [x_1(t) \cdots x_N(t)]^T \tag{2.5}$$

where the superscript ^{T} denotes the transpose operation, and $\boldsymbol{y}(t)$, $\boldsymbol{n_x}(t)$, and $\boldsymbol{n_y}(t)$ are similarly defined $N \times 1$ vectors. A is the $N \times r$ steering matrix whose columns $\boldsymbol{a}(\theta_k) = [a_1(\theta_k) \cdots a_N(\theta_k)]^T$ are composed of the array directional responses for the r wavefront sources, and is presented as below:

$$A \triangleq A(\theta) = [\boldsymbol{a}(\theta_1) \cdots \boldsymbol{a}(\theta_r)] \tag{2.6}$$

$\boldsymbol{s}(t)$ is the $r \times 1$ vector of impinging signals as observed at the reference sensor of subarray Z_X, and is given by:

$$\boldsymbol{s}(t) = [s_1(t) \cdots s_r(t)]^T \tag{2.7}$$

The Φ is a diagonal $r \times r$ matrix of the phase delays between the doublet sensors for the r sources, and is described as:

$$\Phi = diag\{e^{j\varphi_1}, e^{j\varphi_2}, \cdots, e^{j\varphi_r}\} \tag{2.8}$$

where $\varphi_k = \omega_0 \delta \sin(\theta_k)/c$. It is worth to note that Φ is a unitary matrix relating the measurements from subarray Z_X to those from subarray Z_Y. Φ is a simple scaling operator in the complex field. However, it is isomorphic to the real two dimensional rotation operator and is, therefore, referred to as a rotation operator.

Further, we will combine the two subarray output vectors, and define the total array output as $\boldsymbol{z}(t)$, like as:

$$\boldsymbol{z}(t) = \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{y}(t) \end{bmatrix} = \overline{A}\boldsymbol{s}(t) + \boldsymbol{n_z}(t) \tag{2.9}$$

$$\overline{A} = \begin{bmatrix} A \\ A\Phi \end{bmatrix} \tag{2.10}$$

$$\boldsymbol{n_z}(t) = \begin{bmatrix} \boldsymbol{n_x}(t) \\ \boldsymbol{n_y}(t) \end{bmatrix} \tag{2.11}$$

In the next section, we will describe how to exploit the structure of $\overline{A}$ to obtain the estimation of the diagonal elements of Φ without having to know A.

2.3 ESPRIT-The Invariance Approach

The primary concept behind the ESPRIT algorithm relies on taking advantage of the underlying rotational invariance of the signal subspace induced by the translational invariance of the sensor array. The relevant signal subspace is the one composed of the outputs from the two pairwise-matched subarrays described before, Z_X and Z_Y. Synchronous sampling of the output data of the two subarrays induces to two sets of vectors, E_X and E_Y, which span the same signal subspace. Theoretically, the relevant signal subspace spanned by the columns of A. The derivation of the ESPRIT algorithm is based on the following operations for the case in which the underlying $2N$ dimensional signal subspace consisted of the total array output data. The relevant signal subspace can be induced by accumulating a sufficient number of measurement information and determining any set of r linearly independent vectors when the noise is absent. There is a major method to find the applicable signal subspace by utilizing the information of the covariance matrix for the measurement data that composed of r uncorrelated zero-mean complex sinusoids and an additive zero mean white noise process of variance σ^2, as:

$$R_Z = E\{\boldsymbol{z}\boldsymbol{z}^H\} = \overline{A}R_S\overline{A}^H + R_N \tag{2.12}$$

Where $R_S = E\{\boldsymbol{s}\boldsymbol{s}^H\}$, and $R_N = E\{\boldsymbol{n}_z\boldsymbol{n}_z^H\} = \sigma^2 I$. The superscript ^{H} denotes the Hermitian transposition, and $E\{\cdot\}$ denotes the expection. After doing the eigendecomposition of R_Z, we obtain:

$$R_Z = E_S\Lambda_S E_S^H + E_N\Lambda_N E_N^H \tag{2.13}$$

Obviously, the eigenvalues are arranged in a decreasing order and denoted as follow: $\lambda_1 \geq \cdots \geq \lambda_r > \lambda_{r+1} = \cdots = \lambda_{2N} = \sigma^2$. The signal subspace is spanned by E_S, the eigenvectors of R_Z corresponding to the r first eigenvalues; and the noise subspace is spanned by E_N, the eigenvectors corresponding to the remaining $2N - r$ smallest eigenvalues. Clearly, $span\{E_S\} = span\{\overline{A}\}$, and it is the same as $\mathcal{R}\{E_S\} = \mathcal{R}\{\overline{A}\}$, where the $\mathcal{R}$ denotes the range space.

Because $\mathcal{R}\{E_S\} = \mathcal{R}\{\overline{A}\}$, we can find an unique nonsingular matrix T such that

$$E_S = \overline{A}T \tag{2.14}$$

Then the invariance structure of the array indicates that E_S can be divided into $E_X \in \mathcal{C}^{N\times r}$ and $E_Y \in \mathcal{C}^{N\times r}$ such that

$$E_S = \begin{bmatrix} E_X \\ E_Y \end{bmatrix} = \begin{bmatrix} AT \\ A\Phi T \end{bmatrix} \tag{2.15}$$

Obviously, the subspace spanned by the steering matrix A is equivalent to E_X and E_Y, and it is easily seen that

$$\mathcal{R}\{E_X\} = \mathcal{R}\{E_Y\} = \mathcal{R}\{A\} \tag{2.16}$$

We rearrange E_X and E_Y to a new matrix $E_{XY} \in \mathcal{C}^{N\times 2r}$ as

$$E_{XY} \triangleq \begin{bmatrix} E_X & E_Y \end{bmatrix} \tag{2.17}$$

The rank of E_{XY} is r because E_X and E_Y share a common column space. Thus, it implies that we can get an unique rank r matrix $F \in \mathcal{C}^{2r\times r}$ spanned by the null space of E_{XY} such that

$$\begin{aligned} \mathbf{0} = \begin{bmatrix} E_X & E_Y \end{bmatrix} F = \begin{bmatrix} E_X & E_Y \end{bmatrix} \begin{bmatrix} F_X \\ F_Y \end{bmatrix} &= E_X F_X + E_Y F_Y & (2.18) \\ = ATF_X + A\Phi T F_Y & & (2.19) \end{aligned}$$

Further define

$$\Psi \triangleq -F_X[F_Y]^{-1} \tag{2.20}$$

Then we can rearrange equation (2.19) to

$$AT\Psi = A\Phi T \Rrightarrow AT\Psi T^{-1} = A\Phi \tag{2.21}$$

If we assume that A is full rank, we can get

$$T\Psi T^{-1} = \Phi \tag{2.22}$$

Therefore, the diagonal matrix which is composed of the eigenvalues of Ψ is equivalent to Φ. Once the rotational invariance matrix Ψ obtained, the DOA estimation is also accomplished by implementing the eigendecomposition of Ψ. In ESPRIT algorithm, the signal parameters are obtained as nonlinear functions of the eigenvalues of the operator Ψ

that maps one set of vectors E_X that span a N dimensional signal subspace into another E_Y. In other words, $E_X\Psi = E_Y$ (substituting (2.21) for (2.15)) reflects the rotational invariance property of the received signal data subspace of the two subarrays, Z_X and Z_Y. This relationship is the key in the development of ESPRIT and its important characteristic.

2.4 Summary of the TLS ESPRIT Covariance Algorithm

For the sake of understanding the procedure of the ESPRIT algorithm further, a simple instance about the TLS ESPRIT algorithm which is a basic class of ESPRIT to be presented. The summary of the TLS ESPRIT algorithm based on a covariance is formulated as follows.

1. Get an estimation of R_Z, denoted $\hat{R}_Z$, from the measurement data Z.

2. Implement the eigendecomposition of $\hat{R}_Z$,

$$\hat{R}_Z = \overline{E}\Lambda_1\overline{E}^H$$

3. Estimate the number of signal sources $\hat{r}$.

4. Obtain E_S which contains $\hat{r}$ eigenvectors in $\overline{E}$ corresponding to the $\hat{r}$ largest eigenvalues, and denote $E_S = \begin{bmatrix} E_X \\ E_Y \end{bmatrix}$.

5. Rearrange the two submatrices E_Y and E_Y,

$$E_{XY} \triangleq \begin{bmatrix} E_X & E_Y \end{bmatrix}$$

6. Calculate the eigendecomposition,

$$E_{XY}^H E_{XY} = E\Lambda_2 E^H$$

   and separate E into four $\hat{r} \times \hat{r}$ submatrices,

$$E \triangleq \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix}$$

7. Set the matrix $\Psi = -E_{12}E_{22}^{-1}$.

8. Find the eigenvalues of Ψ,

$$\hat{\phi}_k = \lambda_k(\Psi), \quad \forall k = 1, \ldots, \hat{r}$$

9. Estimate $\hat{\theta}_k = f^{-1}(\hat{\phi}_k)$; for DOA estimation, $\hat{\theta}_k = \sin^{-1}\{c \arg(\hat{\phi}_k)/(\omega_0\delta)\}$.

Chapter 3

SUBSPACE TRACKING ALGORITHM

Signal subspace-based high-resolution techniques have become attractive methods for both spatial and temporal domain spectral analysis, and have been successfully applied to many problems for signal processing. These applications widely contain many areas such as DOA estimation, frequency estimation, bearing estimation, digital beamforming, data compression, system identification, data filtering, pattern recognition, and moving target indication. Most of these subspace methods are based on the principle of extracting a low dimensional subspace from the estimated autocorrelation matrix of input signal. The subspace spanned by the eigenvectors corresponding to the larger eigenvalues of the covariance matrix of observations is referred to as the signal subspace, because it usually expresses the statistics of the signal. Respectively, the noise subspace usually represents the statistics of the additive noise and is spanned by the eigenvectors corresponding to the smaller eigenvalues of the covariance matrix of observations. This thesis is focused on the spatial problem of DOA estimation of plane waves impinging on an antenna array solved by ESPRIT algorithm, and it mainly utilizes the properties of the signal subspace.

However, implementations of these subspace-based approaches have been carried out by using batch eigenvalue decomposition (ED) of the sample covariance matrix or singular value decomposition (SVD) of the data matrix. In the application of real-time processing, this method is unacceptable since it is necessary to process repeated ED/SVD, which is very time consuming and computation costly. If these subspaces are extracted without calculating the associated eigenvalues or eigenvectors, a significant computational effort can be realized. In order to achieve this purpose, a number of adaptive algorithms for subspace tracking has been proposed, and subspace tracker has become an important tool in real-time signal processing recently. There are several approaches for tracking the signal subspace in [2]-[3], [5]-[6]. This thesis is based on the orthonormal projection approximation subspace tracking (OPAST) and we compare it with the method based on the low-rank adaptive filter (LORAF) subspace tracker. Hence we review the two subspace tracking algorithms in this chapter.

3.1 LORAF Subspace Tracking

The low rank adaptive filters (LORAF) technique proposed by Peter Strobach [5] is a class of fast subspace tracking approaches. The principal concept of this subspace tracker is based on sequential orthogonal iteration. Assume $X(t)$ is an $L \times N$ data matrix, where L is a finite number of time snapshots and N is the number of array sensors. Define $\Gamma(t)$ as the $N \times N$ sample covariance matrix of the data matrix $X(t)$.

$$\Gamma(t) = X^H(t)X(t) \tag{3.1}$$

Then consider an $N \times r$ recursion matrix $Q(t)$ composed of orthonormal column vectors, where r is the number of the signal sources. Subsequently, we set the equation as:

$$Q(t)R(t) = \Gamma(t)Q(t-1) \tag{3.2}$$

where $Q(t)$ and $R(t)$ are the factor components of a QR-decomposition of the matrix product $A(t) = \Gamma(t)Q(t-1)$. So we have the following recursion as

$$A(t) = \Gamma(t)Q(t-1) \tag{3.3}$$

$$A(t) = Q(t)R(t) \tag{3.4}$$

This idea is an important concept known as simultaneous orthogonal iteration [13]. If $\Gamma(t)$ does not vary with time, the sequence of recursive matrix Q created by the recursion (3.3) and (3.4) will converge to the subspace consisted of the principal eigenvectors of $\Gamma(t)$. Furthermore, the sequence of triangular matrix R will converge to the diagonal matrix composed of the dominant eigenvalues of $\Gamma(t)$.

The representation above is the basic concept of the development of LORAF subspace tracker. We just divide it into two class algorithms, LORAF2 and LORAF3, according to their computational complexity. The complexity for implementing subspace tracking by LORAF2 is $\mathcal{O}(Nr^2)$ at each time step. LORAF3 just requires $\mathcal{O}(Nr)$ per iteration by modifying LORAF2 with some assumption of approximation. Since these algorithms are not the priority of this thesis, we just simply describe them in following sections.

3.1.1 LORAF2 Algorithm

In the first, we present the major important structural components that are usually relative to the most proposed fast subspace tracking techniques. The development of LORAF2 algorithm based on sequential orthogonal iteration utilizes these concepts naturally.

The so called 'initial data compaction' procedure is the fundamental implementing process in all fast subspace tracking techniques. Consider all operations occurred in the condition that the localizations of signal sources or the characteristics of data change slightly with time. Then, assume $Q(t-1) \in \mathcal{C}^{N\times r}$ be an estimate of the delayed basis matrix. Thus, we can utilize $Q(t-1)$ as a 'data compressor' on real input signal vector as:

$$\boldsymbol{h}(t) = Q^H(t-1)\boldsymbol{z}(t) \tag{3.5}$$

Obviously, all information about the relative signal in $\boldsymbol{z}(t)$ is exactly mapped into the much smaller vector $\boldsymbol{h}(t)$, which has the dimension r equal to the rank supplied by the number of independently active signal sources. After this compaction, principal matrix updating or recursion is achieved in the dimension r which is the rank of signal subspace. This operation demands a quantification component of the innovation in input data vector $\boldsymbol{z}(t)$. This innovation is indicated from the complement of the orthogonal projection of $\boldsymbol{z}(t)$ onto the delayed subspace spanned by $Q(t-1)$ when implementing the algorithms of fast subspace tracking. So we can define this complement vector as:

$$\boldsymbol{z}_{\perp}(t) = \boldsymbol{z}(t) - Q(t-1)\boldsymbol{h}(t) \tag{3.6}$$

Further, we can obtain the updated estimate $Q(t)$ of the actual basis matrix by utilizing the operation of subspace rotation as:

$$\begin{bmatrix} Q(t) & \boldsymbol{q}(t) \end{bmatrix} = \begin{bmatrix} Q(t-1) & \overline{\boldsymbol{z}}_{\perp}(t) \end{bmatrix} G^{H}(t) \tag{3.7}$$

where $\overline{\boldsymbol{z}}_{\perp}(t) = \|\boldsymbol{z}_{\perp}(t)\|_2^{-1}\boldsymbol{z}_{\perp}(t)$ is the normalized complement vector, $G(t) \in \mathcal{C}^{(r+1)\times(r+1)}$ is a subspace rotor, and $\boldsymbol{q}(t)$ is a component of unweighted quantity. Note that the $\|\cdot\|$ is the Euclidean vector norm. Deduce from (3.7) that we can write the subspace rotor in a partitioned form as:

$$\begin{aligned} G^{H}(t) &= \begin{bmatrix} Q^{H}(t-1)Q(t) & Q^{H}(t-1)\boldsymbol{q}(t) \\ \overline{\boldsymbol{z}}_{\perp}^{H}(t)\boldsymbol{Q}(t) & \overline{\boldsymbol{z}}_{\perp}^{H}(t)\boldsymbol{q}(t) \end{bmatrix} \\ &= \begin{bmatrix} \Theta(t) & \star \\ \boldsymbol{f}^{H}(t) & \star \end{bmatrix} \end{aligned} \tag{3.8}$$

where '$\star$' indicates that they are uninteresting quantities. Consequently, replacing the $G^{H}(t)$ in (3.7) by (3.8) leads to an equivalent expression form as:

$$Q(t) = Q(t-1)\Theta(t) + \overline{\boldsymbol{z}}_{\perp}^{H}(t)\boldsymbol{f}^{H}(t) \tag{3.9}$$

The main difference between the various methods is how to find this subspace rotor. The major task of the sequence of unitary plane rotations $G(t)$ is to triangularize the matrix $\overline{R}(t) \in \mathcal{C}^{(r+1)\times r}$ with full rank as:

$$\begin{bmatrix} R(t) \\ 0\cdots 0 \end{bmatrix} = G(t)\overline{R}(t) \tag{3.10}$$

where $R(t)$ is specified as an upper-right triangular matrix usually. $\overline{R}(t)$ has the different structure according to the particular type of algorithm. For subspace tracking in LORAF2 algorithm here, $\overline{R}(t)$ has the form as:

$$\overline{R}(t) = \begin{bmatrix} \alpha R(t-1)\Theta(t) + (1-\alpha)\boldsymbol{h}(t)\boldsymbol{h}^{H}(t) \\ (1-\alpha)\|\boldsymbol{z}_{\perp}(t)\|_2\boldsymbol{h}^{H}(t) \end{bmatrix} \tag{3.11}$$

where $0 \leq \alpha \leq 1$ is well known a exponential forgetting factor.

Fortunately, the diagonal elements of $R(t)$ converge to the principal eigenvalues of the underlying data covariance matrix when implementing subspace tracking by LORAF2 algorithm or the other algorithms using the concepts of eigenvalue decomposition and orthogonal iteration. We can find that $R(t)$ has the same structure in the case of subspace tracking utilizing singular value decomposition concept [15], and the diagonal elements of $R(t)$ converge to the dominant singular values of the underlying data matrix. The LORAF2 algorithm requires $\mathcal{O}(Nr^2)$ complex arithmetic operations per iteration to implement subspace tracking. All the steps of the LORAF2 algorithm for subspace tracking is summarized in Table 3.1. Note that we will do some comparisons of the computation costs and storage sizes by implementing DOA estimation in Chapter 5, so the input vector $\boldsymbol{z}(t)$ shown in the Table 3.1 must be set the dimension as $\mathcal{C}^{2N\times 1}$. The dimensions of other matrices in Table 3.1 must be adjusted to match the input vector.

Note that the way to reduce $\overline{R}(t)$ shown in (3.11) to triangular form $R(t)$ (3.10) is to utilize a full set of Givens plane rotations herein. All elementary rotations required here are of the 'annihilate bottom component by complex circular plane rotation' type. The elementary plane rotation is given as:

$$\begin{bmatrix} x_1' \\ 0 \end{bmatrix} = \begin{bmatrix} c & s^* \\ -s & c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{3.12}$$

where x_1 , x_2, and x_1' are complex numbers. In the rotor, $c = \frac{\|x_1\|}{\rho}$ is a real variable, $s = c\frac{x_2}{x_1}$ is a complex variable, and $\rho = (x_1^2 + x_2^2)^{1/2}$. Note taht the superscript * denotes the complex conjugation. The elaboration of utilizing the Givens plane rotations to triangularize $\overline{R}(t)$ is shown in [4]-[5].

3.1.2 LORAF3 Algorithm

The LORAF3 algorithm for subspace tracking is next developed by requiring an assumption of approximation to modify the LORAF2 method. Consider that $\Theta(t) = Q^H(t-1)Q(t)$ shown in (3.8) can be regarded as a matrix of cosines of angles between the basis vectors of successive subspaces, $Q(t-1)$ and $Q(t)$. Usually, we require the exponential forgetting factor α close to 1 in practice, then the angles between the associated basis vectors in two consecutive subspaces must be very small. Thus, it is reasonable to substitute $\Theta(t)$ in (3.11) for the identity matrix with slight or no performance penalty. This approximation results in that $\overline{R}(t)$ has a new form as:

$$\overline{R}(t) = \begin{bmatrix} \alpha R(t-1) + (1-\alpha)\boldsymbol{h}(t)\boldsymbol{h}^H(t) \\ (1-\alpha)\|\boldsymbol{z}_\perp(t)\|_2 \boldsymbol{h}^H(t) \end{bmatrix} \tag{3.13}$$

Obviously, this $\overline{R}(t)$ has a formation of the 'triangular plus rank one'. This operation reaches the purpose of saving computation since we utilize a succession of only $2r-1$ unitary Givens plane rotations to reduce $\overline{R}(t)$ to a triangular form $R(t)$.

However, the computational complexity for doing subspace tracking by the LORAF3 algorithm is only $\mathcal{O}(Nr)$ per time update. There is a complete elaboration of the LORAF algorithm shown in [5], and is summarized in Table 3.2. The input vector $\boldsymbol{z}(t)$ shown in the Table 3.2 must be set the dimension as $\mathcal{C}^{2N\times 1}$ for implementing the DOA estimation. The dimensions of other matrices in Table 3.2 must be adjusted to match the input vector.

3.2 PAST Subspace Tracking

The projection approximation subspace tracking (PAST) algorithm proposed by Bin Yang [2] is a technique for tracking the signal subspace. The different interpretation of the signal subspace as the solution of an unconstrained minimization problem is the major concept of PAST algorithm. This minimization work can simplify to the exponentially weighted least squares problem successfully by introducing the idea of appropriate projection approximation. Therefore, the Recursive least squares (RLS) technique can be utilized to track the signal subspace efficiently, and this is the basis of PAST algorithm. Consequently, the PAST algorithm can help us obtain the signal subspace by a computational complexity $\mathcal{O}(Nr)$.

The PAST approach is a robust and efficient method to find the signal subspace, and quickly converges to an orthonormal matrix spanning the signal subspace in most case. But the PAST method is unable to ensure the orthonormality of the weight matrix at each iteration, and probably oscillates without convergence in some case. In order to assure

Table 3.1: LORAF2 SUBSPACE TRACKING ALGORITHM

Equation	Computations	Storage Sizes
Initialize: $Q(0) = \begin{bmatrix} I_{r\times r} \\ 0_{(2N-r)\times r} \end{bmatrix}$; $R(0) = 0_{r\times r}$		
$\Theta(0) = 0_{r\times r}$; $0 \leq \alpha \leq 1$		$2Nr + 2r^2 + 1$
Input: $\boldsymbol{z}(t) \quad t = 1, 2, \cdots$		$2N$
$\boldsymbol{h}(t) = Q^H(t-1)\boldsymbol{z}(t)$	$2Nr$	r
$\boldsymbol{z}_\perp(t) = \boldsymbol{z}(t) - Q(t-1)\boldsymbol{h}(t)$	$2Nr$	
$\lVert\boldsymbol{z}_\perp(t)\rVert_2 = \sqrt{\boldsymbol{z}_\perp^H(t)\boldsymbol{z}_\perp(t)}$	$2N$	1
$\overline{\boldsymbol{z}}_\perp(t) = \lVert\boldsymbol{z}_\perp(t)\rVert_2\boldsymbol{z}_\perp(t)$	$2N$	
$R_1(t) = \begin{bmatrix} \alpha R(t-1)\Theta(t-1) + (1-\alpha)\boldsymbol{h}(t)\boldsymbol{h}^H(t) \\ (1-\alpha)\lVert\boldsymbol{z}_\perp(t)\rVert_2\boldsymbol{h}^H(t) \end{bmatrix}$	$\frac{r^3}{2} + 2r^2 + \frac{5r}{2}$	$r^2 + r$
$\begin{bmatrix} R(t) \\ 0 \end{bmatrix} = G(t)R_1(t)$	$\frac{r(r+1)(2r+1)}{3}$	$(r+1)^2$ $(forG(t))$
$G^H(t) = \begin{bmatrix} \Theta(t) & * \\ \boldsymbol{f}^H(t) & * \end{bmatrix} \overrightarrow{extract} \quad \Theta(t)$; $\boldsymbol{f}(t)$		
$Q(t) = Q(t-1)\Theta(t) + \overline{\boldsymbol{z}}_\perp^H(t)\boldsymbol{f}^H(t)$	$2Nr^2 + 2Nr$	

Table 3.2: LORAF3 SUBSPACE TRACKING ALGORITHM

Equation	Computations	Storage Sizes
Initialize: $Q(0) = \begin{bmatrix} I_{r\times r} \\ 0_{(2N-r)\times r} \end{bmatrix}$; $R(0) = 0_{r\times r}$		
$0 \leq \alpha \leq 1$		$2Nr + r^2 + 1$
Input: $\boldsymbol{z}(t) \quad t = 1, 2, \cdots$		$2N$
$\boldsymbol{h}(t) = Q^H(t-1)\boldsymbol{z}(t)$	$2Nr$	r
$\boldsymbol{z}_\perp(t) = \boldsymbol{z}(t) - Q(t-1)\boldsymbol{h}(t)$	$2Nr$	
$\lVert\boldsymbol{z}_\perp(t)\rVert_2 = \sqrt{\boldsymbol{z}_\perp^H(t)\boldsymbol{z}_\perp(t)}$	$2N$	1
$\overline{\boldsymbol{z}}_\perp(t) = \lVert\boldsymbol{z}_\perp(t)\rVert_2\boldsymbol{z}_\perp(t)$	$2N$	
$R_1(t) = \begin{bmatrix} \alpha R(t-1) + (1-\alpha)\boldsymbol{h}(t)\boldsymbol{h}^H(t) \\ (1-\alpha)\lVert\boldsymbol{z}_\perp(t)\rVert_2\boldsymbol{h}^H(t) \end{bmatrix}$	$\frac{3r^2}{2} + \frac{5r}{2}$	$r^2 + r$
$\begin{bmatrix} R(t) \\ 0 \end{bmatrix} = G(t)R_1(t)$	$3r^2 - r$	$(r+1)^2$ $(forG(t))$
$\begin{bmatrix} Q(t) & * \end{bmatrix} = \begin{bmatrix} Q(t-1) & \overline{\boldsymbol{z}}_\perp(t) \end{bmatrix} G^H(t)$	$8Nr - 4N$	

the orthonormality per iteration and to guarantee a global convergence characteristic, the orthonormal PAST (OPAST) algorithm is presented by K. Abed-Meraim [3]. Attractively, the OPAST algorithm has the same computational complexity as the PAST algorithm, and the advantages of orthonormality and global convergence. We shortly review the PAST and OPAST algorithms in the following subsections subsequently.

3.2.1 PAST Algorithm

Assume that $\boldsymbol{x} \in \mathcal{C}^N$ is a random vector process with the correlation matrix $C = E\{\boldsymbol{x}\boldsymbol{x}^H\}$. We consider the following scalar function

$$\begin{aligned} J(W) &= E\{\lVert\boldsymbol{x} - WW^H\boldsymbol{x}\rVert^2\} \\ &= tr(C) - 2tr(W^HCW) + tr(W^HCW \cdot W^HW) \end{aligned} \tag{3.14}$$

Where $W \in \mathcal{C}^{N\times r}(r < N)$ is a matrix argument, and it is assumed to have full rank r. The symbol $tr(\cdot)$ denotes the trace operator. It is worth to note that there is no constraint on W. Particularly, we do not set any restriction on the norm of W. Thus, it is impossible to maximize $J(W)$ since $J(W)$ is unbounded if the elements of W is close to infinity. So we are focused on the minimum of $J(W)$. There are two important theorems are proposed in [2]

Theorem 1 : W is a stationary point of $J(W)$ if and only if $W = U_r Q$where $U_r \in \mathcal{C}^{N\times r}$ contains any r distinct eigenvectors of C and $Q \in \mathcal{C}^{r\times r}$ is an arbitrary unitary matrix. At each stationary point, $J(W)$ equals the sum of eigenvalues whose eigenvectors are not involved in U_r.

Theorem 2 : All stationary points of $J(W)$ are saddle points except when U_r contains the r dominant eigenvectors of C. In this case, $J(W)$ attains the global minimum.

Some important comments are presented to remark on the theorems as follow.

1. If we find the signal subspace of C by minimizing $J(W)$ through iterative techniques, the guarantee of a global convergence is promised. Because $J(W)$ has a global minimum at which the column span of W equals the signal subspace and no other local minima.

2. There are not any restrictions on the orthonormality of the columns of W. The two theorems demonstrate that a solution W with orthonormal columns will automatically result from minimizing $J(W)$ in (3.14). This signal subspace property does a different exposition from those in the literature where the orthonormality $W^H W = I$ is always required exactly in terms of an optimization constraint.

Then we further describe the development of the so called projection approximation subspace tracking (PAST) algorithm based on the expression above. The first step is to replace the expectation in (3.14) with the exponentially weighted sum, yielding

$$\begin{aligned} J(W(t)) &= \sum_{i=1}^{t} \beta^{t-i} \|\boldsymbol{x}(i) - W(t)W^H(t)\boldsymbol{x}(i)\|^2 \\ &= tr[C(t)] - 2tr[W^H(t)C(t)W(t)] + tr[W^H(t)C(t)W(t)W^H(t)W(t)] \end{aligned} \tag{3.15}$$

The task of estimating the signal subspace at the time instant is affected by all input data vectors obtainable in the time interval $1 \leq i \leq t$. The introducing of the forgetting factor $0 < \beta \leq 1$ is meant to guarantee that input data in the distant past are down-weighted in order to afford the tracking capability when the system works in a nonstationary environment.

Apparently, $J(W(t))$ in (3.15) is the same as $J(W)$ in (3.14) excluding the utilization of the exponentially weighted sample correlation matrix

$$C(t) = \sum_{i=1}^{t} \beta^{t-i} \boldsymbol{x}(i)\boldsymbol{x}^H(i) = \beta C(t-1) + \boldsymbol{x}(i)\boldsymbol{x}^H(i) \tag{3.16}$$

in place of $C = E\{\boldsymbol{x}\boldsymbol{x}^H\}$.

Consequently, both theorems presented before also apply to $J(W(t))$. On the other hand, an orthonormal basis of the signal subspace spanned by the r dominant eigenvectors of $C(t)$ is constructed by the set of column vectors of $W(t)$ which minimizes $J(W(t))$.

Unfortunately, $J(W(t))$ is a fourth-order function of the elements of $W(t)$ such that it is necessary to minimize $J(W(t))$ by iterative approaches. The major key point of the PAST algorithm is approximating $W^H(t)\boldsymbol{x}(i)$ in (3.15), the unknown projection of $\boldsymbol{x}(i)$ onto the column vectors of $W(t)$ by the expression $\boldsymbol{y}(i) = W^H(i-1)\boldsymbol{x}(i)$, which can be computed for $1 \leq i \leq t$ at the time instant t. This idea produces a modified cost function

$$J'(W(t)) = \sum_{i=1}^{t} \beta^{t-i} \|\boldsymbol{x}(i) - W(t)\boldsymbol{y}(i)\|^2 \tag{3.17}$$

which is quadratic in the elements of $W(t)$.

This concept of projection approximation is the reason of the name PAST, and it modifies the error performance surface of $J(W(t))$. Nevertheless, $W^H(t)\boldsymbol{x}(i)$ is different from $W^H(i-1)\boldsymbol{x}(i)$ slightly in the environment for stationary or slowly varying signals, particularly when i is intended to t. Perhaps this dissimilarity is significant in the distant past with $i \ll t$. But the effect of past input data to the cost function is decreasing for increasing t. Thus, it is predictable that $J'(W(t))$ is a satisfactory approximation for $J(W(t))$ and the matrix $W(t)$ minimizing $J'(W(t))$ is also a suitable estimate for the signal subspace of $C(t)$.

The principal advantage of the PAST algorithm is the exponentially weighted least squares criterion (3.17), which is well studied in the topic of adaptive filter. $J'(W(t))$ is minimized if

$$W(t) = C_{xy}(t)C_{yy}^{-1}(t) \tag{3.18}$$

$$C_{xy}(t) = \sum_{i=1}^{t} \beta^{t-i}\boldsymbol{x}(i)\boldsymbol{y}^H(i) = \beta C_{xy}(t-1) + \boldsymbol{x}(t)\boldsymbol{y}^H(t) \tag{3.19}$$

$$C_{yy}(t) = \sum_{i=1}^{t} \beta^{t-i}\boldsymbol{y}(i)\boldsymbol{y}^H(i) = \beta C_{yy}(t-1) + \boldsymbol{y}(t)\boldsymbol{y}^H(t) \tag{3.20}$$

A recursive computational operation of the $N \times r$ matrix $C_{xy}(t)$ and the $r \times r$ matrix $C_{yy}(t)$ just needs the computational complexity with only $\mathcal{O}(Nr)$ and $\mathcal{O}(r^2)$. But the computational complexity of calculation $W(t)$ from $C_{xy}(t)$ and $C_{yy}(t)$ requires additional $\mathcal{O}(Nr^2) + \mathcal{O}(r^3)$. In order to implement a more efficient and more robust computation, the RLS algorithm must be introduced.

The quasicode listing of the PAST algorithm for tracking the signal subspace is summarized in the Table 3.3. Note that we set the dimension of input vector $\boldsymbol{z}(t)$ shown in the Table 3.3 as $\mathcal{C}^{2N\times 1}$ in order to implement the DOA estimation. The other matrices in Table 3.3 must be adjusted their dimensions to match the input vector. The operator $Tri\{\cdot\}$ expresses that only the upper (or lower) triangular part of $P(t) = C_{yy}^{-1}(t)$ is computed and its Hermitian transposed version is copied to the another lower (or upper) triangular part. This RLS method can decrease the computational complexity and retain the Hermitian symmetry of $P(t)$ in occurrence of rounding errors.

Some conditions for the choice of $P(0)$ and $W(0)$ as that $P(0)$ must be a Hermitian positive definite matrix and $W(0)$ should contain r orthonormal vectors. Both matrices can be computed from arbitrary initial data. However, the simplest choice is to define $P(0)$ to the $r \times r$ identity matrix and the column vectors of $W(0)$ to the r leading unit vectors of the $N \times N$ identity matrix. Interestingly, these initial values merely affect the transient behavior but not the steady state performance.

Note that this PAST algorithm also has two disadvantages. We minimize $J'(W(t))$ in place of $J(W(t))$, so we can not obtain the exact signal subspace of $C(t)$. Second, there is no promise that orthonormality of the columns of $W(t)$ exists each recursion.

Table 3.3: PAST SUBSPACE TRACKING ALGORITHM

Equation	Computations	Storage Sizes
Initialize: $W(0) = \begin{bmatrix} I_{r\times r} \\ 0_{(2N-r)\times r} \end{bmatrix}$; $P(0) = I_{r\times r}$		
$0 \leq \beta \leq 1$		$2Nr + r^2 + 1$
Input: $\boldsymbol{z}(t) \quad t = 1, 2, \cdots$		$2N$
$\boldsymbol{y}(t) = W^H(t-1)\boldsymbol{z}(t)$	$2Nr$	r
$\boldsymbol{h}(t) = P(t-1)\boldsymbol{y}(t)$	r^2	r
$\boldsymbol{g}(t) = \frac{\boldsymbol{h}(t)}{\beta + \boldsymbol{y}^H(t)\boldsymbol{h}(t)}$	$2r$	r
$P(t) = \frac{1}{\beta} Tri\{P(t-1) - \boldsymbol{g}(t)\boldsymbol{h}^H(t)\}$	$2r^2$	
$\boldsymbol{e}(t) = \boldsymbol{z}(t) - W(t-1)\boldsymbol{y}(t)$	$2Nr$	
$W(t) = W(t-1) + \boldsymbol{e}(t)\boldsymbol{g}^H(t)$	$2Nr$	

3.2.2 OPAST Algorithm

The orthonormal PAST (OPAST) algorithm [3] is a modification of the PAST algorithm. The principal purpose of this algorithm is to ensure the column vectors of $W(t)$ are exactly orthonormal per iterative step. However, it has the advantages of orthonormality and global convergence with the same computational complexity as the PAST algorithm. Since this algorithm is developed from the PAST technique, undoubtedly, the two both theorems presented by PAST algorithm also hold herein.

The OPAST algorithm is composed of the PAST algorithm and an additional orthonormalization procedure of the weight matrix per updating step

$$W_O(t) \triangleq W(t)[W^H(t)W(t)]^{-1/2} \tag{3.21}$$

Where $W_O(t)$ means that orthonromalized $W(t)$, and $(W^H(t)W(t))^{-1/2}$ indicates an inverse square root of $(W^H(t)W(t))$. This updating equation of $W(t)$ is utilized to implement the following computation. Noting that $W_O(t-1)$ is an orthonormal matrix now, we obtain

$$W^H(t)W(t) = I + \|\boldsymbol{p}(t)\|^2\boldsymbol{q}(t)\boldsymbol{q}^H(t) = I + \boldsymbol{v}\boldsymbol{v}^H$$

where $W(t) = W_O(t-1) + \boldsymbol{p}(t)\boldsymbol{q}^H(t)$, and $\boldsymbol{p}(t)$, and $\boldsymbol{q}(t)$ are shown in the Table 3.4.

The fact that $W_O^H(t-1)\boldsymbol{p}(t) = \mathbf{0}$ is used in this computation. We define $\boldsymbol{v} \triangleq \|\boldsymbol{p}(t)\|\boldsymbol{q}(t)$, and I is the identity matrix. Thus, we get

$$\begin{aligned} [W^H(t)W(t)]^{-1/2} &= I + \frac{1}{\|\boldsymbol{v}\|^2}\left(\frac{1}{\sqrt{1+\|\boldsymbol{v}\|^2}} - 1\right)\boldsymbol{v}\boldsymbol{v}^H \\ &= I + \tau(t)\boldsymbol{q}(t)\boldsymbol{q}^H(t) \end{aligned} \tag{3.22}$$

where

$$\tau(t) \triangleq \frac{1}{\|\boldsymbol{q}(t)\|^2}\left(\frac{1}{\sqrt{1+\|\boldsymbol{p}(t)\|^2\|\boldsymbol{q}(t)\|^2}} - 1\right)$$

Utilizing the equations (3.21),(3.22), and the updating equation of $W(t)$, we have a new recursive version of $W_O(t)$ as

$$\begin{aligned} W_O(t) &= (W_O(t-1) + \boldsymbol{p}(t)\boldsymbol{q}^H(t))(I + \tau(t)\boldsymbol{q}(t)\boldsymbol{q}^H(t)) \\ &= W_O(t-1) + \boldsymbol{p}'(t)\boldsymbol{q}^H(t) \end{aligned} \tag{3.23}$$

where

$$\boldsymbol{p}'(t) = \tau(t)W_O(t-1)\boldsymbol{q}(t) + (1 + \tau(t)\|\boldsymbol{q}(t)\|^2)\boldsymbol{p}(t) \tag{3.24}$$

The summary of the OPAST algorithm is expressed in Table 3.4. Note that the input vector $\boldsymbol{z}(t)$ shown in the Table 3.4 has the dimension with $\mathcal{C}^{2N\times 1}$ for implementing the DOA estimation. The dimensions of other matrices in Table 3.4 must be also adjusted to match the input vector.

The OPAST algorithm ensures exactly the orthonormality of the weight matrix $W_O(t)$ per iteration, while the PAST algorithm merely converges to an orthonormal matrix asymptotically. There is a simulation in [3] to demonstrate that the OPAST algorithm and the PAST algorithm have the identical asymptotic performance for the tracking the subspace. In the discussion of computational complexity, the OPAST costs slightly more than the PAST, but the order of operations are both identical to $\mathcal{O}(Nr) + \mathcal{O}(r^2)$. Furthermore, an example [14] is exhibited next to demonstrate that the OPAST ensure having a global convergence characteristic. Implement the eigendecomposition of C as

$$C = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \Sigma_2 \end{bmatrix} \begin{bmatrix} U_1^H \\ U_2^H \end{bmatrix} \tag{3.25}$$

where the signal subspace has a subscript with 1 and the noise subspace has a subscript with 2. Set a novel coordinate system for the weight matrix $W(t)$

$$\begin{bmatrix} U_1 & U_2 \end{bmatrix}^H W(t) = \begin{bmatrix} U_1^H W(t) \\ U_2^H W(t) \end{bmatrix} = \begin{bmatrix} M_1(t) \\ M_2(t) \end{bmatrix} \tag{3.26}$$

Consider this new coordinate system (3.26), the equation $W(t) = CW(t-1)(W^H(t-1)CW(t-1))^{-1}$ shown in [3] is equivalent to

$$\begin{aligned} \begin{bmatrix} M_1(t) \\ M_2(t) \end{bmatrix} = & \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \Sigma_2 \end{bmatrix} \begin{bmatrix} M_1(t-1) \\ M_2(t-1) \end{bmatrix} \\ & \cdot (\begin{bmatrix} M_1(t-1) \\ M_2(t-1) \end{bmatrix}^H \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \Sigma_2 \end{bmatrix} \begin{bmatrix} M_1(t-1) \\ M_2(t-1) \end{bmatrix})^{-1} \end{aligned} \tag{3.27}$$

In the condition of $\Sigma_2 = \mathbf{0}$, the equation result in

$$\begin{cases} M_1(t) = ((M_1(t-1))^H)^{-1} \\ M_2(t) = \mathbf{0} \end{cases} \tag{3.28}$$

This result indicates that the matrix $M_1(t)$ oscillates between two matrix values $M_1(t-1)$ and $((M_1(t-1))^H)^{-1}$. The advantage of the OPAST algorithm is having the ability to prevent this oscillation since the matrix $M_1(t-1)$ becomes unitary. In other words, $M_1(t-1) = ((M_1(t-1))^H)^{-1}$.

Table 3.4: OPAST SUBSPACE TRACKING ALGORITHM

Equation	Computations	Storage Sizes
Initialize: $W_O(0) = \begin{bmatrix} I_{r\times r} \\ 0_{(2N-r)\times r} \end{bmatrix}$; $P(0) = I_{r\times r}$		
$0 \leq \beta \leq 1$		$2Nr + r^2 + 1$
Input: $\boldsymbol{z}(t) \quad t = 1, 2, \cdots$		$2N$
$\boldsymbol{y}(t) = W_O^H(t-1)\boldsymbol{z}(t)$	$2Nr$	r
$\boldsymbol{q}(t) = \frac{1}{\beta}P(t-1)\boldsymbol{y}(t)$	$r^2 + r$	r
$\gamma(t) = \frac{1}{1+\boldsymbol{y}^H(t)\boldsymbol{q}(t)}$	r	1
$\boldsymbol{p}(t) = \gamma(t)[\boldsymbol{z}(t) - W_O(t-1)\boldsymbol{y}(t)]$	$2Nr + 2N$	
$\tau_1(t) = \boldsymbol{q}^H(t)\boldsymbol{q}(t)$	r	1
$\tau_2(t) = \boldsymbol{p}^H(t)\boldsymbol{p}(t)$	$2N$	1
$\tau(t) = \frac{1}{\tau_1(t)}[\frac{1}{\sqrt{1+\tau_1(t)\tau_2(t)}} - 1]$	2	
$\boldsymbol{p}'(t) = \tau(t)W_O(t-1)\boldsymbol{q}(t) + [1 + \tau(t)\tau_1(t)]\boldsymbol{p}(t)$	$2Nr + 4N + 1$	
$Q_r(t) = \boldsymbol{q}(t)\boldsymbol{q}^H(t)$	r^2	r^2
$P(t) = \frac{1}{\beta}P(t-1) - \gamma(t)Q_r(t)$	$2r^2$	
$W_O(t) = W_O(t-1) + \boldsymbol{p}'(t)\boldsymbol{q}^H(t)$	$2Nr$	

Chapter 4

ADAPTIVE ESPRIT ALGORITHM

In the practical applications, the characteristic parameters of signal sources usually are not stationary but vary with time, such as the frequency or the position of signal source. Therefore the direction of arrival (DOA) of signal source is usually different each unit time. However, it is an important and necessary task to acquire the instantaneous information of signal source in many communication signal processing applications. In order to satisfy this requirement, the development of adaptive ESPRIT algorithm for DOA estimation is needed.

In all the adaptive ESPRIT algorithms, they usually utilize the real-time subspace tracking to replace the eigenvalue decomposition of the data correlation matrix first, and then process the following procedure by the batch methods or the adaptive techniques. The subspace tracking algorithms have been described in Chapter 3, so we develop the following adaptive ESPRIT techniques using the given signal subspace in this chapter. The least-square (LS) based description of ESPRIT algorithm is represented in order to suit the ensuing presentation. Then there is a simple introduction for the adaptive ESPRIT algorithms using LORAF subspace tracking. Finally we develop an adaptive ESPRIT algorithm utilizing OPAST subspace tracking technique. Our method has the performance equivalent to other techniques and just requires only $\mathcal{O}(Nr)$ computational complexity per iteration.

4.1 The Description of ESPRIT Using LS Concpet

In order to develop the following adaptive ESPRIT algorithms, a different description of ESPRIT utilizing least-square (LS) concept is represented [1],[4]. Assume that $V_s \in \mathcal{C}^{2N \times r}$ is the signal subspace which is estimated from the subspace tracking. The definitions of N and r are identical to the data modal described in Section 2.2. From the presentation of the Section 2.3, the fact that $span\{V_s\} = span\{\overline{A}\}$ the same as $\mathcal{R}\{V_s\} = \mathcal{R}\{\overline{A}\}$ is existed exactly. Thus, undoubtedly, there must exist an unique nonsingular matrix $T_r \in \mathcal{C}^{r \times r}$ such that

$$\overline{A} = V_s T_r \tag{4.1}$$

This equation has equivalent significance as (2.14) , but with different form. By considering the equation (2.10) and (4.1), it is reasonable to separate V_s into the two ‘split subspace’ matrices $V_x \in \mathcal{C}^{N \times r}$ and $V_y \in \mathcal{C}^{N \times r}$ as

$$A = V_x T_r \tag{4.2}$$

$$A\Phi = V_y T_r \tag{4.3}$$

Apparently, the fact that

$$\mathcal{R}\{V_x\} = \mathcal{R}\{V_y\} = \mathcal{R}\{A\} \tag{4.4}$$

which similar to (2.16) is also held here. Further, the next step is replacing A in (4.3) by (4.2) such that

$$V_x T_r \Phi = V_y T_r \tag{4.5}$$

However, this equation is an attractive form since the array steering matrix A is absent here. The nonsingular subspace rotor matrix T_r is invertible, so we can apply T_r^{-1} to obtain

$$V_x T_r \Phi T_r^{-1} = V_y \tag{4.6}$$

The matrix V_x^H is multiplied on the left of (4.6) in both sides, yielding

$$V_x^H V_x (T_r \Phi T_r^{-1}) = V_x^H V_y \tag{4.7}$$

Note that the fact from (4.4), the subspace spanned by the steering matrix A assumed full rank in Section 2.3 is equivalent to V_x, so V_x is full rank with rank r. Obviously, $V_x^H V_x$ is also nonsingular, thus

$$T_r \Phi T_r^{-1} = (V_x^H V_x)^{-1} V_x^H V_y = \Psi_r \tag{4.8}$$

Finally, the diagonal matrix Φ can be expressed as

$$\Phi = T_r^{-1} \Psi_r T_r \tag{4.9}$$

Thus, we can find the phase delay matrix Φ to accomplish the DOA estimation by implementing the eigenvalue decomposition or diagonalization of Ψ_r. The principal task in this chapter is to develop an adaptive technique to obtain Ψ_r requiring the given signal subspace estimated from the subspace tracking approaches.

However, this LS based description of the ESPRIT algorithm which is slightly different from the expression in Chapter 2. It is worth to note that A and $A\Phi$ reflect the rotational invariance property because the signal subspaces of the two subarrays are identical. This is the reason of that the ESPRIT algorithm called ‘rotational invariance techniques’.

4.2 Adaptive ESPRIT Algorithms Using LORAF Subspace Tracking

A class of fast recursive adaptive ESPRIT algorithms based on LORAF subspace tracking is proposed by Peter Strobach [4]. These algorithms are developed by utilizing an especial QR-reduction that connects with the requirements of the recursive concept, and are declared that they are extremely fast, well-structured, reliable, and unconditionally stable. Because these algorithms are not our focuses, we only present simple introductions about two approaches with $\mathcal{O}(Nr^2)$ and $\mathcal{O}(Nr)$ complexity respectively in this section.

4.2.1 QR-Reduction Concept

In this subsection, we introduce the basic idea of QR-reduction for the ESPRIT algorithm. Utilize $T_r^H V_x^H$ to multiply both sides of (4.5) as

$$T_r^H V_x^H V_x T_r \Phi = T_r^H V_x^H V_y T_r \tag{4.10}$$

Designate the subspace rotor T_r to satisfy that $T_r^H V_x^H V_x T_r = I_r$, then we can get $\Phi = T_r^H V_x^H V_y T_r$. Assume that $V_x = Q_x R_x$, where $Q_x \in \mathcal{C}^{N\times r}$ is a matrix composed of orthonormal columns, and $R_x \in \mathcal{C}^{r\times r}$ is an upper triangular matrix. Find a matrix $Q \in \mathcal{C}^{r\times r}$ such that

$$Q = R_x T_r \tag{4.11}$$

It is obviously seen that $Q^H Q = T_r^H R_x^H R_x T_r = T_r^H V_x^H V_x T_r = I_r$. Replace V_x in (4.5) by $V_x = Q_x R_x$ to get

$$Q_x R_x T_r \Phi = V_y R_x^{-1} R_x T_r \tag{4.12}$$

and consider (4.11) and (4.12) to procure

$$Q_x Q \Phi = V_y R_x^{-1} Q \tag{4.13}$$

$$\Rightarrow Q\Phi = Q_x^H V_y R_x^{-1} Q \tag{4.14}$$

Then, we can find the diagonal matrix Φ as

$$\Phi = Q^H \Psi_q Q \tag{4.15}$$

where $\Psi_q \in \mathcal{C}^{r\times r}$ is

$$\Psi_q = Q_x^H V_y R_x^{-1} \tag{4.16}$$

However, the basic concepts of the recursive adaptive ESPRIT approaches would describe in the following subsection composed of (4.15), (4.16), and the QR-factorization form $V_x = Q_x R_x$.

4.2.2 Adaptive ESPRIT Algorithms with $\mathcal{O}(Nr^2)$ Complexity

According to the conclusion described above, we obtain a simple idea to do parameters estimation. Assume that we obtain the signal subspace $Q(t)$ estimated from subspace tracker, we can divide it into two submatrices as

$$Q(t) = \begin{bmatrix} V_x(t) \\ V_y(t) \end{bmatrix} \tag{4.17}$$

Then implement the QR-factorization of $V_x(t)$ such as

$$V_x(t) = Q_x(t) R_x(t) \tag{4.18}$$

From (4.16), the $\Psi_q(t)$ is given as

$$\Psi_q(t) = Q_x^H(t)V_y(t)R_x^{-1}(t) \tag{4.19}$$

Thus, we can solve this problem by implementing the eigenvalue decomposition. However, this is a concept of batch processing and required a lot of computations.

Subsequently, we introduce an adaptive ESPRIT algorithm using the LORAF2 subspace tracking. The first step is to separate $\overline{\boldsymbol{z}}_\perp(t)$ defined in Section 3.1.1 into two subvectors as

$$\overline{\boldsymbol{z}}_\perp(t) = \begin{bmatrix} \boldsymbol{z}_x(t) \\ \boldsymbol{z}_y(t) \end{bmatrix} \tag{4.20}$$

Then apply (3.9), (4.17), and (4.20) to develop the time-update recursions for the split-subspace as

$$V_x(t) = V_x(t-1)\Theta(t) + \boldsymbol{z}_x(t)\boldsymbol{f}^H(t) \tag{4.21}$$

$$V_y(t) = V_y(t-1)\Theta(t) + \boldsymbol{z}_y(t)\boldsymbol{f}^H(t) \tag{4.22}$$

Furthermore, replace $V_x(t)$ in (4.21) by the QR-factorization form in (4.18) to get

$$Q_x(t)R_x(t) = Q_x(t-1)R_x(t-1)\Theta(t) + \boldsymbol{z}_x(t)\boldsymbol{f}^H(t) \tag{4.23}$$

The next step is to specify

$$\boldsymbol{h}_x(t) = Q_x^H(t-1)\boldsymbol{z}_x(t) \tag{4.24}$$

$$\boldsymbol{z}_x^\perp(t) = \boldsymbol{z}_x(t) - Q_x(t-1)\boldsymbol{h}_x(t) \tag{4.25}$$

$$\overline{\boldsymbol{z}}_x^\perp(t) = \|\boldsymbol{z}_x^\perp(t)\|_2^{-1}\boldsymbol{z}_x^\perp(t) \tag{4.26}$$

and to use these equation to represent $\boldsymbol{z_x}(t)$ in the form as

$$\boldsymbol{z}_x(t) = \|\boldsymbol{z}_x^\perp(t)\|_2\overline{\boldsymbol{z}}_x^\perp(t) + Q_x(t-1)\boldsymbol{h}_x(t) \tag{4.27}$$

Replace $\boldsymbol{z_x}(t)$ in (4.23) with (4.27) and rearrange the form to obtain

$$Q_x(t)R_x(t) = \begin{bmatrix} Q_x(t-1) & \overline{\boldsymbol{z}}_x^\perp(t) \end{bmatrix} \begin{bmatrix} R_x(t-1)\Theta(t) + \boldsymbol{h}_x(t)\boldsymbol{f}^H(t) \\ \|\boldsymbol{z}_x^\perp(t)\|_2\boldsymbol{f}^H(t) \end{bmatrix} \tag{4.28}$$

Now, introduce a complex multiple Givens plane rotation matrix $G_x(t)$ and apply it to transform (4.28) into two updated forms as

$$\begin{bmatrix} R_x(t) \\ 0\cdots 0 \end{bmatrix} = G_x(t) \begin{bmatrix} R_x(t-1)\Theta(t) + \boldsymbol{h}_x(t)\boldsymbol{f}^H(t) \\ \|\boldsymbol{z}_x^{\perp}(t)\|_2 \boldsymbol{f}^H(t) \end{bmatrix} \tag{4.29}$$

$$\begin{bmatrix} Q_x(t) & \boldsymbol{q}_x(t) \end{bmatrix} = \begin{bmatrix} Q_x(t-1) & \overline{\boldsymbol{z}}_x^{\perp}(t) \end{bmatrix} G_x^H(t) \tag{4.30}$$

For the purpose of finding the pure updated form of $Q_x(t)$, we extract two components $\Theta_x(t)$ and $\boldsymbol{f}_x(t)$ from the rotor $G_x(t)$ as

$$\begin{aligned} G_x^H(t) &= \begin{bmatrix} Q_x^H(t-1) \\ (\overline{\boldsymbol{z}}_x^{\perp}(t))^H \end{bmatrix} \begin{bmatrix} Q_x(t) & \boldsymbol{q}_x(t) \end{bmatrix} \\ &= \begin{bmatrix} \Theta_x(t) & Q_x^H(t-1)\boldsymbol{q}_x(t) \\ \boldsymbol{f}_x^H(t) & (\overline{\boldsymbol{z}}_x^{\perp}(t))^H \boldsymbol{q}_x(t) \end{bmatrix} \end{aligned} \tag{4.31}$$

Thus, we can obtain the pure updated form of $R_x(t)$ shown in (4.29) and the pure updated form of $Q_x(t)$ as

$$Q_x(t) = Q_x(t-1)\Theta_x(t) + \overline{\boldsymbol{z}}_x^{\perp}(t)\boldsymbol{f}_x^H(t) \tag{4.32}$$

These are two important components of this algorithm based on the concept of QR reduction. From the equation (4.19), an idea that directly update the matrix product $H_w(t)$ as the form

$$H_w(t) = Q_x^H(t)V_y(t) \tag{4.33}$$

is presented. Substitute (4.22) and (4.32) into (4.33) to obtain the new representation for $H_w(t)$ and deduce it to the recursive form as

$$\begin{aligned} H_w(t) = {} & [\Theta_x^H(t)H_w(t-1) + \boldsymbol{f}_x(t)\boldsymbol{h}_{vx}^H(t)]\Theta(t) \\ & + [\Theta_x^H(t)\boldsymbol{h}_{qy}(t) + \gamma_{xy}(t)\boldsymbol{f}_x(t)]\boldsymbol{f}^H(t) \end{aligned} \tag{4.34}$$

where $\boldsymbol{h}_{vx}(t)$, $\boldsymbol{h}_{qy}(t)$, and $\gamma_{xy}(t)$ are defined as that

$$\boldsymbol{h}_{vx}(t) = V_y^H(t-1)\overline{\boldsymbol{z}}_x^{\perp}(t) \tag{4.35}$$

$$\boldsymbol{h}_{qy}(t) = Q_x^H(t-1)\boldsymbol{z}_y(t) \tag{4.36}$$

$$\gamma_{xy}(t) = (\overline{\boldsymbol{z}}_x^{\perp}(t))^H \boldsymbol{z}_y(t) \tag{4.37}$$

Finally, we can find the desired matrix Ψ_q by

$$\Psi_q(t) = H_w(t)R_x^{-1}(t) \tag{4.38}$$

The remained procedure for parameter estimation is to solve the eigenvalue decomposition of Ψ_q.

Although this approach requires $\mathcal{O}(Nr^2)$ computational complexity per time update, it exhibits the useful adaptive processing for parameters estimation, and its quasicode is listed in Tabel 4.1.

4.2.3 Adaptive ESPRIT Algorithms with $\mathcal{O}(Nr)$ Complexity

In this subsection, we will introduce another adaptive ESPRIT algorithm also proposed in [4]. This method utilizing the LORAF 3 subspace tracking has the same basic concept as the one described in the last subsection. The difference from the last one with $\mathcal{O}(Nr^2)$ complexity is the approximation of $\Theta(t) = I$ and the 'triangular plus rank one' form of $\overline{R}(t)$ in (3.13) which are both presented in LORAF3 subspace tracking .

We use the similar procedure in (4.21) and (4.22) to directly divide (3.7) into two split form as

$$\begin{bmatrix} V_x(t) & \boldsymbol{v}_x(t) \end{bmatrix} = \begin{bmatrix} V_x(t-1) & \boldsymbol{z}_x(t) \end{bmatrix} G^H(t) \tag{4.39}$$

$$\begin{bmatrix} V_y(t) & \boldsymbol{v}_y(t) \end{bmatrix} = \begin{bmatrix} V_y(t-1) & \boldsymbol{z}_y(t) \end{bmatrix} G^H(t) \tag{4.40}$$

To replace $V_x(t)$ and $V_x(t-1)$ by their QR factors to obtain

$$\begin{aligned} &\begin{bmatrix} Q_x(t)R_x(t) & \boldsymbol{v}_x(t) \end{bmatrix} \\ &= \begin{bmatrix} Q_x(t-1)R_x(t-1) & \|\boldsymbol{z}_x^{\perp}(t)\|_2 \overline{\boldsymbol{z}}_x^{\perp}(t) + Q_x(t-1)\boldsymbol{h}_x(t) \end{bmatrix} G^H(t) \end{aligned} \tag{4.41}$$

Then rewrite (4.41) by product form of partitioned matrices as

$$\begin{aligned} &\begin{bmatrix} Q_x(t) & \boldsymbol{q}_x(t) \end{bmatrix} \begin{bmatrix} R_x(t) & \boldsymbol{r}_x(t) \\ 0 \cdots 0 & r_x(t) \end{bmatrix} \\ &= \begin{bmatrix} Q_x(t-1) & \overline{\boldsymbol{z}}_x^{\perp}(t) \end{bmatrix} \begin{bmatrix} R_x(t-1) & \boldsymbol{h}_x(t) \\ 0 \cdots 0 & \|\boldsymbol{z}_x^{\perp}(t)\|_2 \end{bmatrix} G^H(t) \end{aligned} \tag{4.42}$$

Table 4.1: ADAPTIVE ESPRIT ALGORITHM $\mathcal{O}(Nr^2)$ USING LORAF 2 SUBSPACE TRACKING

Equation	Computations	Storage Sizes
Initialize: $Q_x(0) = \begin{bmatrix} I_{r\times r} \\ 0_{(N-r)\times r} \end{bmatrix}$; $R_x(0) = I_{r\times r}$		
$H_w(0) = 0_{r\times r}$		$Nr + 2r^2$
Input from Subspace Tracker: $Q(t-1)$; $\Theta(t)$		
$\overline{\boldsymbol{z}}_\perp(t)$; $\boldsymbol{f}(t)$		
Partitions: $Q(t-1) = \begin{bmatrix} V_x(t-1) \\ V_y(t-1) \end{bmatrix}$		
$\overline{\boldsymbol{z}}_\perp(t) = \begin{bmatrix} \boldsymbol{z}_x(t) \\ \boldsymbol{z}_y(t) \end{bmatrix}$		
$\boldsymbol{h}_x(t) = Q_x^H(t-1)\boldsymbol{z}_x(t)$	Nr	r
$\boldsymbol{z}_x^\perp(t) = \boldsymbol{z}_x(t) - Q_x(t-1)\boldsymbol{h}_x(t)$	Nr	
$\Vert\boldsymbol{z}_x^\perp(t)\Vert_2 = [\boldsymbol{z}_x^\perp(t)]^H\boldsymbol{z}_x^\perp(t)$	N	1
$\overline{\boldsymbol{z}}_x^\perp(t) = \Vert\boldsymbol{z}_x^\perp(t)\Vert_2^{-1}\boldsymbol{z}_x^\perp(t)$	N	
$R_{x1}(t) = \begin{bmatrix} \alpha R_x(t-1)\Theta(t) + \boldsymbol{h}_x(t)\boldsymbol{f}^H(t) \\ \Vert\boldsymbol{z}_\perp(t)\Vert_2\boldsymbol{f}^H(t) \end{bmatrix}$	$\frac{r^3}{2} + 2r^2 + \frac{3r}{2}$	$r^2 + r$
$\begin{bmatrix} R_x(t) \\ 0 \end{bmatrix} = G_x(t)R_{x1}(t)$	$\frac{r(r+1)(2r+1)}{3}$	$(r+1)^2$ $(for G_x(t))$
$G_x^H(t) = \begin{bmatrix} \Theta_x(t) & Q_x^H(t-1)\boldsymbol{q}_x(t) \\ \boldsymbol{f}_x^H(t) & (\overline{\boldsymbol{z}}_x^\perp(t))^H\boldsymbol{q}_x(t) \end{bmatrix}$		
$\overrightarrow{extract}$ $\Theta_x(t)$; $\boldsymbol{f}_x(t)$		
$\boldsymbol{h}_{vx}(t) = V_y^H(t-1)\overline{\boldsymbol{z}}_x^\perp(t)$	Nr	r
$\boldsymbol{h}_{qy}(t) = Q_x^H(t-1)\boldsymbol{z}_y(t)$	Nr	r
$\gamma_{xy}(t) = (\overline{\boldsymbol{z}}_x^\perp(t))^H\boldsymbol{z}_y(t)$	N	1
$Q_x(t) = Q_x(t-1)\Theta_x(t) + \overline{\boldsymbol{z}}_x^\perp(t)\boldsymbol{f}_x^H(t)$	$Nr^2 + Nr$	
$H_w(t) = [\Theta_x^H(t)H_w(t-1) + \boldsymbol{f}_x(t)\boldsymbol{h}_{vx}^H(t)]\Theta(t)$		
$+[\Theta_x^H(t)\boldsymbol{h}_{qy}(t) + \gamma_x y(t)\boldsymbol{f}_x(t)]\boldsymbol{f}^H(t)$	$2r^3 + 3r^2 + r$	
$\Psi_q(t) = H_w(t)R_x^{-1}(t)$	$\frac{r^3}{2}$	r^2

where $\boldsymbol{v}_x(t) = Q_x(t)\boldsymbol{r}_x(t) + \boldsymbol{q}_x(t)r_x(t)$. Further, from (4.42), we can use the Givens plane rotation $G_x(t)$ to structure the updating form as

$$\begin{bmatrix} R_x(t) & \boldsymbol{r}_x(t) \\ 0\cdots 0 & r_x(t) \end{bmatrix} = G_x(t) \begin{bmatrix} R_x(t-1) & \boldsymbol{h}_x(t) \\ 0\cdots 0 & \|\boldsymbol{z}_x^{\perp}(t)\|_2 \end{bmatrix} G^H(t) \tag{4.43}$$

$$\begin{bmatrix} Q_x(t) & \boldsymbol{q}_x(t) \end{bmatrix} = \begin{bmatrix} Q_x(t-1) & \overline{\boldsymbol{z}}_x^{\perp}(t) \end{bmatrix} G_x^H(t) \tag{4.44}$$

This case is the same as the LORAF3 subspace tracking, so requires only $2r-1$ unitary Givens plane rotations to reduce the $R_x(t-1)$ to $R_x(t)$. In order to find the matrix $H_w(t) = Q_x^H(t)V_y(t)$, we previous consider the conjugated transposed form of (4.44) to get

$$\begin{bmatrix} Q_x^H(t) \\ \boldsymbol{q}_x^H(t) \end{bmatrix} = G_x(t) \begin{bmatrix} Q_x^H(t-1) \\ (\overline{\boldsymbol{z}}_x^{\perp}(t))^H \end{bmatrix} \tag{4.45}$$

Thus, the recursive form of $H_w(t)$ can be deduce as follow

$$\begin{aligned} \begin{bmatrix} H_w(t) & Q_x^H(t)\boldsymbol{v}_y(t) \\ \boldsymbol{q}_x^H(t)V_y(t) & \boldsymbol{q}_x^H(t)\boldsymbol{v}_y(t) \end{bmatrix} &= \begin{bmatrix} Q_x^H(t) \\ \boldsymbol{q}_x^H(t) \end{bmatrix} \begin{bmatrix} V_y(t) & \boldsymbol{v}_y(t) \end{bmatrix} \\ &= G_x(t) \begin{bmatrix} Q_x^H(t-1) \\ (\overline{\boldsymbol{z}}_x^{\perp}(t))^H \end{bmatrix} \begin{bmatrix} V_y(t-1) & \boldsymbol{z}_y(t) \end{bmatrix} G^H(t) \\ &= G_x(t) \begin{bmatrix} H_w(t-1) & \boldsymbol{h}_{qy}(t) \\ \boldsymbol{h}_{vx}^H(t) & \gamma_{xy}(t) \end{bmatrix} G^H(t) \end{aligned} \tag{4.46}$$

The final step is equivalent to the last approach above.

The complete implementing procedure is summarized in Table 4.2. Attractively, this technique utilizes only $\mathcal{O}(Nr)$ computational complexity each update in time, so it is faster than the last one.

4.3 Fast Adaptive ESPRIT Algorithm with $\mathcal{O}(Nr)$ Complexity Using OPAST Subspace Tracking

The adaptive ESPRIT algorithms described above can help us to handle many problems about signal parameters estimation exactly, particularly for the spatial problem of estimating the DOA of signals. Since the techniques of adaptive signal processing are applied,

Table 4.2: FAST ADAPTIVE ESPRIT ALGORITHM $\mathcal{O}(Nr)$ USING LORAF 3 SUBSPACE TRACKING

Equation	Computations	Storage Sizes
Initialize: $Q_x(0) = \begin{bmatrix} I_{r\times r} \\ 0_{(N-r)\times r} \end{bmatrix}$; $R_x(0) = I_{r\times r}$		
$H_w(0) = 0_{r\times r}$		$Nr + 2r^2$
Input from Subspace Tracker: $Q(t-1)$; $G(t)$ $\overline{\boldsymbol{z}}_{\perp}(t)$		
Partitions: $Q(t-1) = \begin{bmatrix} V_x(t-1) \\ V_y(t-1) \end{bmatrix}$		
$\overline{\boldsymbol{z}}_{\perp}(t) = \begin{bmatrix} \boldsymbol{z}_x(t) \\ \boldsymbol{z}_y(t) \end{bmatrix}$		
$\boldsymbol{h}_x(t) = Q_x^H(t-1)\boldsymbol{z}_x(t)$	Nr	r
$\boldsymbol{z}_x^{\perp}(t) = \boldsymbol{z}_x(t) - Q_x(t-1)\boldsymbol{h}_x(t)$	Nr	
$\lVert \boldsymbol{z}_x^{\perp}(t) \rVert_2 = [\boldsymbol{z}_x^{\perp}(t)]^H \boldsymbol{z}_x^{\perp}(t)$	N	1
$\overline{\boldsymbol{z}}_x^{\perp}(t) = \lVert \boldsymbol{z}_x^{\perp}(t) \rVert_2^{-1} \boldsymbol{z}_x^{\perp}(t)$	N	
$R_{x1}(t) = \begin{bmatrix} \alpha R_x(t-1) & \boldsymbol{h}_x(t) \\ 0 & \lVert \boldsymbol{z}_{\perp}(t) \rVert_2 \end{bmatrix} G^H(t)$	$\frac{r^3}{2} + \frac{5r^2}{2} + 3r + 1$	$(r+1)^2$
$\boldsymbol{h}_{vx}(t) = V_y^H(t-1)\overline{\boldsymbol{z}}_x^{\perp}(t)$	Nr	r
$\boldsymbol{h}_{qy}(t) = Q_x^H(t-1)\boldsymbol{z}_y(t)$	Nr	r
$\gamma_{xy}(t) = (\overline{\boldsymbol{z}}_x^{\perp}(t))^H \boldsymbol{z}_y(t)$	N	1
$\begin{bmatrix} R_x(t) & \boldsymbol{r}_x(t) \\ 0 \cdots 0 & r_x(t) \end{bmatrix} = G_x(t)R_{x1}(t)$	$3r^2 - r$	$(r+1)^2$ $(forG_x(t))$
$\begin{bmatrix} Q_x(t) & \boldsymbol{q}_x(t) \end{bmatrix} = \begin{bmatrix} Q_x(t-1) & \overline{\boldsymbol{z}}_x^{\perp}(t) \end{bmatrix} G_x^H(t)$	$4Nr - 2N$	
$\begin{bmatrix} H_w(t) & Q_x^H(t)\boldsymbol{v}_y(t) \\ \boldsymbol{q}_x^H(t)V_y(t) & \boldsymbol{q}_x^H(t)\boldsymbol{v}_y(t) \end{bmatrix}$ $= G_x(t) \begin{bmatrix} H_w(t-1) & \boldsymbol{h}_{qy}(t) \\ \boldsymbol{h}_{vx}^H(t) & \gamma_{xy}(t) \end{bmatrix} G^H(t)$	$2(r+1)^3$	
$\Psi_q(t) = H_w(t)R_x^{-1}(t)$	$\frac{r^3}{2}$	r^2

the computational complexity can be low down successfully. It was proved that the performances of these two approaches are satisfactory [4]. Especially, the second technique reduce the computational complexity to $\mathcal{O}(Nr)$ at each time step, so the cost of computation is saving very much.

In this section, we propose a new adaptive ESPRIT technique utilizing the OPAST subspace tracking. This method is very simple and is developed intuitively without any complex principle. We only consider the description presented in Section 4.1, and represent the processing of our technique in the following text.

The first step is the same as other approaches. When we obtain the signal subspace $W_O(t) \in \mathcal{C}^{2N\times r}$ from the OPAST subspace tracking, we divide it into two split submatrices $V_x(t) \in \mathcal{C}^{N\times r}$ and $V_y(t) \in \mathcal{C}^{N\times r}$as

$$W_O(t) = \begin{bmatrix} V_x(t) \\ V_x(t) \end{bmatrix} \tag{4.47}$$

Simultaneously, we separate the vector $\boldsymbol{p}'(t) \in \mathcal{C}^{2N\times 1}$ given from (3.24) into two split subvectors as

$$\boldsymbol{p}'(t) = \begin{bmatrix} \boldsymbol{p}'_x(t) \\ \boldsymbol{p}'_y(t) \end{bmatrix} \tag{4.48}$$

where $\boldsymbol{p}'_x(t)$ and $\boldsymbol{p}'_y(t)$ both have the same dimension $(\in \mathcal{C}^{N\times 1})$. Consider (3.23), (4.47), and (4.48) together, we can get two updating recursions as

$$V_x(t) = V_x(t-1) + \boldsymbol{p}'_x(t)\boldsymbol{q}^H(t) \tag{4.49}$$

$$V_y(t) = V_y(t-1) + \boldsymbol{p}'_y(t)\boldsymbol{q}^H(t) \tag{4.50}$$

where $\boldsymbol{q}(t) \in \mathcal{C}^{r\times 1}$ is come from the OPAST algorithm. Obviously, (4.49), and (4.50) can be regarded as the update form of the split signal subspaces respectively effected by the subarrays, Z_X and Z_Y. Review the basic concept described in Section 4.1, it is not difficult to find the effort shown in (4.8). There is an instinctive idea about that we can obtain an adaptive processing form with low complexity by directly updating the matrix products $V_x^H V_x$ and $V_x^H V_y$. For the purpose of implementing the idea, we define two matrix products $H_1(t) \in \mathcal{C}^{r\times r}$ and $H_2(t) \in \mathcal{C}^{r\times r}$ as

$$H_1(t) = V_x^H(t)V_x(t) \tag{4.51}$$

$$H_2(t) = V_x^H(t)V_y(t) \tag{4.52}$$

Then we substitute the equation (4.49) into (4.51) to obtain the form as

$$\begin{aligned} H_1(t) &= [V_x(t-1) + \boldsymbol{p}'_x(t)\boldsymbol{q}^H(t)]^H[V_x(t-1) + \boldsymbol{p}'_x(t)\boldsymbol{q}^H(t)] \\ &= H_1(t-1) \\ &+ V_x^H(t-1)\boldsymbol{p}'_x(t)\boldsymbol{q}^H(t) \\ &+ \boldsymbol{q}(t)[\boldsymbol{p}'_x(t)]^H V_x(t-1) \\ &+ p_1(t)Q_r(t) \end{aligned} \tag{4.53}$$

where $p_1(t) \in \mathcal{C}$and $Q_r(r) \in \mathcal{C}^{r\times r}$ are defined as

$$p_1(t) = [\boldsymbol{p}'_x(t)]^H\boldsymbol{p}'_x(t) \tag{4.54}$$

$$Q_r(t) = \boldsymbol{q}(t)\boldsymbol{q}^H(t) \tag{4.55}$$

It is successful that we find an updating recursion of the matrix product $H_1(t)$. We further apply (4.49), (4.50) and (4.52) to do the similar work to obtain the form as

$$\begin{aligned} H_2(t) &= [V_x(t-1) + \boldsymbol{p}'_x(t)\boldsymbol{q}^H(t)]^H[V_y(t-1) + \boldsymbol{p}'_y(t)\boldsymbol{q}^H(t)] \\ &= H_2(t-1) \\ &+ V_x^H(t-1)\boldsymbol{p}'_y(t)\boldsymbol{q}^H(t) \\ &+ \boldsymbol{q}(t)[\boldsymbol{p}'_x(t)]^H V_y(t-1) \\ &+ p_2(t)Q_r(t) \end{aligned} \tag{4.56}$$

where $p_2(t) \in \mathcal{R}$ has the relation as

$$p_2(t) = [\boldsymbol{p}'_x(t)]^H\boldsymbol{p}'_y(t) \tag{4.57}$$

Consequently, the purpose for finding the direct updating recursions is achieved. The final step is that utilize (4.53) and (4.56) to find the matrix product $\Psi_h(t) \in \mathcal{C}^{r\times r}$ as

$$\Psi_h(t) = [H_1(t)]^{-1}H_2(t) \tag{4.58}$$

Thus, the desired phase delay or wanted signal parameters can be estimated by implementing the eigenvalue decomposition of $\Psi_h(t)$. It is worth to note that the initial values of $H_1(0)$ and $H_2(0)$ must be set suitably. Since the initial value of signal subspace matrix $W_O(0)$ has been set in the OPAST algorithm, we must compute the matched $H_1(0)$ and $H_2(0)$ as

$$H_1(0) = W_O^H(0) \begin{bmatrix} I_{N\times N} \\ 0_{N\times N} \end{bmatrix} \begin{bmatrix} I_{N\times N} & 0_{N\times N} \end{bmatrix} W_O(0) \tag{4.59}$$

$$H_2(0) = W_O^H(0) \begin{bmatrix} I_{N\times N} \\ 0_{N\times N} \end{bmatrix} \begin{bmatrix} 0_{N\times N} & I_{N\times N} \end{bmatrix} W_O(0) \tag{4.60}$$

However, our technique is very simple and intuitive without introducing any complex concepts of ‘QR-reduction’ and ‘Givens plane rotation’. Although this approach is also required $\mathcal{O}(Nr)$ computational complexity per time update, the amounts of computations of this technique is practically less than the approaches described in the last section. The effort of computation costs saving is more obvious when the signal source number r is more large. This is why we call it ‘fast adaptive ESPRIT’. Furthermore, our simulations demonstrate the fact that the performance of our approach is identical to the first presented method with $\mathcal{O}(Nr^2)$ complexity above. In the next chapter, we will show the results of simulations and do a discussion of comparing the amounts of computations and storage sizes with three techniques. We will also exhibit the summary of our adaptive ESPRIT approach in Table 4.3.

Of course, our technique is also suitable to apply the PAST subspace tracking. The amounts of computations are slightly less than our technique using the OPAST subspace tracking since the inherent property in the algorithm for subspace tracking. However, it lacks the advantages of the OPAST subspace tracking. Thus, we will do not consider our technique utilizing the PAST subspace tracking in next chapter.

Table 4.3: FAST ADAPTIVE ESPRIT ALGORITHM $\mathcal{O}(Nr)$ USING OPAST SUBSPACE TRACKING

Equation	Computations	Storage Sizes
Initialize:		
$H_1(0) = W_O^H(0) \begin{bmatrix} I_{N\times N} \\ 0_{N\times N} \end{bmatrix} \begin{bmatrix} I_{N\times N} & 0_{N\times N} \end{bmatrix} W_O(0)$		
$H_2(0) = W_O^H(0) \begin{bmatrix} I_{N\times N} \\ 0_{N\times N} \end{bmatrix} \begin{bmatrix} 0_{N\times N} & I_{N\times N} \end{bmatrix} W_O(0)$		$2r^2$
Input from Subspace Tracker:		
$W_O(t-1) \quad ; \quad \boldsymbol{p}'(t) \quad ; \quad Q_r(t)$		
Partitions:		
$W_O(t-1) = \begin{bmatrix} V_x(t-1) \\ V_y(t-1) \end{bmatrix}$		
$\boldsymbol{p}'(t) = \begin{bmatrix} \boldsymbol{p}'_x(t) \\ \boldsymbol{p}'_y(t) \end{bmatrix}$		
$p_1(t) = [\boldsymbol{p}'_x(t)]^H \boldsymbol{p}'_x(t)$	N	1
$p_2(t) = [\boldsymbol{p}'_x(t)]^H \boldsymbol{p}'_y(t)$	N	1
$H_1(t) = H_1(t-1)$		
$\quad +V_x^H(t-1)\boldsymbol{p}'_x(t)\boldsymbol{q}^H(t)$	$Nr + r^2$	
$\quad +\boldsymbol{q}(t)[\boldsymbol{p}'_x(t)]^H V_x(t-1)$		
$\quad +p_1(t)Q_r(t)$	r^2	
$H_2(t) = H_2(t-1)$		
$\quad +V_x^H(t-1)\boldsymbol{p}'_y(t)\boldsymbol{q}^H(t)$	$Nr + r^2$	
$\quad +\boldsymbol{q}(t)[\boldsymbol{p}'_x(t)]^H V_y(t-1)$	$Nr + r^2$	
$\quad +p_2(t)Q_r(t)$	r^2	
$\Psi_h(t) = (H_1(t))^{-1} H_2(t)$	$\frac{4r^3}{3}$	r^2

Chapter 5

SIMULATION RESULTS AND COMPARISON

In this chapter, computer simulations for DOA estimation by Matlab program demonstrate the applicability and the performance of the adaptive ESPRIT algorithm utilizing OPAST subspace tracking which we propose in Section 4.3. Simultaneously, we also simulate the two adaptive ESPRIT algorithms using LORAF subspace trackers described in Section 4.2 for comparison. For simplicity, the discussed three adaptive ESPRIT algorithms are called ESPRIT-OPAST, ESPRIT-LORAF2, and ESPRIT-LORAF3 respectively. Simulation results show that ESPRIT-OPAST has the tracking performance almost identical to ESPRIT-LORAF2 and ESPRIT-LORAF3. Then we also compare the required computational complexity and memory sizes to realize each adaptive ESPRIT algorithm.

5.1 Simulation Results

Consider the desired parameters $\varphi_k = \omega_0 \delta \sin(\theta_k)/c$ for DOA estimation shown in Section 2.2, the c is the speed of light obviously. If we assume that the frequency $f_0 = \omega_0/(2\pi)$ is $150MHz$ and the displacement δ is equal to $2m$, we can obtain a relation $\varphi_k = 2\pi \sin(\theta_k) = 2\pi\nu_k$. Hence, for simplicity in our simulations, our task is to estimate $\nu_k = \sin(\theta_k)$ instead of θ_k. The received data is generated according to the data model described in Section 2.2 ((2.1) to (2.9)), when the number of source r and the SNR are given. Furthermore, the forgetting factors α and β in all algorithms are set equal to 0.98 in the overall simulations here.

In the first experiment, the signal sources with constant ν_k are observed. Consider three cases, one signal sources collected by 6 sensor doublets, two signal sources collected by 10 sensor doublets, and four signal sources collected by 50 sensor doublets. Each case is simulated in both conditions, $3dB$ SNR and $0dB$ SNR. The simulation results are show in Figure 5.1 to Figure 5.6. We can find that the tracking curves of three algorithms are very close.

In the second experiment, we compare the performance of the algorithms in tracking two signals with crossed phase delay ν_k by 10 sensor doublets. Figure 5.7 and Figure 5.8 show that the signal sources with the larger slope and the shorter snapshots in two kinds of environments,$3dB$ SNR and $-3dB$ SNR respectively. Then observe the signal sources with the smaller slope and the longer snapshots. All three algorithms are simulated in both $3dB$ SNR and $-3dB$ SNR, and exhibit their results in Figure 5.9 and Figure 5.14. The results show that three algorithms have almost the same performance as each other.

Finally, We now consider two cases for that four signal sources impinging on the array composed of 50 sensor pairs. One case is tracking the suddenly varied phase delay ν_k, and the other one is tracking the smoothly varied one. Both two kinds of noise conditions, $3dB$ SNR and $-3dB$ SNR, are set in each case for three algorithms. Figure 5.15 to Figure 5.26 display the simulation results. From the figures, we can find that three approaches almost have the same performance since the tracking curves of three algorithms are almost overlapped.

However, ESPRIT-OPAST has the performance nearly identical to ESPRIT-LORAF2 and ESPRIT-LORAF3.

5.2 Computational Complexity

In the comparison of the computational complexity, we have known that ESPRIT-LORAF2 requires $\mathcal{O}(Nr^2)$ complexity per time updating, but ESPRIT-OPAST and ESPRIT-LORAF3 both only need $\mathcal{O}(Nr)$ complexity each time step. Now we want to know the amounts

Table 5.1: COMPARISON IN COMPUTATIONAL COMPLEXITY

	computations
ESPRIT-LORAF2	$3Nr^2 + 11Nr + 7N + \frac{29r^3}{6} + 9r^2 + \frac{17r}{3}$
ESPRIT-LORAF3	$20Nr + N + 3r^3 + 16r^2 + \frac{19r}{2} + 3$
ESPRIT-OPAST	$11Nr + 10N + \frac{4r^3}{3} + 9r^2 + 3r + 1$

Table 5.2: COMPARISON IN STORAGE

	storage sizes
ESPRIT-LORAF2	$3Nr + 2N + 9r^2 + 10r + 6$
ESPRIT-LORAF3	$3Nr + 2N + 8r^2 + 11r + 7$
ESPRIT-OPAST	$2Nr + 2N + 5r^2 + 2r + 6$

of computations per time recursion for three kinds of techniques respectively. The computations of every operation are shown in the Tables which are the summaries of the algorithms. We now calculate the sum of all the procedures for each of the three algorithms, ESPRIT-LORAF2, ESPRIT-LORAF3 and ESPRIT-OPAST, and show in the Table 5.1. While one 'givens plane rotor' operation is equal to two 'mac' operations in the two approaches, LORAF2 and LORAF3. However, the effects of the $\mathcal{O}(r^2)$ and $\mathcal{O}(r)$ are slight since the fact $N \gg r$ exists exactly in the most practical applications. The required amounts of computations for ESPRIT-LORAF3 are about $20Nr + N + \mathcal{O}(r^3)$, but for ESPRIT-OPAST are only about $11Nr + 10N + \mathcal{O}(r^3)$. Thus, It is obviously that ESPRIT-OPAST requires the least amounts of computations each time updating.

5.3 Storage Size

In the discussion of the amounts of the storage sizes, we attempt to understand that which one of these algorithms requires the least amounts of storage spaces. We also list the storage spaces required in every operation in the Tables. The sum of all necessary storage sizes for each algorithm is show in the Table 5.2. ESPRIT-OPAST needs $2Nr + 2N + 5r^2 + 2r + 6$ storage sizes which are smaller than others. Undoubtedly, ESPRIT-OPAST saves the storage sizes most.

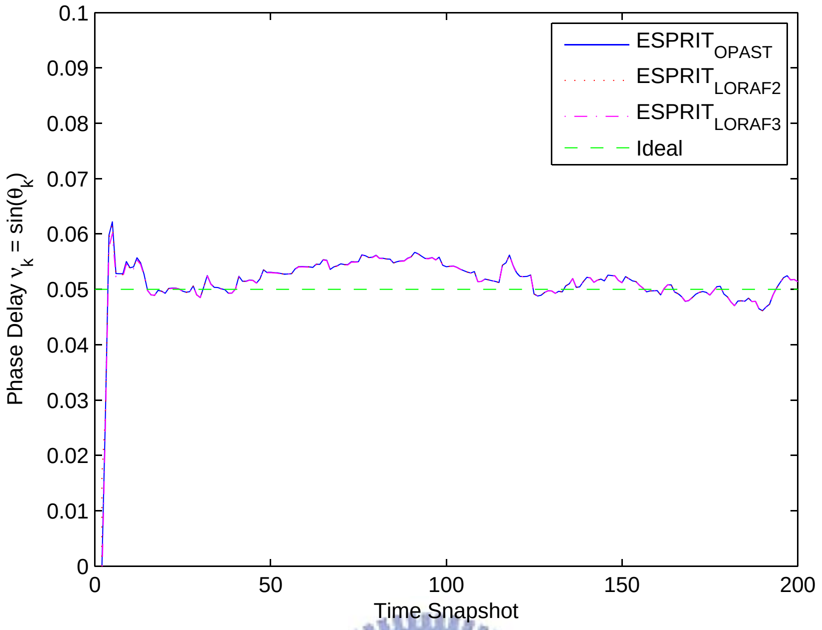

Figure 5.1: DOA estimation for one signal source with the constant phase, N = 6 , SNR = 3dB

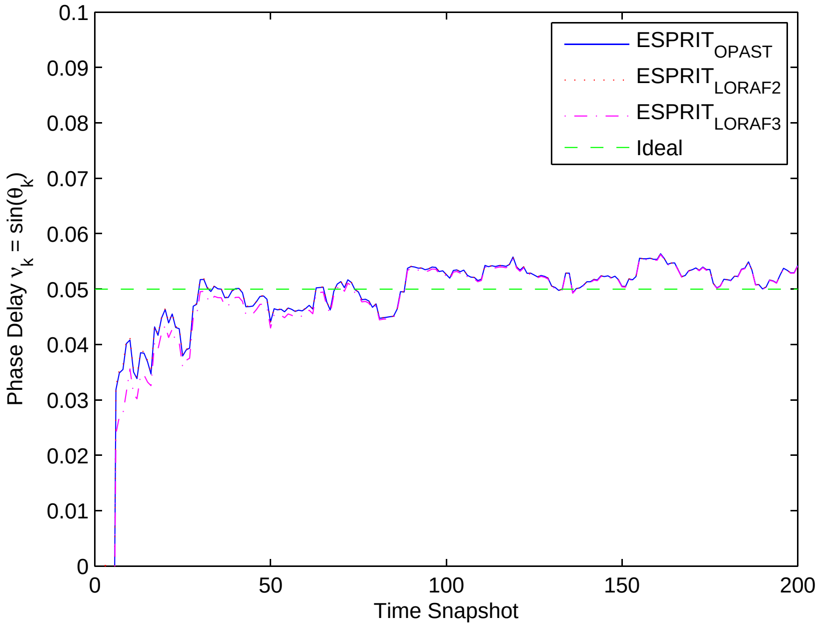

Figure 5.2: DOA estimation for one signal source with the constant phase, N = 6 , SNR = 0dB

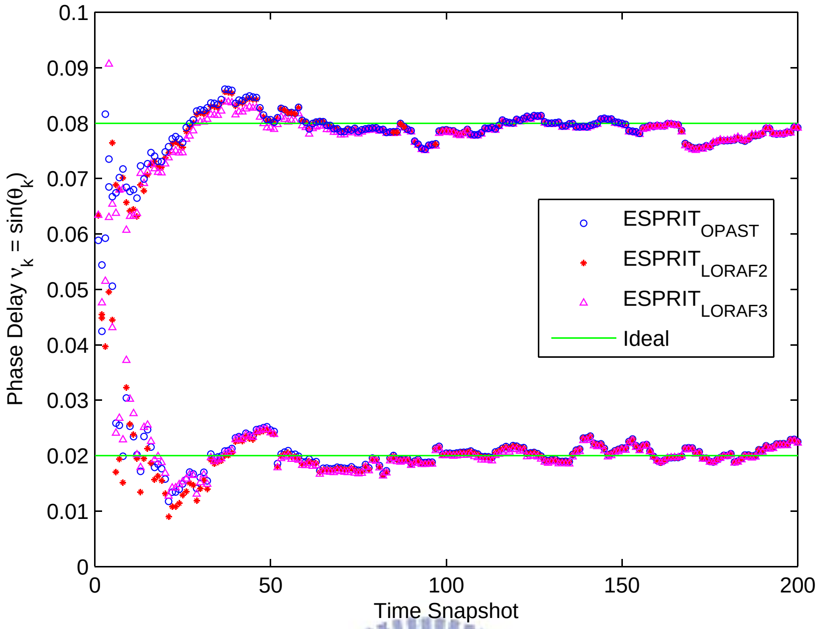

Figure 5.3: DOA estimation for two signal sources with the constant phase delays , N = 10 , SNR = 3dB

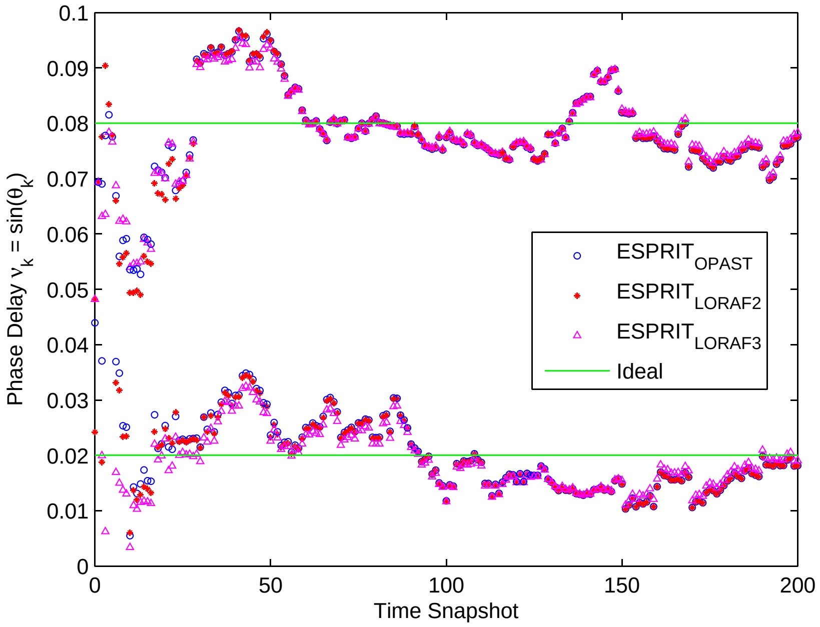

Figure 5.4: DOA estimation for two signal sources with the constant phase delays , N = 10 , SNR = 0dB

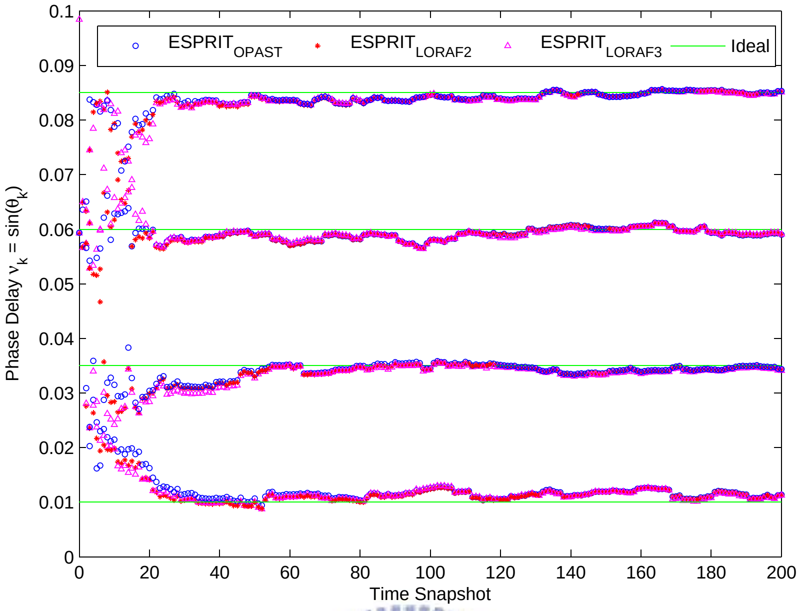

Figure 5.5: DOA estimation for four signal sources with the constant phase delays , N = 50 , SNR = 3dB

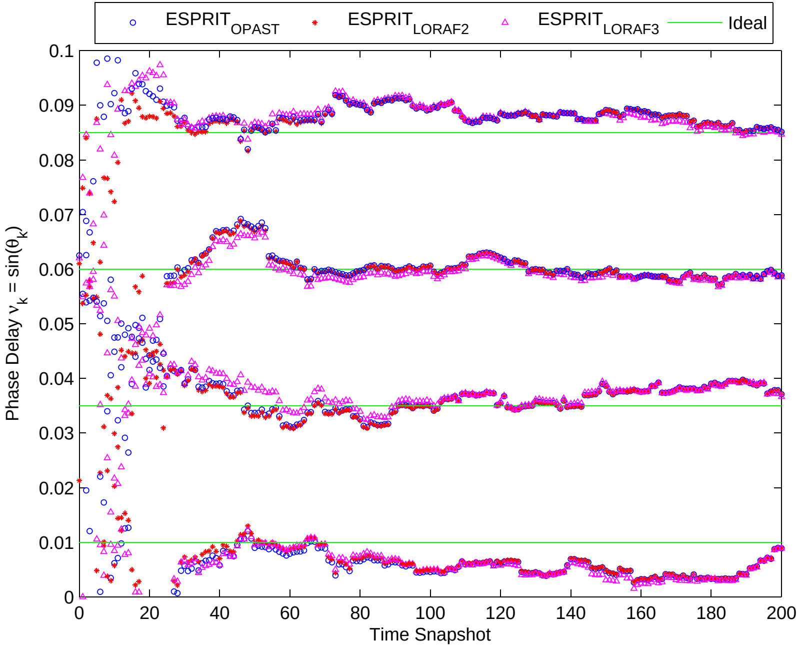

Figure 5.6: DOA estimation for four signal sources with the constant phase delays , N = 50 , SNR = 0dB

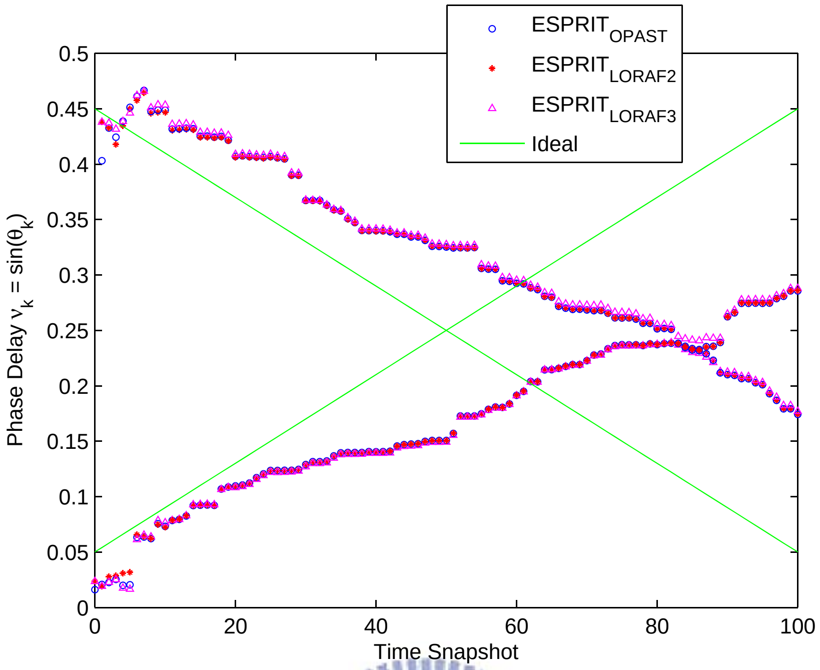

Figure 5.7: DOA estimation for two signal sources with the crossed phase delays , N = 10 , SNR = 3dB

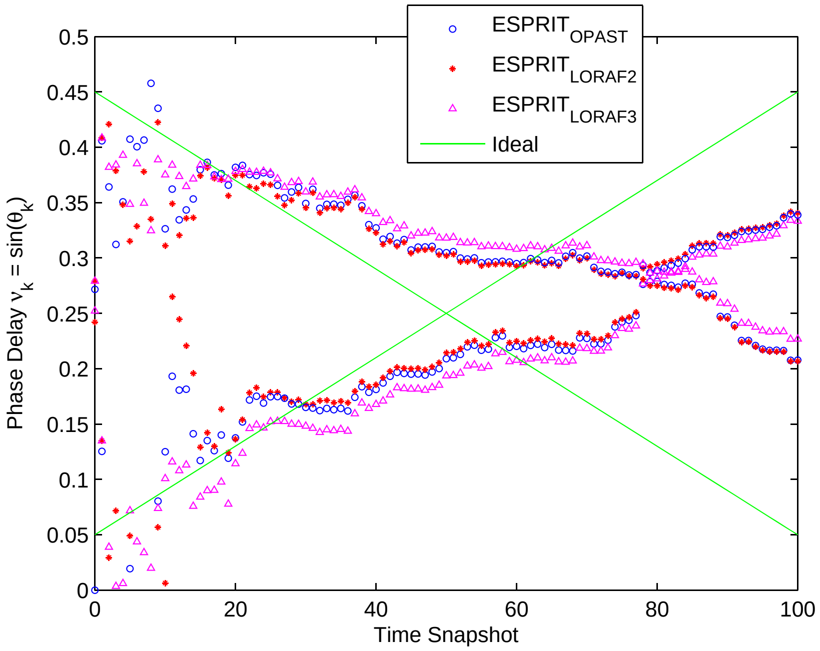

Figure 5.8: DOA estimation for two signal sources with the crossed phase delays , N = 10 , SNR = -3dB

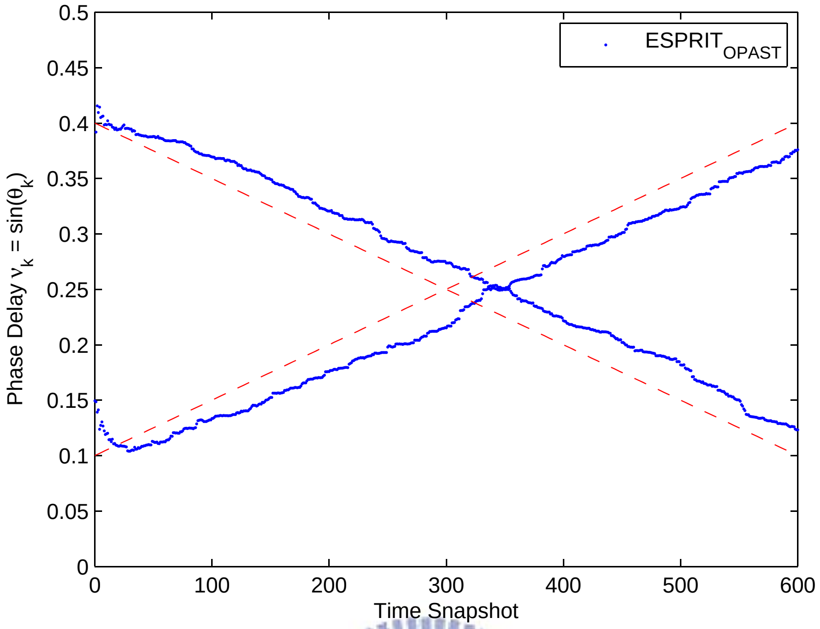

Figure 5.9: DOA estimation for two signal sources with the crossed phase delays via ESPRIT-OPAST, N = 10 , SNR = 3dB

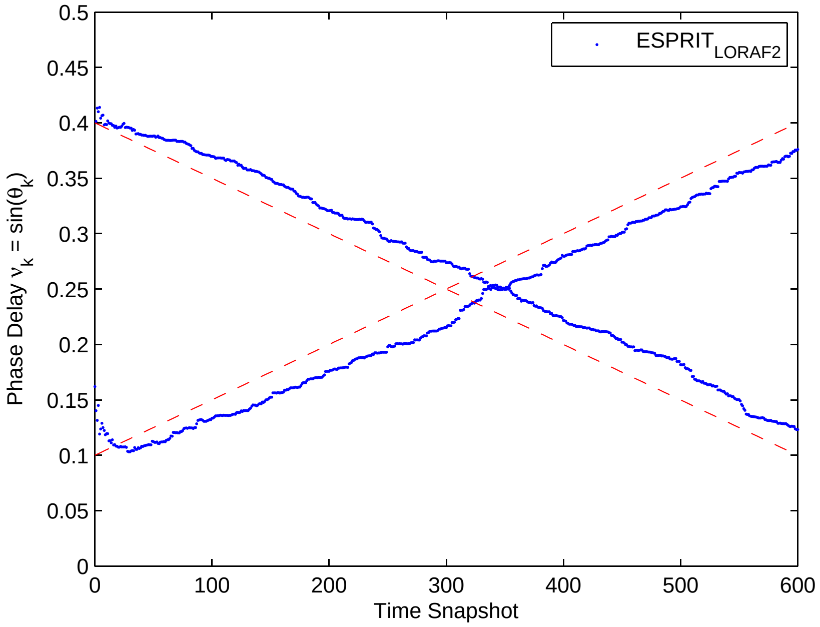

Figure 5.10: DOA estimation for two signal sources with the crossed phase delays via ESPRIT-LORAF2, N = 10 , SNR = 3dB

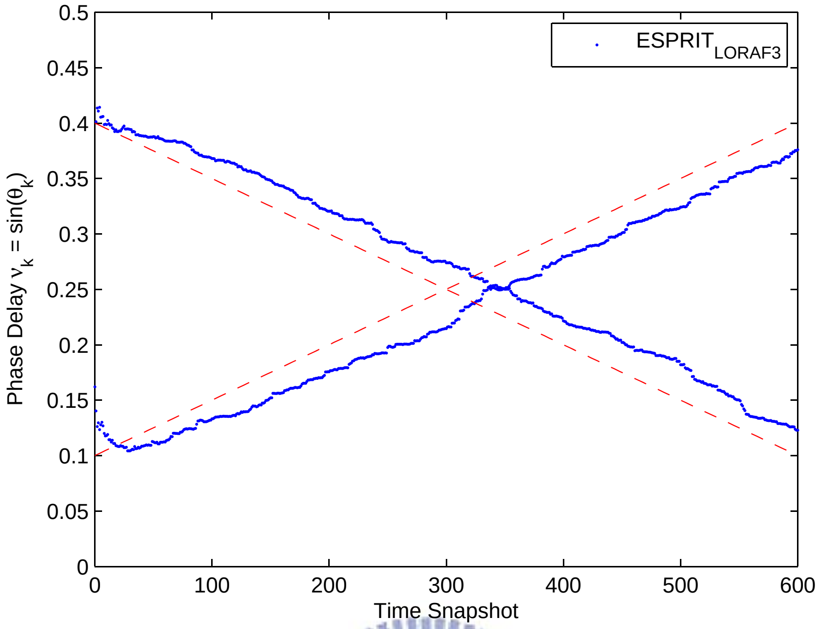

Figure 5.11: DOA estimation for two signal sources with the crossed phase delays via ESPRIT-LORAF3, N = 10 , SNR = 3dB

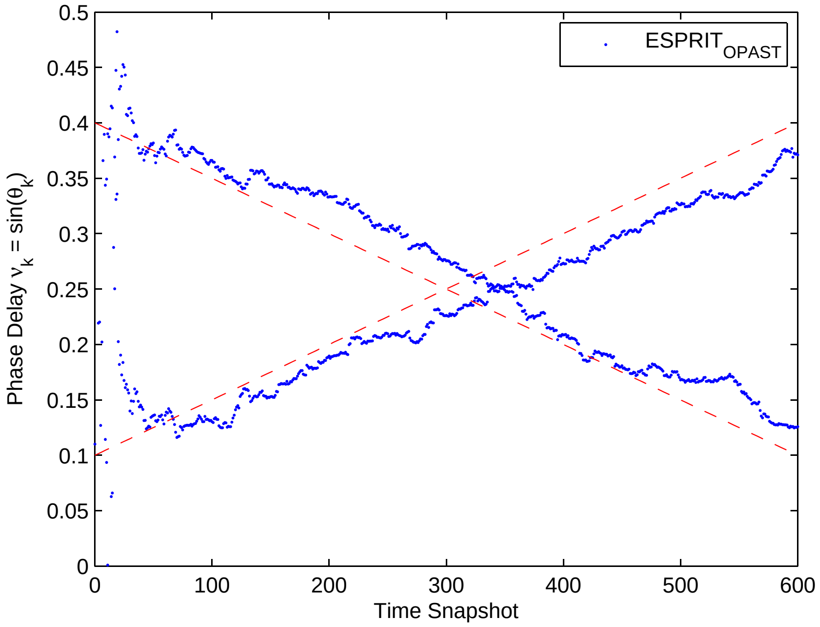

Figure 5.12: DOA estimation for two signal sources with the crossed phase delays via ESPRIT-OPAST, N = 10 , SNR = -3dB

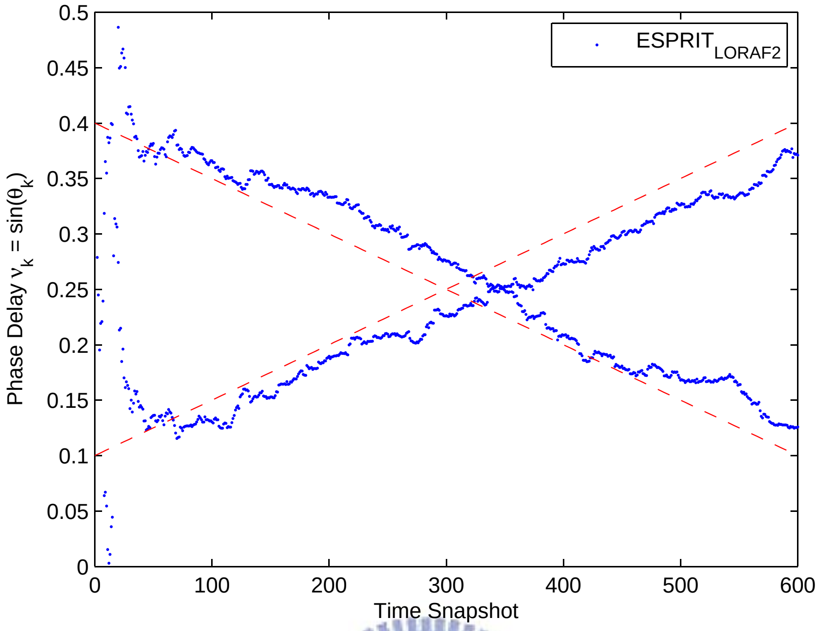

Figure 5.13: DOA estimation for two signal sources with the crossed phase delays via ESPRIT-LORAF2, N = 10 , SNR = -3dB

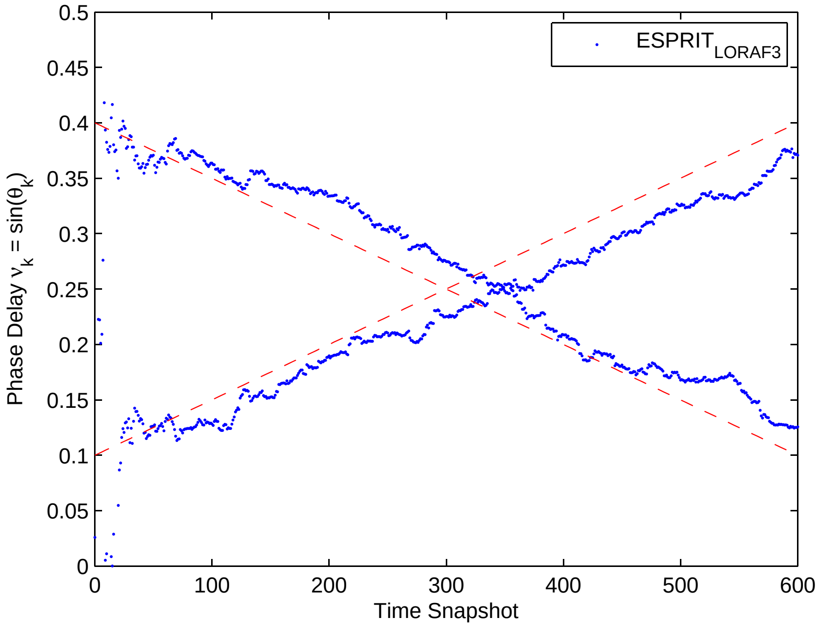

Figure 5.14: DOA estimation for two signal sources with the crossed phase delays via ESPRIT-LORAF3, N = 10 , SNR = -3dB

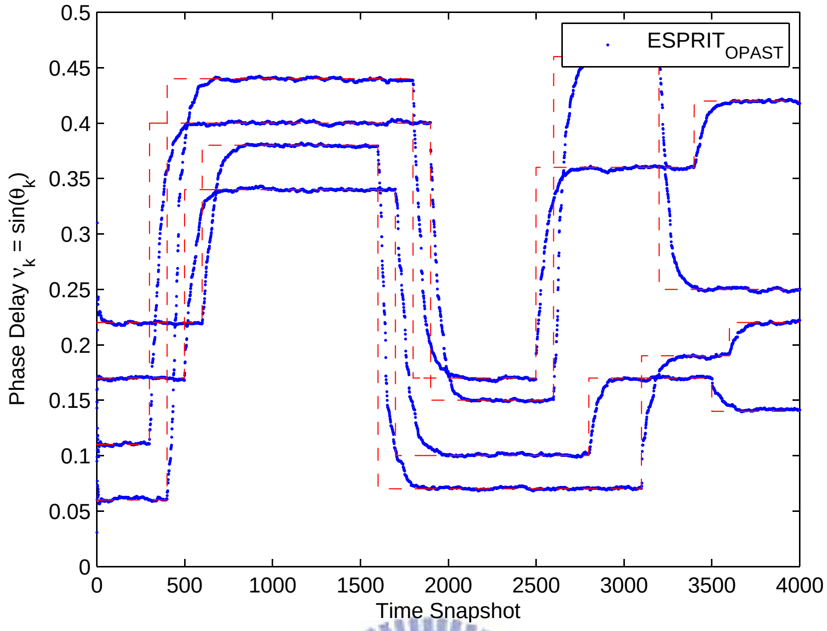

Figure 5.15: DOA estimation for four signal sources with the phase delays varies suddenly via ESPRIT-OPAST, N = 50 , SNR = 3dB

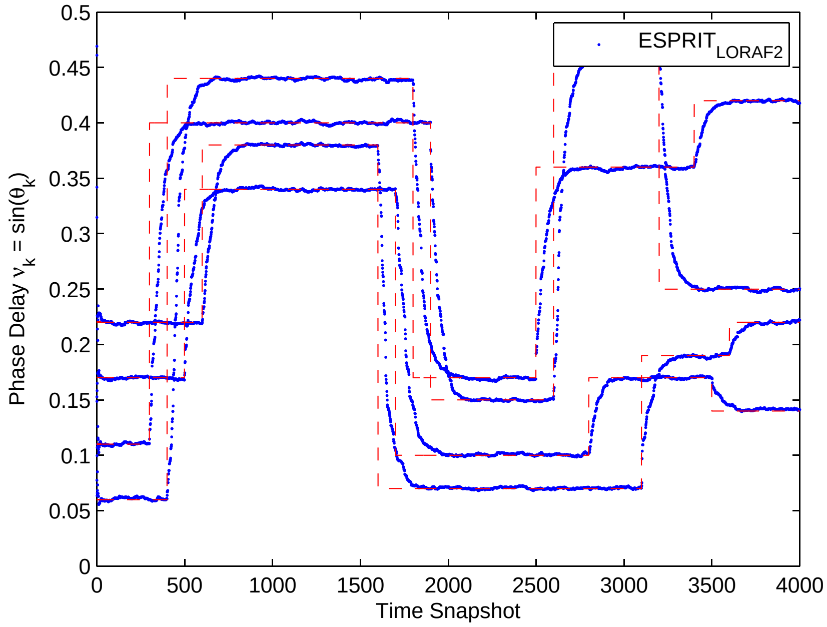

Figure 5.16: DOA estimation for four signal sources with the phase delays varies suddenly via ESPRIT-LORAF2, N = 50 , SNR = 3dB

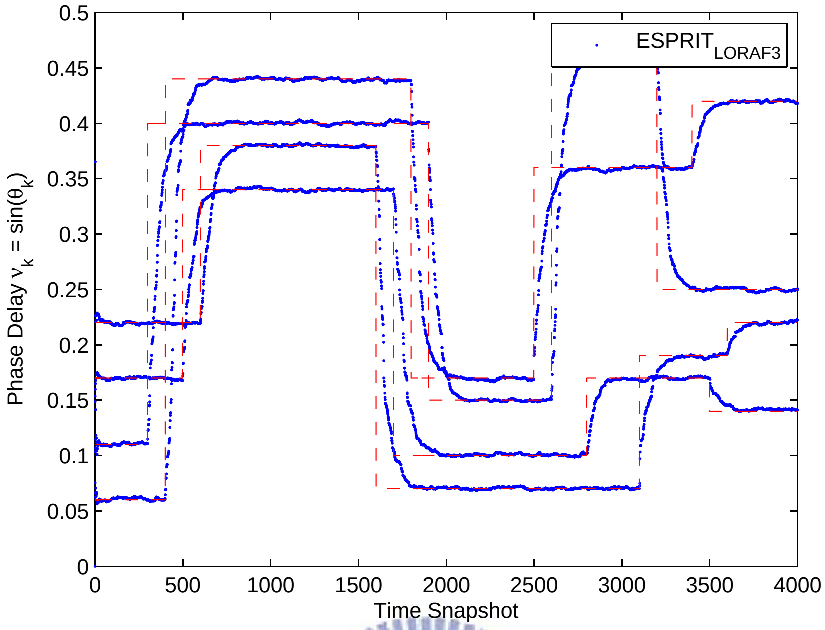

Figure 5.17: DOA estimation for four signal sources with the phase delays varies suddenly via ESPRIT-LORAF3, N = 50 , SNR = 3dB

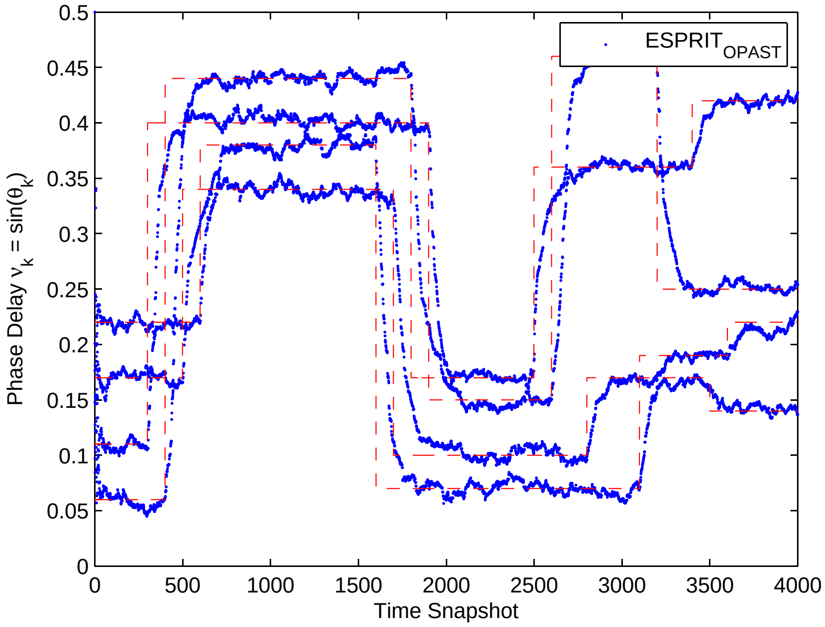

Figure 5.18: DOA estimation for four signal sources with the phase delays varies suddenly via ESPRIT-OPAST, N = 50 , SNR = -3dB

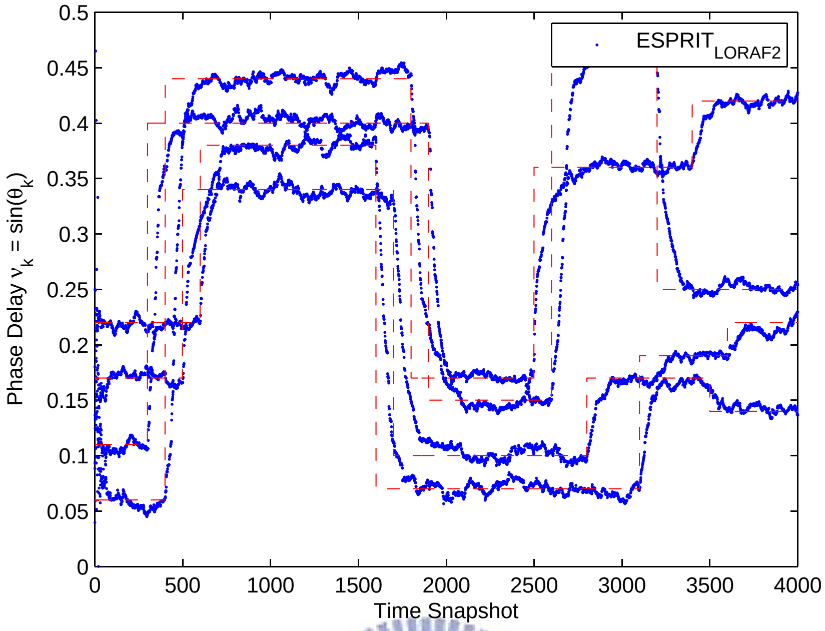

Figure 5.19: DOA estimation for four signal sources with the phase delays varies suddenly via ESPRIT-LORAF2, N = 50 , SNR = -3dB

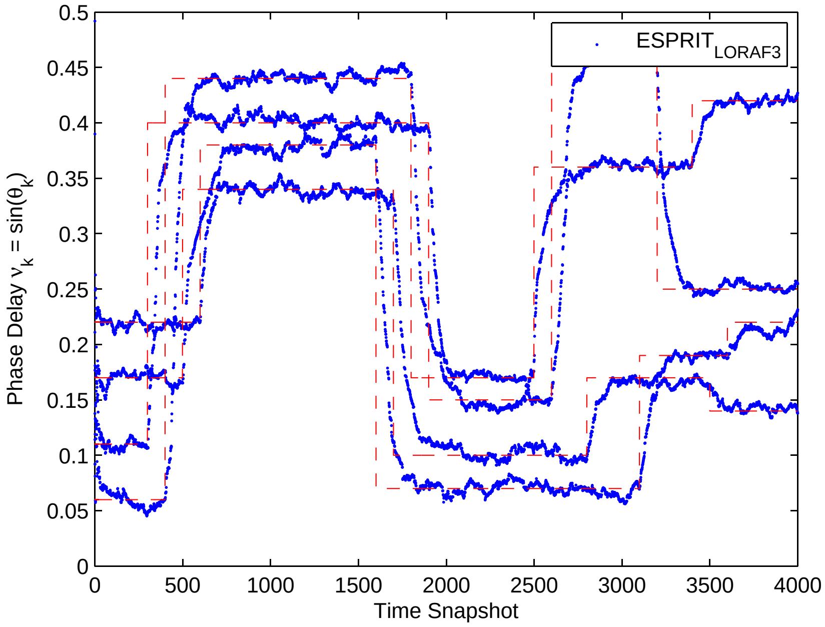

Figure 5.20: DOA estimation for four signal sources with the phase delays varies suddenly via ESPRIT-LORAF3, N = 50 , SNR = -3dB

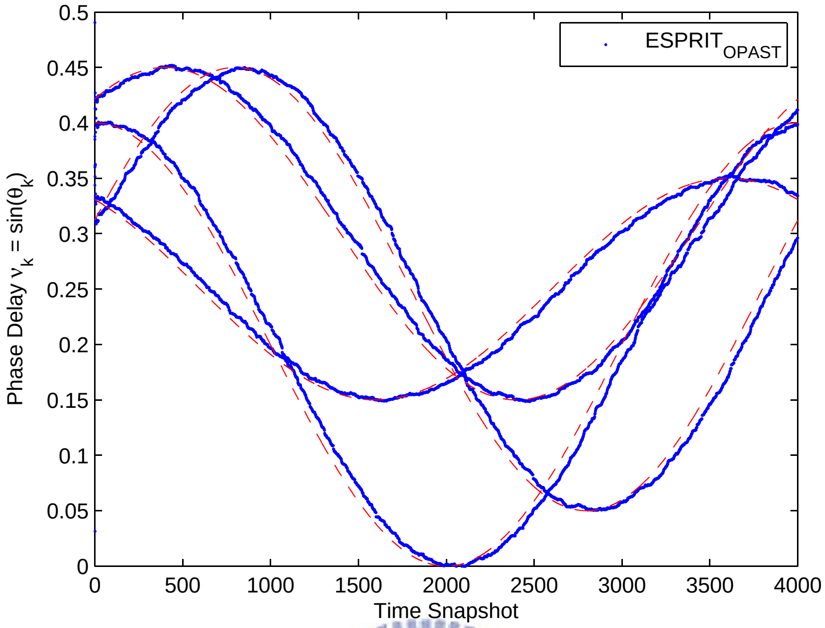

Figure 5.21: DOA estimation for four signal sources with the phase delays varies smoothly via ESPRIT-OPAST, N = 50 , SNR = 3dB

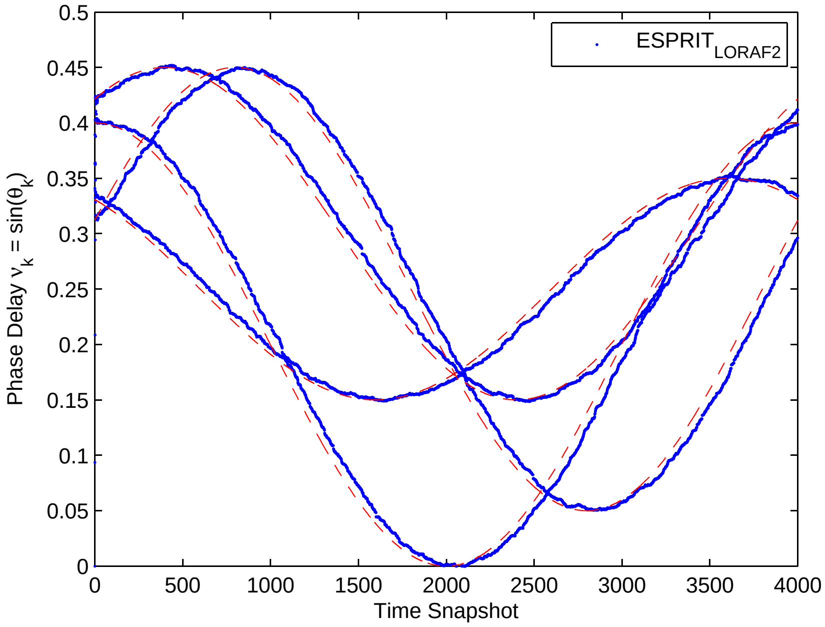

Figure 5.22: DOA estimation for four signal sources with the phase delays varies smoothly via ESPRIT-LORAF2, N = 50 , SNR = 3dB

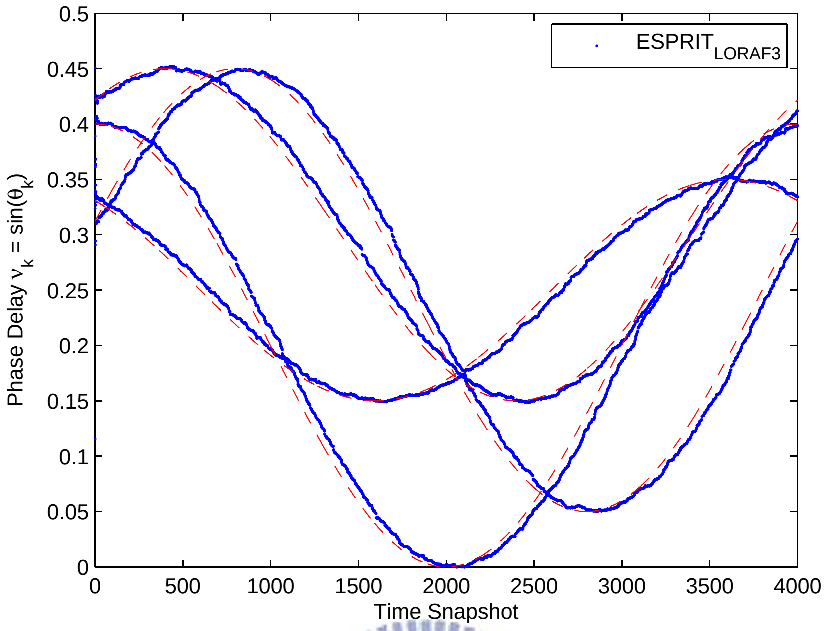

Figure 5.23: DOA estimation for four signal sources with the phase delays varies smoothly via ESPRIT-LORAF3, N = 50 , SNR = 3dB

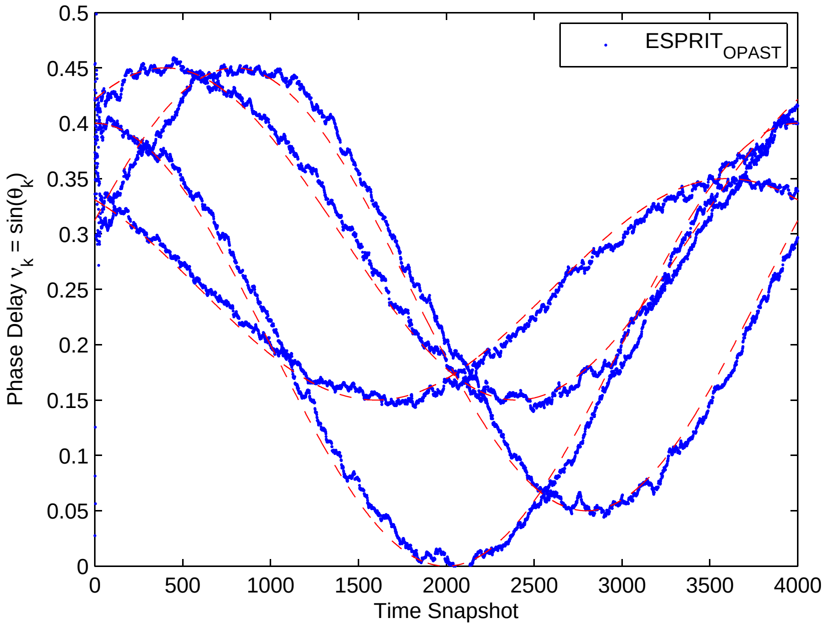

Figure 5.24: DOA estimation for four signal sources with the phase delays varies smoothly via ESPRIT-OPAST, N = 50 , SNR = -3dB

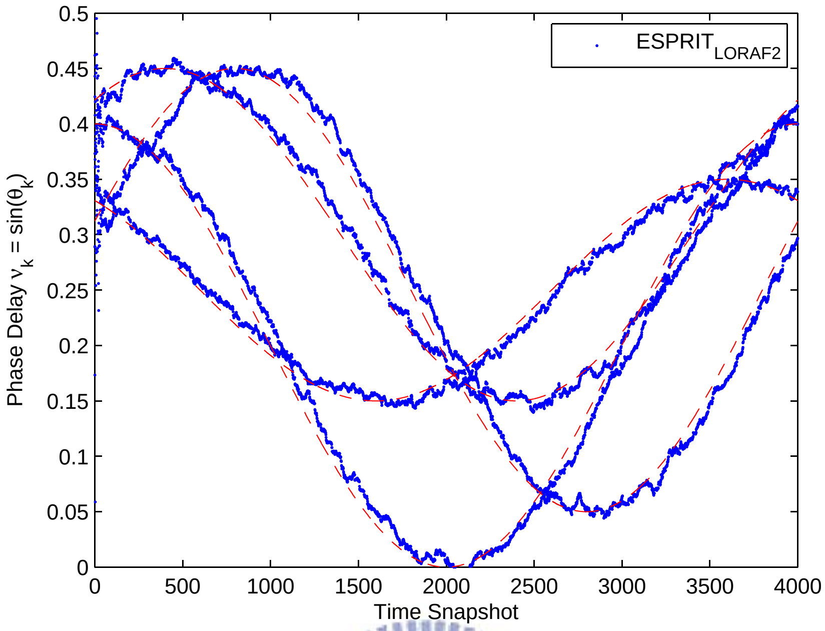

Figure 5.25: DOA estimation for four signal sources with the phase delays varies smoothly via ESPRIT-LORAF2, N = 50 , SNR = -3dB

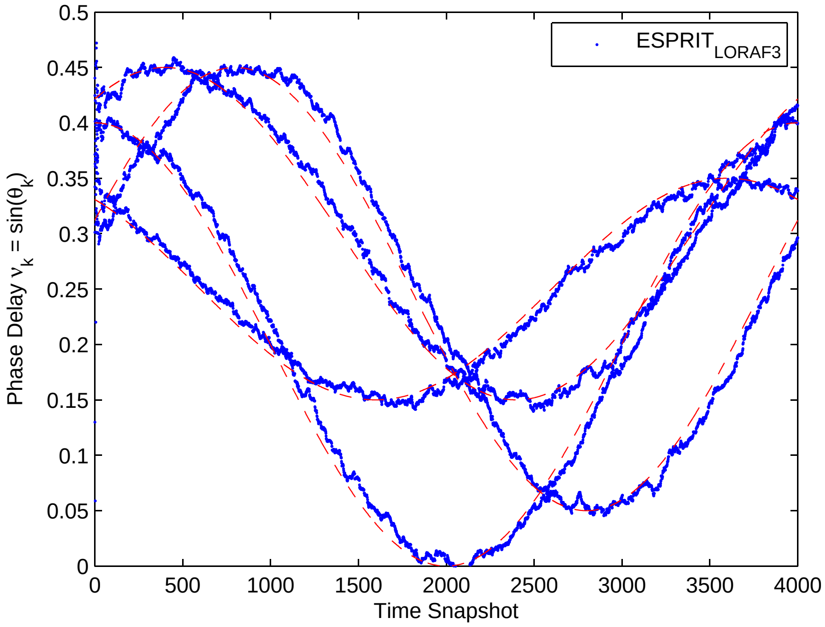

Figure 5.26: DOA estimation for four signal sources with the phase delays varies smoothly via ESPRIT-LORAF3, N = 50 , SNR = -3dB

Chapter 6

CONCLUSIONS

In this thesis, the OPAST subspace tracking is employed to develop a new and fast adaptive ESPRIT algorithm with $\mathcal{O}(Nr)$ computational complexity. The derived approach is straightforward simple and intuitive in no need of QR-reduction, sequential orthogonal iteration, and Givens plane rotation which are used in developing adaptive ESPRIT algorithms existing in literature. Our method requires only about $11Nr + 10N + \mathcal{O}(r^3)$ computational complexity every update and saves the more amounts of computations than ESPRIT-LORAF2 and ESPRIT-LORAF3. It is also examined that this method requires the least storage spaces compared with ESPRIT-LORAF2 and ESPRIT-LORAF3. Computer simulations further demonstrate that for DOA estimation our algorithm manifests nearly the same performance as both ESPRIT-LORAF2 and ESPRIT-LORAF3.

Bibliography

[1] R. Roy and T. Kailath, “ESPRIT-Estimation of Signal Parameters via Rotational Invariance Techniques,” *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 37, no. 7, pp.984-995, July 1989.

[2] B. Yang, “Projection Approximation Subspace Tracking,” *IEEE Trans. Signal Processing,* vol. 44, no. 1, pp.95-107, Jan. 1995.

[3] K. Abed-Meraim, A. Chkeif, and Y. Hua, “Fast Orthonormal PAST Algorithm,” *IEEE Signal Processing Letters,* vol. 7, no. 3, pp. 60-62, Mar. 2000

[4] P. Strobach, “Fast recursive subspace adaptive ESPRIT algorithms,” *IEEE Trans. Signal Processing,* vol. 46, no. 9, pp. 2413-2430, Sept. 1998.

[5] P. Strobach, “Low-rank adaptive filters,” *IEEE Trans. Signal Processing,* vol. 44, no. 12, pp. 2932-2947, Dec. 1996.

[6] P. Strobach, “Bi-iteration SVD subspace tracking algorithms,” *IEEE Trans. Signal Processing,* vol. 45, no. 5, pp. 1222-1240, May 1997.

[7] M. Moonen, F. J. Vanpoucke, and E. F. Deprettere, “Parallel and adaptive high-resolution direction finding,” *IEEE Trans. Signal Processing,* vol. 42, no. 9, pp. 2439-2448, Sept. 1994.

[8] A. Paulraj, R. Roy, and T. Kailath, “A subspace rotation approach to signal parameter estimation,” *Proc. IEEE.,* vol. 74, no. 7, pp. 1044-1046, July 1986.

[9] R. Roy, A. Paulraj, and T. Kailath, “ESPRIT-A subspace rotation approach to estimation of parameters ofcisoids in noise,” *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-34, pp. 1340-1342, Oct. 1986.

[10] R. O. Schmidt, “A signal subspace approach to multiple emitter location and spectral estimation,” Ph.D. dissertation. Stanford Univ., Stanford, CA, 1981.

[11] R. O. Schmidt, “Multiple Emitter Location and Signal Parameter Estimation,” *IEEE Trans. Acoust., Speech, Signal Processing,* Vol. 34, pp. 276-280, 1986.

[12] R. Kumaresan and D. W. Tufts, “Estimating the angles of arrival of multiple plane waves,” *IEEE Trans. Aerospace Electron. Syst.,* Vol. AES-19, pp. 134-139, 1983.

[13] G. W. Stewart, “Simultaneous iteration for computing invariant subspaces of non-Hermitian matrices,” *Numer. Math.,* vol. 25, pp. 123-136, 1976.

[14] Y. Hua, T. Chen, and Y. Miao, “A unifying view of a class of subspace tracking methods,” *ISSPR’ 98,* Hong Kong, vol. 2, pp. 27-32, Sept. 1998

[15] G. H. Golub and C. F. Van Loan, *MATRIX COMPUTATIONS*, Johns Hopkins University Press, 1996.

[16] S. Haykin, *Adaptive Filter Theory*, 2nd, Prentice-Hall, 1991.