

國立交通大學

電信工程學系

博士論文



應用於直接序列分碼多工系統中碼擷取之
適應性濾波技術
Adaptive Filtering Techniques for DS/CDMA
Code Acquisition

研究生：楊華龍

指導教授：吳文榕

中華民國九十五年十月

應用於直接序列分碼多工系統中碼擷取之適應性
濾波技術

Adaptive Filtering Techniques for DS/CDMA Code
Acquisition

研究生: 楊華龍

Student: Hua-Lung Yang

指導教授: 吳文榕 博士

Advisor: Dr. Wen-Rong Wu

國立交通大學

電信工程學系

博士論文



A Dissertation

Submitted to Department of Communication Engineering
College of Electrical Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Communication Engineering

October 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年十月

應用於直接序列分碼多工系統中碼擷取之適應性 濾波技術

研究生: 楊華龍

指導教授: 吳文榕

國立交通大學電信工程學系

摘要

在直接序列分碼多工系統中，碼擷取是一個很重要的課題。傳統上，相關器常被用來解決此問題。然而，其碼擷取的性能卻因多重使用者干擾而嚴重惡化。我們熟知在接收機端架設陣列天線可以有效的壓制多重使用者干擾。但是，在多天線系統下，大多數的碼擷取系統依舊是倚賴相關器。由於相關器固有的性質，使得基於相關器所設計的碼擷取系統通常擁有較長的平均碼擷取時間。最近，適應性濾波技術被應用於此問題上。儘管，此技術能提供較佳的碼擷取性能，但是它的計算複雜度卻隨著延遲不確定性變大而增加。而且，在適應性多天線系統下，有效率的碼擷取系統尚未被研究。在本論文中，我們發展新的適應性演算法來解決前述問題。

為了對付長延遲的問題，首先，我們提出了一個多率(multirate)碼擷取系統。它是由多個不同碼擷取單元所組成的，而這些單元擁有不同的信號處理速率。拜多率信號處理中抽取(decimation)性質之賜，整體的計算複雜度可以被大量降低。我們亦分析了適應性濾波器的收斂以及平均碼擷取時間。由實驗結果可以看出，當多率碼擷取系統與傳統適應性濾波碼擷取系統性能相近時，多率系統僅需非常低的計算複雜度。

在適應性多天線系統下，我們提出的系統可以遠優於傳統上基於相關器所設計的系統。提出的系統包含了兩個適應性濾波器，一個適應性空間濾波器與一個適應性時間濾波器。經由特別的設計，空間濾波器可以作為波束成型器，而時間濾波器可以

作為碼延遲的估計器。基於最小均方差的準則，我們利用隨機梯度坡降法(stochastic gradient decent method)來同時調整這兩個濾波器。我們亦仔細的分析了其碼擷取性能與收斂行為。由模擬結果可以看出：提出系統的平均碼擷取時間遠低於傳統相關器所設計的系統，以及理論的分析是正確的。

最後，我們設計了兩個演算法來優化前述的適應性多天線碼擷取系統。首先，在第一個研究裡，我們藉由連續搜尋技術，可以大量的縮短時間濾波器的長度，同時可以達到降低計算複雜度。我們對此低複雜度系統分析並得到相關的固定表示式。由模擬結果，可以看出此系統可以藉由稍微犧牲性能以大量降低計算複雜度。此外，我們發現在多路徑通道中，前述的適應性多天線碼擷取系統有收斂變慢的傾向。為了解決收斂便慢的問題，在第二個研究裡，我們提出了一個基於共軛梯度法所設計的適應性演算法。與傳統的共軛梯度法相比，所提出的方法是低計算複雜度的，這是由於我們利用輸入信號的相關性矩陣中特殊結構。由模擬結果可以看出，與傳統的共軛梯度法相比，所提出的低複雜度共軛梯度適應性演算法擁有相近的收斂速度。



Adaptive Filtering Techniques for DS/CDMA Code Acquisition

Student: Hua-Lung Yang

Advisor: Dr. Wen-Rong Wu

Department of Communication Engineering
National Chiao Tung University

Abstract

Code acquisition has been an important issue in direct-sequence/code-division multiple access (DS/CDMA) systems. The conventional solution to this problem is to use the correlator. However, the corresponding acquisition performance is significantly degraded when multiple access interference (MAI) is present. It is well-known that the receiver equipped with an antenna array can effectively suppress MAI. However, most code acquisition schemes for array systems still rely on the correlator structure. Due to the inherent property, the mean acquisition time of the correlator-based approaches is usually large. Recently, adaptive-filtering technique was applied to solve the problem. Although adaptive-filtering systems can provide better performance, its computational complexity becomes high when the delay uncertainty becomes large. Also, effective adaptive array systems for code acquisition have not been investigated yet. In this dissertation, we have developed novel adaptive algorithms solving the problems mentioned above.

To cope with the large code delay problem, we first propose a multirate acquisition system, which is comprised of several acquisition units operating in different processing rates. Thanks to the decimation property in multirate processing, the overall computational complexity can be greatly reduced. Theoretical analysis of filter convergence and mean acquisition time is also provided. Experimental results show that while the proposed scheme can have comparable performance with respect to the conventional adaptive filtering scheme, its computational complexity is much lower. We then propose an adaptive array system having superior performance than the conventional correlator-based system. The proposed scheme comprises two adaptive filters, an adaptive spatial and an

adaptive temporal filter. With a specially designed structure, the spatial filter can act as a beamformer suppressing interference, while the temporal filter can act as a code-delay estimator. A mean squared error (MSE) criterion is proposed such that these filters can be simultaneously adjusted by a stochastic gradient descent method. The performance as well as the convergence behavior of the proposed algorithm are analyzed in detail. Simulations show that the mean acquisition time of the proposed algorithm is much shorter than that of the correlator-based approach, and the derived theoretical expressions are accurate.

Finally, we develop algorithms refining the proposed adaptive array acquisition system. The first approach is to incorporate a serial-search technique. By this way, we are able to significantly reduce the size of the temporal filter, so does the computational complexity. We also analyze the proposed low-complexity system and derive related closed-form expressions. Simulations show that while the refined system slightly compromises the performance, the computational complexity is much lower. In multipath-channel environments, the convergence of the proposed adaptive array system tends to be slow. In the second approach, we propose an adaptive algorithm with the conjugate gradient (CG) method to solve the problem. Unlike the original CG method, the proposed method, exploiting the special structure inherent in the input correlation matrix, requires a low computational-complexity. Simulation results show that the performance of adaptive array code acquisition with the proposed CG method is comparable to that with the original CG method.

Acknowledgements

First, many thanks to my advisor, Professor Wen-Rong Wu, who has provided me with guidance and insightful direction over the years. During the period of time, he showed full support to me. With his valuable comments and forceful guidance, then I could finish the papers and this dissertation. His active attitude has taught me so much. I feel that I am so lucky to work with such a nice man.

Then, I appreciate all the committee members for this dissertation. Their suggestions and thoughtful comments make this work more complete and inspire me some future works.

I also would like to thank my colleagues, including some former members, generally for their friendship and company. Special thanks to Ren-Jr Chen, Chao-Yuan Hsu, Yin-man Lee, Jeff Lee, Shou-Sheu Lin, String Hsiao, and Mavin Wu, whose friendship has encouraged me over these years.

I deeply appreciate my parents for their love and support giving me strength. Also, special thanks to my elder sister, Hsiu-Min Yang, who understands me and agrees with my decision.

There are several more people whose contribution to my academic life should also be greatly appreciated. They are Pei-Yi Chu, James Kuo, Rich Chen, and An Sheen, whose care warms me.

Though words are not even enough to express my gratitude, I would thank the people who help and understand me.

Contents

| | |
|---|-------------|
| <i>Abstract</i> | ii |
| <i>Acknowledgements</i> | iv |
| <i>Contents</i> | iv |
| <i>List of Tables</i> | vii |
| <i>List of Figures</i> | viii |
| <i>Glossary</i> | xii |
| <i>Notations</i> | xv |
| 1 Introduction | 1 |
| 2 Multirate Adaptive Filtering for Low-Complexity Code Acquisition | 5 |
| 2.1 Conventional Adaptive Code Acquisition | 8 |
| 2.2 Proposed Adaptive Multirate Code Acquisition | 11 |
| 2.2.1 2R Scheme | 11 |
| 2.2.2 3R Scheme | 21 |
| 2.3 Performance Analysis | 25 |
| 2.3.1 Computational Complexity | 25 |



| | | |
|----------|--|-----------|
| 2.3.2 | Probability of Acquisition Error | 26 |
| 2.3.3 | Mean Acquisition Time | 30 |
| 2.4 | Simulations | 31 |
| 2.5 | Conclusions | 36 |
| 3 | <i>Adaptive Antenna Array Code Acquisition</i> | 39 |
| 3.1 | Correlator-Based Adaptive Array Code Acquisition | 41 |
| 3.2 | Proposed Adaptive Array Code Acquisition | 43 |
| 3.2.1 | Signal Model and Algorithm Development | 43 |
| 3.2.2 | Code Acquisition with AWGN Channel | 47 |
| 3.2.3 | Code Acquisition with Multipath Channel | 50 |
| 3.3 | Adaptive Implementation and Convergence Analysis | 54 |
| 3.3.1 | Constrained LMS and Convergence Analysis | 54 |
| 3.3.2 | Steady-state MSE Analysis | 56 |
| 3.3.3 | Output SINR of Beamformer | 58 |
| 3.4 | Performance Analysis | 59 |
| 3.4.1 | Probability of Correct Acquisition | 59 |
| 3.4.2 | Mean Acquisition Time | 62 |
| 3.5 | Simulations | 62 |
| 3.5.1 | AWGN Channel | 63 |
| 3.5.2 | Multipath Channels | 68 |
| 3.5.3 | Performance Comparison | 69 |
| 3.6 | Conclusions | 72 |
| 4 | <i>Low-Complexity Adaptive Array Code Acquisition</i> | 73 |
| 4.1 | Proposed Low-Complexity Code Acquisition | 74 |
| 4.2 | Adaptive Implementation and Convergence Analysis | 79 |
| 4.2.1 | Constrained LMS and Convergence Issue | 79 |

| | | |
|----------|--|------------|
| 4.2.2 | Steady-state MSE Analysis | 82 |
| 4.2.3 | Output SINR at Beamformer for an Inphase Cell | 83 |
| 4.3 | Performance Analysis | 84 |
| 4.3.1 | Mean Acquisition Time | 84 |
| 4.3.2 | Probabilities Derivation | 88 |
| 4.4 | Simulations | 90 |
| 4.5 | Conclusions | 96 |
| 5 | <i>Conjugate Gradient Algorithm for Adaptive Array Code Acquisition</i> | 98 |
| 5.1 | Proposed CG Adaptive Algorithm | 99 |
| 5.2 | Simulation Results | 107 |
| 5.3 | Conclusions | 112 |
| 6 | <i>Conclusions and Future Works</i> | 114 |
| | <i>Bibliography</i> | 116 |



List of Tables

| | | |
|-----|--|-----|
| 2.1 | Computational Complexity Comparison for $D = 4$ | 32 |
| 2.2 | Computational Complexity comparison for $D = 8$ | 32 |
| 2.3 | Computational Complexity Comparison for $D = 16$ | 32 |
| 3.1 | Parameters used for simulations in multipath scenario | 69 |
| 5.1 | Computational Complexity Comparison for Constrained CCG, MCG, and LCG Algorithms | 105 |



List of Figures

| | | |
|-----|--|----|
| 2.1 | Conventional 1R code acquisition system, where $x_1(n - qM_c)$ is user-1's PN sequence at q -th cell with $q = 0, \dots, Q - 1$ | 9 |
| 2.2 | Proposed 2R code acquisition system with (a) LRU and (b) HRU. Note that LRU and HRU interact only when $n = mD$. The dash-lines indicate feedforward and feedback operations. | 13 |
| 2.3 | Proposed 3R code acquisition system with (a) LRU, (b) MRU, and (c) HRU. Again, all units interact only when $n = mD$ and the dash-lines are for feedforward and feedback operations. | 23 |
| 2.4 | Markov chain model for multirate code acquisition schemes. The right hand side figure illustrates an equivalent model, where z is a delay operator, P_E the probability of acquisition error, T_p penalty time, and ACQ the correct acquisition state. | 30 |
| 2.5 | Experimental and theoretical P_E [(2.9), (2.90), and (2.96)] versus step size μ ($D = 8$, $D_M = 4$, $U = 128$, and $\text{SINR}_c = -13$ dB). | 33 |
| 2.6 | Experimental mean acquisition time T_{acq} versus step size μ ($T_p = 1.28 \times 10^4$, $D = 8$, $D_M = 4$, and $\text{SINR}_c = -13$ dB). | 34 |
| 2.7 | Experimental P_E versus SINR_c ($\mu = 5 \times 10^{-3}$). | 35 |
| 2.8 | Experimental mean acquisition time T_{acq} versus SINR_c ($\mu = 5 \times 10^{-3}$). | 36 |
| 2.9 | Experimental mean acquisition time T_{acq} versus N (expressed as a multiple of time-constant) and step size μ | 37 |

| | | |
|------|---|----|
| 3.1 | Correlator-based adaptive array code acquisition system. Note that $\mathbf{w}_c(N_c - 1) \triangleq [w_{c,0}, \dots, w_{c,M-1}]^T$. | 42 |
| 3.2 | Proposed adaptive array code acquisition system. | 44 |
| 3.3 | Convergence curve for squared temporal filter tap-weights, $ w_{t,p}(n) ^2$ and $ w_{t,p+2}(n) ^2$. Theoretical values are shown with horizontal lines and obtained from (3.39). | 64 |
| 3.4 | Convergence curve for MSE. Theoretical values are obtained from (3.74). | 64 |
| 3.5 | Convergence curve for SINR. Theoretical values are obtained from (3.78) and (3.79). | 65 |
| 3.6 | MSE for fractional delay estimate in AWGN and multipath channel. | 65 |
| 3.7 | Probability of correct acquisition versus step size. Theoretical values are obtained from (3.92) and (3.97). | 67 |
| 3.8 | Experimental mean acquisition time (in chips) versus step size for AWGN channel. | 67 |
| 3.9 | Experimental mean acquisition time (in chips) versus N and step size for AWGN channel. | 68 |
| 3.10 | Experimental and theoretical beam-patterns for multipath channel. Arrow signs indicate DoAs of all users; the labeled are DoAs of the desired user. | 70 |
| 3.11 | Mean acquisition time (in chips) comparison. | 71 |
| 3.12 | Mean acquisition time (in chips) versus M (size of antenna array). | 71 |
| 4.1 | System diagram of the proposed system, where $Z^q = \ \mathbf{w}_s^q(N)\ ^2$, $q = 0, \dots, Q-1$. | 75 |
| 4.2 | Circular state diagram of the proposed system. | 85 |
| 4.3 | Simplified state diagram. | 85 |
| 4.4 | Convergence curve for MSE when $\mu = 3 \times 10^{-3}$ and $M_t = 8$. Theoretical value is from (4.43). | 91 |
| 4.5 | Convergence curve for SINR of $\gamma(n)$ at an inphase cell, $\mu = 3 \times 10^{-3}$, and $M_t = 8$. Theoretical value is from (4.48). | 91 |

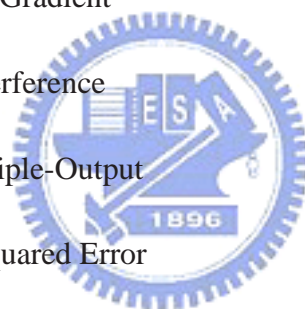
| | | |
|------|---|-----|
| 4.6 | Experimental and theoretical beam-patterns for an inphase cell ($N = 2000$, $M_t = 8$, and $M = 8$). Arrow signs indicate the DoAs associated with all users. The labeled one, ϕ_1 , is the DoA of the desired user. | 92 |
| 4.7 | Convergence curve for squared temporal filter-weights $ w_{t,j}^\nu(n) ^2$ for $j = 0, \dots, M_t - 1$ and $M_t = 8$. Theoretical value is obtained from (4.19). | 92 |
| 4.8 | Experimental and theoretical probabilities of outphase false alarm P_{Fo} versus threshold ζ | 93 |
| 4.9 | Experimental and theoretical probabilities of P_C versus threshold ζ | 94 |
| 4.10 | Experimental mean acquisition time (chips) versus ζ | 95 |
| 4.11 | Mean acquisition time comparison for $U = 256$ | 96 |
| 5.1 | Ratio of the computational complexity of MCG and LCG versus U and M ($U = 2^x, x \in \{3, 4, \dots, 9\}$). | 106 |
| 5.2 | Learning curves for constrained CG algorithms ($N_r = U + M$). | 108 |
| 5.3 | Learning curves for constrained LMS algorithm. | 108 |
| 5.4 | Convergence curves of $ w_{t,\tau_3}(n) ^2$ for constrained CG algorithms ($N_r = U + M$). | 110 |
| 5.5 | Convergence curves of $ w_{t,\tau_3}(n) ^2$ for constrained LMS algorithm. | 110 |
| 5.6 | Convergence curve of SINR for constrained CG algorithms ($N_r = U + M$). | 111 |
| 5.7 | Beam-patterns for constrained CG algorithms. Arrow signs indicate DoAs of all users; labeled ones are DoAs of the desired user. | 111 |
| 5.8 | Effect of N_r on convergence of $ w_{t,\tau_3}(n) ^2$ | 112 |
| 5.9 | Probability of correct acquisition ($N_r = U + M$). | 113 |

Glossary

| | |
|----------------|---|
| 1R | One-Rate |
| 2R | Two-Rate |
| 3G | Third Generation |
| 3R | Three-Rate |
| AWGN | Additive White Gaussian Noise |
| BPSK | Binary Phase Shift Keying |
| CCG | Conventional Conjugate Gradient |
| CG | Conjugate Gradient |
| DoA | Direction-of-Arrival |
| DPTF | Delay- and phase-Tuning Filter |
| DS/CDMA | Direct-Sequence/Code-Division Multiple Access |
| DTF | Delay-Tuning Filter |
| HRAF | High-Rate Adaptive Filter |
| HRU | High-Rate Unit |



| | |
|---------------|---|
| i.i.d. | Independent and Identically Distributed |
| LCG | Low-Complexity Conjugate Gradient |
| LPF | Lowpass Filter |
| LRAF | Low-Rate Adaptive Filter |
| LRU | Low-Rate Unit |
| LSML | Large Sample Maximum Likelihood |
| LMS | Least-Mean-Square |
| MCG | Modified Conjugate Gradient |
| MAI | Multiple Access Interference |
| MIMO | Multiple-Input-Multiple-Output |
| MMSE | Minimized Mean-Squared Error |
| MRAF | Medium-Rate Adaptive Filter |
| MRU | Medium-Rate Unit |
| MSE | Mean Squared Error |
| MUSIC | Multiple Signal Classification |
| PDM | Projection Degree Measurement |
| PTF | Phase-Tuning Filter |
| PN | Pseudo-Noise |
| RLS | Recursive Least Square |



SINR Signal to interference-plus-Noise Ratio

SNR Signal-to-Noise Ratio

SRRC Squared-Root-Raised-Cosine

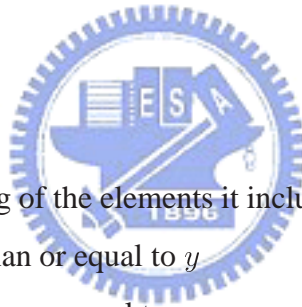
TF Transfer Function



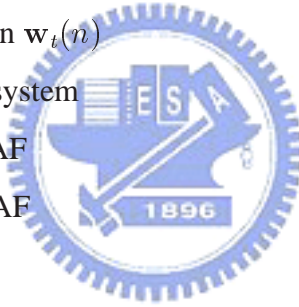


Notations

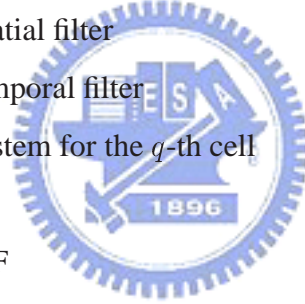
| | |
|------------------------|---|
| $E\{\cdot\}$ | Statistical expectation operator |
| $\ \cdot\ $ | Vector norm |
| i | $\sqrt{-1}$ |
| $(\cdot)^*$ | Complex-conjugate operator |
| $(\cdot)^T$ | Transpose operator |
| $(\cdot)^H$ | Hermitian operator |
| $\text{tr}\{\cdot\}$ | Trace operator |
| $\text{diag}\{\cdot\}$ | Diagonal matrix consisting of the elements it includes |
| $\lceil y \rceil$ | Smallest integer greater than or equal to y |
| $\lfloor y \rfloor$ | Largest integer smaller than or equal to y |
| $\mathbf{A}_{u,v}$ | uv -th entry of a matrix \mathbf{A} |
| U | Code delay uncertainty/Spread-spectrum processing gain |
| D | Decimation factor of LRU |
| D_M | Decimation factor of MRU |
| Q | Number of code-delay cell |
| K | Number of users |
| M | Number of array |
| N | Number of iteration |
| L_k | Number of multipath for k -th user's channel |
| τ_k | Code-delay of the k -th user |
| A_k | Amplitude of the k -th user's signal in AWGN channels |



| | |
|------------------|--|
| $p(t)$ | Rectangular chip-pulse |
| M_c | Filter-length of 1R system |
| M_p | Filter-length of LRAF |
| $r(n)$ | Received signal |
| $e(n)$ | Error signal |
| J_{min} | MMSE |
| $J_{ex}(\infty)$ | Excess MSE |
| J_{ss} | Steady-state MSE |
| ξ | Lagrange multiplier |
| ζ | Threshold |
| w_c | Tap-weight corresponding to the code-delay of the desired user |
| $w_{t,j}(n)$ | j -th tap-weight in $\mathbf{w}_t(n)$ |
| μ_c | Step size of 1R system |
| μ_L | Step size of LRAF |
| μ_H | Step size of HRAF |
| $Q(\cdot)$ | Q-function |
| $\Gamma(\cdot)$ | Gamma-function |
| $I_x(\cdot)$ | x -th order modified Bessel function of the first kind |
| σ_v^2 | Variance of $v(n)$ |
| σ_w^2 | Variance of $w(n)$ |
| $\phi_{k,l}$ | DoA for the l -th channel path of the k -th user |
| $\alpha_{k,l}$ | Channel gain for the l -th channel path of the k -th user |
| T_{acq} | Mean acquisition time |
| T_p | Penalty time |
| $H_{acq}(\cdot)$ | Transfer function of a Markov chain model |
| ACQ | State of correct acquisition |
| FA | State of false alarm |
| s_q | q -th state on a Markov chain model |



| | |
|-----------------------------|--|
| P_M | Probability of missing |
| P_{Fo} | Probability of outphase false alarm |
| P_{Fi} | Probability of inphase false alarm |
| \mathbf{f}_L | Filter-weight of LPF in LRU |
| \mathbf{f}_M | Filter-weight of LPF in MRU |
| $\mathbf{w}^q(n)$ | Filter-weight of 1R system for the q -th cell |
| $\mathbf{w}_L(m)$ | Filter-weight of LRAF |
| $\mathbf{w}_M(s)$ | Filter-weight of MRAF |
| $\mathbf{w}_H(n)$ | Filter-weight of HRAF |
| $\mathbf{w}_s(n)$ | Filter-weight of spatial filter |
| $\mathbf{w}_t(n)$ | Filter-weight of temporal filter |
| $\mathbf{w}_{s,o}$ | Optimal-weight of spatial filter |
| $\mathbf{w}_{t,o}$ | Optimal-weight of temporal filter |
| $\mathbf{x}^q(n)$ | Input-vector of 1R system for the q -th cell |
| $\mathbf{x}_L(m)$ | Input-vector of LRAF |
| $\mathbf{x}_M(s)$ | Input-vector of MRAF |
| $\mathbf{x}_H(n)$ | Input-vector of HRAF |
| $\mathbf{m}_{(U \times 1)}$ | mean vector of dimension $U \times 1$ |
| $\mathbf{r}(n)$ | Received signal vector |
| \mathbf{a} | Steering vector of the desired user |
| $\mathbf{a}_{k,l}$ | Steering vector of the l -th path for k -th user |
| $\mathbf{I}_{(U \times U)}$ | Identity matrix of dimension U |
| $\mathbf{C}_{(U \times U)}$ | Covariance matrix of dimension U |
| \mathbf{R}_x | Correlation matrix $\mathbf{x}(n)$ |
| \mathbf{R}_x^{-1} | Matrix inversion of \mathbf{R}_x |



Chapter 1

Introduction

DIRECT-sequence/code-division multiple access (DS/CDMA) is a promising multiple access technique for wireless mobile communication. However, one problem associated with the approach is the presence of multiple access interference (MAI). It has been shown that MAI is the main performance bottleneck for CDMA systems. Unfortunately, MAI not only affects data detection, but also code synchronization, which is the first step that CDMA systems have to perform. Code synchronization can be further divided into code acquisition and code tracking. Code acquisition can be considered as a coarse code synchronization process, aligning the received signal and the local code sequence with an error less than a chip duration. After successful code acquisition, other operations such as channel estimation, code tracking, and data detection can follow. Thus, code acquisition is a critical task in DS/CDMA systems. In this dissertation, we consider code acquisition in MAI environments.

Code acquisition has been widely studied in the literature. The conventional approach to this problem is the well-known correlator-based method. However, the correlator-based method is only optimal for the single-user case. The acquisition performance degrades greatly when MAI is present, especially in near-far environments. It has been shown that the acquisition-based capacity of the correlator system is less than the bit-error-rate-based capacity. The acquisition-based capacity is a performance measure for an acquisition scheme, defined as the maximum

number of users that a system can serve with certain acquisition performance. This implies that code acquisition may become a limiting factor for a CDMA system capacity. It is well-known that antenna array can significantly improve the performance of a receiver. Still, code acquisition in array systems is usually solved with correlator-based methods. Due to the inherent property of the associated serial-search scheme, its mean acquisition time is large, especially in strong MAI environments.

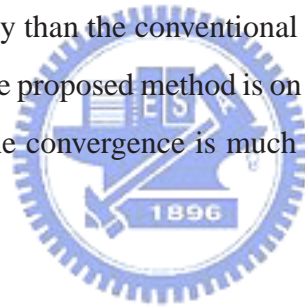
Recently, adaptive-filtering technique was proposed to solve the acquisition problem. It has been shown that the acquisition-based capacity associated with this approach can be enhanced significantly. However, its computational complexity may become higher when the delay uncertainty becomes large. Also, array systems employing the adaptive-filtering technique for code acquisition have not been investigated yet. This motivates us to develop novel adaptive algorithms solving the problems mentioned above.

In Chapter 2, we first propose a multirate adaptive-filtering scheme to deal with the large code delay problem. Compared with the conventional adaptive-filtering scheme, the multirate system requires much lower computational complexity. The proposed system comprised of several acquisition units operating in different processing rates. Theoretical analysis of the multirate adaptive filters and the mean acquisition time of the acquisition system is also investigated.

Then, we consider adaptive code acquisition with an array system, and propose a novel adaptive array code acquisition scheme in Chapter 3. The proposed system has two adaptive filters, spatial and temporal filters. The former, acting as a beamformer, can suppress MAI and the latter can simultaneously estimate the code-delays of the desired user. A mean squared error (MSE) criterion is proposed such that these filters can be simultaneously adjusted by a stochastic gradient descent method. The performance as well as the convergence behavior of the proposed algorithm are analyzed in detail. Closed-form expressions for optimum filter weights, optimum beamformer signal-to-interference-plus-noise ratio (SINR), steady-state MSE, and mean acquisition time are derived for the additive white Gaussian noise (AWGN) channel. It is shown that

the proposed system can significantly outperform the conventional correlator-based acquisition system.

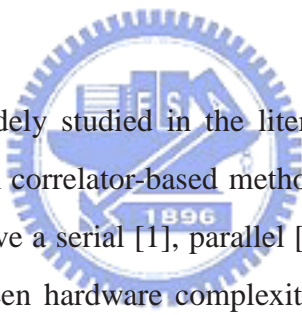
The computational complexity of temporal filter in the proposed array system may become higher when the delay uncertainty is large. To solve this, we employ the serial-search technique to develop a low-complexity system. This is described in Chapter 4. Detail analysis and experimental simulations are also made. Apart from the complexity issue, in the adaptive array code acquisition system, we find that the filter convergence is slow in multipath channel environments. Although the recursive-least-squares (RLS) algorithm can be applied, the computational complexity will be greatly increased. To alleviate this, we employ the conjugate gradient (CG) algorithm to accelerate the convergence in Chapter 5. Taking the advantage of the special structure in the correlation matrix of the input signal, we proposed a CG algorithm having much lower computational complexity than the conventional CG algorithm. It can be shown that the computational complexity of the proposed method is on the same order of the least-mean-square (LMS) algorithm. However, the convergence is much faster. Finally, we draw conclusions in Chapter 6.





Chapter 2

Multirate Adaptive Filtering for Low Complexity Code Acquisition



Code acquisition has been widely studied in the literature. The conventional approach to this problem is the well-known correlator-based method [1]– [10], [32], [36] (and references therein). The correlator can have a serial [1], parallel [2]– [4], or hybrid search structure providing an easy trade-off between hardware complexity and acquisition time. However, the correlator-based method is only optimal for the single-user case. The acquisition performance degrades greatly when MAI presents, especially in near-far environments [5]– [6]. To evaluate the performance of an acquisition scheme, a measure called acquisition-based capacity was defined in [7]. This capacity corresponds to the maximum number of users that a system can serve (with certain acquisition performance). It was shown in [7] and [19] that the asymptotic acquisition-based capacity for the correlator is $U/[2 \ln(U)]$, where U is the processing gain. The quantity is less than the bit-error-rate-based capacity [8] which is proportional to U . This implies that code acquisition may become a limiting factor for a CDMA system capacity. Another discussion on the acquisition-based capacity for the correlator can be found in [9].

Another category of the acquisition technique employs subspace- or matrix-based methods [11]– [18]. The advantage of subspace-based approaches is that it does not require train-

ing sequences. However, these methods usually have to estimate, decompose, and inverse the autocorrelation matrix of the received signal vector. This often demands high computational complexity, especially at a large processing gain. The projection degree measurement (PDM) algorithm [11] observes two successive symbols in order to obtain the complete information of one desired symbol. As a consequence, the PDM has to estimate and inverse an autocorrelation matrix of dimension $2U$ -by- $2U$. The multiple signal classification (MUSIC) algorithm has also been applied to code acquisition [12]– [14]. The MUSIC algorithm has to carry out eigen-decompositions and extract eigen-vectors corresponding to noise subspace. Despite of the oversampling operation in [12], the computational complexity of the MUSIC algorithm is with $\mathcal{O}(U^3)$. Besides, this algorithm is constrained under $2K < U$, where K is the number of users. A matrix-based method [18] called a large sample maximum likelihood (LSML) acquisition algorithm, provides excellent performance and robustness against the near-far problem. However, it requires a large amount of received bit signals and, again, pays a high computational complexity in the matrix operations. Notably, these methods are specifically designed for CDMA systems with periodic spreading codes (i.e., the spreading code repeats itself for every bit) and may not straightforwardly apply to the aperiodic-code systems (i.e., the periodicity of the spreading code is great than a bit interval).

Recently, the adaptive filter technique [19]– [26] was proposed to solve the acquisition problem in the presence of MAI. The method [19]– [24] separates the delay uncertainty into several regions, named (delay) cells. The input to the adaptive filter is the desired user's pseudo-noise (PN) sequence with a code delay associated to a cell. Each cell is then sequentially tested and the code delay can then be estimated with the location of the maximum convergent tap-weight. This method can also have a serial or parallel searching structure trading performance with computational complexity. It was addressed in [19]– [20] that the adaptive filtering scheme can have a much higher acquisition-based capacity than the correlator. Apart from the maximum weight testing, architectures with the threshold testing were also considered [21]– [22]. The threshold can be set for the mean-squared error (MSE) or for the maximum tap-weight (in a cell). It was

found in [23] that the tap-weight testing can bring better performance than the MSE testing. The acquisition performance with fading channels was analyzed and reported in [24]. Yet, another adaptive receiver structure reported in [26] performs an exhaustive search to find the integer chip delay, and then solve a quadratic equation to find the corresponding fractional chip delay. The drawback of this approach is that its complexity is high particularly for a large processing gain.

In this chapter, we propose a code acquisition algorithm using a multirate adaptive filtering technique. Similar to the original adaptive filter approach [19]– [20], our structure is valid for periodic as well as for aperiodic spreading codes. In fact, many commercial CDMA systems, including IS-95 standard [28], CDMA-2000 proposal [29], and 3G CDMA-based wireless networks [30]– [31], adopt aperiodic codes for spreading. The fundamental structure of the proposed algorithm is similar to that in [19]– [20]; however, the proposed scheme contains several adaptive filters operating in different rates. The adaptive filters with low rates will search the code delay in low resolutions. The adaptive filters with higher rates will then resolve the code delay in higher resolutions. The adaptive filter with the highest rate, say the chip-rate, can finally identify the original code delay. The proposed multirate processing can have a much lower computational complexity than the conventional adaptive filtering approaches in [19]– [20]. This is particularly true in the applications where the processing gain as well as the delay uncertainty is large.

The rest of this chapter is organized as follows. Section 2.1 reviews the conventional adaptive code acquisition scheme. Section 2.2 describes the proposed multirate code acquisition scheme. Section 2.3 analyzes the performance of the proposed scheme, and Section 2.4 reports simulation results. Finally, we draw conclusions in Section 2.5.

§ 2.1 Conventional Adaptive Code Acquisition

In this section, we briefly review the conventional adaptive code acquisition scheme [19]– [20]. Fig. 2.1 shows the structure of this scheme. For reference convenience, we name this scheme as a one-rate (1R) scheme since only one processing rate (i.e., chip-rate) is used. The baseband chip-rate sampled received signal can be expressed as

$$r(n) = \sum_{k=1}^K A_k x_k(n - \tau_k) + w(n), \quad (2.1)$$

where K , τ_k , A_k , $x_k(n)$, and $w(n)$ denotes the number of user, the code delay, the signal amplitude, the transmitted signal of user- k , and channel noise, respectively. The channel noise is assumed to be additive white Gaussian and its mean is zero. The transmitted signal of user- k can be expressed as

$$x_k(n) = \sum_{j=-\infty}^{\infty} d_k(j) \sum_{l=0}^{U-1} c_{k,j}(l) p(n - l - jU), \quad k = 1, \dots, K \quad (2.2)$$

where $d_k(j)$ denotes the j -th BPSK signal of user- k and $c_{k,j}(l) \in \{1, -1\}$ corresponds to the l -th chip signal in $d_k(j)$. Also, U denotes the processing gain and $p(n)$ the chip-rate sampled pulse. Before proceed further, we list assumptions to be used in the sequel:

- a) User-1's code delay is of interest and $A_1 = 1$.
- b) The code delay is an integer multiples of the chip-duration and smaller than U .
- c) Carrier synchronization is established before code acquisition.
- d) No data are modulated for user-1's signal in the period of code acquisition, i.e., $d_1(j) = 1$.
- e) The chip-pulse is considered as a rectangular pulse with unit amplitude.
- f) The code sequence $c_{k,j}$ has a period much higher than the processing gain such that the input to the adaptive filter can be viewed as statistically white.

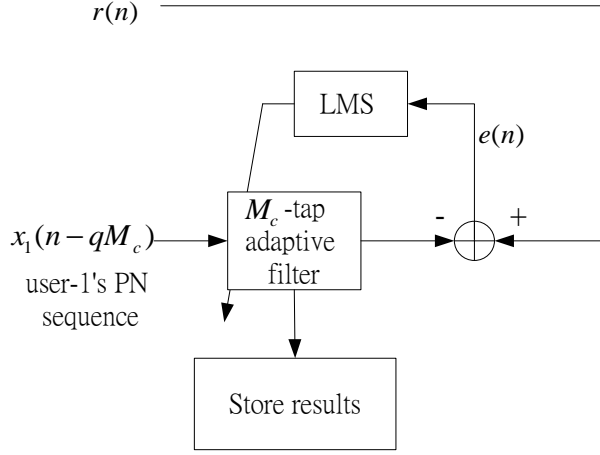


Figure 2.1: Conventional 1R code acquisition system, where $x_1(n - qM_c)$ is user-1's PN sequence at q -th cell with $q = 0, \dots, Q - 1$.

g) Only the additive white Gaussian noise (AWGN) channel is considered and the summation of MAI and white Gaussian noise can be modeled as another white Gaussian noise [32].

The 1R scheme first divides U into $Q = \lceil U/M_c \rceil$ cells, where M_c is the length of the adaptive filter. The adaptive filter then serially searches the code delay in these cells. The least-mean-square (LMS) algorithm is employed to minimize the MSE between the received signal $r(n)$ and the adaptive filter output (see Fig. 2.1). The tap-weight update equations are given by

$$\mathbf{w}^q(n+1) = \mathbf{w}^q(n) + \mu_c e(n) \mathbf{x}^q(n) \quad (2.3)$$

$$e(n) = r(n) - [\mathbf{w}^q(n)]^T \mathbf{x}^q(n), \quad q \in \{0, \dots, Q-1\} \quad (2.4)$$

where μ_c denotes the step size controlling the convergence of the adaptive filter, $\mathbf{w}^q(n) = [w_0^q(n), w_1^q(n), \dots, w_{M_c-1}^q(n)]^T$ the filter tap-weight vector for the q -th cell, and $\mathbf{x}^q(n) = [x_1(n - qM_c), x_1(n - qM_c - 1), \dots, x_1(n - qM_c - M_c + 1)]^T$ the corresponding input vector. Here, q is sequentially increased from zero to $Q - 1$. The tap-weight vector $\mathbf{w}^q(n)$ for a particular q is stored after some iterations, say N_1 chips. Then, an estimation of τ_1 can be derived with the tap-index of the maximum tap-weight (among all cells). Let the $\hat{\Delta}_c$ -th tap ($0 \leq \hat{\Delta}_c < M_c$)

of the adaptive filter in the $\hat{\alpha}_c$ -th cell has the maximum value. Then, we can have the delay estimation $\hat{\tau}_1 = \hat{\alpha}_c M_c + \hat{\Delta}_c$. Combine $\mathbf{w}^q(N_1)$, $q = 0, 1, \dots, Q - 1$ into a big vector \mathbf{w} , i.e., $\mathbf{w} = [[\mathbf{w}^0(N_1)]^T, [\mathbf{w}^1(N_1)]^T, \dots, [\mathbf{w}^{Q-1}(N_1)]^T]^T$. It can be shown that the probability of acquisition error is

$$P_e = 1 - P_b(w_c \geq w_j), \quad c \neq j, \quad \{c, j\} \in \{0, 1, \dots, U - 1\}, \quad (2.5)$$

where w_j denotes the j -th element of \mathbf{w} and w_c the tap-weight corresponding to the true code delay τ_1 (i.e., $c = \tau_1$). To evaluate (2.5), we need to know the stochastic properties of the tap-weights. It has been shown in [33] that these tap-weights at convergence have Gaussian distributions with a mean vector of

$$\mathbf{m}_{(U \times 1)} = \mathbf{w}_o, \quad (2.6)$$

and a covariance matrix of

$$\mathbf{C}_{(U \times U)} \approx \frac{\mu_c}{2} J_{min} \mathbf{I}_{(U \times U)} \quad (2.7)$$

$$\triangleq \sigma_w^2 \mathbf{I}, \quad (2.8)$$

where \mathbf{w}_o is the optimum solution of \mathbf{w} solved with the Wiener equations [34], J_{min} is the corresponding minimum mean-squared error (MMSE), and σ_w^2 is the variance of each tap-weight. Let $\mathbf{R}^q = E\{\mathbf{x}^q(n)[\mathbf{x}^q(n)]^T\}$ and $\mathbf{p}^q = E\{\mathbf{x}^q(n)r(n)\}$. Since the input is white, $\mathbf{R}^q = \mathbf{I}_{(M_c \times M_c)}$. It is well known that $\mathbf{w}_o^q = (\mathbf{R}^q)^{-1} \mathbf{p}^q$. Let $\tau_1 = \alpha_c M_c + \Delta_c$, $0 \leq \Delta_c < M_c$, and p_j^q is the j -th entry of \mathbf{p}^q ($j \in \{0, 1, \dots, M_c - 1\}$). It is simple to show that $p_j^q = 1$ when $q = \alpha_c$ and $j = \Delta_c$, and $p_j^q = 0$ otherwise. This is to say that a unique peak with value one will appear in w_c , and all other weights are zeros. Thus, we can have $J_{min} = E\{r^2(n)\} - 1$. Using (2.6) and (2.8), we can rewrite (2.5) as

$$P_e = 1 - \int_{-\infty}^{\infty} \left[1 - Q\left(\frac{w_c}{\sigma_w}\right) \right]^{U-1} \exp\left(-\frac{(w_c - 1)^2}{2\sigma_w^2}\right) dw_c, \quad (2.9)$$

where $Q(\cdot)$ denotes the Q -function [35]. It is known that an M_c -tap adaptive filter (with the LMS algorithm) requires $2M_c$ multiplications per iteration. Thus, the computational complexity is proportional to the filter size.

§ 2.2 Proposed Adaptive Multirate Code Acquisition

To understand our idea easier, we start our development with a two-rate (2R) system. Then, we will extend it to a three-rate (3R) system.

§ 2.2.1 2R Scheme

Following the assumptions given in Section 2.1, we express (2.1) as

$$\begin{aligned} r(n) &= \sum_{k=1}^K A_k x_k(n - \tau_k) + w(n) \\ &= x_1(n - \tau_1) + v(n), \end{aligned} \quad (2.10)$$

where

$$v(n) = \sum_{k=2}^K A_k x_k(n - \tau_k) + w(n) \quad (2.11)$$

denotes the sum of MAI and white Gaussian noise. Let the variance of $v(n)$ be σ_v^2 . For notational simplification, we will omit the subscripts of $x_1(n)$ and τ_1 in following derivations. Fig. 2.2 shows the architecture of the proposed 2R acquisition system. As we can see, the system contains two units with two different processing rates. We call the unit in Fig. 2.2 (a) as a low-rate unit (LRU). In this unit, the adaptive filter updates its tap-weights with a low rate. For this reason, we refer to the adaptive filter in this unit as a low-rate adaptive filter (LRAF). By contrast, we call the unit in Fig. 2.2 (b) a high-rate unit (HRU). The adaptive filter in this unit updates its tap-weights with a high rate. We refer to the adaptive filter in this unit as a high-rate adaptive filter (HRAF). Note that the high-rate here denotes the chip-rate. There are feedforward and feedback operations in the system. We now describe the fundamental feed-forward operation. First, consider Fig. 2.2 (a). The system passes the received signal $r(n)$ and the locally generated user-1's signal $x(n)$ through lowpass filters (LPFs) to obtain $r_{LPF}(n)$ and $x_{LPF}(n)$ respectively. Then, it downsamples these signals with a factor of D and feeds

the resultant signals to the LRAF. Let $M_p = \lceil U/D \rceil$. Then, the code delay can be rewritten as $\tau = \alpha D + \Delta$ where $\alpha \in \{0, 1, \dots, M_p\}$ and $-D/2 < \Delta \leq D/2$. Note that the ranges of α and Δ are defined different from that in the previous section. The LRAF will adapt to estimate a low-resolution τ having the value in $\{0, D, \dots, M_p D\}$. Similar to the 1R system, we select the tap-index associating with the maximum tap-weight value. Note that $M_p + 1$ is the filter length of the LRAF and $(M_p + 1)D$ must be great or equal to U . Let the index with the maximum tap-weight in the LRAF be $\hat{\alpha}$. The HRU in Fig. 2.2 (b) then delays $x(n)$ with $\hat{\alpha}D$ chips. We call the device to perform the delay function as the delay-tuning filter (DTF). With this operation, the HRAF adapts to refine the code-delay resolution. After convergence, we select the tap-index $\hat{\Delta}$ with the maximum tap-weight. It is easy to see that the index should be in the range of $\pm D/2$. Combing these two tap-weight indices, we can finally obtain a code-delay estimate. In summary, the LRU attempts to acquire τ in a multi-chip level (low resolution), while the HRU in a chip level (high resolution).

We now examine some properties of the 2R feedforward operation. For low complexity consideration, we let the LPF filtered $r(n)$ [in (2.10)] as

$$\begin{aligned} r_{LPF}(n) &= \sum_{j=0}^{D-1} r(n-j) \\ &= \sum_{j=0}^{D-1} x(n-\tau-j) + v_{LPF}(n), \end{aligned} \quad (2.12)$$

where

$$v_{LPF}(n) = \sum_{j=0}^{D-1} v(n-j). \quad (2.13)$$

It is simple to see that this is just an averaging operation with a D -tap filter (apart from a constant). In Fig. 2.2 (a), \mathbf{f}_L indicates a vector consisting of the impulse response of the LPF. As shown, each element of \mathbf{f}_L has the value of one. Substituting $\tau = \alpha D + \Delta$, we can rewrite (2.12) as

$$r_{LPF}(n) = \sum_{j=0}^{D-1} x(n - \alpha D - \Delta - j) + v_{LPF}(n). \quad (2.14)$$

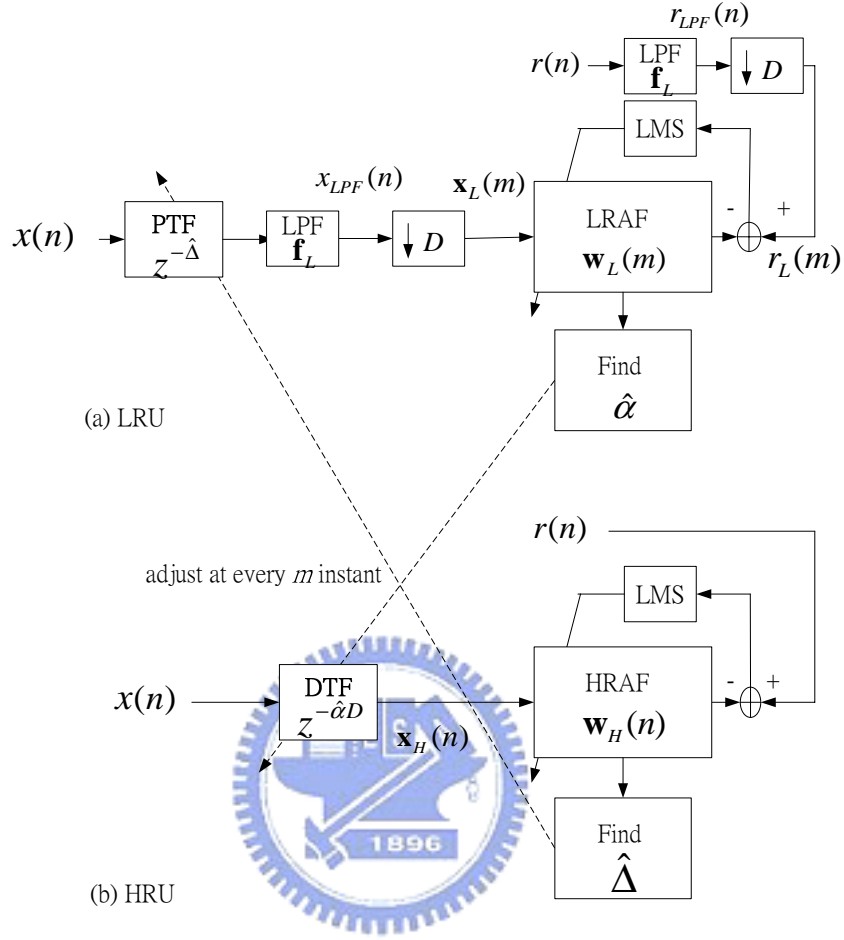


Figure 2.2: Proposed 2R code acquisition system with (a) LRU and (b) HRU. Note that LRU and HRU interact only when $n = mD$. The dash-lines indicate feedforward and feedback operations.

Downsampling (2.14) with a factor of D , we then have

$$\begin{aligned}
 r_L(m) &\triangleq r_{LPF}(n)|_{n=mD} \\
 &= \sum_{j=0}^{D-1} x((m-\alpha)D - \Delta - j) + v_L(m),
 \end{aligned} \tag{2.15}$$

where we let $m = \lfloor n/D \rfloor$ and $v_L(m) = v_{LPF}(mD)$. Similarly, we can average $x(n)$ to obtain

$$x_{LPF}(n) = \sum_{j=0}^{D-1} x(n-j), \quad (2.16)$$

and downsample $x_{LPF}(n)$ to obtain

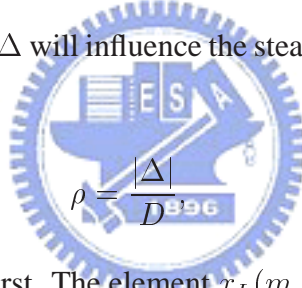
$$\begin{aligned} x_L(m) &\triangleq x_{LPF}(n)|_{n=mD} \\ &= \sum_{j=0}^{D-1} x(mD-j). \end{aligned} \quad (2.17)$$

Let the input vector of the LRAF be $\mathbf{x}_L(m)$. Then, we have

$$\mathbf{x}_L(m) = [x_L(m), x_L(m-1), \dots, x_L(m-M_p)]^T. \quad (2.18)$$

In what follows, we will find out how Δ will influence the steady-state tap-weights of the LRAF.

Let



$$\rho = \frac{|\Delta|}{D} \quad (2.19)$$

and consider the case where $\Delta \geq 0$ first. The element $x_L(m-\epsilon)$, $\epsilon \in \{0, 1, \dots, M_p\}$ in (2.18) can be rewritten as

$$\begin{aligned} x_L(m-\epsilon) &= \sum_{j=0}^{D-1} x((m-\epsilon)D-j) \\ &= \sum_{j=0}^{\Delta-1} x((m-\epsilon)D-j) + \sum_{j=\Delta}^{D-1} x((m-\epsilon)D-j) \\ &= \sum_{j=0}^{\Delta-1} x((m-\epsilon)D-j) + \sum_{j=0}^{D-\Delta-1} x((m-\epsilon)D-\Delta-j) \\ &= \sqrt{D\rho} \left\{ \frac{1}{\sqrt{D\rho}} \sum_{j=0}^{\Delta-1} x((m-\epsilon)D-j) \right\} \\ &\quad + \sqrt{D(1-\rho)} \left\{ \frac{1}{\sqrt{D(1-\rho)}} \sum_{j=0}^{D-\Delta-1} x((m-\epsilon)D-\Delta-j) \right\}. \end{aligned} \quad (2.20)$$

Let

$$\theta(m) = \frac{1}{\sqrt{D\rho}} \sum_{j=0}^{\Delta-1} x(mD - j) \quad (2.21)$$

$$\phi(m) = \frac{1}{\sqrt{D(1-\rho)}} \sum_{j=0}^{D-\Delta-1} x(mD - \Delta - j), \quad (2.22)$$

$\Theta(m) = [\theta(m), \theta(m-1), \dots, \theta(m-M_p)]^T$, and $\Phi(m) = [\phi(m), \phi(m-1), \dots, \phi(m-M_p)]^T$.

Thus, (2.20) can be written as

$$x_L(m - \epsilon) = \sqrt{D\rho}\theta(m - \epsilon) + \sqrt{D(1-\rho)}\phi(m - \epsilon), \quad \epsilon \in \{0, 1, \dots, M_p\} \quad (2.23)$$

and (2.18) as

$$\mathbf{x}_L(m) = \sqrt{D\rho}\Theta(m) + \sqrt{D(1-\rho)}\Phi(m). \quad (2.24)$$

Note that $\theta(m)$, $\phi(m)$ and $v_L(m)$ are zero mean, mutually uncorrelated, and

$$\begin{aligned} E\{\theta(m)\theta(m-\epsilon)\} &= \delta(\epsilon) \\ E\{\phi(m)\phi(m-\epsilon)\} &= \delta(\epsilon) \\ E\{v_L(m)v_L(m-\epsilon)\} &= D\sigma_v^2\delta(\epsilon), \end{aligned} \quad (2.25)$$

where $\delta(\cdot)$ denotes a Kronecker Dirac delta function. Using (2.21)–(2.22), we can also express (2.15) as

$$\begin{aligned} r_L(m) &= \sum_{j=0}^{D-1} x((m-\alpha)D - \Delta - j) + v_L(m) \\ &= \sum_{j=0}^{D-\Delta-1} x((m-\alpha)D - \Delta - j) + \sum_{j=D-\Delta}^{D-1} x((m-\alpha)D - \Delta - j) + v_L(m) \\ &= \sum_{j=0}^{D-\Delta-1} x((m-\alpha)D - \Delta - j) + \sum_{j=0}^{\Delta-1} x((m-\alpha-1)D - j) + v_L(m) \\ &= \sqrt{D(1-\rho)}\phi(m-\alpha) + \sqrt{D\rho}\theta(m-\alpha-1) + v_L(m). \end{aligned} \quad (2.26)$$

Let the tap-weights of the LRAF be $\mathbf{w}_L(m)$ and $\mathbf{w}_L(m) = [w_{L,0}(m), w_{L,1}(m), \dots, w_{L,M_p}(m)]^T$. Also, let the corresponding optimal solution be $\mathbf{w}_{L,o}$. Using the corresponding Wiener equations, we can have

$$\mathbf{w}_{L,o} = \mathbf{R}_L^{-1} \mathbf{p}_L, \quad (2.27)$$

where $\mathbf{p}_L \triangleq E\{\mathbf{x}_L(m)r_L(m)\}$ and $\mathbf{R}_L \triangleq E\{\mathbf{x}_L(m)\mathbf{x}_L^T(m)\}$. It is simple to see that

$$\mathbf{R}_L = D\mathbf{I}. \quad (2.28)$$

Substituting (2.24), (2.26) and (2.28) into (2.27), we obtain

$$w_{L,o,\epsilon} = \begin{cases} 1 - \rho, & \epsilon = \alpha \\ \rho, & \epsilon = \alpha + 1 \\ 0, & \text{otherwise} \end{cases} \quad \epsilon \in \{0, 1, \dots, M_p\} \quad (2.29)$$

where $w_{L,o,\epsilon}$ is the ϵ -th element of $\mathbf{w}_{L,o}$. Let the MSE that the Wiener filter minimizes be $J_L(m)$. Then,

$$\begin{aligned} J_L(m) &= E\{[r_L(m) - \mathbf{w}_L^T(m)\mathbf{x}_L(m)]^2\} \\ &= E\{r_L^2(m)\} - 2\mathbf{w}_L^T(m)\mathbf{p}_L + \mathbf{w}_L^T(m)\mathbf{R}_L\mathbf{w}_L(m), \end{aligned} \quad (2.30)$$

where $E\{r_L^2(m)\} = DE\{r^2(n)\} = D(\sigma_v^2 + 1)$. Replacing $\mathbf{w}_L(m)$ with $\mathbf{w}_{L,o}$, we can obtain the corresponding MMSE, $J_{L,min}$, as

$$\begin{aligned} J_{L,min} &= DE\{r^2(n)\} - D[(1 - \rho)^2 + \rho^2] \\ &= D[\sigma_v^2 + 2\rho(1 - \rho)]. \end{aligned} \quad (2.31)$$

From (2.31), we can see that a nonzero ρ will produce an extra term in the MMSE. We now

proceed to find the MSE yielded by the LMS algorithm. Using (2.24) and (2.29), we derive

$$\begin{aligned} \mathbf{x}_L^T(m)\mathbf{w}_{L,o} &= (1 - \rho)\{\sqrt{D\rho}\theta(m - \alpha) + \sqrt{D(1 - \rho)}\phi(m - \alpha)\} \\ &\quad + \rho\{\sqrt{D\rho}\theta(m - \alpha - 1) + \sqrt{D(1 - \rho)}\phi(m - \alpha - 1)\} \end{aligned} \quad (2.32)$$

$$\begin{aligned} &= \sqrt{D\rho}\theta(m - \alpha) + \sqrt{D(1 - \rho)}\phi(m - \alpha) \\ &\quad - \rho\sqrt{D\rho}\theta(m - \alpha) - \rho\sqrt{D(1 - \rho)}\phi(m - \alpha) \\ &\quad + \rho\sqrt{D\rho}\theta(m - \alpha - 1) + \rho\sqrt{D(1 - \rho)}\phi(m - \alpha - 1). \end{aligned} \quad (2.33)$$

Substituting (2.26) into (2.33), we obtain

$$\begin{aligned} \mathbf{x}_L^T(m)\mathbf{w}_{L,o} &= r_L(m) - v_L(m) + (1 - \rho)\sqrt{D\rho}\{\theta(m - \alpha) - \theta(m - \alpha - 1)\} \\ &\quad - \rho\sqrt{D(1 - \rho)}\{\phi(m - \alpha) - \phi(m - \alpha - 1)\}. \end{aligned} \quad (2.34)$$

Rewriting (2.34), we have

$$\begin{aligned} r_L(m) &= \mathbf{x}_L^T(m)\mathbf{w}_{L,o} + v_L(m) \\ &\quad \underbrace{-(1 - \rho)\sqrt{D\rho}[\theta(m - \alpha) - \theta(m - \alpha - 1)] + \rho\sqrt{D(1 - \rho)}[\phi(m - \alpha) - \phi(m - \alpha - 1)]}_{\triangleq \xi(m)}, \end{aligned} \quad (2.35)$$

where $\xi(m)$ is zero mean and its variance is $\sigma_\xi^2 = 2D\rho(1 - \rho)$. The LMS tap-weight update equation for the LRAF is given by

$$\mathbf{w}_L(m) = \mathbf{w}_L(m - 1) + \mu_L \mathbf{x}_L(m) [r_L(m) - \mathbf{x}_L^T(m)\mathbf{w}_L(m - 1)], \quad (2.36)$$

where μ_L is the step size. Substituting (2.35) into (2.36), we have

$$\mathbf{w}_L(m) = \mathbf{w}_L(m - 1) + \mu_L \mathbf{x}_L(m) [\mathbf{x}_L^T(m)\mathbf{w}_{L,o} + v_L(m) + \xi(m) - \mathbf{x}_L^T(m)\mathbf{w}_L(m - 1)]. \quad (2.37)$$

Subtracting $\mathbf{w}_{L,o}$ on the both sides of (2.37) and letting $\Delta\mathbf{w}_L(m) = \mathbf{w}_L(m) - \mathbf{w}_{L,o}$, we can

rewrite (2.37) as

$$\begin{aligned}
\Delta \mathbf{w}_L(m) &= \Delta \mathbf{w}_L(m-1) - \mu_L \mathbf{x}_L(m) \mathbf{x}_L^T(m) \Delta \mathbf{w}_L(m-1) \\
&\quad + \mu_L \mathbf{x}_L(m) v_L(m) + \mu_L \mathbf{x}_L(m) \xi(m) \\
&= [\mathbf{I} - \mu_L \mathbf{x}_L(m) \mathbf{x}_L^T(m)] \Delta \mathbf{w}_L(m-1) + \mu_L \mathbf{x}_L(m) v_L(m) + \mu_L \mathbf{x}_L(m) \xi(m). \quad (2.38)
\end{aligned}$$

Let $\mathbf{Q}(m) = E\{\Delta \mathbf{w}_L(m) \Delta \mathbf{w}_L^T(m)\}$. Then,

$$\begin{aligned}
\mathbf{Q}(m) &= E\{[\mathbf{I} - \mu_L \mathbf{x}_L(m) \mathbf{x}_L^T(m)] \Delta \mathbf{w}_L(m-1) \\
&\quad \times \Delta \mathbf{w}_L^T(m-1) [\mathbf{I} - \mu_L \mathbf{x}_L(m) \mathbf{x}_L^T(m)]^T\} \\
&\quad + \mu_L^2 E\{v_L^2(m) \mathbf{x}_L(m) \mathbf{x}_L^T(m)\} \\
&\quad + \mu_L^2 E\{\xi^2(m) \mathbf{x}_L(m) \mathbf{x}_L^T(m)\}. \quad (2.39)
\end{aligned}$$

Equation (2.39) can be written as

$$\begin{aligned}
\mathbf{Q}(m) &= (\mathbf{I} - \mu_L \mathbf{R}_L) \mathbf{Q}(m-1) (\mathbf{I} - \mu_L \mathbf{R}_L) \\
&\quad + \mu_L^2 D \sigma_v^2 \mathbf{R}_L + \mu_L^2 \sigma_\xi^2 \mathbf{R}_L. \quad (2.40)
\end{aligned}$$

Note that in (2.40) we implicitly assume that $\xi^2(m)$ and $\mathbf{x}_L(m) \mathbf{x}_L^T(m)$ are uncorrelated. The ϵ -th entry on the main diagonal of $\mathbf{Q}(m)$ is

$$\mathbf{Q}_{\epsilon,\epsilon}(m) = (1 - \mu_L D)^2 \mathbf{Q}_{\epsilon,\epsilon}(m-1) + \mu_L^2 D^2 \sigma_v^2 + \mu_L^2 D \sigma_\xi^2. \quad (2.41)$$

When $m \rightarrow \infty$, we have the asymptotic result as

$$\begin{aligned}
\mathbf{Q}_{\epsilon,\epsilon}(m) &\approx \frac{\mu_L}{2} (D \sigma_v^2 + \sigma_\xi^2) \\
&= \frac{\mu_L D}{2} [\sigma_v^2 + 2\rho(1 - \rho)] \quad \epsilon \in \{0, 1, \dots, M_p\}. \quad (2.42)
\end{aligned}$$

Using (2.31) and (2.42), we can have the MSE for the LMS algorithm in steady-state [34] as

$$\begin{aligned}
J_L(\infty) &= J_{L,\min} + \frac{(M_p + 1) \mu_L D}{2} [\sigma_v^2 + 2\rho(1 - \rho)] \\
&= \left[1 + \frac{(M_p + 1) \mu_L}{2} \right] J_{L,\min}. \quad (2.43)
\end{aligned}$$

Next, let us consider the case where $\Delta < 0$. We define a new set of $\theta(m)$ and $\phi(m)$ as

$$\theta(m) \triangleq \frac{1}{\sqrt{D(1-\rho)}} \sum_{j=0}^{D-|\Delta|-1} x(mD - j) \quad (2.44)$$

$$\phi(m) \triangleq \frac{1}{\sqrt{D\rho}} \sum_{j=0}^{|\Delta|-1} x((m-1)D + |\Delta| - j). \quad (2.45)$$

Then, we can have $x_L(m - \epsilon)$ as

$$x_L(m - \epsilon) = \sum_{j=0}^{D-1} x((m - \epsilon)D - j) \quad (2.46)$$

$$= \sum_{j=0}^{D-|\Delta|-1} x((m - \epsilon)D - j) + \sum_{j=D-|\Delta|}^{D-1} x((m - \epsilon)D - j) \quad (2.47)$$

$$= \sum_{j=0}^{D-|\Delta|-1} x((m - \epsilon)D - j) + \sum_{j=0}^{|\Delta|-1} x((m - \epsilon - 1)D + |\Delta| - j) \quad (2.48)$$

$$= \sqrt{D(1-\rho)}\theta(m - \epsilon) + \sqrt{D\rho}\phi(m - \epsilon), \quad (2.49)$$

and (2.15) as

$$r_L(m) = \sum_{j=0}^{D-1} x((m - \alpha)D + |\Delta| - j) + v_L(m) \quad (2.50)$$

$$= \sum_{j=0}^{|\Delta|-1} x((m - \alpha)D + |\Delta| - j) + \sum_{j=|\Delta|}^{D-1} x((m - \alpha)D + |\Delta| - j) + v_L(m) \quad (2.51)$$

$$= \sum_{j=0}^{|\Delta|-1} x((m - \alpha)D + |\Delta| - j) + \sum_{j=0}^{D-|\Delta|-1} x((m - \alpha)D - j) + v_L(m) \quad (2.52)$$

$$= \sqrt{D\rho}\phi(m - \alpha + 1) + \sqrt{D(1-\rho)}\theta(m - \alpha) + v_L(m). \quad (2.53)$$

Following the similar procedure for the case that $\Delta \geq 0$, we can derive

$$w_{L,o,\epsilon} = \begin{cases} 1 - \rho, & \epsilon = \alpha \\ \rho, & \epsilon = \alpha - 1 \\ 0, & \text{otherwise} \end{cases} \quad \epsilon \in \{0, 1, \dots, M_p\} \quad (2.54)$$

and

$$\begin{aligned} J_L(\infty) &= J_{L,min} + \frac{(M_p + 1)\mu_L D}{2} [\sigma_v^2 + 2\rho(1 - \rho)] \\ &= \left[1 + \frac{(M_p + 1)\mu_L}{2} \right] J_{L,min}. \end{aligned} \quad (2.55)$$

As shown in (2.31), $J_{L,min}$ is a function of ρ . Since $0 \leq \rho \leq 1/2$, when ρ gets larger, $J_{L,min}$ will become larger. This will also make the steady-state MSE in (2.43) larger. Furthermore, from (2.29) and (2.54), we see that a nonzero ρ will produce two nonzero weights and make the value of the peak tap-weight smaller than one. Combining these effects, we can conclude that the larger the ρ , the worse the acquisition performance. The worst case occurs when $\rho = 1/2$ yielding two nonzero equal weights. In what follows, we will develop a system that can null ρ .

Now, let us consider operations in the HRU. As Fig. 2.2 (b) shows, the input to the HRAF is $x(n - \hat{\alpha}D)$. As mentioned, the optimal filter of the LRAF may have two nonzero weights with the same value. Thus, the peak position can be α or $\alpha + 1$. In other words, we need at least $D + 1$ taps for the HRAF. To simplify our analysis, we let $\hat{\alpha} = \alpha$. It is simple to see that the optimal weights of the HRAF will have a unique peak at $\hat{\Delta}$. Since the analysis of HRAF is straightforward, we only provide the results without detailed derivations. Let

$$\begin{aligned} \mathbf{x}_H(n) &= [x(n - \hat{\alpha}D + D/2), \dots, x(n - \hat{\alpha}D), \dots, x(n - \hat{\alpha}D - D/2)]^T \\ &\triangleq [x_{H,-D/2}(n), \dots, x_{H,0}(n), \dots, x_{H,D/2}(n)]^T \end{aligned} \quad (2.56)$$

$$\mathbf{w}_H(n) \triangleq [w_{H,-D/2}(n), \dots, w_{H,0}(n), \dots, w_{H,D/2}(n)]^T, \quad (2.57)$$

where we assume that $D/2$ is an integer (for notational convenience). Notice that $\mathbf{R}_H \triangleq E\{\mathbf{x}_H(n)\mathbf{x}_H^T(n)\} = \mathbf{I}$. We then have the optimum weights listed below:

$$w_{H,o,j} = \begin{cases} 1, & j = \Delta \\ 0, & \text{otherwise} \end{cases}, \quad (2.58)$$

where $w_{H,o,j}$ is the j -th element of $\mathbf{w}_{H,o}$, and $\mathbf{w}_{H,o}$ is the optimal solution of $\mathbf{w}_H(n)$. We then

have the MMSE and steady-state MSE as

$$\begin{aligned}
J_{H,min} &= E\{[r(n) - \mathbf{w}_H^T(n)\mathbf{x}_H(n)]^2\}_{\mathbf{w}_H(n)=\mathbf{w}_{H,o}} \\
&= E\{r^2(n)\} - 2\mathbf{w}_{H,o}^T E\{\mathbf{x}_H(n)r(n)\} + \mathbf{w}_{H,o}^T \mathbf{R}_H \mathbf{w}_{H,o} \\
&= \sigma_v^2
\end{aligned} \tag{2.59}$$

$$J_H(\infty) = \left[1 + \frac{(D+1)\mu_H}{2}\right] J_{H,min}, \tag{2.60}$$

where μ_H is the step size used in the HRAF.

The main problem associated with the 2R scheme described above is that sampling phases for $r_{LPF}(n)$ and $x_{LPF}(n)$ may not be synchronized (i.e., $\Delta \neq 0$). As analyzed, the acquisition performance can be greatly affected when Δ is not equal to zero. Our remedy to this problem is to adjust the sampling phase of $x(n)$ during filter adaptation. This is possible if Δ estimated by the HRAF can be feedback to the LRAF. To realize this thought, we use a device, namely phase-tuning filter (PTF), to tune the input phase with $\hat{\Delta}$ chips (see the feedback operation in Fig. 2.2). The PTF can advance or lag the phase of its input signal. With this structure, the sampling phases for $r_{LPF}(n)$ and $x_{LPF}(n)$ can be synchronized. Note that the LRU and HRU interact only when $n = mD$. Letting $\Delta = 0$ (i.e., $\rho = 0$) in (2.31) and (2.42), we have

$$J_{L,min} = D\sigma_v^2, \tag{2.61}$$

$$\mathbf{Q}_{\epsilon,\epsilon}(m) = \frac{\mu_L}{2} D\sigma_v^2 \quad \epsilon \in \{0, \dots, M_p\}. \tag{2.62}$$

Thus, steady-state MSEs of the LRAF and the HRAF are

$$J_L(\infty) = \left[1 + \frac{(M_p+1)\mu_L}{2}\right] D\sigma_v^2, \tag{2.63}$$

$$J_H(\infty) = \left[1 + \frac{(D+1)\mu_H}{2}\right] \sigma_v^2. \tag{2.64}$$

§ 2.2.2 3R Scheme

In the previous subsection, we have proposed a 2R scheme that is able to null ρ . Since the HRAF operates in a high processing rate, it dominates the overall computational complexity.

This becomes an important issue when the tap-length $D + 1$ is large. We can solve the problem by introducing a unit with another processing rate. We call this unit as a medium-rate unit (MRU). This unit contains a medium-rate adaptive filter (MRAF) sharing the computational loading of the HRAF. As shown in Fig. 2.3 (b), the LPFs \mathbf{f}_M average $r(n)$ and $x(n)$ with a window side of D_M , and the decimators downsample the resultant signals with a factor of D_M . Let the DPTF denote the device cascading the DTF and PTF. Here, the processing rate of the MRU is D/D_M times faster than that of the LRU, but D_M times slower than that of the HRU.

With the additional MRU, we have three resolutions to work with. We can express the code delay as $\tau = \alpha D + \beta D_M + \delta$, where

$$\alpha \in \{0, 1, \dots, M_p\}, \quad (2.65)$$

$$\beta \in \{-D/(2D_M), \dots, 0, \dots, D/(2D_M)\}, \quad (2.66)$$

$$\delta \in \{-D_M/2, \dots, 0, \dots, D_M/2\}. \quad (2.67)$$

Again, for convenience we assume that $D/(2D_M)$ and $D_M/2$ are integers. Then, we use the LRU, MRU and HRU to estimate $\{\alpha, \beta, \delta\}$, respectively. Note that $-(D+D_M)/2 \leq \beta D_M + \delta \leq (D + D_M)/2$, where $D_M \geq 2$. In other words, the MRAF and HRAF can span a delay region greater than $D + 1$. Define the tap-weight vector and the input vector of the MRAF as

$$\mathbf{w}_M(s) \triangleq [w_{M,-D/(2D_M)}(s), \dots, w_{M,0}(s), \dots, w_{M,D/(2D_M)}(s)]^T \quad (2.68)$$

$$\mathbf{x}_M(s) \triangleq [x_{M,-D/(2D_M)}(s), \dots, x_{M,0}(s), \dots, x_{M,D/(2D_M)}(s)]^T, \quad (2.69)$$

where $s = \lfloor n/D_M \rfloor$. The update equation for the MRAF is given by

$$\mathbf{w}_M(s) = \mathbf{w}_M(s-1) + \mu_M \mathbf{x}_M(s) [r_M(s) - \mathbf{x}_M^T(s) \mathbf{w}_M(s-1)], \quad (2.70)$$

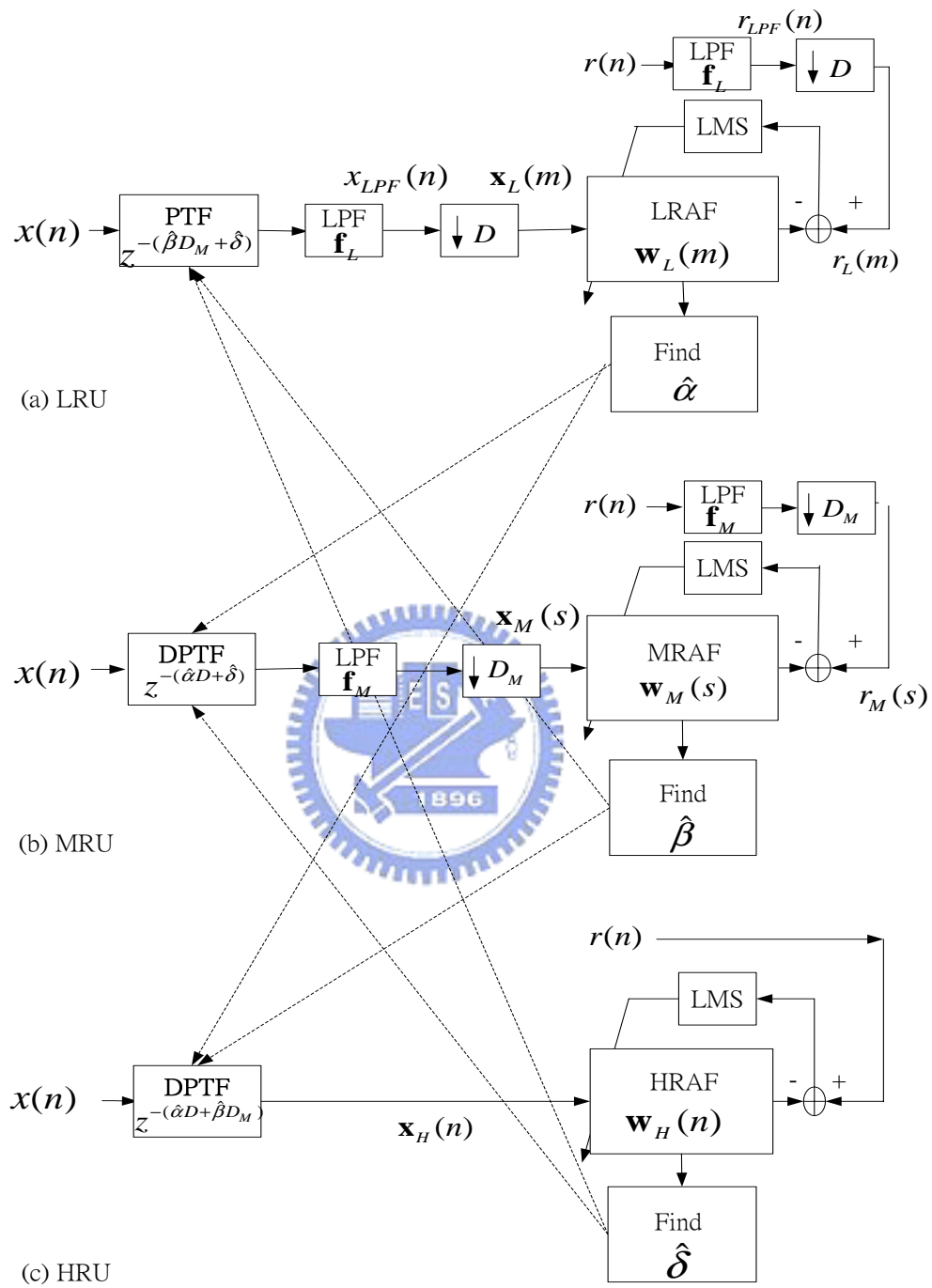


Figure 2.3: Proposed 3R code acquisition system with (a) LRU, (b) MRU, and (c) HRU. Again, all units interact only when $n = mD$ and the dash-lines are for feedforward and feedback operations.

where μ_M is the corresponding step size and

$$r_M(s) = \sum_{j=0}^{D_M-1} r(n-j)|_{n=sD_M}, \quad (2.71)$$

$$x_{M,\epsilon}(s) = \sum_{j=0}^{D_M-1} x(n - \hat{\alpha}D - \hat{\delta} - \epsilon D_M - j)|_{n=sD_M}$$

$$\epsilon \in \{-D/(2D_M), \dots, 0, \dots, D/(2D_M)\}, \quad (2.72)$$

where we have used $\hat{\alpha}D$ and $\hat{\delta}$ obtained from other two units. The weight adaptations for the LRAF and HRAF are similar to (2.36).

We have analyzed the performance of the HRAF and LRAF in a 2R system previously. The performance of the MRU in a 3R system can be done in a similar way. We can treat the MRU as a special LRU, and replace D with D_M for the formulas derived for the LRAF. Since this is straightforward, we omit the detailed results here. Note that in Fig. 2.3 all units update parameters in their PTFs or DTFs simultaneously at $n = mD$. Let the estimates for α , β , and δ at the instant $n = mD$ be $\hat{\alpha}(m)$, $\hat{\beta}(s)$, and $\hat{\delta}(n)$, respectively. When $n = mD$, the PTF in the LRU delays $x(n)$ by $\hat{\beta}(s)D_M + \hat{\delta}(n)$ chips, the DPTF in the MRU delays $x(n)$ by $\hat{\alpha}(m)D + \hat{\delta}(n)$ chips, and the DPTF in the HRU delays $x(n)$ by $\hat{\alpha}(m)D + \hat{\beta}(s)D_M$ chips. We can extend the idea to a four-rate or higher rate system; however, the system architecture will become complex. For typical applications, a 2R or 3R system will be sufficient. As described, all the filters are adjusted using the LMS algorithm. As shown later, the tap-weight of an adaptive filter can be treated as a random variable. Thus, α , β , or δ may be incorrectly estimated during the adaptation, which we call a decision error. Note that the LMS algorithm changes the filter-weight values slowly. For most cases, the estimation error can be corrected shortly. There are only few cases that the error will propagate between adaptive filters and the overall effect may lower the final amplitudes of the peak tap-weights. To alleviate the decision error problem, we can let the LRU operate for a short period of time without feedback at the initial. Simulations show that the error propagation effect only slightly slows the convergence.

§ 2.3 Performance Analysis

To compare the proposed schemes with the 1R system in Section 2.2, we employ some performance measures such as the required computational complexity (per iteration), acquisition error probability, and mean acquisition time.

§ 2.3.1 Computational Complexity

To have a fair comparison, we let $N = N_1 = N_M$, where N_M denotes the iteration time of the multirate system. Also, we let $D = Q$ such that the filter size in the 1R system is approximately equal to that of the LRAF ($M_c \approx M_p + 1$). Since the main operation in filtering is multiplication, we only take this into account. We first calculate the total multiplications required in N iterations and then divide the result by N .

a) 1R scheme

As mentioned in Section 2.2, the M_c -tap adaptive filter will require $2M_c$ multiplications per iteration. Then, the computational complexity of the 1R system, denoted as C_1 , is $2M_c = 2[U/D]$.

b) 2R scheme

For the 2R scheme, we have to take both the LRAF and HRAF into account. Since the HRAF has $D + 1$ taps and operates in the chip-rate, it requires $2(D + 1)$ multiplications per iteration. On the other hand, the LRAF has $M_p + 1$ taps operating in a rate D times slower. Thus, the required multiplications per iteration for a 2R scheme, C_2 , is given by

$$C_2 = \frac{2(M_p + 1)\frac{N}{D} + 2(D + 1)N}{N} \quad (2.73)$$

$$= \frac{2(M_p + 1)}{D} + 2(D + 1). \quad (2.74)$$

c) 3R scheme

Similarly, we take the LRAF, MRAF, and HRAF into account. The required multiplication

per iteration for a 3R system, C_3 , turns out to be

$$C_3 = \frac{2(M_p + 1)\frac{N}{D} + 2\lceil \frac{D+1}{D_M} \rceil \frac{N}{D_M} + 2(D_M + 1)N}{N} \quad (2.75)$$

$$= \frac{2(M_p + 1)}{D} + \frac{2}{D_M} \lceil \frac{D+1}{D_M} \rceil + 2(D_M + 1) \quad (2.76)$$

where $\lceil (D+1)/D_M \rceil$ is the minimum required tap-length for the MRAF.

§ 2.3.2 Probability of Acquisition Error

For the 1R system, we let the adaptive filter run for a period of four time-constants to reach the steady-state (for each cell). The time-constant for a LMS filter with white input and a step size μ_c can be estimated as $\lceil 1/\mu_c \rceil$ (see pp. 348 in [34]). The overall convergence time for the 1R system, denoted as N_1 , is then $4Q\lceil 1/\mu_c \rceil$. Let the step size for the adaptive filter in the 1R system and that in the HRAF be the same (i.e., $\mu_c = \mu_H \triangleq \mu$). For multirate systems described in Section 2.2, we further let $\mu_L = \mu/D$ and $\mu_M = \mu/D_M$. In this way, the variances of these adaptive filter taps are the same [see (2.7) and (2.62)].

a) 2R scheme

An acquisition error may occur due to $\hat{\alpha} \neq \alpha$, $\hat{\delta} \neq \delta$, or both. If we assume that there are no decision errors, the probability of acquisition error for the time instant n , denoted as $P_e(n)$, can be written as

$$P_e(n) = 1 - P_{L,c}(m)P_{H,c}(n) \quad (2.77)$$

$$P_{L,c}(m) = P(w_{L,c}(m) \geq w_{L,j}(m)) \quad c \neq j, \{c, j\} \in \{0, 1, \dots, M_p\} \quad (2.78)$$

$$P_{H,c}(n) = P(w_{H,c}(n) \geq w_{H,j}(n)) \quad c \neq j, \{c, j\} \in \{-D/2, \dots, 0, \dots, D/2\}, \quad (2.79)$$

where $P_{L,c}(m)$ and $P_{H,c}(n)$ denote the correct acquisition probabilities of the LRAF and HRAF, respectively. Also, $w_{L,c}(m)$ and $w_{H,c}(n)$ denote the taps whose tap-indices correspond to the actual code delay.

Using the transient analysis of LMS algorithms in [33], we have the mean weight vector of

the LRAF as

$$E\{\mathbf{w}_L(m)\} = [\mathbf{I} - (\mathbf{I} - \mu_L \mathbf{R}_L)^m] \mathbf{w}_{L,o}, \quad (2.80)$$

and the $(M_p + 1)$ -by- $(M_p + 1)$ covariance matrix as

$$\mathbf{C}_L(m) = \frac{\mu_L D \sigma_v^2}{2} [\mathbf{I} - (\mathbf{I} - 2\mu_L \mathbf{R}_L)^m]. \quad (2.81)$$

Since $\mathbf{R}_L = D\mathbf{I}$, we can let $\mathbf{C}_L(m) = \sigma_{w_L}^2(m)\mathbf{I}$ where $\sigma_{w_L}^2(m)$ is an equivalent variance that can be derived from (2.81). Here, $\mathbf{w}_L(m)$ and $\mathbf{w}_H(n)$ are assumed to be Gaussian distributed. Similarly, we can have the mean weight vector and the covariance matrix of $\mathbf{w}_H(n)$ as

$$E\{\mathbf{w}_H(n)\} = [\mathbf{I} - (\mathbf{I} - \mu_H \mathbf{R}_H)^n] \mathbf{w}_{H,o}, \quad (2.82)$$

and

$$\mathbf{C}_H(n) = \frac{\mu_H \sigma_v^2}{2} [\mathbf{I} - (\mathbf{I} - 2\mu_H \mathbf{R}_H)^n]. \quad (2.83)$$

Since $\mathbf{R}_H = \mathbf{I}$, we can let $\mathbf{C}_H(n) = \sigma_{w_H}^2(n)\mathbf{I}$. Similarly, $\sigma_{w_H}^2(n)$ is an equivalent variance that can be derived from (2.83). From (2.81) and (2.83), we find that tap-weights are independent and identically distributed. As mentioned, both the HRAF and LRAF are run for N chips. For notational simplicity, we let A_L as the peak in $E\{\mathbf{w}_L(\lfloor N/D \rfloor)\}$, $\sigma_L^2 = \sigma_{w_L}^2(\lfloor N/D \rfloor)$, $P_L = P_{L,c}(\lfloor N/D \rfloor)$, A_H as the peak in $E\{\mathbf{w}_H(N)\}$, $\sigma_H^2 = \sigma_{w_H}^2(N)$, $P_H = P_{H,c}(N)$, and $P_e = P_e(N)$. The probabilities in (2.78)–(2.79) at $n = N$ turn out to be

$$P_L = \frac{1}{\sqrt{2\pi\sigma_L^2}} \int_{-\infty}^{\infty} \left[1 - Q\left(\frac{w}{\sigma_L}\right)\right]^{M_p} \exp\left(-\frac{(w - A_L)^2}{2\sigma_L^2}\right) dw, \quad (2.84)$$

$$P_H = \frac{1}{\sqrt{2\pi\sigma_H^2}} \int_{-\infty}^{\infty} \left[1 - Q\left(\frac{w}{\sigma_H}\right)\right]^D \exp\left(-\frac{(w - A_H)^2}{2\sigma_H^2}\right) dw. \quad (2.85)$$

Finally, we obtain

$$P_e = 1 - P_L P_H. \quad (2.86)$$

As mentioned in Section 2.2, incorrect decisions can occur and the error propagation between the HRAF and LRAF will lower the peak amplitudes of final tap-weights. Thus, the results in (2.84)–(2.86) may be too optimistic. However, the exact analysis of the error propagation effect turns out to be very difficult, if not impossible. In what follows, we propose a simple approximation method to overcome the problem. We first assume that the error propagation affects the mean of a tap-weight much more serious than the variance. As a result, we only consider the variation of mean weight vectors. For an adaptation period, a decision error can occur in any instant and the error sequence can have many patterns. For simplicity, we only investigate those affecting performance most. Consider the LRAF. It is simple to see that if there are κ decision errors during the adaptation period (i.e., between $m = 0$ and $m = \lfloor N/D \rfloor$), the error pattern corresponding to the worst performance will be the one when all errors occur between $m = \lfloor N/D \rfloor - \kappa + 1$ and $m = \lfloor N/D \rfloor$. In other words, once a decision error occurs, the error will continue to the end of the adaptation period. This will make the peak weight value of the LRAF decrease from $m = \lfloor N/D \rfloor - \kappa + 1$ monotonically. We then use this pattern to represent all possible error patterns having κ decision errors. From (2.80), we have $A_L = 1 - (1 - \mu_L D)^{\lfloor N/D \rfloor}$. Taking the decision errors into account, we may then rewrite A_L as

$$A_L(\kappa) = [1 - (1 - \mu_L D)^{\lfloor N/D \rfloor - \kappa}] \cdot \exp\left(-\frac{\kappa}{\eta_L}\right), \quad (2.87)$$

where $\eta_L^{-1} = \mu_L \lambda_L$ and $\lambda_L = D$ is the eigenvalue of \mathbf{R}_L [34]. We may treat κ as a random variable with a binomial distribution as

$$p(\kappa) = \binom{\lfloor N/D \rfloor}{\kappa} (1 - P_L)^\kappa P_L^{\lfloor N/D \rfloor - \kappa}, \quad (2.88)$$

where P_L is the correct probability in (2.84). We then use (2.87)–(2.88) to calculate the mean value of $A_L(\kappa)$, denoted as \bar{A}_L . It is given by

$$\bar{A}_L = \sum_{\kappa=0}^{\lfloor N/D \rfloor} A_L(\kappa) p(\kappa). \quad (2.89)$$

Then, the probability of correct acquisition for the LRAF, denoted as \bar{P}_L , can be obtained by substituting \bar{A}_L into (2.84). Similarly, we can use the same procedure to obtain the probability

of correct acquisition for the HRAF, \bar{P}_H . Finally, the probability of acquisition error for a 2R system, denoted as P_E , is obtained by

$$P_E = 1 - \bar{P}_L \bar{P}_H. \quad (2.90)$$

It is worth mentioning that P_L and P_H are the correct acquisition probabilities without decision errors. Thus, these values essentially correspond to two upper bounds of the correct acquisition probabilities. Using these values in the calculation of $p(\kappa)$ [as that in (2.88)] will underestimate the acquisition error probability. On the other hand, we only take the worst decision error patterns into consideration and this will overestimate the acquisition error probability. Thus, (2.90) is a result corresponding a compromise of these two extreme cases.

b) 3R scheme

Using the similar idea, we can have the probability of acquisition error for the decision-error-free case as

$$P_e(n) = 1 - P_{L,c}(m)P_{M,c}(s)P_{H,c}(n) \quad (2.91)$$

$$P_{L,c}(m) = P(w_{L,c}(m) \geq w_{L,j}(m)) \quad c \neq j; \{c, j\} \in \{0, 1, \dots, M_p\} \quad (2.92)$$

$$P_{M,c}(s) = P(w_{M,c}(s) \geq w_{M,j}(s)) \quad c \neq j; \{c, j\} \in \{-D/(2D_M), \dots, 0, \dots, D/(2D_M)\} \quad (2.93)$$

$$P_{H,c}(n) = P(w_{H,c}(n) \geq w_{H,j}(n)) \quad c \neq j; \{c, j\} \in \{-D_M/2, \dots, 0, \dots, D_M/2\}, \quad (2.94)$$

where $P_{L,c}(m)$, $P_{M,c}(s)$, and $P_{H,c}(n)$ are the correct acquisition probabilities of the LRU, MRU, and HRU, respectively; $w_{L,c}(m)$, $w_{M,c}(s)$, $w_{H,c}(n)$ denote the taps whose tap-indices correspond to the actual code delay. Note that $s = \lfloor n/D_M \rfloor$. Let $P_L = P_{L,c}(\lfloor N/D \rfloor)$, $P_M = P_{M,c}(\lfloor N/D_M \rfloor)$, $P_H = P_{H,c}(N)$, and $P_e = P_e(N)$. Then P_L can be calculated as that in (2.84), while P_M and P_H are given by

$$\begin{aligned} P_M &= \frac{1}{\sqrt{2\pi\sigma_M^2}} \int_{-\infty}^{\infty} \left[1 - Q\left(\frac{w}{\sigma_M}\right) \right]^{D/D_M} \exp\left(-\frac{(w - A_M)^2}{2\sigma_M^2}\right) dw \\ P_H &= \frac{1}{\sqrt{2\pi\sigma_H^2}} \int_{-\infty}^{\infty} \left[1 - Q\left(\frac{w}{\sigma_H}\right) \right]^{D_M} \exp\left(-\frac{(w - A_H)^2}{2\sigma_H^2}\right) dw, \end{aligned} \quad (2.95)$$

where A_M and σ_M^2 can be obtained as that described in (2.80)–(2.81). Then, we have $P_e = 1 - P_L P_M P_H$. Again, P_e does not consider the decision error propagation effect. We can follow the same notation definitions and procedures outlined in the previous subsection to obtain $\{\bar{P}_L, \bar{P}_M, \bar{P}_H\}$. Finally, we have the probability of acquisition error for the 3R system as

$$P_E = 1 - \bar{P}_L \bar{P}_M \bar{P}_H. \quad (2.96)$$

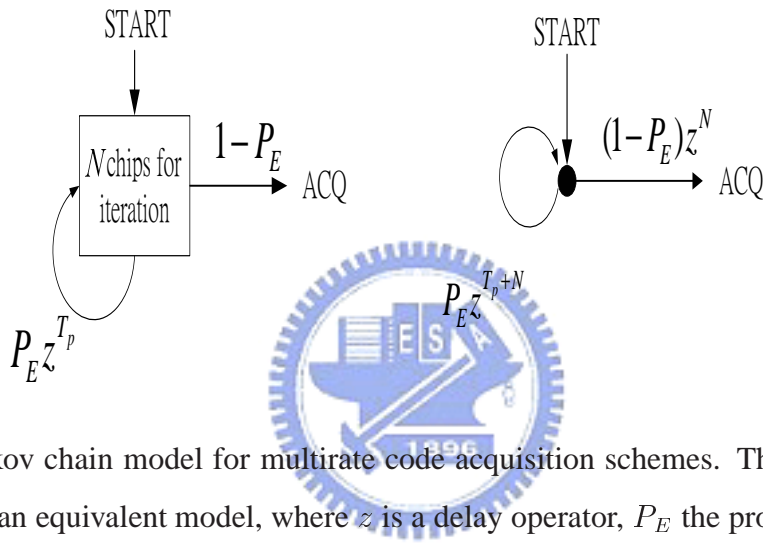


Figure 2.4: Markov chain model for multirate code acquisition schemes. The right hand side figure illustrates an equivalent model, where z is a delay operator, P_E the probability of acquisition error, T_p penalty time, and ACQ the correct acquisition state.

§ 2.3.3 Mean Acquisition Time

Mean acquisition time analysis is generally derived with a Markov chain model [36]. Since our multirate systems is different from the correlator with serial search, the commonly used model [10] cannot be applied here. Fig. 2.4 shows the model derived for our systems. As the figure shows, the system iterates for N chips to obtain $\hat{\tau}$ and the probability of acquisition error is P_E . If the acquisition fails, it will wait for a period of time T_p (chips) before the system re-starts the acquisition. Here, T_p is generally referred to as the penalty time [32]. For our schemes, $\hat{\tau}$ is constructed from $\{\hat{\alpha}, \hat{\Delta}\}$ or $\{\hat{\alpha}, \hat{\beta}, \hat{\delta}\}$ at $n = N$. If $\hat{\tau} \neq \tau$, the receiver will

re-initialize acquisition after a time interval of T_p chips. We can have the transfer function of the Markov chain model in Fig. 2.4 as [27], [36]

$$H_{acq}(z) = \frac{(1 - P_E)z^N}{1 - P_E z^{T_p + N}}, \quad (2.97)$$

where z is a delay operator and P_E is the probability of acquisition error formulated above. The mean acquisition time can then be easily found as

$$T_{acq} \triangleq \frac{d}{dz} H_{acq}(z) \Big|_{z=1} \quad (2.98)$$

$$= N + \frac{(T_p + N)P_E}{(1 - P_E)}. \quad (2.99)$$

Note that the unit of T_{acq} is chip.

§ 2.4 Simulations

In this section, we conduct computer simulations to demonstrate the effectiveness of the proposed algorithms. First, we investigate the computational complexity issue. Using (2.74) and (2.76), we can evaluate the computational complexity requirement per chip versus D for 1R, 2R, and 3R schemes. We list the results in Table 2.1, 2.2, and 2.3 for $D = 4, 8,$ and $16,$ respectively. The numbers inside the parentheses in these tables indicate the values of D_M used for the 3R system. Also, the last two rows of the tables give the complexity ratio defined as C_2/C_1 and $C_3/C_1,$ respectively. From these tables, we can have several observations. Firstly, the larger the processing gain, the higher efficiency the multirate system can achieve. Secondly, the 3R system is always more efficient than the 2R system. Lastly, there exists an optimum D for a given processing gain $U.$ For example, when $U = 1024$ and $D = 8,$ the computational complexity of the 2R system is about 20% of the 1R system. For the same processing gain with $D = 16,$ the complexity of the 3R system is only about 16% of the 1R system. These outcomes state that the multirate system can be much more efficient than the 1R system for large $U.$

Table 2.1: Computational Complexity Comparison for $D = 4$.

| | | | | |
|-----------|---------|---------|---------|----------|
| U | 128 | 256 | 512 | 1024 |
| C_1 | 64 | 128 | 256 | 512 |
| C_2 | 26.50 | 42.50 | 74.50 | 138.50 |
| C_3 | 25.5(2) | 41.5(2) | 73.5(2) | 137.5(2) |
| C_2/C_1 | 0.414 | 0.332 | 0.291 | 0.271 |
| C_3/C_1 | 0.398 | 0.324 | 0.287 | 0.269 |

Table 2.2: Computational Complexity comparison for $D = 8$.

| | | | | |
|-----------|----------|----------|----------|----------|
| U | 128 | 256 | 512 | 1024 |
| C_1 | 32 | 64 | 128 | 256 |
| C_2 | 22.25 | 26.25 | 34.25 | 50.25 |
| C_3 | 14.25(3) | 18.25(3) | 26.25(3) | 42.25(3) |
| C_2/C_1 | 0.695 | 0.410 | 0.268 | 0.196 |
| C_3/C_1 | 0.445 | 0.285 | 0.205 | 0.165 |

Table 2.3: Computational Complexity Comparison for $D = 16$.

| | | | | |
|-----------|-----------|-----------|-----------|-----------|
| U | 128 | 256 | 512 | 1024 |
| C_1 | 16 | 32 | 64 | 128 |
| C_2 | 35.125 | 36.125 | 38.125 | 42.125 |
| C_3 | 13.125(3) | 14.125(3) | 16.125(3) | 20.125(3) |
| C_2/C_1 | 2.195 | 1.129 | 0.596 | 0.329 |
| C_3/C_1 | 0.821 | 0.441 | 0.252 | 0.157 |

We then compare the probabilities of acquisition error for 1R, 2R, and 3R systems. We set signal-to-interference-plus-noise ratio (SINR_c), which is defined $1/\sigma_v^2$, as -13 dB (about 20 users with equal power). Also, $U = 128$, $D = 8$, $D_M = 4$, $\mu = \mu_H = \mu_L D = \mu_M D/2$, and

$N = 4D \lceil 1/\mu \rceil$. The code-delay τ is uniformly and randomly selected from $[0, U)$. We conduct 10^4 independent trials and show the results in Fig. 2.5. Also shown in the figure is the theoretical results derived in Section 2.3. Experimental results in Fig. 2.5 indicate that the performance of multirate systems are slightly better than that of the 1R system. Theoretical predictions for all systems are accurate particularly when the step size is large. For the 1R system, the deviation between experimental and theoretical values is smaller than that in 2R and 3R systems. This is not surprising, since the 1R system does not have the error propagation problem.

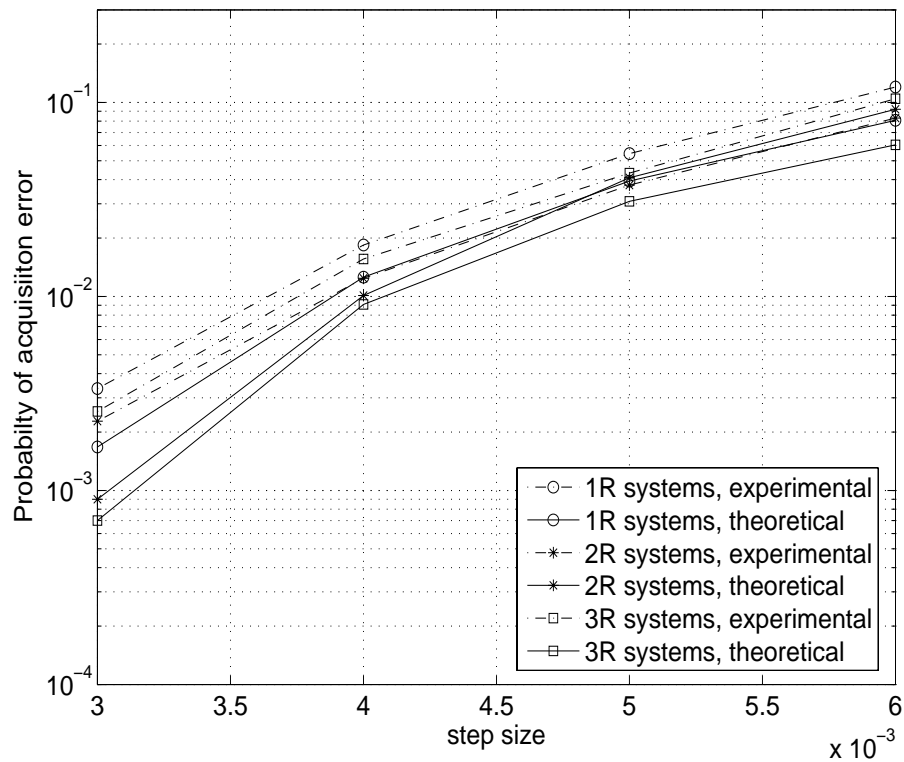


Figure 2.5: Experimental and theoretical P_E [(2.9), (2.90), and (2.96)] versus step size μ ($D = 8$, $D_M = 4$, $U = 128$, and $\text{SINR}_c = -13$ dB).

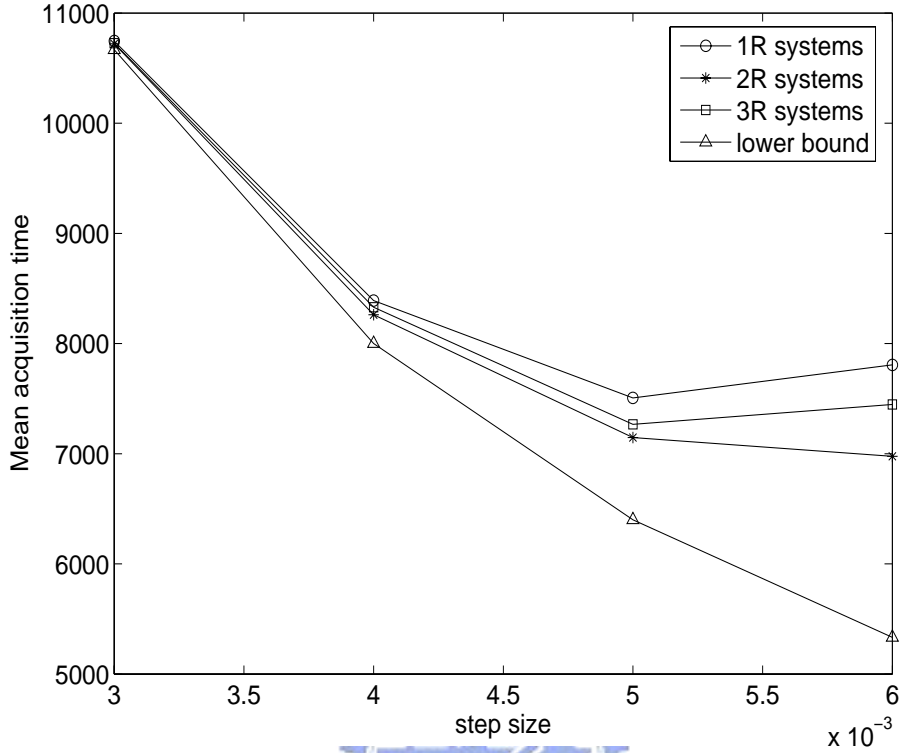


Figure 2.6: Experimental mean acquisition time T_{acq} versus step size μ ($T_p = 1.28 \times 10^4$, $D = 8$, $D_M = 4$, and $\text{SINR}_c = -13$ dB).

As mentioned, an important acquisition performance measure is the mean acquisition time. To derive the mean acquisition time, T_{acq} , we first set $T_p = 1.28 \times 10^4$ chips (100 bits) and substitute the experimental acquisition error probabilities obtained from Fig. 2.5 into (2.99). Fig. 2.6 shows the mean acquisition time for all systems. The lower bound in Fig. 2.6 corresponds to the case that no acquisition errors occur. In this case, $T_{acq} = N = 4D \lceil 1/\mu \rceil$ and this can serve as a performance bound for comparison. As we can see, initially the mean acquisition time decreases when the step size increases. When the step size is larger than $\mu = 5 \times 10^{-3}$, the mean acquisition time starts to increase. For the setting here, the optimal step size is around $\mu = 5 \times 10^{-3}$. In this case, T_{acq} for the 1R system is about 7500 chips, that for the 2R system is about 7150 chips, and that for the 3R system is about 7250 chips. We also examine the prob-

ability of acquisition error for various SINR_c . Fig. 2.7 shows the experimental results. Here, we let $\mu = 5 \times 10^{-3}$, $U = 128$, $D = 8$, $D_M = 4$, and $N = 4D \lceil 1/\mu \rceil$. We find that all systems have similar performance. Also, the higher the SINR_c , the better performance we can have. The 2R system behaves slightly better than the others. Fig. 2.8 shows the corresponding mean acquisition time. In terms of the mean acquisition time, we have the same conclusion that all systems have similar performance.

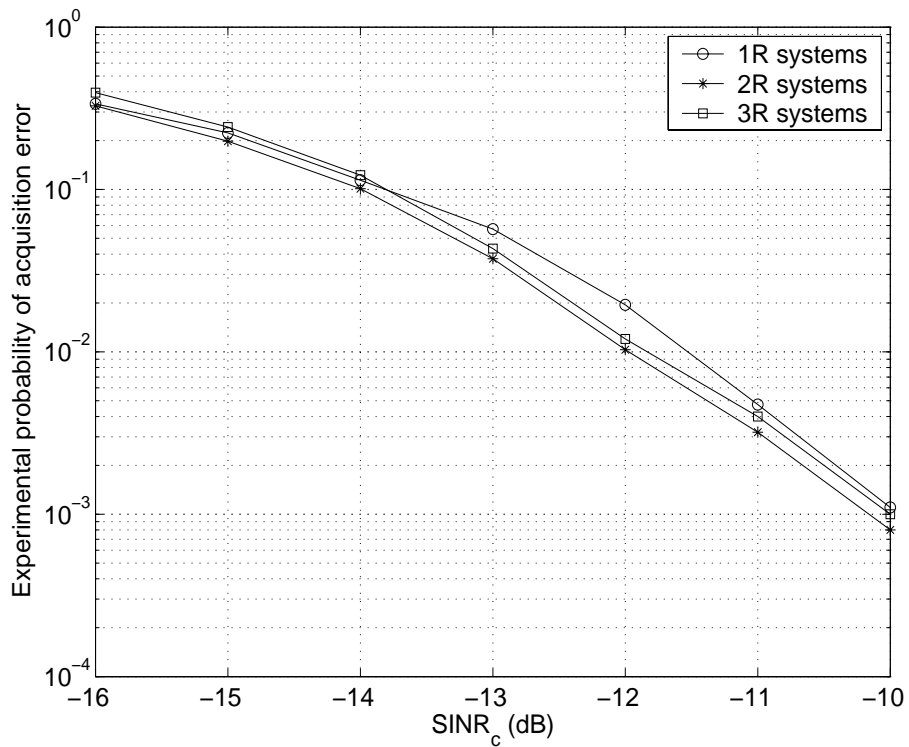


Figure 2.7: Experimental P_E versus SINR_c ($\mu = 5 \times 10^{-3}$).

For all simulations shown above, we have fixed $N = 4D \lceil 1/\mu \rceil$ for the systems. In terms of mean acquisition time, this choice may not be optimal. Fig. 2.9 shows the mean acquisition time for various N (SINR_c is -13 dB). As we can see, there are optimum N 's for multirate systems. For $\mu = 5 \times 10^{-3}$, we find that the mean acquisition time of the 3R system increases quicker than that of the 2R system when N is smaller than the optimum value. This is because

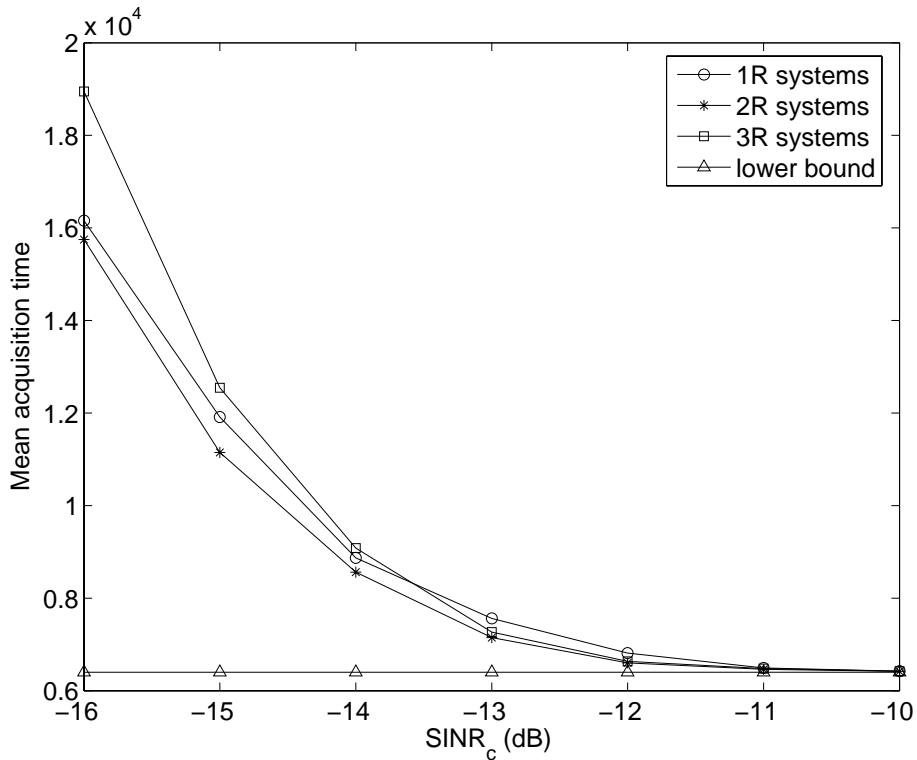


Figure 2.8: Experimental mean acquisition time T_{acq} versus SINR_c ($\mu = 5 \times 10^{-3}$).

the performance of low-rate units depends on N more strongly. When N is larger than the optimum value, the mean acquisition times of both systems approach the lower bounds. We can observe the same behaviors when $\mu = 3 \times 10^{-3}$. From the figure, we also find that the optimal N is about 2 and 2.5 time-constants for $\mu = 3 \times 10^{-3}$ and $\mu = 5 \times 10^{-3}$, respectively. In these cases, $T_{acq} = 6 \times 10^3$ chips (47 bits) for both step sizes.

§ 2.5 Conclusions

The performance of conventional code acquisition in a CDMA system degrades greatly when MAI presents. The adaptive filtering approach proposed recently has been proven to be MAI-resistant. In this chapter, we propose a multirate adaptive code acquisition scheme that can

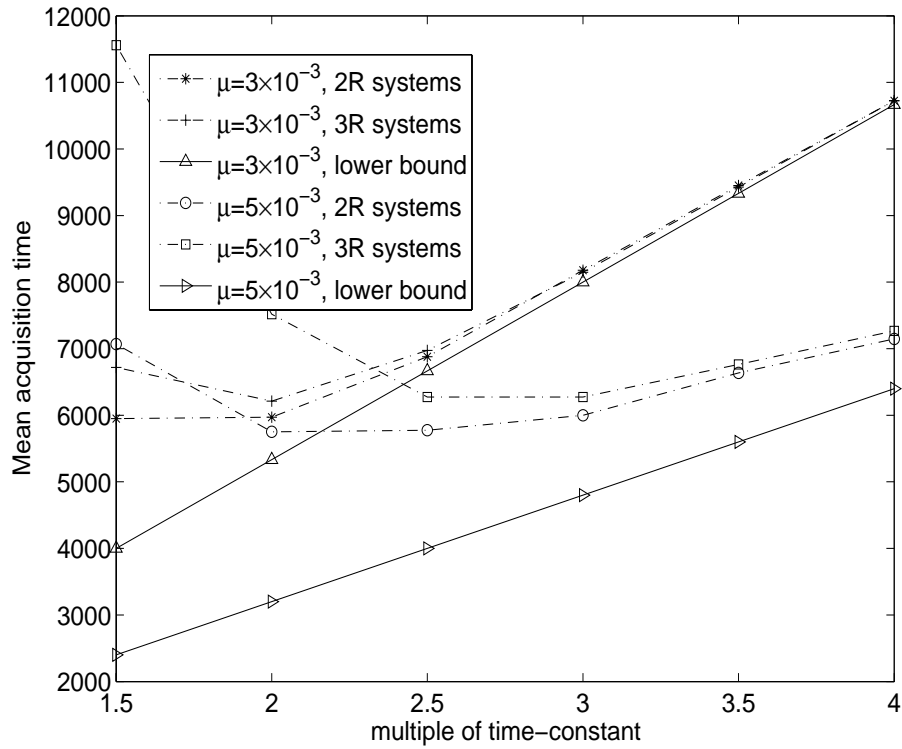


Figure 2.9: Experimental mean acquisition time T_{acq} versus N (expressed as a multiple of time-constant) and step size μ .

significantly reduce the required computational complexity. We have specifically studied the 2R and 3R systems and theoretically analyzed their performance; this includes the filter convergence properties, acquisition error rate, and mean acquisition time. Experimental results show that while the proposed schemes can perform similarly with the conventional adaptive acquisition, the computational complexity is much lower. The proposed scheme is specially suitable for CDMA systems operating in large propagation delay environments. With proper choice of D or D_M , the multirate code acquisition scheme can achieve an efficient compromise between the mean acquisition time and computational complexity. The proposed scheme can also be used in a carrier-phase unsynchronized system. In this circumstance, we have to take the in-phase as well as quadrature components of tap-weights into account. For fractional delay, we

can conduct a sub-chip level acquisition using an oversampled receive signal. This will enhance acquisition performance, but it increases the computational complexity too. In this chapter, we only consider the scenario of the AWGN channel. It is straightforward to extend the use of the proposed scheme to a frequency selective fading channel. In this case, the proposed scheme will acquire the code delay of the strongest channel path. These problems can serve as potential topics for further research.



Chapter 3

Adaptive Antenna Array Code Acquisition

Antenna arrays, being able to perform beamforming, can be used to enhance acquisition performance [38]– [42]. Code acquisition with an array system is conventionally solved with the correlator-based structure. In [38], each array element is equipped with a correlator, and the correlator outputs are used as the input to a beamformer. If an assumed code-phase is correct, the output of the optimum beamformer will exceed a pre-set threshold. Then, acquisition is claimed. Otherwise, the code-phase is changed and optimum beamformer weights are recalculated. A frequently considered MAI scenario is called directional MAI, in which MAI signals arrive at the array in some incident angles. When the interference is present, direct matrix inversion is needed to derive the optimum beamformer weights [38]. In [39], an adaptive beamformer is used to avoid this problem. However, the beamformer has to converge for each trial code-phase. It requires a long adaptation time in an MAI environment and acquisition is then slow. The approach in [42] uses a simple noncoherent correlator performing a two-dimensional search. This method serially searches a cell corresponding to a specified delay and an angular region. Since the search is performed in two-dimension, it often requires longer mean acquisition time if better angular resolution is desired. Besides, the acquisition performance degrades when directional interference exists, as addressed in [42]– [43]. A remedy with an additional algorithm was proposed in [43]. Apart from that, there are approaches treating acquisition as a

channel estimation problem [44]– [49]. These methods provide good performance and usually require matrix computations that are not desirable in real-world implementation. In [50]– [51], a correlator-bank exploiting multipath signals was used for acquisition. However, its structure is quite complicated.

In this chapter, we propose a novel scheme for code acquisition with antenna array. The proposed algorithm belongs to the category of the adaptive filtering approach. It can be applied in either periodic or aperiodic code systems. The proposed scheme contains two adaptive filters, a spatial and a temporal filter. A MSE criterion is proposed such that both filters can be simultaneously adjusted by a stochastic gradient descent algorithm, called the constrained LMS algorithm. The spatial filter acts as a beamformer to suppress interference while the temporal filter acts as a channel estimator identifying the code-delay. The proposed scheme can form a beam-pattern with multiple main beams collecting the desired user's multipath signals from different directions. We also analyze the signal-to-interference-plus-noise ratio (SINR) at the beamformer output, probability of correct acquisition, and mean acquisition time, and derive closed-form expressions for the AWGN channel. The proposed scheme can deal with fractional code-delay, which is not considered in [38]– [42]. Also, the temporal filter can estimate channel responses of the desired user.

This chapter is organized as follows. Section 3.1 describes the adaptive array acquisition approach in [39]. Section 3.2 develops the proposed schemes for various channel scenarios. Section 3.3 discusses issues of adaptive processing, and Section 3.4 carries out performance analysis. Section 3.5 presents simulation results demonstrating the effectiveness of the proposed scheme. Finally, Section 3.6 gives conclusions. Throughout this chapter, we use \mathbf{I} to denote an identity matrix. Note that the dimension of \mathbf{I} is not explicitly shown; it will be defined whenever necessary.

§ 3.1 Correlator-Based Adaptive Array Code Acquisition

In this section, we briefly review the adaptive array approach in [39] which is a single-dwell serial search method. The block diagram of this approach is shown in Fig. 3.1. As seen, it has an antenna array with M sensors, and uses an individual correlator (or accumulator) for each array element. It is assumed that the array is linear and the M sensors are uniformly placed. Also, the element spacing is half a wave-length. The chip-rate sampled received signal vector in baseband is given by

$$\mathbf{r}(n) = \mathbf{a} \exp(-i\theta)x(n - \tau) + \boldsymbol{\rho}(n), \quad (3.1)$$

where $x(n) \in \{+1, -1\}$ is the PN code sequence of the desired user (i.e., no data are modulated for the desired user during acquisition), θ the carrier-phase offset, τ the corresponding code-delay assumed to be an integer from zero to $U - 1$ where U is the processing gain, and $\boldsymbol{\rho}(n)$ a zero-mean, complex, and white Gaussian noise vector associated with a covariance matrix of $\sigma_\rho^2 \mathbf{I}$. Note that $\boldsymbol{\rho}(n)$ consists of MAI and noise. When the number of interfering users and the number of resolvable multipaths are large, the Gaussian assumption is generally held. The steering vector \mathbf{a} in (3.1) is given by $\mathbf{a} = [1, \exp(-i\pi \sin \phi), \dots, \exp(-i\pi(M - 1) \sin \phi)]^T$, where ϕ denotes the direction-of-arrival (DoA) of the desired user. For a trial code-phase $\hat{\tau}$, the output of the m -th correlator can be obtained as

$$y_m(n) = \sum_{j=0}^n r_m(j)x(j - \hat{\tau}), \quad m = 0, \dots, M - 1, \quad n = 0, \dots, N_c - 1, \quad (3.2)$$

where $r_m(n)$ is the m -th element in $\mathbf{r}(n)$ and N_c is the processing period for each code-phase, selected as $2U$ in [39]. Let $\mathbf{y}(n) \triangleq [y_0(n), \dots, y_{M-1}(n)]^T$ and

$$\bar{\mathbf{y}}(n) = \sqrt{M} \frac{\mathbf{y}(n)}{\|\mathbf{y}(n)\|}. \quad (3.3)$$

Then (3.3) is used as the input to an M -tap adaptive filter, $\mathbf{w}_c(\cdot)$. Consider a specific processing period and let N_s be the starting instant for filter adaptation ($N_s \leq n < N_c - 1$). Define

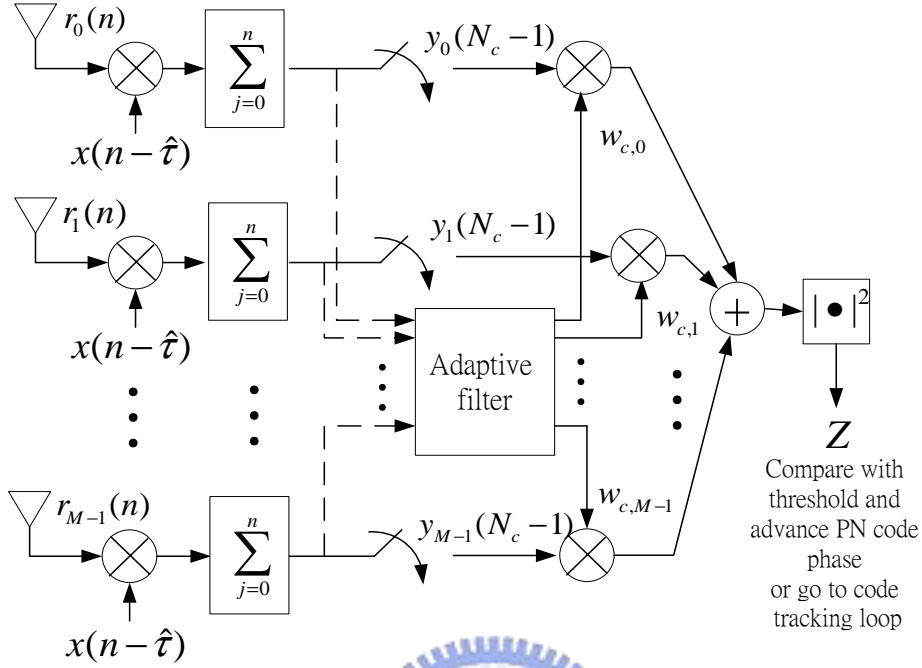


Figure 3.1: Correlator-based adaptive array code acquisition system. Note that $\mathbf{w}_c(N_c - 1) \triangleq [w_{c,0}, \dots, w_{c,M-1}]^T$.

a cost function of $E\{|e_c(n)|^2\}$, where $e_c(n) = M - \mathbf{w}_c^H(n)\bar{\mathbf{y}}(n)$. Using the method of stochastic gradient descent, we can then have the update equation for $\mathbf{w}_c(\cdot)$ as

$$\mathbf{w}_c(n + 1) = \mathbf{w}_c(n) + \mu_c e_c^*(n) \bar{\mathbf{y}}(n) \quad N_s \leq n < N_c - 1, \quad (3.4)$$

where μ_c is the step size controlling the convergence rate. The filter-weight vector $\mathbf{w}_c(N_c - 1)$ is then used for constructing the testing statistic $Z = |\mathbf{w}_c^H(N_c - 1)\mathbf{y}(N_c - 1)|^2$ (see Fig. 3.1). If Z exceeds a pre-set threshold, the system will claim the acquisition and enter the code-tracking phase. Otherwise, the system will advance the trial code-phase and repeat the process all over again. As indicated in [34], the step size is bounded in the range of 0 and $2/\text{tr}\{\mathbf{R}_{\bar{\mathbf{y}}}\}$, where $\mathbf{R}_{\bar{\mathbf{y}}}$ is the correlation matrix of $\bar{\mathbf{y}}(n)$ and $\text{tr}\{\cdot\}$ is the trace operation. In [39], $1/M$ was chosen as a compromise between the convergence rate and stability.

Let the trial code-phase be $\hat{\tau}$. If $\hat{\tau} = \tau$, the decision variable Z will have a noncentral

chi-square distribution. By contrast, if $\hat{\tau} \neq \tau$, the decision variable Z will have a central chi-square distribution. Let κ_a be the threshold for the acquisition claim. Then, we can have the probability of false alarm P_f and the probability of correct detection P_c . Since the false alarm is more harmful to the mean acquisition time, P_f is usually fixed to some level (e.g., $P_f = 0.01$) and the threshold can be calculated accordingly. Finally, the mean acquisition time, denoted as T_{acq} , can be determined as [39]

$$T_{acq} = \left(\frac{(2 - P_c)(1 + K_p P_f)(U - 1)}{2P_c} + \frac{1}{P_c} \right) N_c, \quad (3.5)$$

where K_p is the penalty factor and the unit for (3.5) is chip. Note that if there is no directional interference and the signal-to-noise ratio (SNR) is high, $\mathbf{w}_c(\cdot)$ will converge to the steering vector of the desired user. If the adaptive filter is initialized with $\mathbf{w}_c(N_s) = [1, \dots, 1]^T$, the filter will converge rapidly since each element in the optimum filter-weight is just a phase-rotated version of that in $\mathbf{w}_c(N_s)$. However, if directional interference exists, the relationship cannot be held and the convergence of the adaptive filter becomes slow. As a result, the mean acquisition time becomes large (shown in Section 3.5), especially at strong interference environments.

§ 3.2 Proposed Adaptive Array Code Acquisition

§ 3.2.1 Signal Model and Algorithm Development

Assume that there are K users in a cell and each user is given an aperiodic PN sequence. Aperiodic code means its period is much longer than a symbol period [44], [52]– [53]. The continuously transmitted signal of the k -th user in baseband can be expressed as

$$x_k(t) = \sum_{j=-\infty}^{\infty} d_k(j) \sum_{l=0}^{U-1} c_{k,j}(l) p(t - lT_c - jUT_c) \quad k = 1, \dots, K, \quad (3.6)$$

where $d_k(j)$ is the j -th BPSK symbol of the k -th user, $c_{k,j}(l)$ the l -th chip of the spreading code (assumed to be random) for $d_k(j)$, and $p(t)$ a unit-amplitude rectangular pulse with a chip-duration T_c . Also, let the channel associated with the k -th user has L_k paths, and the

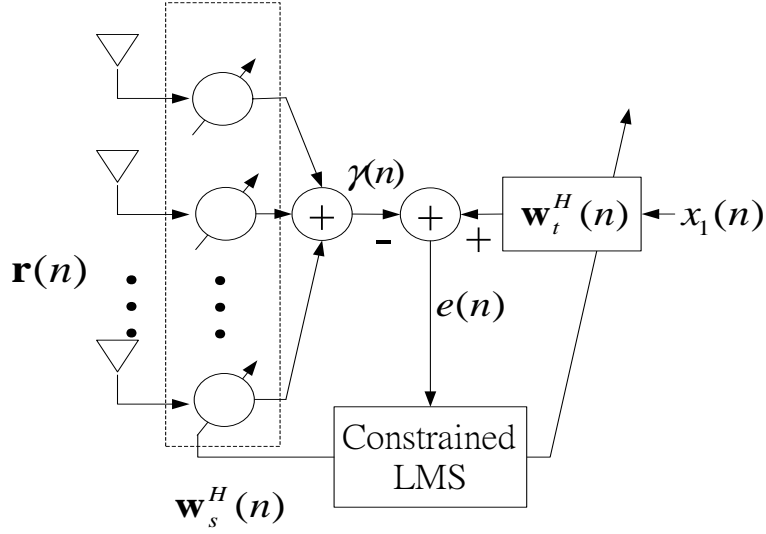


Figure 3.2: Proposed adaptive array code acquisition system.

DoA for each path may be different. Then, the chip-rate sampled received signal vector can be represented as

$$\mathbf{r}(n) = \sum_{k=1}^K \sum_{l=1}^{L_k} \mathbf{a}_{k,l} \alpha_{k,l} \beta_{k,l}(n) \exp(-i\theta_{k,l}) + \boldsymbol{\eta}(n), \quad (3.7)$$

where

$$\beta_{k,l}(n) \triangleq \int_{nT_c}^{(n+1)T_c} x_k(t - \tau_{k,l}T_c) dt, \quad (3.8)$$

$\boldsymbol{\eta}(n)$ is an $M \times 1$ complex Gaussian vector with zero-mean and a covariance matrix of $\sigma_\eta^2 \mathbf{I}$, and $\tau_{k,l}$, $\mathbf{a}_{k,l}$, $\alpha_{k,l}$, $\theta_{k,l}$ denotes the code-delay, steering vector, complex channel gain, and carrier-phase offset, associated with the l -th channel path of the k -th user, respectively. Note that $\theta_{k,l}$ is uniformly distributed over $[-\pi, \pi)$ and $\mathbf{a}_{k,l}$ is given by

$$\mathbf{a}_{k,l} = [1, \exp(-i\pi \sin \phi_{k,l}), \dots, \exp(-i\pi(M-1) \sin \phi_{k,l})]^T, \quad (3.9)$$

where $\phi_{k,l}$ is the DoA for the l -th channel path of the k -th user.

Fig. 3.2 illustrates the structure of the proposed scheme. Without loss of generality, the first user is seen as the desired user. As seen, there are two adaptive filters, a spatial and a

cascaded temporal filter. The spatial filter \mathbf{w}_s combines the array outputs into a single output. The temporal filter \mathbf{w}_t uses $x_1(n)$ as its input signal and the spatial filter output as its reference signal. Note here that $x_1(n)$ is the same as the PN code sequence of the desired user, $c_{1,j}(\cdot)$, since $d_1(j) = 1$ during the acquisition period. In what follows, we will use $x_1(n)$ to denote the PN code sequence of the desired user as well. The spatial filter acts like a beamformer to reject interference, while the temporal filter acts like a channel estimator to estimate the beamformed temporal channel of the desired user. The code-delay can then be estimated from the peak position of \mathbf{w}_t . The difference between these two filter outputs forms the error signal from which we can perform filter adaptation. We propose a cost function as

$$\bar{J} = E\{|\mathbf{w}_t^H \mathbf{x}(n) - \mathbf{w}_s^H \mathbf{r}(n)|^2\}, \quad (3.10)$$

where

$$\mathbf{w}_s \triangleq [w_{s,0}, \dots, w_{s,M-1}]^T \quad (3.11)$$

$$\mathbf{w}_t \triangleq [w_{t,0}, \dots, w_{t,U-1}]^T \quad (3.12)$$

$$\mathbf{x}(n) \triangleq [x_1(n), x_1(n-1), \dots, x_1(n-U+1)]^T. \quad (3.13)$$

Note that the function of the beamformer is to suppress interference and in the ideal case, its output, $\mathbf{w}_s^H \mathbf{r}(n)$, will consist of the beam-formed signal of the desired user and noise. On the other hand, the function of the channel estimator is to identify the beam-formed channel and in the ideal case, its output, $\mathbf{w}_t^H \mathbf{x}(n)$, can form the same beam-formed signal. Thus, minimization of (3.10) will let \mathbf{w}_s and \mathbf{w}_t have the solutions we desire.

Notice that the filter size of \mathbf{w}_t should be larger or equal to the delay uncertainty assumed to be U here. From (3.10), it is simple to find that a minimum \bar{J} (which is zero) occurs when $\mathbf{w}_t = \mathbf{0}$ and $\mathbf{w}_s = \mathbf{0}$, and this is an undesired trivial solution. To avoid that, we pose a unit-norm constraint on the solution. That is

$$\|\mathbf{w}_t\|^2 \triangleq \mathbf{w}_t^H \mathbf{w}_t = 1. \quad (3.14)$$

As a result, (3.10) becomes a constrained optimization problem. We use the Lagrange multiplier method [34] to transform the constrained optimization problem into an unconstrained one. From (3.10) and (3.14), we have an equivalent cost function as

$$\begin{aligned} J &= E\{|\mathbf{w}_t^H \mathbf{x}(n) - \mathbf{w}_s^H \mathbf{r}(n)|^2\} + \xi[1 - \mathbf{w}_t^H \mathbf{w}_t] \\ &= \mathbf{w}_t^H \mathbf{R}_x \mathbf{w}_t + \mathbf{w}_t^H \mathbf{K} \mathbf{w}_s + \mathbf{w}_s^H \mathbf{K}^H \mathbf{w}_t + \mathbf{w}_s^H \mathbf{R}_r \mathbf{w}_s + \xi[1 - \mathbf{w}_t^H \mathbf{w}_t], \end{aligned} \quad (3.15)$$

where

$$\mathbf{K}_{(U \times M)} \triangleq -E\{\mathbf{x}(n)\mathbf{r}^H(n)\}, \quad (3.16)$$

$\mathbf{R}_{r(M \times M)} \triangleq E\{\mathbf{r}(n)\mathbf{r}^H(n)\}$, $\mathbf{R}_{x(U \times U)} \triangleq E\{\mathbf{x}(n)\mathbf{x}^H(n)\}$, and ξ denotes the Lagrange multiplier. Differentiating (3.15) with respect to \mathbf{w}_s^* and \mathbf{w}_t^* and setting the results to be zero-vectors, we can obtain

$$\frac{\partial J}{\partial \mathbf{w}_s^*} = \mathbf{K}^H \mathbf{w}_t + \mathbf{R}_r \mathbf{w}_s = \mathbf{0} \quad (3.17)$$

$$\frac{\partial J}{\partial \mathbf{w}_t^*} = \mathbf{R}_x \mathbf{w}_t + \mathbf{K} \mathbf{w}_s - \xi \mathbf{w}_t = \mathbf{0}. \quad (3.18)$$

Since \mathbf{R}_r is a full-rank matrix, its matrix inversion exists. From (3.17), we have

$$\mathbf{w}_s = -\mathbf{R}_r^{-1} \mathbf{K}^H \mathbf{w}_t. \quad (3.19)$$

Substituting (3.19) into (3.18), we have

$$[\mathbf{R}_x - \mathbf{K} \mathbf{R}_r^{-1} \mathbf{K}^H] \mathbf{w}_t - \xi \mathbf{w}_t = \mathbf{0}. \quad (3.20)$$

It is simple to see that the solution of ξ is an eigenvalue of $\mathbf{R}_x - \mathbf{K} \mathbf{R}_r^{-1} \mathbf{K}^H$, while \mathbf{w}_t is the corresponding eigenvector. Note that an eigenvector satisfies the constraint in (3.14) automatically.

Once \mathbf{w}_t is derived, \mathbf{w}_s can be found using (3.19). Multiplying (3.20) with \mathbf{w}_t^H , we obtain

$$\xi = \mathbf{w}_t^H [\mathbf{R}_x - \mathbf{K} \mathbf{R}_r^{-1} \mathbf{K}^H] \mathbf{w}_t. \quad (3.21)$$

Substituting (3.19) into (3.15) and using (3.21), we have

$$\begin{aligned}
J &= \mathbf{w}_t^H \mathbf{R}_x \mathbf{w}_t - \mathbf{w}_t^H [\mathbf{K} \mathbf{R}_r^{-1} \mathbf{K}^H] \mathbf{w}_t - \mathbf{w}_t^H [\mathbf{K} \mathbf{R}_r^{-1} \mathbf{K}^H] \mathbf{w}_t + \mathbf{w}_t^H [\mathbf{K} \mathbf{R}_r^{-1} \mathbf{K}^H] \mathbf{w}_t \\
&= \mathbf{w}_t^H [\mathbf{R}_x - \mathbf{K} \mathbf{R}_r^{-1} \mathbf{K}^H] \mathbf{w}_t \\
&= \xi.
\end{aligned} \tag{3.22}$$

Let the solutions to (3.17)–(3.18), which are optimum weights, be denoted as $\mathbf{w}_{s,o}$ and $\mathbf{w}_{t,o}$ and the corresponding minimum value of (3.15) be J_{min} . From (3.22), we can conclude that J_{min} is equal to the minimum eigenvalue of $\mathbf{R}_x - \mathbf{K} \mathbf{R}_r^{-1} \mathbf{K}^H$, and $\mathbf{w}_{t,o}$ is the corresponding eigenvector. Substituting $\mathbf{w}_{t,o}$ into (3.19), we can then obtain $\mathbf{w}_{s,o}$. In the sequel, we will apply this result to find $\mathbf{w}_{s,o}$ and $\mathbf{w}_{t,o}$ in various channel scenarios.

§ 3.2.2 Code Acquisition with AWGN Channel

In this subsection, we consider the AWGN channel scenario. In other words, $L_k = 1$ for $k = 1, \dots, K$ in (3.7). For convenience, we drop the subscript l in (3.7). The received signal can be written as

$$\begin{aligned}
\mathbf{r}(n) &= \sum_{k=1}^K \mathbf{a}_k \alpha_k \beta_k(n) \exp(-i\theta_k) + \boldsymbol{\eta}(n) \\
&= \mathbf{a} \cdot 1 \cdot \beta(n) \exp(-i\theta) + \sum_{k=2}^K \mathbf{a}_k \alpha_k \beta_k(n) \exp(-i\theta_k) + \boldsymbol{\eta}(n),
\end{aligned} \tag{3.23}$$

where we let $\mathbf{a}_1 = \mathbf{a}$, $\alpha_1 = 1$, $\beta_1(n) = \beta(n)$, $\theta_1 = \theta$, and $\tau_1 = \tau$.

Here, we let the code-delays of all users have fractional parts. For the desired user, we let

$$\tau = p + \delta, \tag{3.24}$$

where p is an integer, $p \in [0, U)$, and δ is a real number, $\delta \in [0, 1)$. Also, let $x_1(n) = x(n)$.

From (3.8), we can write the received signal of the desired user as [12], [18], [45]

$$\beta(n) = (1 - \delta) \cdot x(n - p) + \delta \cdot x(n - p - 1). \tag{3.25}$$

Note that (3.25) is a weighted sum of $x(n)$ with two successive code-delays p and $p + 1$. This is because $\delta \neq 0$ and a complete received chip in the chip-matched filter crosses two successive chip-intervals. Using (3.25), we can rewrite (3.23) as

$$\begin{aligned} \mathbf{r}(n) = & \mathbf{a} \exp(-i\theta) \left((1 - \delta) \cdot x(n - p) + \delta \cdot x(n - p - 1) \right) \\ & + \sum_{k=2}^K \mathbf{a}_k \alpha_k \beta_k(n) \exp(-i\theta_k) + \boldsymbol{\eta}(n). \end{aligned} \quad (3.26)$$

Substituting (3.26) into (3.16), we find

$$\mathbf{K}_{(U \times M)} = [\mathbf{0}, \dots, \mathbf{0}, -(1 - \delta)\mathbf{a} \exp(-i\theta), -\delta\mathbf{a} \exp(-i\theta), \mathbf{0}, \dots, \mathbf{0}]^H, \quad (3.27)$$

where $-(1 - \delta)\mathbf{a}^H \exp(i\theta)$ and $-\delta\mathbf{a}^H \exp(i\theta)$ are in the $(p + 1)$ - and $(p + 2)$ -th row of \mathbf{K} , respectively. We then apply (3.27) to (3.20) and then obtain

$$\mathbf{I} - \mathbf{K}\mathbf{R}_r^{-1}\mathbf{K}^H = \begin{bmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ \mathbf{e}_1 & \mathbf{e}_2 & \dots & \mathbf{e}_p & 1 - (1 - \delta)^2 \bar{\rho} & -\delta(1 - \delta) \bar{\rho} & \mathbf{e}_{p+3} & \dots & \mathbf{e}_U \\ & & & & -\delta(1 - \delta) \bar{\rho} & 1 - \delta^2 \bar{\rho} & & & \\ & & & & 0 & 0 \\ & & & & \vdots & \vdots \\ & & & & 0 & 0 \end{bmatrix}, \quad (3.28)$$

where we have assumed that $\mathbf{R}_x = \mathbf{I}$ (long-code assumption) and defined

$$\bar{\rho} \triangleq \mathbf{a}^H \mathbf{R}_r^{-1} \mathbf{a} \quad (3.29)$$

$$\mathbf{e}_d \triangleq [\underbrace{0, \dots, 0}_{d-1}, 1, 0, \dots, 0]^T, \quad d = 1, \dots, U. \quad (3.30)$$

Now, we can see that there are two non-unity columns in (3.28) and they are located in the $(p + 1)$ - and $(p + 2)$ -th columns. To derive the eigenvalue ξ , we let

$$\det(\mathbf{I} - \mathbf{K}\mathbf{R}_r^{-1}\mathbf{K}^H - \xi\mathbf{I}) = 0, \quad (3.31)$$

where $w_{t,o,j}$ denotes the j -th element of $\mathbf{w}_{t,o}$ and $j = 0, \dots, U - 1$. Thus, we have

$$\mathbf{w}_{t,o} = \frac{\exp(i\psi)}{\sqrt{(1-\delta)^2 + \delta^2}} [0, \dots, 0, \underbrace{1 - \delta, \delta, 0, \dots, 0}_p]^T, \quad (3.39)$$

where ψ is an arbitrary angle. With the peak position in (3.39), it is simple to see that the code-delay is correctly acquired. As we can see, $\mathbf{w}_{t,o}$ does not have a unique solution. The non-uniqueness of the optimum solution stems from the fact that (3.14) is only an amplitude constraint. Even though the solution is non-unique, it does no harm to our solution since only the amplitude is used in peak finding. Now, let us solve $\mathbf{w}_{s,o}$. Using (3.39) and (3.27) in (3.19), we can derive

$$\mathbf{w}_{s,o} = \mathbf{R}_r^{-1} \mathbf{a} \cdot \exp(i[\psi - \theta]) \sqrt{(1-\delta)^2 + \delta^2}. \quad (3.40)$$

As we can see from (3.40), $\mathbf{w}_{s,o}$ corresponds to the conventional MMSE beamformer ($\mathbf{R}_r^{-1} \mathbf{a}$). Notably, we can estimate the fractional delay δ from $\mathbf{w}_{t,o}$ [see (3.39)] as

$$\delta = \frac{1}{\frac{|w_{t,o,p}|}{|w_{t,o,p+1}|} + 1}. \quad (3.41)$$

A special case considered most in the literature (e.g., [20], [38]–[42]) is that $\delta = 0$ (integer delay). In this case, the results shown above can be further simplified. Substituting $\delta = 0$ into (3.33), (3.39), and (3.40), we then obtain $J_{min} = 1 - \bar{\rho}$, $\mathbf{w}_{t,o} = \mathbf{e}_{\tau+1} \cdot \exp(i\psi)$, and $\mathbf{w}_{s,o} = \mathbf{R}_r^{-1} \mathbf{a} \cdot \exp[i(\psi - \theta)]$.

§ 3.2.3 Code Acquisition with Multipath Channel

In this subsection, we consider the scenario of a general multipath channel. We rewrite (3.7) as

$$\begin{aligned} \mathbf{r}(n) &= \sum_{l=1}^{L_1} \mathbf{a}_{1,l} \alpha_{1,l} \beta_{1,l}(n) + \sum_{k=2}^K \sum_{l=1}^{L_k} \mathbf{a}_{k,l} \alpha_{k,l} \beta_{k,l}(n) + \boldsymbol{\eta}(n) \\ &\triangleq \sum_{l=1}^L \mathbf{a}_l \alpha_l \beta_l(n) + \sum_{k=2}^K \sum_{l=1}^{L_k} \mathbf{a}_{k,l} \alpha_{k,l} \beta_{k,l}(n) + \boldsymbol{\eta}(n), \end{aligned} \quad (3.42)$$

in which we let $L_1 = L$, $\mathbf{a}_{1,l} = \mathbf{a}_l$, $\alpha_{1,l} = \alpha_l$, $\beta_{1,l}(n) = \beta_l(n)$, and $\tau_{1,l} = \tau_l$ for notational simplicity. Note that the transmitted power and carrier-phase offset have been absorbed into the channel gain. We let the multipath delay of the desired user, τ_l , be expressed as

$$\tau_l = p_l + \delta_l, \quad l = 1, \dots, L, \quad (3.43)$$

where $p_l \in [0, U)$ and $\delta_l \in [0, 1)$. Similar to the previous case, we have

$$\beta_l(n) = (1 - \delta_l) \cdot x(n - p_l) + \delta_l \cdot x(n - p_l - 1), \quad l = 1, \dots, L. \quad (3.44)$$

For simplicity, we also assume that $\tau_{l+1} > \tau_l$, $\tau_L < U$, and $p_{l+1} - p_l > 1$ [45] for all channels. Following the procedures described above, we can have (3.16) as

$$\mathbf{K} = \begin{bmatrix} \mathbf{0}_{(1 \times M)} \\ \vdots \\ \mathbf{0}_{(1 \times M)} \\ -\alpha_1^*(1 - \delta_1)\mathbf{a}_1^H \\ -\alpha_1^*\delta_1\mathbf{a}_1^H \\ \vdots \\ -\alpha_L^*(1 - \delta_L)\mathbf{a}_L^H \\ -\alpha_L^*\delta_L\mathbf{a}_L^H \\ \mathbf{0}_{(1 \times M)} \\ \vdots \\ \mathbf{0}_{(1 \times M)} \end{bmatrix}. \quad (3.45)$$

In (3.45), the $(p_1 + 1)$ -th, $(p_1 + 2)$ -th, \dots , $(p_L + 1)$ -th, and $(p_L + 2)$ -th rows are nonzero. Substituting (3.45) into (3.34) and rearranging the results, we have

$$(1 - \xi_{min})w_{t,o,j} = 0 \quad j \neq p_1, p_1 + 1, \dots, p_L, p_L + 1 \quad (3.46)$$

$$[(1 - \xi_{min})\mathbf{I} - \mathbf{\Omega}] \cdot \bar{\mathbf{w}}_{t,o} = \mathbf{0}, \quad (3.47)$$

where $\bar{\mathbf{w}}_{t,o(2L \times 1)} \triangleq [w_{t,o,p_1}, w_{t,o,p_1+1}, \dots, w_{t,o,p_L}, w_{t,o,p_L+1}]^T$,

$$\mathbf{\Omega} \triangleq \begin{bmatrix} \mathbf{\Omega}_{1,1} & \dots & \mathbf{\Omega}_{1,L} \\ & \ddots & \\ \mathbf{\Omega}_{L,1} & \dots & \mathbf{\Omega}_{L,L} \end{bmatrix}_{(2L \times 2L)}, \quad (3.48)$$

$$\mathbf{\Omega}_{u,v} \triangleq \mathbf{a}_u^H \mathbf{R}_r^{-1} \mathbf{a}_v \cdot \alpha_u^* \alpha_v \begin{bmatrix} (1 - \delta_u)(1 - \delta_v) & (1 - \delta_u)\delta_v \\ \delta_u(1 - \delta_v) & \delta_u\delta_v \end{bmatrix}, \quad (3.49)$$

$\mathbf{\Omega}_{u,v} = \mathbf{\Omega}_{v,u}^*$, $\{u, v\} \in \{1, \dots, L\}$, and the dimension of \mathbf{I} here is $2L \times 2L$. From (3.46), we can see that the tap-weights that do not corresponding to multipath delays are all zeros. Also, from (3.47) we know that $\bar{\mathbf{w}}_{t,o}$ is in the null space of $(1 - \xi_{min})\mathbf{I} - \mathbf{\Omega}$. In this general case, however, it is difficult to obtain a closed-form solution for $\bar{\mathbf{w}}_{t,o}$. As shown, an optimum $\bar{\mathbf{w}}_{t,o}$ is the eigenvector associated with the smallest eigenvalue of $\mathbf{I} - \mathbf{K}\mathbf{R}_r^{-1}\mathbf{K}^H$ [or (3.47)]. Once $\bar{\mathbf{w}}_{t,o}$ is obtained, $\mathbf{w}_{s,o}$ can be solved accordingly.

For multipath channels, we can also estimate δ_l , $l = 1, \dots, L$. To show this, we rewrite (3.18) as

$$\mathbf{K}\mathbf{w}_{s,o} + (1 - \xi_{min})\bar{\mathbf{w}}_{t,o} = \mathbf{0}. \quad (3.50)$$

Using (3.45), we have

$$\mathbf{w}_{t,o} = \frac{1}{1 - \xi_{min}} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \alpha_1^*(1 - \delta_1)\mathbf{a}_1^H \mathbf{w}_{s,o} \\ \alpha_1^*\delta_1\mathbf{a}_1^H \mathbf{w}_{s,o} \\ \vdots \\ \alpha_L^*(1 - \delta_L)\mathbf{a}_L^H \mathbf{w}_{s,o} \\ \alpha_L^*\delta_L\mathbf{a}_L^H \mathbf{w}_{s,o} \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (3.51)$$

It is simple to see that δ_l can be estimated by

$$\delta_l = \frac{1}{\frac{|w_{t,o,p_l}|}{|w_{t,o,p_l+1}|} + 1} \quad l = 1, \dots, L, \quad (3.52)$$

which is similar to (3.41). With known $\mathbf{w}_{s,o}$, $\mathbf{w}_{t,o}$, $\{\mathbf{a}_l\}_{l=1}^L$ and $\{\delta_l\}_{l=1}^L$, the channel estimate can be obtained accordingly.

Note that for derivation convenience, $p(t)$ is assumed to be a rectangular pulse. In real-world applications, we can apply other types of pulses as well, e.g., the squared-root-raised-cosine (SRRC) pulse. It can be shown that for the SRRC pulse, the received signal of the desired user has the same form as that in (3.25). The only difference is that the coefficients in (3.25) are replaced with $f(1 - \delta)$ and $f(\delta)$, where $f(\delta)$ stands for the raised-cosine function sampled at δT_c . The derivation is straightforward and then omitted here. All results derived above can be applied accordingly.

§ 3.3 Adaptive Implementation and Convergence Analysis

In Section 3.2, we have proposed a new scheme for code acquisition with the antenna array. Optimal-weights of the system are derived with the eigen-decomposition technique. However, the required computational complexity of the eigen-decomposition is on the order of $\mathcal{O}(U^3)$. In addition, the matrix inverse of \mathbf{R}_r is required in (3.20). To alleviate these problems, we propose to use an adaptive algorithm to approach the optimum filter-weights. The adaptive algorithm we consider is the LMS algorithm which is well-known for its simplicity and robustness. As shown, we have a unit-norm constraint on the temporal filter. Applying this constraint, we then obtain a constrained LMS algorithm. In what follows, we describe the algorithm and examine related issues such as the step size bound and the steady-state mean squared error (MSE). Besides, we also analyze the output SINR of the beamformer [$\gamma(n)$ in Fig. 3.2].

§ 3.3.1 Constrained LMS and Convergence Analysis

Rewriting (3.10), we have

$$\bar{J} = \mathbf{w}_v^H \mathbf{R}_v \mathbf{w}_v, \quad (3.53)$$

where

$$\mathbf{w}_v \triangleq [\mathbf{w}_t^T, \mathbf{w}_s^T]^T \quad (3.54)$$

$$\mathbf{v}(n) \triangleq [\mathbf{x}^T(n), -\mathbf{r}^T(n)]^T \quad (3.55)$$

$$\mathbf{R}_v \triangleq E\{\mathbf{v}(n)\mathbf{v}^H(n)\}. \quad (3.56)$$

The gradient of (3.53) is given by

$$\frac{\partial \bar{J}}{\partial \mathbf{w}_v^*} = \mathbf{R}_v \mathbf{w}_v. \quad (3.57)$$

Using (3.57), we can apply a gradient descent method to obtain the optimum solution, denoted as $\mathbf{w}_{v,o}$. However, \mathbf{R}_v needs to be estimated. The simplest estimate of \mathbf{R}_v is to use the instantaneous value from $\mathbf{v}(n)\mathbf{v}^H(n)$ and this yields a stochastic gradient descent algorithm, called the

LMS algorithm [34]. We then can have the filter adaptation as

$$\mathbf{w}_v(n+1) = \mathbf{w}_v(n) + \mu[-\mathbf{v}(n)\mathbf{v}^H(n)\mathbf{w}_v(n)], \quad (3.58)$$

where μ is the step size controlling the convergence rate. Recall that we have the constraint that $\|\mathbf{w}_t\|^2 = 1$. This constraint can be easily satisfied if normalization is performed on $\mathbf{w}_t(n)$ at each iteration. The overall adaptation procedure is given as:

$$e(n) = \mathbf{w}_v^H(n)\mathbf{v}(n) \quad (3.59)$$

$$\mathbf{H}(n) = \text{diag}\left\{\underbrace{\frac{1}{\|\mathbf{w}_t(n)\|}, \dots, \frac{1}{\|\mathbf{w}_t(n)\|}}_U, \underbrace{1, \dots, 1}_M\right\} \quad (3.60)$$

$$\mathbf{w}_v(n+1) = \mathbf{H}(n)\mathbf{w}_v(n) - \mu\mathbf{v}(n)e^*(n), \quad n = 0, 1, \dots, N-1, \quad (3.61)$$

where $\text{diag}\{\cdot\}$ denotes a diagonal matrix consisting of the arguments it includes, and N the maximum iteration number for the adaptive filter. To complete the acquisition, $|w_{t,j}(N)|^2$, $j = 0, 1, \dots, U-1$, are compared and the position corresponding to the maximum value is used for code-delay estimation. As we can see, $\mathbf{H}(n)$ normalizes $\mathbf{w}_t(n)$ at every iteration. To guarantee convergence, μ has to be selected properly. Here, we perform the mean convergence analysis to derive a step size bound. Subtracting $\mathbf{w}_{v,o} = [\mathbf{w}_{t,o}^T, \mathbf{w}_{s,o}^T]^T$ from both sides of (3.61), we have

$$\begin{aligned} \Delta\mathbf{w}_v(n+1) &= \Delta\mathbf{w}_v(n) + [\mathbf{H}(n) - \mathbf{I}]\mathbf{w}_v(n) - \mu\mathbf{v}(n)[\mathbf{w}_v^H(n)\mathbf{v}(n)]^* \\ &= \Delta\mathbf{w}_v(n) + [\mathbf{H}(n) - \mathbf{I}]\mathbf{w}_v(n) - \mu\mathbf{v}(n)\mathbf{v}^H(n)[\Delta\mathbf{w}_v(n) + \mathbf{w}_{v,o}] \\ &= [\mathbf{I} - \mu\mathbf{v}(n)\mathbf{v}^H(n)]\Delta\mathbf{w}_v(n) + [\mathbf{H}(n) - \mathbf{I}]\mathbf{w}_v(n) - \mu\mathbf{v}(n)e_o^*(n), \end{aligned} \quad (3.62)$$

where

$$e_o(n) \triangleq \mathbf{w}_{v,o}^H\mathbf{v}(n) \quad (3.63)$$

$$\Delta\mathbf{w}_v(n) \triangleq \mathbf{w}_v(n) - \mathbf{w}_{v,o}. \quad (3.64)$$

Note that the dimension of \mathbf{I} here is $(U + M) \times (U + M)$. Taking the statistical expectation of (3.62), applying the direct-averaging method [34], and using the orthogonality principle, we then have

$$E\{\Delta \mathbf{w}_v(n+1)\} = [\mathbf{I} - \mu \mathbf{R}_v]E\{\Delta \mathbf{w}_v(n)\} + [E\{\mathbf{H}(n)\} - \mathbf{I}]E\{\mathbf{w}_v(n)\}. \quad (3.65)$$

Let $\mathbf{\Lambda} = \text{diag}\{\lambda_{v,1}, \dots, \lambda_{v,U+M}\}$ with $\lambda_{v,j}$ being an eigenvalue of \mathbf{R}_v , and \mathbf{Q} be a matrix consisting of the eigenvectors of \mathbf{R}_v . Multiplying (3.65) with \mathbf{Q}^H and letting $\mathbf{g}(n) = \mathbf{Q}^H E\{\Delta \mathbf{w}_v(n)\}$, we have

$$\mathbf{g}(n+1) = [\mathbf{I} - \mu \mathbf{\Lambda}]\mathbf{g}(n) + \mathbf{Q}^H [E\{\mathbf{H}(n)\} - \mathbf{I}]E\{\Delta \mathbf{w}_v(n)\}. \quad (3.66)$$

Since $\mathbf{w}_t(n)$ is normalized at every iteration and the step size is usually small, it is reasonable to assume that $\mathbf{H}(n) \approx \mathbf{I}$ and the second term in the right-hand side of (3.66) can be ignored.

Iterating (3.66), we obtain

$$\mathbf{g}(n) = [\mathbf{I} - \mu \mathbf{\Lambda}]^n \mathbf{g}(0). \quad (3.67)$$

Thus, for (3.66) to converge, the following condition must be satisfied

$$0 < \mu < \frac{2}{\lambda_{v,max}}, \quad (3.68)$$

where $\lambda_{v,max}$ denotes the maximum eigenvalue of \mathbf{R}_v . This result is the same as the conventional LMS algorithm [34]. From (3.67), we can also see that $\mathbf{g}(\infty) = \mathbf{0}$. In other words, $E\{\mathbf{w}_v(n)\} = \mathbf{w}_{v,o}$ when $n \rightarrow \infty$.

Note that while the conventional LMS algorithm requires $2(U + M)$ multiplications per iteration, the constrained LMS algorithm developed here needs extra U multiplications for calculation of $\|\mathbf{w}_t(n)\|$ and extra U divisions for normalization [see (3.61)].

§ 3.3.2 Steady-state MSE Analysis

We now derive the steady-state MSE of the constraint LMS algorithm. Invoking the direct-averaging method [34] and using (3.62), we can write the correlation matrix of the tap-weight

error vector as

$$\begin{aligned}
\mathbf{P}(n+1) &\triangleq E\{\Delta\mathbf{w}_v(n+1)\Delta\mathbf{w}_v^H(n+1)\} \\
&= [\mathbf{I} - \mu\mathbf{R}_v]\mathbf{P}(n)[\mathbf{I} - \mu\mathbf{R}_v] \\
&\quad + \mu^2 J_{min}\mathbf{R}_v + E\{[\mathbf{H}(n) - \mathbf{I}]\mathbf{w}_v(n)\mathbf{w}_v^H(n)[\mathbf{H}(n) - \mathbf{I}]\}. \tag{3.69}
\end{aligned}$$

As stated, $\mathbf{w}_t(n)$ is normalized at every iteration and the step size is usually small. Thus, $\mathbf{H}(n) \approx \mathbf{I}$ and the last term in the right-hand side of (3.69) can be ignored. Let $\bar{\mathbf{P}}(n) \triangleq \mathbf{Q}^H\mathbf{P}(n)\mathbf{Q}$ and observe that $\mathbf{Q}^H\mathbf{R}_v\mathbf{Q} = \Lambda$. Pre-multiplying and post-multiplying both sides of (3.69) with \mathbf{Q}^H and \mathbf{Q} , respectively, we have

$$\bar{\mathbf{P}}(n+1) = [\mathbf{I} - \mu\Lambda]\bar{\mathbf{P}}(n)[\mathbf{I} - \mu\Lambda] + \mu^2 J_{min}\Lambda. \tag{3.70}$$

Let the j -th element on the diagonal of $\bar{\mathbf{P}}(n)$ be $\bar{p}_j(n)$. Then,

$$\bar{p}_j(n+1) = (1 - \mu\lambda_{v,j})^2\bar{p}_j(n) + \mu^2 J_{min}\lambda_{v,j} \quad j = 1, \dots, U+M. \tag{3.71}$$

When $n \rightarrow \infty$, $\bar{p}_j(n+1) \approx \bar{p}_j(n)$. From (3.71), we have

$$\bar{p}_j(\infty) = \frac{\mu J_{min}}{2 - \mu\lambda_{v,j}}. \tag{3.72}$$

The additional MSE due to the use of the LMS algorithm is generally referred to as the excess MSE, denoted as $J_{ex}(\infty)$. From [34], we then have

$$J_{ex}(\infty) = \sum_{j=1}^{U+M} \bar{p}_j(\infty)\lambda_{v,j} = J_{min} \sum_{j=1}^{U+M} \frac{\mu\lambda_{v,j}}{2 - \mu\lambda_{v,j}}. \tag{3.73}$$

Denote the steady-state MSE of the LMS adaptation as J_{ss} . Finally, we have

$$J_{ss} = J_{min} + J_{ex}(\infty). \tag{3.74}$$

§ 3.3.3 Output SINR of Beamformer

Now, let us analyze the output SINR of the beamformer. For the scenario considered in Section 3.2.2, we have the beamformer output as

$$\begin{aligned}\gamma(n) &\triangleq \mathbf{w}_s^H(n)\mathbf{r}(n) \\ &= \mathbf{w}_s^H(n) \left(\mathbf{a} \exp(-i\theta) [(1-\delta) \cdot x(n-p) + \delta \cdot x(n-p-1)] + \mathbf{z}_A(n) \right),\end{aligned}\quad (3.75)$$

where $\mathbf{z}_A(n)$ consists of MAI and noise and

$$\mathbf{z}_A(n) = \sum_{k=2}^K \mathbf{a}_k \alpha_k \beta_k(n) \exp(-i\theta_k) + \boldsymbol{\eta}(n). \quad (3.76)$$

Using (3.40), we can find the output SINR of the optimum beamformer, denoted as SINR_o , as

$$\begin{aligned}\text{SINR}_o &= \frac{\mathbf{w}_{s,o}^H \mathbf{R}_A \mathbf{w}_{s,o}}{\mathbf{w}_{s,o}^H \mathbf{R}_{z_A} \mathbf{w}_{s,o}} \\ &= \frac{[(1-\delta)^2 + \delta^2] \cdot |\mathbf{a}^H \mathbf{R}_r^{-1} \mathbf{a}|^2}{\mathbf{a}^H \mathbf{R}_r^{-1} \mathbf{R}_{z_A} \mathbf{R}_r^{-1} \mathbf{a}},\end{aligned}\quad (3.77)$$

where $\mathbf{R}_A \triangleq [(1-\delta)^2 + \delta^2] \mathbf{a} \mathbf{a}^H$ and $\mathbf{R}_{z_A} \triangleq E\{\mathbf{z}_A(n) \mathbf{z}_A^H(n)\}$. Since we use adaptive filter-weights to approximate the optimum weights, we have to include the excess MSE in the SINR calculation. Thus, we can rewrite (3.77) as

$$\text{SINR}_o = \frac{[(1-\delta)^2 + \delta^2] \cdot |\mathbf{a}^H \mathbf{R}_r^{-1} \mathbf{a}|^2}{\mathbf{a}^H \mathbf{R}_r^{-1} \mathbf{R}_{z_A} \mathbf{R}_r^{-1} \mathbf{a} + J_{ex}(\infty)}, \quad (3.78)$$

where $J_{ex}(\infty)$ is that shown in (3.73). For the special case that $\delta = 0$, (3.78) is reduced to

$$\text{SINR}_o = \frac{|\mathbf{a}^H \mathbf{R}_r^{-1} \mathbf{a}|^2}{\mathbf{a}^H \mathbf{R}_r^{-1} \mathbf{R}_{z_A} \mathbf{R}_r^{-1} \mathbf{a} + J_{ex}(\infty)}. \quad (3.79)$$

Similarly, we can derive the corresponding result for the scenario considered in Section 3.2.3. The beamformer output here is given by

$$\gamma(n) = \mathbf{w}_s^H(n) \left(\sum_{l=1}^L \mathbf{a}_l \alpha_l [(1-\delta_l) \cdot x(n-p_l) + \delta_l \cdot x(n-p_l-1)] + \mathbf{z}_M(n) \right), \quad (3.80)$$

where

$$\mathbf{z}_M(n) = \sum_{k=2}^K \sum_{l=1}^{L_k} \mathbf{a}_{k,l} \alpha_{k,l} \beta_{k,l}(n) + \boldsymbol{\eta}(n). \quad (3.81)$$

The output SINR of the optimum beamformer is then

$$\text{SINR}_o = \frac{\mathbf{w}_{s,o}^H \mathbf{R}_M \mathbf{w}_{s,o}}{\mathbf{w}_{s,o}^H \mathbf{R}_{z_M} \mathbf{w}_{s,o}}, \quad (3.82)$$

where $\mathbf{R}_M \triangleq \sum_{l=1}^L |\alpha_l|^2 [(1 - \delta_l)^2 + \delta_l^2] \mathbf{a}_l \mathbf{a}_l^H$ and $\mathbf{R}_{z_M} \triangleq E\{\mathbf{z}_M(n) \mathbf{z}_M^H(n)\}$. For the adaptive approach, we have the output SINR as

$$\text{SINR}_o = \frac{\mathbf{w}_{s,o}^H \mathbf{R}_M \mathbf{w}_{s,o}}{\mathbf{w}_{s,o}^H \mathbf{R}_{z_M} \mathbf{w}_{s,o} + J_{ex}(\infty)}. \quad (3.83)$$

§ 3.4 Performance Analysis

In general, acquisition performance can be measured by the probability of correct acquisition and the mean acquisition time. In this section, we will evaluate the performance of the proposed scheme with these two measures.



§ 3.4.1 Probability of Correct Acquisition

In the proposed scheme, τ is estimated with the quantities of $|w_{t,j}(N)|^2$, $j = 0, 1, \dots, U - 1$. To evaluate the probability of correct acquisition, we have to characterize the statistical property of $|w_{t,j}(n)|^2$ first. From the analysis shown in the previous section, we see that in the steady-state ($n \rightarrow \infty$), $\mathbf{w}_t(n)$ has a mean vector of

$$E\{\mathbf{w}_t(n)\} = \mathbf{w}_{t,o}. \quad (3.84)$$

Its covariance matrix, denoted as $\mathbf{C} \triangleq E\{[\mathbf{w}_t(n) - \mathbf{w}_{t,o}][\mathbf{w}_t(n) - \mathbf{w}_{t,o}]^H\}$, can be derived as follows. As a common practice, the step size is usually small. Thus, we can use the Taylor expansion to expand $1/(2 - \mu\lambda_{v,j})$ in (3.72) with respect to $\mu\lambda_{v,j} = 0$. Then, we have

$$\frac{1}{(2 - \mu\lambda_{v,j})} = \frac{1}{2} + \frac{1}{4}\mu\lambda_{v,j} + \dots \quad (3.85)$$

From (3.70), it can be seen that the matrix $\bar{\mathbf{P}}(n)$ will become diagonal as $n \rightarrow \infty$. Using this property and truncating the terms higher than the first-order in (3.85), we then have

$$\bar{\mathbf{P}}(n) = \frac{\mu}{2} J_{min} \mathbf{I} + \frac{\mu^2}{4} J_{min} \mathbf{\Lambda}. \quad (3.86)$$

Pre-multiplying and post-multiplying both sides of (3.86) with \mathbf{Q} and \mathbf{Q}^H , we obtain

$$\mathbf{P}(n) = \frac{\mu J_{min}}{2} \mathbf{I} + \frac{\mu^2 J_{min}}{4} \mathbf{R}_v. \quad (3.87)$$

Note that the $U \times U$ upper-left submatrix of $\mathbf{P}(n)$ corresponds to the covariance matrix \mathbf{C} . Finally, we have

$$\mathbf{C} = J_{min} \left[\frac{\mu}{2} + \frac{\mu^2}{4} \right] \mathbf{I} \approx \frac{\mu J_{min}}{2} \mathbf{I}. \quad (3.88)$$

From (3.88), we can see that the filter taps are independent and identically distributed (i.i.d.). Let's consider the AWGN channel with an integer delay first. When $\mathbf{w}_t(n)$ approaches its steady-state, $\mathbf{w}_t(n)$ can be assumed to have a Gaussian distribution [33]. From (3.84)–(3.88), we can find that when n is large, $|w_{t,\tau}(n)|^2$ has a noncentral chi-square distribution with two degrees of freedom, whereas other taps $|w_{t,j}(n)|^2$, $j \neq \tau$, have chi-square distributions. Let $Z_j \triangleq |w_{t,j}(n)|^2|_{n \rightarrow \infty}$, $j = 0, \dots, U-1$. The probability density functions for the filter weights are then

$$p_{Z_\tau}(y) = \frac{1}{2\sigma_w^2} \exp\left(-\frac{|w_{t,o,\tau}|^2 + y}{2\sigma_w^2}\right) I_0\left(\sqrt{y} \frac{|w_{t,o,\tau}|}{\sigma_w^2}\right) \quad y > 0 \quad (3.89)$$

$$p_{Z_j}(y) = \frac{1}{2\sigma_w^2} \exp\left(-\frac{y}{2\sigma_w^2}\right) \quad y > 0, j \neq \tau, \quad (3.90)$$

where $I_0(\cdot)$ is the zero-th order modified Bessel function of the first kind and $\sigma_w^2 \triangleq \mu J_{min}/2$. The conditional probability of correct acquisition for the AWGN channel, denoted as $P_{cc}(\cdot)$, is given by

$$\begin{aligned} P_{cc}(z_\tau) &= \Pr(z_\tau > Z_0, \dots, z_\tau > Z_{\tau-1}, z_\tau > Z_{\tau+1}, \dots, z_\tau > Z_{U-1} | Z_\tau = z_\tau) \\ &= \prod_{j=0, j \neq \tau}^{U-1} \Pr(z_\tau > Z_j | Z_\tau = z_\tau) \\ &= \left[\int_0^{z_\tau} \frac{1}{2\sigma_w^2} \exp\left(-\frac{y}{2\sigma_w^2}\right) dy \right]^{U-1}. \end{aligned} \quad (3.91)$$

Note that the i.i.d. property has been applied in (3.91). The probability of correct acquisition, denoted as P_c , is then

$$P_c = \int_0^\infty P_{cc}(y) \cdot p_{Z_\tau}(y) dy. \quad (3.92)$$

Note that $p_{Z_\tau}(y)$ in (3.92) is the function shown in (3.89) with $|w_{t,o,\tau}|^2 = 1$.

Next, we consider the scenario of the AWGN channel with a fractional delay. In this case, two nonzero successive peaks in optimum filter-weights will result [see (3.39)]. As mentioned, $\tau = p + \delta$. Thus, Z_p and Z_{p+1} will be the peaks. We will claim correct acquisition if either Z_p or Z_{p+1} is the maximum of all Z_j . Define two events as

$$\begin{aligned} \mathbf{E}_p &\triangleq \{z_p > Z_0, \dots, z_p > Z_{p-1}, z_p > Z_{p+2}, \dots, z_p > Z_{U-1} | Z_p = z_p\} \\ \mathbf{E}_{p+1} &\triangleq \{z_{p+1} > Z_0, \dots, z_{p+1} > Z_{p-1}, z_{p+1} > Z_{p+2}, \dots, z_{p+1} > Z_{U-1} | Z_{p+1} = z_{p+1}\}. \end{aligned} \quad (3.93)$$

Thus, correct acquisition corresponds to the event $\mathbf{E}_p \cup \mathbf{E}_{p+1}$. Then, the conditional probability of correct acquisition can be formulated as

$$P_{cc}(z_p, z_{p+1}) \triangleq \Pr(\mathbf{E}_p \cup \mathbf{E}_{p+1}) = \Pr(\mathbf{E}_p) + \Pr(\mathbf{E}_{p+1}) - \Pr(\mathbf{E}_p) \cdot \Pr(\mathbf{E}_{p+1}), \quad (3.94)$$

where

$$\begin{aligned} \Pr(\mathbf{E}_p) &= \prod_{j=0, j \neq p, p+1}^{U-1} \Pr(z_p > Z_j | Z_p = z_p) \\ &= \left[\int_0^{z_p} \frac{1}{2\sigma_w^2} \exp\left(-\frac{y}{2\sigma_w^2}\right) dy \right]^{U-2} \end{aligned} \quad (3.95)$$

$$\begin{aligned} \Pr(\mathbf{E}_{p+1}) &= \prod_{j=0, j \neq p, p+1}^{U-1} \Pr(z_{p+1} > Z_j | Z_{p+1} = z_{p+1}) \\ &= \left[\int_0^{z_{p+1}} \frac{1}{2\sigma_w^2} \exp\left(-\frac{y}{2\sigma_w^2}\right) dy \right]^{U-2}. \end{aligned} \quad (3.96)$$

Note that both Z_p and Z_{p+1} are functions of δ (though the dependence is not shown explicitly).

Thus, the probability of correct acquisition is

$$P_c = \int_0^1 \int_0^\infty \int_0^\infty P_{cc}(\epsilon, \vartheta) \cdot p_{Z_p}(\epsilon) \cdot p_{Z_{p+1}}(\vartheta) d\epsilon d\vartheta d\bar{\delta}, \quad (3.97)$$

where ϵ , ϑ , and $\bar{\delta}$ stand for the dummy variables of Z_p , Z_{p+1} , and δ , respectively. Note that $p_{Z_p}(\cdot)$ and $p_{Z_{p+1}}(\cdot)$ are obtained by replacing $|w_{t,o,\tau}|^2$ with $|w_{t,o,p}|^2$ and $|w_{t,o,p+1}|^2$ in (3.89), respectively.

For general multipath channels, we can also evaluate the probability of correct acquisition. Since the procedure is similar and the result is complicated, we omit the details here.

§ 3.4.2 Mean Acquisition Time

Mean acquisition time usually serves an indicator showing that how fast a receiver can complete the acquisition. It is generally derived with a Markov chain model [1], [10]. Since our system is different from the conventional serial-search correlator, the commonly used model [1], [39] cannot be applied here. Apply the model derived in Fig. 2.4 to our system and let the probability of acquisition error is P_e , which is equal to $1 - P_c$. As the figure shows that if the acquisition fails, the system will wait for a period of time T_p (chips) to re-start the acquisition. Here, T_p is generally referred to as the penalty time [1]. The transfer function of the model can be [27]

$$H_{acq}(\mathbf{z}) = \frac{(1 - P_e)\mathbf{z}}{1 - P_e\mathbf{z}^{T_p+N}}, \quad (3.98)$$

where \mathbf{z} is the unit-delay operator. The mean acquisition time can be found as

$$T_{acq} \triangleq \frac{d}{d\mathbf{z}} H_{acq}(\mathbf{z})|_{\mathbf{z}=1} = N + (N + T_p) \frac{P_e}{1 - P_e}. \quad (3.99)$$

From (3.99), it is easy to see that if $P_e = 0$, $T_{acq} = N$. Thus, N can serve as a performance bound for T_{acq} .

§ 3.5 Simulations

In this section, we report simulation results to demonstrate the effectiveness of the proposed algorithm. We set common parameters used for all simulations as: $M = 8$, $K = 5$, $T_p = 100U$ chips, $\mathbf{w}_s(0) = \mathbf{0}$, and $w_{t,j}(0) = 1/\sqrt{U}$, $j = 0, 1, \dots, U - 1$.

§ 3.5.1 AWGN Channel

Let the power of each jammer be 3 dB stronger than that of the desired user (i.e., $|\alpha| = 1$ and $|\alpha_k| = 2$ for $k = 2, \dots, K$). Also, $\sigma_\eta^2 = 1$, $U = 16$, and $\mu = 6 \times 10^{-3}$. The DoAs are set as $\{\phi_k\}_{k=1}^K = \{0.3152, 0.5586, 0.7754, -0.2014, -0.5236\}$ (radians). Simulations with 400 independent trials are conducted. Figs. 3.3–3.5 show the adaptation results for the proposed algorithm. Also shown in these figures are the corresponding theoretical predictions derived in Section 3.3 and 3.4. Fig. 3.3 shows the convergence curves for averaged $|w_{t,p}(n)|^2$ and $|w_{t,p+2}(n)|^2$. Note that the convergence behaviors for $|w_{t,j}(n)|^2$, $j \neq p, p+1$, are all similar. Two delay scenarios with $\delta = 0$ and $\delta = 0.5$ are considered. As we can see, $|w_{t,p}(n)|^2$ converges to its optimum values, one for $\delta = 0$ and 0.5 for $\delta = 0.5$. By contrast, $|w_{t,p+2}(n)|^2$ converges to a small value close to zero. Fig. 3.4 gives the convergence curves for MSE. As expected, the MSE for $\delta = 0.5$ is larger than that for $\delta = 0$. Using (3.74), we obtain theoretical steady-state MSEs for $\delta = 0$ and 0.5 as 0.158 and 0.278, respectively. From the figure, it is apparent that the experimental result matches the theoretical one quite well. The corresponding output SINR for the beamformer is shown in Fig. 3.5. The theoretical SINRs are calculated with (3.79) and (3.78), and they are 7.34 dB and 4.25 dB for $\delta = 0$ and $\delta = 0.5$, respectively. As seen, the SINR is increased from -12 dB to 7.34 dB in 700 iterations (for $\delta = 0$). The beamformer in the proposed algorithm effectively suppresses the interference.

As mentioned in (3.41), δ can be estimated from $w_{t,p}(n)$ and $w_{t,p+1}(n)$. Simulations are carried out to evaluate the performance of the estimation. We randomly generate p and δ for 500 trials. The MSE, defined as $E\{|\hat{\delta}(N) - \delta|^2\}$, is used as the performance measure, where $\hat{\delta}(N)$ denotes an estimate of δ at $n = N$. Here, we let $\mu = 4 \times 10^{-3}$ and $N = 1000$ chips for filter adaptation. For simplicity, we perform estimation of δ only when $|w_{t,p}(N)|^2$ and $|w_{t,p+1}(N)|^2$ are the first two maximums of all weights. Fig. 3.6 shows the simulation result. As we can see, the estimation errors is small. For an SNR (per chip) of -2 dB, the MSE is only 10^{-3} . The SNR here is defined as $|\alpha|^2/\sigma_\eta^2$ and $|\alpha| = 1$. Note that in the same figure, the result

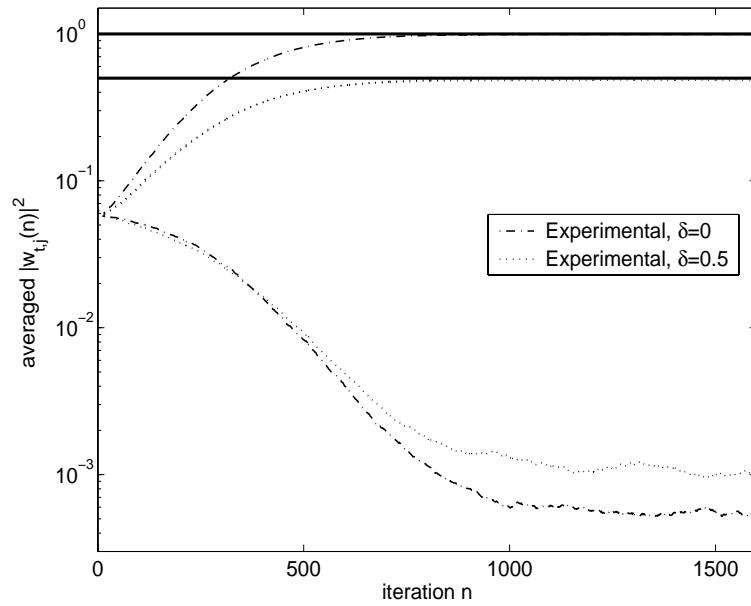


Figure 3.3: Convergence curve for squared temporal filter tap-weights, $|w_{t,p}(n)|^2$ and $|w_{t,p+2}(n)|^2$. Theoretical values are shown with horizontal lines and obtained from (3.39).

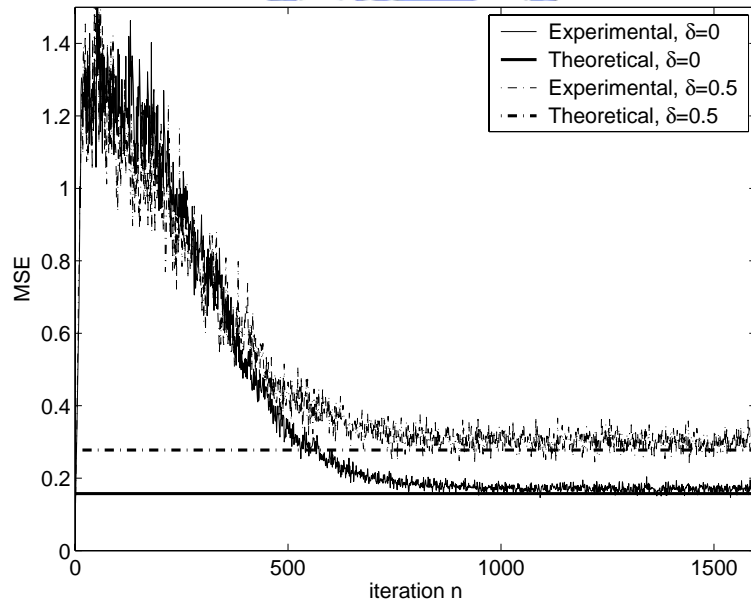


Figure 3.4: Convergence curve for MSE. Theoretical values are obtained from (3.74).

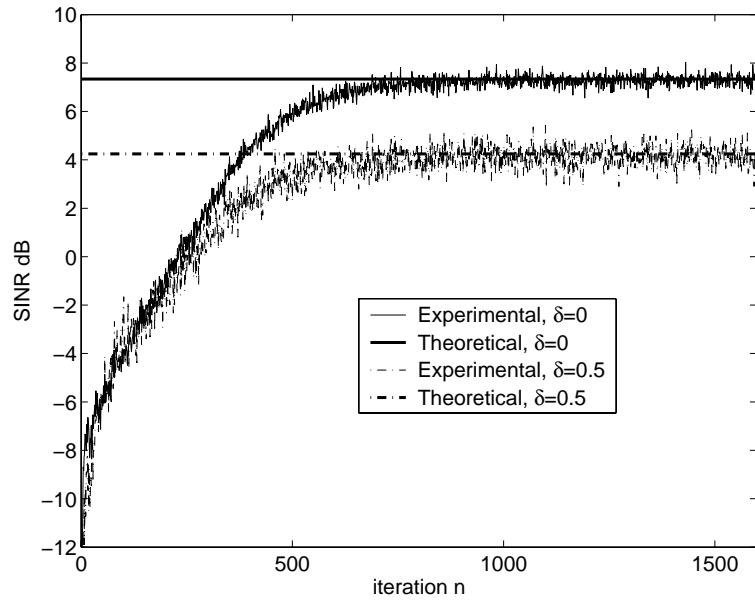


Figure 3.5: Convergence curve for SINR. Theoretical values are obtained from (3.78) and (3.79).

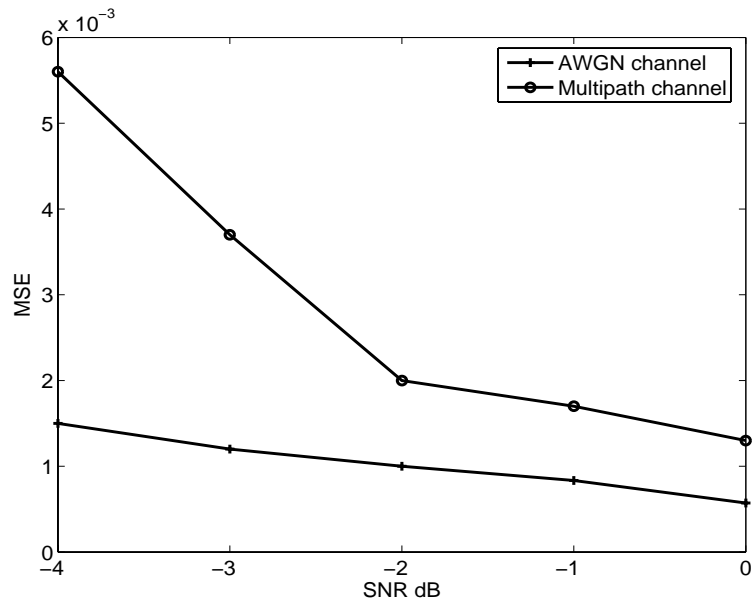


Figure 3.6: MSE for fractional delay estimate in AWGN and multipath channel.

for a multipath channel, which will be discussed later, is also included.

Next, we will consider the performance of acquisition. Let's first examine the probability of correct acquisition [(3.92) and (3.97)]. Fig. 3.7 gives the simulation results for various step sizes. Here, we let $U = 16$, $N = 400$ and 600 , and the array input SINR (per chip) be -15 dB. Theoretical values calculated with (3.92) and (3.97) are also shown for comparison. It is clear from Fig. 3.7 that the experimental results highly agree with the theoretical ones. When $\delta \neq 0$, the probability of correct acquisition is somewhat lower. This is due to the fact that optimum values of $|w_{t,p}(n)|^2$ and $|w_{t,p+1}(n)|^2$ are smaller than one. Also, we can see that the experimental probability of correct acquisition with $N = 400$ is different from the theoretical one. This indicates that N is not sufficiently large and adaptive filters have not reach their steady-states. As we will see below, experimental results of $N = 600$ can be very close to the results calculated with (3.97).

We then substitute the experimental probabilities ($U = 64$) shown in Fig. 3.7 into (3.99) to derive the mean acquisition time. The result is shown in Fig. 3.8. It is seen that the proposed algorithm can acquire an integer delay in a short period of time. For example, T_{acq} is 402 chips when μ is 3×10^{-3} . The mean acquisition time for fractional delay is slightly larger than that for integer delay. In Fig. 3.8, N is selected somewhat arbitrarily and the value may be not optimal. Fig. 3.9 shows the mean acquisition time for various N with $\mu = 3 \times 10^{-3}$ and 5×10^{-3} . The lower bound being $T_{acq} = N$ serves a performance benchmark. For integer code-delay, when N is greater than 350, the mean acquisition time becomes close to the lower bound. Also, we can see that the minimum mean acquisition time is around $N = 300$ for $\mu = 3 \times 10^{-3}$. The acquisition performance for fractional delay will be poorer if N is too small. For N is larger than 400, it becomes close to that in integer delay. The minimum mean acquisition time is around $N = 350$ for $\mu = 5 \times 10^{-3}$.

From Fig. 3.9, we can see that there is an optimum N for a given array input SINR. To let the system be operated in its optimum conditions all the time, we can build a table for optimum N 's (vs. input SINR) off-line, and then obtain an optimum N with a table lookup on-line. If we

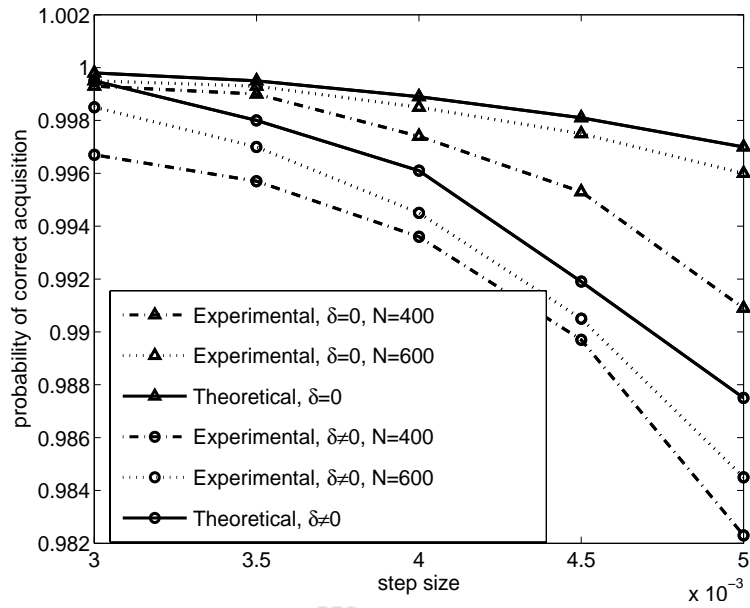


Figure 3.7: Probability of correct acquisition versus step size. Theoretical values are obtained from (3.92) and (3.97).

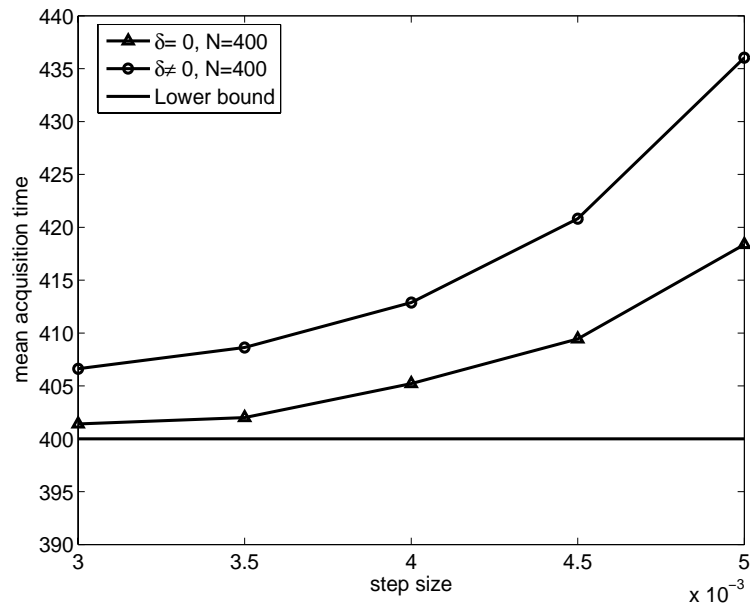


Figure 3.8: Experimental mean acquisition time (in chips) versus step size for AWGN channel.

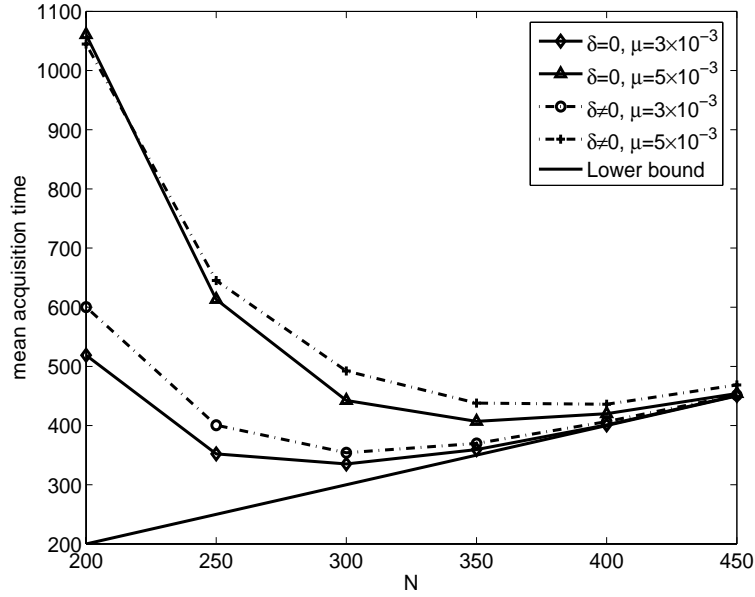


Figure 3.9: Experimental mean acquisition time (in chips) versus N and step size for AWGN channel.

assume that the power of the received signal is dominated by MAI, an estimate of input SINR can be $M \times E\{|\gamma(n)|^2\} / E\{\|\mathbf{r}(n)\|^2\}$. We have found that this SINR estimate can converge fast and provide good results.

§ 3.5.2 Multipath Channels

For the scenario of multipath channels, we let the number of channel paths be two ($L_k = 2$) for all users. Also, let $N = 2600$ and $\mu = 4 \times 10^{-3}$. Other related parameters used in simulations are summarized in Table 3.1. This setting leads the antenna array operating in a heavily loaded case (i.e., the number of overall multipaths is greater than that of array sensors). Simulations with 500 trials are conducted. Fig. 3.10 shows some experimental beam-patterns derived from $\mathbf{w}_s(N)$ and the theoretical beam-pattern from Section 3.2.3. Note that the arrow signs indicate the DoAs of all users, and only the DoAs of the desired user are labeled. The beamformer

Table 3.1: Parameters used for simulations in multipath scenario

| | $p_{k,1}, p_{k,2}$ | $\delta_{k,1}, \delta_{k,2}$ | $\phi_{k,1}, \phi_{k,2}$ (radians) | $ \alpha_{k,1} , \alpha_{k,2} $ |
|---------|--------------------|------------------------------|------------------------------------|----------------------------------|
| $k = 1$ | 3, 7 | 0.44, 0.65 | -0.5236, 0 | 1, 0.85 |
| $k = 2$ | 8, 10 | 0.81, 0.29 | -0.8481, -0.4115 | 1, 1 |
| $k = 3$ | 3, 11 | 0.74, 0.45 | -0.2014, 0.3047 | 1, 1 |
| $k = 4$ | 6, 14 | 0.52, 0.80 | 1.1198, -0.3047 | 1, 1 |
| $k = 5$ | 0, 5 | 0.35, 0.62 | 1.0353, 0.1708 | 1, 1 |

forms two main-beams collecting the desired signals coming from the angles 0 and -0.5236 radians. From Fig. 3.10, we see that some interference cannot be deeply nulled. This is because their incident angles are close to the desire user's DoAs. The convergence behavior of the MSE and SINR is similar to those shown previously and the corresponding figures are then omitted. Specifically, we find that the steady-state MSE is 0.35 and output SINR of the beamformer is 1.98 dB. We then examine the performance of fractional delay estimation. We randomly generate code-delays for all paths and all users for acquisition and calculate the MSE, defined as $E\{\sum_{l=1}^2 |\hat{\delta}_l(N) - \delta_l|^2\}$. The SNR here is defined as $(\sum_{l=1}^2 |\alpha_l|^2)/\sigma_\eta^2$. Fig. 3.6 gives the result with 500 trials. We can see that the performance is worse than that in AWGN case. Also, it is more sensitive to the SNR. When SNR is low, the performance is seriously affected.

§ 3.5.3 Performance Comparison

Finally, we compare the proposed scheme with the correlator-based scheme described in [39]. Since the scheme in [39] does not consider the case with fractional delay, we let the code-delay be integer. Also, the channel is an AWGN channel. We assume that $U = 64$ and 128, $|\alpha_1| = 1, |\alpha_2| = \dots = |\alpha_K|$, and $\sigma_\eta^2 = 2$. The setting of DoAs is the same as that in Section 3.5.1. For the proposed system, we let $P_e = 0.01$. Then, we experimentally search for an optimal set of $\{\mu, N\}$ giving minimal mean acquisition time for each input SINR. For the

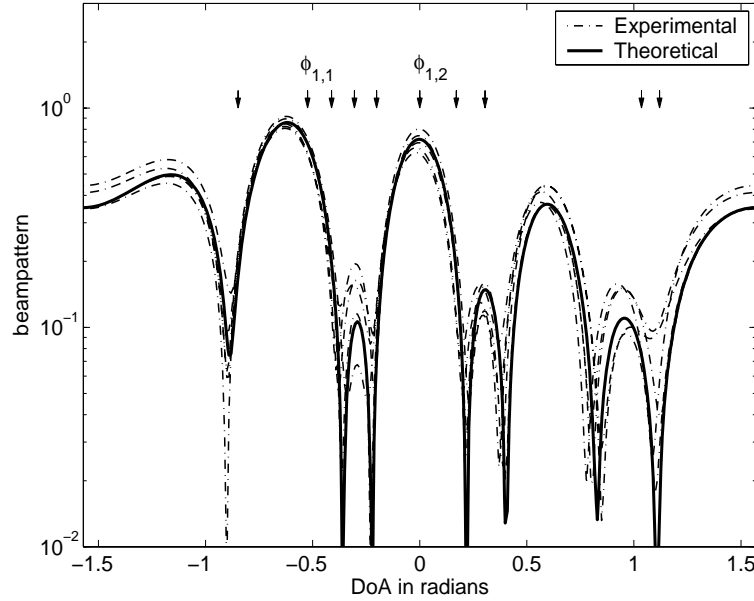


Figure 3.10: Experimental and theoretical beam-patterns for multipath channel. Arrow signs indicate DoAs of all users; the labeled are DoAs of the desired user.

system in [39], we let $P_f = 0.01$ and $N_s = 1$. Note that $K_p = T_p/N_c$ [in (3.5)]. As addressed in [39], the convergent filter-weight vector is not exactly identical to the steering vector of the desired user, and there exists a gap between the experimental and theoretical performance (for $M \geq 2$), especially at low SINR. In other words, the theoretical threshold derived from [39] may not guarantee $P_f = 0.01$. Let $N_c = JU$ where J is a positive scalar. To ensure a fair comparison, we experimentally search for the threshold, J , and the step size to achieve the optimum performance. Fig. 3.11 shows the mean acquisition times versus the array input SINR for the correlator-based and proposed schemes. From the figure, we can see that the proposed system significantly outperforms the correlator-based system, especially for low SINR. For example, when the SINR is -30 dB, the performance gap between the proposed system and the correlator-based system exceeds two orders of magnitude. For the the proposed scheme, we find that the mean acquisition time of $U = 128$ is only slightly larger than that of $U = 64$. While the mean acquisition time for the correlator-based scheme can be significantly enlarged when

U is changed from 64 to 128. The poor performance of the correlator-based algorithm stems

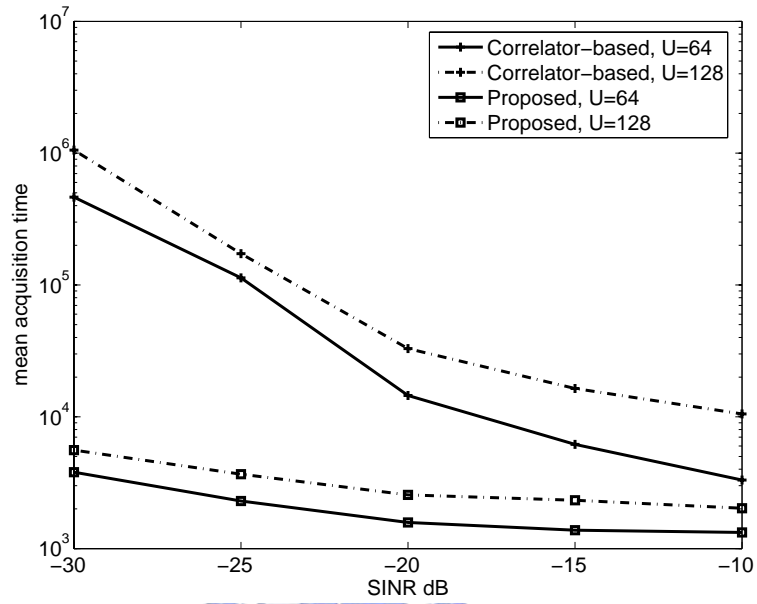


Figure 3.11: Mean acquisition time (in chips) comparison.

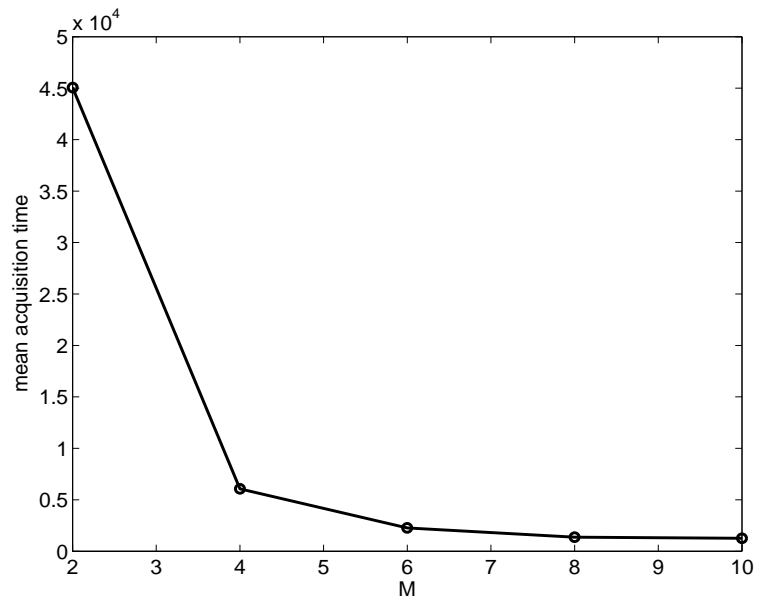


Figure 3.12: Mean acquisition time (in chips) versus M (size of antenna array).

from the slow convergence of the adaptive filter and its necessity for code-phase searching. The proposed scheme simultaneously performs beamforming and code acquisition yielding much better performance in interference suppression and filter convergence.

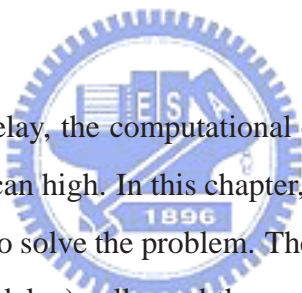
Finally, we consider the effect of M on the mean acquisition time. Here, we let $U = 64$ and $P_e = 0.01$. With the previous setting, we show the result in Fig. 3.12. As seen, the performance can be rapidly degraded when $M < K$ ($K = 5$). When M is greater than K , most interference can be effectively suppressed, and no obvious improvement is observed.

§ 3.6 Conclusions

In this chapter, we propose a novel adaptive antenna array for code acquisition. Unlike the correlator-based serial-search scheme, the proposed system can simultaneously perform beamforming and code acquisition. Another distinct feature is that the proposed algorithm can deal with both integer and fractional code-delays. For multipath channels, the proposed system can acquire multipath delays and serve as a channel estimator. We also theoretically analyze the properties and performance of the proposed algorithm. Closed-form solutions for optimum solutions, steady-state MSE, and SINR are derived. We also show that experimental results highly agree with analytical ones. Simulations results show that the proposed system significantly outperforms the correlator-based one in [39]. In this chapter, we consider the scenario of single transmit antenna. However, the proposed algorithm can also be applied to the scenario of multiple transmit antennas. Acquisition in the multiple-input-multiple-output (MIMO) system is a potential topic for further research.

Chapter 4

Low-Complexity Adaptive Array Code Acquisition



In the scenario of large code-delay, the computational complexity of the temporal filter in the scheme proposed in Chapter 3 can be high. In this chapter, we propose a low-complexity adaptive array code acquisition scheme to solve the problem. The main idea is to divide the whole delay uncertainty range into several (delay) cells, and then sequentially search for the code-delay of the desired user among those cells. This is essentially a serial search technique, being able to shorten the filter-length of the temporal filter. As a result, the computational complexity can be reduced. As that in [41], the proposed system employs a criterion such that both filters can be simultaneously adjusted by a constrained LMS algorithm. However, the acquisition process is more involved than that in [41]. This is because one additional decision has to be made before the code delay can be estimated. For each tested cell, filters are first adapted for a period of time to determine if the code-delay falls into the cell's delay region or not. If it does, the spatial filter will act as an MMSE beamformer and the temporal filter as a code-delay estimator. Thus, the code delay can then be estimated with the peak position of the temporal filter. If not, the next cell is tested and the process is repeated. Note that if the code-delay does not fall into the tested cell's region, the spatial filter will act as a signal blocker with its weights all being zeros.

This property is then used to derive an index for the cell testing. With the choice of the number of cells, we can have an easy tradeoff between performance and complexity. In many cases, however, the complexity reduction is large, but the performance loss is still acceptable.

§ 4.1 Proposed Low-Complexity Code Acquisition

Consider that there are K users in a mobile cell and each user is given an aperiodic PN code sequence with a period much longer than a symbol period. The transmitted signal of the k -th user in baseband can be expressed as

$$x_k(t) = \sum_{j=-\infty}^{\infty} d_k(j) \sum_{l=0}^{U-1} c_{k,j}(l) p(t - lT_c - jUT_c), \quad (4.1)$$

$k = 1, \dots, K$, where $d_k(j)$ is the j -th BPSK symbol of the k -th user, $c_{k,j}(l)$ the l -th chip of the spreading signal for $d_k(j)$, $p(t)$ a unit-amplitude rectangular chip-pulse with a chip-duration T_c , and U the number of chips in a symbol. At the receiver, a uniformly linear array with M sensors is placed and the element spacing is assumed to be half a wave-length. Then, the chip-rate sampled received signal vector in baseband can be represented as

$$\mathbf{r}(n) = \sum_{k=1}^K \mathbf{a}_k \alpha_k x_k(n - \tau_k) \exp(-i\theta_k) + \boldsymbol{\eta}(n), \quad (4.2)$$

where code-delays τ_k , $k = 1, \dots, K$ are assumed to be integers between $[0, U)$, and $\boldsymbol{\eta}(n)$ is an $M \times 1$, complex, and zero-mean Gaussian noise vector with a covariance matrix $\sigma_\eta^2 \mathbf{I}$. Also, $\mathbf{a}_k, \alpha_k, \theta_k$ stand for the steering vector, the amplitude, and the carrier-phase offset, associated with the k -th user, respectively. Note that θ_k is uniformly distributed over $[-\pi, \pi)$ and \mathbf{a}_k is given by $\mathbf{a}_k = [1, \exp(-i\pi \sin \phi_k), \dots, \exp(-i\pi(M-1) \sin \phi_k)]^T$, where ϕ_k denotes the DoA of the k -th user's signal. Without loss of generality, the first user is seen as the desired user. We also assume that $d_1(j) = 1$ during the acquisition period.

As described, the whole delay uncertainty U is divided into cells. Let $Q = \lceil U/M_t \rceil$, where M_t denotes the filter-length of the temporal filter. Among these Q cells, the actual code-delay

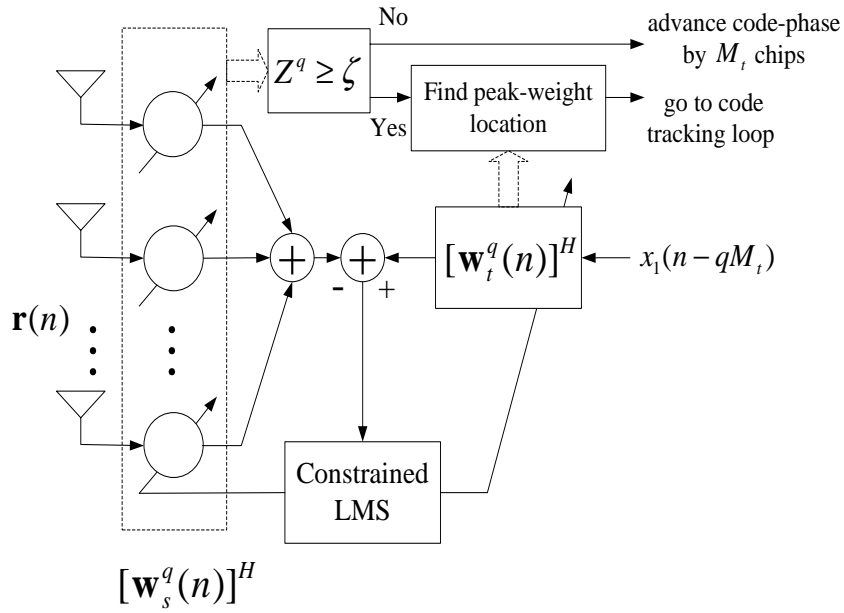


Figure 4.1: System diagram of the proposed system, where $Z^q = \|\mathbf{w}_s^q(N)\|^2$, $q = 0, \dots, Q - 1$.

only falls into the delay region of a certain cell. Let the cell whose delay region includes the desired code delay be the inphase cell and others be outphase cells. Thus, we have one inphase cell and $Q - 1$ outphase cells.

Fig. 4.1 illustrates the block diagram of the proposed system. As seen, the spatial filter \mathbf{w}_s^q combines M array outputs into a single output, where $q = 0, \dots, Q - 1$ denotes the cell index. The temporal filter \mathbf{w}_t^q uses $x_1(n - qM_t)$ as its input signal and the spatial filter output as its reference signal, where $x_1(n)$ is the desired user's PN sequence [since $d_1(j) = 1$]. As far as an inphase cell is concerned, the system is the same as that in [57]. From Fig. 4.1, we can see that the spatial filter can act like a beamformer to reject interference, while the temporal filter a code-delay estimator. In other words, the optimum temporal filter will have a unique peak-weight whose location corresponds to the code-delay [57]. However, for the outphase cells, there is no correlation between the input and the reference signals. The optimum spatial filter will become a signal blocker (all weights are zeros). Using the characteristic, we propose to perform cell detection with the magnitude of the spatial filter-weights. If $\|\mathbf{w}_s^{q'}\|^2$ exceeds a

preset threshold, then the q' -th cell is considered as the inphase cell. Once the inphase cell is identified, the peak-weight in $\mathbf{w}_t^{q'}$ can be located. Let the peak-weight be $w_{t,\hat{\Delta}}^{q'}$, where $w_{t,\hat{\Delta}}^{q'}$ with $0 \leq \hat{\Delta} < M_t$ denotes the $(\hat{\Delta} + 1)$ -th element of $\mathbf{w}_t^{q'}$. Then, the code-delay can be estimated with $\hat{\tau}_1 = q' M_t + \hat{\Delta}$.

As shown, the difference between these two-filter outputs forms the error signal from which we can perform minimization. The cost function to minimize is the same as that in [57]. For each cell, we let

$$\bar{J}^q = E \{ |[\mathbf{w}_t^q]^H \mathbf{x}^q(n) - [\mathbf{w}_s^q]^H \mathbf{r}(n)|^2 \}, \quad (4.3)$$

$q = 0, \dots, Q - 1$, where

$$\mathbf{w}_s^q \triangleq [w_{s,0}^q, \dots, w_{s,M-1}^q]^T, \quad (4.4)$$

$$\mathbf{w}_t^q \triangleq [w_{t,0}^q, \dots, w_{t,M_t-1}^q]^T, \quad (4.5)$$

$$\mathbf{x}^q(n) \triangleq [x_1(n - qM_t), x_1(n - qM_t + 1), \dots, x_1(n - qM_t - M_t + 1)]^T. \quad (4.6)$$

From (4.3), it is simple to observe that a minimum \bar{J}^q (which is zero) occurs at $\mathbf{w}_t^q = \mathbf{0}$ and $\mathbf{w}_s^q = \mathbf{0}$, and this is an undesired trivial solution. To avoid that, we have to make a constraint on the solution. Here, we pose a unit-norm constraint, i.e.,

$$\|\mathbf{w}_t^q\|^2 \triangleq [\mathbf{w}_t^q]^H \mathbf{w}_t^q = 1, \quad q = 0, \dots, Q - 1. \quad (4.7)$$

Thus, minimization of (4.3) turns out to be a constrained optimization problem. We use the Lagrange multiplier method [34] to transform the constrained optimization problem into an unconstrained one. From (4.3) and (4.7), we have an equivalent cost function as

$$\begin{aligned} J^q &= E \{ |[\mathbf{w}_t^q]^H \mathbf{x}^q(n) - [\mathbf{w}_s^q]^H \mathbf{r}(n)|^2 \} + \xi^q \{ 1 - [\mathbf{w}_t^q]^H \mathbf{w}_t^q \} \\ &= [\mathbf{w}_t^q]^H \mathbf{R}_x \mathbf{w}_t^q + [\mathbf{w}_t^q]^H \mathbf{K}^q \mathbf{w}_s^q + [\mathbf{w}_s^q]^H [\mathbf{K}^q]^H \mathbf{w}_t^q \\ &\quad + [\mathbf{w}_s^q]^H \mathbf{R}_r \mathbf{w}_s^q + \xi^q \{ 1 - [\mathbf{w}_t^q]^H \mathbf{w}_t^q \}, \end{aligned} \quad (4.8)$$

where

$$\mathbf{K}_{(M_t \times M)}^q \triangleq - E \{ \mathbf{x}^q(n) \mathbf{r}^H(n) \}, \quad (4.9)$$

$\mathbf{R}_{r(M \times M)} \triangleq E\{\mathbf{r}(n)\mathbf{r}^H(n)\}$, $\mathbf{R}_{x(M_t \times M_t)}^q \triangleq E\{\mathbf{x}^q(n)[\mathbf{x}^q(n)]^H\}$, and ξ^q denotes the Lagrange multiplier for the q -th cell. Differentiating (4.8) with respect to $[\mathbf{w}_s^q]^*$ and $[\mathbf{w}_t^q]^*$ and setting the results to be zero-vectors, we obtain

$$\frac{\partial J^q}{\partial [\mathbf{w}_s^q]^*} = [\mathbf{K}^q]^H \mathbf{w}_t^q + \mathbf{R}_r \mathbf{w}_s^q = \mathbf{0} \quad (4.10)$$

$$\frac{\partial J^q}{\partial [\mathbf{w}_t^q]^*} = \mathbf{R}_x^q \mathbf{w}_t^q + \mathbf{K}^q \mathbf{w}_s^q - \xi^q \mathbf{w}_t^q = \mathbf{0}. \quad (4.11)$$

Since \mathbf{R}_r is a full rank matrix, its matrix inversion exists. From (4.10), we have

$$\mathbf{w}_s^q = -\mathbf{R}_r^{-1}[\mathbf{K}^q]^H \mathbf{w}_t^q. \quad (4.12)$$

Substituting (4.12) into (4.11), we have

$$\{\mathbf{R}_x^q - \mathbf{K}^q \mathbf{R}_r^{-1}[\mathbf{K}^q]^H\} \mathbf{w}_t^q - \xi^q \mathbf{w}_t^q = \mathbf{0}. \quad (4.13)$$

It is simple to observe that the solution of ξ^q in (4.13) denotes the eigenvalue of $\mathbf{R}_x^q - \mathbf{K}^q \mathbf{R}_r^{-1}[\mathbf{K}^q]^H$, while \mathbf{w}_t^q is the corresponding eigenvector. Note that an eigenvector \mathbf{w}_t^q satisfies (4.7) automatically. Once \mathbf{w}_t^q is derived, \mathbf{w}_s^q can be found using (4.12). Multiplying (4.13) with $[\mathbf{w}_t^q]^H$, we obtain

$$\xi^q = [\mathbf{w}_t^q]^H \{\mathbf{R}_x^q - \mathbf{K}^q \mathbf{R}_r^{-1}[\mathbf{K}^q]^H\} \mathbf{w}_t^q. \quad (4.14)$$

Substituting (4.12) into (4.8) and using (4.14), we have

$$\begin{aligned} J^q &= [\mathbf{w}_t^q]^H \mathbf{R}_x^q \mathbf{w}_t^q - [\mathbf{w}_t^q]^H \{\mathbf{K}^q \mathbf{R}_r^{-1}[\mathbf{K}^q]^H\} \mathbf{w}_t^q \\ &\quad - [\mathbf{w}_t^q]^H \{\mathbf{K}^q \mathbf{R}_r^{-1}[\mathbf{K}^q]^H\} \mathbf{w}_t^q + [\mathbf{w}_t^q]^H \{\mathbf{K}^q \mathbf{R}_r^{-1}[\mathbf{K}^q]^H\} \mathbf{w}_t^q \\ &= [\mathbf{w}_t^q]^H \{\mathbf{R}_x^q - \mathbf{K}^q \mathbf{R}_r^{-1}[\mathbf{K}^q]^H\} \mathbf{w}_t^q \\ &= \xi^q, \end{aligned} \quad (4.15)$$

which is identical to (4.14) exactly. Let solutions to (4.10)–(4.11), which are optimum weights, be $\mathbf{w}_{s,o}^q$ and $\mathbf{w}_{t,o}^q$, and the corresponding minimum value of (4.15) be J_{min}^q . We then conclude

that J_{min}^q is equal to the minimum eigenvalue $\mathbf{R}_x^q - \mathbf{K}^q \mathbf{R}_r^{-1} [\mathbf{K}^q]^H$ and $\mathbf{w}_{t,o}^q$ is the corresponding eigenvector. Substituting $\mathbf{w}_{t,o}^q$ into (4.12), we can then obtain $\mathbf{w}_{s,o}^q$.

To simplify notations, we rewrite (4.2) as

$$\mathbf{r}(n) = \sum_{k=1}^K \mathbf{a}_k \alpha_k x_k(n - \tau_k) \exp(-i\theta_k) + \boldsymbol{\eta}(n) \quad (4.16)$$

$$\begin{aligned} &= \mathbf{a} \cdot \exp(-i\theta) \cdot x(n - \nu M_t - \Delta) \\ &+ \sum_{k=2}^K \mathbf{a}_k \alpha_k x_k(n - \tau_k) \exp(-i\theta_k) + \boldsymbol{\eta}(n), \end{aligned} \quad (4.17)$$

where we let $\mathbf{a}_1 = \mathbf{a}$, $\alpha_1 = 1$, $\theta_1 = \theta$, $x_1(n) = x(n)$, and $\tau_1 = \nu M_t + \Delta$. It is simple to see that the inphase cell is the cell that $q = \nu$.

We first consider the scenario of the inphase cell. As mentioned, the proposed system is just the same as that in [57]. From [57], we can have

$$\xi_{min}^\nu = J_{min}^\nu = 1 - \mathbf{a}^H \mathbf{R}_r^{-1} \mathbf{a} \quad (4.18)$$

$$\mathbf{w}_{t,o}^\nu = \underbrace{[0, \dots, 0, 1, 0, \dots, 0]^T}_{\Delta} \exp(i\psi) \quad (4.19)$$

$$\mathbf{w}_{s,o}^\nu = \mathbf{R}_r^{-1} \mathbf{a} \exp(-i[\theta - \psi]), \quad (4.20)$$

where ψ is an arbitrary angle. From (4.19) and (4.20), we can see that both filters do not have unique solutions. This is not surprising since we only pose the magnitude constraint. Also, note that $\mathbf{w}_{s,o}^\nu$ is just the conventional MMSE beamformer ($\mathbf{R}_r^{-1} \mathbf{a}$).

Now, let us consider the rest $Q - 1$ outphase cells. Since $q \neq \nu$, we have $\mathbf{K}^q = \mathbf{0}$ [see (4.9)]. Then, (4.8) becomes

$$J^q = [\mathbf{w}_s^q]^H \mathbf{R}_r \mathbf{w}_s^q + [\mathbf{w}_t^q]^H \mathbf{R}_x^q \mathbf{w}_t^q + \xi^q \{1 - [\mathbf{w}_t^q]^H \mathbf{w}_t^q\}, \quad (4.21)$$

$q \neq \nu$, where $\mathbf{R}_x^q = \mathbf{I}$ (the long-code assumption). Also, (4.10)–(4.11) become

$$\frac{\partial J^q}{\partial [\mathbf{w}_s^q]^*} = \mathbf{R}_r \mathbf{w}_s^q = \mathbf{0} \quad (4.22)$$

$$\frac{\partial J^q}{\partial [\mathbf{w}_t^q]^*} = \mathbf{w}_t^q - \xi^q \mathbf{w}_t^q = \mathbf{0}. \quad (4.23)$$

From (4.22), we have $\mathbf{w}_{s,o}^q = \mathbf{0}$, since \mathbf{R}_r is a full-rank matrix. The spatial filter will block all signal from entering the temporal filter. From (4.23), we can see $\xi_{min}^q = J_{min}^q = 1$, and there is no unique solution for $\mathbf{w}_{t,o}^q$ either. Any vector satisfies the unit-norm constraint can serve as an optimum solution.

§ 4.2 Adaptive Implementation and Convergence Analysis

In lase section, we have proposed a low-complexity code acquisition system modifying the system in [57]. Optimum-weights of the filters are derived with the eigen-decomposition technique. However, the required computational complexity of eigen-decomposition is on the order of $\mathcal{O}(M_t^3)$. In addition, the matrix inversion of \mathbf{R}_r is required in (4.13). To alleviate these problems, we propose to use an adaptive algorithm to derive the optimum filter-weights. The adaptive algorithm we consider is the LMS algorithm which is well-known for its simplicity and robustness. As shown, we have a unit-norm constraint on the temporal filter. Applying this constraint, we then obtain a constrained LMS algorithm. In what follows, we will describe the algorithm and examine related issues such as the step size bound and steady-state MSE. Besides, we also analyze the output SINR of beamformer [see $\gamma(n)$ in Fig. 4.1] for an inphase cell.

§ 4.2.1 Constrained LMS and Convergence Issue

Rewriting (4.3), we have

$$\bar{J}^q(n) = [\mathbf{w}_v^q(n)]^H \mathbf{R}_v^q \mathbf{w}_v^q(n) \quad q = 0, \dots, Q - 1, \quad (4.24)$$

where

$$\mathbf{w}_v^q(n) \triangleq [[\mathbf{w}_t^q(n)]^T, [\mathbf{w}_s^q(n)]^T]^T, \quad (4.25)$$

$$\mathbf{v}^q(n) \triangleq [[\mathbf{x}^q(n)]^T, -\mathbf{r}^T(n)]^T, \quad (4.26)$$

$$\mathbf{R}_v^q \triangleq E\{\mathbf{v}^q(n)[\mathbf{v}^q(n)]^H\}. \quad (4.27)$$

The gradient of (4.24) is by

$$\frac{\partial \bar{J}^q(n)}{\partial [\mathbf{w}_v^q(n)]^*} = \mathbf{R}_v^q \mathbf{w}_v^q(n). \quad (4.28)$$

Using (4.28), we can apply a gradient decent algorithm to obtain the optimum solution, denoted as $\mathbf{w}_{v,o}^q$. However, \mathbf{R}_v^q needs to be estimated. The simplest estimate of \mathbf{R}_v^q is to use instantaneous value from $\mathbf{v}^q(n)[\mathbf{v}^q(n)]^H$ and this yields a stochastic gradient decent algorithm, called the LMS algorithm. We then can have the filter adaptation as

$$\mathbf{w}_v^q(n+1) = \mathbf{w}_v^q(n) + \mu \left\{ -\mathbf{v}^q(n)[\mathbf{v}^q(n)]^H \mathbf{w}_v^q(n) \right\}, \quad (4.29)$$

where μ is the step size controlling the convergence rate. Recall that we have the constraint $\|\mathbf{w}_t^q(n)\| = 1$. This constraint can be easily satisfied if normalization is performed on $\mathbf{w}_t^q(n)$ at every iteration. The overall adaptation procedure is given as

$$e^q(n) = [\mathbf{w}_v^q(n)]^H \mathbf{v}^q(n) \quad (4.30)$$

$$\mathbf{H}^q(n) = \text{diag} \left\{ \underbrace{\frac{1}{\|\mathbf{w}_t^q(n)\|}, \dots, \frac{1}{\|\mathbf{w}_t^q(n)\|}}_{M_t}, \underbrace{1, \dots, 1}_M \right\} \quad (4.31)$$

$$\mathbf{w}_v^q(n+1) = \mathbf{H}^q(n) \mathbf{w}_v^q(n) - \mu \mathbf{v}^q(n) [e^q(n)]^*, \quad (4.32)$$

$n = 0, 1, \dots, N-1; q = 0, \dots, Q-1$, where $\text{diag}\{\cdot\}$ denotes a diagonal matrix consisting of the arguments that it includes, and N the iteration number for each cell. As we can see, $\mathbf{H}^q(n)$ normalizes $\mathbf{w}_t^q(n)$ at every iteration. After training, we have to detect the inphase cell first. To do that, we propose to compare $\|\mathbf{w}_s^q(N)\|^2$ with a preset threshold. If $\|\mathbf{w}_s^q(N)\|^2$ is larger than the threshold, the cell is deemed as the inphase cell. Then, the peak location of $\mathbf{w}_t^q(N)$ is located and the code delay is estimated. Otherwise, we go to the next cell and start the process all over again. To guarantee convergence, μ has to be selected properly. Here, we perform the mean convergence analysis to derive a step size bound. Subtracting $\mathbf{w}_{v,o}^q = [[\mathbf{w}_{t,o}^q]^T, [\mathbf{w}_{s,o}^q]^T]^T$

from both sides of (4.32), we have

$$\begin{aligned}
\Delta \mathbf{w}_v^q(n+1) &= \Delta \mathbf{w}_v^q(n) + [\mathbf{H}^q(n) - \mathbf{I}] \mathbf{w}_v^q(n) \\
&\quad - \mu \mathbf{v}^q(n) \{[\mathbf{w}_v^q(n)]^H \mathbf{v}^q(n)\}^* \\
&= \Delta \mathbf{w}_v^q(n) + [\mathbf{H}^q(n) - \mathbf{I}] \mathbf{w}_v^q(n) \\
&\quad - \mu \mathbf{v}^q(n) [\mathbf{v}^q(n)]^H [\Delta \mathbf{w}_v^q(n) + \mathbf{w}_{v,o}^q] \\
&= \{\mathbf{I} - \mu \mathbf{v}^q(n) [\mathbf{v}^q(n)]^H\} \Delta \mathbf{w}_v^q(n) \\
&\quad + [\mathbf{H}^q(n) - \mathbf{I}] \mathbf{w}_v^q(n) - \mu \mathbf{v}^q(n) [e_o^q(n)]^*, \tag{4.33}
\end{aligned}$$

where $e_o^q(n) \triangleq [\mathbf{w}_{v,o}^q]^H \mathbf{v}^q(n)$ and $\Delta \mathbf{w}_v^q(n) \triangleq \mathbf{w}_v^q(n) - \mathbf{w}_{v,o}^q$. Taking the statistical expectation of (4.33), applying the direct-averaging method [34], and using the orthogonality principle, we then have

$$\begin{aligned}
E\{\Delta \mathbf{w}_v^q(n+1)\} &= [\mathbf{I} - \mu \mathbf{R}_v^q] E\{\Delta \mathbf{w}_v^q(n)\} \\
&\quad + [E\{\mathbf{H}^q(n)\} - \mathbf{I}] E\{\mathbf{w}_v^q(n)\}. \tag{4.34}
\end{aligned}$$

Let $\mathbf{\Lambda}^q = \text{diag}\{\lambda_{v,1}^q, \dots, \lambda_{v,M_t+M}^q\}$ with $\lambda_{v,j}^q$ being an eigenvalue of \mathbf{R}_v^q , and \mathbf{U}^q be a matrix consisting of the eigenvectors of \mathbf{R}_v^q . Multiplying (4.34) with $[\mathbf{U}^q]^H$ and letting $\mathbf{g}^q(n) = [\mathbf{U}^q]^H E\{\Delta \mathbf{w}_v^q(n)\}$, we obtain

$$\begin{aligned}
\mathbf{g}^q(n+1) &= [\mathbf{I} - \mu \mathbf{\Lambda}^q] \mathbf{g}^q(n) \\
&\quad + [\mathbf{U}^q]^H [E\{\mathbf{H}^q(n)\} - \mathbf{I}] E\{\mathbf{w}_v^q(n)\}. \tag{4.35}
\end{aligned}$$

Since $\mathbf{w}_v^q(n)$ is normalized at every iteration and the step size is usually small, it is reasonable to assume that $\mathbf{H}^q(n) \approx \mathbf{I}$ and the second term in the right-hand side of (4.35) can be ignored. Iterating (4.35), we obtain

$$\mathbf{g}^q(n) = [\mathbf{I} - \mu \mathbf{\Lambda}^q]^n \mathbf{g}^q(0). \tag{4.36}$$

Thus, for (4.35) to converge, the following condition must be satisfied

$$0 < \mu < \frac{2}{\lambda_{v,max}^q}, \tag{4.37}$$

where $\lambda_{v,max}^q$ denotes the maximum eigenvalue of \mathbf{R}_v^q . This result is the same as that of the conventional LMS algorithm. From (4.36), we can also see that $\mathbf{g}^q(\infty) = \mathbf{0}$. In other words, $E\{\mathbf{w}_v^q(n)\} = \mathbf{w}_{v,o}^q$, when $n \rightarrow \infty$.

Note that while the conventional LMS algorithm requires $2(M_t + M)$ multiplications per iteration, the constrained LMS algorithm developed here needs extra M_t multiplications for calculation of $\|\mathbf{w}_t^q(n)\|$ and extra M_t divisions for normalization [see (4.32)].

§ 4.2.2 Steady-state MSE Analysis

We now derive the steady-state MSE of the constrained LMS algorithm. Invoking the direct-averaging method [34] and using (4.33), we can write the correlation matrix of the tap-weight error vector as

$$\begin{aligned} \mathbf{P}^q(n+1) &\triangleq E\{\Delta\mathbf{w}_v^q(n+1)[\Delta\mathbf{w}_v^q(n+1)]^H\} \\ &= [\mathbf{I} - \mu\mathbf{R}_v^q]\mathbf{P}^q(n)[\mathbf{I} - \mu\mathbf{R}_v^q] + \mu^2 J_{min}^q \mathbf{R}_v^q \\ &\quad + E\{[\mathbf{H}^q(n) - \mathbf{I}]\mathbf{w}_v^q(n)[\mathbf{w}_v^q(n)]^H[\mathbf{H}^q(n) - \mathbf{I}]\}. \end{aligned} \quad (4.38)$$

As stated, $\mathbf{w}_t^q(n)$ is normalized at every iteration and the step size is usually small. Thus, $\mathbf{H}^q(n) \approx \mathbf{I}$ and the last term in the right-hand side of (4.38) can be ignored. Let $\bar{\mathbf{P}}^q(n) \triangleq [\mathbf{U}^q]^H \mathbf{P}^q(n) \mathbf{U}^q$ and observe that $[\mathbf{U}^q]^H \mathbf{R}_v^q \mathbf{U}^q = \mathbf{\Lambda}^q$. Pre-multiplying and post-multiplying both sides of (4.38) with $[\mathbf{U}^q]^H$ and \mathbf{U}^q , respectively, we have

$$\bar{\mathbf{P}}^q(n+1) = [\mathbf{I} - \mu\mathbf{\Lambda}^q]\bar{\mathbf{P}}^q(n)[\mathbf{I} - \mu\mathbf{\Lambda}^q] + \mu^2 J_{min}^q \mathbf{\Lambda}^q. \quad (4.39)$$

Let the j -th element on the diagonal of $\bar{\mathbf{P}}^q(n)$ be $\bar{p}_j^q(n)$. Then,

$$\bar{p}_j^q(n+1) = (1 - \mu\lambda_{v,j}^q)^2 \bar{p}_j^q(n) + \mu^2 J_{min}^q \lambda_{v,j}^q, \quad (4.40)$$

$j = 1, \dots, M_t + M$. When $n \rightarrow \infty$, $\bar{p}_j^q(n+1) \approx \bar{p}_j^q(n)$. Form (4.40), we derive

$$\bar{p}_j^q(\infty) = \frac{\mu J_{min}^q}{2 - \mu\lambda_{v,j}^q}. \quad (4.41)$$

The additional MSE due to the use of the LMS algorithm is generally referred to as the excess MSE, denoted as $J_{ex}^q(\infty)$. From [34], we then have

$$J_{ex}^q(\infty) = \sum_{j=1}^{M_t+M} \bar{p}_j^q(\infty) \lambda_{v,j}^q = J_{min}^q \sum_{j=1}^{M_t+M} \frac{\mu \lambda_{v,j}^q}{2 - \mu \lambda_{v,j}^q}. \quad (4.42)$$

Denote the steady-state MSE of the LMS adaptation as J_{ss}^q . Finally, we have

$$J_{ss}^q = J_{min}^q + J_{ex}^q(\infty). \quad (4.43)$$

§ 4.2.3 Output SINR at Beamformer for an Inphase Cell

Now, let us analyze the output SINR of the beamformer. We consider the inphase cell ($q = \nu$), and the output is by

$$\gamma(n) \triangleq [\mathbf{w}_s^\nu(n)]^H \mathbf{r}(n) \quad (4.44)$$

$$= [\mathbf{w}_s^\nu(n)]^H (\mathbf{a} \exp(-i\theta) x(n - \nu M_t - \Delta) + \mathbf{z}(n)), \quad (4.45)$$

where $\mathbf{z}(n)$ consists of MAI and noise. Using (4.20), we can find the output SINR of the optimum beamformer, denoted as SINR_o , as

$$\text{SINR}_o = \frac{[\mathbf{w}_{s,o}^\nu]^H \mathbf{R}_A \mathbf{w}_{s,o}^\nu}{[\mathbf{w}_{s,o}^\nu]^H \mathbf{R}_z \mathbf{w}_{s,o}^\nu} \quad (4.46)$$

$$= \frac{\mathbf{a}^H \mathbf{R}_r^{-1} \mathbf{R}_A \mathbf{R}_r^{-1} \mathbf{a}}{\mathbf{a}^H \mathbf{R}_r^{-1} \mathbf{R}_z \mathbf{R}_r^{-1} \mathbf{a}}, \quad (4.47)$$

where $\mathbf{R}_A = \mathbf{a} \mathbf{a}^H$ and $\mathbf{R}_z \triangleq E\{\mathbf{z}(n) \mathbf{z}^H(n)\}$. Since we use adaptive filter-weights to approximate the optimum weights, we have to include the excess MSE in the SINR calculation. Thus, we can rewrite (4.47) as

$$\text{SINR}_o = \frac{\mathbf{a}^H \mathbf{R}_r^{-1} \mathbf{R}_A \mathbf{R}_r^{-1} \mathbf{a}}{\mathbf{a}^H \mathbf{R}_r^{-1} \mathbf{R}_z \mathbf{R}_r^{-1} \mathbf{a} + J_{ex}^\nu(\infty)}, \quad (4.48)$$

where $J_{ex}^\nu(\infty)$ is from (4.42).

§ 4.3 Performance Analysis

The performance of acquisition is generally measured with the mean acquisition time, which is the averaged time for correct acquisition. The mean acquisition time of the proposed system is a function of the probability of false alarm, the probability of missing (denoted as P_M), and the probability of correct acquisition (denoted as P_D). In this section, we will first derive these probabilities and then calculate the mean acquisition time.

§ 4.3.1 Mean Acquisition Time

As mentioned, the proposed scheme performs sequential cell testing. Since there are Q possible cells, there are Q possible states in the system. Label these states as $\{s_0, \dots, s_{Q-1}\}$ in the circular state diagram [1], as shown in Fig. 4.2. In the figure, the state labeled as ACQ indicates the state of correct acquisition. That labeled as FA is the state of false alarm. Using this diagram, we can evaluate the averaged time reaching the ACQ state, i.e., the mean acquisition time. Without loss of generality, we assume s_{Q-1} being the state of an inphase cell, and thus it is connected to the ACQ state. Also, let $Z^q \triangleq \|\mathbf{w}_s^q(N)\|^2$. As described in Section 4.1, the optimum $\mathbf{w}_{s,o}^q$, for outphase cells are all-zero vectors, and the corresponding Z^q should be small. On the other hand, Z^q of the inphase cell should be large. Using this property, we set a threshold ζ for the detection of the inphase cell. Thus, the acquisition problem can be seen as a hypothesis testing problem. Note that correct acquisition means that the inphase cell is correctly detected and at the same time the optimum peak-weight location (Δ) is also correctly estimated. There are two types of false alarm. We name the false alarm occurring in an inphase cell as an inphase false alarm, which means that the inphase cell is correctly detected but the peak location is not (i.e., $\hat{\Delta} \neq \Delta$), and the false alarm occurring in an outphase cell as an outphase false alarm.

From Fig. 4.2, we can see that the transfer function (TF) between s_{Q-1} and ACQ can be expressed as $H_b(z) = P_D z^{N+1}$ [1], [21], [22], where $N + 1$ denotes the time for iteration and cell detection, and z the unit-delay operator. The probability of missing, P_M , is defined as

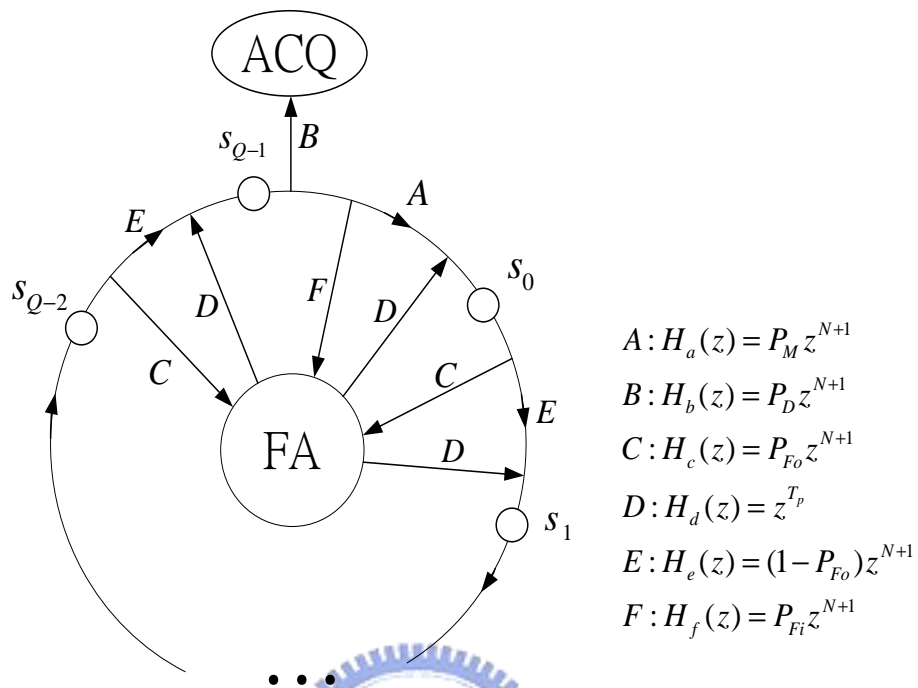


Figure 4.2: Circular state diagram of the proposed system.

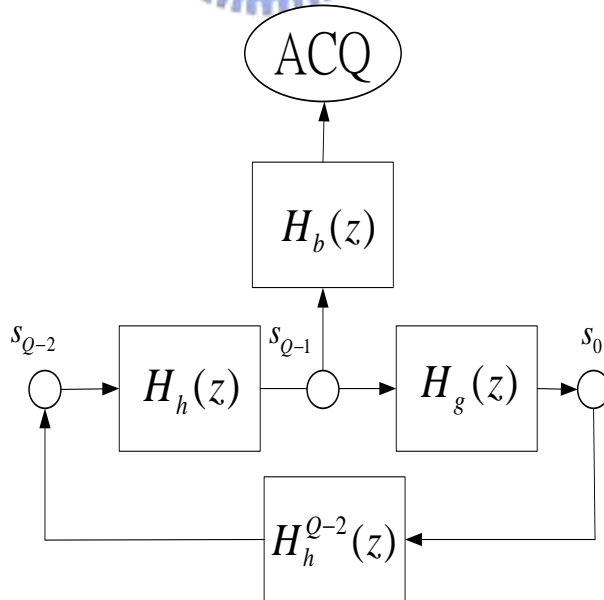


Figure 4.3: Simplified state diagram.

the probability of $Z^{Q-1} < \zeta$. Thus, if only the missing is considered, the TF between s_{Q-1} and s_0 can be expressed as $H_a(\mathbf{z}) = P_M \mathbf{z}^{N+1}$. The probability of the inphase false alarm, denoted as P_{Fi} , is equal to $P_{Fi} = 1 - P_D - P_M$. On the other hand, if only the inphase false alarm is considered, the TF between s_{Q-1} and FA can be expressed $H_f(\mathbf{z}) = P_{Fi} \mathbf{z}^{N+1}$. Note the system has to stay T_p chips once it enters the FA state. The quantity T_p is generally referred to as the penalty time [1]. The TF between the input and the output of the FA state can be described as $H_d(\mathbf{z}) = \mathbf{z}^{T_p}$. Thus, we can have the TF between s_{Q-1} and s_0 as $H_g(\mathbf{z}) \triangleq H_a(\mathbf{z}) + H_f(\mathbf{z})H_d(\mathbf{z})$. The TF between any s_q , $q = 0, 1, \dots, Q-2$, and the FA state will be $H_c(\mathbf{z}) = P_{Fo} \mathbf{z}^{N+1}$, where P_{Fo} is the probability of outphase false alarm. If the outphase false alarm between two consecutive states, s_q and s_{q+1} , $q = 0, 1, \dots, Q-2$, is not considered, then the TF between these two consecutive states can be described as $H_e(\mathbf{z}) = (1 - P_{Fo}) \mathbf{z}^{N+1}$. Thus, we can have the TF between any two consecutive states, s_q and s_{q+1} , $q = 0, 1, \dots, Q-2$, as $H_h(\mathbf{z}) \triangleq H_e(\mathbf{z}) + H_c(\mathbf{z})H_d(\mathbf{z})$.

Using the TFs derived above, we now can redraw the diagram in Fig. 4.2 as that in Fig. 4.3. In what follows, we use Fig. 4.3 to calculate the mean acquisition time. We define the probability of correct acquisition starting from time zero and ending at time n as $P_{ACQ}(n)$. Then, its z-transform is given by

$$P_{ACQ}(\mathbf{z}) = \sum_{n=0}^{\infty} P_{ACQ}(n) \mathbf{z}^n, \quad (4.49)$$

which can be the generating function of acquisition time. Denote the mean acquisition time as T_{acq} and it can be derived from [1]

$$T_{acq} \triangleq \frac{d}{d\mathbf{z}} P_{ACQ}(\mathbf{z}) \Big|_{\mathbf{z}=1}. \quad (4.50)$$

Note that the unit of (4.50) is chip. Assuming that we can start searching at any state in

$\{s_0, \dots, s_{Q-1}\}$ with equal probability $1/Q$, we rewrite (4.49) as

$$P_{ACQ}(\mathbf{z}) = \frac{1}{Q} \sum_{q=0}^{Q-1} P_{q,ACQ}(\mathbf{z}) \quad (4.51)$$

$$= \frac{1}{Q} H_b(\mathbf{z}) \sum_{q=0}^{Q-1} P_{q,Q-1}(\mathbf{z}), \quad (4.52)$$

where $P_{q,ACQ}(\mathbf{z})$ denotes the TF between s_q and ACQ states, and $P_{q,Q-1}(\mathbf{z})$ the TF between s_q and s_{Q-1} . Using Fig. 4.3, we can have

$$P_{q,Q-1}(\mathbf{z}) = \frac{H_h^{Q-1-q}(\mathbf{z})}{1 - H_g(\mathbf{z})H_h^{Q-1}(\mathbf{z})}. \quad (4.53)$$

Substituting (4.53) into (4.52), we obtain

$$P_{ACQ}(\mathbf{z}) = \frac{1}{Q} \frac{H_b(\mathbf{z})}{1 - H_g(\mathbf{z})H_h^{Q-1}(\mathbf{z})} \sum_{q=0}^{Q-1} H_h^{Q-1-q}(\mathbf{z}) \quad (4.54)$$

$$= \frac{1}{Q} \frac{H_b(\mathbf{z})[1 - H_h^Q(\mathbf{z})]}{[1 - H_g(\mathbf{z})H_h^{Q-1}(\mathbf{z})][1 - H_h(\mathbf{z})]}. \quad (4.55)$$

Using (4.55) in (4.50), we finally obtain

$$T_{acq} = \frac{1}{P_D} \left\{ [1 + (Q-1) \frac{2 - P_D}{2}] (N+1) + [P_{Fi} + (Q-1)P_{Fo} \frac{2 - P_D}{2}] T_p \right\}. \quad (4.56)$$

Observing (4.56), we find that a large P_{Fo} and a small P_D can enlarge T_{acq} significantly. It should be noted that P_{Fo} is more harmful to T_{acq} than P_{Fi} . This is because there are $Q-1$ outphase cells and only one inphase cell. For an ideal situation that $P_D = 1$ and $P_{Fi} = P_{Fo} = 0$, we have

$$T_{acq, LB} = \frac{Q+1}{2} (N+1). \quad (4.57)$$

which can serve as the lower bound of (4.56).

§ 4.3.2 Probabilities Derivation

Since Z^q is random, we have to characterize its statistical properties. It is mentioned in [33] that when an adaptive filter approaches the steady-state, its weights have a Gaussian distribution. From the analysis in the previous section, we see that in the steady-state ($N \rightarrow \infty$), $\mathbf{w}_s^q(N)$ has a mean vector of $\mathbf{w}_{s,o}^q$ and $\mathbf{w}_t^q(N)$ has a mean vector $\mathbf{w}_{t,o}^q$. We denote their covariance matrices as $\mathbf{C}_s^q \triangleq E\{[\mathbf{w}_s^q(N) - \mathbf{w}_{s,o}^q][\mathbf{w}_s^q(N) - \mathbf{w}_{s,o}^q]^H\}$ and $\mathbf{C}_t^q \triangleq E\{[\mathbf{w}_t^q(N) - \mathbf{w}_{t,o}^q][\mathbf{w}_t^q(N) - \mathbf{w}_{t,o}^q]^H\}$, respectively. As a common practice, the step size is usually small. Thus, we can use the Taylor expansion to expand $1/(2 - \mu\lambda_{v,j}^q)$ in (4.41) with respect to $\mu\lambda_{v,j}^q = 0$. Then, we can derive

$$\frac{1}{2 - \mu\lambda_{v,j}^q} = \frac{1}{2} + \frac{1}{4}\mu\lambda_{v,j}^q + \cdots \quad (4.58)$$

From (4.39), it can be seen that the matrix $\bar{\mathbf{P}}^q(n)$ will become diagonal as $n \rightarrow \infty$. Using this property and truncating the terms higher than the first-order in (4.58), we then have

$$\bar{\mathbf{P}}^q(n) = \frac{\mu}{2}J_{min}^q\mathbf{I} + \frac{\mu^2}{4}J_{min}^q\mathbf{\Lambda}^q. \quad (4.59)$$

Pre-multiplying and post-multiplying both sides of (4.59) with \mathbf{U}^q and $[\mathbf{U}^q]^H$, we obtain

$$\mathbf{P}^q(n) = \frac{\mu}{2}J_{min}^q\mathbf{I} + \frac{\mu^2}{4}J_{min}^q\mathbf{R}_v^q. \quad (4.60)$$

Note that the $M_t \times M_t$ upper-left submatrix of $\mathbf{P}^q(n)$ corresponds to \mathbf{C}_t^q and the $M \times M$ lower-right submatrix of that can be \mathbf{C}_s^q . Thus, we can write

$$\mathbf{C}_s^q = J_{min}^q \left[\frac{\mu}{2}\mathbf{I} + \frac{\mu^2}{4}\mathbf{R}_r \right] \approx \frac{\mu J_{min}^q}{2}\mathbf{I} \quad (4.61)$$

$$\mathbf{C}_t^q = J_{min}^q \left[\frac{\mu}{2} + \frac{\mu^2}{4} \right] \mathbf{I} \approx \frac{\mu J_{min}^q}{2}\mathbf{I}, \quad (4.62)$$

where J_{min}^q is the MMSE evaluated in Section 4.1. For notational clarity, we let $\sigma_1^2 \triangleq \mu J_{min}^q/2$ and $\sigma_0^2 \triangleq \mu J_{min}^q/2$, $q \neq \nu$. From (4.61)–(4.62), we can see that these filter weights are approximately i.i.d.

Let us calculate P_D now. Since $\mathbf{w}_{s,o}^q = \mathbf{0}$ for $q \neq \nu$, we find that Z^q , $q \neq \nu$ is chi-square distributed with M degrees of freedom, while Z^ν is noncentral chi-square distributed with M degrees of freedom. Thus, the probability of P_{Fo} is given by

$$P_{Fo} = \int_{\zeta}^{\infty} \frac{1}{\sigma_0^M 2^{M/2} \Gamma(M/2)} \beta^{M/2-1} \exp\left(-\frac{\beta}{2\sigma_0^2}\right) d\beta, \quad (4.63)$$

where $\Gamma(\cdot)$ stands for the gamma function [35], and ζ is usually selected on some level to prevent a large P_{Fo} (e.g. $P_{Fo} = 0.01$). Let P_C be the probability of correct inphase cell detection. Then,

$$P_C \triangleq \int_{\zeta}^{\infty} \frac{1}{2\sigma_1^2} \left(\frac{\beta}{s^2}\right)^{(M-2)/4} \exp\left(-\frac{s^2 + \beta}{2\sigma_1^2}\right) I_{M/2-1}\left(\sqrt{\beta} \frac{s}{\sigma_1}\right) d\beta, \quad (4.64)$$

where $s^2 \triangleq \|\mathbf{w}_{s,o}^\nu\|^2$ and $I_{M/2-1}(\cdot)$ the $(M/2 - 1)$ -th order modified Bessel function of the first kind [35]. Next, we evaluate the probability of $\hat{\Delta} = \Delta$, say P_Δ . Let $Y_j = |w_{t,j}^\nu(N)|^2$ for $j = 0, \dots, M_t - 1$. When N is large enough, Y_Δ has a noncentral chi-square distribution with two degrees of freedom, whereas Y_j , $j \neq \Delta$, has a chi-square distribution. The corresponding probability density functions can be shown as

$$p_{Y_\Delta}(y) = \frac{1}{2\sigma_1^2} \exp\left(-\frac{|w_{t,o,\Delta}^\nu|^2 + y}{2\sigma_1^2}\right) I_0\left(\sqrt{y} \frac{|w_{t,o,\Delta}^\nu|}{\sigma_1}\right) \quad (4.65)$$

$$p_{Y_j}(y) = \frac{1}{2\sigma_1^2} \exp\left(-\frac{y}{2\sigma_1^2}\right) \quad y > 0, j \neq \Delta, \quad (4.66)$$

where $w_{t,o,\Delta}^\nu$ denotes the $(\Delta + 1)$ -th element in $\mathbf{w}_{t,o}^\nu$ with $|w_{t,o,\Delta}^\nu|^2 = 1$. With (4.65)–(4.66), P_Δ is given by

$$P_\Delta = \Pr(Y_j < Y_\Delta) \quad (4.67)$$

$$= \int_0^\infty \left[\int_0^\beta p_{Y_j}(\beta') d\beta' \right]^{M_t-1} p_{Y_\Delta}(\beta) d\beta, \quad j \neq \Delta \quad (4.68)$$

where the i.i.d. property has been applied in (4.68). Finally, we can have $P_D = P_C P_\Delta$, $P_M = 1 - P_C$, $P_{Fi} = 1 - P_D - P_M$. Then, (4.56) can be evaluated.

§ 4.4 Simulations

To demonstrate the effectiveness of the proposed system, we report some simulation results in this section. First, we set common parameters used in simulations as follows: $U = 256$, $M = 8$, $K = 8$, $T_p = 100U$ chips, $\sigma_\eta^2 = 1$, $\mu = 3 \times 10^{-3}$, $\mathbf{w}_s^q(0) = \mathbf{0}$, and $\mathbf{w}_t^q(0) = (1/\sqrt{M_t})[1, \dots, 1]^T$ for $q = 0, \dots, Q - 1$. Also, for convenience, the DoAs are fixed to be $\{\phi_k\}_{k=1}^K = \{0.41, 0.56, 0.78, -0.20, -0.52, -1.12, 1.12, 0.94\}$ (radians) in all simulations.

In the first set of simulations, we examine the convergence behaviors of the the proposed adaptive system. This includes the MSE convergence of the system, the SINR convergence behavior of the spatial filter, and the weight convergence of the temporal filter. All experimental results are derived from an average of 400 trials. In these experiments, we let $M_t = 8$, the array input SINR be -10 dB ($\alpha = 1$), and the powers of jammers be equal. Fig. 4.4 shows the MSE convergence curve for the proposed system with the inphase cell ($q = \nu$). It can be seen that the steady-state MSE value approaches to the theoretical value 0.28 around $n = 1300$. The theoretical MSE value is calculated from (4.43). In the same figure, we also show the MSE with an outphase cell ($q \neq \nu$). It is apparent that the experimental MSE is more fluctuating. This is because that $J_{min}^q, q \neq \nu$, is much greater than J_{min}^ν making the corresponding excess MSE larger. Fig. 4.5 illustrates the SINR convergence curve for the beamformer output [see $\gamma(n)$ in (4.44)]. The SINR starts from -11 dB and eventually reaches the optimum value 4.18 dB. The theoretical value is derived from (4.48) and is shown with the horizontal line in the figure. We omit the results for $q \neq \nu$, in which the experimental SINR is around -10 dB. This indicates that the spatial filter can not suppress interference for outphase cells. From the figure, we conclude that the adaptive spatial filter can effectively suppress interference when it operates in the inphase cell. Fig. 4.6 presents several experimental beam-patterns calculated from $\mathbf{w}_s^\nu(N)$. Here, we let $N = 2000$. The optimum beam-pattern, derived from (4.20), is also shown. Note that the arrow signs indicate the signal DoAs and only the DoA of the desired user, ϕ_1 , is labeled. As seen, the spatial filter, acting as a beamformer, can steer the main-beam

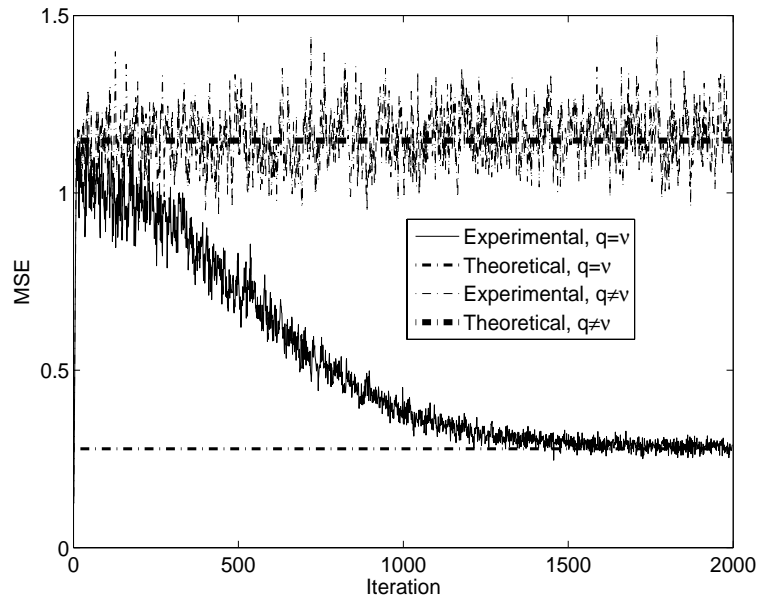


Figure 4.4: Convergence curve for MSE when $\mu = 3 \times 10^{-3}$ and $M_t = 8$. Theoretical value is from (4.43).

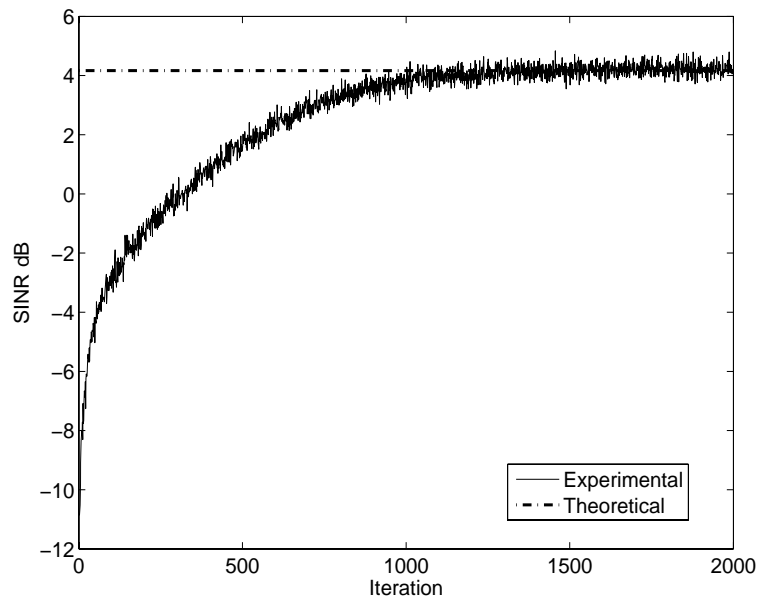


Figure 4.5: Convergence curve for SINR of $\gamma(n)$ at an inphase cell, $\mu = 3 \times 10^{-3}$, and $M_t = 8$. Theoretical value is from (4.48).

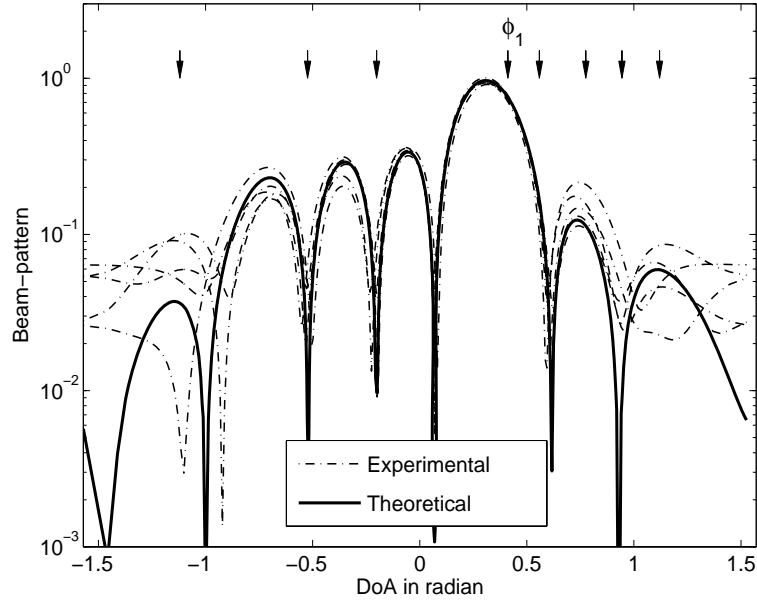


Figure 4.6: Experimental and theoretical beam-patterns for an inphase cell ($N = 2000$, $M_t = 8$, and $M = 8$). Arrow signs indicate the DoAs associated with all users. The labeled one, ϕ_1 , is the DoA of the desired user.

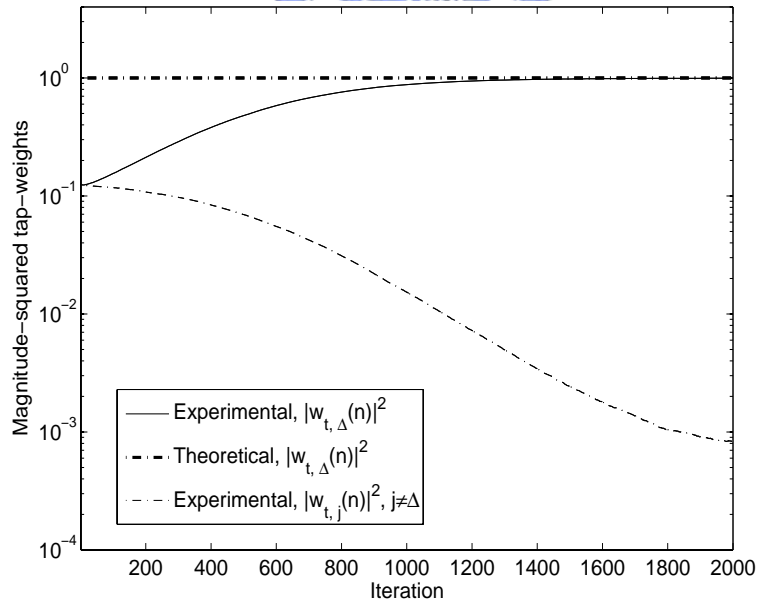


Figure 4.7: Convergence curve for squared temporal filter-weights $|w_{t,j}^v(n)|^2$ for $j = 0, \dots, M_t - 1$ and $M_t = 8$. Theoretical value is obtained from (4.19).

to the incident direction ϕ_1 and put nullities in the directions of interference. The convergence behavior of the temporal filter weights is shown in Fig. 4.7. We can see that the tap-weight whose indices correspond to the code delay, $|w_{i,\Delta}^\nu(n)|^2$, converges to unity, while other weights, $|w_{i,j}^\nu(n)|^2, j \neq \Delta$, converge to a very small value (only one weight is shown in the figure).

In the figures shown above, we can see that the theoretical results calculated using derived formulas all match the simulated ones very well. Then, we calculate the mean acquisition time of the proposed system. Before that, we have to evaluate related probabilities. Fig. 4.8 shows the comparison of experimental and theoretical P_{Fo} (versus ζ).

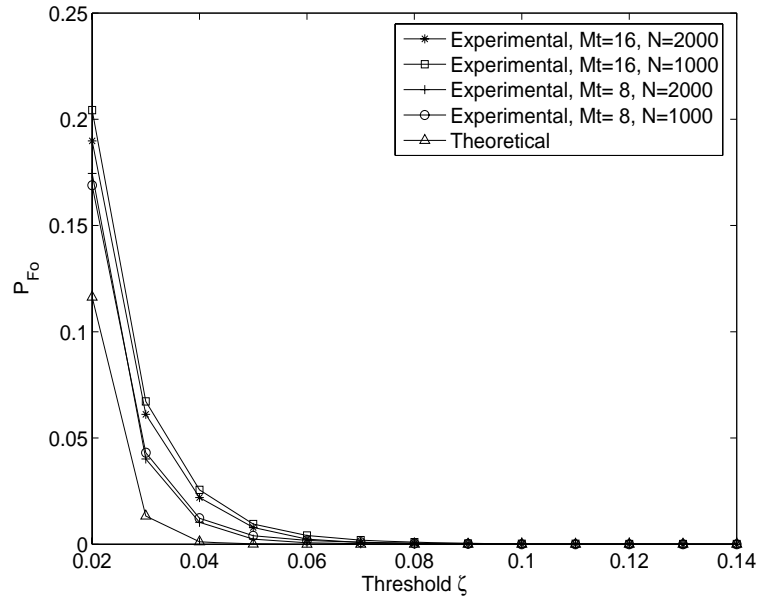


Figure 4.8: Experimental and theoretical probabilities of outphase false alarm P_{Fo} versus threshold ζ .

Here, the array input SINR is set as -10 dB. As we can see, P_{Fo} decreases rapidly as the threshold increases. The experimental results with $M_t = 8$ match the theoretical results [in (4.63)] better than those with $M_t = 16$. We also see that the experimental results for $N = 1000$ and $N = 2000$ are close. Fig. 4.9 shows the similar comparison for P_C .

Here, experiment and theoretical results agree very well for $N = 2000$. However, they agree

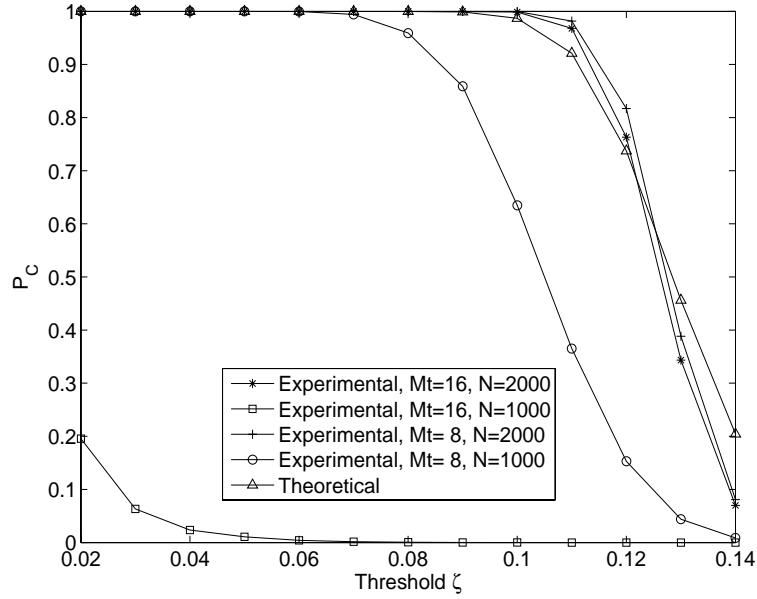


Figure 4.9: Experimental and theoretical probabilities of P_C versus threshold ζ .

poorly for $N = 1000$. This is because the spatial filter has not converged with the given number of iterations, and Z^ν tends to be smaller than the threshold. This behavior is different from that in P_{F_o} calculation. From Figs. 4.8 and 4.9, we can see that the best ζ is around 0.07. Using this value, P_{F_o} can be close to zero and P_C to one. The theoretical value of P_Δ is usually very close to one. With 10^4 trials, we find $P_\Delta = 1$ ($N = 2000$, $M_t = 8$ or $M_t = 16$). As a result, we can let $P_C \approx P_D$. Since the interference is mainly suppressed by the spatial filter, P_{F_i} is close to zero. Substituting derived experimental probabilities into (4.56), we can then calculate the mean acquisition time. Fig. 4.10 shows the result.

In the figure, lower bounds derived from (4.57), are also shown. It is simple to see that if ζ is too small, P_{F_o} will be large, leading to large mean acquisition time. On the contrary, if ζ is too large, P_M being equal to $1 - P_C$ will be large, leading to large mean acquisition time also. From the figure, we can observe that ζ can be chosen in a wide range of value such that mean acquisition times can approach lower bounds.

Finally, we conduct performance comparison for the correlator-based scheme in [39], the

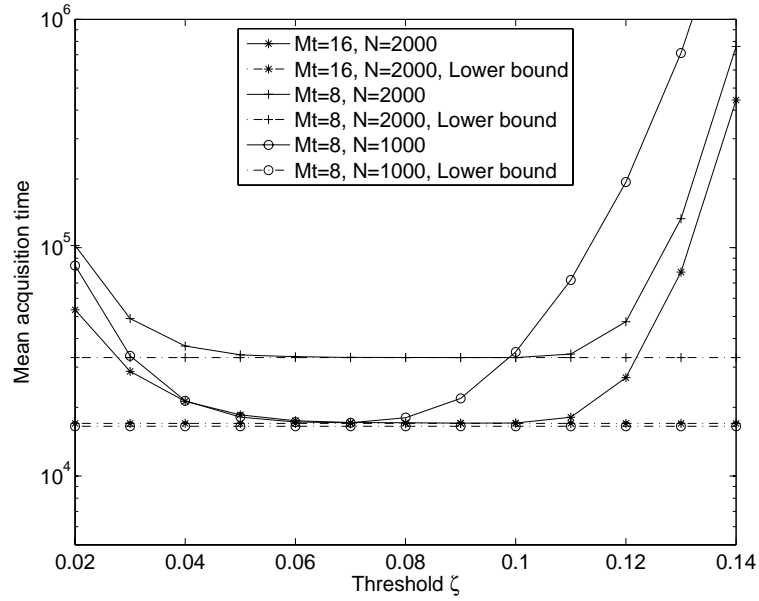


Figure 4.10: Experimental mean acquisition time (chips) versus ζ .

adaptive array system in [57], and the proposed system. We let $U = 256$, $\sigma_\eta^2 = 1$, $\alpha = 1$, and the powers of all jammers be equal. As addressed in [39], the derived theoretical threshold is not accurate enough to guarantee that a designated probability of false alarm (set as 0.01 here) can be achieved. Thus, we experimentally search for the threshold, processing period (for adaptation), and step size that gives the minimal mean acquisition time (for each array input SINR). To ensure a fair comparison, we also search for an optimum set $\{\mu, N, \zeta\}$ that provides the optimum performance for the proposed system ($M_t = 8$ or 16). Here, P_{Fo} is set as 0.01. Similarly, for the system in [57], the performance is optimized over $\{\mu, N\}$. Fig. 4.11 shows the performance comparison for these systems in various SINRs. From the figure, we first can see that the correlator-based system has the worst performance. This is because the beamformer training cannot be accomplished in a short processing period, especially in serious MAI environments. The system in [57] exhibits the best performance. It can outperform the correlator-based system by an order of magnitude. Comparing to that in [57], the proposed system somewhat compromises the performance. However, its computational complexity is

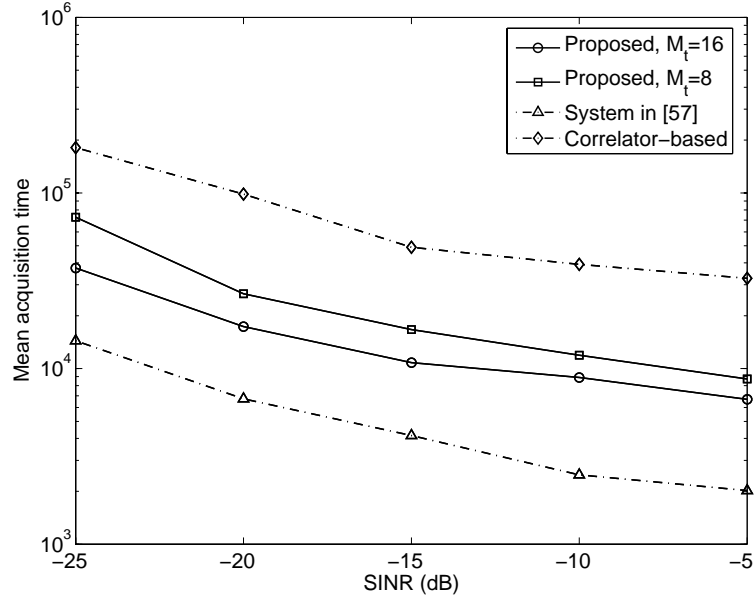


Figure 4.11: Mean acquisition time comparison for $U = 256$.

much lower. For example, with $M_t = 8$, the temporal-filter size is just 1/32 of that in [57]. We also can see that for the proposed system with $M_t = 8$ performs slightly worse than that with $M_t = 16$. We can expect that the larger the M_t , the smaller the performance loss. Thus, we can have an easy tradeoff between performance and computational complexity.

§ 4.5 Conclusions

In this chapter, we proposed a low-complexity adaptive array code acquisition scheme, especially being suited to large-delay channel environments. Applying the serial search technique, we can greatly reduce the temporal filter size, so does the computational complexity. The proposed scheme also allows an easy tradeoff between performance and computational complexity. With the special designed structure, the proposed system is able to suppress MAI and estimate code-delay simultaneously. It can outperform the conventional correlator-based system. We also analyze the convergence behavior and the mean acquisition time of the proposed scheme,

and derive related closed-form expressions. Simulations verify that theoretical and experimental results agree well. In this chapter, we only consider the AWGN integer chip-delay channels. With minor modification, the proposed system can be easily extended to apply in the multipath yet fractional chip-delay channels [57]. This issue may serve as a topic for further research.



Chapter 5

Conjugate Gradient Algorithm for Array Code Acquisition

The adaptive array proposed in Chapter 3 has a simple structure and can simultaneously perform adaptive beamforming and code-delay estimation. For simplicity, the well-known LMS adaptive algorithm was used in Chapter 3. However, the convergence of the adaptive filters becomes very slow when their dimensions become large or they are operated in multipath channel environments. The recursive-least-squares (RLS) algorithm can be applied; however, the computational complexity will be increased dramatically.

To solve the dilemma, in this chapter we propose the application of the conjugate gradient (CG) algorithm [59] in the adaptive array code acquisition problem. The CG algorithm has been well-developed in optimization theory and used in many fields, [60]– [61]. It can iteratively solve a quadratic minimization problem in just κ steps, where κ is the number of unknown parameters. However, it requires the exact knowledge of the second-order derivative of the quadratic cost function, which cannot be known in general signal processing problems. With an estimated derivative, the CG algorithm is degenerated [62], which means the finite-step convergence property may be not held. Nevertheless, the convergence of the CG algorithm is still fast. Note that the straightforward application of the original CG algorithm in adaptive

signal processing will require a very high computational complexity. In [62], a simple method was proposed to alleviate this problem. Still, the computational complexity is on the order of $\mathcal{O}(U^2)$, where U is the filter dimension. Exploiting the special structure of the correlation matrix, we propose a low-complexity CG algorithm for the array acquisition scheme in [57]. It is shown that the computational complexity can be reduced to the order of $\mathcal{O}(U)$, same as that of the LMS algorithm. However, the convergence of the proposed CG algorithm is significantly faster than that of the LMS algorithm.

§ 5.1 Proposed CG Adaptive Algorithm

In Chapter 3, we have proposed the adaptive antenna arrays code acquisition system. Optimal-weights of the system are derived with the eigen decomposition. However, the required computational complexity is on the order of $\mathcal{O}(U^3)$. In addition, the matrix inversion of \mathbf{R}_r is required in (3.20). To alleviate these problems, a constrained LMS algorithm is used in [57] to approach the optimum filter-weights. However, the LMS algorithm converges slowly, especially when the constraint is applied. In this section, we propose a CG algorithm to speed up the convergence. As mentioned, the RLS type of algorithms can be applied for the same purpose. However, the computational complexity of the RLS is on the order of $\mathcal{O}(U^2)$, which is still computationally expensive. It is known that the main effort in the RLS algorithm is to estimate the inverse of the correlation matrix. With a different perspective, the CG algorithm estimates the input correlation matrix itself. As a result, the CG algorithm will be more stable than the RLS algorithm. In its original form, the CG algorithm [59] is not suitable for adaptive processing. In [62], a simple modification allows the CG algorithm to reduce its complexity to the order of $\mathcal{O}(U^2)$. We will use the special structure inherent in the correlation matrix and propose a low-complexity CG algorithm with the computational complexity of order $\mathcal{O}(U)$. As shown, we have a unit-norm constraint on the temporal filter-weights. Applying this constraint, we then obtain a constrained

CG algorithm for adaptive filtering. Rewriting (3.10), we can have

$$\bar{J}(n) = \mathbf{w}_v^H(n) \mathbf{R}_v \mathbf{w}_v(n) \quad (5.1)$$

where

$$\mathbf{w}_v(n) \triangleq [\mathbf{w}_t^T(n), \mathbf{w}_s^T(n)]^T \quad (5.2)$$

$$\mathbf{v}(n) \triangleq [\mathbf{x}^T(n), -\mathbf{r}^T(n)]^T \quad (5.3)$$

$$\begin{aligned} \mathbf{R}_v &\triangleq E\{\mathbf{v}(n)[\mathbf{v}(n)]^H\} \\ &= \begin{bmatrix} \mathbf{I}_{(U \times U)} & \mathbf{K} \\ \mathbf{K}^H & \mathbf{R}_r \end{bmatrix}. \end{aligned} \quad (5.4)$$

Thus, the optimum solution of the filter-weights, denoted as $\mathbf{w}_{v,o} \triangleq [\mathbf{w}_{t,o}^T, \mathbf{w}_{s,o}^T]^T$, can also be expressed as

$$\mathbf{w}_{v,o} = \arg \min_{\mathbf{w}_v} \mathbf{w}_v^H \mathbf{R}_v \mathbf{w}_v \quad (5.5)$$

$$\text{s.t. } \|\mathbf{w}_{t,o}\| = 1. \quad (5.6)$$

Before our development, we briefly describe the constrained LMS algorithm used in [57].

It can be summarized as:

Given $\mu, \mathbf{w}_v(0)$

for $n = 0, 1, 2, \dots, N - 1$

$$e(n) = \mathbf{w}_v^H(n) \mathbf{v}(n) \quad (5.7)$$

$$\mathbf{H}(n) = \text{diag}\left\{ \underbrace{\frac{1}{\|\mathbf{w}_t(n)\|}, \dots, \frac{1}{\|\mathbf{w}_t(n)\|}}_U, \underbrace{1, \dots, 1}_M \right\} \quad (5.8)$$

$$\mathbf{w}_v(n+1) = \mathbf{H}(n) \mathbf{w}_v(n) - \mu \mathbf{v}(n) e^*(n) \quad (5.9)$$

end

where $\text{diag}\{\cdot\}$ denotes a diagonal matrix consisting of the argument it includes, μ the step size, and N the iteration number. Note that in (5.8) $\mathbf{H}(n)$ normalizes $\mathbf{w}_t(n)$ at every iteration. By

doing so, we can force $\mathbf{w}_i(n)$ to satisfy the unit-norm constraint. Due to the small step size and the normalization procedure, we can assume $\mathbf{H}(n) \approx \mathbf{I}$, and thus the excess mean-squared error (MSE) and step size bound are similar to those of the conventional LMS algorithm. While the conventional LMS algorithm requires $2(U + M) + 1$ multiplications per iteration, the constrained one mentioned above needs extra U multiplications and extra U divisions for normalization.

As mentioned, the original CG algorithm [59] iteratively solves a quadratic cost function (with a constant symmetric positive-definite matrix) and can converge in finite steps (the number of unknown parameters). It is well-known that it converges faster than steepest-descent methods, and has lower computational-complexity than Newton's method. Consider our problem here. The cost function in (5.5) to minimize is a purely quadratic cost function. Its second-order derivative is the correlation matrix \mathbf{R}_v , which is unknown in practice. This implies that an estimate, say $\hat{\mathbf{R}}_v(n)$, has to be used. As a result, the CG algorithm minimizes $\mathbf{w}_v^H(n) \hat{\mathbf{R}}_v(n) \mathbf{w}_v(n)$ subject to the unit-norm constraint. Similar to the constrained LMS algorithm, we can normalize the temporal filter-weight at each CG iteration, resulting a constrained CG algorithm. Note that the filter dimension is $U + M$ and we have one new input-vector at each time-instant. In other words, $\hat{\mathbf{R}}_v(n)$ is updated for each time-instant, and for each update of $\hat{\mathbf{R}}_v(n)$, $\kappa_i = U + M$ iterations have to be conducted for the CG algorithm. We refer this approach as the conventional CG (CCG) algorithm. Let $\mathbf{u}(j)$ be a $(U + M) \times 1$ vector for the iteration j , where $j = 0, \dots, \kappa_i - 1$. The application of the CCG algorithm in our problem can be summarized below.

$$\text{Given } \mathbf{u}(0), \bar{\mathbf{R}}_v(-1) = \mathbf{0}$$

$$\text{for } n = 0, 1, 2, \dots, N - 1$$

$$\bar{\mathbf{R}}_v(n) = \bar{\mathbf{R}}_v(n - 1) + \mathbf{v}(n) \mathbf{v}^H(n) \quad (5.10)$$

$$\hat{\mathbf{R}}_v(n) = \frac{1}{n} \bar{\mathbf{R}}_v(n) \quad (5.11)$$

$$\mathbf{g}_v(0) = \hat{\mathbf{R}}_v(n) \mathbf{u}(0) \quad (5.12)$$

$$\mathbf{d}(0) = -\mathbf{g}_v(0) \quad (5.13)$$

for $j = 0, 1, \dots, \kappa_i - 1$

$$\mu_C(j) = \frac{\|\mathbf{g}_v(j)\|^2}{\mathbf{d}^H(j)\hat{\mathbf{R}}_v(n)\mathbf{d}(j)} \quad (5.14)$$

$$\mathbf{u}(j+1) = \mathbf{u}(j) + \mu_C(j)\mathbf{d}(j) \quad (5.15)$$

$$\mathbf{u}(1:U, j+1) = \frac{\mathbf{u}(1:U, j+1)}{\|\mathbf{u}(1:U, j+1)\|} \quad (5.16)$$

$$\mathbf{g}_v(j+1) = \hat{\mathbf{R}}_v(n)\mathbf{u}(j+1) \quad (5.17)$$

$$\beta_g(j) = \frac{\|\mathbf{g}_v(j+1)\|^2}{\|\mathbf{g}_v(j)\|^2} \quad (5.18)$$

$$\mathbf{d}(j+1) = -\mathbf{g}_v(j+1) + \beta_g(j)\mathbf{d}(j) \quad (5.19)$$

end

$$\mathbf{w}_v(n) = \mathbf{u}(\kappa_i) \quad (5.20)$$

$$\mathbf{u}(0) = \mathbf{u}(\kappa_i) \quad (5.21)$$

end

Here, $\mathbf{g}_v(\cdot)$ stands for the gradient of (5.1), $\mathbf{d}(\cdot)$ the update direction, $\mu_C(\cdot)$ the optimum step size. Usually, we let $\mathbf{u}(0) = (1/\sqrt{U})[1, \dots, 1, 0, \dots, 0]^T$. In (5.10)–(5.11), the CCG first calculates $\hat{\mathbf{R}}_v(n)$ with the average in a rectangular window. Then, for each n the CCG iterates $\mathbf{u}(\cdot)$ by κ_i times and only $\mathbf{u}(\kappa_i)$ is of interest. Note that $\mathbf{u}(\cdot)$ is normalized for its first U elements at each iteration. Finally, $\mathbf{u}(\kappa_i)$ serves as $\mathbf{w}_v(n)$ and the initial $\mathbf{u}(0)$ for the next time-instant. Note that $\beta_g(j)$ attempts to provide $\hat{\mathbf{R}}_v(n)$ -conjugacy for $\mathbf{d}(j+1)$ with respect to previous directions $\{\mathbf{d}(j), \dots, \mathbf{d}(0)\}$ [59], [62]. It should be emphasized that the CCG performs minimization over $\hat{\mathbf{R}}_v(n)$ (not true \mathbf{R}_v) for each n . When n is large, $\hat{\mathbf{R}}_v(n)$ will approach \mathbf{R}_v , and $\mathbf{w}_v(n)$ approaches $\mathbf{w}_{v,o}$. As seen, the CCG resets the update direction at each time-instant [see (5.12) and (5.13)]. As mentioned, the correlation matrix is estimated and the property of the finite-step convergence is not held anymore. In our application, the problem becomes more apparent due to the weight normalization operation. Even with these problems, the convergence of the CCG algorithm is still fast. The CCG algorithm requires $2\kappa_i + 1$ matrix-vector multiplications for

every time-instant. Thus, the computational complexity is very high, and it is not suitable for sample-by-sample adaptive filtering application.

For sample-by-sample adaptive filtering, a modified CG algorithm [62] was developed to alleviate the high complexity problem. This algorithm updates the weights only one time per time-instant, and its computational complexity is lower. We call the algorithm as a modified CG algorithm (MCG). Applying the MCG algorithm to our problem, we have

$$\text{Given } \mathbf{w}_v(0), \bar{\mathbf{R}}_v(-1) = \mathbf{0}$$

$$\text{for } n = 0, 1, 2, \dots, N - 1$$

$$\bar{\mathbf{R}}_v(n) = \bar{\mathbf{R}}_v(n-1) + \mathbf{v}(n)\mathbf{v}^H(n) \quad (5.22)$$

$$\hat{\mathbf{R}}_v(n) = \frac{1}{n} \bar{\mathbf{R}}_v(n) \quad (5.23)$$

$$\left. \begin{array}{l} \mathbf{g}_v(n) = \hat{\mathbf{R}}_v(n)\mathbf{w}_v(n) \\ \mathbf{d}(n) = -\mathbf{g}_v(n) \end{array} \right\} \begin{array}{l} \text{applied only when} \\ n/N_r \text{ is an integer} \end{array} \quad (5.24)$$

$$\mu_C(n) = \frac{\|\mathbf{g}_v(n)\|^2}{\mathbf{d}^H(n)\hat{\mathbf{R}}_v(n)\mathbf{d}(n)} \quad (5.25)$$

$$\mathbf{w}_v(n+1) = \mathbf{w}_v(n) + \mu_C(n)\mathbf{d}(n) \quad (5.26)$$

$$\mathbf{w}_v(1:U, n+1) = \frac{\mathbf{w}_v(1:U, n+1)}{\|\mathbf{w}_v(1:U, n+1)\|} \quad (5.27)$$

$$\mathbf{g}_v(n+1) = \hat{\mathbf{R}}_v(n)\mathbf{w}_v(n+1) \quad (5.28)$$

$$\beta_g(n) = \frac{\|\mathbf{g}_v(n+1)\|^2}{\|\mathbf{g}_v(n)\|^2} \quad (5.29)$$

$$\mathbf{d}(n+1) = -\mathbf{g}_v(n+1) + \beta_g(n)\mathbf{d}(n) \quad (5.30)$$

end

Here, N_r denotes the length of a reset period. Inside the period, the correlation matrix remains the same. At the moment that the correlation matrix is updated, $\mathbf{d}(\cdot)$ is reset to $-\mathbf{g}_v(\cdot)$.

Although the computational complexity of the MCG is lower, it is still much higher than the LMS algorithm. In our application, there is a special structure in the correlation matrix. This can be seen from (5.4), in which the upper-left $U \times U$ sub-matrix in \mathbf{R}_v is an identity matrix.

Also note that in typical array acquisition, $U \gg M$. Using these properties, we now propose a CG algorithm that can effectively reduce the computational complexity to the order similar to the LMS algorithm. We call the algorithm as the low-complexity CG (LCG) algorithm. The algorithm is summarized as below.

$$\text{Given } \mathbf{w}_v(0), \bar{\mathbf{K}}(-1) = \mathbf{0}, \bar{\mathbf{R}}_r(-1) = \mathbf{0}$$

$$\text{for } n = 0, 1, 2, \dots, N - 1$$

$$\bar{\mathbf{K}}(n) = \bar{\mathbf{K}}(n - 1) - \mathbf{x}(n)\mathbf{r}^H(n) \quad (5.31)$$

$$\bar{\mathbf{R}}_r(n) = \bar{\mathbf{R}}_r(n - 1) + \mathbf{r}(n)\mathbf{r}^H(n) \quad (5.32)$$

$$\hat{\mathbf{K}}(n) = \frac{1}{n}\bar{\mathbf{K}}(n) \quad (5.33)$$

$$\hat{\mathbf{R}}_r(n) = \frac{1}{n}\bar{\mathbf{R}}_r(n) \quad (5.34)$$

$$\hat{\mathbf{R}}_v(n) = \begin{bmatrix} \mathbf{I}_{(U \times U)} & \hat{\mathbf{K}}(n) \\ \hat{\mathbf{K}}(n)^H & \hat{\mathbf{R}}_r(n) \end{bmatrix} \quad (5.35)$$

$$\left. \begin{array}{l} \mathbf{g}_v(n) = \hat{\mathbf{R}}_v(n)\mathbf{w}_v(n) \\ \mathbf{d}(n) = -\mathbf{g}_v(n) \end{array} \right\} \begin{array}{l} \text{applied only when} \\ n/N_r \text{ is an integer} \end{array} \quad (5.36)$$

$$\mu_C(n) = \frac{\|\mathbf{g}_v(n)\|^2}{\mathbf{d}^H(n)\hat{\mathbf{R}}_v(n)\mathbf{d}(n)} \quad (5.37)$$

$$\mathbf{w}_v(n + 1) = \mathbf{w}_v(n) + \mu_C(n)\mathbf{d}(n) \quad (5.38)$$

$$\mathbf{w}_v(1 : U, n + 1) = \frac{\mathbf{w}_v(1 : U, n + 1)}{\|\mathbf{w}_v(1 : U, n + 1)\|} \quad (5.39)$$

$$\mathbf{g}_v(n + 1) = \hat{\mathbf{R}}_v(n)\mathbf{w}_v(n + 1) \quad (5.40)$$

$$\beta_g(n) = \frac{\|\mathbf{g}_v(n + 1)\|^2}{\|\mathbf{g}_v(n)\|^2} \quad (5.41)$$

$$\mathbf{d}(n + 1) = -\mathbf{g}_v(n + 1) + \beta_g(n)\mathbf{d}(n) \quad (5.42)$$

end

The key idea of the proposed algorithm is in (5.35), where the correlation matrix is only partially calculated. We approximate the upper-left $U \times U$ sub-matrix in $\hat{\mathbf{R}}_v(n)$ as an identity matrix, i.e.,

$\mathbf{I}_{(U \times U)}$. Thus, only small correlation matrices \mathbf{K} and \mathbf{R}_r are required to be estimated. In such a manner, the matrix-vector multiplications in (5.36), (5.37), and (5.40) can be significantly simplified. Since the elements in $\mathbf{x}(n)$ are binary, (i.e., ± 1), the calculation of $\mathbf{x}(n)\mathbf{r}^H(n)$ in (5.31) is also simple to implement. As that in the MCG algorithm, the direction vector is periodically reset [in (5.36)]. The choice of N_r will influence the performance [62]. For a small N_r , the performance will be better, but the complexity is higher.

In this paragraph, we analyze the computational-complexity requirement for the CCG, MCG, and LCG algorithms. The complexity considered here is the required multiplications and divisions for each time-instant. Note that the complexity for the MCG and LCG algorithms is considered at the time instants that reset is applied. We show the results in Table 5.1.

Table 5.1: Computational Complexity Comparison for Constrained CCG, MCG, and LCG Algorithms

| | Multiplications & divisions |
|-----|---|
| CCG | $2(U + M)^3 + 6(U + M)^2 + (2U + 1)(U + M)$ |
| MCG | $3(U + M)^2 + 7U + UM + M^2 + 6M$ |
| LCG | $7U(M + 1) + 4M^2 + 6M$ |

As we can see from the table, the complexity of the CCG algorithm is on the order of $\mathcal{O}(U^3)$, that of MCG is $\mathcal{O}(U^2)$, and that of LCG is $\mathcal{O}(U)$. Fig. 5.1 shows the ratio of the required computational complexity for the LCG and MCG algorithms. As we can see, the ratio is decreasing along with the increase of U . When $U = 128$ and $M = 4$, the complexity of the LCG algorithm is only 8.5% of that of the MCG algorithm. We then conclude that the LCG algorithm is much more efficient than the MCG algorithm.

The SINR at beamformer output $\gamma(n)$ [see Fig. 3.2] can serve as an indicator for the effec-

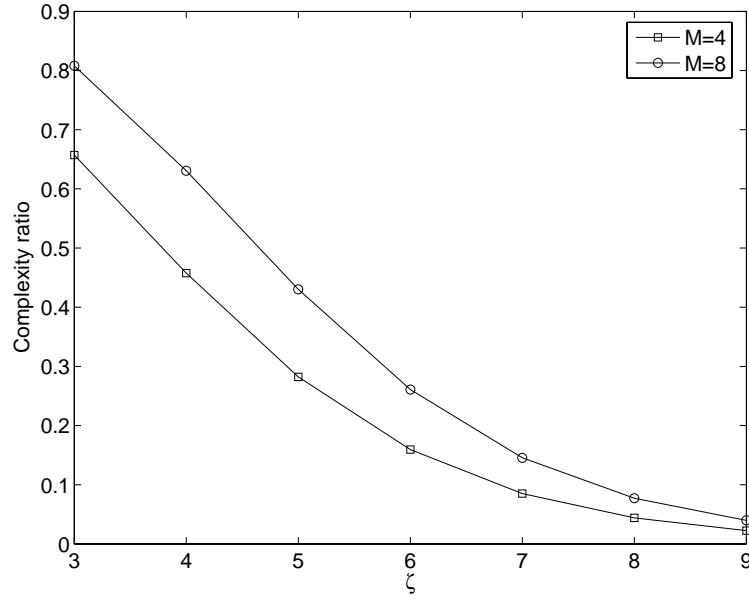


Figure 5.1: Ratio of the computational complexity of MCG and LCG versus U and M ($U = 2^x, x \in \{3, 4, \dots, 9\}$).

tiveness of interference suppression. The beamformer output here is given by

$$\gamma(n) = \mathbf{w}_s^H(n) \left(\sum_{l=1}^L \mathbf{a}_l \alpha_l x(n - \tau_l) + \mathbf{z}_M(n) \right), \quad (5.43)$$

where

$$\mathbf{z}_M(n) \triangleq \sum_{k=2}^K \sum_{l=1}^{L_k} \mathbf{a}_{k,l} \alpha_{k,l} x_k(n - \tau_{k,l}) + \boldsymbol{\eta}(n). \quad (5.44)$$

The output SINR of the optimum beamformer can then be expressed as

$$\text{SINR}_o = \frac{\mathbf{w}_{s,o}^H \mathbf{R}_s \mathbf{w}_{s,o}}{\mathbf{w}_{s,o}^H \mathbf{R}_{z_M} \mathbf{w}_{s,o}}, \quad (5.45)$$

where $\mathbf{R}_s \triangleq \sum_{l=1}^L |\alpha_l|^2 \mathbf{a}_l \mathbf{a}_l^H$, $\mathbf{R}_{z_M} \triangleq E\{\mathbf{z}_M(n) \mathbf{z}_M^H(n)\}$, and $\mathbf{w}_{s,o}$ denotes the optimum beamformer weights derived above.

§ 5.2 Simulation Results

To demonstrate the effectiveness of the proposed algorithm, we report some simulation results in this section. We let commonly used parameters as $U = 16$, $M = 8$, $K = 5$, and $\sigma_\eta^2 = 1$. The array input SINR is set as -10 dB, with equal power for each jammer. Also, we let the desired user's channel be a three-path channel ($L = 3$) with $\tau_1 = 2$, $\tau_2 = 5$, and $\tau_3 = 10$. The channels for jammers are assumed to have single paths (i.e., $L_k = 1$ for $k \neq 1$). The DoAs for the jammers and desired user are assumed to be $\{\phi_{k,1}\}_{k=2}^K = \{-0.2014, -0.5236, 0, 1.1198\}$ and $\{\phi_{1,l}\}_{l=1}^L = \{-0.6561, 0.5586, 0.7754\}$ radians, respectively. The path gains associated with the desired user are fixed to be $\{|\alpha_l|\}_{l=1}^L = \{0.66, 0.45, \sqrt{1 - 0.66^2 - 0.45^2}\}$.

For the LCG algorithm, the estimated correlation matrix may not be positive definite when the number of input vectors is small. This will lead to large $\|\mathbf{g}_v(\cdot)\|$ affecting convergence greatly. To mitigate this problem, we limit the values of $\mu_C(n)$ and $\beta_g(n)$ when n is small. For the simulations conducted below, we let $0 < \mu_C(n) \leq 0.01$ and $0 < \beta_g(n) \leq 0.99$ when $n < 100$. Also let $N_r = U + M$, and simulation results be derived with 200 trials.

Figs. 5.2-5.3 show the learning curves for the constrained CCG, MCG, LCG, and LMS algorithms. In these figures, the minimum MSE, J_{min} , being equal to 0.232, is also shown [see (3.22)]. As we can see from Fig. 5.2, the CCG can approach J_{min} very rapidly. The convergence of the MCG and LCG algorithms is slower. Also, the convergence of the LCG is slightly slower than that of the MCG. This stems from the fact that LCG approximates the sub-matrix in $\hat{\mathbf{R}}_v(n)$ as an identity matrix. We find that both MCG and LCG can reach steady-state around 350 iterations. Fig. 5.3 shows the learning results for the constrained LMS algorithm. Two step sizes are used. The first one is the maximum allowable step size which is 1.3×10^{-2} . This step size will let the LMS algorithm have the fastest convergence. However, the corresponding steady-state MSE is also large (0.85). In this case, the LMS converges around 1000 iterations. To obtain a comparable steady-state MSE with that of the LCG, we use another step size which is 5×10^{-3} . With this step size, the LMS algorithm converges around 2000 iterations. From

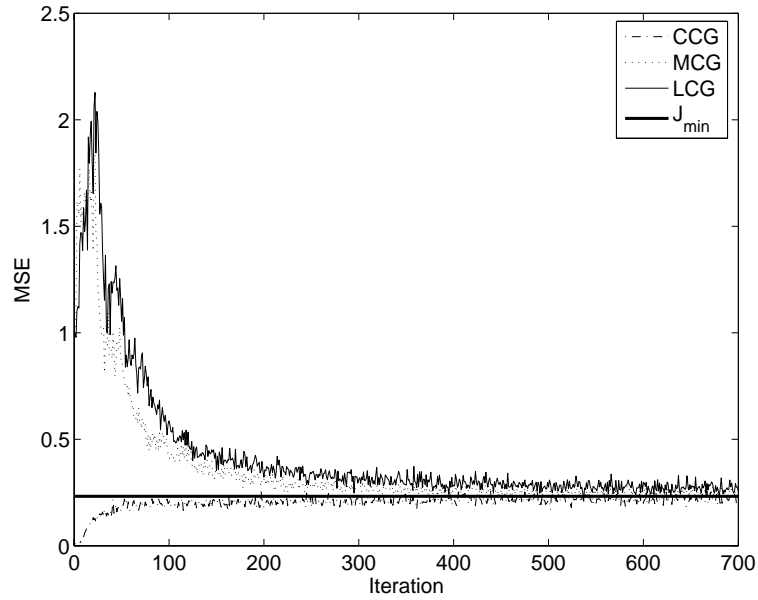


Figure 5.2: Learning curves for constrained CG algorithms ($N_r = U + M$).

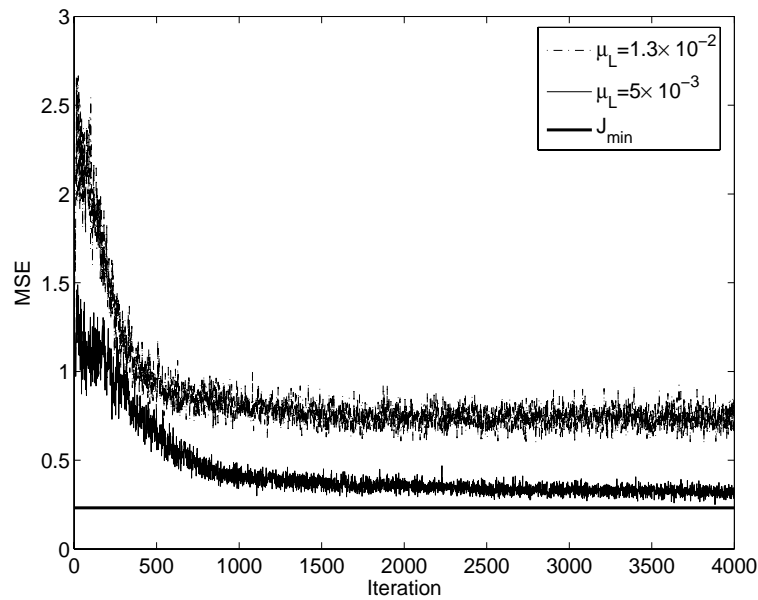


Figure 5.3: Learning curves for constrained LMS algorithm.

these results, we can clearly see that while the computational complexity of the LCG algorithm is comparable to that of the LMS algorithm, the convergence is much faster.

Next, let us examine the convergence of temporal filter-weights. Note that the magnitudes of those weights directly influence the performance of code acquisition. In Fig. 5.4, we only show a weight, which is $|w_{t,\tau_3}(n)|^2$, for clarity. In the scenario considered, $|w_{t,o,\tau_3}|^2$ corresponds to a significant tap in the desired beamformed channel. For the filter-weights that do not correspond to multipath delay positions, their values will decay to a very small value close to zero [57]. As seen, the CCG converges rapidly, and its steady-state value is close to the optimum. The MCG algorithm converges slower, but its steady-state value is the same as the CCG one. The convergence of LCG is similar to that of MCG. However, its steady-state value has a small bias. Fig. 5.5 shows the results for the constrained LMS algorithm. The LMS algorithm requires around 2000 and 4000 iterations to convergence for $\mu = 1.3 \times 10^{-2}$ and 5×10^{-3} , respectively. We also can see that the LMS algorithm has the bias problem too. The larger the step size, the larger the bias.

To examine the effectiveness of the beamformer, we show the SINR of the spatial filter output in Fig. 5.6. The SINR of CCG rapidly approaches the theoretical value of 5.2 dB [see (5.45)]. Similar to previous results, the SINR convergence for MCG and LCG is slower; the difference between these two algorithms are not obvious. Fig. 5.7 shows the beam-patterns formed by the algorithms (derived with $n = 200$). As expected, the beampatterns have multiple main-beams to collect the multipath signals of the desired user and put nullities to the DoAs of interfering signals. We find that all algorithms can deeply null the interference. Yet, the performance of LCG is slightly poorer.

In above simulations, we let $N_r = U + M$. As mentioned, N_r will influence the adaptation performance. Fig. 5.8 shows the convergence curve of $|w_{t,\tau_3}(n)|^2$ for $N_r = U + M$ and $3(U + M)$. We can clearly see that for both the MCG and LCG with $N_r = 3(U + M)$, the weight grows faster only when the update direction is reset. The convergence for $N_r = 3(U + M)$ is then slower than that for $N_r = U + M$.

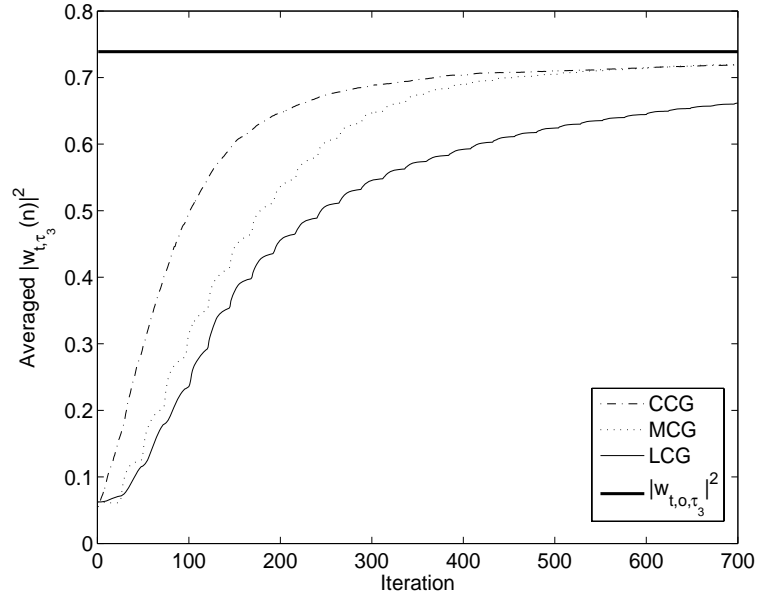


Figure 5.4: Convergence curves of $|w_{t,\tau_3}(n)|^2$ for constrained CG algorithms ($N_r = U + M$).

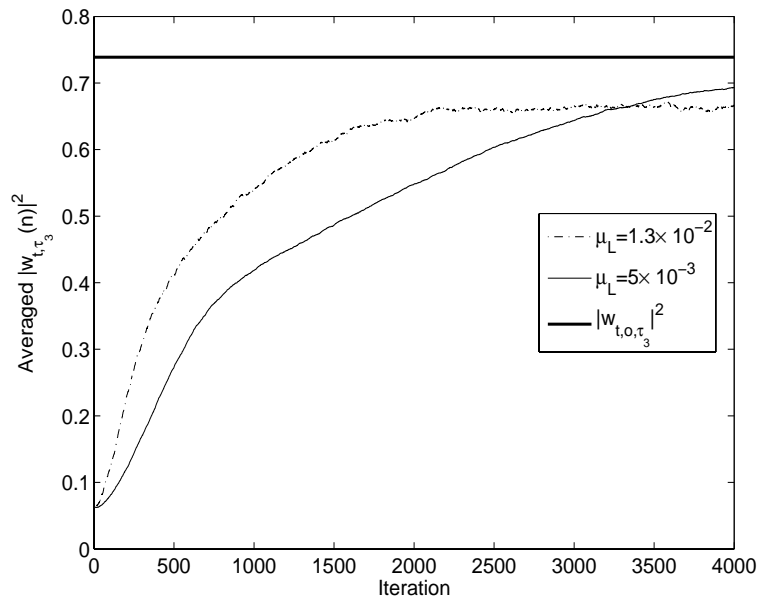


Figure 5.5: Convergence curves of $|w_{t,\tau_3}(n)|^2$ for constrained LMS algorithm.

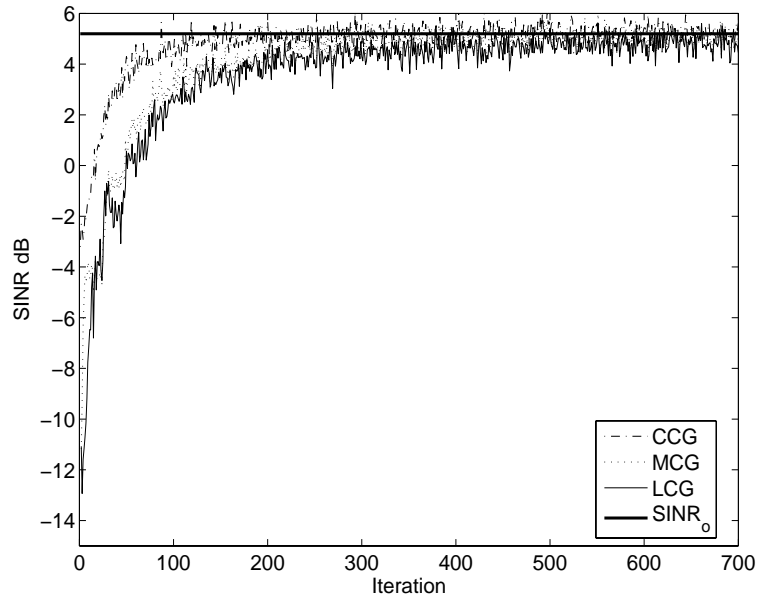


Figure 5.6: Convergence curve of SINR for constrained CG algorithms ($N_r = U + M$).

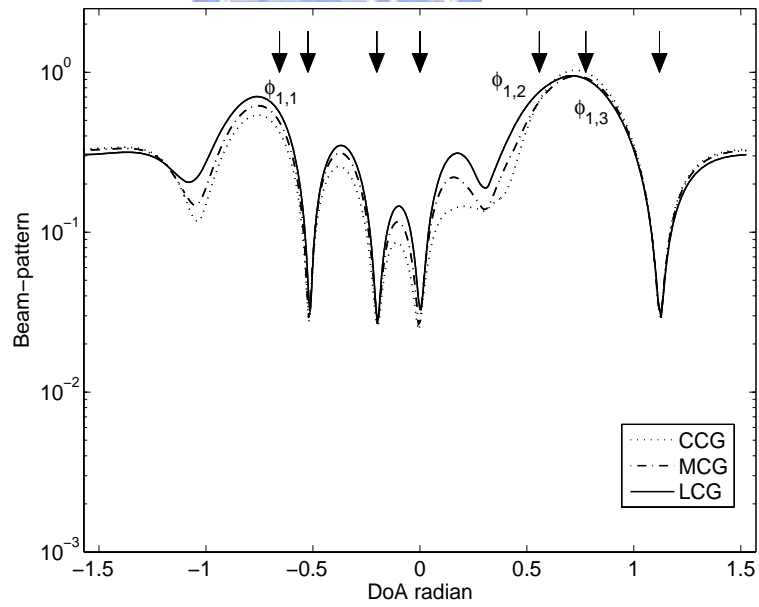


Figure 5.7: Beam-patterns for constrained CG algorithms. Arrow signs indicate DoAs of all users; labeled ones are DoAs of the desired user.

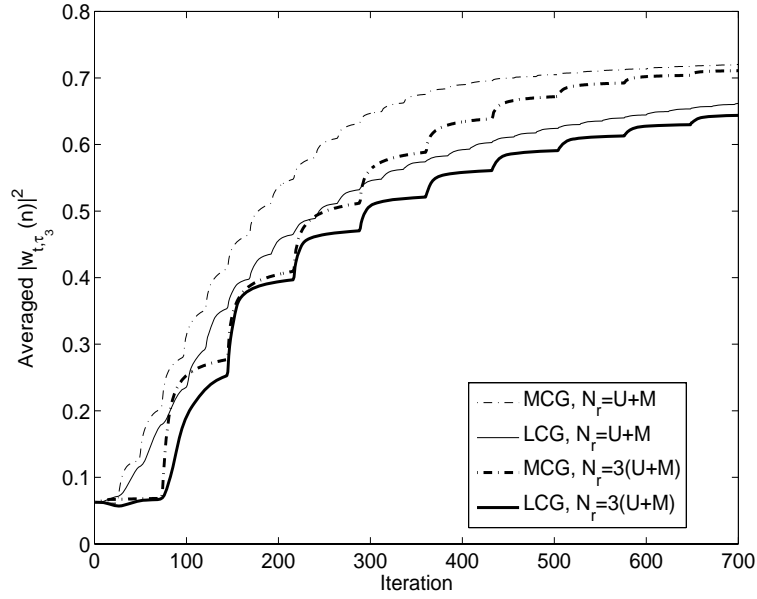


Figure 5.8: Effect of N_f on convergence of $|w_{t,\tau_3}(n)|^2$.

Finally, we consider the performance of code acquisition. We define the correct acquisition as the situation that $\{|w_{t,\tau_1}(N)|^2, |w_{t,\tau_2}(N)|^2, |w_{t,\tau_3}(N)|^2\}$ correspond to the first three maximal filter-weights in $\mathbf{w}_t(N)$. Also, denote the probability of correct acquisition as P_c . Fig. 5.9 shows P_c versus different N (array input SINR = -10 dB) for various algorithms. The results here are derived with 10^4 trials. As seen, the constrained LMS gives very poor performance when the training period, N , is short, while the MCG and LCG are not sensitive to the period. Also, the performance gap between the CG algorithm is very small justifying the effectiveness of the LCG algorithm. For the LCG, we can then use a small N effectively shortening the acquisition time.

§ 5.3 Conclusions

In this chapter, we propose an adaptive algorithm with the conjugate gradient algorithm to solve the slow convergence problem associated with the adaptive array code acquisition in [57]. Un-

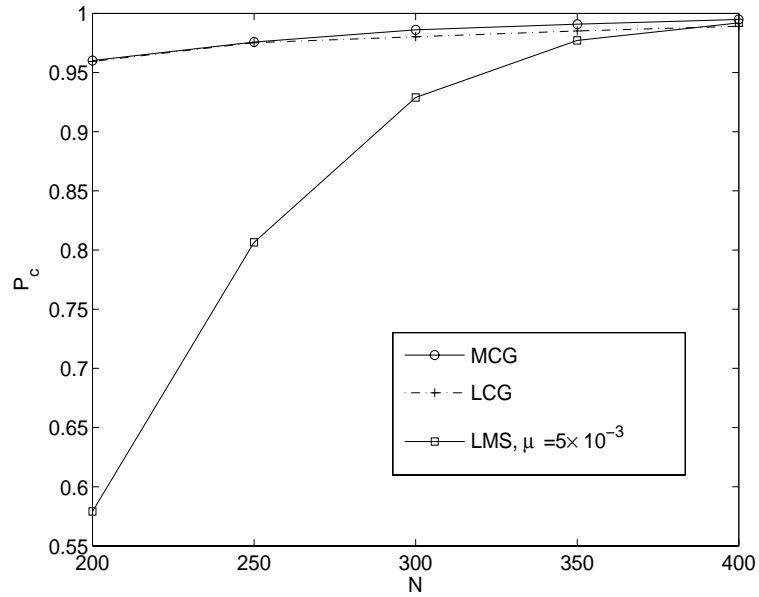


Figure 5.9: Probability of correct acquisition ($N_r = U + M$).

like the MCG algorithm, the proposed method, exploiting the special structure inherent in the correlation matrix, requires a low computational-complexity. We have shown that the computational complexity of the proposed method is on the same order of the LMS algorithm. However, the convergence rate is much faster. Simulation results show that the performance of adaptive array code acquisition with the proposed CG algorithm is comparable to that with the MCG algorithm. In this chapter, only integer code-delays are considered. However, with minor modifications, the proposed algorithm can also be applied to a scenario with fractional multipath-delays [57]. Besides, it may be feasible to apply the proposed algorithm in MIMO CDMA systems. These may serve as topics for future research.

Chapter 6

Conclusions and Future Works

In this dissertation, we consider code acquisition with adaptive filtering techniques. From the analysis and experimental results, we conclude that proposed adaptive-filtering based code acquisition algorithms can either significantly reduce the computational complexity or enhance the performance of conventional correlator-based algorithms.

In Chapter 2, we first proposed a multirate adaptive-filtering scheme for single-antenna systems. Using the decimation property in the multirate signal processing, we are able to significantly reduce the computational complexity of the conventional adaptive-filtering scheme. However, in this work, we assume that the channel is AWGN, the code-delay is an integer, and carrier frequency offset is not present. To be applicable in real-world, multipath channels, carrier frequency offset, and fractional code-delay have to be taken into considered.

To deal with code acquisition in array systems, we proposed a novel adaptive array system in Chapter 3. The system can simultaneously suppress MAI and estimate the code-delays. We have shown that its mean acquisition time is much smaller than the correlator-based system. We have also analyzed the proposed system and derived related closed-form expressions. Simulations show that theoretical results for MSE, probability of acquisition error, and beamformer output SINR agree with experimental results very well. Multiple-input-multiple-output (MIMO) systems have become more popular nowadays. With some modification, the proposed

adaptive array code acquisition can be applied in MIMO CDMA systems, serving as a good topic for further research.

As mentioned before, the complexity of the temporal filter increases with large delay uncertainty. In Chapter 4, we use the serial-search technique to solve the problem. With the proposed structure, it is easy to obtain a compromise between performance and computational complexity. Except for the serial-search scheme, another possibility for computational complexity reduction may be the multirate processing technique developed in Chapter 2.

In Chapter 5, we employed the CG algorithm to cope with the slow convergence problem of the proposed adaptive array system. Although the proposed algorithm, referred to as LCG, has the same order of computational complexity as that of the LMS algorithm, it can provide much faster convergence performance. The proposed CG algorithm can also be applied to the scenario of fractional multipath-delays or in MIMO environments.



Bibliography

- [1] A. Polydoros and C. L. Weber, "A unified approach to serial search spread-spectrum code acquisition—part II: a matched-filter receiver," *IEEE Trans. on Commun.*, vol. 32, No. 5, pp. 550-560, May 1984.
- [2] R. R. Rick and L. B. Milstein, "Parallel acquisition in mobile DS-CDMA systems," *IEEE Trans. on Commun.*, vol. 45, No. 11, pp. 1466-1476, Nov. 1997.
- [3] Yu T. Su, "Rapid code acquisition algorithms employing PN matched filters," *IEEE Trans. on Commun.*, vol. 36, No. 6, pp. 724-733, June 1988.
- [4] E. Sourour and S. C. Gupta, "Direct-sequence spread-spectrum parallel acquisition in a fading mobile channel," *IEEE Trans. on Commun.*, vol. 38, no. 7, pp. 992-998, 1990.
- [5] T. K. Moon, R. T. Short, and C. K. Rushforth, "Average acquisition time for SSMA channels," in *IEEE Military Communication Conference*, pp. 1042-1046, 1991.
- [6] G. E. Corazza and V. Degli-Esposti, "Acquisition-based capacity estimates for CDMA with imperfect power control," in *IEEE International Symposium on Spread Spectrum Techniques and Applications*, vol. 1, pp. 325 -329, July 1994.
- [7] U. Madhow and M. B. Pursley, "Acquisition in direct-sequence spread-spectrum communication networks: an asymptotic analysis," *IEEE Trans. on Information Theory*, vol. 39, No. 3, pp. 903-912, May 1993.

- [8] R. L. Pickholtz, L. B. Milstein, and D. L. Schilling, "Spread spectrum for mobile communication," *IEEE Trans. on Vehicular Technology*, Vol. 40, No. 2, pp. 313-322, May 1991.
- [9] A. G. Dabak, "Acquisition based capacity of a synchronous direct sequence spread spectrum multiple access technique," in *IEEE International Symposium on Information Theory*, pp. 141, 1994.
- [10] J. K. Holmes and C. C. Chen, "Acquisition time performance of PN spread-spectrum systems," *IEEE Trans. on Commun.*, vol. 25, No. 8, pp. 778-783, May 1977.
- [11] Ho-Chi Hwang and Che-Ho Wei, "A new blind adaptive interference suppression scheme for acquisition and MMSE demodulation of DS/CDMA signals," *IEEE Trans. on Vehicular Technology*, vol. 49, No. 3, pp. 875-884, May 2000.
- [12] E. G. Ström, S. Parkvall, S. L. Miller, and Björn E. Ottersten, "Propagation delay estimation in asynchronous direct-sequence code-division multiple access systems," *IEEE Trans. on Commun.*, vol. 44, No. 1, pp. 84-93, Jan. 1996.
- [13] S. E. Bensley and B. Aazhang, "Subspace-based channel estimation for code division multiple access communication systems," *IEEE Trans. on Commun.*, vol. 44, No. 8, pp. 1009-1020, Aug. 1996.
- [14] Sangchoon Kim, "Improved MUSIC algorithm for the code-timing estimation of DS-CDMA multipath-fading channels in multiantenna systems," *IEEE Trans. on Vehicular Technology*, vol. 53, No. 5, pp. 1354-1369, Sept. 2004.
- [15] Peter K. P. Cheung and P. B. Rapajic, "CMA-based code acquisition scheme for DS-CDMA systems," *IEEE Trans. on Commun.*, vol. 48, No. 5, pp. 852-862, May 2000.

- [16] R. Wang, H. Li, and T. Li, "Code-timing estimation for CDMA systems with bandlimited chip waveforms," *IEEE Trans. on Wireless Commun.*, vol. 3, No. 4, pp. 1338-1348, July 2004.
- [17] Y. Ma, K. H. Li, A. C. Kot, and G. Ye, "A blind code timing estimator and its implementation for DS-CDMA signals in unknown colored noise," *IEEE Trans. on Vehicular Technology*, vol. 51, No. 6, pp. 1600-1607, Nov. 2002.
- [18] D. Zheng, J. Li, S. L. Miller, and E. G. Ström, "An efficient code-timing estimator for DS-CDMA signals," *IEEE Trans. on Signal Processing*, vol. 45, pp. 82-89, Jan. 1997.
- [19] M. G. El-Tarhuni and Asrar U. H. Sheikh, "PN code acquisition in CDMA systems using a MMSE adaptive filter," in *IEEE Canadian Conference on Electrical and Computer Engineering*, vol. 2, pp. 746-749, May 1998.
- [20] —, "An adaptive filtering PN code acquisition scheme with improved acquisition based capacity in DS/CDMA," in *9th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 3, pp. 1486-1490, Sept. 1998.
- [21] —, "Adaptive synchronization for spread spectrum systems," in *46th IEEE Vehicular Technology Conference*, vol. 1, pp. 170-174, 1996.
- [22] M. G. El-Tarhuni, Application of adaptive filtering to direct-sequence spread-spectrum code synchronization, Ph. D. Thesis proposal, Department of System and Computer Engineering, Carleton university, Canada, Jan. 1996.
- [23] T. Yu, J. Kwun, H. Jeon, D. Hong, and C. Kang, "Noncoherent adaptive code synchronization for DS/CDMA systems," in *IEEE Global Telecommunications Conference*, vol. 6, pp. 3311-3315, Nov. 2001.

- [24] M. G. El-Tarhuni and Asrar U. H. Sheikh, "Code acquisition of DS/SS signals in fading channels using an LMS adaptive filter," *IEEE Communication Letters*, vol. 2, No. 4, pp. 85-88, April 1998.
- [25] H. L. Yang and W. R. Wu, "Multirate adaptive filtering for DS/CDMA code acquisition," in *IEEE International Symposium on Signal Processing and Information Technology*, pp. 363-366, Dec. 2003.
- [26] R. F. Smith and S. L. Miller, "Acquisition performance of an adaptive receiver for DS-CDMA," *IEEE Trans. on Commun.*, vol. 47, No. 9, pp. 1416-1424, Sept. 1999.
- [27] H. R. Park and B. J. Kang, "On the performance of a maximum-likelihood code-acquisition technique for preamble search in a CDMA reverse link," *IEEE Trans. on Vehicular Technology*, vol. 47, No. 1, pp. 65-74, Feb. 1998.
- [28] TIA/EIA/IS95, *Mobile station-base station compatibility standard for dual-mode wide-band spread spectrum cellular system*: Telecommun. Industry Assoc., July, 1993.
- [29] TIA cdma.2000, *Wideband cdmaOne radio transmission technology proposal*: Int. Telecommun. Union, Radiocommun. Study Groups, June 1998.
- [30] E. Dahlman et al., "WCDMA-The radio interface for future mobile multimedia communications," *IEEE Trans. on Vehicular Technology*, vol. 47, pp. 1105-1118, Nov. 1998.
- [31] P. Taaghoul et al., "Satellite UMTS/IMT2000 W-CDMA air interfaces," *IEEE Commun. Mag.*, vol. 37, pp. 116-126, Sept. 1999.
- [32] A. J. Viterbi, *Principle of Spread Spectrum Communications*. New York: Addison-Wesley, 1995.
- [33] N. J. Bershad and L. Z. Qu, "On the probability density function of the LMS adaptive filter weights," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 37, No. 1, pp. 43-56, Jan. 1989.

- [34] Haykin, S., *Adaptive Filter Theory*, 3rd. ed., Prentice-Hall, 1996.
- [35] J. G. Proakis, *Digital Communications*, 4th ed., McGraw-Hill, 2000.
- [36] J. K. Holmes, *Coherent Spread Spectrum Systems*. New York: Wiley, 1982.
- [37] M. Simon et al., *Spread Spectrum Communications Handbooks*, Revised Edition, McGraw-Hill, Inc., New York, 1994.
- [38] Y. Zhang, L. Zhang, and G. Liao, "PN code acquisition and beamforming weight acquisition for DS-CDMA systems with adaptive array," in *14th IEEE Int. Symp. on Personal, Indoor, and Mobile Radio Communications*, vol. 2, pp. 1385-1389, 2003.
- [39] B. Wang and H. M. Kwon, "PN code acquisition using smart antenna for spread-spectrum wireless communications—part I," *IEEE Trans. Veh. Technol.*, vol. 52, No. 1, pp. 142-149, Jan. 2003.
- [40] —, "PN code acquisition using smart antenna for spread-spectrum wireless communications—part II," *IEEE Trans. Wireless Commun.*, vol. 2, No. 1, pp. 108-117, Jan. 2003.
- [41] H. L. Yang and W. R. Wu, "A novel adaptive code acquisition using antenna array for DS/CDMA systems," in *IEEE Int. Workshop on Antenna Technology: Small Antennas and Novel Metamaterials*, pp. 458-461, Mar. 2005.
- [42] M. D. Katz, J. Inatti, and S. Glisic, "Two-dimensional code acquisition in time and angular domains," *IEEE J. Select. Areas Commun.*, vol. 19, No. 12, pp. 2441-2451, Dec. 2001.
- [43] —, "Two-dimensional code acquisition in environments with a spatially nonuniform distribution of interference: algorithms and performance," *IEEE Trans. Wireless Commun.*, vol. 3, No. 1, pp. 1-7, Jan. 2004.

- [44] S. Buzzi and H. V. Poor, "On parameter estimation in long-code DS/CDMA systems: cramer-rao bounds and least-squares algorithms," *IEEE Trans. Signal Processing*, vol. 51, No. 2, pp. 545-559, Feb. 2003.
- [45] C. Sengupta, J. R. Cavallaro, and B. Aazhang, "On multipath channel estimation for CDMA systems using multiple sensors," *IEEE Trans. Commun.*, vol. 49, No. 3, pp. 543-553, June 2001.
- [46] S. Affes and P. Mermelstein, "A new receiver structure for asynchronous CDMA: STAR—the spatial-temporal array-receiver," *IEEE J. Select. Areas Commun.*, vol. 16, No. 8, pp. 1411-1421, Oct. 1998.
- [47] J. Ramos, M. D. Zoltowski, and H. Liu, "Low-complexity space-time processor for DS-CDMA Communications," *IEEE Trans. Signal Processing*, vol. 48, No. 1, pp. 39-52, Jan. 2000.
- [48] Z. Liu, J. Li, and S. L. Miller, "An efficient code-timing estimator for receiver diversity DS-CDMA systems," *IEEE Trans. Commun.*, vol. 46, No. 6, pp. 826-835, June 1998.
- [49] G. Seco, A. L. Swindlehurst, and D. Astely, "Exploiting antenna arrays for synchronization," in *Signal Processing Advances in Wireless and Mobile Communications*, G. B. Giannakis et al., Eds. Upper Saddle River, NJ: Prentice-Hall, vol. 2, 2001.
- [50] O. S. Shin and K. B. Lee, "Utilization of multipath for spread-spectrum code acquisition in frequency-selective rayleigh fading channels," *IEEE Trans. Commun.*, vol. 49, pp. 734-743, Apr. 2001.
- [51] H. W. Je, O.-S. Shin, and K. B. Lee, "Acquisition of DS/CDMA systems with multiple antennas in frequency-selective fading channels," *IEEE Trans. Wireless Commun.*, vol. 2, pp. 787-798, July 2003.

- [52] TIA/EIA/IS95, *Mobile station-base station compatibility standard for dual-mode wide-band spread spectrum cellular system*: Telecommun. Industry Assoc., July, 1993.
- [53] E. H. Diana, B. Jabbari, and G. Mason, "Spreading codes for direct sequence CDMA and wideband CDMA cellular networks," *IEEE Commun. Mag.*, vol. 36, pp. 48-54, Sept. 1998.
- [54] —, "Multirate adaptive filtering for low complexity DS/CDMA code acquisition," submitted to *IEEE Trans. Veh. Technol.*, Aug. 2005.
- [55] Y. Zhang, L. Zhang and G. Liao, "PN code acquisition and beamforming weight acquisition for DS-CDMA systems with adaptive array," in 14th IEEE Int. Symp. on Personal, Indoor, and Mobile Radio Communications, vol. 2, pp. 1385-1389, 2003.
- [56] Sangchoon Kim, "Approximate maximum likelihood approach for code acquisition in DS-CDMA systems with multiple antennas," *IEICE Trans. Commun.*, vol. E88-B, no. 3, pp. 1054-1065, Mar. 2005.
- [57] H. L. Yang and W. R. Wu, "A novel adaptive antenna array for DS/CDMA code acquisition," *IEEE Trans. Signal Process.*, Sept. 2006, accepted.
- [58] —, "A low-complexity adaptive antenna array for DS/CDMA code acquisition," submitted to *IEICE Trans. Commun.*, Apr. 2006.
- [59] E. K. P. Chong and S. H. Zak, *An introduction to optimization*, 2nd ed. NY: Wiley, 2001.
- [60] G. K. Boray and M. D. Srinath, "Congugate gradient techniques for adaptive filtering," *IEEE Trans. Circuit and System*, vol. 39, pp. 1-10, Jan. 1992.
- [61] P. S. Chang and A. N. Willson, Jr., "Adaptive spectral estimation using the conjugate gradient algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Atlanta, GA, pp. 2979-2982, May 1996.

- [62] —, “Analysis of conjugate gradient algorithms for adaptive filtering,” *IEEE Trans. Signal Processing*, vol. 48, pp. 409-418, Feb. 2000.

