A STUDY OF EFFICIENT ADAPTIVE FILTERING ALGORITHMS AND THEIR APPLICATIONS TO ACOUSTIC AND COMMUNICATION SYSTEMS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF
COMMUNICATIONS & INTEGRATED SYSTEMS
OF TOKYO INSTITUTE OF TECHNOLOGY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF ENGINEERING

Masahiro YUKAWA
June 2006

Acknowledgements

This thesis completes my studies in the fascinating research group of Sakaniwa & Yamada Laboratory in Tokyo Institute of Technology since April 2001.

Having in mind that words cannot fully describe feelings, I would like to express my deepest gratitude to my advisor Associate Professor Isao Yamada for his guidance, support, and friendship through all these years.

I would like also to express my deep gratitude to my advisor Professor Kohichi Sakaniwa for his support and for giving me the chance to pursue my graduate studies in Tokyo Institute of Technology.

It is my pleasure to deeply thank an Associate Professor Tomoharu Shibuya of NIME (National Institute of Multimedia Education), who had been an Assistant Professor in Sakaniwa & Yamada Laboratory until March 2003, for his interesting seminar on coding theory. It is also my pleasure to deeply thank an Associate Professor Hiroshi Hasegawa of Nagoya University, who had been also an Assistant Professor in Sakaniwa & Yamada Laboratory until March 2005, for fruitful discussions, support, and his invaluable help often needed to overcome many, unsolvable for me, computer network puzzles. I can never forget to express my hearty thanks to Dr. Konstantinos Slavakis of University of Athens, Greece, who had been a JSPS Research Fellow in Sakaniwa & Yamada Laboratory since April 2004 until March 2006, for warm friendship and helpful advises.

I feel thankful for having the luck of accepting the friendship and helpful advises of Dr. Kenta Kasai and Ms. Hiroko Ohta. My deep thanks are also addressed to all the students of Sakaniwa & Yamada Laboratory, whom I met these years, for providing with a fun-to-work environment.

I would like also to deeply thank Professors Yoshinori Sakai, Hiroshi Suzuki, and Tomohiko Uyematsu of Tokyo Institute of Technology, and an Associate Professor Kazuhiko Fukawa of Tokyo Institute of Technology for serving as members

of the examining committee of this thesis.

Contents

List of Figures

Abstract

This thesis presents efficient techniques to achieve the following objective: develop an *efficient* adaptive filtering algorithm that enjoys (i) fast and stable convergence and (ii) good steady-state performance with (iii) low computational complexity [more precisely, linear computational complexity]. The term 'stable' here stands for robustness against corrupted noise and/or possible nonstationarity of environments. The efficacy of the proposed techniques is verified in applications to acoustic and communication systems.

Firstly, a family of efficient linearly-constrained adaptive filtering algorithms is presented; the family is named *Adaptive Parallel Constrained Projection (A-PCP) method*. Two efficient blind adaptive algorithms, which belong to A-PCP, are proposed for Multiple Access Interference (MAI) suppression in DS/CDMA wireless communication systems. The proposed algorithms utilize rough estimates of an amplitude and transmitted bits of a desired user. Due to the *embedded constraint* and *parallel structure*, the proposed algorithms realize fast and stable convergence and low bit error rate at steady-state with linear order complexity.

Secondly, *Pairwise Optimal Weight Realization* (POWER), an efficient adaptive weighting technique for the adaptive-PSP algorithm, is presented. The POWER technique employs the projection onto the intersection of two closed half-spaces as a basic tool for an efficient approximation of an ideal direction of update. The weights realized by POWER turn out to be optimal in the sense of (i) pairwise and (ii) worst-case optimization. The proposed algorithm enjoys fast and stable convergence and good steady-state performance while keeping linear complexity.

Thirdly, an efficient Stereophonic Acoustic Echo Cancellation (SAEC) scheme is presented, based on two key ideas. The first idea is simultaneous use of data

from multiple input-states provided by preprocessing. The second idea is to employ POWER in an efficient manner for further acceleration while keeping linear computational complexity. In fact, the POWER technique turns out to exert far-reaching effects in the SAEC problem. The proposed technique exhibits excellent convergence and tracking behavior after a change of the echo paths in extensive simulations.

Finally, a family of very flexible adaptive algorithms based on quadratic-metric is presented. Following two adaptive algorithms with metrics constant in time, an algorithm with a variable metric is presented. One of the constant-metric algorithms is based on the parallel projection onto data-dependent closed convex sets, thus it is named *Adaptive Parallel Quadratic-metric Projection (APQP) algorithm*. The other constant-metric algorithm selectively utilizes critical ones among those convex sets based on a simple min-max criterion, thus it is named *Adaptive Parallel Min-max Quadratic-metric Projection (APMQP) algorithm*. The variable-metric algorithm is named *Adaptive Parallel Variable-metric Projection (APVP) algorithm*, being a natural extension of APQP. The efficacy of the proposed algorithms is verified in the acoustic echo canceling application.

The techniques developed in this thesis are based on the framework of *Adaptive Projected Subgradient Method (APSM)* [Yamada *et al.*, 2003], which generates a solution (a sequence of adaptive filtering vectors) to the following formulation: minimize asymptotically a sequence of nonnegative convex objective functions over a closed convex subset of a real Hilbert space. APSM has been proven to have the following properties: monotone approximation, asymptotic optimality, strong convergence etc. The proposed techniques are of course endowed with those remarkable properties. The consequence of this thesis supports the applicability of APSM to real-world signal processing problems.

Chapter 1

General Introduction

My Ph.D. study has been devoted to pursue an *efficient* adaptive filtering algorithm that enjoys (i) fast and stable convergence and (ii) good steady-state performance with (iii) low computational complexity [more precisely, $O(N)$, where N is the length of adaptive filter]. The term 'stable' here stands for robustness against corrupted noise and/or possible nonstationarity of environments. This study yields various techniques, to achieve the above goal, of which the efficacy is verified in applications to acoustic and communication systems. This short chapter provides with an outline of the fruits of this study. Let us first look back briefly at the history of adaptive filtering algorithms developed over the last half a century (See, e.g., [56, 97] for more details).

1.1 RLS and LMS

In the middle of 20 century, two classical adaptive filtering algorithms have been invented: the *Recursive Least Squares (RLS)* algorithm[1] by Plackett (1950) [91] and the *Least Mean Square (LMS)* algorithm by Widrow and Hoff (1960) [122]. Although it has widely been adopted in commercial products because mainly of its stability and computational simplicity [$O(N)$], the LMS algorithm suffers from slow convergence. On the other hand, the RLS algorithm has relatively fast convergence property and is insensitive to the variation of eigenvalue spread of the

[1] Although the original work of the RLS algorithm is often credited to Plackett in modern times, it is mentioned in some literature that Gauss in the late of 18 century had already formulated the recursive least squares solution (see e.g., [97, § 11.7, § 12.6]).

auto-correlation matrix of input signal (unlike LMS). However, RLS often suffers from instability especially for nonstationary input signal (see, e.g., [53], [56, §9.9]) and requires high computational complexity [$O(N^2)$]. The *regularization* technique [114], also called *diagonal loading* in adaptive beamforming [19], alleviates the instability issue at the cost of $O(N^3)$ computational complexity. To reduce the computational complexity of RLS, simplified versions, called Fast-RLS (FRLS), have been proposed by Ljung, Morf, and Falconer (1978), Carayannis, Manolakis, and Kalouptsidis (1983), and Cioffi and Kailath (1984) [18, 23, 72]. It has been pointed out that FRLS suffers from intrinsic instability (see [50] and the references therein). A great deal of effort has been devoted to stabilize FRLS with a moderate increase of computational complexity [7, 14, 40, 82, 83, 105]. In particular, the Fast Newton Transversal Filters (FNTF), proposed by Moustakides and Theodoridis (1991) [83], is a class of efficient adaptive estimation algorithms, being an efficient extension of the method in [18]. Unfortunately, even if we employ the existing stabilized versions, the filter has a risk of divergence, and thus, monitoring and reinitialization are always required [11, p. 77], [14, 82, 105], [42, p. 40]. Numerical instability still remains as a common issue to overcome in the RLS-type algorithms. One of the other remaining issues in RLS is that there are situations where RLS has tracking inferiority to LMS [35, 57], [97, pp. 383]. To overcome these issues, adaptive controlling techniques of *forgetting factor*, a parameter governing the estimates of statistics, have been studied (see, e.g., [36, 71, 104, 106, 109, 115]). Nevertheless, there still remain some issues due mainly to its dependency on estimates of statistical information. The previous extensive studies on RLS suggest that the simple LMS-type algorithms (which relies *not* on such statistics *but* just on instantaneous measurements) will be playing a central roll in nonstationary signal processing applications. Let us now switch our focus to the LMS-type algorithms.

The LMS algorithm iteratively generates the sequence of adaptive filtering vector $(\boldsymbol{h}_k)_{k\in\mathbb{N}} \subset \mathbb{R}^N$ by the following recursion (k: iteration number): $\boldsymbol{h}_{k+1} := \boldsymbol{h}_k - \lambda_k e_k(\boldsymbol{h}_k)\boldsymbol{u}_k$, where λ_k is a small positive constant called *step size*, $\boldsymbol{u}_k \in \mathbb{R}^N$ the input vector, and $e_k(\boldsymbol{h}) := \boldsymbol{u}_k^T\boldsymbol{h} - d_k \in \mathbb{R}$, $\forall \boldsymbol{h} \in \mathbb{R}^N$, with the observable output data $d_k \in \mathbb{R}$. Since the amount of update depends on input energy (i.e., $\|\boldsymbol{u}_k\|$), a very small step size should be used to prevent divergence, resulting in very slow convergence. To improve the convergence speed, the Normalized LMS (NLMS) algorithm has been proposed by Nagumo and Noda (1967) [84], Albert and Gardner

(1967) [1], where the amount of update is normalized as $\boldsymbol{h}_{k+1} := \boldsymbol{h}_k - \lambda_k \frac{e_k(\boldsymbol{h}_k)\boldsymbol{u}_k}{\|\boldsymbol{u}_k\|^2}$, $\lambda_k \in [0, 2]$ (Other ways of normalization have also been proposed; see [97] and the references therein). This simple modification not only allows us to use larger step size but also gives us an interesting geometric interpretation in the space of filtering vector. Namely, the NLMS algorithm can be interpreted as an *iterative relaxed projection* (exact metric projection if $\lambda_k = 1$) onto data-dependent hyperplanes on which the output error is zero. Mathematically, the update equation of NLMS can be rewritten as $\boldsymbol{h}_{k+1} := \boldsymbol{h}_k + \lambda_k [P_{H_k}(\boldsymbol{h}_k) - \boldsymbol{h}_k]$, where $P_{H_k}(\boldsymbol{h}_k)$ denotes the projection of $\boldsymbol{h}_k$ onto the hyperplane $H_k := \{\boldsymbol{h} \in \mathbb{R}^N : e_k(\boldsymbol{h}) = 0\}$. Thus it is categorized into a family of projection-based algorithms. However, when the auto-correlation matrix $E(\boldsymbol{u}_k \boldsymbol{u}_k^T)$ (E: expectation, superscript T: transposition) has spreading eigenvalues (i.e., the input signal is highly colored), the normal vectors of the hyperplanes which are defined with data obtained within a short period are close to each other. This causes slow convergence of NLMS for highly colored input signals [15, 50], although NLMS is robust against noise (which is supported by the H^∞ theory [54, 55]).

To raise the convergence speed of NLMS, the *Affine Projection Algorithm (APA)* has been proposed by Hinamoto[2] and Maekawa (1975) [58], Ozeki and Umeda (1984) [89]. The basic idea is to replace the hyperplane H_k by the linear variety $V_k := \bigcap_{i=k-r+1}^{k} H_i$, where $(0 <) r \in \mathbb{N}$ is recently called *data reusing factor*. The update equation of APA can be written as $\boldsymbol{h}_{k+1} := \boldsymbol{h}_k + \lambda_k [P_{V_k}(\boldsymbol{h}_k) - \boldsymbol{h}_k]$. To analyze the convergence property of APA, a lot of effort has been done (see e.g., [93, 95, 103, 116]). In noiseless situations, the vector sequence generated by APA monotonically approaches (by Pythagorean theorem) an estimandum $\boldsymbol{h}^* (\in V_k)$, a system to be estimated. In practically noisy situations, however, probability that $\boldsymbol{h}^*$ belongs to V_k is approximately zero for $r \geq 3$, which clearly explains the noise sensitivity of APA [129]. To reduce the computational complexity, a simplified version, called *Fast Affine Projection (FAP)* algorithm, has been proposed by Gay and Tavathia (1995) [43]. Unfortunately, however, FAP keeps the noise sensitivity problem unsettled. All the aforementioned algorithms utilize *linear* operators as their basic tools.

[2]Although, in most literature, APA is credited to Ozeki and Umeda, the original idea has been given by Hinamoto and Maekawa [58].

The history gives us a question: "Is the *linearity* necessary to deal with challenging problems in adaptive filtering?". In this study, *convexity*, an important special case of nonlinearity, is a key rather than linearity.

1.2 This Study

The techniques developed in this study are based on the framework of *Adaptive Projected Subgradient Method (APSM)* proposed by Yamada *et al.* (2003) [124, 126], in which a fairly new formulation has been established: minimize asymptotically a sequence of nonnegative convex objective functions $(\Theta_k)_{k\in\mathbb{N}}$ over a closed convex subset C of a real Hilbert space $\mathcal{H}$. APSM reproduces various projection-based algorithms such as NLMS [1, 84], APA [58, 89], set-membership NLMS/APA [51, 121], constrained NLMS/APA [27, 28, 67] (in the *embedded* sense [124, 126]), adaptive Parallel Subgradient Projection (adaptive-PSP) [129, 135], embedded-constraint parallel projection [132] algorithms. Thus, APSM, as a side effect, provides a substantial framework to understand a wide range of set-theoretic adaptive filtering algorithms in a unified fashion. Chapter 2 provides notation employed throughout this thesis, and briefly introduces APSM with a significant extension by Slavakis *et al.* [99, 100].

Chapter 3 presents a family of novel linearly-constrained adaptive filtering algorithms; the family is named *Adaptive Parallel Constrained Projection (A-PCP) method*. Two efficient blind adaptive algorithms, which belong to A-PCP, are proposed for Multiple Access Interference (MAI) suppression in DS/CDMA wireless communication systems. Due to the *embedded constraint* and the *parallel structure*, the proposed algorithms realize fast and stable convergence and low Bit Error Rate (BER) at steady-state with linear order complexity. The proposed algorithms utilize rough estimates of an amplitude and transmitted bits of a desired user. To clarify the advantage of the proposed algorithms over the conventional methods, a geometric interpretation as well as simulation results is provided. The simulation results demonstrate that the proposed algorithms attain approximately 10 times faster convergence than the conventional blind algorithms [67, 90] and, simultaneously, BER performance close to the (suboptimal) linear filter.

Chapter 4 presents an inventive idea for further acceleration of an efficient

adaptive filtering algorithm, named *the adaptive-PSP algorithm* [129], by optimizing the weighting parameters in a *pairwise* fashion, while keeping its computational efficiency. Adaptive-PSP utilizes, at each iteration, multiple, say $q \in \mathbb{N} \setminus \{0\}$, closed half-spaces $H^-_{\iota_1}(\boldsymbol{h}_k), \cdots, H^-_{\iota_q}(\boldsymbol{h}_k)$ $(\iota_j \leq k,\ \forall j = 1, 2, \cdots, q)$ that contain $\boldsymbol{h}^*$ with high probability even in noisy situations. Hence, monotonicity is highly expected to hold, opening a pathway to resolution of the noise sensitivity problem. Uniform weights have commonly been used in adaptive-PSP mainly for simplicity, which suggests that further improvements of convergence could be achieved by establishing an effective weight design. To bring out the potential of the adaptive-PSP algorithm aggressively but without losing its computational efficiency, an efficient adaptive weighting technique named *Pairwise Optimal Weight Realization* (POWER) is proposed. The POWER technique employs the projection onto the intersection of two closed half-spaces as a basic tool for an efficient approximation of an ideal direction of update. The weights realized by POWER turn out to be optimal in the sense of (i) pairwise and (ii) maximum minimization. The proposed algorithm enjoys fast and stable convergence and good steady-state performance while keeping linear complexity.

Chapter 5 presents a class of efficient fast adaptive filtering algorithms for Stereophonic Acoustic Echo Cancellation (SAEC) problem, which is a central issue when we design high-quality, hands-free and full-duplex systems (e.g., advanced teleconferencing etc.) [8, 11, 13, 16, 34, 37, 42, 44, 45, 65, 107, 108, 111]. The SAEC problem has received significant attention because there exists a structural problem caused by high cross-correlation between two input signals observed at two microphones. To decorrelate the input signals, a great deal of effort has been devoted to devise preprocessing of the inputs [2, 12, 13, 46, 47, 49, 60, 66, 98, 108, 112]. The remaining major challenges in SAEC with preprocessing are twofold: (i) fast tracking of the echo paths and (ii) low computational complexity due to the necessity to adapt 4 echo cancelers with a few thousands taps [42]. The first idea is simultaneous use of data from multiple input-states provided by preprocessing [138]. The second idea is to employ POWER in an efficient manner for further acceleration while keeping linear computational complexity [134]. In fact, the POWER technique turns out to exert far-reaching effects in the SAEC problem. The proposed technique exhibits excellent convergence and tracking behavior after a change of the echo paths in the extensive simulations.

Chapter 6 presents a family of very flexible adaptive algorithms based on quadratic-metric. First, two adaptive algorithms in which the metric is constant in time are presented. One is based on the parallel projection onto data-dependent closed convex sets, thus it is named *Adaptive Parallel Quadratic-metric Projection (APQP) algorithm*. The other selectively utilizes critical ones among those convex sets based on a simple min-max criterion, thus it is named *Adaptive Parallel Min-max Quadratic-metric Projection (APMQP) algorithm*. These two algorithms are naturally derived by employing quadratic-norms in the APSM. Secondly, although the 'constancy' in the metric design is crucial to ensure important properties such as asymptotic optimality, a more general form of APQP where the metric itself changes in time is presented. In other words, the algorithm is based on variable-metric, thus it is named *Adaptive Parallel Variable-metric Projection (APVP) algorithm*. The proposed algorithms (APQP/APMQP/APVP) has the valuable monotone property. By employing an efficient metric, the overall computational complexity of the proposed algorithms is kept linear with respect to (w.r.t.) the filter length. The efficacy of the proposed algorithms is verified by simulations in the acoustic echo canceling application.

Chapter 7 concludes this thesis with some remarks.

Chapter 2

Preliminaries

Throughout the thesis, the following notation is used. Let $\mathbb{R}$, $\mathbb{N}$, and $\mathbb{N}^*$ denote the sets of all real numbers, nonnegative integers, and positive integers, respectively. Vectors (matrices) are represented by bold-faced lower-case (upper-case) letters. The identity and zero matrices are denoted by $\boldsymbol{I}$ and $\boldsymbol{O}$, respectively. For any matrix $\boldsymbol{A}$, the $\boldsymbol{A}^T$ stands for the transposition of $\boldsymbol{A}$. A real Hilbert space $\mathcal{H}$ equipped with an inner product $\langle \cdot, \cdot \rangle$ will be denoted by $(\mathcal{H}, \langle \cdot, \cdot \rangle)$. Its induced norm is given by $\|\boldsymbol{x}\| := \langle \boldsymbol{x}, \boldsymbol{x} \rangle^{1/2}$, $\forall \boldsymbol{x} \in \mathcal{H}$. The Euclidean space $\mathbb{R}^N$ $(N \in \mathbb{N}^*)$ is the simplest finite-dimensional Hilbert space, which is often the stage in practical applications. The notation $|S|$ stands for the cardinality of a set S.

Given $\boldsymbol{v} \in \mathcal{H}$ ($\mathcal{H}$: real Hilbert space) and a closed subspace $M \subset \mathcal{H}$, the translation of M by $\boldsymbol{v}$ defines the *linear variety* $V := \boldsymbol{v} + M := \{\boldsymbol{v} + \boldsymbol{m} : \boldsymbol{m} \in M\}$. If $\dim(M^\perp) = 1$, V is called a *hyperplane*, which can be expressed as $(V =)\Pi = \{\boldsymbol{x} \in \mathcal{H} : \langle \boldsymbol{a}, \boldsymbol{x} \rangle = c\}$ for some $(\boldsymbol{0} \neq)\boldsymbol{a} \in \mathcal{H}$, and $c \in \mathbb{R}$. Here, for any linear subspace $M \subset \mathcal{H}$, $M^\perp \subset \mathcal{H}$ is defined as $M^\perp := \{\boldsymbol{x} \in \mathcal{H} : \langle \boldsymbol{x}, \boldsymbol{m} \rangle = 0,\ \forall \boldsymbol{m} \in M\}$. $\Pi^- := \{\boldsymbol{x} \in \mathcal{H} : \langle \boldsymbol{a}, \boldsymbol{x} \rangle \leq c\}$ is called a *closed half-space* with its boundary Π.

A set $C \subset \mathcal{H}$ is said to be *convex* if $\nu\boldsymbol{x} + (1-\nu)\boldsymbol{y} \in C$, $\forall \boldsymbol{x}, \boldsymbol{y} \in C$, $\forall \nu \in (0, 1)$. A function $\Theta : \mathcal{H} \to \mathbb{R}$ is said to be *convex* if $\Theta(\nu\boldsymbol{x} + (1-\nu)\boldsymbol{y}) \leq \nu\Theta(\boldsymbol{x}) + (1-\nu)\Theta(\boldsymbol{y})$, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{H}$, $\forall \nu \in (0, 1)$.

Given a mapping $T : \mathcal{H} \to \mathcal{H}$, $\mathrm{Fix}(T) := \{\boldsymbol{y} \in \mathcal{H} : T(\boldsymbol{y}) = \boldsymbol{y}\}$ is called the fixed point set of T. A mapping T is said to be *nonexpansive* if $\|T(\boldsymbol{x}) - T(\boldsymbol{y})\| \leq \|\boldsymbol{x} - \boldsymbol{y}\|$, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{H}$. If, in addition, $\mathrm{Fix}(T) \neq \emptyset$ and there exists $\eta > 0$ such that (s.t.) $\eta \|\boldsymbol{x} - T(\boldsymbol{x})\|^2 \leq \|\boldsymbol{x} - \boldsymbol{f}\|^2 - \|T(\boldsymbol{x}) - \boldsymbol{f}\|^2$, $\forall \boldsymbol{x} \in \mathcal{H}$, $\forall \boldsymbol{f} \in \mathrm{Fix}(T)$, then T is said to be *strongly* or *η-attracting nonexpansive*. The identity mapping

$I : \mathcal{H} \to \mathcal{H}$, $\boldsymbol{x} \mapsto \boldsymbol{x}$, can be considered as an η-attracting nonexpansive mapping for an arbitrary $\eta > 0$ with $\mathrm{Fix}(I) = \mathcal{H}$. Moreover, a mapping $T : \mathcal{H} \to \mathcal{H}$ is called α-averaged nonexpansive if there exist $\alpha \in [0, 1)$ and a nonexpansive mapping $\mathcal{N} : \mathcal{H} \to \mathcal{H}$ s.t. $T = (1 - \alpha)I + \alpha\mathcal{N}$.

Fact 2.0.1 (Selected properties of nonexpansive mapping [127]).

(a) *If a nonexpansive mapping $T : \mathcal{H} \to \mathcal{H}$ has at least one fixed point,* $\mathrm{Fix}(T) \subset \mathcal{H}$ *is closed convex.*

(b) *Let $\alpha > 0$. Then, $T : \mathcal{H} \to \mathcal{H}$ is α-averaged nonexpansive iff (if and only if) T is $\dfrac{1-\alpha}{\alpha}$-attracting nonexpansive.*

(c) *Let $T_1 : \mathcal{H} \to \mathcal{H}$ and $T_2 : \mathcal{H} \to \mathcal{H}$ be nonexpansive with* $\mathrm{Fix}(T_1) \cap \mathrm{Fix}(T_2) \neq \emptyset$. *Then, for $\alpha \in (0, 1)$, (i) $(1 - \alpha)T_1 + \alpha T_2$ is nonexpansive with* $\mathrm{Fix}((1 - \alpha)T_1 + \alpha T_2) = \mathrm{Fix}(T_1) \cap \mathrm{Fix}(T_2)$. *In addition, (ii) if either T_1 or T_2 is attracting nonexpansive, then $(1-\alpha)T_1+\alpha T_2$ is also attracting nonexpansive. Moreover, (iii) if T_1 is η_1-attracting nonexpansive and T_2 is η_2-attracting nonexpansive, then $(1-\alpha)T_1+\alpha T_2$ is $\left(\dfrac{(\eta_1+1)(\eta_2+1)}{(1-\alpha)\eta_2+\alpha\eta_1+1} - 1\right)$-attracting nonexpansive.*

(d) *Let $T_1 : \mathcal{H} \to \mathcal{H}$ be nonexpansive and $T_2 : \mathcal{H} \to \mathcal{H}$ attracting nonexpansive with* $\mathrm{Fix}(T_1) \cap \mathrm{Fix}(T_2) \neq \emptyset$. *Then, (i) $T_2T_1 : \mathcal{H} \to \mathcal{H}$ is nonexpansive with* $\mathrm{Fix}(T_2T_1) = \mathrm{Fix}(T_1) \cap \mathrm{Fix}(T_2)$. *In addition, (ii) if $T_1 : \mathcal{H} \to \mathcal{H}$ is also attracting nonexpansive, then T_2T_1 is attracting nonexpansive. Moreover, (iii) if T_1 is η_1-attracting nonexpansive and T_2 is η_2-attracting nonexpansive, then T_2T_1 is $\dfrac{\eta_1\eta_2}{\eta_1+\eta_2}$-attracting nonexpansive.*

(e) *Let $T : \mathcal{H} \to \mathcal{H}$ be 1-attracting nonexpansive, which is also called firmly nonexpansive, with* $\mathrm{Fix}(T) \neq \emptyset$. *Then, for $\mu \in (0, 2)$, $(1 - \mu)I + \mu T : \mathcal{H} \to \mathcal{H}$ is $\dfrac{2-\mu}{\mu}$-attracting nonexpansive with* $\mathrm{Fix}((1 - \mu)I + \mu T) = \mathrm{Fix}(T)$. *Moreover, T is firmly nonexpansive (iff T is $\dfrac{1}{2}$-averaged nonexpansive) iff $2T - I$ is nonexpansive.*

Note that Fact 2.0.1 also holds for a wider class of mappings called *quasi-nonexpansive*; a mapping $T : \mathcal{H} \to \mathcal{H}$ with $\mathrm{Fix}(T) \neq \emptyset$ is called *quasi-nonexpansive* if $\|T(\boldsymbol{x}) - T(\boldsymbol{f})\| \leq \|\boldsymbol{x} - \boldsymbol{f}\|$ for all $(\boldsymbol{x}, \boldsymbol{f}) \in \mathcal{H} \times \mathrm{Fix}(T)$.

Given a nonempty closed convex set $C \subset \mathcal{H}$, the mapping that assigns every point in $\mathcal{H}$ to its unique nearest point in C is called *metric projection* onto C and is denoted by P_C. Mathematically, one can state $P_C : \mathcal{H} \to C$, $\boldsymbol{x} \mapsto P_C(\boldsymbol{x}) \in \arg\inf_{\boldsymbol{y}\in C} \|\boldsymbol{x} - \boldsymbol{y}\|$. P_C has the following properties: $\mathrm{Fix}(P_C) = C$; P_C is 1-attracting nonexpansive; and $\|\boldsymbol{x} - P_C(\boldsymbol{x})\| = \mathrm{d}(\boldsymbol{x}, C) := \inf_{\boldsymbol{y}\in C} \|\boldsymbol{x} - \boldsymbol{y}\|$, $\forall \boldsymbol{x} \in \mathcal{H}$.

Given a continuous convex function $\Theta : \mathcal{H} \to \mathbb{R}$, the *subdifferential* of Θ at any $\boldsymbol{y} \in \mathcal{H}$, the set of all the *subgradients* of Θ at $\boldsymbol{y}$; $\partial\Theta(\boldsymbol{y}) := \{\boldsymbol{a} \in \mathcal{H} : \langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{a}\rangle + \Theta(\boldsymbol{y}) \leq \Theta(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{H}\}$, is nonempty. Given a continuous convex function $\Theta : \mathcal{H} \to \mathbb{R}$, suppose that there exists $\boldsymbol{x} \in \mathcal{H}$ s.t. $\Theta(\boldsymbol{x}) \leq 0$. Using one of its subgradients $\Theta' : \mathcal{H} \to \mathcal{H}$ (i.e., $\Theta'(\boldsymbol{x}) \in \partial\Theta(\boldsymbol{x})$, $\forall \boldsymbol{x} \in \mathcal{H}$), define a mapping $T_{\mathrm{sp}(\Theta)} : \mathcal{H} \to \mathcal{H}$ by

$$T_{\mathrm{sp}(\Theta)} : \boldsymbol{x} \mapsto \begin{cases} \boldsymbol{x} - \dfrac{\Theta(\boldsymbol{x})}{\|\Theta'(\boldsymbol{x})\|^2}\Theta'(\boldsymbol{x}) & \text{if } \Theta(\boldsymbol{x}) > 0, \\ \boldsymbol{x} & \text{otherwise.} \end{cases}$$

Then the mapping $T_{\mathrm{sp}(\Theta)}$ is called a subgradient projection relative to Θ, where $\mathrm{Fix}(T_{\mathrm{sp}(\Theta)}) = \mathrm{lev}_{\leq}\Theta := \{\boldsymbol{x} \in \mathcal{H} : \Theta(\boldsymbol{x}) \leq 0\}$ [5, 117, 127]. Note that $T_{\mathrm{sp}(\Theta)}$ is *not* nonexpansive *but* quasi-nonexpansive [5, 117, 127].

Let $\Theta_k : \mathcal{H} \to [0, \infty)$, $k \in \mathbb{N}$, be a continuous convex function and $\partial\Theta_k(\boldsymbol{y})$ the subdifferential of Θ_k at $\boldsymbol{y}$. Also let $T : \mathcal{H} \to \mathcal{H}$ denote an η-attracting nonexpansive mapping. Define a mapping $\Phi_k : \mathcal{H} \to \mathcal{H}$, $\forall k \in \mathbb{N}$, as $\Phi_k := T\left[(1-\lambda_k)I + \lambda_k T_{\mathrm{sp}(\Theta_k)}\right]$. The following scheme, an extension of the scheme in [124, 126], provides a vector sequence that minimizes asymptotically the sequence of objective functions $(\Theta_k)_{k\in\mathbb{N}}$ over $\mathrm{Fix}(T)$.

Scheme 2.0.2 (Extended Adaptive Projected Subgradient Method [99, 100])**.** *For an arbitrarily given $\boldsymbol{h}_0 \in \mathcal{H}$, generate a sequence $(\boldsymbol{h}_k)_{k\in\mathbb{N}} \subset \mathcal{H}$ by*

$$\boldsymbol{h}_{k+1} := \begin{cases} \Phi_k(\boldsymbol{h}_k) = T\left[\boldsymbol{h}_k - \lambda_k \dfrac{\Theta_k(\boldsymbol{h}_k)}{\|\Theta_k'(\boldsymbol{h}_k)\|^2}\Theta_k'(\boldsymbol{h}_k)\right] & \textit{if } \Theta_k'(\boldsymbol{h}_k) \neq \boldsymbol{0}, \\ T(\boldsymbol{h}_k) & \textit{otherwise,} \end{cases}$$

where $\Theta_k'(\boldsymbol{h}_k) \in \partial\Theta_k(\boldsymbol{h}_k)$, $\lambda_k \in [0, 2]$, $\forall k \in \mathbb{N}$, and $\boldsymbol{0}$ denotes the zero vector.

Replacing T with a metric projection operator, Scheme 2.0.2 is reduced to the original APSM [124, 126].

Fact 2.0.3 (Selected properties of extended APSM [99, 100]). *Define* $\Omega_k := \left\{ \boldsymbol{h} \in \mathrm{Fix}(T) : \Theta_k(\boldsymbol{h}) = \Theta_k^* := \inf_{\boldsymbol{x} \in \mathrm{Fix}(T)} \Theta_k(\boldsymbol{x}) \right\}$, $\forall k \in \mathbb{N}$. *The sequence* $(\boldsymbol{h}_k)_{k \in \mathbb{N}}$ *generated by Scheme* 2.0.2 *enjoys the following properties.*

(a) *(Monotone Approximation) Assume the following conditions:*

1. $\exists k_0 \in \mathbb{N}$ *s.t. (i)* $\Omega := \bigcap_{k \geq k_0} \Omega_k \neq \emptyset$ *and (ii)* $\Theta_k^* = 0$, $\forall k \geq k_0$, *and*
2. $\lambda_k \in (0, 2)$, $\forall k \geq k_0$.

   Then,

$$\left\| \boldsymbol{h}_{k+1} - \boldsymbol{h}^{*(k)} \right\| \leq \left\| \boldsymbol{h}_k - \boldsymbol{h}^{*(k)} \right\|, \ \forall \boldsymbol{h}^{*(k)} \in \Omega_k, \forall k \geq k_0.$$

   In addition, assume that

3. $\boldsymbol{h}_k \notin \Omega_k$ *for* $k \geq k_0$.

   Then,

$$\left\| \boldsymbol{h}_{k+1} - \boldsymbol{h}^{*(k)} \right\| < \left\| \boldsymbol{h}_k - \boldsymbol{h}^{*(k)} \right\|, \ \forall \boldsymbol{h}^{*(k)} \in \Omega_k.$$

(b) *(Strong convergence) Assume* 2.0.3.a.1 *and the following:*

1. $\exists \epsilon_1, \epsilon_2 > 0$ *s.t.* $\lambda_k \in [\epsilon_1, 2 - \epsilon_2]$, $\forall k \geq k_0$, *and*
2. *There exists a hyperplane* $\Pi \subset \mathcal{H}$ *s.t.* $\mathrm{ri}_\Pi(\Omega) \neq \emptyset$, *where* $\mathrm{ri}_\Pi(\Omega) := \{\boldsymbol{h} \in \Omega : \exists \epsilon > 0 \text{ s.t. } B(\boldsymbol{h}, \epsilon) \cap \Pi \subset \Omega\}$ *is the relative interior of* Ω *w.r.t.* Π. *Here,* $B(\boldsymbol{h}, \epsilon) := \{\boldsymbol{x} \in \mathcal{H} : \|\boldsymbol{x} - \boldsymbol{h}\| < \epsilon\}$ *is an open ball.*

Then, the sequence $(\boldsymbol{h}_k)_{k \in \mathbb{N}}$ *converges (strongly) to a point* $\widehat{\boldsymbol{h}} \in \mathrm{Fix}(T)$.

(c) *(Asymptotic optimality) Assume* 2.0.3.a.1, 2.0.3.b.1, *and*

1. $(\Theta_k'(\boldsymbol{h}_k))_{k \in \mathbb{N}}$ *is bounded.*

   Then, $\lim_{k \to \infty} \Theta_k(\boldsymbol{h}_k) = 0$. *Assume also* 2.0.3.b.2, *and*
2. $(\Theta_k'(\widehat{\boldsymbol{h}}))_{k \in \mathbb{N}}$ *is bounded.*

   Then, $\lim_{k \to \infty} \Theta_k(\widehat{\boldsymbol{h}}) = 0$.

Note that the Θ_k'*s appearing in the remaining of this thesis are automatically bounded* [126, Proposition 3-(a)].

Chapter 3

Efficient Blind MAI Suppression in DS/CDMA Systems by Embedded Constraint Parallel Projection Techniques

Summary

This chapter presents two novel blind set-theoretic adaptive filtering algorithms for suppressing "Multiple Access Interference (MAI)", which is one of the central burdens in DS/CDMA systems. We naturally formulate the problem of MAI suppression as an asymptotic minimization of a sequence of cost functions under some linear constraint defined by the desired user's signature. The proposed algorithms embed the constraint into the direction of update, and thus the adaptive filter moves toward the optimal filter without stepping away from the constraint set. In addition, using parallel processors, the proposed algorithms attain excellent performance with linear computational complexity. Geometric interpretation clarifies an advantage of the proposed methods over existing methods. Simulation results demonstrate that the proposed algorithms achieve (i) much higher speed of convergence with rather better bit error rate performance than other blind methods and (ii) much higher speed of convergence than the non-blind NLMS algorithm (indeed, the speed of convergence of the proposed algorithms is comparable to the

non-blind RLS algorithm).

3.1 Introduction

The goal of this chapter is to develop a blind Multiple Access Interference (MAI) suppressing algorithm, being "efficient" in the sense of (i) low computational complexity and (ii) high speed of convergence, for Direct Sequence Code-Division Multiple-Access (DS/CDMA) systems.

One of the noticeable advantages of CDMA systems is that users can share time and frequency by exploiting distinct spreading codes, or, in other words, users can transmit their information symbols at the same time and frequency. CDMA receivers, on the other hand, are usually affected by interference originated from transmitted symbols of other users. This is commonly referred to as MAI and it is known to deteriorate the overall capacity. A great deal of effort has been devoted to MAI suppression [38, 39, 62, 63, 74–76, 79, 90, 92, 94, 118, 119].

To realize high throughput systems, blind methods for MAI suppression, which do not require a training sequence (or pilot signals), have been particularly in great demand [62, 63, 90, 92, 94, 119]. In 1995, Honig *et al.* proposed a blind adaptive multiuser detection method [62], in which the problem is formulated as a constrained optimization with a linear constraint defined by the desired user's signature. In 1997, Park and Doherty have proposed a simple set-theoretic blind method called *Space Alternating Generalized Projection (SAGP)* [90], which utilizes generalized projections onto non-convex sets (see Remark 3.3.4 and [110]). The SAGP exhibits better performance in the steady state at the expense of slower convergence rate than the method in [62]. In [88], it is reported that fast algorithms are necessary to keep good performance especially in wireless communications.

In 1998, Apolinário Jr. *et al.* have proposed the *Constrained Normalized Least Mean Square (CNLMS) algorithm* [67], which embeds the constraint used in [62] into the direction of update, providing fast convergence. Unfortunately, the CNLMS does not yet achieve sufficient speed of convergence because it takes just one datum into account at each iteration. In 2004, on the other hand, a fast blind MAI suppression method has been proposed [20], which we call *Blind Parallel Projection (B2P) algorithm*. The B2P developed the idea of the SAGP by using a certain *parallel structure* and *convexification*, leading to excellent performance.

The filter recursion (update) of the B2P is constructed by two steps at each iteration (cf. Remark 3.3.4): (i) shift the filter in descent directions of cost functions and (ii) enforce it in the constraint set.

This chapter presents two embedded constraint blind algorithms for an adaptive MAI suppression filter. *Embedded constraint* and *parallel structure* are the keys to realize fast convergence with linear order complexity (see Remarks 3.3.3 and 3.3.4). The proposed algorithms develop the idea of the CNLMS for acceleration of convergence by taking into account more than one datum with several parallel processors at each iteration. Actually, the algorithms are derived from a set-theoretic adaptive filtering scheme named *Adaptive Projected Subgradient Method (APSM)* [124–126] (see Chapter 2), which has been successfully applied to *the stereophonic acoustic echo cancellation problem* [133, 138] (see Chapter 5). Roughly speaking, the algorithms minimize asymptotically a sequence of cost functions that are defined by the received data at every sampling time. Each iteration is constructed by two stages as follows. The first stage of the algorithms estimates the amplitude of the transmitted signal (as in [90]) and the transmitted bits. By using these estimates and the constraint used in [62], closed convex sets called *stochastic property sets* [see (3.3.7) in Sec. 3.3] are newly designed and, based on the distances to these sets, a reasonable cost function is defined. The second stage updates the MAI suppression filter in a descent direction of the cost function. The proposed algorithms have no need to enforce the filter in the constraint set unlike the SAGP or the B2P, since the constraint is embedded into the direction of update; i.e., the filter does not step away from the constraint set. Geometric interpretation clarifies an advantage of the proposed algorithms over the CNLMS, the SAGP and the B2P algorithms (see Remark 3.3.4). Simulation results exemplify dramatical improvements expected by the geometric interpretation.

Preliminary versions of this chapter are presented in [130, 131].

3.2 System Model

A Binary Phase-Shift Keying (BPSK) short-code DS/CDMA system is briefly summarized below. The system model considered in this chapter is exactly the same as the one presented in [20, 90, 94]. Without loss of generality, assume that the desired user's signature $\boldsymbol{s}_1$ satisfies $\|\boldsymbol{s}_1\| = 1$ as in [90]. The received data

process $(\boldsymbol{r}[i])_{i\in\mathbb{N}} \subset \mathbb{R}^N$ (N: the length of signature) is

$$\boldsymbol{r}[i] = A_1 b_1[i]\boldsymbol{s}_1 + \sum_{l=2}^{L} A_l \bar{b}_l[i]\bar{\boldsymbol{s}}_l + \boldsymbol{n}[i], \ \forall i \in \mathbb{N}, \tag{3.2.1}$$

where

$$\begin{aligned}
&A_1 > 0 : \text{amplitude of the 1st (desired) user}\\
&b_1[i] \in \{-1, 1\} : i\text{th transmitted bit of the desired user}\\
&\boldsymbol{s}_1 \in \left\{-\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}\right\}^N : \text{signature of the desired user}\\
&\boldsymbol{n}[i] \in \mathbb{R}^N : i\text{th noise vector.}
\end{aligned}$$

Moreover, A_l $(2 \le l \le L)$ is the amplitude of the lth interference, and $\bar{b}_l[i]$ and $\bar{\boldsymbol{s}}_l$ are respectively the ith interfering symbol bit and the interfering vector generated by lth interfering user's parameters such as associated data bits and signature. In the presence of K users, the number of interferences $L-1$ can range from $K-1$ to $2(K-1)$, due to relative delays of the $K-1$ interfering users [76].

The problem addressed in this chapter is to suppress efficiently the MAI, $\sum_{l=2}^{L} A_l\ \bar{b}_l[i]\bar{\boldsymbol{s}}_l$ in (3.2.1), with a linear filter without amplifying the noise $\boldsymbol{n}[i]$ severely.

3.3 Proposed Embedded-Constraint Blind Adaptive Algorithms

This section provides two set-theoretic algorithms for adaptation of a blind MAI suppression filter $\boldsymbol{h}_k \in \mathbb{R}^N$, where $k \in \mathbb{N}$ denotes the iteration number. All available data for the adaptation are assumed to be the sequence of received vectors $(\boldsymbol{r}[i])_{i\in\mathbb{N}}$ and the desired user's signature $\boldsymbol{s}_1$ (NOTE: In the absence of Inter-Chip Interference (ICI), the signature coincides with the spreading code and may be readily available [63]).

3.3.1 Set Design

To avoid the *self-nulling*[1] (i.e., canceling the desired user's signals), the following constraint is commonly imposed on the filter (e.g., [62]):

$$\boldsymbol{h}_k \in C_s := \{\boldsymbol{h} \in \mathbb{R}^N : \langle \boldsymbol{h}, \boldsymbol{s}_1 \rangle = 1\},\ \forall k \in \mathbb{N}. \tag{3.3.1}$$

Actually, $\langle \boldsymbol{h}_k, \boldsymbol{s}_1 \rangle$ can be any positive constant, however, for simplicity, we let $\langle \boldsymbol{h}_k, \boldsymbol{s}_1 \rangle = 1$. For any $\boldsymbol{h}_k \in C_s$, $\forall i \in \mathbb{N}$,

$$\langle \boldsymbol{h}_k, \boldsymbol{r}[i] \rangle = A_1 b_1[i] + \sum_{l=2}^{L} A_l \bar{b}_l[i] \langle \boldsymbol{h}_k, \bar{\boldsymbol{s}}_l \rangle + \langle \boldsymbol{h}_k, \boldsymbol{n}[i] \rangle . \tag{3.3.2}$$

For suppressing the MAI without amplifying noise severely, the second and third terms on the right side of (3.3.2) should be reduced as much as possible. Thus, a Minimum Mean-Squared Error (MMSE) filter is defined as follows [119]:

$$\boldsymbol{h}^* \in \operatorname*{argmin}_{\boldsymbol{h} \in C_s} E\left\{ (\langle \boldsymbol{h}, \boldsymbol{r}[i] \rangle - A_1 b_1[i])^2 \right\}, \tag{3.3.3}$$

where $E\{\cdot\}$ denotes the expectation; see the appendix for the relationship between the MMSE and the Minimum Output Energy (MOE) optimal filters. Since A_1 and $b_1[i]$ in (3.3.3) are not available, we use the following estimates [90]:

$$\widehat{A}_{1,k+1} := \widehat{A}_{1,k} + \gamma \left(|\langle \boldsymbol{h}_k, \boldsymbol{r}[k] \rangle| - \widehat{A}_{1,k} \right),\ \forall k \in \mathbb{N}, \tag{3.3.4}$$

$$\widehat{b}_{1,k}[i] := \operatorname{sgn} \langle \boldsymbol{h}_k, \boldsymbol{r}[i] \rangle ,\ \forall k \in \mathbb{N}, \tag{3.3.5}$$

where $\widehat{A}_{1,k}$ ($\widehat{A}_{1,0} = 0$) and $\widehat{b}_{1,k}[i]$ are respectively estimates of the amplitude A_1 and the ith transmitted bit $b_1[i]$ at iteration number k, and $\gamma \in (0, 1]$ is the *forgetting factor*; see Remark 3.3.5. For simplicity, we define the signum function sgn: $\mathbb{R} \to \{-1, 1\}$ as, if $a > 0$, sgn $a = 1$, otherwise, sgn $a = -1$ ($\forall a \in \mathbb{R}$). With the estimates in (3.3.4) and (3.3.5), the problem is reformulated as finding a point

[1] In the case when the amplitude of some interference is greater than that of a desired user, the filter may track not the desired user but the interference. In such a case, the desired user's signal is suppressed. The set C_s can avoid such a situation.

in

$$\underset{\boldsymbol{h}\in C_s}{\operatorname{argmin}} E\left\{\left(\langle \boldsymbol{h}, \boldsymbol{r}[i]\rangle - \widehat{A}_{1,k+1}\widehat{b}_{1,k}[i]\right)^2\right\}. \tag{3.3.6}$$

Instead of the expectation in (3.3.6), we newly introduce the following stochastic property sets [cf. Remark 3.A.2-(d)]:

$$\begin{aligned} C_\rho^{(k)}[i] := \left\{\boldsymbol{h}\in\mathbb{R}^N : \left(\langle \boldsymbol{h}, \boldsymbol{r}[i]\rangle - \widehat{A}_{1,k+1}\widehat{b}_{1,k}[i]\right)^2 \le \rho\right\}, \\ \forall k\in\mathbb{N},\ \forall i\in\mathcal{I}_k := \{k, k-1, \cdots, k-q+1\}, \end{aligned} \tag{3.3.7}$$

where $\mathcal{I}_k$ is the so-called *control sequence* (cf. [22]) with q elements (see Remark 3.3.5) and $\rho \ge 0$ is a parameter that determines the reliability of the set to contain the MMSE optimal filter $\boldsymbol{h}^*$ in (3.3.3). Intuitively, an increase of ρ inflates the set $C_\rho^{(k)}[i]$, and thus we call ρ *inflation parameter* (ρ should be described as $\rho_{n,i}$ because it can be designed independently for each set; in the following, however, such subscripts are omitted for notational simplicity).

Since C_s is completely reliable to contain $\boldsymbol{h}^*$, our strategy is to use C_s as a hard (absolute) constraint set and $\{C_\rho^{(k)}[i]\}_{i\in\mathcal{I}_k}$ as a collection of sets to which the distances should be reduced.

3.3.2 Proposed Algorithms

Let us derive the proposed algorithms from Scheme 2.0.2 with the sets in (3.3.1) and (3.3.7). Given $q \in \mathbb{N}\setminus\{0\}$, let $\{\omega_\iota^{(k)}\}_{\iota\in\mathcal{I}_k} \subset (0,1]$ satisfying $\sum_{\iota\in\mathcal{I}_k}\omega_\iota^{(k)} = 1$, $\forall k\in\mathbb{N}$, be the weights. Define the cost function

$$\Theta_k(\boldsymbol{h}) := \begin{cases} \displaystyle\sum_{\iota\in\mathcal{I}_k}\frac{\omega_\iota^{(k)}}{L_k^{(1)}}\mathrm{d}(\boldsymbol{h}_k, C_\rho^{(k)}[\iota]\cap C_s)\mathrm{d}(\boldsymbol{h}, C_\rho^{(k)}[\iota]\cap C_s), \\ \text{if } L_k^{(1)} := \displaystyle\sum_{\iota\in\mathcal{I}_k}\omega_\iota^{(k)}\mathrm{d}(\boldsymbol{h}_k, C_\rho^{(k)}[\iota]\cap C_s) \neq 0, \\ 0, \qquad \text{otherwise}, \end{cases} \tag{3.3.8}$$

where $\mathrm{d}(\boldsymbol{h}, C_\rho^{(k)}[\iota]\cap C_s)\left(= \left\|\boldsymbol{h} - P_{C_\rho^{(k)}[\iota]\cap C_s}(\boldsymbol{h})\right\|\right)$, $\forall \iota \in \mathcal{I}_k$, denotes the distance from the variable vector $\boldsymbol{h}\in\mathbb{R}^N$ to the set $C_\rho^{(k)}[\iota]\cap C_s$ (which should be reduced).

Table 3.3.1: Adaptive blind algorithms. $P_{C_s}(\boldsymbol{x}) = \boldsymbol{Q_s}\boldsymbol{x} + \boldsymbol{s}_1$, $\forall \boldsymbol{x} \in \mathbb{R}^N$.

Algorithm	Adaptation rule
OPM-GP	$\boldsymbol{h}_{k+1} = P_{C_s}\left[\boldsymbol{h}_k + \mu\left(P_{H_k}(\boldsymbol{h}_k) - \boldsymbol{h}_k\right)\right] = \boldsymbol{h}_k - \mu\dfrac{\langle \boldsymbol{h}_k, \boldsymbol{r}[k]\rangle}{\|\boldsymbol{r}[k]\|^2}\boldsymbol{Q_s}\boldsymbol{r}[k]$ where $H_k := \{\boldsymbol{h} : \langle \boldsymbol{h}, \boldsymbol{r}[k]\rangle = 0\}$
SAGP	$\widehat{A}_{1,k+1}$ & $\widehat{b}_{1,k}$ are computed by (3.3.4) & (3.3.5) $\boldsymbol{h}_{k+1} = P_{C_s}\left[\boldsymbol{h}_k + \mu\left(P^{(g)}(\boldsymbol{h}_k) - \boldsymbol{h}_k\right)\right]$, where $P^{(g)}(\boldsymbol{h}_k) := \begin{cases} P_{H_k^{(+)}}(\boldsymbol{h}_k) = \boldsymbol{h}_k - \dfrac{\langle \boldsymbol{h}_k, \boldsymbol{r}[k]\rangle - \widehat{A}_{1,k+1}}{\|\boldsymbol{r}[k]\|^2}\boldsymbol{r}[k], \\ \text{if } \langle \boldsymbol{h}_k, \boldsymbol{r}[k]\rangle > 0, \\ P_{H_k^{(-)}}(\boldsymbol{h}_k) = \boldsymbol{h}_k - \dfrac{\langle \boldsymbol{h}_k, \boldsymbol{r}[k]\rangle + \widehat{A}_{1,k+1}}{\|\boldsymbol{r}[k]\|^2}\boldsymbol{r}[k], \\ \text{otherwise,} \end{cases}$ with $H_k^{(\pm)} := \{\boldsymbol{h} : \langle \boldsymbol{h}, \boldsymbol{r}[k]\rangle = \pm\widehat{A}_{1,k+1}\}$
CNLMS	$\widehat{A}_{1,k+1}$ & $\widehat{b}_{1,k}$ are computed by (3.3.4) & (3.3.5) $\boldsymbol{h}_{k+1} = \boldsymbol{h}_k + \mu\left(P_{C_0^{(k)}[k]\cap C_s}(\boldsymbol{h}_k) - \boldsymbol{h}_k\right)$
B2P	$\widehat{A}_{1,k+1}$ is computed by (3.3.4) $\boldsymbol{h}_{k+1} = P_{C_s}\left[\boldsymbol{h}_k + \lambda_k\left(\sum_{j=0}^{q-1} w_j^{(k)} P_{C_j^{(k)}}(\boldsymbol{h}_k) - \boldsymbol{h}_k\right)\right]$ where $C_j^{(k)} := \{\boldsymbol{h} : \lvert\langle \boldsymbol{h}, \boldsymbol{r}[k-j]\rangle\rvert \le \widehat{A}_{1,k+1}\}$

When $L_k^{(1)} \neq 0$ ($\Leftrightarrow \boldsymbol{h}_k \notin \bigcap_{\iota\in\mathcal{I}_k} C_\rho^{(k)}[\iota] \cap C_s$), the weighting $\dfrac{\omega_\iota^{(k)}}{L_k^{(1)}}\mathrm{d}(\boldsymbol{h}_k, C_\rho^{(k)}[\iota] \cap C_s)$ is given to each distance function, where $L_k^{(1)}$ is the normalizing factor; the sets far from $\boldsymbol{h}_k$ have large weighting. When $L_k^{(1)} = 0$, we have $\boldsymbol{h}_k \in \bigcap_{\iota\in\mathcal{I}_k} C_\rho^{(k)}[\iota] \cap C_s$, hence nothing is left to do in this case. A subgradient of Θ_k at $\boldsymbol{h}_k$ is given by $\Theta_n'(\boldsymbol{h}_k) = \dfrac{1}{L_k^{(1)}}\sum_{\iota\in\mathcal{I}_k}\omega_\iota^{(k)}\left(\boldsymbol{h}_k - P_{C_\rho^{(k)}[\iota]\cap C_s}(\boldsymbol{h}_k)\right) \in \partial\Theta_k(\boldsymbol{h}_k)$ if $L_k^{(1)} \neq 0$; for details, see [126, p.607, Example 3].

Application of $T = I$ and $\Theta_k(\boldsymbol{h})$ in (3.3.8) to Scheme 2.0.2 yields the following algorithm.

Algorithm 3.3.1 (Blind Parallel Constrained Projection Algorithm).

Requirements: *the control sequence* $\mathcal{I}_k$, *the weights* $\omega_\iota^{(k)} > 0$ *s.t.* $\sum_{\iota\in\mathcal{I}_k}\omega_\iota^{(k)} = 1$, *the signature* $\boldsymbol{s}_1$, *the projection matrix* $\boldsymbol{Q_s} := \boldsymbol{I} - \boldsymbol{s}_1\boldsymbol{s}_1^T$ *(NOTE:* $\|\boldsymbol{s}_1\| = 1$*), the*

inflation parameter $\rho \geq 0$, *the step size* $\lambda_k \in [0,2]$ *and the forgetting factor* $\gamma \in (0,1]$.

Initialization: $\widehat{A}_{1,0} = 0$, $\boldsymbol{h}_0 = \boldsymbol{s}_1 \in C_s$

Algorithm:

1) Estimation of A_1 *and* $b_1[\iota]$

$$\begin{aligned} \widehat{A}_{1,k+1} &= \widehat{A}_{1,k} + \gamma \left(|\langle \boldsymbol{h}_k, \boldsymbol{r}[k] \rangle| - \widehat{A}_{1,k} \right) \\ \widehat{b}_{1,k}[\iota] &= sgn \langle \boldsymbol{h}_k, \boldsymbol{r}[\iota] \rangle, \quad \iota \in \mathcal{I}_k \end{aligned}$$

2) Update of filter

$$\boldsymbol{h}_{k+1} = \boldsymbol{h}_k + \lambda_k \mathcal{M}_k^{(1)} \left(\sum_{\iota \in \mathcal{I}_k} \omega_\iota^{(k)} P_{C_\rho^{(k)}[\iota] \cap C_s}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right), \tag{3.3.9}$$

where, for any $\boldsymbol{h} \in C_s$,

$$P_{C_\rho^{(k)}[\iota] \cap C_s}(\boldsymbol{h}) = \begin{cases} \boldsymbol{h} - \dfrac{\langle \boldsymbol{h}, \boldsymbol{r}[\iota] \rangle - \widehat{A}_{1,k+1} \widehat{b}_{1,k}[\iota] - \sqrt{\rho}}{\boldsymbol{r}[\iota]^T \boldsymbol{Q_s} \boldsymbol{r}[\iota]} \boldsymbol{Q_s} \boldsymbol{r}[\iota], \\ \quad if \langle \boldsymbol{h}, \boldsymbol{r}[\iota] \rangle - \widehat{A}_{1,k+1} \widehat{b}_{1,k}[\iota] > \sqrt{\rho}, \\ \boldsymbol{h} - \dfrac{\langle \boldsymbol{h}, \boldsymbol{r}[\iota] \rangle - \widehat{A}_{1,k+1} \widehat{b}_{1,k}[\iota] + \sqrt{\rho}}{\boldsymbol{r}[\iota]^T \boldsymbol{Q_s} \boldsymbol{r}[\iota]} \boldsymbol{Q_s} \boldsymbol{r}[\iota], \\ \quad if \langle \boldsymbol{h}, \boldsymbol{r}[\iota] \rangle - \widehat{A}_{1,k+1} \widehat{b}_{1,k}[\iota] < -\sqrt{\rho}, \\ \boldsymbol{h}, \qquad otherwise, \end{cases} \tag{3.3.10}$$

$$\mathcal{M}_k^{(1)} := \begin{cases} \dfrac{\sum_{\iota \in \mathcal{I}_k} \omega_\iota^{(k)} \left\| P_{C_\rho^{(k)}[\iota] \cap C_s}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\|^2}{\left\| \sum_{\iota \in \mathcal{I}_k} \omega_\iota^{(k)} P_{C_\rho^{(k)}[\iota] \cap C_s}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\|^2}, & if\ \boldsymbol{h}_k \notin \bigcap_{\iota \in \mathcal{I}_k} C_\rho^{(k)}[\iota] \cap C_s, \\ 1, & otherwise. \end{cases}$$

NOTE: For all $k \in \mathbb{N}$, $\boldsymbol{h}_k \in C_s$ holds, since (i) $\boldsymbol{h}_0 \in C_s$ and (ii) $\boldsymbol{h}_k \in C_s \Rightarrow \boldsymbol{h}_{k+1} \in C_s$ from (3.3.1) and (3.3.9). The proof of (3.3.10) is given in 3.3.3. A weighted average of multiple projections as in (3.3.9) is referred to as *parallel projection* [24], since it can be computed in parallel by using q concurrent processors.

On the other hand, application of $T = I$ and $\Theta_k(\boldsymbol{h}) := \Psi_k(P_{C_s}(\boldsymbol{h}))$, where

$$\Psi_k(\boldsymbol{h}) := \begin{cases} \displaystyle\sum_{\iota\in\mathcal{I}_k} \frac{\omega_\iota^{(k)}}{L_k^{(2)}} \mathrm{d}(\boldsymbol{h}_k, C_\rho^{(k)}[\iota])\mathrm{d}(\boldsymbol{h}, C_\rho^{(k)}[\iota]), & \text{if } L_k^{(2)} := \sum_{\iota\in\mathcal{I}_k} \omega_\iota^{(k)} \mathrm{d}(\boldsymbol{h}_k, C_\rho^{(k)}[\iota]) \neq 0, \\ 0, & \text{otherwise,} \end{cases}$$

to Scheme 2.0.2 yields the following algorithm (for details about the derivation of the algorithm, see [126, p.610, Example 5]).

Algorithm 3.3.2 (Blind Constrained Parallel Projection Algorithm).

Requirements & Initialization: *the same as Algorithm* 3.3.1

Algorithm:

1) Estimation of A_1 *and* $b_1[\iota]$*: the same as Algorithm* 3.3.1

2) Update of filter

$$\boldsymbol{h}_{k+1} = \boldsymbol{h}_k + \lambda_k \mathcal{M}_k^{(2)} P_{\widetilde{C}_s}\left(\sum_{\iota\in\mathcal{I}_k} \omega_\iota^{(k)} P_{C_\rho^{(k)}[\iota]}(\boldsymbol{h}_k) - \boldsymbol{h}_k\right), \tag{3.3.11}$$

where $\widetilde{C}_s := \{\boldsymbol{h} \in \mathbb{R}^N : \langle \boldsymbol{h}, \boldsymbol{s}_1\rangle = 0\}$ *is a translated linear subspace of* C_s *and*

$$P_{\widetilde{C}_s}(\boldsymbol{h}) = \boldsymbol{Q_s h},$$

$$P_{C_\rho^{(k)}[\iota]}(\boldsymbol{h}) = \begin{cases} \boldsymbol{h} - \dfrac{\langle \boldsymbol{h}, \boldsymbol{r}[\iota]\rangle - \widehat{A}_{1,k+1}\widehat{b}_{1,k}[\iota] - \sqrt{\rho}}{\|\boldsymbol{r}[\iota]\|^2}\boldsymbol{r}[\iota], & \textit{if } \langle \boldsymbol{h}, \boldsymbol{r}[\iota]\rangle - \widehat{A}_{1,k+1}\widehat{b}_{1,k}[\iota] > \sqrt{\rho}, \\ \boldsymbol{h} - \dfrac{\langle \boldsymbol{h}, \boldsymbol{r}[\iota]\rangle - \widehat{A}_{1,k+1}\widehat{b}_{1,k}[\iota] + \sqrt{\rho}}{\|\boldsymbol{r}[\iota]\|^2}\boldsymbol{r}[\iota], & \textit{if } \langle \boldsymbol{h}, \boldsymbol{r}[\iota]\rangle - \widehat{A}_{1,k+1}\widehat{b}_{1,k}[\iota] < -\sqrt{\rho}, \\ \boldsymbol{h}, & \textit{otherwise,} \end{cases}$$

$$\mathcal{M}_k^{(2)} := \begin{cases} \dfrac{\sum_{\iota\in\mathcal{I}_k} \omega_\iota^{(k)} \left\| P_{C_\rho^{(k)}[\iota]}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\|^2}{\left\| P_{\widetilde{C}_s}\left(\sum_{\iota\in\mathcal{I}_k} \omega_\iota^{(k)} P_{C_\rho^{(k)}[\iota]}(\boldsymbol{h}_k) - \boldsymbol{h}_k\right)\right\|^2}, & \textit{if } \sum_{\iota\in\mathcal{I}_k} \omega_\iota^{(k)} P_{C_\rho^{(k)}[\iota]}(\boldsymbol{h}_k) - \boldsymbol{h}_k \notin \widetilde{C}_s^{\perp}, \\ 1, & \textit{otherwise.} \end{cases}$$

Algorithm 3.3.2 belongs to the family of *Embedded Constraint Adaptive Projected Subgradient Method (EC-APSM)* [124–126]. Moreover, Algorithm 3.3.1 can

be generalized into a new family of embedded constraint algorithms shown in Sec. 3.3.4.

Remark 3.3.3 (Computational complexity). *Note that the computation of* $\boldsymbol{Q}_{\boldsymbol{s}}\boldsymbol{a} = \boldsymbol{a} - \boldsymbol{s}_1(\boldsymbol{s}_1^T\boldsymbol{a})$, $\forall \boldsymbol{a} \in \mathbb{R}^N$, *requires* $2N$ *multiplications. Moreover,* $\forall \iota \in \mathcal{I}_k \setminus \{n\}$, *"*$\boldsymbol{r}[\iota]^T\boldsymbol{Q}_{\boldsymbol{s}}\boldsymbol{r}[\iota]$ *and* $\boldsymbol{Q}_{\boldsymbol{s}}\boldsymbol{r}[\iota]$ *in Algorithm* 3.3.1*" and "*$\|\boldsymbol{r}[\iota]\|^2$ *in Algorithm* 3.3.2*" are computed at the previous iterations. Hence, we see that both Algorithms* 3.3.1 *and* 3.3.2 *require* $(4q+5)N$ *multiplications at each iteration. Furthermore, note that each term in the summation in* (3.3.9) *[or* (3.3.11)*] can be computed in parallel (independently). Therefore, with* q *concurrent processors, the number of multiplications imposed on each processor is reduced to* $9N$ *no matter how many projections are used; i.e., the complexity order imposed on each processor is linear. This implies that the proposed algorithms are suitable for real-time implementation. On the other hand, the Recursive Least Squares (RLS)-based-MMSE method* [38, 39, 92] *and the subspace approach* [119]*, which are well-known blind methods, require* $O(N^2)$ *and* $(4L+3)N + O(L)$ *multiplications, respectively. Moreover, for good performance, the subspace approach needs to detect the exact number of strong interferences, which increases the overall system complexity.*

Table 3.3.1 gives a unified view, with projection operators, to the following blind algorithms: the normalized OPM-based gradient projection (OPM-GP) [62, 90], the SAGP [90], the blind CNLMS that is based on the idea of [67] combined with our defining sets in (3.3.7), and the B2P [20]. The OPM-GP [90] is a normalized version of the blind MOE algorithm [62]; the algorithms are called respectively *projected NLMS* and *projected LMS* in [28]. It is not hard to see that the CNLMS is a special case of Algorithm 3.3.1 with $q = 1$ and $\rho = 0$. It should be remarked that the steady-state performance of the B2P and the SAGP may be different, since the algorithms use different sets as shown in Table 3.3.1. The SAGP utilizes the so-called *generalized projection* $P^{(g)}(\boldsymbol{h}_k)$ (see e.g., [110]), which gives a nearest point from $\boldsymbol{h}_k$ in the non-convex set $H_k^{(+)} \cup H_k^{(-)}$. The generalized projection is not a strict projection because it is not always unique (cf. the definition of projection in Chapter 2). In fact, if $\langle \boldsymbol{h}_k, \boldsymbol{r}[k] \rangle = 0$, there exist two nearest points from $\boldsymbol{h}_k$ in $H_k^{(+)} \cup H_k^{(-)}$; $P_{H_k^{(+)}}(\boldsymbol{h}_k)$ and $P_{H_k^{(-)}}(\boldsymbol{h}_k)$. Fortunately, a geometric comparison of the SAGP with the proposed algorithms is possible (see Remark 3.3.4), since $C_0^{(k)}[k]$ coincides with $H_k^{(+)}$ [or $H_k^{(-)}$] when $\mathrm{sgn}\langle \boldsymbol{h}_k, \boldsymbol{r}[k] \rangle = 1$

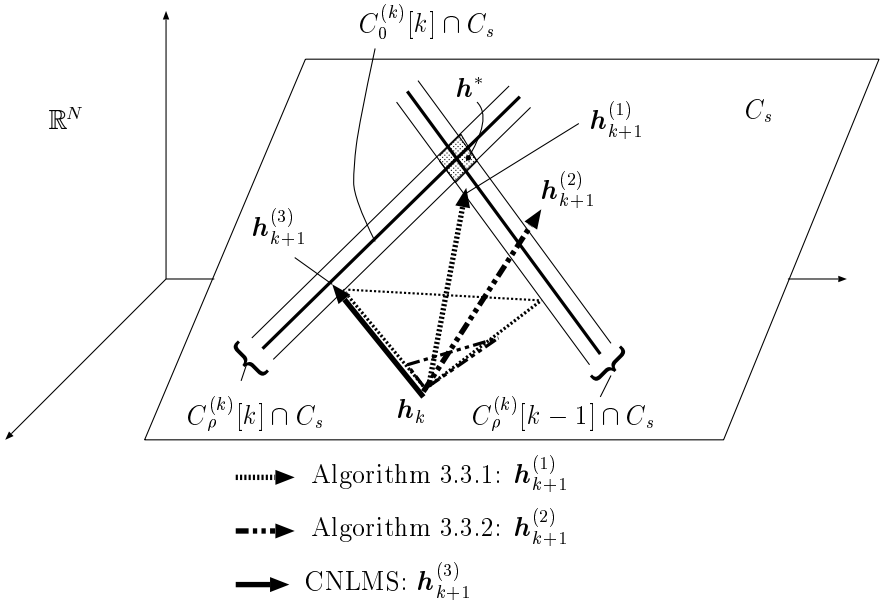


Figure 3.3.1: A geometric interpretation of embedded constraint methods: the proposed algorithms and the CNLMS algorithm. The dotted area shows $C_\rho^{(k)}[k] \cap C_\rho^{(k)}[k-1] \cap C_s$.

(or $\mathrm{sgn}\langle \boldsymbol{h}_k, \boldsymbol{r}[k] \rangle = -1$), by (3.3.5), (3.3.7) and Table 3.3.1. It is easily seen that $C_0^{(k)}$ used in the B2P is a closed convex set bounded by the hyperplanes $H_k^{(+)}$ and $H_k^{(-)}$ used in the SAGP.

Figures 3.3.1 and 3.3.2 illustrate geometric interpretations of the proposed algorithms compared with a simple embedded constraint method (the CNLMS) and non-embedded constraint methods (the SAGP and the B2P), respectively. A geometric interpretation of the OPM-GP is also possible; the set H_k is nothing but the translated subspace of $H_k^{(+)}$ [or $H_k^{(-)}$]. For visual clarity, however, it is omitted. For the proposed algorithm and the B2P, the uniform weights, $\omega_\iota^{(k)} = 1/2$ ($\forall \iota = 1, 2$), are employed with $q = 2$ parallel processors. For the B2P, the step size is set to $\mathcal{M}_k$. For the other methods, the step sizes are set to 1. The MMSE optimal filter $\boldsymbol{h}^*$ is assumed to satisfy $\boldsymbol{h}^* \in C_\rho^{(k)}[k] \cap C_\rho^{(k)}[k-1] \cap C_s$. All algorithms are assumed to have, if necessary, a common amplitude estimation $\widehat{A}_{1,k+1}$ and a correct bit estimation $\widehat{b}_{1,k}[k]$. A remark on geometric comparisons is given below.

Remark 3.3.4 (Geometric comparisons). *Referring to Fig.* 3.3.1*, we see that the proposed algorithms generate closer points to the MMSE optimal filter* $\boldsymbol{h}^*$ *than the CNLMS due to its parallel structure; i.e., the proposed algorithms utilize multiple data simultaneously. As also seen in the figure, Algorithm* 3.3.1 *takes an averaged*

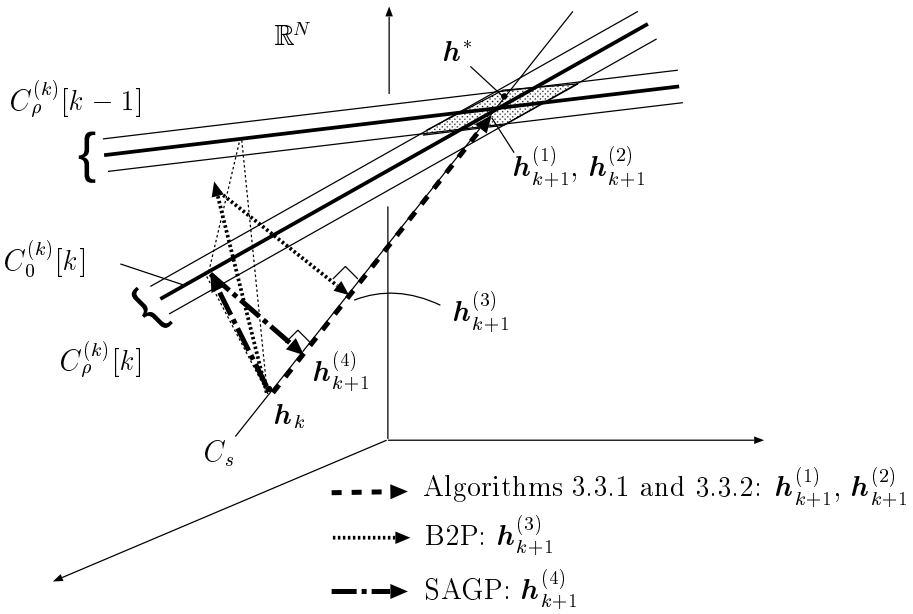


Figure 3.3.2: A geometric interpretation of non-embedded constraint methods (the SAGP and the B2P) and the proposed methods. The dotted area shows $C_\rho^{(k)}[k] \cap C_\rho^{(k)}[k-1]$.

direction of exact projections onto $\{C_\rho^{(k)}[\iota] \cap C_s\}_{\iota \in \mathcal{I}_k}$, *while Algorithm* 3.3.2 *takes an averaged direction of relaxed projections. The "relaxation" depends on the angle between* $\boldsymbol{s}_1$ *(the normal vector of the hyperplane* C_s*) and* $\boldsymbol{r}[\iota]$ *(the one of the boundary hyperplanes of* $C_\rho^{(k)}[\iota]$*).*

Referring to Fig. 3.3.2*, we see that the B2P generates a closer point to* $\boldsymbol{h}^*$ *than the SAGP due to its parallel structure. The proposed algorithms generate even closer points than the B2P due to its embedded constraint structure in addition to its parallel structure. We also see that the SAGP and the B2P are constructed by two steps; the second step* $P_{C_s}(\cdot)$ *in Table* 3.3.1 *is to enforce the filter in the constraint set. On the other hand, the CNLMS and the proposed algorithms update the filter along the constraint set, and hence they are constructed by one step.*

The discussion in Remark 3.3.4 will be supported by simulation in Sec. 3.4. Finally, from our observation, a simple strategy for the design of γ and q [cf. (3.3.4) and (3.3.7)] is given below.

Remark 3.3.5 (On design of γ and q). *From Remark* 3.A.2-(d) *in the appendix,* $\widehat{A}_{1,k+1} \approx A_1$ *should be valid for good steady-state performance, which can be obtained with small* γ*, although it may decrease the speed of convergence* [90]*. From our experience,* q *leads to good performance when* $T_{\text{a.c.}}/qT_{\text{b}} > 0.1$*, where* $T_{\text{a.c.}}$ *and* T_{b} *denote the period when the channels are almost constant and the bit period, respectively.*

3.3.3 Proof of Equation (3.3.10)

Suppose $\boldsymbol{h} \in C_s$. For notational simplicity, in this section, we represent the stochastic property set $C_\rho^{(k)}[\iota]$ as C [see (3.3.7)]. The set C is a closed convex set bounded by two hyperplanes

$$\begin{aligned} H_+ &:= \{\boldsymbol{x} \in \mathbb{R}^N : \langle \boldsymbol{x}, \boldsymbol{r}[\iota]\rangle - \widehat{A}_{1,k+1}\widehat{b}_{1,k}[\iota] = \sqrt{\rho}\}, \\ H_- &:= \{\boldsymbol{x} \in \mathbb{R}^N : \langle \boldsymbol{x}, \boldsymbol{r}[\iota]\rangle - \widehat{A}_{1,k+1}\widehat{b}_{1,k}[\iota] = -\sqrt{\rho}\}. \end{aligned}$$

(a) Assume $-\sqrt{\rho} \le \langle \boldsymbol{h}, \boldsymbol{r}[\iota]\rangle - \widehat{A}_{1,k+1}\widehat{b}_{1,k}[\iota] \le \sqrt{\rho}$ ($\Leftrightarrow \boldsymbol{h} \in C$). In this case,

$$P_{C\cap C_s}(\boldsymbol{h}) = \boldsymbol{h}.$$

In the other cases, $P_{C\cap C_s}(\boldsymbol{h}) = P_{H_{\text{sgn}}\cap C_s}(\boldsymbol{h})$, where H_{sgn} (sgn: $+$ or $-$) is the nearest hyperplane, from $\boldsymbol{h}$, of the two H_+ and H_-.

(b) Assume $\langle \boldsymbol{h}, \boldsymbol{r}[\iota]\rangle - \widehat{A}_{1,k+1}\widehat{b}_{1,k}[\iota] > \sqrt{\rho}$ ($\Rightarrow \boldsymbol{h} \notin C$). In this case, the nearest hyperplane is obviously H_+, and hence $P_{C\cap C_s}(\boldsymbol{h}) = P_{H_+\cap C_s}(\boldsymbol{h})$. Since

$$H_+ \cap C_s = \left\{\boldsymbol{x} : \boldsymbol{x}^T [\boldsymbol{s}_1, \boldsymbol{r}[\iota]] = \left[1, \widehat{A}_{1,k+1}\widehat{b}_{1,k}[\iota] + \sqrt{\rho}\right]\right\},$$

we have (cf. e.g., [73, p.65 Theorem 2])

$$P_{H_+\cap C_s}(\boldsymbol{h}) = \boldsymbol{h} - \boldsymbol{G}(\boldsymbol{G}^T\boldsymbol{G})^{-1}(\boldsymbol{G}^T\boldsymbol{h} - \boldsymbol{v}),$$

where $\boldsymbol{G} := [\boldsymbol{r}[\iota], \boldsymbol{s}_1]$ and $\boldsymbol{v} := \begin{bmatrix} \widehat{A}_{1,k+1}\widehat{b}_{1,k}[\iota] + \sqrt{\rho} \\ 1 \end{bmatrix}$. Using $\langle \boldsymbol{s}_1, \boldsymbol{h}\rangle = 1$, $\|\boldsymbol{s}_1\| = 1$ and $\boldsymbol{I} - \boldsymbol{s}_1\boldsymbol{s}_1^T = \boldsymbol{Q}_{\boldsymbol{s}}$ (see Requirements in Algorithm 3.3.1), we

obtain

$$P_{C\cap C_s}(\boldsymbol{h}) = \boldsymbol{h} - \frac{\langle \boldsymbol{h}, \boldsymbol{r}[\iota]\rangle - \widehat{A}_{1,k+1}\widehat{b}_{1,k}[\iota] - \sqrt{\rho}}{\boldsymbol{r}[\iota]^T \boldsymbol{Q}_s \boldsymbol{r}[\iota]} \boldsymbol{Q}_s \boldsymbol{r}[\iota].$$

(c) Assume $\langle \boldsymbol{h}, \boldsymbol{r}[\iota]\rangle - \widehat{A}_{1,k+1}\widehat{b}_{1,k}[\iota] < -\sqrt{\rho}$ ($\Rightarrow \boldsymbol{h} \notin C$). In this case, the nearest hyperplane is obviously H_-, and hence $P_{C\cap C_s}(\boldsymbol{h}) = P_{H_-\cap C_s}(\boldsymbol{h})$. In analogy with (b), we can verify

$$P_{C\cap C_s}(\boldsymbol{h}) = \boldsymbol{h} - \frac{\langle \boldsymbol{h}, \boldsymbol{r}[\iota]\rangle - \widehat{A}_{1,k+1}\widehat{b}_{1,k}[\iota] + \sqrt{\rho}}{\boldsymbol{r}[\iota]^T \boldsymbol{Q}_s \boldsymbol{r}[\iota]} \boldsymbol{Q}_s \boldsymbol{r}[\iota],$$

which completes the proof. □

3.3.4 New Family of Embedded Constraint Algorithms

Let us consider the following problem.

Problem 3.3.6. *Suppose q sets $\{S_\iota(k)\}_{\iota=1}^q \subset \mathbb{R}^N$ are defined for each $k \in \mathbb{N}$. Find a sequence $(\boldsymbol{h}_k)_{k\in\mathbb{N}} \subset \mathbb{R}^N$ that asymptotically minimizes the distance to $(\{S_\iota(k)\}_{\iota=1}^q)_{k\in\mathbb{N}}$ over a linear variety V.*

Setting $V = C_s$ and $S_\iota(k) = C_\rho^{(k)}[\iota]$, $\forall k \in \mathbb{N}$, $\forall \iota \in \mathcal{I}_k (:= \{1, 2, \cdots, q\})$, Problem 3.3.6 is reduced to the one in Sec. 3.3. Conversely, using V and $S_\iota(k)$ instead of C_s and $C_\rho^{(k)}[\iota]$, $\forall k \in \mathbb{N}$, $\forall \iota \in \mathcal{I}_k$, in Algorithm 3.3.1, respectively, we obtain the following scheme to solve Problem 3.3.6.

Scheme 3.3.7 (Adaptive Parallel Constrained Projection [A-PCP] Method)**.** *Generate a sequence $(\boldsymbol{h}_k)_{k\in\mathbb{N}}$ by*

$$\boldsymbol{h}_{k+1} = \boldsymbol{h}_k + \lambda_k \mathcal{M}_k \left(\sum_{\iota=1}^q \omega_\iota^{(k)} P_{S_\iota(k)\cap V}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right),$$

$\forall k \in \mathbb{N}$, where $\boldsymbol{h}_0 \in V$, $\lambda_k \in [0, 2]$ and

$$\mathcal{M}_k := \begin{cases} \dfrac{\sum_{\iota=1}^q \omega_\iota^{(k)} \left\| P_{S_\iota(k)\cap V}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\|^2}{\left\| \sum_{\iota=1}^q \omega_\iota^{(k)} P_{S_\iota(k)\cap V}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\|^2}, & \text{if } \boldsymbol{h}_k \notin \bigcap_{\iota=1}^q S_\iota(k) \cap V, \\ 1, & \text{otherwise.} \end{cases}$$

If, in Scheme 3.3.7, the projection onto $S_\iota(k) \cap V$ is computationally expensive, an outer approximating closed half-space can be used instead of $S_\iota(k)$ as in the *adaptive parallel subgradient projection algorithm* (see Example 6.3.2). When $S_\iota(k)$ ($\forall k \in \mathbb{N}$) is a hyperplane, the choice of $q = 1$ in Scheme 3.3.7 derives the CNLMS algorithm [67].

3.4 Numerical Examples

This section provides the results of some computer simulations, all of which are performed under the following conditions. The number of interfering users is $(K-1) = 5$, and all users have amplitude 10 times greater than the amplitude of the desired signal $A_1 = 1$. Signals are modulated by length-31 Gold sequences ($N = 31$), which are chosen randomly. For all algorithms, $\boldsymbol{h}_0 = \boldsymbol{s}_1 \in C_s$ is employed and, if the estimation of the amplitude is needed, the forgetting factor is set to $\gamma = 0.01$, by following the way in [20, 90].

3.4.1 Effects of Inflation Parameter

First, the effects of the inflation parameter ρ in (3.3.7) are examined. Figure 3.4.3 compares the output Signal to Interference-plus-Noise Ratio (SINR) performance of Algorithm 3.3.1, which at the kth iteration is obtained by

$$\mathrm{SINR}_k := \frac{\sum_{u=1}^{U} \left\langle \boldsymbol{h}_k^{(u)}, \boldsymbol{s}_1 \right\rangle^2}{\sum_{u=1}^{U} \left[\frac{\left\langle \boldsymbol{h}_k^{(u)}, \boldsymbol{r}^{(u)}[k] - A_1^{(u)} b_1^{(u)}[k] \boldsymbol{s}_1 \right\rangle}{A_1^{(u)}} \right]^2}.$$

Here $\boldsymbol{h}_k^{(u)}$ and $\boldsymbol{r}^{(u)}[k]$ are the respective vectors at the uth realization, $A_1^{(u)}$ and $b_1^{(u)}[k]$ are respectively the amplitude and the kth transmitted bits of the desired user at the uth realization, and $U = 500$ is the number of realizations. For simplicity, the path delays of users 2 to 6 are integer multiples of the chip rate; i.e., $aT_\mathrm{c} \in [0, T_\mathrm{b}]$, $a \in \mathbb{N}$ (T_c: chip period). The delays were chosen randomly with equal probability among given multiples at every realizations. The simulations are performed under Signal (or bit energy) to Noise Ratio (SNR) $:= 10 \log_{10} \frac{A_1^2}{\sigma_\mathrm{n}^2}$ =15

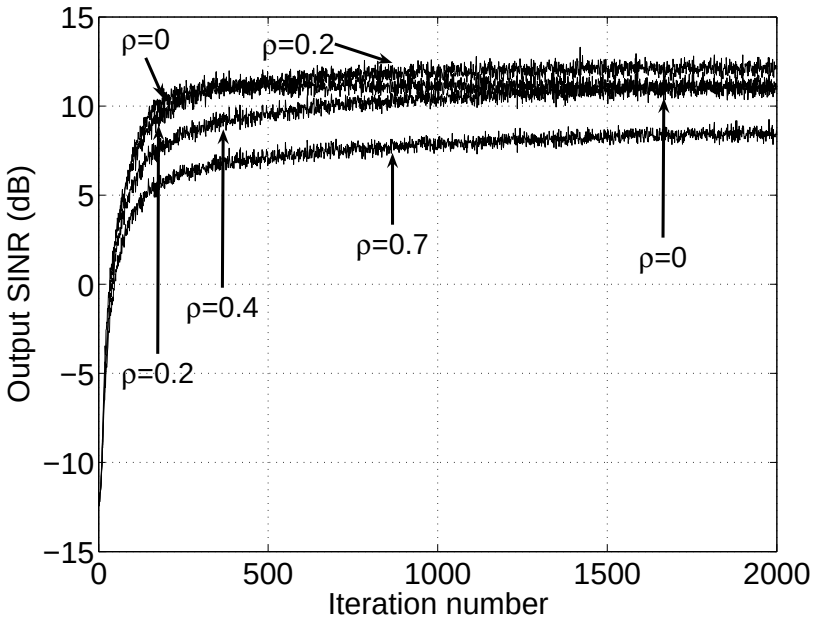


Figure 3.4.3: SINR curves of Algorithm 3.3.1 with different values of inflation parameter ρ under SNR=15 dB.

dB, where $\sigma_{\rm n}^2$ is the variance of additive noise. Different fixed values, $\rho = 0$, 0.2, 0.4 and 0.7, are assigned to the inflation parameter. For simplicity, we set $\boldsymbol{r}[i] = \boldsymbol{r}[1]$ for $i \leq 1$, and $\omega_\iota^{(k)} = \dfrac{1}{q}$, $\forall \iota \in \mathcal{I}_k$. The step size $\lambda_k = 0.2$ (see also below) is employed with $q = 16$ parallel projections.

We observe that, although "$\rho = 0$" exhibits the fastest initial convergence in the experiments, "$\rho = 0.2$" achieves better steady-state performance ("$\rho = 0.4$" and "$\rho = 0.7$" are also expected to achieve higher SINR than "$\rho = 0$" after more iterations). Considering the performance in the initial and steady states, "$\rho = 0.2 \approx 6\sigma_{\rm n}^2$" may be an effective fixed value in this simulation. Note, however, that ρ should be designed by taking into account influence of MAI and estimation errors in $\widehat{A}_{1,k+1}$ & $\widehat{b}_{1,k}[i]$ as well as noise. Hence, the design of inflation parameter needs additional discussion, which will be addressed in a future work; a simple fundamental analysis on this designing problem is reported in [139] (see also [140]). With an appropriately designed inflation parameter, the step size λ_k can naturally be set to 1; $\boldsymbol{h}^*$ may not belong to the simple sets we designed herein, and $\lambda_k = 0.2$ realizes robustness against such a situation in our simulations.

The proposed algorithms are next compared with the *Generalized Projection*

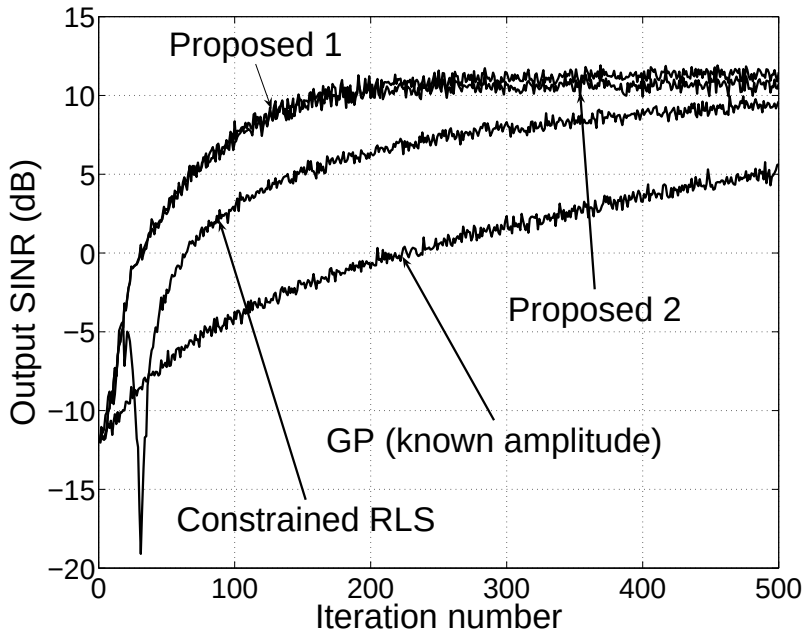


Figure 3.4.4: Proposed algorithms versus GP and constrained RLS in SINR under SNR=15 dB.

(GP) algorithm[2] [90] and the constrained RLS algorithm based on the so-called *orthogonal matched filtering detector* [38, 39] (see [92] for a convergence analysis of the blind RLS-based method). For the GP and proposed algorithms, the step sizes are set to 0.2. For the proposed algorithms, we set $\rho = 0$ and the other parameters to the same as in the previous simulation shown in Fig. 3.4.3. The results are depicted in Fig. 3.4.4. We observe that the proposed algorithms exhibit much faster convergence than the conventional algorithms, which suggests the advantage of the proposed algorithms in tracking ability.

A fresh look at Fig. 3.4.3 brings a natural suggestion that excellent performance in both initial and steady states will be simultaneously realized by assigning "$\rho = 0$" at the beginning and "an appropriate value of ρ" after convergence; this suggestion is consistent with the results in [139]. To verify this suggestion, additional experiments are performed below.

[2]The GP algorithm is classified into the family of *semi-blind*, since the algorithm is based on the assumption that the amplitude as well as the signature of a desired user is known. The algorithm is given by replacing the estimated amplitude with the known one in the SAGP algorithm, which is shown in Table 3.3.1.

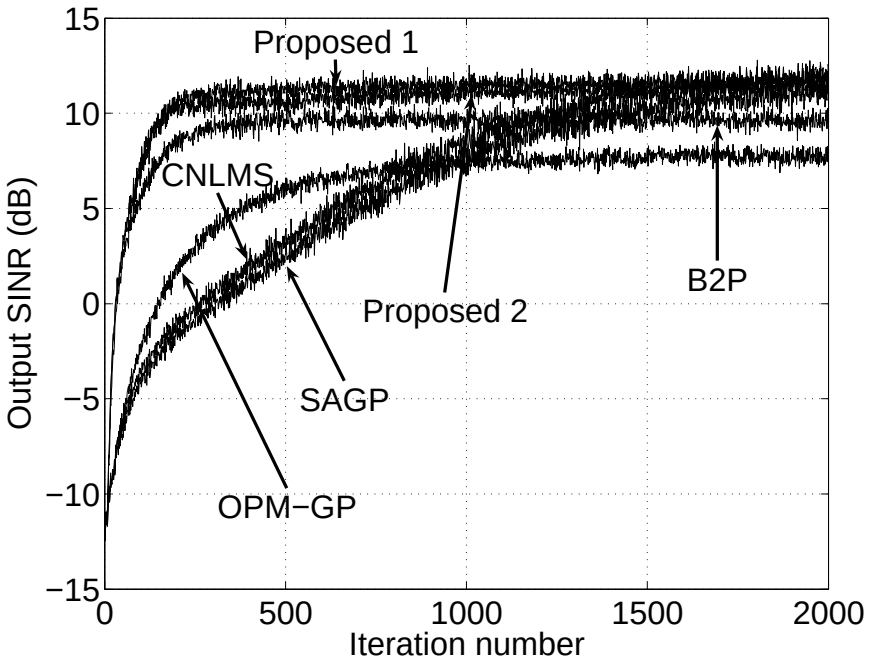


Figure 3.4.5: Proposed algorithms versus other blind methods in SINR under SNR=15 dB.

3.4.2 Proposed Methods with Change of Inflation Parameter & Comparison with Other Blind Methods

We assign 0 at the beginning and 0.7 after iteration number 500 to the inflation parameter ρ, and the other parameters are the same as employed in Fig. 3.4.3. Figure 3.4.5 compares the SINR performance, under SNR =15 dB, of the proposed algorithms with the ones presented in Table 3.3.1 (See [20] for comparisons with another major blind method, the Constant Modulus with Amplitude Estimation (CMAE) [94]). For Algorithm 3.3.2 and the B2P, the same parameters as Algorithm 3.3.1 are employed (For the B2P, the step size is set to $\lambda_k = 0.2\mathcal{M}_k$). For the OPM-GP, the SAGP and the CNLMS, step sizes are set to 0.2 for a fair comparison. As expected from Remark 3.3.4, we observe that the proposed algorithms outperform all other methods in terms of speed of convergence, while attaining excellent SINR in the steady state. Moreover, the additional computational complexity imposed by the proposed algorithms can be somehow alleviated by using processors in parallel (see Remark 3.3.3). As suggested in the end of Sec. 3.4.1, we observe that the steady-state performance of Algorithm 3.3.1 is improved by

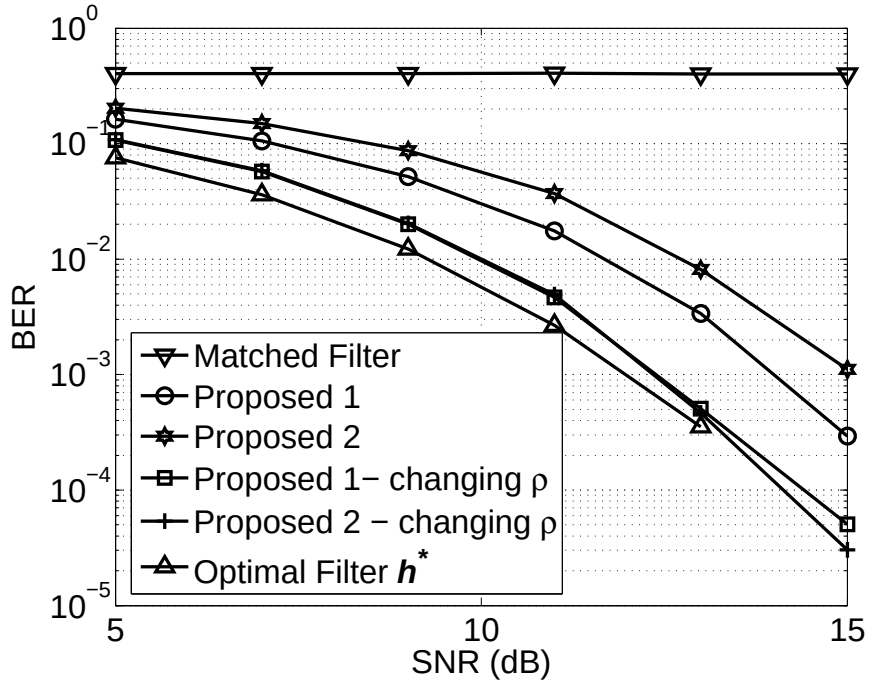


Figure 3.4.6: BER curves of the proposed algorithms with ρ (i) fixed throughout simulations and (ii) switched after convergence.

approximately 1 dB, although, judged from Fig. 3.4.3, the choice of $\rho = 0.7$ may not be the best.

To highlight the steady-state performance, the Bit Error Rate (BER) performance is depicted in Figs. 3.4.6 and 3.4.7 over SNR ranging from 5 to 15 dB. To capture the steady-state performance in a fair manner, 6000 bits are transmitted at each realization and the last 1000 bits for 100 realizations are used to calculate the BER. For a comparison, the line by the optimal filter $\boldsymbol{h}^*$ is depicted, which is computed by (3.A.1) and $\boldsymbol{R_r} = A_1^2 \boldsymbol{s}_1 \boldsymbol{s}_1^T + \sum_{l=2}^{L} A_l^2 \bar{\boldsymbol{s}}_l \bar{\boldsymbol{s}}_l^T + \sigma_{\mathrm{n}}^2 \boldsymbol{I}$ (see the appendix), with full information, based on the independence assumption.

Figure 3.4.6 compares the BER of the proposed algorithms with "changing the inflation parameter ρ as in Fig. 3.4.5" and "fixing ρ to 0". We see that the BER performance is significantly improved due to the change of ρ. In Fig. 3.4.7, the BER performance of the proposed algorithms with changing ρ is compared with the blind methods employed in Fig. 3.4.5. Referring to Figs. 3.4.5 and 3.4.7, we observe that the proposed algorithms achieve much faster convergence in SINR than the SAGP and the CNLMS as well as almost the same BER performance as

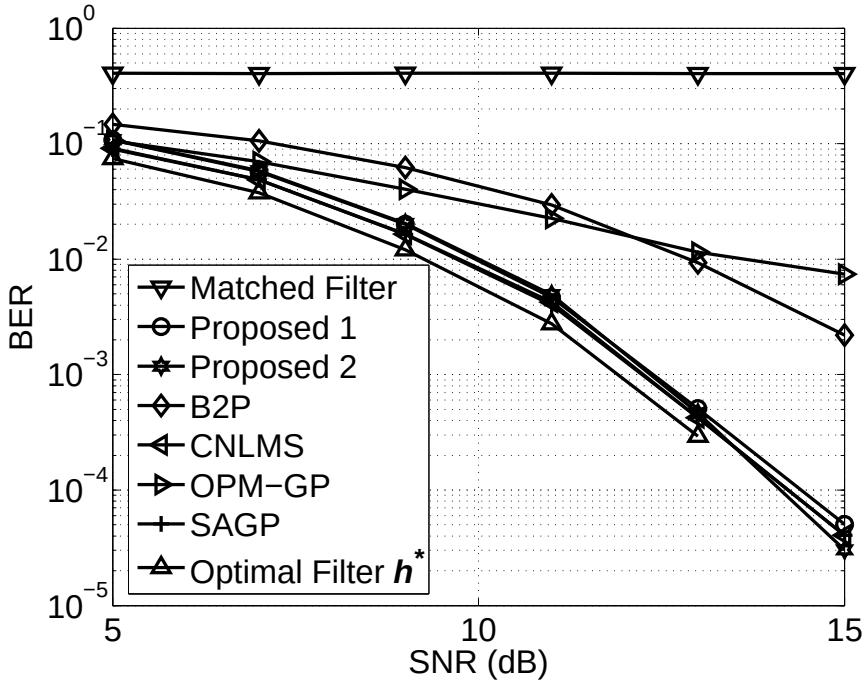


Figure 3.4.7: Proposed algorithms versus other blind methods in BER.

the optimal filter. Also we observe that the proposed algorithms drastically outperform the OPM-GP and the B2P in BER. Reviewing Fig. 3.4.3 and considering that the CNLMS is a special case of Algorithm 3.3.1 with $q = 1$ [see (3.3.9) and Table 3.3.1], another suggestion is brought that the steady-state performance of Algorithm 3.3.1 will also be improved by switching q to 1 after convergence.

To verify this second suggestion, further experiments for the proposed algorithms are performed under SNR = 15 dB, where the number of parallel projections is set to $q = 16$ at the beginning and it is switched to 1 at iteration number 500 and the inflation parameter is fixed to $\rho = 0$ throughout the simulations. The other parameters are the same as in Fig. 3.4.5. Figure 3.4.8 compares the SINR performance of the proposed algorithms with the blind methods used in Fig. 3.4.5. We observe that the performance in the steady state is efficiently improved by decreasing the number of parallel projections after convergence, as expected by the second suggestion. This switching strategy is easily realized in hardware implementation.

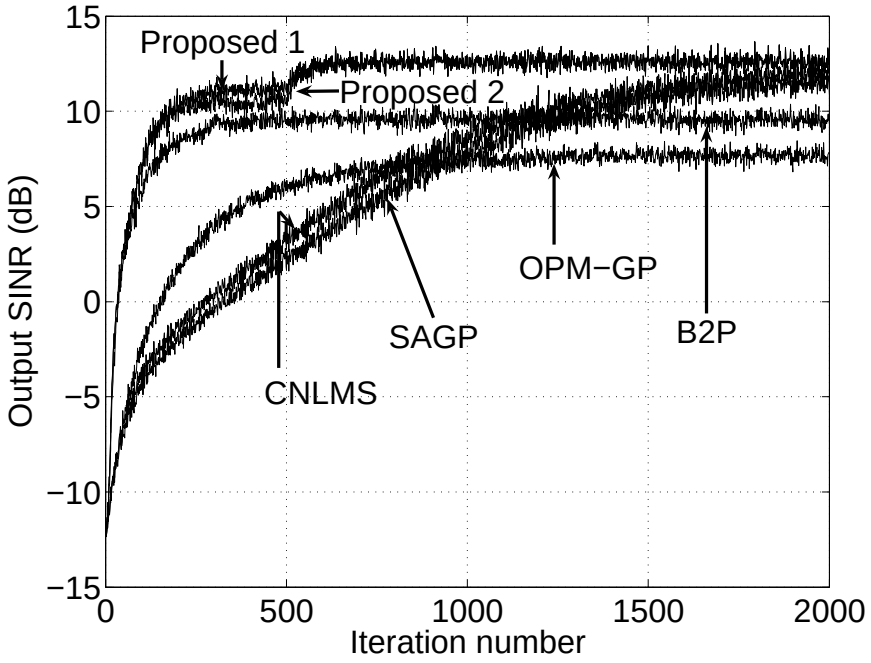


Figure 3.4.8: "Proposed algorithms with q switched to 1 after convergence" versus other blind methods in SINR under SNR=15 dB.

3.4.3 Comparison with Non-Blind Methods

Finally, Fig. 3.4.9 compares the proposed algorithms, under SNR=15 dB, with the non-blind (semi-blind) algorithms; the GP algorithm [90] with known amplitude of desired user, and the Normalized Least Mean Square (NLMS) and RLS[3] algorithms [56] with training sequences. For the non-blind methods, parameters are adjusted to achieve the fastest noticeable rate of convergence. More precisely, we set $\delta = 1.0 \times 10^{-4}$ and $\lambda = 0.98$ for RLS; and $\mu = 0.6$ for NLMS and GP. For the proposed algorithms and the B2P, the employed parameters are the same as in Fig. 3.4.5. We observe that the proposed algorithms achieve rather faster convergence than the non-blind NLMS, and exhibit comparable speed of convergence to the non-blind RLS. These remarkable improvements are accomplished by the embedded constraint and parallel structures.

[3]It should be stressed that the non-blind RLS algorithm here is different from the (blind) constrained RLS algorithm employed in Sec. 3.4.1.

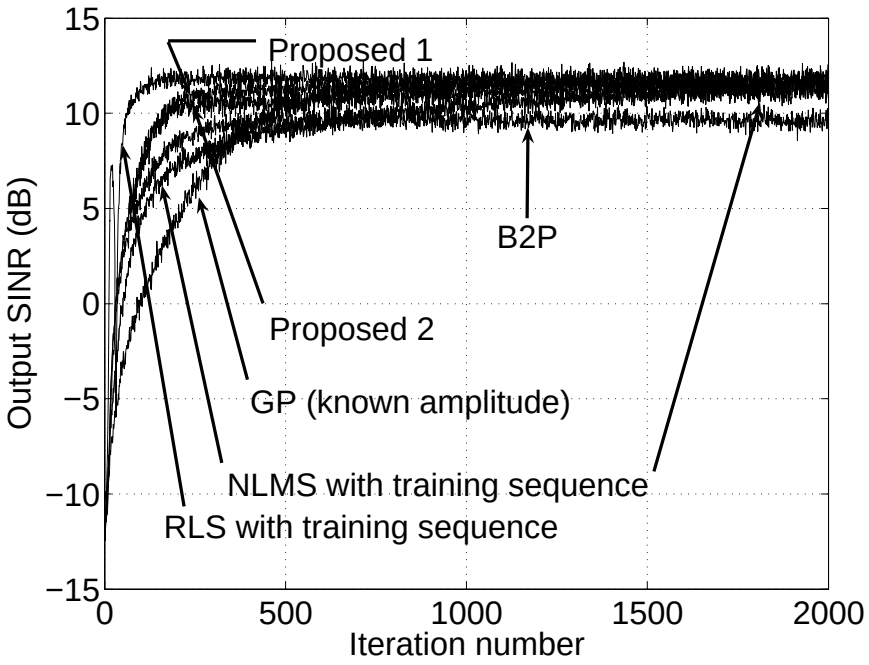


Figure 3.4.9: Proposed algorithms versus non-blind methods in SINR under SNR=15 dB.

3.5 Conclusion

This chapter has presented two blind adaptive filtering algorithms for the MAI suppression in DS/CDMA systems. Since the proposed algorithms are based on the parallel projection with the embedded constraint structure, they achieve closer points to the MMSE optimal filter than the existing methods at each iteration. Simulation results have demonstrated that the proposed algorithms exhibit excellent performance.

The extensive experiments in this chapter suggest that the A-PCP (see Sec. 3.3.4) and the EC-APSM may include excellent embedded constraint algorithms. Those two families of embedded constraint algorithms (i.e., A-PCP and EC-APSM) are expected to be useful not only in communications but also in a wide range of applications. In the presented simulations, we focus on the uniform weights ($\omega_\iota^{(k)} = \frac{1}{q}$, $\forall \iota \in \mathcal{I}_k$; see the previous section) for simplicity. For further improvements, an efficient strategic weighting technique such as the one shown in the following chapter would be effective. An efficient extension of the proposed algorithms to complex-valued cases has been presented in [140] in the adaptive

beamforming application. Moreover, in situations where good phase synchronization is difficult to obtain, one should use the differential PSK modulation; such issues have been addressed in [21]. Finally, in multipath fading environments, the proposed algorithms should be combined with channel estimation techniques, and must be robust against channel-estimation errors.

Appendix

3.A MMSE and MOE Detectors

Let us show a simple observation.

Observation 3.A.1. *Suppose (I) the auto-correlation matrix* $\boldsymbol{R}_{\boldsymbol{r}} := E\{\boldsymbol{r}[i]\boldsymbol{r}[i]^T\}$ *is full rank (*$\Rightarrow \boldsymbol{h}^*$ *is unique), and (II)* $\boldsymbol{y}_{\boldsymbol{r}b} := E\{\boldsymbol{r}[i]b_1[i]\} = \beta \boldsymbol{s}_1$, $\exists \beta \in \mathbb{R}$. *Then, for any given* $\alpha \in \mathbb{R}$,

$$\boldsymbol{h}^* = \frac{\boldsymbol{R}_{\boldsymbol{r}}^{-1}\boldsymbol{s}_1}{\boldsymbol{s}_1^T\boldsymbol{R}_{\boldsymbol{r}}^{-1}\boldsymbol{s}_1} = \underbrace{\underset{\boldsymbol{h}\in C_s}{\operatorname{argmin}} E\left\{(\langle \boldsymbol{h}, \boldsymbol{r}[i]\rangle - \alpha b_1[i])^2\right\}}_{=:\, \boldsymbol{h}_\alpha^*}. \tag{3.A.1}$$

Sketch of proof:

By the condition *(I)* and "Lagrangian multiplier" methodology (e.g., [61]), we can easily obtain

$$\boldsymbol{h}_\alpha^* = \frac{\boldsymbol{R}_{\boldsymbol{r}}^{-1}\boldsymbol{s}_1}{\boldsymbol{s}_1^T\boldsymbol{R}_{\boldsymbol{r}}^{-1}\boldsymbol{s}_1} + \alpha\left[\boldsymbol{R}_{\boldsymbol{r}}^{-1}\boldsymbol{y}_{\boldsymbol{r}b} - \frac{\boldsymbol{s}_1^T\boldsymbol{R}_{\boldsymbol{r}}^{-1}\boldsymbol{y}_{\boldsymbol{r}b}}{\boldsymbol{s}_1^T\boldsymbol{R}_{\boldsymbol{r}}^{-1}\boldsymbol{s}_1}\boldsymbol{R}_{\boldsymbol{r}}^{-1}\boldsymbol{s}_1\right],$$

and, by imposing the condition *(II)*, we readily verify

$$\boldsymbol{h}_\alpha^* = \boldsymbol{h}^* = \frac{\boldsymbol{R}_{\boldsymbol{r}}^{-1}\boldsymbol{s}_1}{\boldsymbol{s}_1^T\boldsymbol{R}_{\boldsymbol{r}}^{-1}\boldsymbol{s}_1}.$$

□

Remark 3.A.2.

(a) *Without the condition (I),* $\boldsymbol{h}_\alpha^*$ *is not necessarily unique.*

(b) *The condition (II) holds under slowly time-varying fading environments with the following assumption:* $E\{b_1[i]b_l[i]\} = 0$, $\forall l \in \{2, 3, \cdots, L\}$, *and* $E\{b_1[i]\boldsymbol{n}[i]\} = \mathbf{0}$.

(c) *The filters* $\boldsymbol{h}_0^*$ *and* $\boldsymbol{h}_{A_1}^*(= \boldsymbol{h}^*)$ *are called the MOE detector and the (constrained) MMSE detector, respectively. Observation* 3.A.1 *shows that the MMSE and MOE detectors coincide under (I) and (II).*

(d) *By* $\boldsymbol{h}^* = \boldsymbol{h}_\alpha^*$ $(\forall \alpha \in \mathbb{R})$ *under (I) and (II), a natural question would be: Does the set*

$$\widetilde{C}_\rho^{(k)}[i] := \left\{ \boldsymbol{h} \in \mathbb{R}^N : \left(\langle \boldsymbol{h}, \boldsymbol{r}[i] \rangle - \alpha \widehat{b}_{1,k}[i] \right)^2 \leq \rho \right\}$$

with an arbitrarily chosen α *contain the optimal filter* $\boldsymbol{h}^*$*? If "yes", we could get an optimistic conclusion that the amplitude estimation* $\widehat{A}_{1,k+1}$ *is not necessary. Unfortunately, however, the answer is "no", of which the reason is as follows. By* (3.3.2), $\langle \boldsymbol{h}, \boldsymbol{r}[i] \rangle - \alpha \widehat{b}_{1,k}[i]$ *has the term* $A_1 b_1[i] - \alpha \widehat{b}_{1,k}[i]$ *in addition to the terms of MAI and noise. Hence, bounding* $\left(\langle \boldsymbol{h}, \boldsymbol{r}[i] \rangle - \alpha \widehat{b}_{1,k}[i] \right)^2$ *by small* ρ *does not necessarily suppress MAI sufficiently (without amplifying noise severely) when* $|A_1 - \alpha| \gg 0$*, which implies, from the context between* (3.3.2) *and* (3.3.3)*, that* α *should be close to* A_1 *in order to ensure* $\boldsymbol{h}^* \in \widetilde{C}_\rho^{(k)}[i]$*. Therefore, high accuracy of the estimation of* A_1 *is essential for good steady-state performance.*

Chapter 4

Pairwise Optimal Weight Realization —Acceleration Technique for Set-Theoretic Adaptive Parallel Subgradient Projection Algorithm

Summary

The adaptive Parallel Subgradient Projection (PSP) algorithm has been proposed in 2002 as a set-theoretic adaptive filtering algorithm providing fast and stable convergence, robustness against noise, and low computational complexity by using weighted parallel projections onto multiple time-varying closed half-spaces.

In this chapter, we present a novel weighting technique named Pairwise Optimal Weight Realization (POWER) for further acceleration of the adaptive-PSP algorithm. A simple closed-form formula is derived to compute the projection onto the intersection of two closed half-spaces defined by a triplet of vectors. Using the formula inductively, the proposed weighting technique realizes a good direction of update. The resulting weights turn out to be *pairwise optimal* in a certain sense. The proposed algorithm has the inherently parallel structure composed of q primitive functions, hence its total computational complexity $O(qrN)$ is reduced

to $O(rN)$ with q concurrent processors (r: a constant positive integer, N: filter length). Numerical examples demonstrate that the proposed technique for $r = 1$ yields significantly faster convergence than not only adaptive-PSP with uniform weights, APA, and FNTF, but also the regularized RLS algorithm.

4.1 Introduction

Adaptive filtering is a fundamental tool with applications in many fields such as acoustics, communications, seismology, geophysics, astrophysics, and biomedicine [11, 56, 97]. The demand is increasing for algorithms with high speed of convergence to a reasonable approximation of the *estimandum* (system to be estimated). The adaptive Parallel Subgradient Projection (PSP) algorithm [129] has been introduced as an advanced *time-varying set-theoretic approach* to meet this growing demand. (As a different stream of set-theoretic methods, the so-called *Set-Membership* (SM) approach, e.g., SM-NLMS [51], Frequency-domain-SM-NLMS [52], has been independently developed.) The basic idea of this approach is as follows: (i) design multiple closed convex sets of which the intersection is sufficiently small but contains the estimandum with high probability and then (ii) find a point in the intersection. The approach does not essentially require the estimation of statistics of random processes, and hence it is expected to play a leading role in adaptive filtering for possibly nonstationary random processes. In this chapter, we present an inventive idea for further acceleration of the algorithm in [129] while keeping $O(rN)$ complexity [N: filter length, ($N \gg$) r: a positive constant]. To clarify the orientation of the present study from a wider viewpoint, we start with a brief review on the development of adaptive filtering algorithms.

The Recursive Least Squares (RLS) [56, 97] is a well-known family of adaptive filtering algorithms, which exhibit fast convergence even for highly colored (stationary) inputs. Unfortunately, however, RLS is computationally-intensive $O(N^2)$ [or $O(N^3)$ for its regularized version], and it is reported that the algorithm shows inferior performance to the classical Least Mean Square (LMS) algorithm [56] for nonstationary inputs [54, 56]. To achieve convergence as fast as RLS with $O(N)$ complexity, simplified versions, called Fast-RLS (FRLS), have been proposed [18, 23, 72]. It has been pointed out that FRLS suffers from intrinsic instability (see [50] and the references therein). A great deal of effort has been

devoted to stabilize FRLS with a moderate increase of computational complexity [7, 14, 40, 82, 105]. Moreover, a class of adaptive estimation algorithms called Fast Newton Transversal Filters (FNTF) has also been proposed [83] as an efficient extension of the method in [18] from AR(N) to AR(L_{FNTF}), $L_{\mathrm{FNTF}} \in \{0, 1, \cdots, N\}$, where AR($n$) stands for an autoregressive (AR) process of order n. Unfortunately, even if we employ the existing stabilized versions, the filter has a risk of divergence, and thus, monitoring and reinitialization are always required [11, p. 77], [14, 82, 105], [42, p. 40]. Numerical instability is still a common issue for the RLS-type algorithms to overcome.

The Affine Projection Algorithm (APA) [43, 56, 58, 89, 93, 95, 97, 103] can be interpreted as an example of a set-theoretic adaptive filtering algorithm. APA has originally been proposed around three decades ago [58] to increase the speed of convergence of the popular Normalized LMS (NLMS) algorithm [1, 56, 84, 97], particularly for highly-colored input signals. The APA algorithm is based on an iterative relaxed projection onto a series of linear varieties $V_k \subset \mathbb{R}^N$ ($k \in \mathbb{N}$: time index), which is generated as the intersection of a certain number, say $r \in \mathbb{N} \setminus \{0\}$, of hyperplanes that are determined by instantaneous input-output relations. If the estimandum $\boldsymbol{h}^* \in \mathbb{R}^N$ belongs to V_k, *monotonicity* $\|\boldsymbol{h}_{k+1} - \boldsymbol{h}^*\| \leq \|\boldsymbol{h}_k - \boldsymbol{h}^*\|$ holds (by the Pythagorean theorem), which is an important property for stable performance ($\boldsymbol{h}_k \in \mathbb{R}^N$: adaptive filter at time k). In noise free situations, $\boldsymbol{h}^*$ always belongs to V_k; thus the convergence speed can be raised by increasing r at the cost of $2rN + O(r^3)$ complexity (efficient versions called fast-APA, or FAP, have been proposed to reduce the complexity [50], [42, Chapter2], [43]). In noisy situations, an increase of r makes the membership probability Prob$\{\boldsymbol{h}^* \in V_k\}$ close to zero, thus causing serious instability (for details, see [129]). It is strongly desired to establish an algorithm simultaneously achieving (i) fast convergence, (ii) $O(N)$ computational complexity, (iii) numerical stability, and (iv) robustness against noise.

The *inherent parallelism*[1] concept [4, 17, 22, 24] would be a key in adaptive filtering problems as well [50], to break the persistent deadlock existing among the above requirements. The adaptive-PSP algorithm [129] enjoys exactly such

[1] As in the preface in [17], the term "inherently parallel algorithms" means those which are logically (i.e., in their mathematical formulations) parallel, not just parallelizable under some conditions, such as when the underlying problem is decomposable in a certain manner.

a parallel structure. Indeed, it utilizes, for each update, relaxed projections onto multiple, say $q \in \mathbb{N} \setminus \{0\}$, closed half-spaces $H^-_{\iota_1}(\boldsymbol{h}_k), \cdots, H^-_{\iota_q}(\boldsymbol{h}_k)$ ($\iota_j \leq k$, $\forall j = 1, 2, \cdots, q$) that contain $\boldsymbol{h}^*$ with high probability even in noisy situations (see Sec. 4.3.2). [To give a unified view of set-theoretic adaptive algorithms such as NLMS/APA/adaptive-PSP and their embedded versions, the Adaptive Projected Subgradient Method (APSM) has recently been established [124–126, 128] (see Chapter 2) and has successfully been applied to challenging real-world problems such as stereophonic echo cancellation [134, 138] (see Chapter 5), and blind multiuser detection in DS/CDMA systems [20, 132] (see Chapter 3).] Hence, monotonicity is highly expected to hold, opening a pathway to resolution of the noise sensitivity problem. So far, just for simplicity, uniform weights have commonly been used in adaptive-PSP, which suggests that further improvements of convergence could be achieved by establishing an effective weight design.

The goal of this chapter is to bring out the potential of the adaptive-PSP algorithm aggressively but without losing its computational efficiency. We propose an efficient adaptive weighting technique named *Pairwise Optimal Weight Realization* (POWER), which uses projection onto the intersection of two closed half-spaces as a basic tool for an efficient approximation of an ideal direction of update. We first present an expression of a pair of half-spaces with a triplet of vectors; by this expression a closed-form formula of the projection onto the intersection is derived. The proposed weighting technique inductively utilizes the formula to find a good direction of update. The resulting weights turn out to meet certain optimality conditions [see Sec. 4.3.3.C]. In addition, by engaging q processors, the overall complexity is kept $O(rN)$, imposed on each processor, including the weight design (see Secs. 4.3.4 and 4.6). There is a large variety of ways to utilize the proposed weighting technique, among which we present two algorithms named POWER-PSP type I and type II.

Numerical examples verify that the proposed technique significantly accelerates the convergence speed while inheriting the notable stability from adaptive-PSP. Moreover, the proposed algorithm achieves, with low computational cost, even faster convergence than the (computationally-intensive) regularized RLS algorithm as well as the APA and FNTF algorithms.

Preliminary short versions of this chapter were partially introduced at conferences [128, 137].

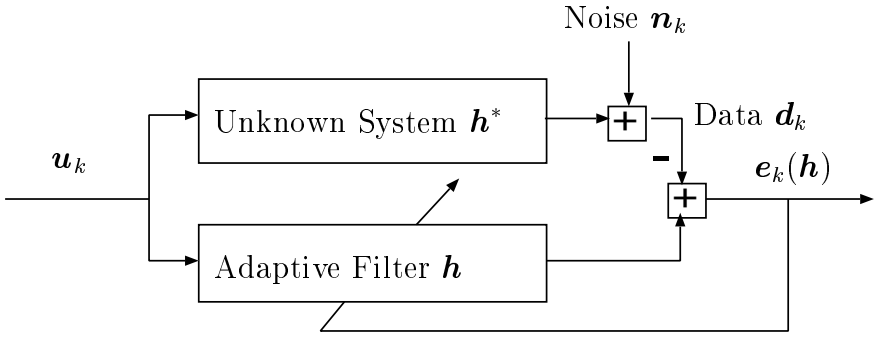


Figure 4.2.1: Adaptive filtering scheme.

4.2 Adaptive Filtering Problem

The adaptive filtering problem is addressed in a real Hilbert space equipped with the inner product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^T \boldsymbol{y}$, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{H} := \mathbb{R}^N$ ($N \in \mathbb{N}^*$), and its induced norm $\|\boldsymbol{x}\| := \left(\boldsymbol{x}^T \boldsymbol{x}\right)^{1/2}$, $\forall \boldsymbol{x} \in \mathcal{H}$ (see Fig. 4.2.1). Let $(u_k)_{k\in\mathbb{N}} \subset \mathbb{R}$ be the input sequence (k: time index), and define the sequence of input vectors $(\boldsymbol{u}_k)_{k\in\mathbb{N}} \subset \mathcal{H}$ as $\boldsymbol{u}_k := [u_k, u_{k-1}, \cdots, u_{k-N+1}]^T \in \mathcal{H}$, $\forall k \in \mathbb{N}$. For $r \in \mathbb{N}^*$, let $\boldsymbol{U}_k := [\boldsymbol{u}_k, \boldsymbol{u}_{k-1}, \cdots, \boldsymbol{u}_{k-r+1}] \in \mathbb{R}^{N\times r}$, $\forall k \in \mathbb{N}$. From $(\boldsymbol{U}_k)_{k\in\mathbb{N}}$, the observable data process $(\boldsymbol{d}_k)_{k\in\mathbb{N}} \subset \mathbb{R}^r$ is produced as $\boldsymbol{d}_k := \boldsymbol{U}_k^T \boldsymbol{h}^* + \boldsymbol{n}_k$, $\forall k \in \mathbb{N}$, where $\boldsymbol{h}^* \in \mathcal{H}$ is the estimandum (unknown system) and $\boldsymbol{n}_k := [n_k, n_{k-1}, \cdots, n_{k-r+1}]^T \in \mathbb{R}^r$, $\forall k \in \mathbb{N}$, is the noise vector. The estimation residual function is defined as $\boldsymbol{e}_k$: $\mathcal{H} \to \mathbb{R}^r$, $\boldsymbol{h} \mapsto \boldsymbol{U}_k^T \boldsymbol{h} - \boldsymbol{d}_k$, $\forall k \in \mathbb{N}$. The problem is to approximate $\boldsymbol{h}^*$ by the adaptive filter $\boldsymbol{h} \in \mathcal{H}$ with the input output relations $\{(\boldsymbol{U}_\iota, \boldsymbol{d}_\iota)\}_{\iota\le k}$.

4.3 Proposed Pairwise Optimal Weight Realization Technique

In Sec. 4.3.1, we first present a simple explicit formula to give the projection onto the intersection of a pair of closed half-spaces, which are defined by a triplet of vectors. In Sec. 4.3.2, by using the formula as a basic tool, we present the Pairwise Optimal Weight Realization (POWER) technique, a computationally-efficient scheme to yield a good direction of update for more than two half-spaces. In Sec. 4.3.3, for the proposed POWER technique, we finally present A) geometric

interpretations, B) weight realizations, and C) optimality.

For convenience, we define, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{H}$,

$$\begin{aligned} \Pi(\boldsymbol{x}, \boldsymbol{y}) &:= \{\boldsymbol{z} \in \mathcal{H} : \langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{z} - \boldsymbol{y} \rangle = 0\} \subset \mathcal{H}, \\ \Pi^-(\boldsymbol{x}, \boldsymbol{y}) &:= \{\boldsymbol{z} \in \mathcal{H} : \langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{z} - \boldsymbol{y} \rangle \leq 0\} \subset \mathcal{H}. \end{aligned} \tag{4.3.1}$$

Remark 4.3.1. *If $\boldsymbol{x} = \boldsymbol{y}$, $\Pi(\boldsymbol{x}, \boldsymbol{y}) = \Pi^-(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{H}$, otherwise, $\Pi^-(\boldsymbol{x}, \boldsymbol{y}) \not\ni \boldsymbol{x}$ is a closed half-space with its boundary hyperplane $\Pi(\boldsymbol{x}, \boldsymbol{y})$. $P_{\Pi^-(\boldsymbol{x}, \boldsymbol{y})}(\boldsymbol{x}) = \boldsymbol{y}$ holds in any case.*

4.3.1 Projection onto Intersection of Pair of Closed Half-Spaces

We first define two important operators.

Definition 4.3.2. *Given an ordered triplet[2] $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{b}) \in \mathcal{H}^3$ s.t. $\Pi^-(\boldsymbol{s}, \boldsymbol{a}) \cap \Pi^-(\boldsymbol{s}, \boldsymbol{b}) \neq \emptyset$, we define the following two operators:*

(a) *$\mathcal{Q} : [0, 1] \times [0, \infty) \times \mathcal{H}^3 \to \mathcal{H}$ is defined by $\mathcal{Q}(\omega, \mu, \boldsymbol{s}, \boldsymbol{a}, \boldsymbol{b}) := \boldsymbol{s} + \mu\,[\omega \boldsymbol{a} + (1 - \omega)\boldsymbol{b} - \boldsymbol{s}]$, and*

(b) *$\mathcal{P} : \mathcal{H}^3 \to \mathcal{H}$ is defined by $\mathcal{P}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{b}) := P_{\Pi^-(\boldsymbol{s}, \boldsymbol{a}) \cap \Pi^-(\boldsymbol{s}, \boldsymbol{b})}(\boldsymbol{s}) \in \mathcal{H}$.*

We show a simple explicit formula of $\mathcal{P}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{b})$ below.

Proposition 4.3.3. *Given $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{b}) \in \mathcal{H}^3$, let $\xi := \|\boldsymbol{a} - \boldsymbol{s}\|^2$, $\zeta := \|\boldsymbol{b} - \boldsymbol{s}\|^2$, and $\eta := \langle \boldsymbol{a} - \boldsymbol{s}, \boldsymbol{b} - \boldsymbol{s} \rangle$. Then, the following hold.*

(a) *The following conditions are equivalent:*

$$\text{C1}: \quad \Pi^-(\boldsymbol{s}, \boldsymbol{a}) \cap \Pi^-(\boldsymbol{s}, \boldsymbol{b}) = \emptyset, \quad \text{C2}: \quad \eta = -\sqrt{\xi\zeta} \neq 0$$

(b) *If $\Pi^-(\boldsymbol{s}, \boldsymbol{a}) \cap \Pi^-(\boldsymbol{s}, \boldsymbol{b}) \neq \emptyset$, we have an expression:*

$$\mathcal{P}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{b}) = \mathcal{Q}(\omega^*, \mu^*, \boldsymbol{s}, \boldsymbol{a}, \boldsymbol{b}), \tag{4.3.2}$$

[2] In [5], a different three-point expression is presented for a pair of closed half-spaces. Our three-point expression covers, e.g., the case $\boldsymbol{b} \in \Pi^-(\boldsymbol{a}, \boldsymbol{c})$, while the one in [5] does *not*.

where

$$\mu^* := \begin{cases} 1, & \text{if } \eta \geq \min(\xi,\zeta), \\ \dfrac{2\xi\zeta-(\xi+\zeta)\eta}{\xi\zeta-\eta^2} > 0, & \text{otherwise}, \end{cases} \tag{4.3.3}$$

$$\omega^* := \begin{cases} 1, & \text{if } \eta \geq \zeta, \\ 0, & \text{if } \zeta > \eta \geq \xi, \\ \dfrac{\zeta(\xi-\eta)}{2\xi\zeta-(\xi+\zeta)\eta} \in [0,1], & \text{if } \eta < \min(\xi,\zeta). \end{cases} \tag{4.3.4}$$

Proof: See Sec. 4.4.1. □

Another characterization of $\mathcal{P}(\boldsymbol{s},\boldsymbol{a},\boldsymbol{b})$ is given in Proposition 4.3.10.

4.3.2 Pairwise Optimal Weight Realization (POWER)

The adaptive-PSP algorithm [129] uses weighted parallel projections onto multiple time-varying half-spaces, which are designed as follows. Define first the following closed convex sets called *stochastic property sets*:

$$C_k(\rho) := \left\{\boldsymbol{h} \in \mathcal{H} : g_k(\boldsymbol{h}) := \|\boldsymbol{e}_k(\boldsymbol{h})\|^2 - \rho \leq 0\right\}, \forall k \in \mathbb{N}, \tag{4.3.5}$$

where $\rho \geq 0$ is called the *inflation parameter*. Noting that $\boldsymbol{h}^* \in C_k(\rho) \Leftrightarrow \|\boldsymbol{n}_k\|^2 \leq \rho$, the use of a large ρ guarantees the membership $\boldsymbol{h}^* \in C_k(\rho)$ with high probability, which leads to stable behavior of the algorithm (see [129]). To reduce the computational complexity, the projection $P_{H_k^-(\boldsymbol{h})}(\boldsymbol{h}) \cong P_{C_k(\rho)}(\boldsymbol{h})$, onto an approximating closed half-space[3] $H_k^-(\boldsymbol{h}) \supset C_k(\rho)$, is used with the following simple closed-form expression [$g_k(\boldsymbol{h})$ is defined in (4.3.5)]:

$$P_{H_k^-(\boldsymbol{h})}(\boldsymbol{h}) = \begin{cases} \boldsymbol{h} - \dfrac{g_k(\boldsymbol{h})}{\|\nabla g_k(\boldsymbol{h})\|^2}\nabla g_k(\boldsymbol{h}), & \text{if } \boldsymbol{h} \notin H_k^-(\boldsymbol{h}), \\ \boldsymbol{h}, & \text{otherwise}, \end{cases} \tag{4.3.6}$$

$$H_k^-(\boldsymbol{h}) := \{\boldsymbol{x} \in \mathcal{H} : \langle \boldsymbol{x}-\boldsymbol{h}, \nabla g_k(\boldsymbol{h})\rangle + g_k(\boldsymbol{h}) \leq 0\}. \tag{4.3.7}$$

[3]Other closed half-spaces are also presented in [87] [see also Example 6.3.2-(b)], which can also be used with the proposed weighting technique.

Note that $\nabla g_k(\boldsymbol{h}) = \boldsymbol{U}_k \boldsymbol{e}_k(\boldsymbol{h})$, $\forall k \in \mathbb{N}$. Since $\mathrm{Prob}\{\boldsymbol{h}^* \in \bigcap H_j^-(\boldsymbol{h})\}$ is expected to be sufficiently high, $P_{\bigcap H_j^-(\boldsymbol{h})}(\boldsymbol{h})$ would be a natural candidate for appropriate approximations of $\boldsymbol{h}^*$. For more than two half-spaces, however, finding the weights to give $P_{\bigcap H_j^-(\boldsymbol{h})}(\boldsymbol{h})$ is computationally intensive (cf. [3, p. 86]). In such a case, more reasonable and strategic weight design than uniform weights has not yet been established.

As an efficient, aggressive weighting strategy for adaptive-PSP, we propose to apply Proposition 4.3.3 in a pairwise manner. We call this the *POWER* technique, whereby we compute reasonable weights that realize a good direction of update. How to exploit Proposition 4.3.3 could depend on applications. In the following, two simple realizations of the POWER technique are presented.

A. POWER-PSP Type I Algorithm

We introduce a realization of the POWER technique named the POWER-PSP Type I algorithm, which is constructed in M stages. Now, given $q \in \mathbb{N}^*$, define the control sequence [22] $(\bar{\mathcal{I}}_k =)\ \mathcal{I}_k^{(0)} := \{\iota_{k,1}, \iota_{k,2}, \cdots, \iota_{k,q}\} \subset \mathbb{N}$, $\forall k \in \mathbb{N}$, where $\iota_{k,i}$ $(i = 1, 2, \cdots, q)$ denotes the time index used at the 0th stage at time k. We use herein the notation $\bar{\mathcal{I}}_k$ for the proposed algorithm instead of $\mathcal{I}_k$ since some of the overall weights $\{w_j^{(k)}\}_{j \in \bar{\mathcal{I}}_k}$ realized by the proposed algorithm can be zero when observed data are *inconsistent* (see Proposition 4.3.7 and the paragraph just after Algorithm 4.3.4), whereas the weights used in adaptive-PSP are positive. Next define inductively the control sequence used at the mth stage $(m = 1, 2, \cdots, M)$ as $\mathcal{I}_k^{(m)} \subset \{(\iota_1, \iota_2) : \iota_1, \iota_2 \in \mathcal{I}_k^{(m-1)}, \iota_1 \neq \iota_2\}$ $(\forall k \in \mathbb{N})$ satisfying $1 = |\mathcal{I}_k^{(M)}| < |\mathcal{I}_k^{(M-1)}| \leq \cdots \leq |\mathcal{I}_k^{(1)}| \leq |\mathcal{I}_k^{(0)}| = q$. The proposed algorithm is given as follows.

Algorithm 4.3.4 (POWER-PSP Type I). *Suppose that a sequence of closed convex sets $(C_k(\rho))_{k \in \mathbb{N}}$ is defined as in (4.3.5) and that $\left(H_j^-(\boldsymbol{h}_k)\right)_{j \in \bar{\mathcal{I}}_k}$, $\forall k \in \mathbb{N}$, is defined as in (4.3.7). Let $\boldsymbol{h}_0 \in \mathcal{H}$ be an arbitrarily chosen initial vector and $\lambda_k \in [0, 2]$ the step size. Then, a sequence $(\boldsymbol{h}_k)_{k \in \mathbb{N}} \subset \mathcal{H}$ is generated iteratively; at iteration k, $\boldsymbol{h}_k$ is updated to $\boldsymbol{h}_{k+1}$ by the following three steps.*

Step 1) *0th Stage: Projections onto q Half-Spaces*

$$\boldsymbol{h}_{k,\iota}^{(0)} := P_{H_\iota^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k), \quad \forall k \in \mathbb{N},\ \forall \iota \in \bar{\mathcal{I}}_k, \tag{4.3.8}$$

where $P_{H_\iota^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k)$ *is computed by* (4.3.6).

Step 2) *1st to Mth Stage: Find a Good Direction*

for $m := 1$ **to** M **do**

$$\boldsymbol{h}_{k,\iota}^{(m)} := \begin{cases} \boldsymbol{h}_k, & \text{if } \eta_{k,\iota}^{(m)} = -\sqrt{\xi_{k,\iota}^{(m)}\zeta_{k,\iota}^{(m)}} \neq 0, \\ \mathcal{P}\left(\boldsymbol{h}_k, \boldsymbol{h}_{k,\iota_1}^{(m-1)}, \boldsymbol{h}_{k,\iota_2}^{(m-1)}\right), & \text{otherwise}, \end{cases}$$
$$\forall k \in \mathbb{N},\ \forall \iota = (\iota_1, \iota_2) \in \mathcal{I}_k^{(m)}, \tag{4.3.9}$$

where $\eta_{k,\iota}^{(m)} := \left\langle \boldsymbol{h}_{k,\iota_1}^{(m-1)} - \boldsymbol{h}_k, \boldsymbol{h}_{k,\iota_2}^{(m-1)} - \boldsymbol{h}_k \right\rangle$, $\xi_{k,\iota}^{(m)} := \left\| \boldsymbol{h}_{k,\iota_1}^{(m-1)} - \boldsymbol{h}_k \right\|^2$, *and* $\zeta_{k,\iota}^{(m)} := \left\| \boldsymbol{h}_{k,\iota_2}^{(m-1)} - \boldsymbol{h}_k \right\|^2$.
$\mathcal{P}\left(\boldsymbol{h}_k, \boldsymbol{h}_{k,\iota_1}^{(m-1)}, \boldsymbol{h}_{k,\iota_2}^{(m-1)}\right)$ *is computed by Proposition* 4.3.3. *The reason why we use* $\mathcal{P}\left(\boldsymbol{h}_k, \boldsymbol{h}_{k,\iota_1}^{(m-1)}, \boldsymbol{h}_{k,\iota_2}^{(m-1)}\right)$ *is clearly discussed in Remark* 4.3.12.

end;

Step 3) *Final Stage: Update to the Good Direction*

$$\boldsymbol{h}_{k+1} := \boldsymbol{h}_k + \lambda_k(\boldsymbol{h}_{k,\iota}^{(M)} - \boldsymbol{h}_k),\ \forall k \in \mathbb{N}. \tag{4.3.10}$$

From Proposition 4.3.3-(a) and Lemma 4.3.9, we can verify that $\eta_{k,\iota}^{(m)} = -\sqrt{\xi_{k,\iota}^{(m)}\zeta_{k,\iota}^{(m)}} \neq 0 \Leftrightarrow \exists \delta < 0$ s.t. $\boldsymbol{h}_{k,\iota_1}^{(m-1)} - \boldsymbol{h}_k = \delta(\boldsymbol{h}_{k,\iota_2}^{(m-1)} - \boldsymbol{h}_k) \neq \boldsymbol{0}$, $\forall k \in \mathbb{N}$, $\forall \iota = (\iota_1, \iota_2) \in \mathcal{I}_k^{(m)}$, $\forall m \in \{1, 2, \cdots, M\}$. This happens when we receive inconsistent data, since the data indicated by ι_1 suggests the opposite direction from the ones indicated by ι_2. In such a case, the proposed algorithm does not update the filter, hence there is no problem, which is the same for the POWER-PSP Type I I algorithm presented in the following.

A systematic design of control sequences in Algorithm 4.3.4 is given below.

Example 4.3.5 (Binary-tree construction). *Given* $\mathcal{I}_k^{(0)}$ $(= \bar{\mathcal{I}}_k)$, $\forall k \in \mathbb{N}$, *we suggest a systematic design of control sequences* $(\mathcal{I}_k^{(m)})_{m=1}^M$ *as shown in Fig.* 4.3.2-(a), *which we call the binary-tree construction of Algorithm* 4.3.4. *For example, with* $(\widetilde{\mathcal{I}}_k =)\mathcal{I}_k^{(0)} = \{1, 2, \cdots, 8\}$, *the control sequences at each stage are given as* $\mathcal{I}_k^{(1)} = \{(1,2), (3,4), (5,6), (7,8)\}$, $\mathcal{I}_k^{(2)} = \{[(1,2), (3,4)], [(5,6), (7,8)]\}$, *and* $\mathcal{I}_k^{(3)} = \{[[(1,2), (3,4)], [(5,6), (7,8)]]\}$ *(*$\mathcal{I}_k^{(3)}$ *is a singleton). Note that the pairs are not necessarily selected successively, e.g.,* $\mathcal{I}_k^{(1)}$ *can be* $\{(1,5), (2,6), (3,7), (4,8)\}$.

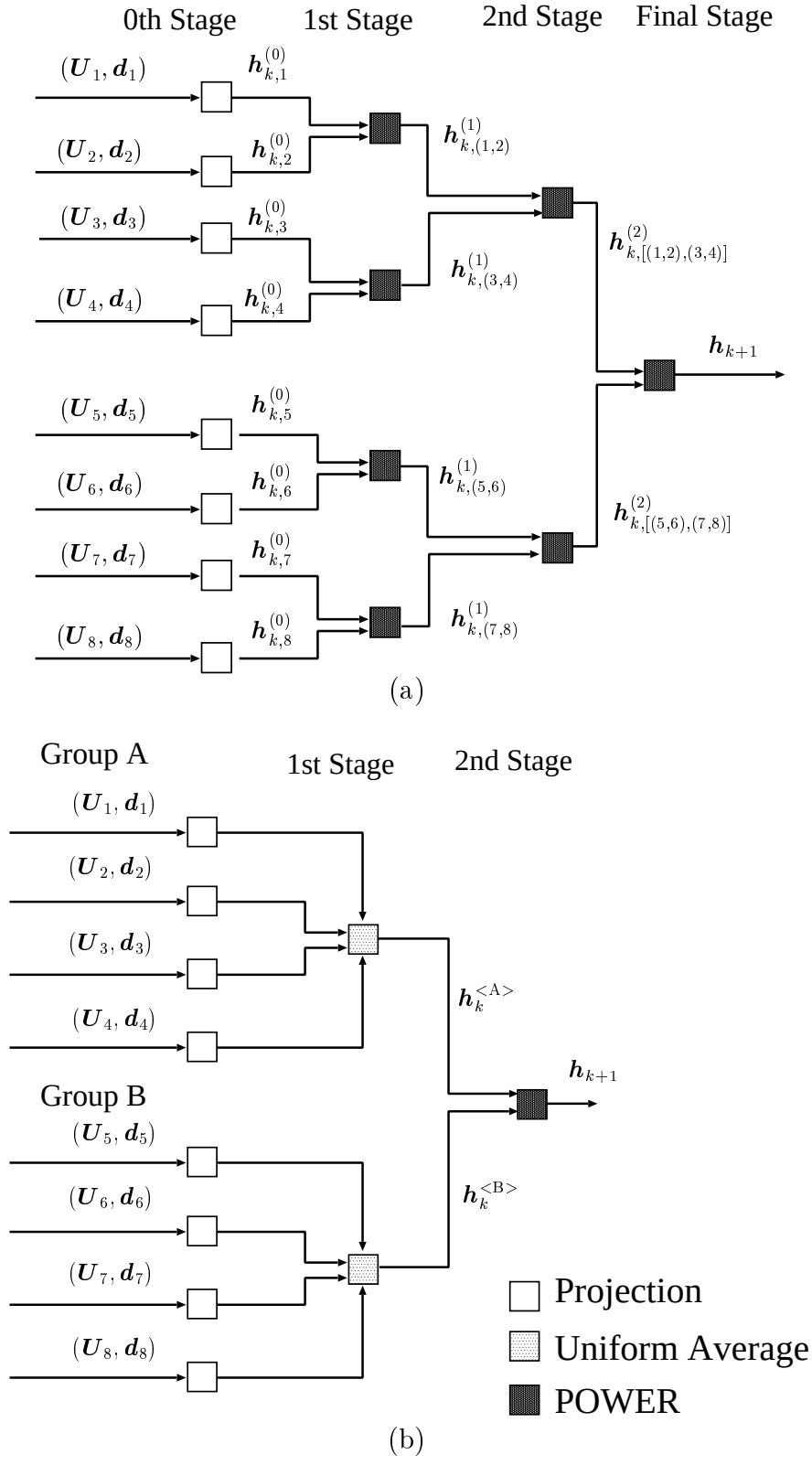


Figure 4.3.2: Simple system models with eight parallel processors ($q = 4$) to implement (a) POWER-PSP I and (b) POWER-PSP II.

B. POWER-PSP Type II Algorithm

We introduce an efficient realization of the POWER technique named the POWER-PSP Type II algorithm. Let us define two groups of closed half-spaces, say group A: $\left(H_j^-(\boldsymbol{h}_k)\right)_{j\in\mathcal{I}_k^{<\mathrm{A}>}}$ and group B: $\left(H_j^-(\boldsymbol{h}_k)\right)_{j\in\mathcal{I}_k^{<\mathrm{B}>}}$, where $\mathcal{I}_k^{<\mathrm{A}>}, \mathcal{I}_k^{<\mathrm{B}>} \neq \emptyset$, $\mathcal{I}_k^{<\mathrm{A}>} \cup \mathcal{I}_k^{<\mathrm{B}>} = \bar{\mathcal{I}}_k$ and $\mathcal{I}_k^{<\mathrm{A}>} \cap \mathcal{I}_k^{<\mathrm{B}>} = \emptyset$. At the 1st stage, for each group, take the uniformly weighted average of projections onto $\left(H_j^-(\boldsymbol{h}_k)\right)_{j\in\mathcal{I}_k^{<\mathrm{g}>}}$ ($\mathrm{g}\in\{\mathrm{A},\mathrm{B}\}$), say $\boldsymbol{h}_k^{<\mathrm{A}>}$ and $\boldsymbol{h}_k^{<\mathrm{B}>}$, for saving the computational complexity. At the 2nd stage, the filter $\boldsymbol{h}_k$ is updated by exploiting $\mathcal{P}(\boldsymbol{h}_k, \boldsymbol{h}_k^{<\mathrm{A}>}, \boldsymbol{h}_k^{<\mathrm{B}>})$ with Proposition 4.3.3. The algorithm is given below.

Algorithm 4.3.6 (POWER-PSP Type II). *Suppose that a sequence of closed convex sets $(C_k(\rho))_{k\in\mathbb{N}}$ is defined as in (4.3.5) and that $\left(H_j^-(\boldsymbol{h}_k)\right)_{j\in\bar{\mathcal{I}}_k}$, $\forall k\in\mathbb{N}$, is defined as in (4.3.7). Let $\boldsymbol{h}_0\in\mathcal{H}$ be an arbitrarily chosen initial vector and $\lambda_k\in[0,2]$ the step size. Then, a sequence $(\boldsymbol{h}_k)_{k\in\mathbb{N}}\subset\mathcal{H}$ is generated iteratively; at iteration k, $\boldsymbol{h}_k$ is updated to $\boldsymbol{h}_{k+1}$ as follows.*

1st Stage)

$$\boldsymbol{h}_k^{<\mathrm{g}>} := \boldsymbol{h}_k + \mathcal{M}_k^{<\mathrm{g}>}\left[\sum_{j\in\mathcal{I}_k^{<\mathrm{g}>}} w_k^{<\mathrm{g}>} P_{H_j^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k) - \boldsymbol{h}_k\right], \forall k\in\mathbb{N},\ \forall \mathrm{g}\in\{\mathrm{A},\mathrm{B}\}, \tag{4.3.11}$$

where $w_k^{<\mathrm{g}>} = 1/|\mathcal{I}_k^{<\mathrm{g}>}|$ is the uniform weight (which does not depend on j) and $\mathcal{M}_k^{<\mathrm{g}>}$ is a constant to determine appropriate relaxation, defined as

$$\mathcal{M}_k^{<\mathrm{g}>} := \begin{cases} \dfrac{\sum_{j\in\mathcal{I}_k^{<\mathrm{g}>}} w_k^{<\mathrm{g}>}\left\|P_{H_j^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k) - \boldsymbol{h}_k\right\|^2}{\left\|\sum_{j\in\mathcal{I}_k^{<\mathrm{g}>}} w_k^{<\mathrm{g}>} P_{H_j^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k) - \boldsymbol{h}_k\right\|^2}, & \text{if } \boldsymbol{h}_k \notin \bigcap_{j\in\mathcal{I}_k^{<\mathrm{g}>}} H_j^-(\boldsymbol{h}_k), \\ 1, & \text{otherwise.} \end{cases}$$

2nd Stage)

$$\boldsymbol{h}_{k+1} := \begin{cases} \boldsymbol{h}_k, & \text{if } \eta_k = -\sqrt{\xi_k\zeta_k} \neq 0, \\ \boldsymbol{h}_k + \lambda_k\left[\mathcal{P}\left(\boldsymbol{h}_k, \boldsymbol{h}_k^{<\mathrm{A}>}, \boldsymbol{h}_k^{<\mathrm{B}>}\right) - \boldsymbol{h}_k\right], & \text{otherwise,} \end{cases} \quad \forall k\in\mathbb{N}, \tag{4.3.12}$$

where $\eta_k := \left\langle \boldsymbol{h}_k^{<\mathrm{A}>} - \boldsymbol{h}_k, \boldsymbol{h}_k^{<\mathrm{B}>} - \boldsymbol{h}_k \right\rangle$, $\xi_k := \left\| \boldsymbol{h}_k^{<\mathrm{A}>} - \boldsymbol{h}_k \right\|^2$ *and* $\zeta_k := \left\| \boldsymbol{h}_k^{<\mathrm{B}>} - \boldsymbol{h}_k \right\|^2$.
The reason why we use $\mathcal{P}\left(\boldsymbol{h}_k, \boldsymbol{h}_k^{<\mathrm{A}>}, \boldsymbol{h}_k^{<\mathrm{B}>}\right)$ *is clearly discussed in Remark* 4.3.12.

A simple system model to implement Algorithm 4.3.6 is depicted in Fig. 4.3.2-(b), in which the control sequences are defined as $\mathcal{I}_k^{<\mathrm{A}>} = \{1, 2, 3, 4\}$ and $\mathcal{I}_k^{<\mathrm{B}>} = \{5, 6, 7, 8\}$. It is seen that POWER-PSP II is computationally more efficient than POWER-PSP I. As seen from Propositions 4.3.7 and 4.3.8 in the following section, Algorithms 4.3.4 and 4.3.6 can be interpreted as the adaptive-PSP algorithm with specially designed weights. Since adaptive-PSP is derived by APSM [125], [126], the proposed algorithms inherit the notable properties from APSM, e.g., *monotone approximation* (which implies the numerical stability), *asymptotic optimality*, *strong convergence*, etc. (see [126, Theorem 2]). Moreover, the proposed algorithms can be implemented with $O(rN)$ computational complexity by q parallel processors (see Secs. 4.3.4 and 4.6).

4.3.3 Properties of POWER-PSP I and POWER-PSP II

A. Geometric Interpretation

To see how the proposed algorithms and uniform weight PSP (UW-PSP), i.e., adaptive-PSP with the weights $w_j^{(k)} = 1/|\mathcal{I}_k|$, perform, we show a geometric interpretation of the algorithms in Fig. 4.3.3. The step size and the number of parallel projections (or the number of closed half-spaces used in each update) are set to $\lambda_k = 1$ and $q = 4$, respectively. The control sequences are designed as $\bar{\mathcal{I}}_k = \mathcal{I}_k^{(0)} = \{1, 2, 3, 4\}$, $\mathcal{I}_k^{(1)} = \{(1, 2), (3, 4)\}$, and $\mathcal{I}_k^{(2)} = \{[(1, 2), (3, 4)]\}$ for Type I, and $\mathcal{I}_k^{<\mathrm{A}>} = \{1, 2\}$ and $\mathcal{I}_k^{<\mathrm{B}>} = \{3, 4\}$ for Type II. In Fig. 4.3.3, the shaded triangle area shows the "target" set, i.e., the intersection $\bigcap_{j \in \bar{\mathcal{I}}_k} H_j^-(\boldsymbol{h}_k)$. The triangle automatically contains $C_0 := \bigcap_{j \in \bar{\mathcal{I}}_k} C_j(\rho)$, which is assumed to contain the estimandum $\boldsymbol{h}^*$.

Referring to Fig. 4.3.3-(c), UW-PSP: (i) computes projections (thin dotted lines) onto four closed half-spaces H_1^-, H_2^-, H_3^- and H_4^-; (ii) takes the uniformly weighted average (thick dotted lines); and, (iii) moves in the direction of the average (arrow). The update is not in a good direction towards the target triangle. POWER-PSP II, on the other hand, achieves a better direction of update than

UW-PSP. Referring to Fig. 4.3.3-(b), POWER-PSP II: (i) computes the projections (thin dotted lines); (ii) moves in the direction of the uniformly weighted average of the projections in each group (two dotted arrows); and, (iii) constructs new closed half-spaces $\Pi^-(\boldsymbol{h}_k, \boldsymbol{h}_k^{<\mathrm{A}>})$ and $\Pi^-(\boldsymbol{h}_k, \boldsymbol{h}_k^{<\mathrm{B}>})$ (bounded by thick dotted lines) and moves towards the projection onto their intersection (thick arrow). Furthermore, POWER-PSP I achieves the ideal direction in this simple case, i.e., the projection onto the target triangle. Referring to Fig. 4.3.3-(a), POWER-PSP I: (i) computes the projections (thin dotted lines); (ii) computes the projection (dotted arrows) onto the intersection of each pair of closed half-spaces, (H_1^-, H_2^-) and (H_3^-, H_4^-); and, (iii) moves towards the projection (thick arrow) onto the intersection of new closed half-spaces $\Pi^-(\boldsymbol{h}_k, \boldsymbol{h}_{k,(1,2)}^{(1)})$ and $\Pi^-(\boldsymbol{h}_k, \boldsymbol{h}_{k,(3,4)}^{(1)})$ (bounded by thick dotted lines).

B. Weight Realizations

Let us present the overall weights realized by the POWER-PSP I algorithm below.

Proposition 4.3.7 (Weight realization by POWER-PSP I). *Let $(\boldsymbol{h}_k)_{k\in\mathbb{N}} \subset \mathcal{H}$ be a sequence of filtering vectors generated by Algorithm 4.3.4. Then, $\boldsymbol{h}_{k+1}$ is represented in the form of adaptive-PSP with the weights $w_j^{(k)} = w_{j,\iota,M}^{(k)}$, $j \in \mathcal{I}_k$, $\forall k \in \mathbb{N}$, defined by the following simple recursive form $\left[\mathcal{I}_k := \{j \in \bar{\mathcal{I}}_k : w_{j,\iota,M}^{(k)} > 0\}\right]$:*
if $\eta_{k,\iota}^{(m)} = -\sqrt{\xi_{k,\iota}^{(m)} \zeta_{k,\iota}^{(m)}} \neq 0$, then $w_{j,\iota,m}^{(k)} = 0$ ($\forall m = 1, 2, \cdots, M$, $\forall \iota \in \mathcal{I}_k^{(m)}$, $\forall j \in \bar{\mathcal{I}}_k$), otherwise,

$$
\begin{aligned}
w_{j,\iota,1}^{(k)} &:= \begin{cases} \omega^*, & \text{if } \iota = (j, \cdot), \\ 1-\omega^*, & \text{if } \iota = (\cdot, j), \\ 0, & \text{otherwise}, \end{cases} \qquad \forall j \in \bar{\mathcal{I}}_k,\ \forall \iota \in \mathcal{I}_k^{(1)}, \\
w_{j,\iota,m}^{(k)} &:= \frac{\omega^* \mu_1^* w_{j,\iota_1,m-1}^{(k)} + (1-\omega^*)\mu_2^* w_{j,\iota_2,m-1}^{(k)}}{\omega^* \mu_1^* + (1-\omega^*)\mu_2^*}, \\
&\qquad\qquad \forall j \in \bar{\mathcal{I}}_k, \forall \iota = (\iota_1, \iota_2) \in \mathcal{I}_k^{(m)},\ \forall m = 2, 3, \cdots, M,
\end{aligned}
$$

where ω^* for $w_{j,\iota,m}^{(k)}$ ($\forall m = 1, 2, \cdots, M$) denotes the weight to calculate $\boldsymbol{h}_{k,\iota}^{(m)}$ $\left[= \mathcal{P}(\boldsymbol{h}_k, \boldsymbol{h}_{k,\iota_1}^{(m-1)}, \boldsymbol{h}_{k,\iota_2}^{(m-1)})\right]$ and μ_i^* ($i = 1, 2$) for $w_{j,\iota,m}^{(k)}$ ($\forall m = 2, 3, \cdots, M$) denotes

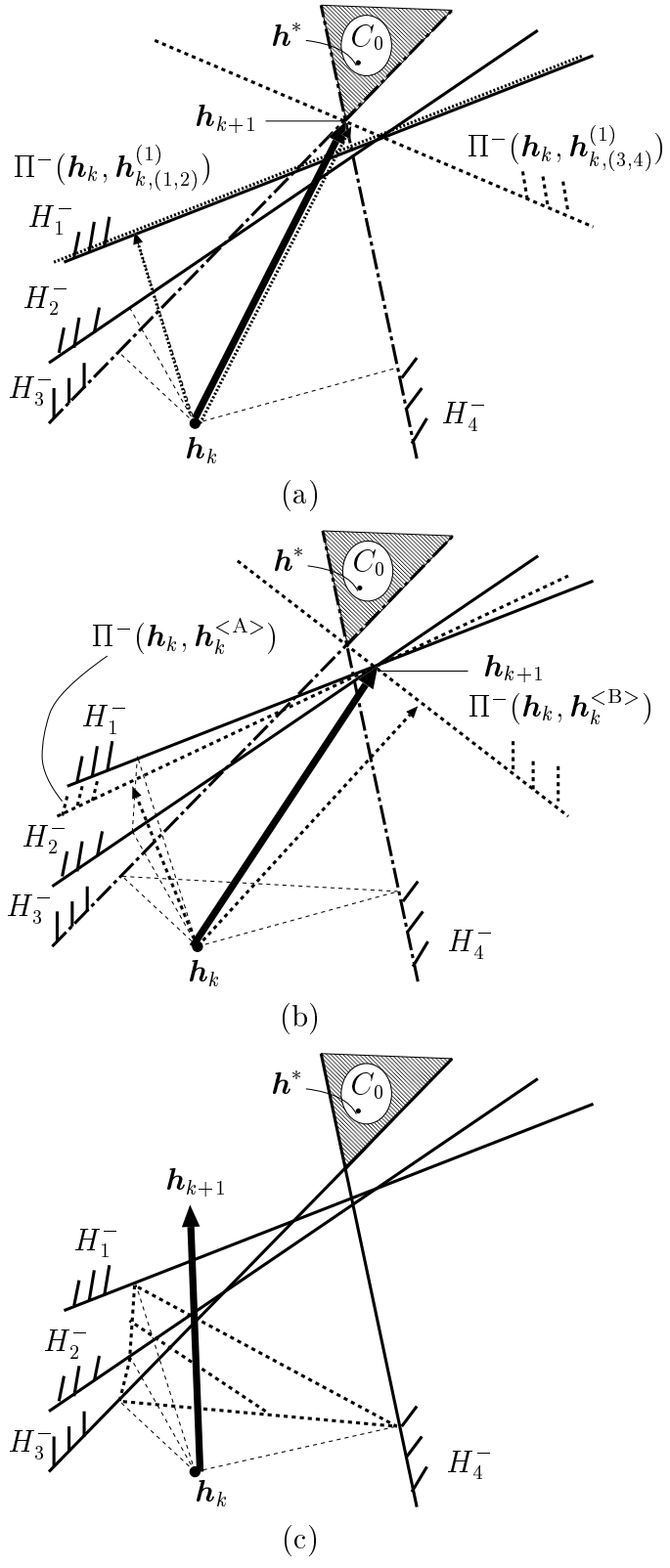


Figure 4.3.3: A geometric interpretation of (a) POWER-PSP I, (b) POWER-PSP II, and (c) UW-PSP algorithms. In this example, $\Pi^-(\boldsymbol{h}_k, \boldsymbol{h}^{(1)}_{k,(1,2)}) = H_1^-$.

the relaxation parameter to calculate $\boldsymbol{h}_{k,\iota_i}^{(m-1)}$ *[see* (4.3.3) *and* (4.3.4)*]. Note that, since* $|\mathcal{I}_k^{(M)}| = 1$*, the overall weights* $w_{j,\iota,M}^{(k)}$*,* $\forall j \in \mathcal{I}_k$*, do not depend on* ι*.*

Proof: The proof is omitted because it is verified by simple algebra. □

The weights $w_j^{(k)}$, $j \in \mathcal{I}_k$, realized by POWER-PSP I satisfy $w_j^{(k)} \in (0, 1]$ and $\sum_{j\in\mathcal{I}_k} w_j^{(k)} = 1$ if $\mathcal{I}_k \neq \emptyset$, which holds with a few exceptions that occur only when we receive inconsistent data, e.g., $\eta_{k,\iota}^{(1)} = -\sqrt{\xi_{k,\iota}^{(1)}\zeta_{k,\iota}^{(1)}} \neq 0$, $\forall \iota \in \mathcal{I}_k^{(1)}$ (see after Algorithm 4.3.4).

The weights realized by the POWER-PSP II algorithm are given below.

Proposition 4.3.8 (Weight realization by POWER-PSP II)**.** *Let* $(\boldsymbol{h}_k)_{k\in\mathbb{N}} \subset \mathcal{H}$ *be a sequence of filtering vectors generated by Algorithm* 4.3.6*. Then,* $\boldsymbol{h}_{k+1}$ *is represented in the form of adaptive-PSP with the weights* $w_j^{(k)}$*,* $j \in \mathcal{I}_k$*, given as follows* $\left[\mathcal{I}_k := \{j \in \mathcal{I}_k^{<\mathrm{A}>} \cup \mathcal{I}_k^{(\mathrm{B})} : w_j^{(k)} > 0\}\right]$*:*
if $\eta_k = -\sqrt{\xi_k\zeta_k} \neq 0$*, then* $w_j^{(k)} = 0$*,* $\forall j \in \bar{\mathcal{I}}_k (= \mathcal{I}_k^{<\mathrm{A}>} \cup \mathcal{I}_k^{(\mathrm{B})})$*, otherwise,*

$$w_j^{(k)} = \begin{cases} \dfrac{\omega^* \mathcal{M}_k^{<\mathrm{A}>} w_k^{<\mathrm{A}>}}{\alpha_k}, & \forall j \in \mathcal{I}_k^{<\mathrm{A}>}, \\ \dfrac{(1-\omega^*)\mathcal{M}_k^{<\mathrm{B}>} w_k^{<\mathrm{B}>}}{\alpha_k}, & \forall j \in \mathcal{I}_k^{<\mathrm{B}>}, \end{cases} \tag{4.3.13}$$

with $\alpha_k := \omega^* \mathcal{M}_k^{<\mathrm{A}>} + (1-\omega^*)\mathcal{M}_k^{<\mathrm{B}>}$ *and the weight* ω^* *to calculate* $\mathcal{P}\left(\boldsymbol{h}_k, \boldsymbol{h}_k^{<\mathrm{A}>}, \boldsymbol{h}_k^{<\mathrm{B}>}\right)$ *[see* (4.3.4) *and* (4.3.12)*].*

Proof: The proof is omitted because it is verified by simple algebra. □
Except for the case $\eta_k = -\sqrt{\xi_k\zeta_k} \neq 0$, we have $w_j^{(k)} \in (0, 1]$ and $\sum_{j\in\mathcal{I}_k} w_j^{(k)} = 1$. In this exceptional case, there is no problem for the same reason as for POWER-PSP I.

C. Optimality

Let us first present a lemma, followed by a proposition showing the optimality of the weights given in (4.3.4) in the sense of a certain max-min criterion.

Lemma 4.3.9. *The following conditions are equivalent:*

$$\begin{aligned} &\mathrm{C3}: \quad \Pi^-(\boldsymbol{s},\boldsymbol{a}) \cap \Pi^-(\boldsymbol{s},\boldsymbol{b}) \neq \emptyset \text{ and } \Pi(\boldsymbol{s},\boldsymbol{a}) \cap \Pi(\boldsymbol{s},\boldsymbol{b}) = \emptyset \\ &\mathrm{C4}: \quad \exists \delta \in (0,1) \cup (1,\infty) \text{ s.t. } \boldsymbol{s} - \boldsymbol{a} = \delta(\boldsymbol{s} - \boldsymbol{b}) \neq \boldsymbol{0}. \end{aligned}$$

Proof: See Sec. 4.4.2. □

Thanks to the geometric simplicity of the half-spaces, we can show the optimality of the weight ω^* and the relaxation parameter μ^* (For another investigation on optimal weight design, see, e.g., [3, Ch. 7]).

Proposition 4.3.10 (Optimality of ω^* and μ^*)**.** *Given* $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{b}) \in \mathcal{H}^3$ *s.t.* $\Pi^-(\boldsymbol{s}, \boldsymbol{a}) \cap \Pi^-(\boldsymbol{s}, \boldsymbol{b}) \neq \emptyset$, *let* $\phi(\omega, \mu, \boldsymbol{z}) := \|\boldsymbol{s} - \boldsymbol{z}\|^2 - \|\mathcal{Q}(\omega, \mu, \boldsymbol{s}, \boldsymbol{a}, \boldsymbol{b}) - \boldsymbol{z}\|^2$. *Then, the following hold.*

(a)

$$\begin{aligned} \widehat{\phi}(\omega, \mu) &:= \min_{\boldsymbol{z} \in \Pi^-(\boldsymbol{s}, \boldsymbol{a}) \cap \Pi^-(\boldsymbol{s}, \boldsymbol{b})} \phi(\omega, \mu, \boldsymbol{z}) \\ &= -\mu^2 \left\| \omega(\boldsymbol{a} - \boldsymbol{s}) + (1 - \omega)(\boldsymbol{b} - \boldsymbol{s}) \right\|^2 \\ &\quad + 2\mu \left[\omega \left\| \boldsymbol{a} - \boldsymbol{s} \right\|^2 + (1 - \omega) \left\| \boldsymbol{b} - \boldsymbol{s} \right\|^2 + \psi(\omega) \right], \end{aligned} \tag{4.3.14}$$

where, referring to Lemma 4.3.9*,*

$$\psi(\omega) := \begin{cases} 0, & \text{if } \Pi(\boldsymbol{s}, \boldsymbol{a}) \cap \Pi(\boldsymbol{s}, \boldsymbol{b}) \neq \emptyset, \\ \delta(1 - \delta)\omega \left\| \boldsymbol{b} - \boldsymbol{s} \right\|^2, & \text{if } \exists \delta \in (0, 1) \text{ s.t. } (\boldsymbol{s} - \boldsymbol{a}) = \delta(\boldsymbol{s} - \boldsymbol{b}), \\ \dfrac{1}{\delta}\left(1 - \dfrac{1}{\delta}\right)(1 - \omega) \left\| \boldsymbol{a} - \boldsymbol{s} \right\|^2, & \text{if } \exists \delta \in (1, \infty) \text{ s.t. } (\boldsymbol{s} - \boldsymbol{a}) = \delta(\boldsymbol{s} - \boldsymbol{b}). \end{cases}$$

(b) (ω^*, μ^*) *in* (4.3.3) *and* (4.3.4) *are optimal in the sense that*

$$(\omega^*, \mu^*) \in \operatorname*{argmax}_{(\omega, \mu) \in [0,1] \times [0, \infty)} \widehat{\phi}(\omega, \mu). \tag{4.3.15}$$

Moreover,

$$\widehat{\phi}(\omega^*, \mu) \geq 0, \ \forall \mu \in [0, 2\mu^*]. \tag{4.3.16}$$

Proof: See Sec. 4.4.3. □

Intuitively, (ω^*, μ^*) achieves a worst-case optimization, or, in other words, (4.3.15) implies that (ω^*, μ^*) is a solution to the max-min problem: firstly minimize $\phi(\omega, \mu, \boldsymbol{z})$ over $\boldsymbol{z}$ and then maximize the minimum over (ω, μ).

The following proposition guarantees the membership of the estimandum $\boldsymbol{h}^*$ to the closed half-spaces constructed by the proposed POWER technique, provided

that $\boldsymbol{h}^*$ belongs to the primitive half-spaces $\left(H_j^-(\boldsymbol{h}_k)\right)_{j\in\bar{\mathcal{I}}_k}$.

Proposition 4.3.11. *Suppose* $\boldsymbol{h}^* \in \bigcap_{j\in\bar{\mathcal{I}}_k} H_j^-(\boldsymbol{h}_k)$ $[\Leftarrow \boldsymbol{h}^* \in \bigcap_{j\in\bar{\mathcal{I}}_k} C_j(\rho)]$, $k \in \mathbb{N}$. *Then, the following hold.*

(a) *For* $(\boldsymbol{h}_k, \boldsymbol{h}_{k,\iota}^{(m)})$, $\forall \iota \in \mathcal{I}_k^{(m)}$, $\forall m \in \{0, 1, \cdots, M-1\}$ *given by POWER-PSP I, we have* $\boldsymbol{h}^* \in \Pi^-(\boldsymbol{h}_k, \boldsymbol{h}_{k,\iota}^{(m)})$.

(b) *For* $(\boldsymbol{h}_k, \boldsymbol{h}_k^{<\mathrm{A}>}, \boldsymbol{h}_k^{<\mathrm{B}>})$ *given by POWER-PSP II, we have* $\boldsymbol{h}^* \in \Pi^-(\boldsymbol{h}_k, \boldsymbol{h}_k^{<\mathrm{A}>}) \cap \Pi^-(\boldsymbol{h}_k, \boldsymbol{h}_k^{<\mathrm{B}>})$.

Proof: See Sec. 4.4.4. □

A remark on the proposed algorithms is given below.

Remark 4.3.12 (On optimality). *For the POWER-PSP I algorithm,* $\forall m = 1, 2, \cdots, M$, $\mathcal{P}\left(\boldsymbol{h}_k, \boldsymbol{h}_{k,\iota_1}^{(m-1)}, \boldsymbol{h}_{k,\iota_2}^{(m-1)}\right)$ *in* (4.3.9) *and* (4.3.10) *is the best among* $\mathcal{Q}\left(\omega, \mu, \boldsymbol{h}_k, \boldsymbol{h}_{k,\iota_1}^{(m-1)}, \boldsymbol{h}_{k,\iota_2}^{(m-1)}\right)$, $\forall(\omega, \mu) \in [0, 1] \times [0, \infty)$, *by Proposition* 4.3.10, *and hence it is especially better than* $\boldsymbol{h}_{k,\iota_1}^{(m-1)} = \mathcal{Q}\left(1, 1, \boldsymbol{h}_k, \boldsymbol{h}_{k,\iota_1}^{(m-1)}, \boldsymbol{h}_{k,\iota_2}^{(m-1)}\right)$ *and* $\boldsymbol{h}_{k,\iota_2}^{(m-1)} = \mathcal{Q}\left(0, 1, \boldsymbol{h}_k, \boldsymbol{h}_{k,\iota_1}^{(m-1)}, \boldsymbol{h}_{k,\iota_2}^{(m-1)}\right)$ *if* $\boldsymbol{h}^* \in \Pi^-(\boldsymbol{h}_k, \boldsymbol{h}_{k,\iota_1}^{(m-1)}) \cap \Pi^-(\boldsymbol{h}_k, \boldsymbol{h}_{k,\iota_2}^{(m-1)})$. *Fortunately,* $\boldsymbol{h}^* \in \Pi^-(\boldsymbol{h}_k, \boldsymbol{h}_{k,\iota_1}^{(m-1)}) \cap \Pi^-(\boldsymbol{h}_k, \boldsymbol{h}_{k,\iota_2}^{(m-1)})$ *is automatically guaranteed by Proposition* 4.3.11, *provided that* $\boldsymbol{h}^* \in \bigcap_{j\in\bar{\mathcal{I}}_k} H_j^-(\boldsymbol{h}_k)$. *This implies that, at each stage, the direction of update is improved thanks to Proposition* 4.3.3 *(POWER). Hence, POWER-PSP I realizes pairwise optimal weights at each stage. With a similar discussion, we can say that POWER-PSP II realizes pairwise optimal weights at the second stage.*

4.3.4 Matrix-Form Formulae of Proposed Algorithms

The projection-based formulation of the proposed algorithms is convenient to interpret geometrically (see Sec. 4.3.3.A) but would be unfamiliar to most readers. Hence, we show the matrix-form formulae of POWER-PSP I and II and their cost (the number of multiplications/divisions) in Tables 4.3.1 and 4.3.2, respectively. The presented cost assumes the use of q parallel processors. Note that the computation in the 0th stage of POWER-PSP I (or 1st stage of POWER-PSP II) can be computed in parallel.

Table 4.3.1: Efficient computation for the POWER-PSP I algorithm.
Require: the control sequences $\mathcal{I}_k^{(m)}$ $(m = 0, 1, \cdots, M)$, the inputs $\boldsymbol{U}_\iota$, the outputs $\boldsymbol{d}_\iota$ $(\iota \in \mathcal{I}_k^{(0)})$, the inflation parameter ρ, the step size λ_k, and any $\boldsymbol{h}_0 \in \mathcal{H}$.

Time update at time k	Cost
0th stage: $\forall \iota \in \mathcal{I}_k^{(0)}$	
$\boldsymbol{e}_\iota^{(k)} = \boldsymbol{U}_\iota^T \boldsymbol{h}_k - \boldsymbol{d}_\iota$	rN
$g_\iota^{(k)} = \|\boldsymbol{e}_\iota^{(k)}\|^2 - \rho$	r
$\boldsymbol{\alpha}_\iota^{(k)} = \begin{cases} \boldsymbol{U}_\iota \boldsymbol{e}_\iota^{(k)} & \text{if } g_\iota^{(k)} > 0 \\ \boldsymbol{0} & \text{otherwise} \end{cases}$	rN
$\beta_\iota^{(k)} = \begin{cases} -\dfrac{g_\iota^{(k)}}{2\|\boldsymbol{\alpha}_\iota^{(k)}\|^2} & \text{if } g_\iota^{(k)} > 0 \\ 0 & \text{otherwise} \end{cases}$	$N+1$
mth stage $(m = 1, 2, \cdots, M)$: $\forall \iota = (\iota_1, \iota_2) \in \mathcal{I}_k^{(m)}$	
$\eta_{k,\iota}^{(m)} = \begin{cases} \beta_{\iota_1}^{(k)} \beta_{\iota_2}^{(k)} {\boldsymbol{\alpha}_{\iota_1}^{(k)}}^T \boldsymbol{\alpha}_{\iota_2}^{(k)} & \text{for } m = 1 \\ {\boldsymbol{\gamma}_{k,\iota_1}^{(m-1)}}^T \boldsymbol{\gamma}_{k,\iota_2}^{(m-1)} & \text{for } m \geq 2 \end{cases}$	$N(+2)$
$\xi_{k,\iota}^{(m)} = \begin{cases} {\beta_{\iota_1}^{(k)}}^2 \|\boldsymbol{\alpha}_{\iota_1}^{(k)}\|^2 & \text{for } m = 1 \\ [\mu_{k,\iota_1}^{(m-1)} \omega_{k,\iota_1}^{(m-1)}]^2 \xi_{k,\iota_1}^{(m-1)} + \left[\mu_{k,\iota_1}^{(m-1)}(1 - \omega_{k,\iota_1}^{(m-1)})\right]^2 \zeta_{k,\iota_1}^{(m-1)} & \\ \quad +2[\mu_{k,\iota_1}^{(m-1)}]^2 \omega_{k,\iota_1}^{(m-1)}(1 - \omega_{k,\iota_1}^{(m-1)}) \eta_{k,\iota_1}^{(m-1)} & \text{for } m \geq 2 \end{cases}$	7 (2)
$\zeta_{k,\iota}^{(m)} = \begin{cases} {\beta_{\iota_2}^{(k)}}^2 \|\boldsymbol{\alpha}_{\iota_2}^{(k)}\|^2 & \text{for } m = 1 \\ [\mu_{k,\iota_2}^{(m-1)} \omega_{k,\iota_2}^{(m-1)}]^2 \xi_{k,\iota_2}^{(m-1)} + \left[\mu_{k,\iota_2}^{(m-1)}(1 - \omega_{k,\iota_2}^{(m-1)})\right]^2 \zeta_{k,\iota_2}^{(m-1)} & \\ \quad +2[\mu_{k,\iota_2}^{(m-1)}]^2 \omega_{k,\iota_2}^{(m-1)}(1 - \omega_{k,\iota_2}^{(m-1)}) \eta_{k,\iota_2}^{(m-1)} & \text{for } m \geq 2 \end{cases}$	7 (2)
if $\eta_{k,\iota}^{(m)} = -\sqrt{\xi_{k,\iota}^{(m)} \zeta_{k,\iota}^{(m)}} \neq 0$	
then $\mu_{k,\iota}^{(m)} = \omega_{k,\iota}^{(m)} = 0$, $\boldsymbol{\gamma}_{k,\iota}^{(m)} = \boldsymbol{0}$	
else	
$\mu_{k,\iota}^{(m)} := \begin{cases} 1 & \text{if } \eta_{k,\iota}^{(m)} \geq \xi_{k,\iota}^{(m)} \text{ or } \eta_{k,\iota}^{(m)} \geq \zeta_{k,\iota}^{(m)} \\ \dfrac{2\xi_{k,\iota}^{(m)} \zeta_{k,\iota}^{(m)} - [\xi_{k,\iota}^{(m)} + \zeta_{k,\iota}^{(m)}] \eta_{k,\iota}^{(m)}}{\xi_{k,\iota}^{(m)} \zeta_{k,\iota}^{(m)} - {\eta_{k,\iota}^{(m)}}^2} & \text{if } \eta_{k,\iota}^{(m)} < \min(\xi_{k,\iota}^{(m)}, \zeta_{k,\iota}^{(m)}) \end{cases}$	5
$\omega_{k,\iota}^{(m)} := \begin{cases} 1 & \text{if } \eta_{k,\iota}^{(m)} \geq \zeta_{k,\iota}^{(m)} \\ 0 & \text{if } \eta_{k,\iota}^{(m)} \geq \xi_{k,\iota}^{(m)} \\ \dfrac{\zeta_{k,\iota}^{(m)} [\xi_{k,\iota}^{(m)} - \eta_{k,\iota}^{(m)}]}{2\xi_{k,\iota}^{(m)} \zeta_{k,\iota}^{(m)} - [\xi_{k,\iota}^{(m)} + \zeta_{k,\iota}^{(m)}] \eta_{k,\iota}^{(m)}} & \text{if } \eta_{k,\iota}^{(m)} < \min(\xi_{k,\iota}^{(m)}, \zeta_{k,\iota}^{(m)}) \end{cases}$	1
$\boldsymbol{\gamma}_{k,\iota}^{(m)} = \begin{cases} \mu_{k,\iota}^{(m)} \omega_{k,\iota}^{(m)} \boldsymbol{\gamma}_{k,\iota_1}^{(m-1)} + \mu_{k,\iota}^{(m)}(1 - \omega_{k,\iota}^{(m)}) \boldsymbol{\gamma}_{k,\iota_2}^{(m-1)} & \text{for } m = 1, 2, \cdots, M-1 \\ \qquad \text{(where } \boldsymbol{\gamma}_{k,\iota_i}^{(0)} := \beta_{\iota_i}^{(k)} \boldsymbol{\alpha}_{\iota_i}^{(k)},\ i = 1, 2) & \\ \lambda_k \mu_{k,\iota}^{(m)} \omega_{k,\iota}^{(m)} \boldsymbol{\gamma}_{k,\iota_1}^{(m-1)} + \lambda_k \mu_{k,\iota}^{(m)}(1 - \omega_{k,\iota}^{(m)}) \boldsymbol{\gamma}_{k,\iota_2}^{(m-1)} & \text{for } m = M \end{cases}$	$2N+1$
endif;	
Update the filter: $\boldsymbol{h}_{k+1} = \boldsymbol{h}_k + \boldsymbol{\gamma}_{k,\iota}^{(M)}$	0
Total:	$(3M + 2r + 1)N$ + 21M + r

Table 4.3.2: Efficient computation for the POWER-PSP II algorithm. Require: the control sequences $\mathcal{I}_k^{<\mathrm{A}>}$, $\mathcal{I}_k^{<\mathrm{B}>}$, the inputs $\boldsymbol{U}_\iota$, the outputs $\boldsymbol{d}_\iota$ ($\iota \in \mathcal{I}_k^{<\mathrm{A}>} \cup \mathcal{I}_k^{<\mathrm{B}>}$), the inflation parameter ρ, the step size λ_k, the uniform weights $w_k^{<\mathrm{A}>}$, $w_k^{<\mathrm{B}>}$ and any $\boldsymbol{h}_0 \in \mathcal{H}$.

Time update at time k	Cost
1st stage:	
(a) compute $\boldsymbol{e}_\iota^{(k)}$, $g_\iota^{(k)}$, $\boldsymbol{\alpha}_\iota^{(k)}$, and $\beta_\iota^{(k)}$, $\forall \iota \in \mathcal{I}_k^{<\mathrm{A}>} \cup \mathcal{I}_k^{<\mathrm{B}>}$	
in the same way as in Table 4.3.1, and $\forall \iota \in \mathcal{I}_k^{<\mathrm{g}>}$ ($\mathrm{g} \in \{\mathrm{A},\mathrm{B}\}$)	$(2r+1)N + r + 1$
$\boldsymbol{\tau}_\iota^{(k)} = w_k^{<\mathrm{g}>} \beta_\iota^{(k)} \boldsymbol{\alpha}_\iota^{(k)}$	$N+1$
$\upsilon_\iota^{(k)} = w_k^{<\mathrm{g}>} g_\iota^{(k)} \beta_\iota^{(k)}$	2
(b) compute $\forall \mathrm{g} \in \{\mathrm{A},\mathrm{B}\}$	
$\boldsymbol{\nu}_k^{<\mathrm{g}>} = \sum_{\iota \in \mathcal{I}_k^{<\mathrm{g}>}} \boldsymbol{\tau}_\iota^{(k)}$	0
$\pi_k^{<\mathrm{g}>} = \sum_{\iota \in \mathcal{I}_k^{<\mathrm{g}>}} \upsilon_\iota^{(k)}$	0
$\mathcal{M}_k^{<\mathrm{g}>} = \begin{cases} \dfrac{\pi_k^{<\mathrm{g}>}}{2\|\boldsymbol{\nu}_k^{<\mathrm{g}>}\|^2} & \text{if } \boldsymbol{\nu}_k^{<\mathrm{g}>} \neq \boldsymbol{0} \\ 1 & \text{otherwise} \end{cases}$	$N+1$
2nd stage:	
compute	
$\eta_k = \mathcal{M}_k^{<\mathrm{A}>} \mathcal{M}_k^{<\mathrm{B}>} \boldsymbol{\nu}_k^{<\mathrm{A}>T} \boldsymbol{\nu}_k^{<\mathrm{B}>}$	$N+2$
$\xi_k = \mathcal{M}_k^{<\mathrm{A}>2} \|\boldsymbol{\nu}_k^{<\mathrm{A}>}\|^2$	2
$\zeta_k = \mathcal{M}_k^{<\mathrm{B}>2} \|\boldsymbol{\nu}_k^{<\mathrm{B}>}\|^2$	2
and μ_k, ω_k as in $\mu_{k,\iota}^{(m)}$, $\omega_{k,\iota}^{(m)}$ in Table 4.3.1	6
Update the filter:	
$\boldsymbol{h}_{k+1} = \boldsymbol{h}_k + \lambda_k \mu_k \omega_k \mathcal{M}_k^{<\mathrm{A}>} \boldsymbol{\nu}_k^{<\mathrm{A}>} + \lambda_k \mu_k (1-\omega_k) \mathcal{M}_k^{<\mathrm{B}>} \boldsymbol{\nu}_k^{<\mathrm{B}>}$	$2N+4$
Total:	$(2r+6)N + r$

4.4 Proofs

4.4.1 Proof of Proposition 4.3.3

Let us start with the following simple lemma.

Lemma 4.4.1. *Let Π_i^- $(\subset \mathcal{H})$, $i = 1, 2$, be a closed half-space s.t. $\Pi_1^- \cap \Pi_2^- \neq \emptyset$. Then, the following hold.*

(a) *For any $\boldsymbol{x} \in \mathcal{H}$,*

$$P_{\Pi_1^- \cap \Pi_2^-}(\boldsymbol{x}) = \begin{cases} P_{\Pi_1^-}(\boldsymbol{x}), & \text{if } P_{\Pi_1^-}(\boldsymbol{x}) \in \Pi_2^-, \\ P_{\Pi_2^-}(\boldsymbol{x}), & \text{if } P_{\Pi_2^-}(\boldsymbol{x}) \in \Pi_1^-, \\ P_{\Pi_1 \cap \Pi_2}(\boldsymbol{x}), & \text{if } P_{\Pi_2^-}(\boldsymbol{x}) \notin \Pi_1^- \text{ and } P_{\Pi_1^-}(\boldsymbol{x}) \notin \Pi_2^-, \end{cases}$$

where Π_i $(i = 1, 2)$ denotes the boundary hyperplane of Π_i^-.

(b) *If $\exists \boldsymbol{x} \in \mathcal{H}$ s.t. $P_{\Pi_1^-}(\boldsymbol{x}) \notin \Pi_2^-$ and $P_{\Pi_2^-}(\boldsymbol{x}) \notin \Pi_1^-$, then, $\Pi_1 \cap \Pi_2 \neq \emptyset$.*

Proof of Lemma 4.4.1*:* The proof is omitted because it is verified by simple algebra. □

The following lemma gives a closed-form formula of $P_{\Pi_1 \cap \Pi_2}(\boldsymbol{x})$ in terms of $\boldsymbol{x}$, $P_{\Pi_1}(\boldsymbol{x})$, and $P_{\Pi_2}(\boldsymbol{x})$.

Lemma 4.4.2. *Let $(\boldsymbol{h}, \boldsymbol{h}_1, \boldsymbol{h}_2) \in \mathcal{H}^3 (= \mathcal{H} \times \mathcal{H} \times \mathcal{H})$ satisfy $\boldsymbol{h} \notin \{\boldsymbol{h}_1, \boldsymbol{h}_2\}$, $\boldsymbol{h}_1 \neq \boldsymbol{h}_2$, and $\Pi_1 \cap \Pi_2 \neq \emptyset$, where $\Pi_i := \Pi(\boldsymbol{h}, \boldsymbol{h}_i)$ $(i = 1, 2)$. Then,*

$$P_{\Pi_1 \cap \Pi_2}(\boldsymbol{h}) = \boldsymbol{h} + \mu_{\text{opt}}[w_{\text{opt}} \boldsymbol{h}_1 + (1 - w_{\text{opt}}) \boldsymbol{h}_2 - \boldsymbol{h}], \tag{4.4.1}$$

where, with $\boldsymbol{x}_i := \boldsymbol{h}_i - \boldsymbol{h}$ $(i = 1, 2)$,

$$\mu_{\text{opt}} := \frac{2 \|\boldsymbol{x}_1\|^2 \|\boldsymbol{x}_2\|^2 - \left(\|\boldsymbol{x}_1\|^2 + \|\boldsymbol{x}_2\|^2 \right) \langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle}{\|\boldsymbol{x}_1\|^2 \|\boldsymbol{x}_2\|^2 - \langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle^2}, \tag{4.4.2}$$

$$w_{\text{opt}} := \frac{\|\boldsymbol{x}_2\|^2 \left(\|\boldsymbol{x}_1\|^2 - \langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle \right)}{2 \|\boldsymbol{x}_1\|^2 \|\boldsymbol{x}_2\|^2 - \left(\|\boldsymbol{x}_1\|^2 + \|\boldsymbol{x}_2\|^2 \right) \langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle} \tag{4.4.3}$$

$$= \frac{\cos\theta_1 \sin\theta_1}{\cos\theta_1 \sin\theta_1 + \cos\theta_2 \sin\theta_2}, \tag{4.4.4}$$

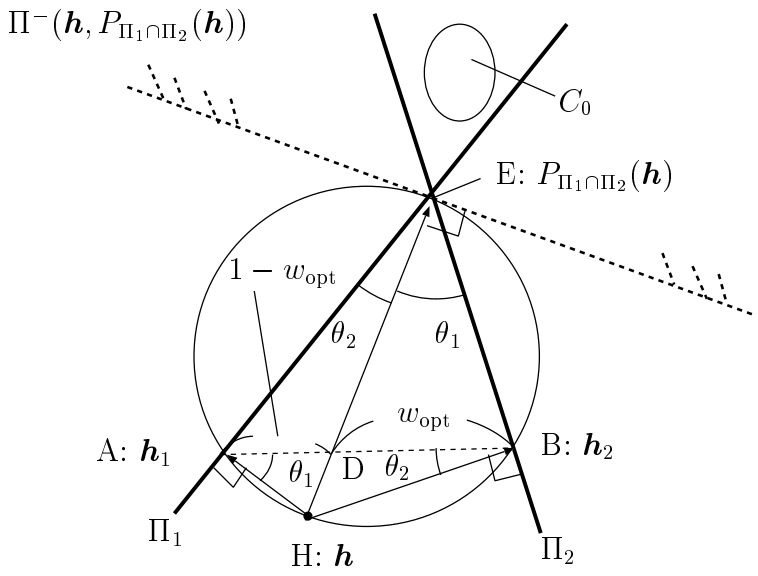


Figure 4.4.4: Geometric interpretation of Lemma 4.4.2 in the case of $\cos\theta_1$, $\cos\theta_2 > 0$. C_0 $[:= \bigcap_{j\in\mathcal{I}_k} C_j(\rho)]$ is automatically contained by $\Pi^-(\boldsymbol{h}, \boldsymbol{h}_1) \cap \Pi^-(\boldsymbol{h}, \boldsymbol{h}_2)$ and $\Pi^-[\boldsymbol{h}, P_{\Pi_1\cap\Pi_2}(\boldsymbol{h})]$.

$$\begin{aligned} \cos\theta_1 &:= \frac{\langle \boldsymbol{x}_1, \boldsymbol{x}_1 - \boldsymbol{x}_2\rangle}{\|\boldsymbol{x}_1\|\,\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|},\ \cos\theta_2 := \frac{\langle \boldsymbol{x}_2, \boldsymbol{x}_2 - \boldsymbol{x}_1\rangle}{\|\boldsymbol{x}_2\|\,\|\boldsymbol{x}_2 - \boldsymbol{x}_1\|}, \\ \sin\theta_i &:= \sqrt{1-\cos^2\theta_i}, \quad i = 1, 2. \end{aligned} \tag{4.4.5}$$

Proof of Lemma 4.4.2*:* By $\Pi_1 \cap \Pi_2 \neq \emptyset$, $\boldsymbol{h} \notin \{\boldsymbol{h}_1, \boldsymbol{h}_2\}$ and $\boldsymbol{h}_1 \neq \boldsymbol{h}_2$, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are linearly independent, and hence $\boldsymbol{G}^T\boldsymbol{G}$ is nonsingular, where $\boldsymbol{G} := [\boldsymbol{x}_1\ \boldsymbol{x}_2]$. Moreover, (cf. e.g., [73, p. 65 Theorem 2])

$$P_{\Pi_1\cap\Pi_2}(\boldsymbol{h}) = \boldsymbol{h} + \boldsymbol{G}(\boldsymbol{G}^T\boldsymbol{G})^{-1}(\boldsymbol{v} - \boldsymbol{G}^T\boldsymbol{h}), \tag{4.4.6}$$

where $\boldsymbol{v} := \begin{bmatrix} \langle \boldsymbol{x}_1, \boldsymbol{h}_1\rangle \\ \langle \boldsymbol{x}_2, \boldsymbol{h}_2\rangle \end{bmatrix}$. By simple manipulation, (4.4.1)–(4.4.3) are deduced from (4.4.6). The expression in (4.4.4) is obvious by (4.4.5). □

Referring to Fig. 4.4.4, the following remark gives a geometric interpretation of Lemma 4.4.2.

Remark 4.4.3 (Geometric interpretation of Lemma 4.4.2)**.** *Suppose* $\cos\theta_i > 0$, $i = 1, 2$. *By* $\widehat{HAE} = \widehat{HBE} = \pi/2$ *(rad) and the fact of elementary geometry on inscribed angle, we have* $\widehat{HEB} = \widehat{HAB} = \theta_1$ *and* $\widehat{HEA} = \widehat{HBA} = \theta_2$ *(see Fig.* 4.4.4*), where* $\widehat{ABC}$ *denotes the inscribed angle between* AB *and* CB *for a given triangle*

ABC. Then, noting the areas of the triangles HEB and HEA in Fig. 4.4.4, *we obtain*

$$w_{\text{opt}} : 1 - w_{\text{opt}} = HB\cos\theta_1 : HA\cos\theta_2 = \cos\theta_1 \sin\theta_1 : \cos\theta_2 \sin\theta_2,$$

which provides a geometric verification of (4.4.4). *A similar interpretation is possible for* $\cos\theta_1 \leq 0$ *or* $\cos\theta_2 \leq 0$.

Proof of Proposition 4.3.3*:*
Proof of (a): By (4.3.1) and the equality condition in the Cauchy-Schwarz inequality, it is sufficient to show

$$\begin{aligned} & \text{C1:} \quad \langle \boldsymbol{s}-\boldsymbol{a}, \boldsymbol{y}-\boldsymbol{a}\rangle > 0 \text{ or } \langle \boldsymbol{s}-\boldsymbol{b}, \boldsymbol{y}-\boldsymbol{b}\rangle > 0, \ \forall \boldsymbol{y} \in \mathcal{H} \\ \Leftrightarrow \ & \text{C2:} \quad \exists \delta < 0 \text{ s.t. } \boldsymbol{s}-\boldsymbol{a} = \delta(\boldsymbol{s}-\boldsymbol{b}) \neq \boldsymbol{0}. \end{aligned} \tag{4.4.7}$$

(I) **Proof of C2 $\Rightarrow$ C1:** Assume C2 and $\langle \boldsymbol{s}-\boldsymbol{b}, \boldsymbol{y}-\boldsymbol{b}\rangle \leq 0$. Then, we obtain

$$\begin{aligned} & \langle \boldsymbol{s}-\boldsymbol{a}, \boldsymbol{y}-\boldsymbol{a}\rangle = \delta\left(\langle \boldsymbol{s}-\boldsymbol{b}, \boldsymbol{y}-\boldsymbol{b}\rangle - \langle \boldsymbol{s}-\boldsymbol{b}, \boldsymbol{a}-\boldsymbol{b}\rangle\right) \geq -\delta \langle \boldsymbol{s}-\boldsymbol{b}, \boldsymbol{a}-\boldsymbol{b}\rangle \\ & = -\delta \langle \boldsymbol{s}-\boldsymbol{b}, \boldsymbol{s}-\boldsymbol{b}-(\boldsymbol{s}-\boldsymbol{a})\rangle = -\delta(1-\delta)\|\boldsymbol{s}-\boldsymbol{b}\|^2 > 0. \end{aligned}$$

(II) **Proof of C1 $\Rightarrow$ C2:** By the contraposition, we will show

$$\begin{aligned} & [\exists \delta \geq 0 \text{ s.t. } \boldsymbol{s}-\boldsymbol{a} = \delta(\boldsymbol{s}-\boldsymbol{b})] \ \text{ or } \ [\forall \delta \in \mathbb{R}, \ \boldsymbol{s}-\boldsymbol{a} \neq \delta(\boldsymbol{s}-\boldsymbol{b})] \\ & \Rightarrow \exists \boldsymbol{y} \in \mathcal{H} \text{ s.t. } \langle \boldsymbol{s}-\boldsymbol{a}, \boldsymbol{y}-\boldsymbol{a}\rangle \leq 0 \text{ and } \langle \boldsymbol{s}-\boldsymbol{b}, \boldsymbol{y}-\boldsymbol{b}\rangle \leq 0. \end{aligned} \tag{4.4.8}$$

(i) Assume $\exists \delta \geq 0$ s.t. $\boldsymbol{s}-\boldsymbol{a} = \delta(\boldsymbol{s}-\boldsymbol{b})$. Then, we have

$$\langle \boldsymbol{s}-\boldsymbol{a}, \boldsymbol{y}-\boldsymbol{a}\rangle = \delta \langle \boldsymbol{s}-\boldsymbol{b}, \boldsymbol{y}-\boldsymbol{s}+\boldsymbol{s}-\boldsymbol{a}\rangle = \delta\left(\langle \boldsymbol{s}-\boldsymbol{b}, \boldsymbol{y}-\boldsymbol{s}\rangle + \delta\|\boldsymbol{s}-\boldsymbol{b}\|^2\right).$$

Letting specially $\boldsymbol{y} = \boldsymbol{s} - (1+\delta)(\boldsymbol{s}-\boldsymbol{b}) \in \mathcal{H}$ yields

$$\langle \boldsymbol{s}-\boldsymbol{a}, \boldsymbol{y}-\boldsymbol{a}\rangle = \langle \boldsymbol{s}-\boldsymbol{b}, \boldsymbol{y}-\boldsymbol{b}\rangle = -\delta\|\boldsymbol{s}-\boldsymbol{b}\|^2 \leq 0.$$

(ii) Assume $\boldsymbol{s}-\boldsymbol{a} \neq \delta(\boldsymbol{s}-\boldsymbol{b}),\ \forall \delta \in \mathbb{R}$. In this case, we have

$$\dim\left\{\mathcal{R}\left(\left[\begin{array}{c}(\boldsymbol{s}-\boldsymbol{a})^T\\(\boldsymbol{s}-\boldsymbol{b})^T\end{array}\right]\right)\right\}=2,\ \text{hence}\ \mathcal{R}\left(\left[\begin{array}{c}(\boldsymbol{s}-\boldsymbol{a})^T\\(\boldsymbol{s}-\boldsymbol{b})^T\end{array}\right]\right)\ni\left[\begin{array}{c}(\boldsymbol{s}-\boldsymbol{a})^T\boldsymbol{a}\\(\boldsymbol{s}-\boldsymbol{b})^T\boldsymbol{b}\end{array}\right],$$

where $\mathcal{R}(\cdot)$ denotes the *range space* of the columns of a matrix. This implies

$$\exists \boldsymbol{y}\in\mathcal{H}\ \text{s.t.}\ \left[\begin{array}{c}(\boldsymbol{s}-\boldsymbol{a})^T\\(\boldsymbol{s}-\boldsymbol{b})^T\end{array}\right]\boldsymbol{y}=\left[\begin{array}{c}(\boldsymbol{s}-\boldsymbol{a})^T\boldsymbol{a}\\(\boldsymbol{s}-\boldsymbol{b})^T\boldsymbol{b}\end{array}\right],$$

from which it follows that $\exists \boldsymbol{y} \in \mathcal{H}$ s.t. $\langle \boldsymbol{s}-\boldsymbol{a}, \boldsymbol{y}-\boldsymbol{a}\rangle = \langle \boldsymbol{s}-\boldsymbol{b}, \boldsymbol{y}-\boldsymbol{b}\rangle = 0$ [This implies $\Pi(\boldsymbol{s},\boldsymbol{a})\cap\Pi(\boldsymbol{s},\boldsymbol{b})\neq\emptyset$ by (4.3.1)].

The above discussion verifies (4.4.7).

Proof of (b):

(I) Assume $\eta \geq \zeta$. In this case, $\langle \boldsymbol{s}-\boldsymbol{b}, \boldsymbol{a}-\boldsymbol{b}\rangle \leq 0$, which implies $\boldsymbol{a}\in\Pi^-(\boldsymbol{s},\boldsymbol{b})$ by (4.3.1). Hence, we have $\mathcal{P}(\boldsymbol{s},\boldsymbol{a},\boldsymbol{b}) = P_{\Pi^-(\boldsymbol{s},\boldsymbol{a})}(\boldsymbol{s}) = \boldsymbol{a}$, where the first equality follows from Lemma 4.4.1-(a) and the second from Remark 4.3.1.

(II) Assume $(\zeta >)\eta \geq \xi$. This case is proved in analogy with (I).

(III) Assume $\eta < \min(\xi,\zeta)$. In this case, we have $\langle \boldsymbol{s}-\boldsymbol{b}, \boldsymbol{a}-\boldsymbol{b}\rangle > 0$ and $\langle \boldsymbol{s}-\boldsymbol{a}, \boldsymbol{b}-\boldsymbol{a}\rangle > 0$, which implies $P_{\Pi^-(\boldsymbol{s},\boldsymbol{a})}(\boldsymbol{s}) = \boldsymbol{a} \notin \Pi^-(\boldsymbol{s},\boldsymbol{b})$ and $P_{\Pi^-(\boldsymbol{s},\boldsymbol{b})}(\boldsymbol{s}) = \boldsymbol{b} \notin \Pi^-(\boldsymbol{s},\boldsymbol{a})$, respectively. Thus, by Lemma 4.4.1-(a), $P_{\Pi^-(\boldsymbol{s},\boldsymbol{a})\cap\Pi^-(\boldsymbol{s},\boldsymbol{b})}(\boldsymbol{s}) = P_{\Pi(\boldsymbol{s},\boldsymbol{a})\cap\Pi(\boldsymbol{s},\boldsymbol{b})}(\boldsymbol{s})$, which allows us to use Lemma 4.4.2. The equations (4.3.2)–(4.3.4) for this case are readily verified by substituting $\langle \boldsymbol{x}_1,\boldsymbol{x}_2\rangle = \eta$, $\|\boldsymbol{x}_1\|^2 = \xi$, and $\|\boldsymbol{x}_2\|^2 = \zeta$ into (4.4.1)–(4.4.3) in Lemma 4.4.2. All that we have to do is to check that the conditions of Lemma 4.4.2 are satisfied, i.e.,

$$\boldsymbol{s}\notin\{\boldsymbol{a},\boldsymbol{b}\}\ \text{ and }\ \boldsymbol{a}\neq\boldsymbol{b} \tag{4.4.9}$$

$$\Pi(\boldsymbol{s},\boldsymbol{a})\cap\Pi(\boldsymbol{s},\boldsymbol{b})\neq\emptyset. \tag{4.4.10}$$

By Lemma 4.4.1-(b), it is not hard to verify (4.4.9) and (4.4.10), which completes the proof. □

4.4.2 Proof of Lemma 4.3.9

(I) **Proof of C4 $\Rightarrow$ C3:** Assume C4 holds. In this case, we have $\eta = \delta \|\boldsymbol{s}-\boldsymbol{b}\|^2 > 0$, which implies that $\eta \neq -\sqrt{\xi\zeta}$, leading to $\Pi^-(\boldsymbol{s},\boldsymbol{a}) \cap \Pi^-(\boldsymbol{s},\boldsymbol{b}) \neq \emptyset$ by Proposition 4.3.3-(a). Moreover,

$$\begin{aligned}\Pi(\boldsymbol{s},\boldsymbol{a}) &= \{\boldsymbol{x}\in\mathcal{H} : \langle \boldsymbol{s}-\boldsymbol{b}, \boldsymbol{x}-\boldsymbol{s}+\delta(\boldsymbol{s}-\boldsymbol{b})\rangle = 0\} \\ &= \left\{\boldsymbol{x}\in\mathcal{H} : \langle \boldsymbol{s}-\boldsymbol{b}, \boldsymbol{x}-\boldsymbol{b}\rangle = (1-\delta)\|\boldsymbol{s}-\boldsymbol{b}\|^2\right\}, \\ \Pi(\boldsymbol{s},\boldsymbol{b}) &= \{\boldsymbol{x}\in\mathcal{H} : \langle \boldsymbol{s}-\boldsymbol{b}, \boldsymbol{x}-\boldsymbol{b}\rangle = 0\}, \end{aligned} \tag{4.4.11}$$

from which it follows, from $(1-\delta)\|\boldsymbol{s}-\boldsymbol{b}\|^2 \neq 0$, that $\Pi(\boldsymbol{s},\boldsymbol{a}) \cap \Pi(\boldsymbol{s},\boldsymbol{b}) = \emptyset$.

(II) **Proof of C3 $\Rightarrow$ C4:** Assume C3. In this case, by (II)-(ii) in the proof of (a) in Sec. 4.4.1, $\exists\delta \in \mathbb{R}$ s.t. $\boldsymbol{s}-\boldsymbol{a} = \delta(\boldsymbol{s}-\boldsymbol{b})$. It is obvious that $\delta \notin \{0,1\}$. Hence, it is sufficient to verify $\delta \notin (-\infty, 0)$. Suppose $\delta \in (-\infty, 0)$. Then, we obtain

$$\begin{aligned}\Pi^-(\boldsymbol{s},\boldsymbol{a}) &= \{\boldsymbol{x}\in\mathcal{H} : \langle \boldsymbol{x}-\boldsymbol{a}, \delta(\boldsymbol{s}-\boldsymbol{b})\rangle \leq 0\} \\ &= \{\boldsymbol{x}\in\mathcal{H} : \langle \boldsymbol{x}-\boldsymbol{b}+\boldsymbol{b}-\boldsymbol{a}, \boldsymbol{s}-\boldsymbol{b}\rangle \geq 0\} \\ &= \{\boldsymbol{x}\in\mathcal{H} : \langle \boldsymbol{x}-\boldsymbol{b}, \boldsymbol{s}-\boldsymbol{b}\rangle \geq \langle \boldsymbol{a}-\boldsymbol{b}, \boldsymbol{s}-\boldsymbol{b}\rangle\} \\ &= \{\boldsymbol{x}\in\mathcal{H} : \langle \boldsymbol{x}-\boldsymbol{b}, \boldsymbol{s}-\boldsymbol{b}\rangle \geq (1-\delta)\|\boldsymbol{s}-\boldsymbol{b}\|^2\},\end{aligned}$$

where $(1-\delta)\|\boldsymbol{s}-\boldsymbol{b}\|^2 > 0$. This implies $\Pi^-(\boldsymbol{s},\boldsymbol{a}) \cap \Pi^-(\boldsymbol{s},\boldsymbol{b}) = \emptyset$, which violates C3. Hence, we can verify $\delta \in (0,1) \cup (1,\infty)$. □

4.4.3 Proof of Proposition 4.3.10

Proof of (a): For any $\boldsymbol{z} \in \Pi^-(\boldsymbol{s},\boldsymbol{a}) \cap \Pi^-(\boldsymbol{s},\boldsymbol{b})$,

$$\begin{aligned}\phi(\omega,\mu,\boldsymbol{z}) &= \|\boldsymbol{s}-\boldsymbol{z}\|^2 - \|\boldsymbol{s}-\boldsymbol{z}+\mu[\omega\boldsymbol{a}+(1-\omega)\boldsymbol{b}-\boldsymbol{s}]\|^2 \\ &= -\mu^2\|\omega(\boldsymbol{a}-\boldsymbol{s})+(1-\omega)(\boldsymbol{b}-\boldsymbol{s})\|^2 \\ &\quad +2\mu\langle \boldsymbol{s}-\boldsymbol{z}, \omega(\boldsymbol{s}-\boldsymbol{a})+(1-\omega)(\boldsymbol{s}-\boldsymbol{b})\rangle. \end{aligned} \tag{4.4.12}$$

Here,

$$\begin{aligned}
&\langle \boldsymbol{s}-\boldsymbol{z}, \omega(\boldsymbol{s}-\boldsymbol{a})+(1-\omega)(\boldsymbol{s}-\boldsymbol{b})\rangle \\
&= \omega \langle \boldsymbol{s}-\boldsymbol{a}+\boldsymbol{a}-\boldsymbol{z}, \boldsymbol{s}-\boldsymbol{a}\rangle + (1-\omega)\langle \boldsymbol{s}-\boldsymbol{b}+\boldsymbol{b}-\boldsymbol{z}, \boldsymbol{s}-\boldsymbol{b}\rangle \\
&= \omega \|\boldsymbol{s}-\boldsymbol{a}\|^2 + (1-\omega)\|\boldsymbol{s}-\boldsymbol{b}\|^2 - \{\omega\langle \boldsymbol{z}-\boldsymbol{a}, \boldsymbol{s}-\boldsymbol{a}\rangle + (1-\omega)\langle \boldsymbol{z}-\boldsymbol{b}, \boldsymbol{s}-\boldsymbol{b}\rangle\}.
\end{aligned} \tag{4.4.13}$$

By (4.4.12) and (4.4.13), all that we have to prove is

$$\min_{\boldsymbol{z}\in \Pi^-(\boldsymbol{s},\boldsymbol{a})\cap\Pi^-(\boldsymbol{s},\boldsymbol{b})} \widehat{\psi}(\omega, \boldsymbol{z}) = \psi(\omega), \tag{4.4.14}$$

where $\widehat{\psi}(\omega, \boldsymbol{z}) := -\omega\langle \boldsymbol{z}-\boldsymbol{a}, \boldsymbol{s}-\boldsymbol{a}\rangle - (1-\omega)\langle \boldsymbol{z}-\boldsymbol{b}, \boldsymbol{s}-\boldsymbol{b}\rangle$. By $\boldsymbol{a} = P_{\Pi^-(\boldsymbol{s},\boldsymbol{a})}(\boldsymbol{s})$ and $\boldsymbol{b} = P_{\Pi^-(\boldsymbol{s},\boldsymbol{b})}(\boldsymbol{s})$ (see Remark 4.3.1),

$$\widehat{\psi}(\omega, \boldsymbol{z}) \geq 0, \forall \boldsymbol{z} \in \Pi^-(\boldsymbol{s},\boldsymbol{a})\cap\Pi^-(\boldsymbol{s},\boldsymbol{b}). \tag{4.4.15}$$

(I) Assume $\Pi(\boldsymbol{s},\boldsymbol{a})\cap\Pi(\boldsymbol{s},\boldsymbol{b}) \neq \emptyset$. In this case,

$$\widehat{\psi}(\omega, \boldsymbol{z}) = 0, \forall \boldsymbol{z} \in \Pi(\boldsymbol{s},\boldsymbol{a})\cap\Pi(\boldsymbol{s},\boldsymbol{b}). \tag{4.4.16}$$

By (4.4.15) and (4.4.16),

$$\min_{\boldsymbol{z}\in \Pi^-(\boldsymbol{s},\boldsymbol{a})\cap\Pi^-(\boldsymbol{s},\boldsymbol{b})} \widehat{\psi}(\omega, \boldsymbol{z}) = 0.$$

(II) Assume $\Pi(\boldsymbol{s},\boldsymbol{a})\cap\Pi(\boldsymbol{s},\boldsymbol{b}) = \emptyset$ $\left[\Leftrightarrow \exists \delta \in (0,1)\cup(1,\infty) \text{ s.t. } \boldsymbol{s}-\boldsymbol{a} = \delta(\boldsymbol{s}-\boldsymbol{b}) \neq \boldsymbol{0}\right.$ by Lemma 4.3.9$\left.\right]$. Referring to (4.4.11),

$$\Pi^-(\boldsymbol{s},\boldsymbol{a}) = \left\{\boldsymbol{x}\in\mathcal{H} : \langle \boldsymbol{s}-\boldsymbol{b}, \boldsymbol{x}-\boldsymbol{b}\rangle \leq (1-\delta)\|\boldsymbol{s}-\boldsymbol{b}\|^2\right\}. \tag{4.4.17}$$

(i) Assume $\delta \in (0,1)$. In this case, $(1-\delta)\|\boldsymbol{s}-\boldsymbol{b}\|^2 > 0$, which yields $\Pi^-(\boldsymbol{s},\boldsymbol{b}) \subset \Pi^-(\boldsymbol{s},\boldsymbol{a})$, and thus

$$\Pi^-(\boldsymbol{s},\boldsymbol{a})\cap\Pi^-(\boldsymbol{s},\boldsymbol{b}) = \Pi^-(\boldsymbol{s},\boldsymbol{b}). \tag{4.4.18}$$

On the other hand,

$$\begin{aligned}\widehat{\psi}(\omega, \boldsymbol{z}) &= -\omega \langle \boldsymbol{z} - \boldsymbol{b} + \boldsymbol{b} - \boldsymbol{a}, \delta(\boldsymbol{s} - \boldsymbol{b}) \rangle - (1-\omega) \langle \boldsymbol{z} - \boldsymbol{b}, \boldsymbol{s} - \boldsymbol{b} \rangle \\ &= (\omega - 1 - \delta\omega) \langle \boldsymbol{z} - \boldsymbol{b}, \boldsymbol{s} - \boldsymbol{b} \rangle + \delta\omega \langle \boldsymbol{a} - \boldsymbol{s} + \boldsymbol{s} - \boldsymbol{b}, \boldsymbol{s} - \boldsymbol{b} \rangle \\ &= (\omega - 1 - \delta\omega) \langle \boldsymbol{z} - \boldsymbol{b}, \boldsymbol{s} - \boldsymbol{b} \rangle + \delta(1-\delta)\omega \|\boldsymbol{s} - \boldsymbol{b}\|^2 .\end{aligned}$$

By $\omega \in [0,1]$ and $\delta > 0$, we have $\omega - 1 - \delta\omega < 0$, which yields, with (4.4.18),

$$\widehat{\psi}(\omega, \boldsymbol{z}) \geq \delta(1-\delta)\omega \|\boldsymbol{s} - \boldsymbol{b}\|^2 , \forall \boldsymbol{z} \in \Pi^-(\boldsymbol{s}, \boldsymbol{a}) \cap \Pi^-(\boldsymbol{s}, \boldsymbol{b}) \, (= \Pi^-(\boldsymbol{s}, \boldsymbol{b})),$$

where the equality holds if $\boldsymbol{z} \in \Pi(\boldsymbol{s}, \boldsymbol{b}) \subset \Pi^-(\boldsymbol{s}, \boldsymbol{a}) \cap \Pi^-(\boldsymbol{s}, \boldsymbol{b})$. Hence, we can verify

$$\min_{\boldsymbol{z} \in \Pi^-(\boldsymbol{s}, \boldsymbol{a}) \cap \Pi^-(\boldsymbol{s}, \boldsymbol{b})} \widehat{\psi}(\omega, \boldsymbol{z}) = \delta(1-\delta)\omega \|\boldsymbol{b} - \boldsymbol{s}\|^2 .$$

(ii) Assume $\delta \in (1, \infty)$. In a similar way to (i), we can verify

$$\min_{\boldsymbol{z} \in \Pi^-(\boldsymbol{s}, \boldsymbol{a}) \cap \Pi^-(\boldsymbol{s}, \boldsymbol{b})} \widehat{\psi}(\omega, \boldsymbol{z}) = \frac{1}{\delta}\left(1 - \frac{1}{\delta}\right)(1-\omega) \|\boldsymbol{a} - \boldsymbol{s}\|^2 .$$

Proof of (b):

(I) Assume $\Pi(\boldsymbol{s}, \boldsymbol{a}) \cap \Pi(\boldsymbol{s}, \boldsymbol{b}) = \emptyset$. In this case, by Lemma 4.3.9, $\exists \delta \in (0,1) \cup (1, \infty)$ s.t. $\boldsymbol{s} - \boldsymbol{a} = \delta(\boldsymbol{s} - \boldsymbol{b}) \neq \boldsymbol{0}$.

(i) Assume $\delta \in (0,1)$. In this case, $\eta \geq \xi$, hence $(\omega^*, \mu^*) = (0,1)$ by Proposition 4.3.3. From (I), we obtain

$$\begin{aligned}\widehat{\phi}(\omega, \mu) &= -\mu \left\{ \mu(\omega\delta + 1 - \omega)^2 \|\boldsymbol{b} - \boldsymbol{s}\|^2 - 2[\omega\delta^2 + 1 - \omega + \delta(1-\delta)\omega] \|\boldsymbol{b} - \boldsymbol{s}\|^2 \right\} \\ &= -(\omega\delta + 1 - \omega)^2 \|\boldsymbol{b} - \boldsymbol{s}\|^2 \mu \left(\mu - \frac{2}{\omega\delta + 1 - \omega} \right) \\ &= -(\omega\delta + 1 - \omega)^2 \|\boldsymbol{b} - \boldsymbol{s}\|^2 \left(\mu - \frac{1}{\omega\delta + 1 - \omega} \right)^2 + \|\boldsymbol{b} - \boldsymbol{s}\|^2 .\end{aligned}$$

By $(\omega\delta + 1 - \omega)^2 \|\boldsymbol{b} - \boldsymbol{s}\|^2 > 0$ and $\omega\delta + 1 - \omega > 0$, $\phi(\omega, \mu)$ is maximized by $\mu = 1/(\omega\delta + 1 - \omega), \forall \omega \in [0,1]$, from which it follows that

$(\omega^*, \mu^*) \in \arg\max_{(\omega,\mu)\in[0,1]\times[0,\infty)} \widehat{\phi}(\omega, \mu)$ and $\widehat{\phi}(\omega^*, \mu^*) = \|\boldsymbol{b} - \boldsymbol{s}\|^2 > 0$. By $\widehat{\phi}(\omega^*, 0) = \widehat{\phi}(\omega^*, 2\mu^*) = 0$, (4.3.16) holds.

(ii) Assume $\delta \in (1, \infty)$. In this case, $\eta \geq \zeta$, hence $(\omega^*, \mu^*) = (1, 1)$. In a similar way to (a), we obtain

$$\widehat{\phi}(\omega, \mu) = \|\boldsymbol{a} - \boldsymbol{s}\|^2 - [\omega + (1-\omega)/\delta]^2 \|\boldsymbol{a} - \boldsymbol{s}\|^2 \left[\mu - \frac{1}{\omega + (1-\omega)/\delta}\right]^2.$$

By $[\omega + (1-\omega)/\delta]^2 \|\boldsymbol{a} - \boldsymbol{s}\|^2 > 0$ and $\omega + (1-\omega)/\delta > 0$, $\widehat{\phi}(\omega, \mu)$ is maximized by

$$\mu = \frac{1}{\omega + (1-\omega)/\delta}, \ \forall \omega \in [0, 1],$$

from which it follows that

$$(\omega^*, \mu^*) \in \underset{(\omega,\mu)\in[0,1]\times[0,\infty)}{\arg\max} \widehat{\phi}(\omega, \mu)$$

and $\widehat{\phi}(\omega^*, \mu^*) = \|\boldsymbol{a} - \boldsymbol{s}\|^2 > 0$. By $\widehat{\phi}(\omega^*, 0) = \widehat{\phi}(\omega^*, 2\mu^*) = 0$, (4.3.16) holds.

(II) Assume $\Pi(\boldsymbol{s}, \boldsymbol{a}) \cap \Pi(\boldsymbol{s}, \boldsymbol{b}) \neq \emptyset$. In this case, by (I),

$$\widehat{\phi}(\omega, \mu) = -\mu^2 \|\omega(\boldsymbol{a} - \boldsymbol{s}) + (1-\omega)(\boldsymbol{b} - \boldsymbol{s})\|^2 + 2\mu \left[\omega \|\boldsymbol{a} - \boldsymbol{s}\|^2 + (1-\omega) \|\boldsymbol{b} - \boldsymbol{s}\|^2\right].$$

Here, it is easily verified that $\|\omega(\boldsymbol{a} - \boldsymbol{s}) + (1-\omega)(\boldsymbol{b} - \boldsymbol{s})\|^2 = 0 \Leftrightarrow$ one of the following conditions hold:

(i) $\boldsymbol{a} = \boldsymbol{b} = \boldsymbol{s} \{\Leftrightarrow \widehat{\phi}(\omega, \mu) = 0, \ \forall \omega \in [0, 1], \ \forall \mu \in [0, \infty)\}$
(ii) $\boldsymbol{a} = \boldsymbol{s} \neq \boldsymbol{b}, \ \omega = 1 \ \{\Leftrightarrow \widehat{\phi}(\omega, \mu) = 0, \ \forall \mu \in [0, \infty)\}$
(iii) $\boldsymbol{b} = \boldsymbol{s} \neq \boldsymbol{a}, \ \omega = 0 \ \{\Leftrightarrow \widehat{\phi}(\omega, \mu) = 0, \ \forall \mu \in [0, \infty)\}$.

Since i): $\widehat{\phi}(\omega, 0) = 0$, $\forall \omega \in [0, 1]$, and ii): $\max_{(\omega,\mu)\in[0,1]\times[0,\infty)} \widehat{\phi}(\omega, \mu) \geq 0$ [by i)], we can assume $\|\omega(\boldsymbol{a} - \boldsymbol{s}) + (1-\omega)(\boldsymbol{b} - \boldsymbol{s})\|^2 \neq 0$ in the following.

Now, we obtain

$$\begin{aligned}\widehat{\phi}(\omega,\mu) &= -\left\|\omega(\boldsymbol{a}-\boldsymbol{s})+(1-\omega)(\boldsymbol{b}-\boldsymbol{s})\right\|^2\left(\mu-\frac{\omega\left\|\boldsymbol{a}-\boldsymbol{s}\right\|^2+(1-\omega)\left\|\boldsymbol{b}-\boldsymbol{s}\right\|^2}{\left\|\omega(\boldsymbol{a}-\boldsymbol{s})+(1-\omega)(\boldsymbol{b}-\boldsymbol{s})\right\|^2}\right)^2 \\ &\quad +\frac{\left[\omega\left\|\boldsymbol{a}-\boldsymbol{s}\right\|^2+(1-\omega)\left\|\boldsymbol{b}-\boldsymbol{s}\right\|^2\right]^2}{\left\|\omega(\boldsymbol{a}-\boldsymbol{s})+(1-\omega)(\boldsymbol{b}-\boldsymbol{s})\right\|^2}, \end{aligned} \tag{4.4.19}$$

which yields

$$\mu^*_\omega := \frac{\omega\left\|\boldsymbol{a}-\boldsymbol{s}\right\|^2+(1-\omega)\left\|\boldsymbol{b}-\boldsymbol{s}\right\|^2}{\left\|\omega(\boldsymbol{a}-\boldsymbol{s})+(1-\omega)(\boldsymbol{b}-\boldsymbol{s})\right\|^2} \in \underset{\mu\in[0,\infty)}{\arg\max}\,\widehat{\phi}(\omega,\mu).$$

All that we have to prove for (4.3.15) is that $\omega^* \in \arg\max_{\omega\in[0,1]}\widetilde{\phi}(\omega)$ and $\mu^*_{\omega^*} = \mu^*$, where $\widetilde{\phi}(\omega) := \widehat{\phi}(\omega,\mu^*_\omega)$. It is verified that

$$\begin{aligned}\widetilde{\phi}(\omega) &= \frac{\left[\omega\left\|\boldsymbol{a}-\boldsymbol{s}\right\|^2+(1-\omega)\left\|\boldsymbol{b}-\boldsymbol{s}\right\|^2\right]^2}{\left\|\omega(\boldsymbol{a}-\boldsymbol{s})+(1-\omega)(\boldsymbol{b}-\boldsymbol{s})\right\|^2} = \frac{[\omega\xi+(1-\omega)\zeta]^2}{\omega^2\xi+2\omega(1-\omega)\eta+(1-\omega)^2\zeta} \\ &= \frac{[\omega(\xi-\zeta)+\zeta]^2}{(\xi+\zeta-2\eta)\omega^2+2(\eta-\zeta)\omega+\zeta},\end{aligned}$$

of which the derivative is as follows:

$$\begin{aligned}\left[\text{denominator of }\frac{\partial\widetilde{\phi}(\omega)}{\partial\omega}\right] &= \left\|\omega(\boldsymbol{a}-\boldsymbol{s})+(1-\omega)(\boldsymbol{b}-\boldsymbol{s})\right\|^4 > 0 \\ \left[\text{numerator of }\frac{\partial\widetilde{\phi}(\omega)}{\partial\omega}\right] &= 2\left[\omega(\xi-\zeta)+\zeta\right](\xi-\zeta)\left[(\xi+\zeta-2\eta)\omega^2+2(\eta-\zeta)\omega+\zeta\right] \\ &\quad -2\left[\omega(\xi-\zeta)+\zeta\right]^2\left[(\xi+\zeta-2\eta)\omega+(\eta-\zeta)\right] \\ &= 2\left[\omega(\xi-\zeta)+\zeta\right]\left\{\left[(\xi-\zeta)(\eta-\zeta)-\zeta(\xi+\zeta-2\eta)\right]\omega\right. \\ &\quad \left.+(\xi-\zeta)\zeta-\zeta(\eta-\zeta)\right\} \\ &= 2\left[\omega\xi+(1-\omega)\zeta\right]\Big\{\underbrace{\left[\eta(\xi+\zeta)-2\xi\zeta\right]\omega+\zeta(\xi-\eta)}_{=:\,f(\omega)}\Big\},\end{aligned} \tag{4.4.20}$$

where it is obvious that

$$\omega\xi+(1-\omega)\zeta \geq 0. \tag{4.4.21}$$

(i) Assume $\eta < \min(\xi, \zeta)$. In this case, it is not hard to see that $\xi, \zeta > 0$, from which it follows that $\eta(\xi+\zeta) - 2\xi\zeta = \xi(\eta-\zeta) + \zeta(\eta-\xi) < 0$. Hence, we have

$$f(\omega) \begin{cases} \geq 0, & \text{if } \omega \leq \dfrac{\zeta(\xi-\eta)}{2\xi\zeta - \eta(\xi+\zeta)} (= \omega^*), \\ < 0, & \text{otherwise.} \end{cases} \tag{4.4.22}$$

By (4.4.21), (4.4.22), and $\zeta(\xi-\eta)/[2\xi\zeta-\eta(\xi+\zeta)] \in (0,1)$, we obtain

$$\omega^* = \frac{\zeta(\xi-\eta)}{2\xi\zeta - (\xi+\zeta)\eta} = \arg\max_{\omega\in[0,1]} \widetilde{\phi}(\omega).$$

Moreover, we have

$$\begin{aligned} \mu^*_{\omega^*} &= \frac{[2\xi\zeta - (\xi+\zeta)\eta][(\xi-\eta)\zeta\xi + (\zeta-\eta)\xi\zeta]}{(\xi-\eta)^2\zeta^2\xi + 2(\xi-\eta)(\zeta-\eta)\zeta\xi\eta + (\zeta-\eta)^2\xi^2\zeta} \\ &= \frac{[2\xi\zeta - (\xi+\zeta)\eta](\xi+\zeta-2\eta)}{(\xi-\eta)^2\zeta + 2(\xi-\eta)(\zeta-\eta)\eta + (\zeta-\eta)^2\xi}, \end{aligned}$$

where

$$\begin{aligned} & (\xi-\eta)^2\zeta + 2(\xi-\eta)(\zeta-\eta)\eta + (\zeta-\eta)^2\xi \\ = \; & (\xi-\eta)^2\zeta + (\zeta-\eta)^2\xi + (\xi-\eta)(\zeta-\eta)(\xi+\zeta) - (\xi-\eta)(\zeta-\eta)(\xi+\zeta-2\eta) \\ = \; & (\xi-\eta+\zeta-\eta)\left[(\xi-\eta)\zeta + (\zeta-\eta)\xi - (\xi-\eta)(\zeta-\eta)\right] = (\xi+\zeta-2\eta)(\xi\zeta-\eta^2), \end{aligned}$$

from which it follows that

$$\mu^*_{\omega^*} = \frac{2\xi\zeta - (\xi+\zeta)\eta}{\xi\zeta - \eta^2} = \mu^*.$$

(ii) Assume $\eta \geq \xi$ ($\Rightarrow \eta \leq \zeta$). In this case, $f(0) = \zeta(\xi-\eta) \leq 0$, $f(1) = \xi(\eta-\zeta) \leq 0$, which implies that $f(\omega) \leq 0$, $\forall \omega \in [0,1]$, since $f(\omega)$ is *monotone*. Hence, by (4.4.20) and (4.4.21), we have $\partial\widetilde{\phi}(\omega)/\partial\omega \leq 0$, $\forall \omega \in [0,1]$, from which it follows that $\omega^* = 0 \in \arg\max_{\omega\in[0,1]} \widetilde{\phi}(\omega)$, $\mu^*_{\omega^*} = 1 = \mu^*$.

(iii) Assume $\eta \geq \zeta$ ($\Rightarrow \eta \leq \xi$). In a similar way to (ii), we have $\partial\widetilde{\phi}(\omega)/\partial\omega \geq 0$, $\forall \omega \in [0,1]$, from which it follows that $\omega^* = 1 \in \arg\max_{\omega\in[0,1]} \widetilde{\phi}(\omega)$,

$\mu^*_{\omega^*} = 1 = \mu^*$.

Finally, in cases (i)–(iii), by (4.4.19), $\widehat{\phi}(\omega^*, \mu)$ is a quadratic function in terms of μ with its maximum at $\mu = \mu^*$. Thus, over $[0, 2\mu^*]$, $\widehat{\phi}(\omega^*, \mu)$ is minimized by $\mu = 0$ or $\mu = 2\mu^*$, and moreover, $\widehat{\phi}(\omega^*, 0) = 0$ by (4.3.14). This verifies (4.3.16), which completes the proof. □

4.4.4 Proof of Proposition 4.3.11

Proof of (a): The proof is given by mathematical induction on m.

(I) $m = 0$: By the assumption and (4.3.8), we have $\left\langle \boldsymbol{h}_k - \boldsymbol{h}^{(0)}_{k,\iota}, \boldsymbol{h}^* - \boldsymbol{h}^{(0)}_{k,\iota} \right\rangle \leq 0$, $\forall \iota \in \mathcal{I}^{(0)}_k$, which shows nothing but $\boldsymbol{h}^* \in \Pi^-(\boldsymbol{h}_k, \boldsymbol{h}^{(0)}_{k,\iota})$, $\forall \iota \in \mathcal{I}^{(0)}_k$.

(II) Assume $\boldsymbol{h}^* \in \Pi^-(\boldsymbol{h}_k, \boldsymbol{h}^{(m-1)}_{k,\iota})$, $\forall \iota \in \mathcal{I}^{(m-1)}_k$, for some $m \in \{1, 2, \cdots, M-1\}$.

(i) If $\eta^{(m)}_{k,\iota} = -\sqrt{\xi^{(m)}_{k,\iota}\zeta^{(m)}_{k,\iota}} \neq 0$, then it is obvious that $\boldsymbol{h}^* \in \Pi^-(\boldsymbol{h}_k, \boldsymbol{h}^{(m)}_{k,\iota}) = \mathcal{H}$ [see (4.3.9)],

(ii) otherwise, by (4.3.9) and Definition 4.3.2,

$$\boldsymbol{h}^{(m)}_{k,\iota} = P_{\Pi^-(\boldsymbol{h}_k, \boldsymbol{h}^{(m-1)}_{k,\iota_1}) \cap \Pi^-(\boldsymbol{h}_k, \boldsymbol{h}^{(m-1)}_{k,\iota_2})}(\boldsymbol{h}_k). \tag{4.4.23}$$

By (4.4.23) and the assumption of (II), we have $\left\langle \boldsymbol{h}_k - \boldsymbol{h}^{(m)}_{k,\iota}, \boldsymbol{h}^* - \boldsymbol{h}^{(m)}_{k,\iota} \right\rangle \leq 0$, which shows nothing but $\boldsymbol{h}^* \in \Pi^-(\boldsymbol{h}_k, \boldsymbol{h}^{(m)}_{k,\iota})$.

(i) and (ii) verify $\boldsymbol{h}^* \in \Pi^-(\boldsymbol{h}_k, \boldsymbol{h}^{(m)}_{k,\iota})$, $\forall \iota \in \mathcal{I}^{(m)}_k$.

(I) and (II) complete the proof.

Proof of (b): Let $\boldsymbol{p}^{<\mathrm{g}>}_k := \sum_{j \in \mathcal{I}^{<\mathrm{g}>}_k} w^{<\mathrm{g}>}_k P_{H^-_j(\boldsymbol{h}_k)}(\boldsymbol{h}_k)$. Then, $\forall \mathrm{g} \in \{\mathrm{A}, \mathrm{B}\}$, we have

$$\begin{aligned}\left\langle \boldsymbol{h}_k - \boldsymbol{h}^{<\mathrm{g}>}_k, \boldsymbol{h}^* - \boldsymbol{h}^{<\mathrm{g}>}_k \right\rangle &= \left\langle -\mathcal{M}^{<\mathrm{g}>}_k(\boldsymbol{p}^{<\mathrm{g}>}_k - \boldsymbol{h}_k), \boldsymbol{h}^* - \boldsymbol{h}_k - \mathcal{M}^{<\mathrm{g}>}_k(\boldsymbol{p}^{<\mathrm{g}>}_k - \boldsymbol{h}_k) \right\rangle \\ &= \mathcal{M}^{<\mathrm{g}>}_k \Bigg[\sum_{j \in \mathcal{I}^{<\mathrm{g}>}_k} w^{<\mathrm{g}>}_k \left\| P_{H^-_j(\boldsymbol{h}_k)}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\|^2 - \left\langle \boldsymbol{p}^{<\mathrm{g}>}_k - \boldsymbol{h}_k, \boldsymbol{h}^* - \boldsymbol{h}_k \right\rangle \Bigg],\end{aligned} \tag{4.4.24}$$

where

$$\begin{aligned}
&\left\langle \boldsymbol{p}_k^{<\mathrm{g}>} - \boldsymbol{h}_k, \boldsymbol{h}^* - \boldsymbol{h}_k \right\rangle \\
&= \sum_{j \in \mathcal{I}_k^{<\mathrm{g}>}} w_k^{<\mathrm{g}>} \left\langle P_{H_j^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k) - \boldsymbol{h}_k, \boldsymbol{h}^* - P_{H_j^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k) + P_{H_j^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\rangle \\
&= \sum_{j \in \mathcal{I}_k^{<\mathrm{g}>}} w_k^{<\mathrm{g}>} \left\| P_{H_j^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\|^2 \\
&\quad - \sum_{j \in \mathcal{I}_k^{<\mathrm{g}>}} w_k^{<\mathrm{g}>} \left\langle \boldsymbol{h}^* - P_{H_j^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k), \boldsymbol{h}_k - P_{H_j^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k) \right\rangle . \qquad (4.4.25)
\end{aligned}$$

By (4.4.24), (4.4.25), and the assumption of Proposition 4.3.11, we obtain $\left\langle \boldsymbol{h}_k - \boldsymbol{h}_k^{<\mathrm{g}>}, \boldsymbol{h}^* - \boldsymbol{h}_k^{<\mathrm{g}>} \right\rangle = \mathcal{M}_k^{<\mathrm{g}>} \sum_{j \in \mathcal{I}_k^{<\mathrm{g}>}} w_k^{<\mathrm{g}>} \left\langle \boldsymbol{h}^* - P_{H_j^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k), \boldsymbol{h}_k - P_{H_j^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k) \right\rangle \leq 0$, which verifies $\boldsymbol{h}^* \in \Pi^-(\boldsymbol{h}_k, \boldsymbol{h}_k^{<\mathrm{g}>})$. □

4.5 Numerical Examples

In this section, POWER-PSP I and POWER-PSP II are compared with the following algorithms: UW-PSP, APA, regularized RLS[4], and FNTF. To examine the potential ability of the proposed algorithms, simulations are performed for the system identification problem to estimate $\boldsymbol{h}^* \in \mathbb{R}^{64}$ ($N = 64$) with the USASI[5] input signal (colored and wide sense stationary) under a silent environment with $\mathrm{SNR} := 10 \log_{10} \left(E\left\{z_k^2\right\} / E\left\{n_k^2\right\} \right) = 100$ dB, where $z_k := \langle \boldsymbol{u}_k, \boldsymbol{h}^* \rangle$ and $E\{\cdot\}$ denotes expectation. We evaluate the system mismatch defined at iteration k as $10 \log_{10} \|\boldsymbol{h}^* - \boldsymbol{h}_k\|^2 / \|\boldsymbol{h}^*\|^2$ [dB], $\forall k \in \mathbb{N}$. Note that, in Chapter 5, it is shown that the proposed technique also exhibits excellent performance for the (more practical) stereo echo cancellation problem with $\boldsymbol{h}^* \in \mathbb{R}^{2000}$, SNR = 25 dB, and speech

[4] Although the regularized RLS algorithm is computationally-intensive (since efficient calculation by the matrix inversion lemma can *not* be utilized because of the regularization), we employ it to examine the potential of the proposed technique. Moreover, in the present simulations, we employ the normal APA rather than its computationally-efficient version called FAP [43], since it is reported that such efficient strategies result in somewhat inferior or similar convergence rate to the original exact algorithms [42, Ch. 2].

[5] The USASI (USA Standards Institute) generation routine is characterized as an ARMA model, which can be found at http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/txt/usasi.txt (see also [129]).

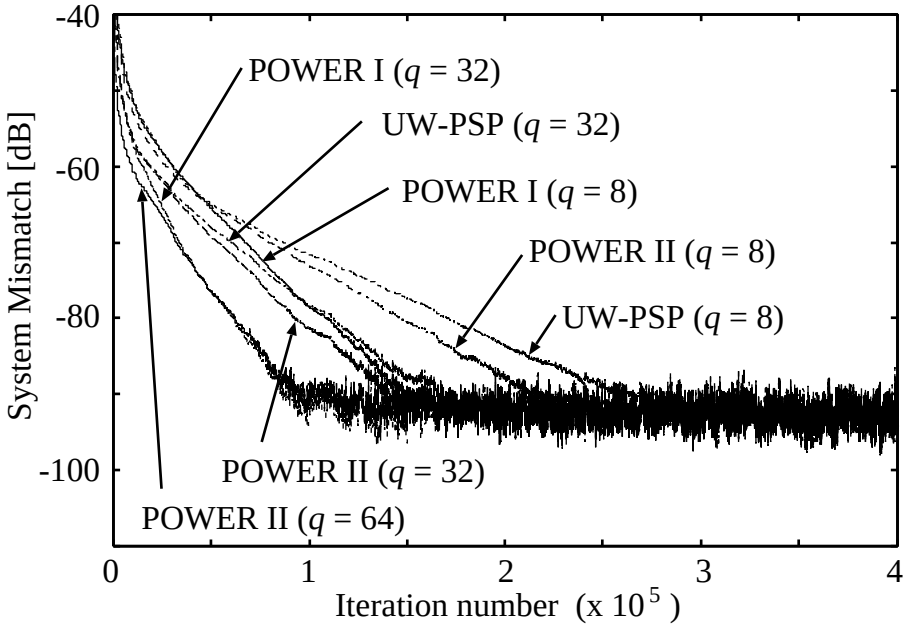


Figure 4.5.5: Performance of the proposed algorithms versus UW-PSP. $r = 1$ and $\rho = 0$ for all algorithms. For POWER-PSP I, (a) $q = 8$, $\lambda_k = 0.4$, and (b) $q = 32$, $\lambda_k = 0.1$. For POWER-PSP II, (a) $q = 8$, $\lambda_k = 0.6$, (b) $q = 32$, $\lambda_k = 0.2$, and (c) $q = 64$, $\lambda_k = 0.2$. For UW-PSP, (a) $q = 8$, $\lambda_k = 1.0$, and (b) $q = 32$, $\lambda_k = 0.2$.

(non-stationary) inputs.

4.5.1 Proposed versus UW-PSP Algorithm

The stochastic property sets are designed, for all employed algorithms, with $r = 1$ and $\rho = \rho_3$ $(= 0)$ (see [129]). The control sequences are designed as (i) $\mathcal{I}_k = \{k, k-1, \cdots, k-q+1\}$ for UW-PSP, (ii) the binary-tree construction for POWER-PSP I (Example 4.3.5) with $\bar{\mathcal{I}}_k$ identical to $\mathcal{I}_k$ for UW-PSP, and (iii) $\mathcal{I}_k^{<\mathrm{A}>} = \{k, k-1, \cdots, k-q/2+1\}$ and $\mathcal{I}_k^{<\mathrm{B}>} = \{k-q/2, \cdots, k-q+1\}$ for POWER-PSP II.

The results are depicted in Fig. 4.5.5. We observe that the proposed algorithms significantly improve the speed of convergence, as expected by the discussion in Sec. 4.3.3.

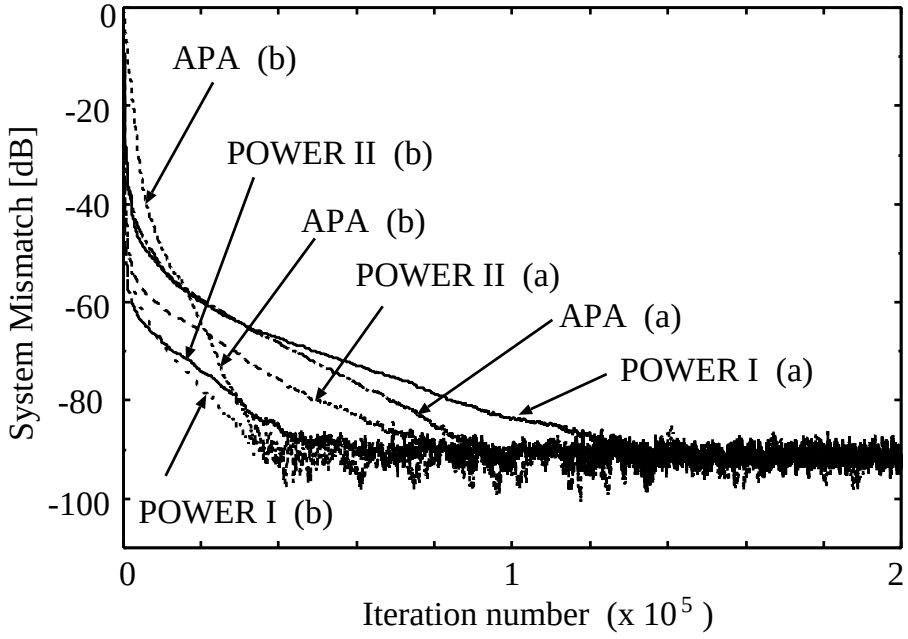


Figure 4.5.6: Performance of the proposed algorithms versus APA. For APA, (a) $r = 4$, $\lambda_k = 0.2$, and (b) $r = 32$, $\lambda_k = 0.002$. For POWER-PSP I, (a) $q = 4$, $\lambda_k = 1.0$, and (b) $q = 128$, $\lambda_k = 0.05$. For POWER-PSP II, (a) $q = 64$, $\lambda_k = 0.6$, and (b) $q = 128$, $\lambda_k = 0.6$. The other parameters are the same as in Fig. 4.5.5.

4.5.2 Proposed versus APA, Regularized RLS, and FNTF

For APA, we set (a) $r = 4$, $\lambda_k = 0.2$, and (b) $r = 32$, $\lambda_k = 0.002$ (see [129, Sec. II-B]). For regularized RLS, we utilize the inverse of the regularized sample covariance matrix $\boldsymbol{R}_k := \lambda \boldsymbol{R}_{k-1} + \boldsymbol{u}_k \boldsymbol{u}_k^T + \delta \boldsymbol{I}$, $\forall k \in \mathbb{N}$. Here, $\lambda = 1 - 1/(3N) = 0.99479 \in (0, 1)$ is the forgetting factor and $\delta = 0.04 \approx 0.0056\sigma_u^2$ is the regularization parameter, where σ_u^2 is the variance of input data $(u_k)_{k \in \mathbb{N}}$ (the choice of $0 < \delta < 0.01\sigma_u^2$ is recommended [56, p. 570]). The matrix is initialized as $\boldsymbol{R}_0 = \mathrm{diag}\{\lambda^{N-1}, \lambda^{N-2}, \cdots, 1\}$ [40]. For FNTF, we set (a) $L_{\mathrm{FNTF}} = 20$, $\lambda = 1 - 1/(3N) = 0.99479$, and (b) $L_{\mathrm{FNTF}} = 64$, $\lambda = 1 - 1/(9N) = 0.99826$, where $L_{\mathrm{FNTF}} \in \{0, 1, \cdots, N\}$ is the order of the AR input process modeled in FNTF. (Although the order of the AR input process is denoted by "N" in [82], the order of FNTF is expressed as "L_{FNTF}" to distinguish from the filter length "N".) In the prediction part, we employ the method in [83, Table I] with a modification for stabilization by following the recommendation in [82]; we use the backward prediction error $e^{\mathrm{b}}_{L_{\mathrm{FNTF}}}(T) = x(T - L_{\mathrm{FNTF}}) - B'_{L_{\mathrm{FNTF}},T-1} X_{L_{\mathrm{FNTF}},T}$ instead of its

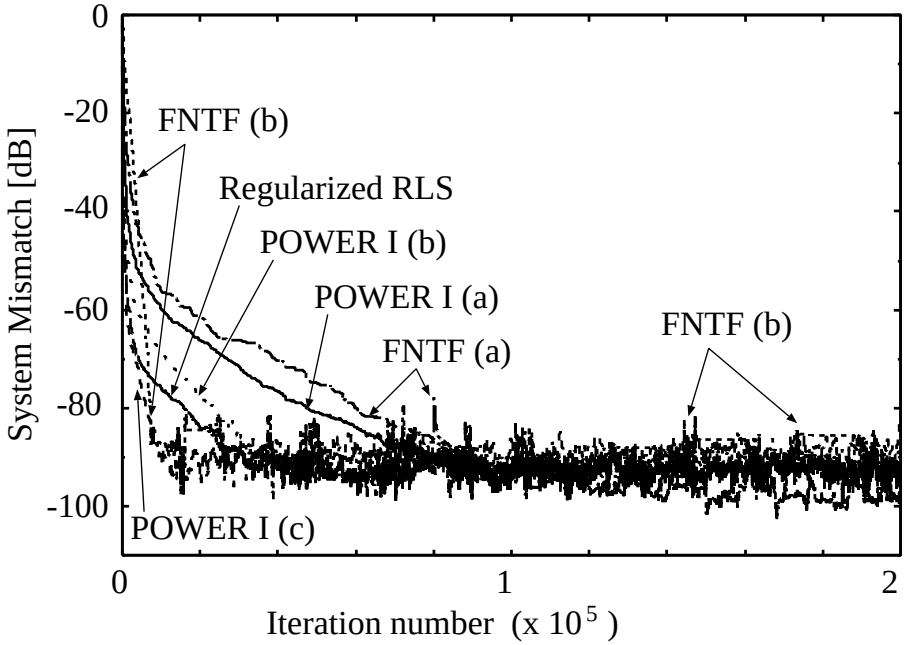


Figure 4.5.7: Performance of POWER-PSP I versus and regularized RLS and FNTF. For regularized RLS, $\lambda = 1-1/(3N)$, $\delta = 0.04$. For FNTF, (a) $L_{\mathrm{FNTF}} = 20$, $\lambda = 1 - 1/(3N)$, and (b) $L_{\mathrm{FNTF}} = 64$, $\lambda = 1 - 1/(9N)$. For POWER-PSP I, (a) $q = 128$, $\lambda_k = 0.05$, (b) $q = 256$, $\lambda_k = 0.05$, and (c) $q = 512$, $\lambda_k = 0.05$. The other parameters are the same as in Fig. 4.5.5.

equivalent form $e^{\mathrm{b}}_{L_{\mathrm{FNTF}}}(T) = -\lambda\beta_{L_{\mathrm{FNTF}}}(T-1)W^{L_{\mathrm{FNTF}}+1}_{L_{\mathrm{FNTF}}+1,T}$ (see [82, Eq. (8) and Table 1] for the definitions of the undefined variables). It is reported that the FRLS algorithms, including FNTF, have a tendency to diverge [50, 83]. Such divergence phenomena were observed in our experiments, and thus, we reinitialize the parameters for prediction based on the monitoring strategy used in [14, 23, 82].

A comparison with APA is depicted in Fig. 4.5.6, and one with regularized RLS and FNTF in Fig. 4.5.7. We see that the proposed algorithms outperform the other existing algorithms. Detailed discussion of the simulation results along with the computational aspects follows below.

4.6 Discussion

The computational complexities of the algorithms used in the simulations are as follows: for POWER-PSP I, $(3M + 2r + 1)N + 21M + r$ $(M = \log_2 q)$; for

POWER-PSP II, $(2r+6)N+r$ (see Sec. 4.3.4); for UW-PSP, $(2r+4)N+r$; for APA, $(r^2+2r)N+r^3+r$ [97, p. 244]; for FAP, $2N+20r$ [50, p. 28]; for RLS, N^2+5N [97, p. 247]; for FNTF, $2N+9L_{\mathrm{FNTF}}$ [50, p. 28]; and, for regularized RLS, N^3 (computation for the inverse of $N \times N$ matrix). For the proposed algorithms and UW-PSP, being *inherently parallel*, we show the complexity with q parallel processors being engaged. The complexities are graphically drawn in Fig. 4.6.8. The horizontal axis corresponds to the value assigned to the parameter of which an increase is expected to improve the performance: q for POWER-PSP I, II, and UW-PSP; r for APA and FAP; and, $L_{\mathrm{FNTF}} \in \{0, 1, \cdots, N\}$ for FNTF. Note that we fix $r = 1$, thus the computational complexities of POWER-PSP II and UW-PSP are constant in terms of q. We see that an increase of q causes a very slight increase of complexity, even compared with the efficient FAP algorithm. Another remarkable feature of adaptive-PSP (including POWER-PSP I, II) is that the algorithm can normally perform with just a slight loss in convergence speed in a situation when some of the engaged processors are damaged. This implies that the algorithm is endowed with a *fault-tolerance* nature thanks to its inherently parallel structure [4, 17, 22, 24]. It should be mentioned that the other conventional algorithms, being *not* inherently parallel, can of course be implemented somehow in parallel. However, we omit to show the complexity with parallel processors for those algorithms, since it depends on many possible ways to decompose tasks.

From Figs. 4.5.5 and 4.5.6, we see that the POWER-PSP II shows excellent convergence behavior for $q = 64$ and $q = 128$ with low computational cost. Moreover, from Figs. 4.5.6 and 4.5.7, POWER-PSP I achieves faster convergence even compared with the regularized RLS algorithm. Among all our experiments, the proposed algorithms always exhibited stable behavior in steady-state, while this did not always occur for APA and FNTF. Actually, APA shows certain instability when we use a relatively large step size for a large r (note: we use a common step size for POWER I with $q = 128$, 256, and 512). This is why we choose such a small step size $\lambda_k = 0.002$ for APA ($r = 32$), which results in the slow convergence at the beginning (as observed in Fig. 4.5.6). On the other hand, in Fig. 4.5.7, we observe non-negligible spikes on the curve of FNTF, which indicates the intrinsic instability of the FRLS algorithms (see, e.g., [11, p. 77], [50], [42, p. 40]). A certain loss for FNTF is observed in the initial convergence speed, even though we use the initialization recommended in [83] and carefully tuned the forgetting factor λ.

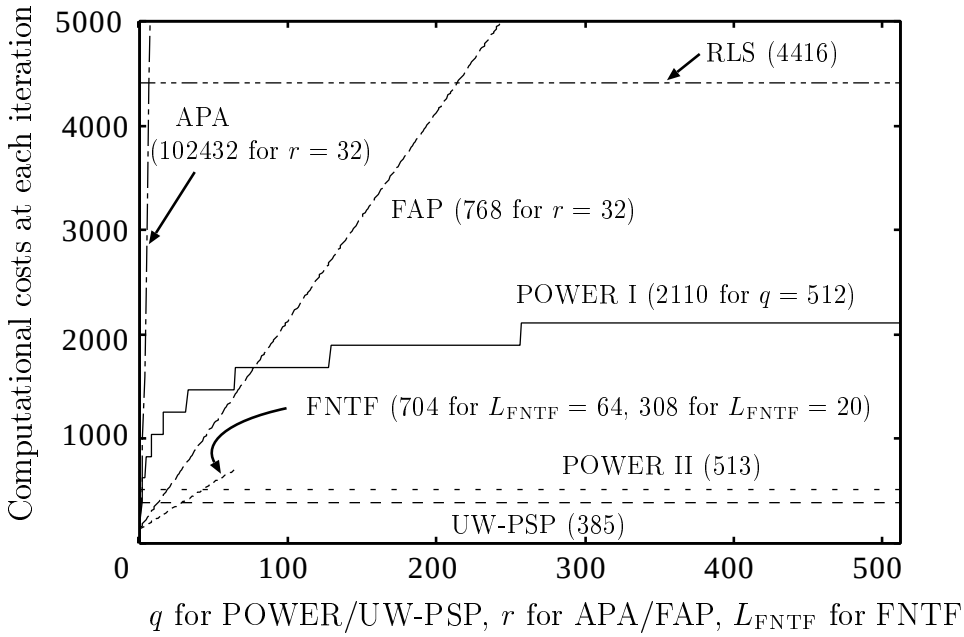


Figure 4.6.8: Computational complexity of the adaptive filtering algorithms for $N = 64$. For the proposed algorithms, we set $r = 1$ and use q parallel processors.

This observed loss in a stationary environment stems from the same reason as the tracking inferiority of RLS in nonstationary environments with a constant λ (as remarked, e.g., in [54, 57, 71]), since initialization contains, in general, some model mismatch.

4.7 Conclusion

To the best of the authors' knowledge, such excellent performance, achieved by the proposed algorithms with a large q, has not yet been previously reported. We believe that the POWER weighting is a promising adaptive filtering technique, for possibly nonstationary inputs, that satisfies the following requirements: (i) fast convergence, (ii) linear computational complexity, (iii) numerical stability, and (iv) robustness against noise. Robustness against noise is naturally expected because of the use of stochastic property sets and, in [134], it has been verified in its application to stereo echo cancellation [108]. We finally remark that the

technique will likely be a useful tool in static convex feasibility problems[6].

[6]The convex feasibility problem is to find a point in the nonempty intersection of a *fixed* collection of closed convex sets [4, 22, 24]. Its applications include matrix estimation, image reconstruction, the inverse problem in radiation therapy treatment, etc. [22].

Chapter 5

Efficient Fast Stereo Acoustic Echo Cancellation Based on Pairwise Optimal Weight Realization Technique

Summary

In Stereophonic Acoustic Echo Cancellation (SAEC) problem, fast and accurate tracking of echo path is strongly required for stable echo cancellation. In this chapter, we propose a class of efficient fast SAEC schemes with linear computational complexity (w.r.t. filter length). The proposed schemes are based on Pairwise Optimal Weight Realization (POWER) technique, thus realizing a "best" strategy (in the sense of pairwise and worst-case optimization) to use multiple state information obtained by preprocessing. Numerical examples demonstrate that the proposed schemes significantly improve the convergence behavior compared with conventional methods in terms of system mismatch as well as Echo Return Loss Enhancement (ERLE).

5.1 Introduction

The ultimate goal of this chapter is to develop an efficient adaptive filtering scheme, with linear computational complexity, to stably cancel acoustic coupling, from loudspeakers to microphones, occurring in telecommunications with stereophonic audio systems. This acoustic coupling is commonly called *acoustic echo* (we just call it echo in the following). The Stereophonic Acoustic Echo Cancellation (SAEC) problem has become a central issue when we design high-quality, hands-free and full-duplex systems (e.g., advanced teleconferencing etc.) [8, 11, 13, 16, 34, 37, 42, 44, 45, 65, 107, 108, 111]. A direct application of a monaural echo canceling algorithm to SAEC usually results in unacceptably slow convergence [37, 107, 108], and this phenomenon is mathematically clarified that the normal equation to be solved for minimization of residual echo is often ill-conditioned or has infinitely-many solutions due to inherent dependency caused by highly cross-correlated stereo input signals [13] (see Sec. 5.2.2).

Decorrelation of the inputs is a pathway to fast and accurate tracking of echo paths (impulse responses), which is necessary for stable echo cancellation [34, 44, 66, 112]. A great deal of effort has been devoted to devise preprocessing of the inputs [2, 12, 13, 46, 47, 49, 60, 66, 98, 108, 112] (see Sec. 5.2.3). In other words, these preprocessing techniques relax the ill-conditioned situation with use of *additional information* provided artificially by feeding less cross-correlated input signals. Based on the preprocessing [13], real-time SAEC systems have been effectively implemented, e.g., in [16, 34]. Under rapidly time-varying situations, however, further convergence acceleration is strongly required. Unfortunately, an increase of decorrelation effects by preprocessing may cause audible acoustic-distortion or loss of stereo sound effects, thus the preprocessing is strictly restricted to only slight modification of the input signal. The remaining major challenges in SAEC with preprocessing are twofold: (i) fast tracking of the echo paths within the above restriction on audio effects and (ii) low computational complexity due to necessity to adapt 4 echo cancelers with a few thousands taps [42] (see Fig. 5.1.1). Now, the time is ripe to move from the early stage of devising preprocessing techniques to the next stage: utilize the additional information provided by preprocessing to the fullest extent possible.

Effective utilization of the additional information is a key to achieve the goal

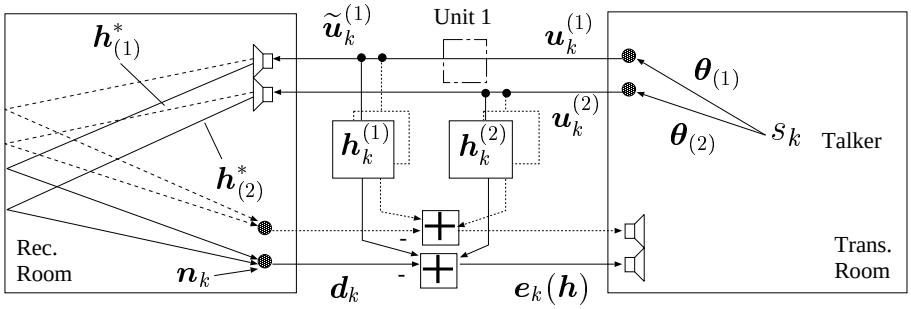


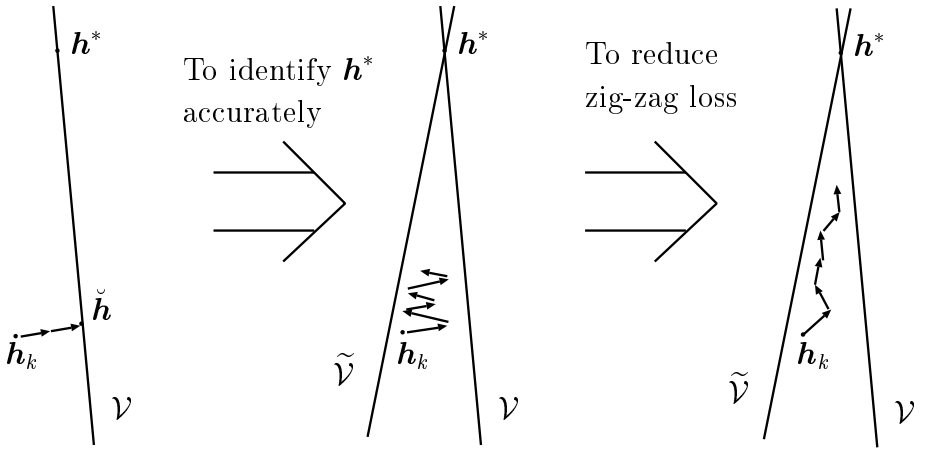
Figure 5.1.1: Stereophonic acoustic echo canceling scheme; Unit 1 is a preprocessing unit (see Sec. 5.2.3). Note that the system is not limited to this special structure but can be any appropriate structure.

shown in the beginning of this introduction. We formulate the SAEC problem as a *time-varying set-theoretic adaptive filtering*, i.e., approximate the *estimandum* $\boldsymbol{h}^*$ (system to be estimated, true echo paths) as a point in the intersection of multiple closed convex sets that are defined with observable data and contain $\boldsymbol{h}^*$ with high probability (see Sec. 5.3.1). As a preliminary step [138], we found a clue to maximally utilize the information given by the preprocessing [66, 112]. The preprocessing in [66, 112] alternately generates certain two states of inputs (see Sec. 5.2.3) and it is reported that it achieves faster convergence in system mismatch[1], at the expense of slower convergence in Echo Return Loss Enhancement (ERLE), than other major preprocessing techniques such as in [13]. The scheme[2] proposed in [138] utilizes the information from the two states of inputs simultaneously at each iteration. The two states can be associated with two states of solution sets (mathematically linear varieties [13]), say $\mathcal{V}$ and $\widetilde{\mathcal{V}}$. By using the adaptive Parallel Subgradient Projection (PSP) algorithm [129] (see Sec. 5.3.1), the scheme fairly reduces the *zig-zag loss*[3] shown in Fig. 5.1.2-(b), and the direction

[1]Recall that the fast and accurate estimation of $\boldsymbol{h}^*$ is necessary in SAEC, hence system mismatch is a very important criterion.

[2]The scheme is derived from *the Adaptive Projected Subgradient Method* [124, 126], a unified framework for various adaptive filtering algorithms, which has also been applied to the multiple access interference suppression problem in DS/CDMA systems successfully [20, 132] (see Chapter 3).

[3]The loss is caused by the "small" angle between $\mathcal{V}$ and $\widetilde{\mathcal{V}}$ due to the restriction of "slight" modification in preprocessing (see e.g., [30, p.197] for angle between subspaces or linear varieties). Similar *zig-zag* behavior can be observed for alternating projection methods known as *Kaczmarz's method* or, more generally, the Projections Onto Convex Sets (POCS) in *convex feasibility problem*; find a point in the nonempty intersection of *fixed* closed convex sets (see



(a) Straightforward (b) Conventional (c) UW-PSP

Figure 5.1.2: A geometric interpretation of existing methods: (a) straightforward: straightforward application of monaural scheme; (b) conventional: preprocessing-based approach with just one state of inputs at each iteration; and (c) UW (Uniform Weight)-PSP: preprocessing-based approach with two state information at each iteration [138]. The solution set $\mathcal{V}$ is periodically changed into $\widetilde{\mathcal{V}}$ by preprocessing ($\mathcal{V}$ and $\widetilde{\mathcal{V}}$ are *linear varieties*). Note that each arrow of "conventional" stands for the update accumulated during a half cycle-period in which the state of inputs is constant.

of its update is governed by certain weighting factors [see Fig. 5.1.2-(c)]. However, the update direction realized by the uniform weights does not sufficiently approximate ideal one. Recently, an efficient strategic weight design called the *Pairwise Optimal Weight Realization (POWER)* has been developed in [135, 137] for the adaptive-PSP algorithm. The POWER technique realizes a best strategy (in the sense of pairwise and worst-case optimization) for the use of multiple information to determine the update direction. This suggests that further drastic acceleration is highly expected by exploiting POWER (see Fig. 5.1.3).

In this chapter, we propose a class of efficient fast SAEC schemes that further accelerate the method in [138] by employing POWER with keeping linear computational complexity. In fact, the POWER technique exerts far-reaching effects in a

e.g., [22] and Sec. 5.3.1). In the case of two subspaces M_1 and M_2, the rate of convergence of alternating projection methods is exactly given as $(\cos(M_1, M_2))^{2k-1}$ [30, 9.31 Theorem], where $\cos(\cdot,\cdot)$ denotes the cosine of the angle between two subspaces and k the iteration number. This provides theoretical verification to slow convergence caused by the zig-zag loss when the angle between two subspaces is small.

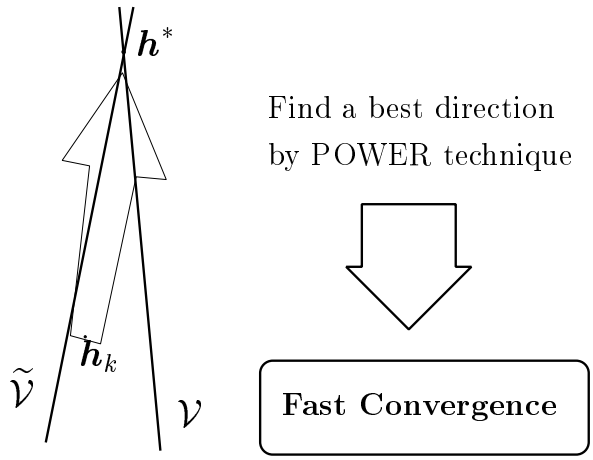


Figure 5.1.3: The direction of the proposed technique.

general adaptive filtering application, especially when the input signals are highly correlated. Hence, as seen from Fig. 5.1.2, POWER is particularly suitable for the SAEC problem. As presented in the previous chapter, the POWER technique is based on a simple formula to give the projection onto the intersection of two *closed half-spaces* that are defined by three vectors (see Proposition 4.3.3). We propose two schemes in the proposed class. The first scheme (Type I) exploits the formula in a combinatorial manner [see Fig. 5.3.5-(a)]. The second scheme (Type II), on the other hand, exploits the formula just once after taking respective uniform averages of projections corresponding to each state of inputs [see Fig. 5.3.5-(b)]. The latter scheme is computationally more efficient than the former one, while overall complexities, including the weight design, of both schemes are kept linear w.r.t. the filter length [see Remark 5.3.3-(a)].

Numerical examples demonstrate that notable improvements are achieved, in system mismatch as well as ERLE, by the use of POWER in place of the uniform weights. Other possible ways to reduce the zig-zag loss would be to employ the Affine Projection Algorithm (APA) [58, 89] or the Recursive Least Squares (RLS) algorithm [56, 97] (The essential difference between our approach and APA is clearly described in Sec. 5.3.2). The proposed schemes are also compared with such other schemes, all of which employ the same preprocessing technique as the proposed schemes do. From our numerical experiments, we verify superiority of the proposed method. Moreover, we confirm that the proposed schemes exhibit

excellent tracking behavior after a change of the echo paths.

5.2 Preliminaries

5.2.1 Stereo Acoustic Echo Cancellation Problem

Let $L_{\text{tran}} \in \mathbb{N}^*$ denote the length (of the impulse response) of the transmission path and $N \in \mathbb{N}^*$ the length of the echo path. For simplicity, let the length of the adaptive filter be N (Analyses for more general cases are presented in [13]). Referring to Fig. 5.1.1, the signals at time $k \in \mathbb{N}$ are expressed as follows.

- speech vector: $\boldsymbol{s}_k \in \mathbb{R}^{L_{\text{tran}}}$
- ith transmission path: $\boldsymbol{\theta}_{(i)} \in \mathbb{R}^{L_{\text{tran}}}$ $(i = 1, 2)$
- ith input: $u_k^{(i)} := \boldsymbol{s}_k^T \boldsymbol{\theta}_{(i)} \in \mathbb{R}$
- ith input vector: $\boldsymbol{u}_k^{(i)} := [u_k^{(i)}, u_{k-1}^{(i)}, \cdots, u_{k-N+1}^{(i)}]^T \in \mathbb{R}^N$
- preprocessed version of $\boldsymbol{u}_k^{(1)}$: $\widetilde{\boldsymbol{u}}_k^{(1)} \in \mathbb{R}^N$
- input vector: $\boldsymbol{u}_k := \begin{bmatrix} \widetilde{\boldsymbol{u}}_k^{(1)} \\ \boldsymbol{u}_k^{(2)} \end{bmatrix} \in \mathcal{H} := \mathbb{R}^{2N}$
- input matrix: $\boldsymbol{U}_k := [\boldsymbol{u}_k, \boldsymbol{u}_{k-1}, \cdots, \boldsymbol{u}_{k-r+1}] \in \mathbb{R}^{2N \times r}$ $(r \in \mathbb{N}^*)$
- ith echo path: $\boldsymbol{h}_{(i)}^* \in \mathbb{R}^N$ $(i = 1, 2)$
- estimandum: $\boldsymbol{h}^* := \begin{bmatrix} \boldsymbol{h}_{(1)}^* \\ \boldsymbol{h}_{(2)}^* \end{bmatrix} \in \mathcal{H}$
- adaptive filter (echo canceler): $\boldsymbol{h}_k := \begin{bmatrix} \boldsymbol{h}_k^{(1)} \\ \boldsymbol{h}_k^{(2)} \end{bmatrix} \in \mathcal{H}$
- noise: $\boldsymbol{n}_k := [n_k, n_{k-1}, \cdots, n_{k-r+1}]^T \in \mathbb{R}^r$
- output: $\boldsymbol{d}_k := \boldsymbol{U}_k^T \boldsymbol{h}^* + \boldsymbol{n}_k \in \mathbb{R}^r$
- residual error function: $\boldsymbol{e}_k(\boldsymbol{h}) := \boldsymbol{U}_k^T \boldsymbol{h} - \boldsymbol{d}_k \in \mathbb{R}^r$

Here, $\mathcal{H}(:= \mathbb{R}^{2N})$ is a real Hilbert space equipped with the inner product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^T \boldsymbol{y}$, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{H}$, and its induced norm $\|\boldsymbol{x}\| := \left(\boldsymbol{x}^T \boldsymbol{x}\right)^{1/2}$, $\forall \boldsymbol{x} \in \mathcal{H}$.

The goal of the SAEC problem is to cancel the echo *stably*; i.e., $\boldsymbol{u}_k^T \boldsymbol{h}^* - \boldsymbol{u}_k^T \boldsymbol{h}_k \approx 0$, $\forall k \in \mathbb{N}$. Since only $\boldsymbol{u}_k$ and $\boldsymbol{d}_k$ are observable, a common alternative goal is to suppress the residual echo; i.e., $\boldsymbol{e}_k(\boldsymbol{h}_k) \approx \boldsymbol{0}$, $\forall k \in \mathbb{N}$.

5.2.2 Non-Uniqueness Problem

In 1991, Sondhi and Morgan found unacceptably slow convergence phenomena in SAEC [107] and, in 1995, Sondhi, Morgan and Hall showed that the primitive solution set, obtained from the normal equation to be solved for minimization of the residual echo, is too large and it depends on the transmission paths (due to inherent dependency caused by highly cross-correlated stereo input signals) [108]. This fundamental difficulty, deeply seated in SAEC, is commonly referred to as *non-uniqueness problem*, which has earned recognition as an intrinsic burden *not* existing in the monaural echo cancellation. In 1998, Benesty, Morgan and Sondhi further clarified this problem, and showed that the normal equation is often ill-conditioned or has infinitely-many solutions [13].

Let us simply explain the non-uniqueness problem mathematically. The input sequence $(u_k^{(i)})_{k\in\mathbb{N}}$, $i = 1, 2$, can be written as

$$u_k^{(i)} = s_k * \boldsymbol{\theta}_{(i)}, \tag{5.2.1}$$

where $*$ denotes *convolution*. Considering the case of $N = L_{\text{tran}}$, for simplicity,

$$\breve{\boldsymbol{h}} := \begin{bmatrix} \breve{\boldsymbol{h}}^{(1)} \\ \breve{\boldsymbol{h}}^{(2)} \end{bmatrix} := \boldsymbol{h}^* + \alpha \begin{bmatrix} \boldsymbol{\theta}_{(2)} \\ -\boldsymbol{\theta}_{(1)} \end{bmatrix}, \ \alpha \in \mathbb{R}, \tag{5.2.2}$$

satisfies

$$\sum_{i=1,2} u_k^{(i)} * \breve{\boldsymbol{h}}^{(i)} = \sum_{i=1,2} u_k^{(i)} * \boldsymbol{h}_{(i)}^*, \tag{5.2.3}$$

which implies, under noiseless environments, that $\boldsymbol{e}_k(\breve{\boldsymbol{h}}) = \boldsymbol{0}$. This is the basic mechanism of the non-uniqueness problem [13] (Precise analysis is possible by using z-transform of (5.2.3) with (5.2.1); see e.g., [65]). From (5.2.2), we see that filter coefficients that cancel the echo depend on the transmission paths $\boldsymbol{\theta}_{(1)}$ and $\boldsymbol{\theta}_{(2)}$. This implies that, without well-approximating $\boldsymbol{h}^*$, echo will relapse by change of $\boldsymbol{\theta}_{(1)}$ and $\boldsymbol{\theta}_{(2)}$ due to talker's alternation etc (see also [138, Appendix A]). Hence, it is strongly desired to keep $\boldsymbol{h}_k$ close to $\boldsymbol{h}^*$ before the transmission paths change drastically.

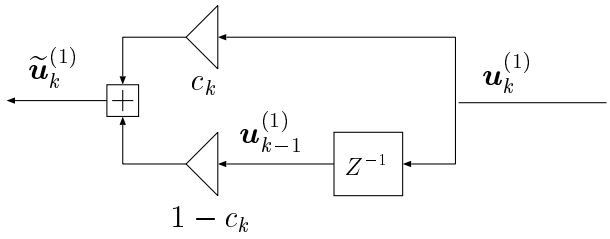


Figure 5.2.4: A preprocessing unit called input sliding. The factor c_k slides between 0 and 1 periodically, and thus, $\widetilde{\boldsymbol{u}}_k^{(1)} := c_k \boldsymbol{u}_k^{(1)} + (1-c_k)\boldsymbol{u}_{k-1}^{(1)}$ is a periodically delayed version of $\boldsymbol{u}_k^{(1)}$.

5.2.3 Preprocessing Techniques

As stated in Sec. 5.2.2, the difficulty of non-uniqueness has been known to be inherent in the SAEC problem. To alleviate this difficulty, several excellent preprocessing techniques[4] were proposed; half-wave rectifier [13] (see [46] for its improved version), comb filtering [12, 108], additive noise [47, 49] and time-varying filtering [2, 66, 112] (see [60] for its generalized version of [66]); another nonlinear preprocessing technique is also proposed in [98]. Indeed, efficacy of several nonlinear preprocessing techniques has been quantified with mutual coherence of the stereo inputs [80].

Figure 5.2.4 illustrates a simple example of the preprocessing unit generating two states of inputs (see also Fig. 5.1.1). In [66, 112], it is reported that periodic one-sample-delays, in one side of stereo inputs (i.e., $\boldsymbol{u}_k^{(1)}$ in Fig. 5.1.1), realize accurate echo path identification without audible degradation in speech. Since $\boldsymbol{u}_k^{(1)}$ is generated by convolution of the talker's speech s_k with the transmission path $\boldsymbol{\theta}_{(1)}$, the periodic delays virtually give one-sample-shift to $\boldsymbol{\theta}_{(1)}$. In other words, the preprocessing technique introduces a slightly modified state of input and alternates two[5] (modified and non-modified) states of inputs periodically, leading to alternation of two states of transmission path, say $\boldsymbol{\theta}_{(1)}$ and $\widetilde{\boldsymbol{\theta}}_{(1)}$. As a result, since the solution set depends on transmission paths as mentioned above, two slightly different solution sets, $\mathcal{V}(\boldsymbol{\theta}_{(1)})$ and $\mathcal{V}(\widetilde{\boldsymbol{\theta}}_{(1)})$ (corresponding to $\mathcal{V}$ and $\widetilde{\mathcal{V}}$

[4]Some non-preprocessing techniques were also proposed with an advantage of no degradation in input signals [59, 68, 70, 96], however, their tracking speed of echo paths is somewhat inferior to some preprocessing techniques.

[5]Although the number of states could be generalized to more than two by generating more than one modified states, we adopt two states for simplicity.

in Fig. 5.1.2, respectively), are generated alternately.

5.3 Proposed Class of Stereo Acoustic Echo Cancellation Schemes

In this section, we present a class of set-theoretic SAEC schemes based on the POWER weighting technique. The proposed approach utilizes *parallel projection* onto certain closed convex sets. First, we provide a brief introduction of set-theoretic adaptive filtering and define the closed convex sets. Then, we show the relationship between the proposed approach and the APA-based method. Finally, we present the proposed schemes in a simple manner.

5.3.1 Set-Theoretic Adaptive Filtering and Convex Set Design

We briefly introduce the basic idea of the *set-theoretic* [24, 25, 124, 126, 129] / *set-membership* [51, 52] approaches in the adaptive filtering. Let us first start with the set-theoretic approach[6] in the static *convex feasibility problem* [4, 22, 24, 25]; find a point in the nonempty intersection of *fixed* closed convex sets S_i, $i \in \mathcal{I} \subset \mathbb{N}$. Each set S_i is designed based on available information, such as noise statistics and observed data, so that S_i contains the estimandum $\boldsymbol{h}^*$ with high probability. Suppose that $\boldsymbol{h}^* \in S_i$, $\forall i \in \mathcal{I}$. Then, it is a natural strategy to find a point in $\bigcap_{i\in\mathcal{I}} S_i$ as an estimate of $\boldsymbol{h}^*$. Due to the nonlinear nature of the problem, certain successive numerical approximations by utilizing the information on each set S_i infinitely-many times are, in general, necessary.

In [129], the adaptive filtering problem is translated into a *time-varying* version of the convex feasibility problem, where multiple closed convex sets $S_i^{(k)}$, $i \in \mathcal{I}_k \subset \mathbb{N}$, are defined by multiple observable data, hence being time-varying (A unified framework for this approach is found in [124, 126]). Namely, the collection of convex sets $(S_i^{(k)})_{i\in\mathcal{I}_k}$ used at time k is varying based on data incoming from one minute to the next (Also $\boldsymbol{h}^*$ is possibly time-varying). Especially in rapidly time-varying environments, it should be reasonable to use a limited number of sets

[6] The difference is clearly stated in [24] between the set-theoretic approach and the conventional approach, i.e., optimize an objective function with or without constraints.

$(S_i^{(k)})_{i\in\mathcal{I}_k}$ that are defined with recently obtained data. This strategy agrees with saving the computational complexity, another requirement in adaptive filtering. This is the basic idea of the set-theoretic adaptive filtering approach.

The adaptive-PSP algorithm [129] has been proposed as an efficient set-theoretic adaptive filtering technique. The algorithm adopts *subgradient projections* as approximations of the exact projections onto the convex sets for saving the computation costs. The multiple (subgradient) projections are computed in parallel, hence the algorithm can save, by engaging parallel processors, the time consumption for each update. Finally, the update direction of filter is determined by taking a weighted average of the projections.

The first step is to define closed convex sets that contain $\boldsymbol{h}^*$ with high probability. A possible choice is as follows [129]:

$$C_\iota(\rho) := \left\{\boldsymbol{h}\in\mathcal{H}(:=\mathbb{R}^{2N}) : g_\iota(\boldsymbol{h}) := \|\boldsymbol{e}_\iota(\boldsymbol{h})\|^2 - \rho \le 0\right\}, \quad \forall \iota \in \mathcal{I}_k \subset \mathbb{N},\ \forall k \in \mathbb{N}, \tag{5.3.1}$$

where $\rho \ge 0$ and $\mathcal{I}_k$ is the control sequence at time k (see Sec. 5.3.3). Assignment of an appropriate value to ρ raises the membership probability $\mathrm{Prob}\{\boldsymbol{h}^* \in C_\iota(\rho)\}$ and, at the same time, keeps $C_\iota(\rho)$ sufficiently small (see Sec. 5.3.2 for detailed discussion). Since the projection onto $C_\iota(\rho)$ requires, in general, very high computational complexity, we instead employ the projection onto the closed half-space[7] [129] $H_\iota^-(\boldsymbol{h}_k) := \{\boldsymbol{x} \in \mathcal{H} : \langle \boldsymbol{x} - \boldsymbol{h}_k, \nabla g_\iota(\boldsymbol{h}_k)\rangle + g_\iota(\boldsymbol{h}_k) \le 0\} \supset C_\iota(\rho)$, which has the following simple closed-form expression:

$$P_{H_\iota^-(\boldsymbol{h}_k)}(\boldsymbol{h}) = \begin{cases} \boldsymbol{h} + \dfrac{-g_\iota(\boldsymbol{h}_k) + (\boldsymbol{h}_k - \boldsymbol{h})^T \nabla g_\iota(\boldsymbol{h}_k)}{\|\nabla g_\iota(\boldsymbol{h}_k)\|^2} \nabla g_\iota(\boldsymbol{h}_k), & \text{if } \boldsymbol{h} \notin H_\iota^-(\boldsymbol{h}_k), \\ \boldsymbol{h}, & \text{otherwise.} \end{cases}$$

Here, $\nabla g_\iota(\boldsymbol{h}_k) = 2\boldsymbol{U}_\iota \boldsymbol{e}_\iota(\boldsymbol{h}_k)$ and $P_{H_\iota^-(\boldsymbol{h}_k)}(\boldsymbol{h}) \cong P_{C_\iota(\rho)}(\boldsymbol{h})$; see [129, Fig. 3]. It should be remarked that $P_{H_\iota^-(\boldsymbol{h}_k)}(\boldsymbol{h})$ requires $O(N)$ complexity. Choosing specially

[7] Tighter closed half-spaces are also presented in [87] [see Example 6.3.2-(b)], which can also be used with the proposed schemes.

$\boldsymbol{h} = \boldsymbol{h}_k$, we have

$$P_{H_\iota^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k) = \begin{cases} \boldsymbol{h}_k - \dfrac{g_\iota(\boldsymbol{h}_k)}{\|\nabla g_\iota(\boldsymbol{h}_k)\|^2}\nabla g_\iota(\boldsymbol{h}_k), & \text{if } \boldsymbol{h}_k \notin H_\iota^-(\boldsymbol{h}_k), \\ \boldsymbol{h}_k, & \text{otherwise.} \end{cases} \tag{5.3.2}$$

5.3.2 Relationship to APA-Based Method and Robustness Issue against Noise

The popular APA [89] can be viewed in the time-varying set-theoretic framework [129] with the linear varieties $V_k := \arg\min_{\boldsymbol{h}\in\mathcal{H}} \|\boldsymbol{e}_k(\boldsymbol{h})\|^2$ $(\forall k \in \mathbb{N})$. The APA generates a sequence of filtering vectors $(\boldsymbol{h}_k)_{k\in\mathbb{N}} \subset \mathcal{H}(:= \mathbb{R}^{2N})$ by (see [129])

$$\boldsymbol{h}_{k+1} = \boldsymbol{h}_k + \lambda_k \left(P_{V_k}(\boldsymbol{h}_k) - \boldsymbol{h}_k\right), \ \forall k \in \mathbb{N}, \tag{5.3.3}$$

where $\lambda_k \in (0, 2)$. In particular, for $r = 1$, (5.3.3) is nothing but the Normalized Least Mean Square (NLMS) algorithm [84], where r is the dimension of affine projection (see Sec. 5.2.1 for the definitions of $\boldsymbol{U}_k \in \mathbb{R}^{2N\times r}$ and $\boldsymbol{d}_k \in \mathbb{R}^r$). A simple comparison of V_k with $C_k(\rho)$ in (5.3.1) implies $V_k = C_k(\delta_k)$, where $\delta_k := \min_{\boldsymbol{h}\in\mathcal{H}} \|\boldsymbol{e}_k(\boldsymbol{h})\|^2$. Note here that we most likely have $\delta_k \approx 0$, since we often have $2N \gg r$ due to long impulse responses of acoustic paths.

The remaining of this section is devoted to the *robustness issue against noise* by highlighting the membership $\boldsymbol{h}^* \in C_k(\rho)$, which ensures the monotone approximation property (for stability); i.e., $\|\boldsymbol{h}_{k+1} - \boldsymbol{h}^*\| \le \|\boldsymbol{h}_k - \boldsymbol{h}^*\|$. Noting that $\boldsymbol{h}^* \in C_k(\rho) \Leftrightarrow \|\boldsymbol{e}_k(\boldsymbol{h}^*)\|^2 = \|\boldsymbol{n}_k\|^2 \le \rho$, we see that ρ governs the reliability on the membership $\boldsymbol{h}^* \in C_k(\rho)$ by $\int_0^\rho f_r(\xi)d\xi$, where $f_r(\xi)$ is the probability density function (pdf) of the random variable $\xi := \|\boldsymbol{n}_k\|^2$ ($f_r(\xi)$ is given in [129, Eq. (9)]). Under the assumption that the noise process is a zero mean i.i.d. Gaussian random variables $\mathcal{N}(0, \sigma^2)$, the random variable ξ follows a χ^2 distribution (of order r), where σ^2 is the variance of noise. The pdf $f_r(\xi)$ is strictly monotone decreasing over $\xi \ge 0$ for $r = 1, 2$, whereas for $r \ge 3$, it has its unique peak at $\xi = (r-2)\sigma^2$ and $f_r(0) = \lim_{\xi\to\infty} f_r(\xi) = 0$. Recall that we most likely have $\delta_k \approx 0$. The above facts imply that, for $r \ge 3$, Prob$\{\boldsymbol{h}^* \in C_k(\delta_k)(= V_k)\}$ is expected to be small, which causes serious sensitivity of the APA to noise for $r \ge 3$ (see Sec. 5.4). For $r = 1, 2$, on the other hand, Prob$\{\boldsymbol{h}^* \in C_k(\delta_k)\}$ is expected

to be relatively large, which suggests robustness of the APA ($r = 1, 2$) against noise (This agrees with the H^∞ optimality [54] of the NLMS, a special case of the APA for $r = 1$). By designing appropriate ρ based on statistics of noise process (see [129, Example 1]), the proposed schemes can fairly raise $\mathrm{Prob}\{\boldsymbol{h}^* \in C_k(\rho)\}$; NOTE: $\mathrm{Prob}\{\boldsymbol{h}^* \in H_k^-(\boldsymbol{h}_k)\} \geq \mathrm{Prob}\{\boldsymbol{h}^* \in C_k(\rho)\}$ because $H_k^-(\boldsymbol{h}_k) \supset C_k(\rho)$. This brings about the noise robustness of POWER I/II, which is presented below.

5.3.3 Novel POWER-Based Stereo Echo Canceler

Given $q \in \mathbb{N}^*$, define the *control sequence* consisting of the q latest time indices as $\mathcal{I}_k^{(\mathrm{c})} := \{k, k-1, \ldots, k-q+1\} \subset \mathbb{N}$. Let $Q_{\mathrm{cyc}} \in \mathbb{N}^*$ denote the cycle period of preprocessing [66, 112], i.e., every $Q_{\mathrm{cyc}}/2$ iterations, the state of inputs is switched. Then, $k - Q_{\mathrm{cyc}}/2$ ($\forall k > Q_{\mathrm{cyc}}/2$) always belongs to the state opposite to k. To utilize data from both states of inputs, we use $\mathcal{I}_k^{(\mathrm{c})} \cup \mathcal{I}_k^{(\mathrm{p})}$ as in [138], where

$$\mathcal{I}_k^{(\mathrm{p})} := \begin{cases} \emptyset, & 0 \leq k \leq Q_{\mathrm{cyc}}/2, \\ \mathcal{I}_{k-Q_{\mathrm{cyc}}/2}^{(\mathrm{c})}, & k > Q_{\mathrm{cyc}}/2. \end{cases}$$

Note that the definitions of $\mathcal{I}_k^{(\mathrm{c})}$ and $\mathcal{I}_k^{(\mathrm{p})}$ can be generalized to any index sets consisting of arbitrary indices chosen from the current and previous states, respectively (see [136]). For simplicity, however, we focus on the above specific definition in the following.

Although given already in the previous chapter, the most important definition is now restated; three points expression of projection onto the intersection of two closed halfspaces. For convenience, let us define, $\forall \boldsymbol{a}, \boldsymbol{b} \in \mathcal{H}$,

$$\Pi^-(\boldsymbol{a}, \boldsymbol{b}) := \{\boldsymbol{y} \in \mathcal{H} : \langle \boldsymbol{a} - \boldsymbol{b}, \boldsymbol{y} - \boldsymbol{b} \rangle \leq 0\} \subset \mathcal{H}, \tag{5.3.4}$$

where $\Pi^-(\boldsymbol{a}, \boldsymbol{b})$ is a closed half-space if $\boldsymbol{a} \neq \boldsymbol{b}$. Then, for a given ordered triplet $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{b}) \in \mathcal{H}^3$ s.t. $\Pi^-(\boldsymbol{s}, \boldsymbol{a}) \cap \Pi^-(\boldsymbol{s}, \boldsymbol{b}) \neq \emptyset$, we define

$$\mathcal{P}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{b}) := P_{\Pi^-(\boldsymbol{s},\boldsymbol{a}) \cap \Pi^-(\boldsymbol{s},\boldsymbol{b})}(\boldsymbol{s}),$$

namely $\mathcal{P}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{b})$ denotes the projection of $\boldsymbol{s}$ onto $\Pi^-(\boldsymbol{s}, \boldsymbol{a}) \cap \Pi^-(\boldsymbol{s}, \boldsymbol{b})$ in $\mathcal{H}$.

We propose a new class of SAEC schemes that utilize $\mathcal{P}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{b})$ (see Proposition 4.3.3) to realize better weights in the method proposed in [138]. Two schemes in the proposed class are presented below, where two families of closed half-spaces, $\{H_\iota^-(\boldsymbol{h}_k)\}_{\iota\in\mathcal{I}_k^{(\mathrm{c})}}$ and $\{H_\iota^-(\boldsymbol{h}_k)\}_{\iota\in\mathcal{I}_k^{(\mathrm{p})}}$, are used in different ways.

A. POWER Type I

A scheme that exploits the POWER technique in a combinatorial manner is presented below [see Fig. 5.3.5-(a)]. Define $\mathcal{I}_k^{(1)} := \{(k-i+1, k-Q_{\mathrm{cyc}}/2-i+1) : i = 1, 2, \cdots, q\} \subset \{(\iota_1, \iota_2) : \ \iota_1 \in \mathcal{I}_k^{(\mathrm{c})}, \ \iota_2 \in \mathcal{I}_k^{(\mathrm{p})}\}$. Also define inductively the control sequences used in each stage as $\mathcal{I}_k^{(m)} \subset \{(\iota_1, \iota_2) : \iota_1, \iota_2 \in \mathcal{I}_k^{(m-1)}, \iota_1 \neq \iota_2\}$, $\forall m \in \{2, 3, \cdots, M\}$, $\forall k \in \mathbb{N}$, satisfying $1 = |\mathcal{I}_k^{(M)}| \lneq |\mathcal{I}_k^{(M-1)}| \leq \cdots \leq |\mathcal{I}_k^{(2)}| \leq |\mathcal{I}_k^{(1)}| = q$. The scheme is given as follows.

Scheme 5.3.1 (POWER Type I). *Suppose that a sequence of closed convex sets $(C_k(\rho))_{k\in\mathbb{N}} \subset \mathcal{H}$ is defined as in* (5.3.1). *Let $\boldsymbol{h}_0 \in \mathcal{H}$ be an arbitrarily chosen initial vector. Then, define a sequence of filtering vectors $(\boldsymbol{h}_k)_{k\in\mathbb{N}} \subset \mathcal{H}$ through multiple stages.*

♠ *0th Stage: Projection onto 2q Half-Spaces*

$$\boldsymbol{h}_{k,\iota}^{(0)} := P_{H_\iota^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k), \quad \forall k \in \mathbb{N}, \ \forall \iota \in \mathcal{I}_k^{(\mathrm{c})} \cup \mathcal{I}_k^{(\mathrm{p})}, \tag{5.3.5}$$

where $P_{H_\iota^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k)$ is computed by (5.3.2).

♠ *1st $\sim$ Mth Stage: Find Good Direction*

for $m := 1$ **to** M **do**

$$\boldsymbol{h}_{k,\iota}^{(m)} := \begin{cases} \boldsymbol{h}_k, & \text{if } \eta_{k,\iota}^{(m)} = -\sqrt{\xi_{k,\iota}^{(m)}\zeta_{k,\iota}^{(m)}} \neq 0, \\ \mathcal{P}\left(\boldsymbol{h}_k, \boldsymbol{h}_{k,\iota_1}^{(m-1)}, \boldsymbol{h}_{k,\iota_2}^{(m-1)}\right), & \text{otherwise,} \end{cases}$$
$$\forall k \in \mathbb{N}, \quad \forall \iota = (\iota_1, \iota_2) \in \mathcal{I}_k^{(m)}, \tag{5.3.6}$$

where $\eta_{k,\iota}^{(m)} := \left\langle \boldsymbol{h}_{k,\iota_1}^{(m-1)} - \boldsymbol{h}_k, \boldsymbol{h}_{k,\iota_2}^{(m-1)} - \boldsymbol{h}_k \right\rangle$, $\xi_{k,\iota}^{(m)} := \left\| \boldsymbol{h}_{k,\iota_1}^{(m-1)} - \boldsymbol{h}_k \right\|^2$ *and* $\zeta_{k,\iota}^{(m)} := \left\| \boldsymbol{h}_{k,\iota_2}^{(m-1)} - \boldsymbol{h}_k \right\|^2$.

end;

♠ *Final Stage: Update to Good Direction*

$$\boldsymbol{h}_{k+1} := \boldsymbol{h}_k + \lambda_k(\boldsymbol{h}_{k,\iota}^{(M)} - \boldsymbol{h}_k),\ \forall k \in \mathbb{N}, \tag{5.3.7}$$

where $\lambda_k \in [0, 2]$ *is the step size.*

Through the multiple stages, the direction of update is improved thanks to the operator $\mathcal{P}(\cdot,\cdot,\cdot)$ (see Remark 4.3.12 for details).

B. POWER Type II

A simple and efficient scheme that exploits the POWER technique just once is given as follows [see Fig. 5.3.5-(b)].

Scheme 5.3.2 (POWER Type II). *Suppose that a sequence of closed convex sets* $(C_\iota(\rho))_{\iota\in\mathcal{I}} \subset \mathcal{H}$ *is defined as in* (5.3.1), *where* $\mathcal{I} := \bigcup_{k\in\mathbb{N}} \left(\mathcal{I}_k^{(\mathrm{c})} \cup \mathcal{I}_k^{(\mathrm{p})}\right)$. *Let* $\boldsymbol{h}_0 \in \mathcal{H}$ *be an arbitrarily chosen initial vector. Then, define a sequence of filtering vectors* $(\boldsymbol{h}_k)_{k\in\mathbb{N}} \subset \mathcal{H}$ *through the following two stages.*

♠ *1st Stage: Uniformly Averaged Directions*

$$\boldsymbol{h}_k^{(\mathrm{g})} := \begin{cases} \boldsymbol{h}_k + \mathcal{M}_k^{(\mathrm{g})} \left(\sum_{\iota\in\mathcal{I}_k^{(\mathrm{g})}} w_k^{(\mathrm{g})} P_{H_\iota^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right), & \text{if } \mathcal{I}_k^{(\mathrm{g})} \neq \emptyset, \\ \boldsymbol{h}_k, & \text{otherwise}, \end{cases}$$
$$\forall k \in \mathbb{N},\ \forall \mathrm{g} \in \{\mathrm{c}, \mathrm{p}\}, \tag{5.3.8}$$

where $w_k^{(\mathrm{g})} := 1/|\mathcal{I}_k^{(\mathrm{g})}| = 1/q$ $(\forall \iota \in \mathcal{I}_k^{(\mathrm{g})})$ *and*

$$\mathcal{M}_k^{(\mathrm{g})} := \begin{cases} \dfrac{\sum_{\iota\in\mathcal{I}_k^{(\mathrm{g})}} w_k^{(\mathrm{g})} \left\| P_{H_\iota^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\|^2}{\left\| \sum_{\iota\in\mathcal{I}_k^{(\mathrm{g})}} w_k^{(\mathrm{g})} P_{H_\iota^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\|^2}, & \text{if } \boldsymbol{h}_k \notin \bigcap_{\iota\in\mathcal{I}_k^{(\mathrm{g})}} H_\iota^-(\boldsymbol{h}_k), \\ 1, & \text{otherwise}. \end{cases}$$

♠ *2nd Stage: Reasonably Averaged Direction by POWER*

$$\boldsymbol{h}_{k+1} := \begin{cases} \boldsymbol{h}_k, & \text{if } \eta_k = -\sqrt{\xi_k \zeta_k} \neq 0, \\ \boldsymbol{h}_k + \lambda_k\{\mathcal{P}(\boldsymbol{h}_k, \boldsymbol{h}_k^{(\mathrm{c})}, \boldsymbol{h}_k^{(\mathrm{p})}) - \boldsymbol{h}_k\}, & \text{otherwise}, \end{cases}$$

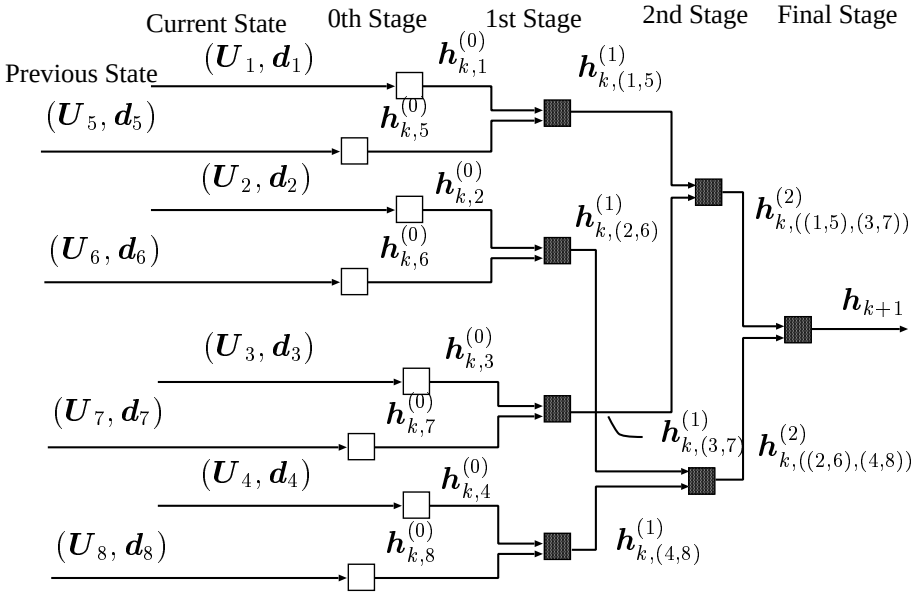
$\forall k \in \mathbb{N}$, *where* $\lambda_k \in [0,2]$ *is the step size,* $\eta_k := \langle \boldsymbol{h}_k^{(\mathrm{c})} - \boldsymbol{h}_k, \boldsymbol{h}_k^{(\mathrm{p})} - \boldsymbol{h}_k \rangle$, $\xi_k := \|\boldsymbol{h}_k^{(\mathrm{c})} - \boldsymbol{h}_k\|^2$ *and* $\zeta_k := \|\boldsymbol{h}_k^{(\mathrm{p})} - \boldsymbol{h}_k\|^2$.

In the 1st stage, for saving the computational complexity, the uniform averages $\boldsymbol{h}_k^{(\mathrm{c})}$ and $\boldsymbol{h}_k^{(\mathrm{p})}$ are computed for two groups corresponding to $\mathcal{I}_k^{(\mathrm{c})}$ and $\mathcal{I}_k^{(\mathrm{p})}$. In the 2nd stage, the POWER technique is exploited to find a good direction of update based on three kinds of information; $\boldsymbol{h}_k$, $\boldsymbol{h}_k^{(\mathrm{c})}$ and $\boldsymbol{h}_k^{(\mathrm{p})}$ (see Sec. 4.3 for details).

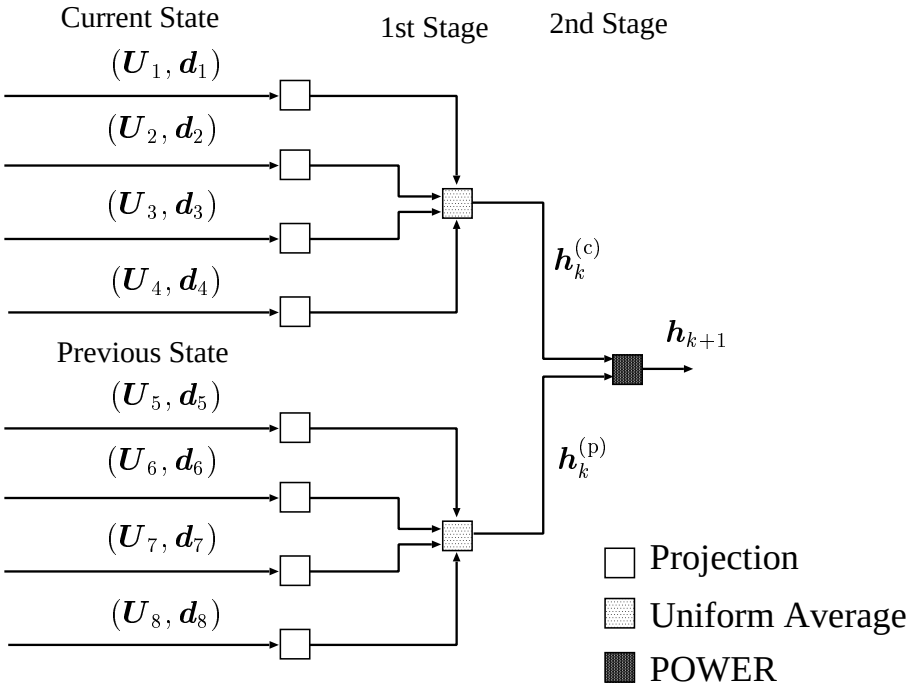
Remark 5.3.3 (On Schemes 5.3.1 and 5.3.2)**.**

(a) *Simple system models to implement the proposed schemes with* $q = 4$ *are shown in Fig.* 5.3.5*. The structure of POWER I is named binary-tree-like construction with its number of stages* $M = \lceil \log_2 q \rceil + 1$*; in this case,* $M = 3$ *(see* [135, 137]*). We see that POWER II is computationally more efficient than POWER I, since it utilizes the POWER technique just once. The projections* $\{P_{H_\iota^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k)\}_{\iota \in \mathcal{I}_k^{(\mathrm{c})} \cup \mathcal{I}_k^{(\mathrm{p})}}$, $\forall k \in \mathbb{N}$, *in* (5.3.5) *and* (5.3.8) *are, respectively, computed simultaneously with* $2q$ *concurrent processors. This implies that the proposed schemes are inherently suitable for implementation with concurrent processors. With such processors, the number of multiplications imposed on each processor is* $(3M+2r+1)N+21M+r$ $(M = \lceil \log_2 q \rceil + 1)$ *for POWER I and* $(2r+6)N+r$ *for POWER II for* $q \geq 2$*; for* $q = 1$*, it is reduced to* $(2r+4)N+r$ *for POWER I/II . In other words, the complexity is kept* $O(N)$*, which is a desired property especially for real-time implementation.*

(b) *Discussions about convergence of the adaptive-PSP algorithm are found in the Adaptive Projected Subgradient Method* [124, 126]*, a more general framework. A geometric interpretation illustrated in Fig.* 5.3.6 *will be rather helpful from a standpoint of application. For simplicity, we set* $q = 2$ *and* $\lambda_k = 1$*. In the figure, the estimandum* $\boldsymbol{h}^*$ *(see Sec.* 5.2.1*) is assumed to belong to the dotted area; i.e.,* $\boldsymbol{h}^* \in \bigcap_{\iota \in \mathcal{I}_k^{(\mathrm{c})} \cup \mathcal{I}_k^{(\mathrm{p})}} H_\iota^-(\boldsymbol{h}_k)$*. This assumption holds if* $C_k(\rho)$ *is defined appropriately (For details, see* [129]*). We see that the schemes realize good directions of update. For visual clarity, the half-spaces* $\Pi^-(\boldsymbol{h}_k, \boldsymbol{h}^{(1)}_{k,(k,k-Q_{\mathrm{cyc}}/2)})$ *and* $\Pi^-(\boldsymbol{h}_k, \boldsymbol{h}^{(1)}_{k,(k-1,k-Q_{\mathrm{cyc}}/2-1)})$ *are omitted. It is not hard to see that* $\boldsymbol{h}_{k+1} = \mathcal{P}(\boldsymbol{h}_k, \boldsymbol{h}^{(1)}_{k,(k,k-Q_{\mathrm{cyc}}/2)}, \boldsymbol{h}^{(1)}_{k,(k-1,k-Q_{\mathrm{cyc}}/2-1)}) = \boldsymbol{h}^{(1)}_{k,(k,k-Q_{\mathrm{cyc}}/2)}$ *in this simple example.*



(a)



(b)

Figure 5.3.5: Simple system models with eight parallel processors ($q = 4$) to implement (a) POWER I and (b) POWER II. For notational simplicity, define the current control sequence $\mathcal{I}_k^{(\mathrm{c})} = \{1, 2, 3, 4\}$ and the previous control sequence $\mathcal{I}_k^{(\mathrm{p})} = \{5, 6, 7, 8\}$. This type of design of control sequences for POWER I is called binary-tree-like construction. It is seen that POWER II is more efficient in computation than POWER I.

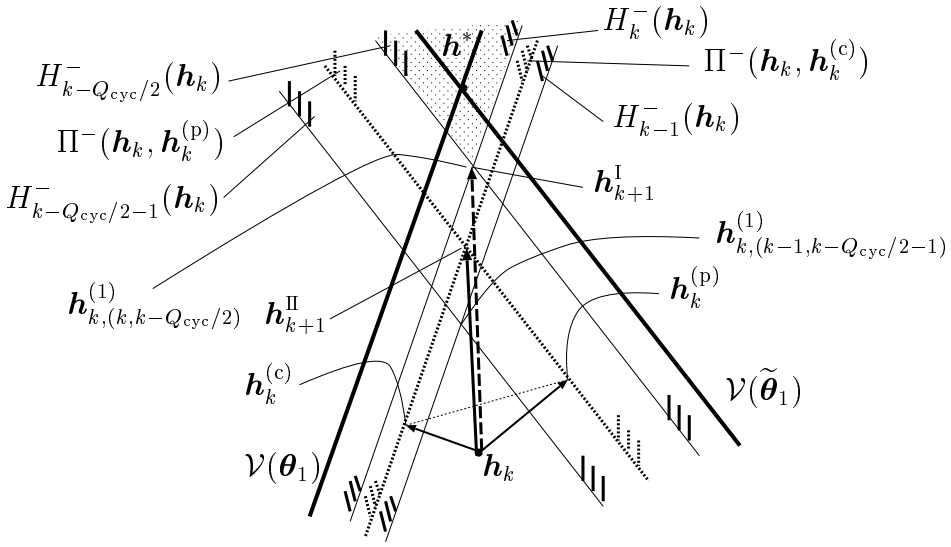


Figure 5.3.6: A geometric interpretation of the proposed schemes. POWER I: $\boldsymbol{h}_{k+1}^{\mathrm{I}}$, POWER II: $\boldsymbol{h}_{k+1}^{\mathrm{II}}$. The control sequences are defined as $\mathcal{I}_k^{(\mathrm{c})} = \{k, k-1\}$ and $\mathcal{I}_k^{(\mathrm{p})} = \{k - Q_{\mathrm{cyc}}/2, k - Q_{\mathrm{cyc}}/2 - 1\}$. The dotted area shows $\bigcap_{\iota \in \mathcal{I}_k^{(\mathrm{c})} \cup \mathcal{I}_k^{(\mathrm{p})}} H_\iota^-(\boldsymbol{h}_k)$.

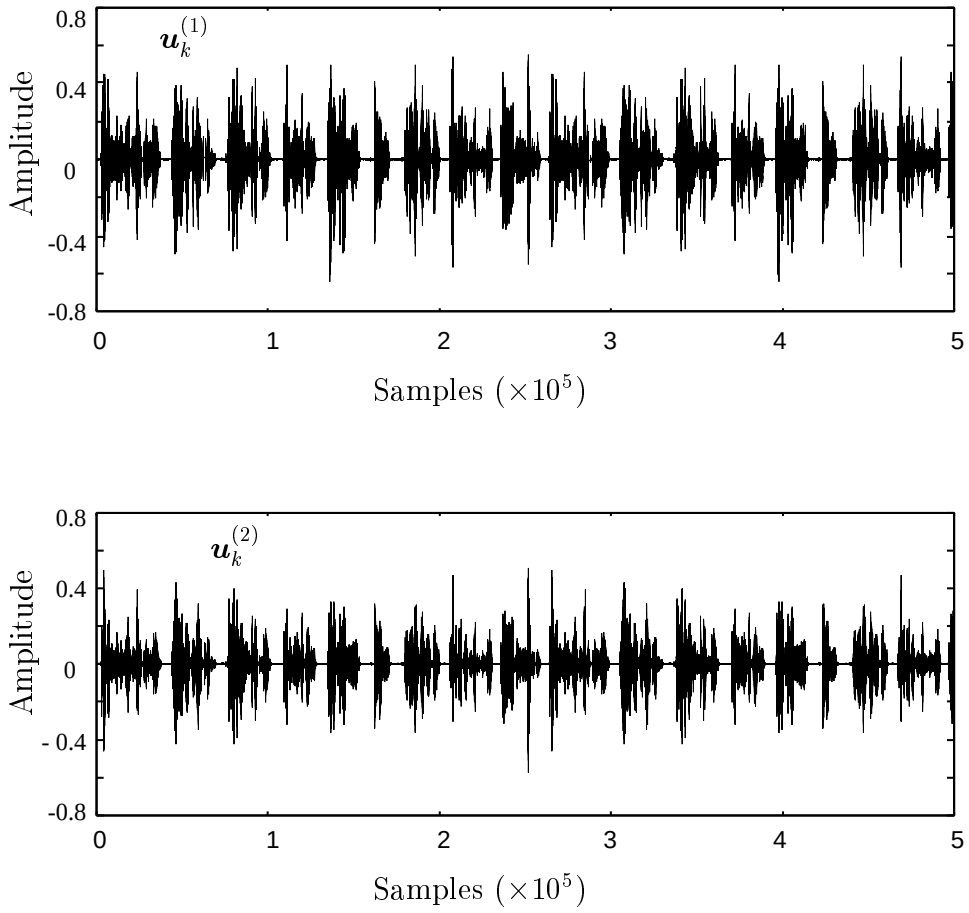


Figure 5.4.7: The input signals $\left(u_k^{(1)}\right)_{k\in\mathbb{N}}$ and $\left(u_k^{(2)}\right)_{k\in\mathbb{N}}$. The signals are generated from a speech signal, sampled at 8 kHz, of an English-native male.

(c) *The proposed schemes realize strategic weight designs for the method in* [138] *in the sense that the schemes give optimal weights, based on a certain max-min criterion, in each stage (see Sec.* 4.3.3*).*

5.4 Numerical Examples

This section presents numerical examples of the proposed schemes, the UW-PSP, APA [58, 89], NLMS [84] and fast RLS (FRLS) [40, 97] algorithms. All the methods are performed with a common preprocessing technique in [66, 112] that periodically delays input signals in the 1st channel with the cycle of preprocessing $Q_{\mathrm{cyc}} = 2000$. The tests are conducted, for estimating $\boldsymbol{h}^* \in \mathcal{H} := \mathbb{R}^{2000}\,(N = L_{\mathrm{tran}} = 1000)$,

under the noise situation of SNR $:= 10\log_{10}(E\{z_k^2\}/E\{n_k^2\}) = 25$ dB, where $z_k := \langle \boldsymbol{u}_k, \boldsymbol{h}^* \rangle$ and $E\{\cdot\}$ denote *pure echo* (i.e., echo without noise) and *expectation*, respectively. We utilize a recorded speech signal of an English native male[8] shown in Fig. 5.4.7, for $(s_k)_{k\in\mathbb{N}}$, which was sampled at 8 kHz. For numerical stability against the poorly excited inputs observed in Fig. 5.4.7, all the algorithms are regularized. The APA is regularized by following the way in [43] with exactly the same parameter as in [129]. The NLMS is regularized by following the way in [56, Eq. (9.144)] with the regularization parameter $\delta = 1.0 \times 10^{-1}$ for better performance. Because the original RLS algorithm is computationally intensive for acoustic echo cancellation applications [11, p.77], a simplified implementation of the regularized RLS [40] is employed with $\xi_k^2 = 20\sigma_u^2$ and $\phi_k = 1$ ($\forall k \in \mathbb{N}$), where σ_u^2 is the variance of $(u_k)_{k\in\mathbb{N}}$. For the proposed schemes and the UW-PSP, the projection in (5.3.2) is regularized as

$$P^{(\delta)}_{H_\iota^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k) := \begin{cases} \boldsymbol{h}_k - \dfrac{g_\iota(\boldsymbol{h}_k)}{\|\nabla g_\iota(\boldsymbol{h}_k)\|^2 + \delta}\nabla g_\iota(\boldsymbol{h}_k), & \text{if } \boldsymbol{h}_k \notin H_\iota^-(\boldsymbol{h}_k), \\ \boldsymbol{h}_k, & \text{otherwise.} \end{cases}$$

where δ is set to 1.0×10^{-6}.

In addition to the regularization for numerical stability against poor excitation, while the signal power is less than a common threshold, we stop the update for all algorithms throughout the simulations (This is the reason of the observable flat intervals in the figures).

To measure the achievement level for echo path identification as well as echo cancellation, the following criteria are adopted:

$$\text{System Mismatch}(k) := 10\log_{10}\frac{\|\boldsymbol{h}^* - \boldsymbol{h}_k\|^2}{\|\boldsymbol{h}^*\|^2}, \forall k \in \mathbb{N},$$
$$\text{ERLE}(k) := 10\log_{10}\frac{\sum_{i=1}^k z_i^2}{\sum_{i=1}^k (z_i - \langle \boldsymbol{u}_i, \boldsymbol{h}_i \rangle)^2}, \forall k \in \mathbb{N}.$$

Simulations are conducted under several conditions.

[8] The speech sample is provided by "Special Research Project of the Typological Investigation into Languages & Cultures of the East & West (LACE)" in University of Tsukuba; the website is found in http://www.modern.tsukuba.ac.jp/~lace/index.html.

5.4.1 Proposed Schemes versus UW-PSP with Different q

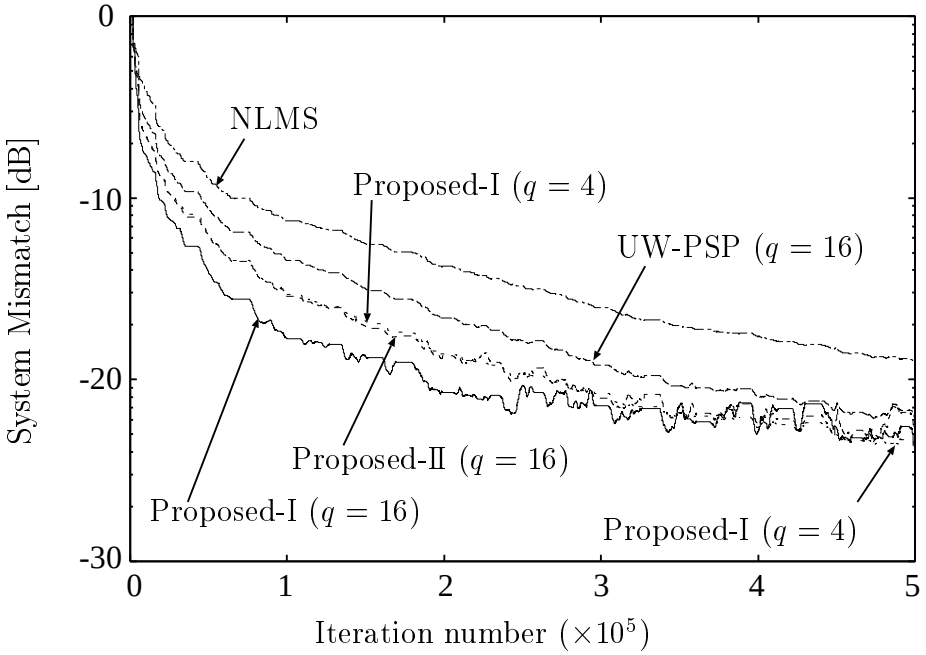
First, we examine the performance of the proposed schemes and the UW-PSP with ($|\mathcal{I}_k^{(\mathrm{c})}| = |\mathcal{I}_k^{(\mathrm{p})}| =$)$q = 4, 16$ in Fig. 5.4.8. For a comparison, the curve of NLMS with the step size $\lambda_k = 0.2$ is drawn, which is a special case of POWER I for $q = 1$, $r = 1$, $\rho = 0$, $\lambda_k = 0.4$, $\mathcal{I}_k^{(0)} = \mathcal{I}_k^{(\mathrm{c})} = \{k\}$, $\mathcal{I}_k^{(1)} = \{(k,k)\}$ ($M = 1$) and $\mathcal{I}_k^{(\mathrm{p})} = \emptyset$. For the proposed schemes, we set $\lambda_k = 0.4$ ($\forall k \in \mathbb{N}$), $r = 1$ and $\rho = \max\{(r-2)\sigma^2, 0\} = 0$; see Sec. 5.3.2 and [129]. The control sequences for POWER I are designed in the same manner as shown in Fig. 5.3.5.

For POWER II and the UW-PSP, the curves of $q = 4$ are omitted for visual clarity, since the difference between $q = 4$ and $q = 16$ is not significant. Referring to Fig. 5.4.8, we see that the increase of q for POWER I significantly improves the convergence speed without serious degradation in steady-state performance in both criteria. We also see that POWER I for $q = 4$ exhibits faster convergence than the UW-PSP for $q = 16$. The above observation suggests that *weight design is the key to attain better performance by increasing q.*

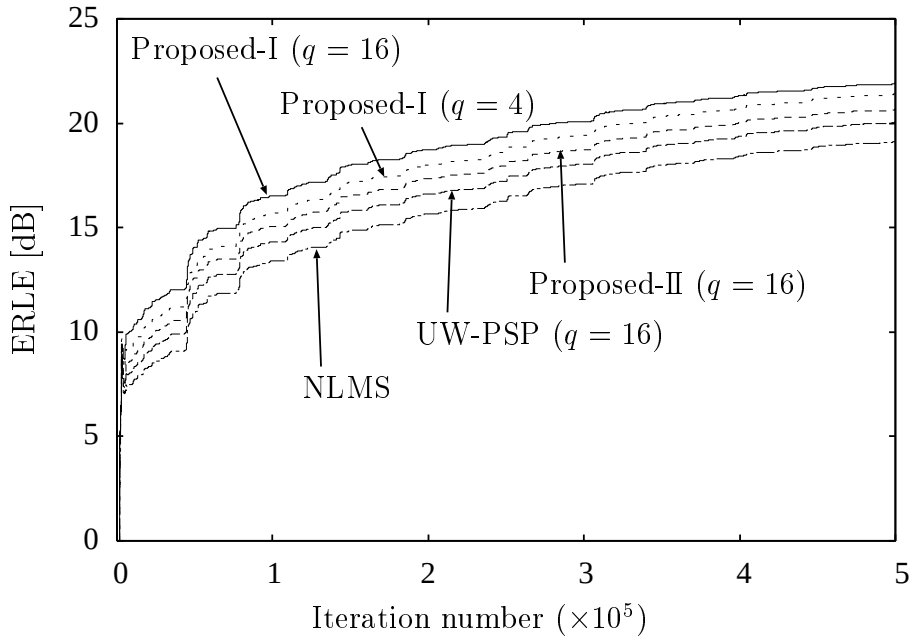
5.4.2 APA-Based Method with Different r

Next, we examine the performance of the APA for $r = 2, 4, 8, 16$ in Fig. 5.4.9, where r is the dimension of affine projection (see Sec. 5.3.2). The APA based method using data from one state of inputs at each iteration is referred to as "APA-I". The step size for $r = 2$ is set to $\lambda_k = 0.2$ for better performance. For $r = 4, 8, 16$, two step sizes are employed; one is fixed to $\lambda_k = 0.2$ (the same step size as $r = 2$), for all r, and the other is individually tuned, for each r, so that the steady-state performance in system mismatch is almost the same as $r = 2$ with $\lambda_k = 0.2$.

Referring to Fig. 5.4.9, the increase of r for the APA-I raises the initial convergence speed at the expense of serious degradation in the steady-state performance in system mismatch, which causes gain loss in ERLE especially for $r = 8, 16$. For the tuned step size, on the other hand, no distinct difference is observed among all r in system mismatch, since, for large r, the small step size for good steady-state performance decreases the initial convergence speed. Comparing Fig. 5.4.9 with Fig. 5.4.8, it is seen that POWER I successfully alleviates the trade-off problem between convergence speed and steady-state performance.

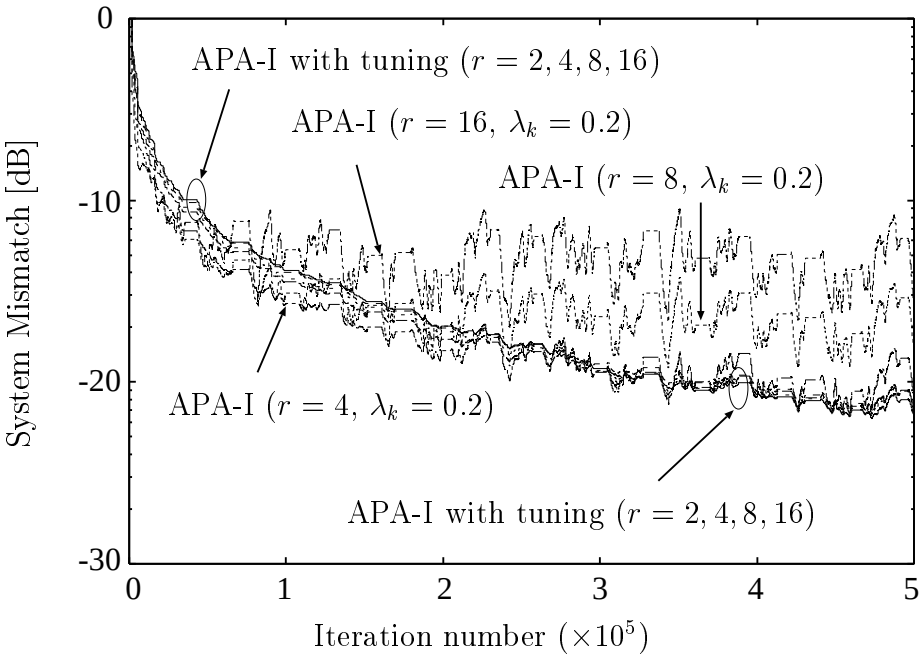


(a)

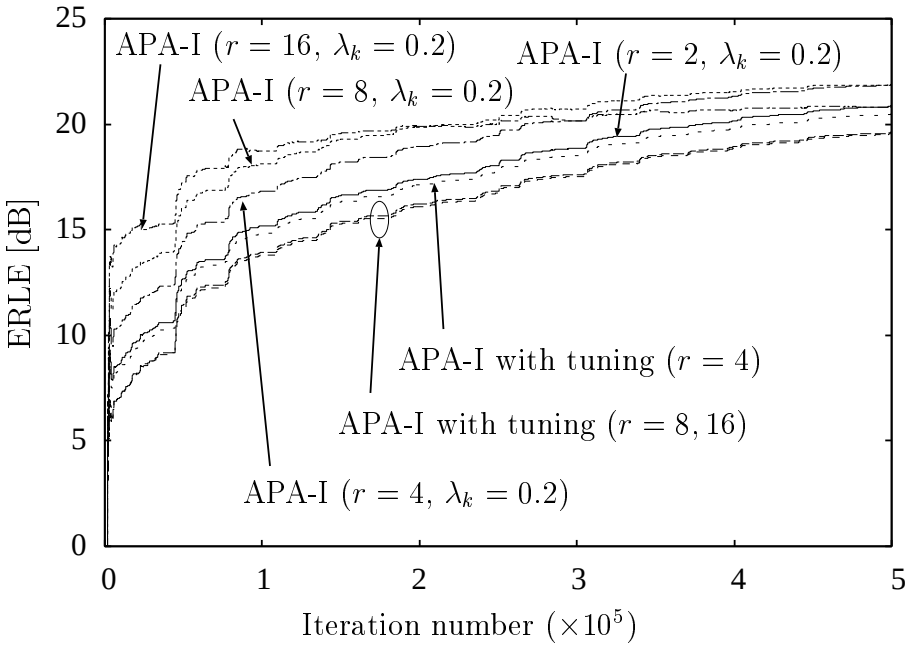


(b)

Figure 5.4.8: Proposed Schemes versus UW-PSP for $r = 1$ and $\lambda_k = 0.4$ under SNR = 25 dB. For a comparison, the performance of NLMS (a special case of the proposed method for $q = 1$) is shown for $\lambda_k = 0.2$.



(a)



(b)

Figure 5.4.9: APA-I for $r = 2, 4, 8, 16$ under SNR = 25 dB. For $r = 2$, we set $\lambda_k = 0.2$. For $r = 4, 8, 16$, we use the same step size $\lambda_k = 0.2$ and individually tuned one; $\lambda_k = 0.1$ for $r = 4$, $\lambda_k = 0.04$ for $r = 8$ and $\lambda_k = 0.022$ for $r = 16$.

It should be remarked that these results do not contradict the results in other publications as mentioned below. Under high SNR situations, it is reported that the increase of r in the APA raises the speed of convergence, especially for highly colored excited input signals, without severely deteriorating the steady-state performance (see e.g., [15, 93, 95, 113]). Under low SNR situations, on the other hand, it is theoretically verified that the increase of r in the APA decreases the membership probability $\boldsymbol{h}^* \in V_k$ (especially for $r \geq 3$, Prob$\{\boldsymbol{h}^* \in V_k\} \approx 0$) [129, Sec. III], which causes serious noise sensitivity of the APA for $r \geq 3$ (see also Sec. 5.3.2).

5.4.3 Proposed Schemes versus UW-PSP, APA, NLMS and FRLS with Fixed and Time-Varying Echo Paths

The proposed schemes are now compared with the UW-PSP, APA-I, NLMS and FRLS algorithms in Figs. 5.4.10 and 5.4.11. For the proposed schemes and the UW-PSP, the parameters are exactly the same as in Fig. 5.4.8 except that $q = 8$. For the NLMS, the step size is set to 0.2 to attain better steady-state performance. For the APA-I, we set $r = 2$ and $\lambda_k = 0.15$ so that the initial convergence speed is the same as the UW-PSP. For the FRLS, the forgetting factor is set to $\gamma = 1 - \frac{1}{18N}$ for the best performance among our experiments. We remark that the FRLS algorithm exhibits severe sensitivity against the choice of the forgetting factor or the regularization parameter ξ_k^2; e.g., once we tried to employ $\gamma = 1 - \frac{1}{15N}$, the speed of convergence was a little faster but the filter diverged around the iteration number 500000. In this simulation, although the steady-state performance is not the same as the proposed schemes, the parameters are tuned carefully.

Figure 5.4.10 depicts the results under the condition of fixed echo paths. We observe that the proposed schemes attain much faster convergence as well as better steady-state than the NLMS, APA-I and FRLS algorithms. The time for POWER I to achieve the system mismatch level of -20 dB is approximately 25 sec. The time for each algorithms is summarized in Table 5.4.1. POWER I is approximately 45 sec., 25 sec. and 3 sec. faster than the NLMS, the APA-I and the FRLS, respectively. Figure 5.4.11 depicts the results under the condition where the echo paths are changed at the iteration number 1.6×10^5. We see that the proposed schemes exhibit excellent tracking behavior against echo path variation. In Figs. 5.4.10 and 5.4.11, the FRLS exhibits poor ERLE performance due to the

Table 5.4.1: Time needed to achieve the system mismatch level of -20 dB.

method	POWER I	POWER II	UW-PSP	FRLS	APA-I	NLMS
sec.	25	31	43	28	50	75

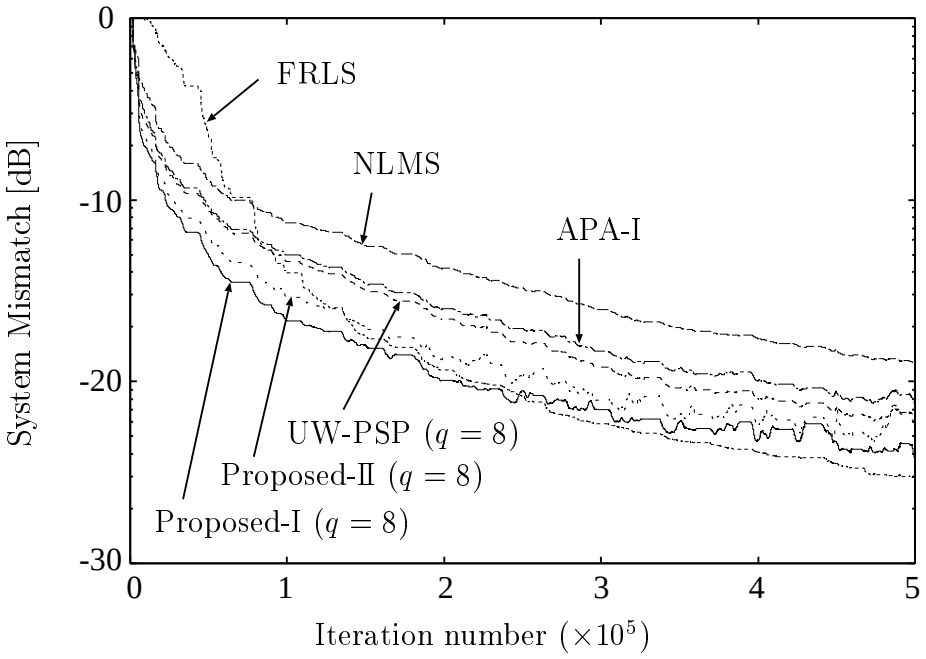
observable instability in system mismatch at the beginning of adaptation. For fairness, we also draw the curves of the FRLS in a different ERLE criterion in which the summations are taken (not from $i = 1$ but) from the moment when its system mismatch becomes less than 0 dB (This new ERLE criterion is referred to as "fair ERLE").

It is reported that the RLS algorithm exhibits, besides its high computational complexity, an instability issue especially for (nonstationary) speech signals, and thus has been discouraged to be used in acoustic echo cancellation [11, p.77]. Also the FRLS algorithms inherit the instability issue, as pointed out in a considerable amount of literature; e.g., [42, p.40], [14, 50, 82, 105]. Moreover, the observable slow initial-convergence of the FRLS stems from the same reason as its tracking inferiority, under nonstationary environments, to the LMS-type algorithms, as remarked, e.g., in [54, 57, 71].

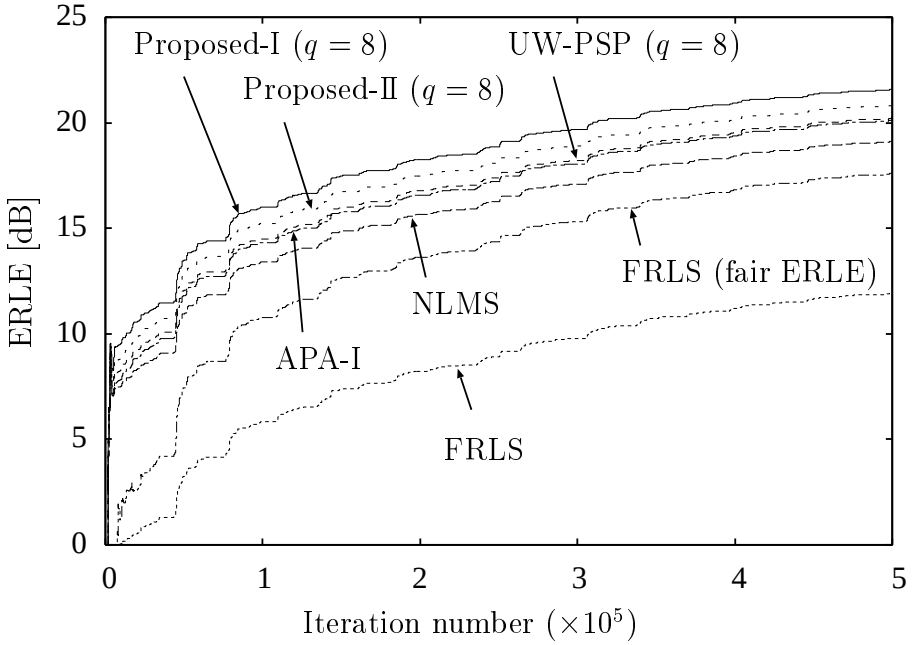
5.4.4 Proposed Schemes versus APA with Simultaneous Use of Data from Two States

Finally, POWER I is compared, in Fig. 5.4.12, with the remaining possibility to resolve the *zig-zag loss* (see Sec. 5.1); i.e., the APA with simultaneous use of data from two states of inputs. Namely, $\forall k \geq Q_{\rm cyc}/2 + r/2$, $\widetilde{\boldsymbol{e}}_k(\boldsymbol{h}) := \widetilde{\boldsymbol{U}}_k^T \boldsymbol{h} - \widetilde{\boldsymbol{d}}_k$ is used to define V_k (see Sec. 5.3.2) instead of $\boldsymbol{e}_k(\boldsymbol{h})$, where $\widetilde{\boldsymbol{U}}_k := [\boldsymbol{u}_k \cdots \boldsymbol{u}_{k-r/2+1} \boldsymbol{u}_{k-Q_{\rm cyc}/2} \cdots \boldsymbol{u}_{k-Q_{\rm cyc}/2-r/2+1}] \in \mathbb{R}^{2N \times r}$ and $\widetilde{\boldsymbol{d}}_k := \widetilde{\boldsymbol{U}}_k^T \boldsymbol{h}^* + \widetilde{\boldsymbol{n}}_k \in \mathbb{R}^r$ with $\widetilde{\boldsymbol{n}}_k := [n_k, \cdots, n_{k-r/2+1}, n_{k-Q_{\rm cyc}/2}, \cdots, n_{k-Q_{\rm cyc}/2-r/2+1}]^T$. This new APA method is referred to as "APA-II". For the proposed scheme, the parameters are the same as in Fig. 5.4.8 (or in Fig. 5.4.10) for $q = 4, 8$. For the APA-II, for fairness, $r = 8, 16$ are employed with the tuned step size $\lambda_k = 0.04, 0.022$, respectively. For a comparison, the curves of APA-I and II with $r = 2$ and $\lambda_k = 0.2$ are also drawn.

In Fig. 5.4.12, we observe that the proposed scheme achieves faster initial convergence and better steady-state performance than the APA-II in both criteria.

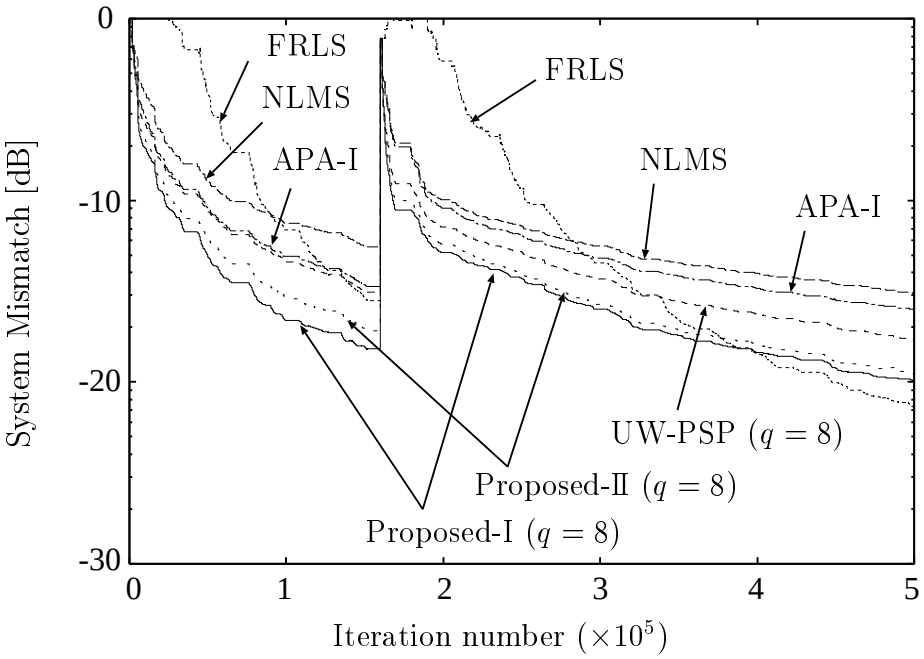


(a)

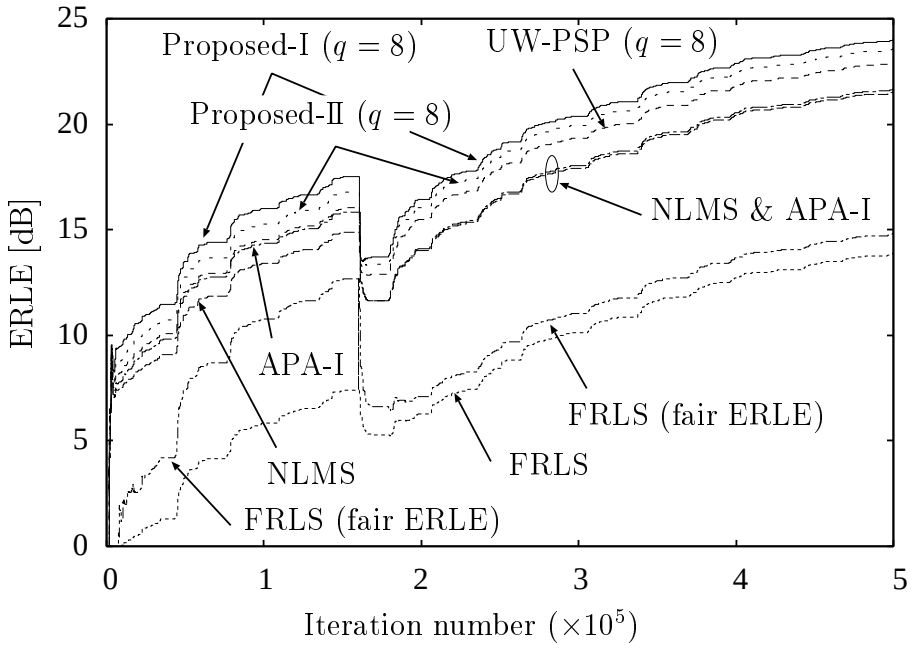


(b)

Figure 5.4.10: Proposed Schemes versus UW-PSP, NLMS, APA-I and FRLS under SNR = 25 dB. For the NLMS, $\lambda_k = 0.2$. For the APA-I, $r = 2$ and $\lambda_k = 0.15$. For the FRLS, $\gamma = 1 - \dfrac{1}{18N}$. For the proposed schemes and the UW-PSP, $r = 1$, $\lambda_k = 0.4$ and $q = 8$.

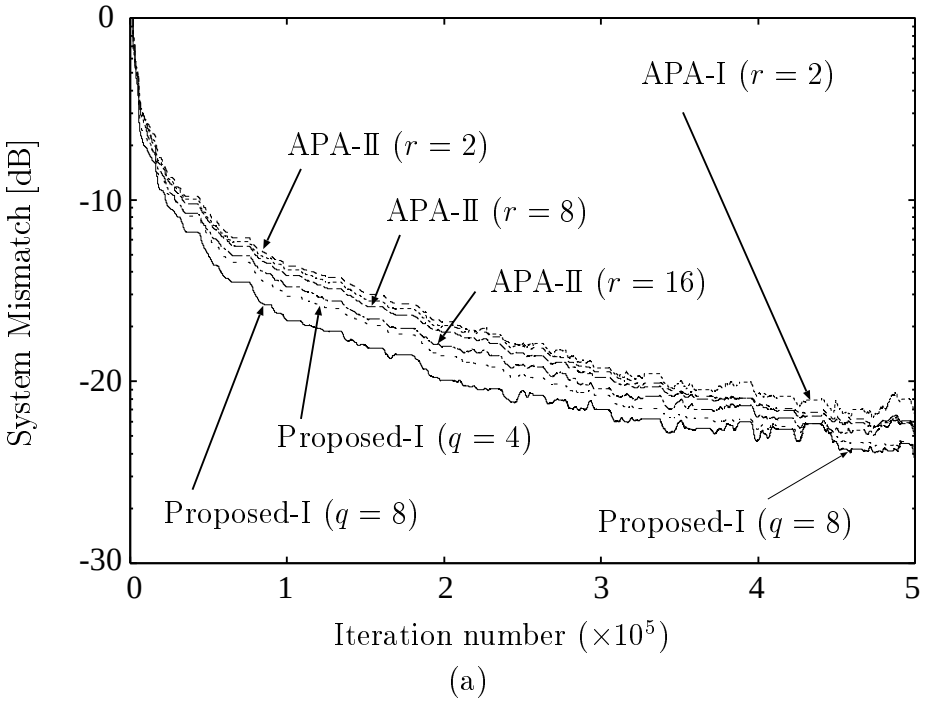


(a)

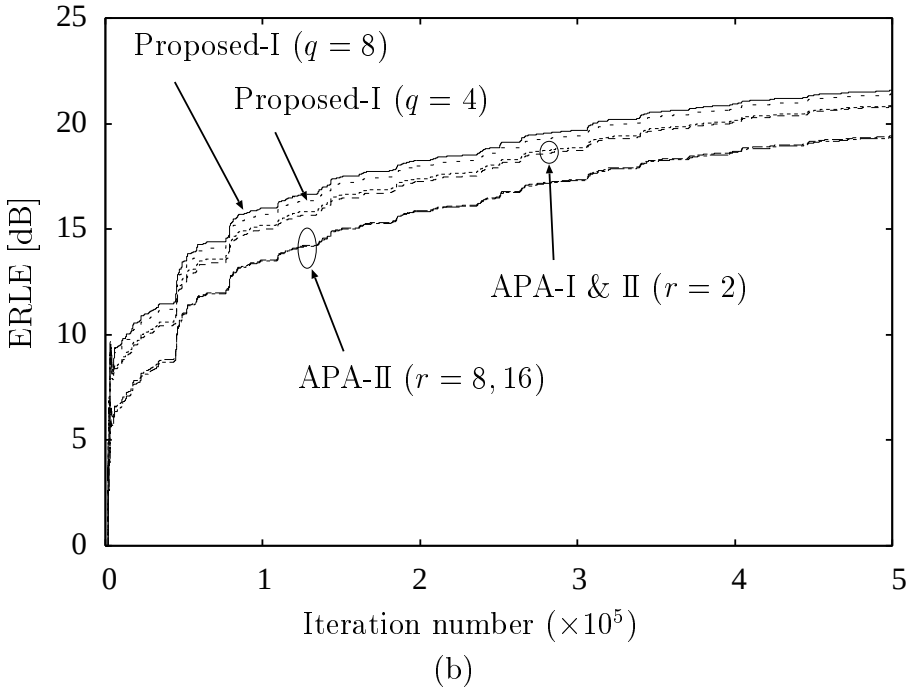


(b)

Figure 5.4.11: Proposed Schemes versus UW-PSP, NLMS, APA-I and FRLS with the echo paths changed at the iteration number 1.6×10^5. The other conditions are the same as in Fig. 5.4.10.



(a)



(b)

Figure 5.4.12: Proposed Schemes ($q = 4, 8$) versus APA-II ($r = 2, 8, 16$) under SNR = 25 dB. For the proposed, we employ the same parameters as in Fig. 5.4.8. For APA-II, $\lambda_k = 0.2, 0.04, 0.022$ for $r = 2, 8, 16$, respectively. For APA-I, $r = 2$ and $\lambda_k = 0.2$.

Moreover, for the APA-II, the increase of r improves the initial convergence speed at the expense of unignorable deterioration in ERLE. On the other hand, for the proposed scheme, the increase of q improves the performance in both criteria, as also shown in Fig. 5.4.8.

5.5 Conclusion

This chapter has presented a class of efficient fast stereophonic acoustic echo canceling schemes based on the POWER weighting technique. The proposed schemes successfully accelerate the convergence with keeping linear complexity and good steady-state performance. Numerical examples have verified the efficacy of the proposed schemes. The results of the extensive simulations suggest the POWER technique is significantly effective especially for the challenging stereophonic echo canceling problem.

Chapter 6

Adaptive Parallel Quadratic-Metric Projection Algorithms

Summary

The goal of this chapter is to show that an appropriate design of metric in the *Adaptive Projected Subgradient Method (APSM)* [Yamada, 2003] leads to significant improvements in adaptive filtering problems. The key is to incorporate a priori (or a posteriori) knowledge on characteristics of an *estimandum*, a system to be estimated, into the metric design. We propose a family of efficient adaptive filtering algorithms based on parallel use of *quadratic-metric projections*, i.e., the best approximation in a closed convex set in the sense of a quadratic norm. We first present two adaptive algorithms in which the metric is constant in time; the algorithms are named Adaptive Parallel Quadratic-metric Projection (APQP) algorithm anD Adaptive Parallel Min-max Quadratic-metric Projection (APMQP) algorithm. APQP is based on the parallel projection, with a quadratic-metric, onto data-dependent closed convex sets containing the estimandum with high reliability. APMQP selectively exploits critical ones among the closed convex sets by a simple min-max criterion. Those algorithms are naturally derived by APSM, thus are endowed with remarkable properties including monotonicity, asymptotic-optimality, and strong convergence.

We then present a more general form of APQP where the metric itself changes in time, thus it is named Adaptive Parallel Variable-metric Projection (APVP) algorithm. The algorithm enjoys a valuable monotone property at each iteration. By employing an efficient metric, the overall computational complexity of the proposed algorithms is kept linear w.r.t the filter length. Numerical examples demonstrate the advantages of the proposed algorithms in the acoustic echo cancellation problem over the conventional algorithms including the Euclidean-metric version of APQP.

6.1 Introduction

Metric projection has been proven to be an effective tool in the robust adaptive signal processing [54, 56, 97, 129]. It is employed in a variety of algorithms such as the Normalized Least Mean Square (NLMS) [1, 84], Affine Projection Algorithm (APA) [58, 81, 89, 93, 103], set-membership NLMS [51], set-membership APA [31, 121], constrained NLMS/APA [27, 28, 67] (in the *embedded* sense [124, 126]), adaptive Parallel Subgradient Projection (adaptive-PSP) [129, 135], embedded-constraint parallel projection [132] algorithms. Although there is great freedom, those algorithms mostly employ the Euclidean metric. *This chapter presents efficient quadratic-metric-projection based algorithms and indicates that a reasonable choice of metric for each application will lead to drastic improvements of performance.*

As a unified guiding principle of the metric-projection-based adaptive algorithms, *Adaptive Projected Subgradient Method (APSM)* has been proposed [124, 126] (see Chapter 2), which minimizes asymptotically a sequence of nonnegative convex objective functions over a closed convex subset of a real Hilbert space[1]. APSM is constructed by combining a metric projection and a relaxed subgradient projection w.r.t. time-varying objective function. The APSM has been proven to be a promising method to derive excellent algorithms in applications to stereophonic acoustic echo cancellation [134, 138] (see Chapter 5) and blind multiple access interference suppression in DS/CDMA systems [20, 132] (see Chapter 3).

[1] Although all the proofs are given for real case in [124, 126], this is *not* a restriction. APSM can also be applied to complex case by defining a bijective mapping between an N-dimensional complex vector space $\mathbb{C}^N$ and $\mathbb{R}^{2N}$ (see, e.g., [21, 140]).

The metric projection operator employed in APSM is extended to a more general nonexpansive mapping called *strongly attracting nonexpansive* [99, 100]. By this extension, we can handle multiple convex constraint sets in asymptotic sense without using the projection onto their intersection. This extended version has been successfully applied to a robust adaptive beamforming problem [101, 102]. For clarity, let us now give some simple examples of reproductions of metric-projection-based algorithms by APSM.

The classical NLMS algorithm is reproduced by defining the objective function at each iteration as the distance to a *hyperplane* containing all possible points that make instantaneous residual error for the latest datum equal to zero. Moreover, the APA algorithm is reproduced by defining the objective function as the distance to a *linear variety* containing all possible points that simultaneously make instantaneous residual errors for a certain number (say r) of data equal to zero. While the linear variety used in APA surely contains the estimandum in *noiseless cases*, it has been analyzed, with the standard noise model of i.i.d. white Gaussian, that the probability that the linear variety for $r \geq 3$ contains the estimandum in *noisy cases* is almost zero and it causes the noise sensitivity of APA [129]. In contrast, the probability that the hyperplane used in NLMS contains the estimandum is relatively high [129], which agrees with the H^∞-optimality of NLMS [54].

Motivated by this analysis, adaptive-PSP, a more efficient adaptive algorithm, has been established [129], where multiple closed convex sets have been introduced instead of a single linear variety (or hyperplane) at each iteration; the closed convex sets are called *stochastic property sets* and designed so as to contain the estimandum with high probability even in noisy cases. The adaptive-PSP algorithm is based on a convex combination (or a weighted average) of the projections onto multiple closed half-spaces, which are outer approximations of the stochastic property sets. The adaptive-PSP algorithm is derived by APSM with the objective function of a weighted sum of the distances to the half-spaces. Usually the algorithms having been derived by APSM employ the Euclidean metric (or norm) defined by the standard inner product: $\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \boldsymbol{a}^T \boldsymbol{b}$ for any $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^N$ $(N \in \mathbb{N}^*)$. One may think that the use of this simple metric is natural because the norm equivalency for finite dimensional cases suggests that the convergence property of a 'given' sequence of vectors is independent of the choice of norm. However, APSM with a different metric (norm) generates a different vector sequence. Therefore,

the choice of metric in APSM governs the convergence property including convergence speed and steady-state performance, which is essential in the adaptive signal processing. The goal of this chapter is to show that an appropriate design of metric in APSM leads to significant improvements in adaptive filtering problems.

In the first part of this chapter (Sec. 6.2), we propose a family of very flexible adaptive algorithms based on quadratic-metric. We firstly present two adaptive algorithms in which the metric is constant in time. The first is based on the parallel projection onto data-dependent closed convex sets, thus it is named *Adaptive Parallel Quadratic-metric Projection (APQP) algorithm* (see Algorithm 6.2.1). APQP includes as its simplest case the *Exponentially weighted Stepsize Projection (ESP) algorithm* [77, 78], developed originally for effective Acoustic Echo Cancellation (AEC) (see Proposition 6.2.4). The second selectively utilizes critical ones among those convex sets by a simple min-max criterion, thus it is named *Adaptive Parallel Min-max Quadratic-metric Projection (APMQP) algorithm* (see Algorithm 6.2.5). These two algorithms are naturally derived by employing quadratic-norms in the APSM, hence those algorithms are equipped with a strongly attracting nonexpansive mapping and are endowed with remarkable properties of APSM; e.g., monotonicity, asymptotic optimality, and strong convergence (see Fact 2.0.3). Although the 'constancy' in the metric design is crucial to ensure the above properties, we secondly present a more general form of APQP where the metric itself changes in time. In other words, the algorithm is based on variable-metric, thus it is named *Adaptive Parallel Variable-metric Projection (APVP) algorithm*, in which the strongly attracting nonexpansive mapping is also variable in time (see Algorithm 6.2.10). APVP includes as its special cases the *Proportionate NLMS/APA (PNLMS/PAPA) algorithms* [9, 10, 29, 33, 41, 48, 64, 86, 120] (see Proposition 6.2.12), which have been proposed originally for a sparse estimandum. The proposed algorithms (APQP/APMQP/APVP) have the valuable monotone property (see Propositions 6.2.9 and 6.2.13). By employing an efficient metric, the overall computational complexity of the proposed algorithms is kept linear w.r.t. the filter length (see Remark 6.2.14). It is clarified that the APA, ESP, and PAPA algorithms are based on iterative relaxed-projections onto the same linear varieties with different metrics, respectively.

In the second part of this chapter (Sec. 6.3), we investigate the effects of the proposed algorithms in a practical application, the AEC problem. We present

an efficient AEC algorithm named *Adaptive Quadratic-metric PSP (AQ-PSP)*. In the design of the metric employed in the AQ-PSP algorithm, we can use the well-known fact that the room impulse responses decay exponentially on average and the exponential factor is modeled by means of sampling frequency and the time interval in which the reverberant sound energy drops down by 60 dB [69] (see Sec. 6.4.2). By employing a simple metric, the additional computational complexity compared with the Euclidean metric is negligible (see Remark 6.2.14). In Sec. 6.4, simulation tests are performed under severely noisy environments with recorded speech signal for the proposed algorithms [AQ-PSP and its variable-metric version named *Adaptive Variable-metric PSP (AV-PSP)*] compared with the ESP and PAPA algorithms. Numerical examples demonstrate the advantages of the proposed algorithms over the conventional algorithms and verify the significance of the choice of metric. Two simple metrics are compared with each other for different exponential factors. Respective suitable situations for AQ-PSP and AV-PSP are also discussed.

A preliminary version is to be presented at the EUSIPCO2006 conference [141].

6.2 Adaptive Parallel Quadratic-Metric Projection Algorithms

Throughout the chapter, to specify an inner product and its induced norm, we respectively use $\langle \boldsymbol{a}, \boldsymbol{b} \rangle_{\boldsymbol{Q}} := \boldsymbol{a}^T \boldsymbol{Q} \boldsymbol{b}$, $\forall \boldsymbol{a}, \boldsymbol{b} \in \mathcal{H}$ $(:= \mathbb{R}^N)$, and $\|\boldsymbol{a}\|_{\boldsymbol{Q}} := \sqrt{\langle \boldsymbol{a}, \boldsymbol{a} \rangle_{\boldsymbol{Q}}}$, $\forall \boldsymbol{a} \in \mathcal{H}$, where $\boldsymbol{Q} \in \mathbb{R}^{N \times N}$ is a positive definite matrix[2] (which will be denoted as $\boldsymbol{Q} \succ 0$). In the real Hilbert space $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\boldsymbol{Q}})$, the distance between arbitrary two elements is given by $\mathrm{d}_{\boldsymbol{Q}}(\boldsymbol{a}, \boldsymbol{b}) := \|\boldsymbol{a} - \boldsymbol{b}\|_{\boldsymbol{Q}}$, $\forall \boldsymbol{a}, \boldsymbol{b} \in \mathcal{H}$. Similarly, the distance between an arbitrary point $\boldsymbol{a} \in \mathcal{H}$ and a closed convex set C is given by $\mathrm{d}_{\boldsymbol{Q}}(\boldsymbol{a}, C) := \min_{\boldsymbol{b} \in C} \|\boldsymbol{a} - \boldsymbol{b}\|_{\boldsymbol{Q}}$, and the projection of $\boldsymbol{h} \in \mathcal{H}$ onto C is given as $P_C^{(\boldsymbol{Q})}(\boldsymbol{h}) := \arg\min_{\boldsymbol{b} \in C} \mathrm{d}_{\boldsymbol{Q}}(\boldsymbol{a}, \boldsymbol{b})$.

[2]Suppose that the original Hilbert space $(\mathcal{H}, \langle \cdot, \cdot \rangle)$ is infinite dimensional with the induced norm $\|\cdot\|$. In this case, we can define a different Hilbert space $(\mathcal{H}, \langle \cdot, \cdot \rangle_Q)$ with a new inner product $\langle \boldsymbol{a}, \boldsymbol{b} \rangle_Q := \langle Q\boldsymbol{a}, \boldsymbol{b} \rangle$, $\forall \boldsymbol{a}, \boldsymbol{b} \in \mathcal{H}$, by employing a self-adjoint bounded linear operator $Q : \mathcal{H} \to \mathcal{H}$ to be *strongly positive*; i.e., there exists $(\mathbb{R} \ni) \alpha > 0$ s.t. $\langle Q\boldsymbol{x}, \boldsymbol{x} \rangle \geq \alpha \|\boldsymbol{x}\|^2$, $\forall \boldsymbol{x} \in \mathcal{H}$. Such a Q is nothing but a positive definite matrix when $\mathcal{H}$ is finite dimensional (see, e.g., [123]).

6.2.1 Constant-Metric Version

Let $(\boldsymbol{h}_k)_{k\in\mathbb{N}}$ be a sequence of adaptive filtering vectors. To compute $\boldsymbol{h}_{k+1}$ from $\boldsymbol{h}_k$ at each $k \in \mathbb{N}$, we use information on the closed convex sets $C_i^{(k)}$, $i = 1, 2, \cdots, q$, that are defined by means of observable data so as to contain the estimandum $\boldsymbol{h}^*$ with high reliability. Define the weights to those data-dependent closed convex sets $w_i^{(k)} \in (0, 1]$, $i = 1, 2, \cdots, q$, $k \in \mathbb{N}$, satisfying $\sum_{i=1}^q w_i^{(k)} = 1$.

Consider here the real Hilbert space $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\boldsymbol{Q}})$, where $\boldsymbol{Q} \succ 0$ is a positive definite matrix designed appropriately by means of a priori knowledge on the estimandum. We now define the objective function $\Theta_k : \mathcal{H} \to \mathbb{R}$, $k \in \mathbb{N}$, as

$$\Theta_k(\boldsymbol{h}) := \begin{cases} \dfrac{1}{L_k}\displaystyle\sum_{i=1}^{q} w_i^{(k)} \mathrm{d}_{\boldsymbol{Q}}[\boldsymbol{h}_k, C_i^{(k)}]\mathrm{d}_{\boldsymbol{Q}}[\boldsymbol{h}, C_i^{(k)}] & \text{if } L_k := \sum_{i=1}^q w_i^{(k)} \mathrm{d}_{\boldsymbol{Q}}[\boldsymbol{h}_k, C_i^{(k)}] \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

In this case, $\mathrm{Fix}(T_{\mathrm{sp}(\Theta_k)}) = \mathrm{lev}_{\le}\Theta_k = \bigcap_{i\in\mathcal{J}_k} C_i^{(k)}$, where $\mathcal{J}_k := \left\{i = 1, 2, \cdots, q : \boldsymbol{h}_k \notin C_i^{(k)}\right\}$. Note that the factor $\mathrm{d}_{\boldsymbol{Q}}[\boldsymbol{h}_k, C_i^{(k)}]$, $i = 1, 2, \cdots, q$, is constant in terms of $\boldsymbol{h}$. Indeed, $\mathrm{d}_{\boldsymbol{Q}}[\boldsymbol{h}_k, C_i^{(k)}]$ is an automatically-determined weighting factor and gives a large weight to a set 'far' from $\boldsymbol{h}_k$ in the sense of the metric $\mathrm{d}_{\boldsymbol{Q}}$. Application of Scheme 2.0.2 to Θ_k derives the following algorithm.

Algorithm 6.2.1 (Adaptive Parallel Quadratic-Metric Projection (APQP) Algorithm). *For an arbitrarily chosen initial vector $\boldsymbol{h}_0 \in \mathcal{H}$, generate a sequence of adaptive filtering vectors $(\boldsymbol{h}_k)_{k\in\mathbb{N}} \subset \mathcal{H}$ as*

$$\boldsymbol{h}_{k+1} := T\left\{\boldsymbol{h}_k + \lambda_k \mathcal{M}_k^{(1)}\left[\sum_{i=1}^{q} w_i^{(k)} P_{C_i^{(k)}}^{(\boldsymbol{Q})}(\boldsymbol{h}_k) - \boldsymbol{h}_k\right]\right\},$$

$\forall k \in \mathbb{N}$, where $\lambda_k \in [0, 2]$ is the step size and

$$\mathcal{M}_k^{(1)} := \begin{cases} \dfrac{\displaystyle\sum_{i=1}^{q} w_i^{(k)} \left\| P_{C_i^{(k)}}^{(\boldsymbol{Q})}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\|_{\boldsymbol{Q}}^2}{\left\| \displaystyle\sum_{i=1}^{q} w_i^{(k)} P_{C_i^{(k)}}^{(\boldsymbol{Q})}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\|_{\boldsymbol{Q}}^2} & \textit{if } \boldsymbol{h}_k \notin \displaystyle\bigcap_{\iota\in\mathcal{I}_k} C_i^{(k)}, \\ 1 & \textit{otherwise.} \end{cases}$$

In the derivation of Algorithm 6.2.1, we use the following fact.

Fact 6.2.2. *Given any closed convex set $S \subset \mathcal{H}$, the following hold.*

(a) $\mathrm{d}_{\boldsymbol{Q}}(\boldsymbol{h}, S) = \left\| \boldsymbol{h} - P_S^{(\boldsymbol{Q})}(\boldsymbol{h}) \right\|_{\boldsymbol{Q}}$, $\forall \boldsymbol{h} \in \mathcal{H}$.

(b) *For the function $\phi : \mathcal{H} \to \mathbb{R}$, $\boldsymbol{h} \mapsto \mathrm{d}_{\boldsymbol{Q}}(\boldsymbol{h}, S)$,*

$$\partial_{(\boldsymbol{Q})}\phi(\boldsymbol{h}) \ni \phi'(\boldsymbol{h}) = \begin{cases} \dfrac{\boldsymbol{h} - P_S^{(\boldsymbol{Q})}(\boldsymbol{h})}{\mathrm{d}_{\boldsymbol{Q}}(\boldsymbol{h}, S)} & \text{if } \boldsymbol{h} \notin S, \\ \boldsymbol{0} & \text{otherwise.} \end{cases}$$

Algorithm 6.2.1 is endowed with the remarkable properties of APSM (see Fact 2.0.3). In fact, Algorithm 6.2.1 is a generalization of some existing algorithms, as shown below.

Example 6.2.3 (Special examples of Algorithm 6.2.1)**.**

(a) *Let $T = I$, $q = 1$, and $C_1^{(k)} := V_k$ be a linear variety. Then we get the following algorithm:*

$$\boldsymbol{h}_{k+1} := \boldsymbol{h}_k + \lambda_k \left[P_{V_k}^{(\boldsymbol{Q})}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right], \tag{6.2.1}$$

which turns out to be a general form of the ESP algorithm [77, 78] *(see Proposition* 6.2.4*).*

(b) *Let $T = I$, $\boldsymbol{Q} = \boldsymbol{I}$, and $C_i(k)$, $i = 1, 2, \cdots, q$, be closed half-spaces, as defined in Sec.* 6.3.2*. Then we obtain the adaptive-PSP algorithm* [129, 135]*. An efficient weight design for adaptive-PSP has been proposed in* [135].

Example 6.2.3-(a) implies that ESP is derived by APSM. A merit to derive an algorithm by APSM is that some remarkable properties of the algorithm come to light immediately with no extra efforts (see Fact 2.0.3).

Proposition 6.2.4. *In (6.2.1), let*[3] $V_k := \arg\min_{\boldsymbol{v} \in \mathcal{H}} \left\| \boldsymbol{U}_k^T \boldsymbol{v} - \boldsymbol{d}_k \right\|_{\boldsymbol{I}}$, $\forall k \in \mathbb{N}$, *where $\boldsymbol{U}_k \in \mathbb{R}^{N \times r}$ $(N > r)$ is the input matrix and $\boldsymbol{d}_k \in \mathbb{R}^r$ the output vector (see Sec. 6.3.1); if $\boldsymbol{U}_k$ has full column rank, then V_k is reduced to $V_k := \left\{ \boldsymbol{v} \in \mathcal{H} : \boldsymbol{U}_k^T \boldsymbol{v} = \boldsymbol{d}_k \right\}$. Then, (i) $\boldsymbol{Q} := \boldsymbol{I}$ yields APA [58, 81, 89, 93, 103], and (ii) $\boldsymbol{Q} := \boldsymbol{A}^{-1}$ with $\boldsymbol{A} := diag(\alpha_1, \alpha_2, \cdots, \alpha_N) \succ 0$ yields the ESP algorithm [77, 78],*

[3] To see V_k is a linear variety, see [129, Appendix B].

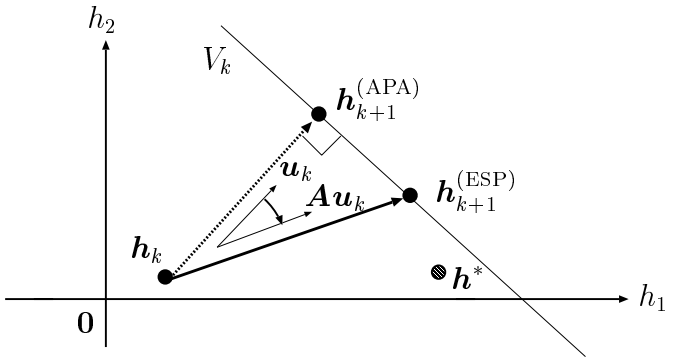


Figure 6.2.1: A geometric interpretation of ESP for $r = 1$ and APA for $r = 1$ (i.e., NLMS).

where $\alpha_i := \alpha_0 \gamma^{i-1}$ $(i = 1, 2, \cdots, N)$ *with a positive constant* α_0 *and the exponential ratio* $\gamma \in (0, 1)$. *The notation diag*$(\cdots)$ *is used for a (block) diagonal matrix; the arguments are scalars or square matrices.*

Proof: See Sec. 6.2.3.A. □

For intuitive understandings to the relations between APA and ESP, we present a geometric interpretation in Fig. 6.2.1. For simplicity, we set $N = 2$. According to the exponentially decaying feature of room impulse responses, we put $\boldsymbol{h}^*$ close to the axis of h_1 rather than that of h_2. For both APA and ESP, we let $r = 1$ and $\lambda_k = 1$, $k \in \mathbb{N}$. In this case, the algorithms can be written as $\boldsymbol{h}_{k+1} := \boldsymbol{h}_k - \lambda_k \left[e_k(\boldsymbol{h}_k) / \left\| \boldsymbol{Q}^{-1}\boldsymbol{u}_k \right\|_{\boldsymbol{Q}}^2 \right] \boldsymbol{Q}^{-1}\boldsymbol{u}_k$, where $\boldsymbol{Q} = \boldsymbol{A}^{-1}$ for ESP and $\boldsymbol{Q} = \boldsymbol{I}$ for NLMS (i.e., APA with $r = 1$). By $V_k = \{\boldsymbol{h} \in \mathcal{H} : \boldsymbol{u}_k^T \boldsymbol{h} = d_k\} = \{\boldsymbol{h} \in \mathcal{H} : \langle \boldsymbol{A}\boldsymbol{u}_k, \boldsymbol{h} \rangle_{\boldsymbol{A}^{-1}} = d_k\}$, one can see that the normal vector of V_k with the metric $\mathrm{d}_{\boldsymbol{A}^{-1}}$ is $\boldsymbol{A}\boldsymbol{u}_k$. Figure 6.2.1 illustrates that ESP gives a better direction of update than APA due to the use of $\mathrm{d}_{\boldsymbol{A}^{-1}}$.

To derive another algorithm based on a *min-max criterion*, we define the objective function $\Theta_k : \mathcal{H} \to \mathbb{R}$, $k \in \mathbb{N}$, as $\Theta_k(\boldsymbol{h}) := \max\limits_{i=1,2,\cdots,q} \mathrm{d}_{\boldsymbol{Q}}(\boldsymbol{h}, C_i^{(k)})$, $\forall k \in \mathbb{N}$. In this case, $\mathrm{Fix}(T_{\mathrm{sp}(\Theta_k)}) = \mathrm{lev}_{\leq} \Theta_k = \bigcap_{i=1}^q C_i^{(k)}$. Define $\mathcal{J}_k(\boldsymbol{h}) := \{i = 1, 2, \cdots, q : \mathrm{d}_{\boldsymbol{Q}}(\boldsymbol{h}, C_i^{(k)}) = \Theta_k(\boldsymbol{h})\}$. Application of Scheme 2.0.2 to Θ_k derives the following algorithm.

Algorithm 6.2.5 (Adaptive Parallel Min-Max Quadratic-Metric Projection (APMQP) Algorithm)**.** *For an arbitrarily chosen initial vector* $\boldsymbol{h}_0 \in \mathcal{H}$, *generate a*

sequence of adaptive filtering vectors $(\boldsymbol{h}_k)_{k\in\mathbb{N}} \subset \mathcal{H}$ *as*

$$\boldsymbol{h}_{k+1} := T\left\{\boldsymbol{h}_k + \lambda_k \mathcal{M}_k^{(2)}\left[\sum_{i\in\mathcal{J}_k(\boldsymbol{h}_k)} \nu_i^{(k)} P_{C_i^{(k)}}^{(\boldsymbol{Q})}(\boldsymbol{h}_k) - \boldsymbol{h}_k\right]\right\},$$

$\forall k \in \mathbb{N}$, *where* $\lambda_k \in [0,2]$ *is the step size,* $\nu_\iota^{(k)} \geq 0$ *satisfies* $\sum_{\iota\in\mathcal{J}_k(\boldsymbol{h}_k)} \nu_\iota^{(k)} = 1$, *and*

$$\mathcal{M}_k^{(2)} := \begin{cases} \dfrac{\max\limits_{i\in\mathcal{J}_k(\boldsymbol{h}_k)} \left\| P_{C_i^{(k)}}^{(\boldsymbol{Q})}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\|_{\boldsymbol{Q}}^2}{\left\| \sum\limits_{i\in\mathcal{J}_k(\boldsymbol{h}_k)} \nu_i^{(k)} P_{C_i^{(k)}}^{(\boldsymbol{Q})}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\|_{\boldsymbol{Q}}^2} & \text{if } \boldsymbol{h}_k \notin \bigcap\limits_{\iota\in\mathcal{I}_k} C_i^{(k)}, \\ 1 & \text{otherwise.} \end{cases}$$

We name Algorithm 6.2.5 *Adaptive Parallel Min-max Quadratic-metric Projection (APMQP) algorithm* because the algorithm asymptotically minimizes the sequence of max functions (see Fact 2.0.3). To derive Algorithm 6.2.5, we use the following fact.

Fact 6.2.6. *Given any closed convex sets* $S_i \subset \mathcal{H}$ *for* $i = 1, 2, \cdots, q$, *define a function* $\phi : \mathcal{H} \to \mathbb{R}$, $\boldsymbol{h} \mapsto \max_{i=1,2,\cdots,q} \mathrm{d}_{\boldsymbol{Q}}(\boldsymbol{h}, S_i)$. *Then,*

$$\partial_{(\boldsymbol{Q})}\phi(\boldsymbol{h}) \ni \phi'(\boldsymbol{h}) := \begin{cases} \phi(\boldsymbol{h})^{-1}\left[\boldsymbol{h} - \sum_{i\in\mathcal{J}(\boldsymbol{h})} \nu_i P_{S_i}^{(\boldsymbol{Q})}(\boldsymbol{h})\right] & \text{if } \boldsymbol{h} \notin \bigcap\limits_{\iota\in\mathcal{J}} S_i, \\ \boldsymbol{0} & \text{otherwise,} \end{cases}$$

where $\mathcal{J}(\boldsymbol{h}) := \{i = 1, 2, \cdots, q : \mathrm{d}_{\boldsymbol{Q}}(\boldsymbol{h}, S_i) = \phi(\boldsymbol{h})\}$ *and* $\nu_i \geq 0$, $\forall i \in \mathcal{J}(\boldsymbol{h})$, *satisfies* $\sum_{i\in\mathcal{J}(\boldsymbol{h})} \nu_i = 1$.

Example 6.2.7 (Design of the strongly attracting nonexpansive mapping T).

(a) *A simplest example of the strongly attracting nonexpansive mapping* T *is a (single) projection operator* $P_K^{(\boldsymbol{Q})}$ *(which corresponds to the original version of APSM* [124, 126]*), where* $K(\ni \boldsymbol{h}^*)$ *is a hard constraint set arising from, e.g., physical aspects etc. In this case, it is obvious that* $\mathrm{Fix}(T) = K$ *and the mapping* $P_K^{(\boldsymbol{Q})}$ *is* 1*-attracting nonexpansive. At any time instant* k, *moreover, it is guaranteed that* $\boldsymbol{h}_k \in K$.

One may have multiple constraint sets K_j, $j = 1, 2, \cdots, m$, to be used; e.g., in case that $P_K^{(\boldsymbol{Q})}$ has no closed-form expression or has prohibitively high computational complexity, one can approximate K by the intersection of multiple 'simpler' closed convex sets $K_j \supset K$, $j = 1, 2, \cdots, m$; here we use 'simpler' in the sense that the projections onto the convex sets require less computational loads. In such scenarios, there are various ways to design T for handling those multiple sets in asymptotic sense without using the projection onto their intersection, as shown in Example 6.2.7-(b)–(d) *below.*

(b) *Let* $T_1 := \prod_{j=1}^{m} \left[(1-\lambda_j)I + \lambda_j P_{K_j}^{(\boldsymbol{Q})}\right]$ *($m \geq 2$), where $\lambda_j \in (0, 2)$, $\forall j = 1, 2, \cdots, m$. Then, T_1 is strongly attracting nonexpansive with* $\mathrm{Fix}(T) = \bigcap_{j=1}^{m} K_j$ [99, 100]. *More precisely, T_1 is* $\left(\sum_{j=1}^{m} \frac{\lambda_j}{2-\lambda_j}\right)^{-1}$*-attracting nonexpansive. In particular, if $\lambda_j = 1$, $\forall j = 1, 2, \cdots, m$, then T_1 is $1/m$-attracting nonexpansive. In the robust adaptive beamforming application, the efficacy of T_1 has been verified* [101, 102], *where $\lambda_j = 1$, $\forall j = 1, 2, \cdots, m$, and $\boldsymbol{Q} = \boldsymbol{I}$ are employed.*

(c) *Let* $T_2 := \sum_{j=1}^{m} \omega_j^{(k)} \left[(1-\lambda_j)I + \lambda_j P_{K_j}^{(\boldsymbol{Q})}\right]$ *($m \geq 2$), where $\lambda_j \in (0, 2)$, $\forall j = 1, 2, \cdots, m$, and the weights $\omega_j^{(k)} > 0$, $\forall k \in \mathbb{N}$, $j = 1, 2, \cdots, m$, satisfy $\sum_{j=1}^{m} \omega_j^{(k)} = 1$. Then, T_2 is strongly attracting nonexpansive with* $\mathrm{Fix}(T) = \bigcap_{j=1}^{m} K_j$. *In particular, if $\lambda_j = \lambda \in (0, 2)$, $\forall j = 1, 2, \cdots, m$, then T_2 is a* $\frac{2-\lambda}{\lambda}$*-attracting nonexpansive mapping.*

Remark 6.2.8 (On the strongly attracting nonexpansive mapping).

(a) *Designing the strongly attracting nonexpansive mapping T as T_1 or T_2 in Example* 6.2.7, *it is guaranteed that the sequence of adaptive filtering vectors $(\boldsymbol{h}_k)_{k\in\mathbb{N}}$ generated by Algorithm* 6.2.1 *converges to a point $\widehat{\boldsymbol{h}} \in \mathrm{Fix}(T) = \bigcap_{j=1}^{m} K_j$ under reasonable conditions (see Fact* 2.0.3*), although there is no guarantee that $\boldsymbol{h}_k \in \mathrm{Fix}(T)$ at each iteration k.*

(b) *In case that one has a single constraint set being a linear variety, there exists a more efficient method named embedded-constraint parallel projection techniques* [127, 132], *and its efficacy has been verified in applications to DS/CDMA wireless communication systems* [132, 140].

(c) *Besides T_1 and T_2, there exists large variety in the choice of T based on Fact* 2.0.1. *Performance evaluations of such design of T are currently under investigation.*

The following proposition suggests the stability of algorithms derived from Scheme 2.0.2 including APQP and APMQP.

Proposition 6.2.9 (Monotone approximation property of Scheme 2.0.2). *Suppose that* $\mathrm{Fix}(T) \cap \mathrm{Fix}(T_{\mathrm{sp}(\Theta_k)}) \neq \emptyset$. *For any initial vector* $\boldsymbol{h}_0 \in \mathcal{H}$, *the sequence* $(\boldsymbol{h}_k)_{k\in\mathbb{N}} \subset \mathcal{H}$ *generated by Scheme* 2.0.2 *satisfies the following.*

(a) *For any* $\boldsymbol{h}^* \in \mathrm{Fix}(\Phi_k)$ $[= \mathrm{Fix}(T) \cap \mathrm{Fix}(T_{\mathrm{sp}(\Theta_k)}) \neq \emptyset]$ *(see p. 11 for the definition of* Φ_k*),*

$$\|\boldsymbol{h}_{k+1} - \boldsymbol{h}^*\| \leq \|\boldsymbol{h}_k - \boldsymbol{h}^*\|, \ \forall k \in \mathbb{N}.$$

(b) *If in particular* $\lambda_k \in [0, 2)$ *and* $\boldsymbol{h}_k \notin \mathrm{Fix}(\Phi_k)$, *then for any* $\boldsymbol{h}^* \in \mathrm{Fix}(\Phi_k)$,

$$\|\boldsymbol{h}_{k+1} - \boldsymbol{h}^*\| < \|\boldsymbol{h}_k - \boldsymbol{h}^*\|, \ \forall k \in \mathbb{N}.$$

Proof: See Sec. 6.2.3.B. □

6.2.2 Variable-Metric Version

The metric $\mathrm{d}_{\boldsymbol{Q}}$ is constant in time in Algorithms 6.2.1 and 6.2.5. Although this is essential to ensure the properties of APSM (see Fact 2.0.3), it would be valuable in practice to present a more general form where the metric itself changes in time. We thus show a variable-metric version of APQP below; a variable-metric version of APMQP can also be given in a similar way.

Algorithm 6.2.10 (Adaptive Parallel Variable-Metric Projection (APVP) Algorithm). *Let T_k, $\forall k \in \mathbb{N}$, be a strongly attracting nonexpansive mapping in the real Hilbert space $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\boldsymbol{Q}_k})$, where $\boldsymbol{Q}_k \succ 0$, $\forall k \in \mathbb{N}$. For an arbitrarily chosen initial vector $\boldsymbol{h}_0 \in \mathcal{H}$, generate a sequence of adaptive filtering vectors $(\boldsymbol{h}_k)_{k\in\mathbb{N}} \subset \mathcal{H}$ as*

$$\boldsymbol{h}_{k+1} := T_k \left\{ \boldsymbol{h}_k + \lambda_k \mathcal{M}_k^{(3)} \left[\sum_{i=1}^{q} w_i^{(k)} P_{C_i^{(k)}}^{(\boldsymbol{Q}_k)}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right] \right\},$$

$\forall k \in \mathbb{N}$, *where* $\lambda_k \in [0,2]$ *is the step size and*

$$\mathcal{M}_k^{(3)} := \begin{cases} \dfrac{\sum_{i=1}^{q} w_i^{(k)} \left\| P_{C_i^{(k)}}^{(\boldsymbol{Q}_k)}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\|_{\boldsymbol{Q}_k}^2}{\left\| \sum_{i=1}^{q} w_i^{(k)} P_{C_i^{(k)}}^{(\boldsymbol{Q}_k)}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\|_{\boldsymbol{Q}_k}^2} & \text{if } \boldsymbol{h}_k \notin \bigcap_{\iota \in \mathcal{I}_k} C_i^{(k)}, \\ 1 & \text{otherwise.} \end{cases}$$

The mapping T_k can be designed in a way similar to Example 6.2.7. Algorithm 6.2.10 is a generalization of some existing algorithms, as shown below.

Example 6.2.11 (Special examples of Algorithm 6.2.10)**.** *Let* $T = I$, $q = 1$, *and* $C_1^{(k)} := V_k$ *be a linear variety. Then we get the following algorithm:*

$$\boldsymbol{h}_{k+1} := \boldsymbol{h}_k + \lambda_k \left[P_{V_k}^{(\boldsymbol{Q}_k)}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right], \tag{6.2.2}$$

which turns out to be a general form of the Proportionate NLMS (PNLMS) algorithm [9, 29, 33, 41, 86] *and the Proportionate APA (PAPA) algorithm* [10, 48, 64, 120] *(see Proposition* 6.2.12*).*

Proposition 6.2.12. *Define* $V_k := \left\{ \boldsymbol{h} \in \mathcal{H} : \boldsymbol{U}_k^T \boldsymbol{h} = \boldsymbol{d}_k \right\}$ *(*$\boldsymbol{U}_k \in \mathbb{R}^{N \times r}$, $\boldsymbol{d}_k \in \mathbb{R}^r$*) as in Proposition* 6.2.4*. Then,* $\boldsymbol{Q}_k := \boldsymbol{G}_k^{-1}$ *yields the Proportionate APA (PAPA) algorithm* [10, 48, 64, 120] *(or the Proportionate NLMS (PNLMS) algorithm* [9, 29, 33, 41, 86] *if* $r = 1$*), where* $\boldsymbol{G}_k := \mathrm{diag}(\alpha_1^{(k)}, \alpha_2^{(k)}, \cdots, \alpha_N^{(k)}) \succ 0$ *is defined as follows* [33, 41]*:*

$$\alpha_i^{(k)} := \frac{\gamma_i^{(k)}}{\sum_{j=1}^{N} \gamma_j^{(k)}}, \ \forall i = 1, 2, \cdots, N,$$

where $\gamma_i^{(k)} := \max\left\{ \sigma L_{\max}^{(k)}, |h_i^{(k)}| \right\}$ *and* $L_{\max}^{(k)} := \max\left\{ \delta, |h_1^{(k)}|, |h_2^{(k)}|, \ \cdots, |h_N^{(k)}| \right\}$. *Here,* σ *and* δ *are small positive constants and* $h_i^{(k)}$ *the* i*th component of* $\boldsymbol{h}_k$*. Modified designs of matrix* $\boldsymbol{G}_k$ *have also been proposed in* [9, 10, 29, 48, 64, 86, 120].

Proof: See Sec. 6.2.3.C. □

In [120], the set-membership PAPA algorithm has been derived by using $\|\cdot\|_{\boldsymbol{G}_k^{-1}}$ with the criterion commonly used to derive the APA. Propositions 6.2.4 and 6.2.12

provide with an interesting interpretation that the APA, ESP, and PAPA algorithms are based on iterative relaxed-projections onto the same linear varieties V_k with the different metrics $\mathrm{d}_{\boldsymbol{I}}$, $\mathrm{d}_{\boldsymbol{A}^{-1}}$, and $\mathrm{d}_{\boldsymbol{G}_k^{-1}}$, respectively.

As a direct consequence of Proposition 6.2.9, we obtain the following proposition on Algorithm 6.2.10.

Proposition 6.2.13 (Monotone approximation property of Algorithm 6.2.10). *Suppose that* $\mathrm{Fix}(T_k) \cap \mathrm{Fix}(T_{\mathrm{sp}(\Theta_k)}) \neq \emptyset$. *For any initial vector* $\boldsymbol{h}_0 \in \mathcal{H}$, *the sequence* $(\boldsymbol{h}_k)_{k\in\mathbb{N}} \subset \mathcal{H}$ *generated by Algorithm* 6.2.10 *satisfies the following.*

(a) *For any* $\boldsymbol{h}^* \in \mathrm{Fix}(\Phi_k)$ $[= \mathrm{Fix}(T_k) \cap \mathrm{Fix}(T_{\mathrm{sp}(\Theta_k)}) \neq \emptyset]$,

$$\|\boldsymbol{h}_{k+1} - \boldsymbol{h}^*\|_{\boldsymbol{Q}_k} \leq \|\boldsymbol{h}_k - \boldsymbol{h}^*\|_{\boldsymbol{Q}_k}, \ \forall k \in \mathbb{N}.$$

(b) *If in particular* $\lambda_k \in (0, 2)$ *and* $\boldsymbol{h}_k \notin \mathrm{Fix}(\Phi_k)$, *then for any* $\boldsymbol{h}^* \in \mathrm{Fix}(\Phi_k)$,

$$\|\boldsymbol{h}_{k+1} - \boldsymbol{h}^*\|_{\boldsymbol{Q}_k} < \|\boldsymbol{h}_k - \boldsymbol{h}^*\|_{\boldsymbol{Q}_k}, \ \forall k \in \mathbb{N}.$$

Proof: See Sec. 6.2.3.D. □

Note that, unlike the monotonicity in Proposition 6.2.9, the approximation is effectively improved from $\boldsymbol{h}_k$ to $\boldsymbol{h}_{k+1}$ in the sense of a reasonable and time-varying metric. A convergence analysis of proportionate-type NLMS algorithms has been presented in [32] under the assumption of sufficiently small step size.

Finally, remarks on Algorithms 6.2.1 and 6.2.10 are given below.

Remark 6.2.14.

(a) *(Inherent parallelism of Algorithm* 6.2.1*) In the update equation of Algorithm* 6.2.1, *each projection in the summation can be computed independently, thus the algorithm has the inherently parallel structure* [4, 17, 22]. *In fact, in addition that the algorithm is relevant to parallel implementation, it has a fault tolerance nature; i.e., a possible trouble in some of the employed processors does not seriously affect the overall performance of the algorithm (which is not true for the other major adaptive algorithms).*

(b) *(Overall complexity of Algorithms* 6.2.1, 6.2.5, *and* 6.2.10*) Note firstly that the complexity shown below excludes the computation to design* $\boldsymbol{Q}_k$ *for Algorithm* 6.2.10 *(e.g., in* [33], *the 'strobe down' technique is introduced in*

PNLMS to reduce the complexity for designing $\boldsymbol{Q}_k$). The complexity of the proposed algorithms depends on the metric employed. By employing an efficient metric (i.e., the matrix $\boldsymbol{Q}$ (or $\boldsymbol{Q}_k$) has a special structure such as diagonal), the overall computational complexity (the number of multiplications/divisions) is kept $O(N)$ [If $\boldsymbol{Q}$ (or $\boldsymbol{Q}_k$) has no special structure, then the algorithms require matrix-vector multiplications]. In addition, by employing q concurrent processors, the computational complexity imposed on each processor at each iteration is approximately $(2r+4)N$. This is the same as the complexity of the adaptive-PSP algorithm with the Euclidean metric [129] *(see* [135] *for a computational comparison among the adaptive-PSP and other major algorithms). The proposed algorithms can significantly raise, by increasing q, convergence speed while keeping low time-consumption, which is very important for real-time applications including AEC. Moreover, although Algorithm* 6.2.5 *should execute the max-function, it is negligible when q is relatively small.*

6.2.3 Proofs

A. Proof of Proposition 6.2.4

The first goal is to show

$$P_{V_k}^{(\boldsymbol{Q})}(\boldsymbol{h}) = \boldsymbol{h} - \boldsymbol{Q}^{-1/2}\left(\boldsymbol{U}_k^T\boldsymbol{Q}^{-1/2}\right)^{\dagger}\boldsymbol{e}_k(\boldsymbol{h}),\ \forall \boldsymbol{h}\in\mathcal{H}, \tag{6.2.3}$$

where the operation $(\cdot)^{\dagger}$ denotes the *Moore-Penrose pseudoinverse* [6, 85]. By the definition of $P_{V_k}^{(\boldsymbol{Q})}$, we have $\boldsymbol{z}^* := P_{V_k}^{(\boldsymbol{Q})}(\boldsymbol{h}) - \boldsymbol{h} \in \arg\min_{\boldsymbol{z}\in V_k-\boldsymbol{h}} \|\boldsymbol{z}\|_{\boldsymbol{Q}}$, where $V_k - \boldsymbol{h} := \{\boldsymbol{v}-\boldsymbol{h} : \boldsymbol{v}\in V_k\} = \arg\min_{\boldsymbol{z}\in\mathcal{H}} \left\|\boldsymbol{U}_k^T\boldsymbol{z} + \boldsymbol{e}_k(\boldsymbol{h})\right\|_{\boldsymbol{I}}$. Defining $\widetilde{\boldsymbol{z}} := \boldsymbol{Q}^{1/2}\boldsymbol{z}$ and $\widetilde{\boldsymbol{U}}_k := \boldsymbol{Q}^{-1/2}\boldsymbol{U}_k$, we have $\|\boldsymbol{z}\|_{\boldsymbol{Q}} = \|\widetilde{\boldsymbol{z}}\|_{\boldsymbol{I}}$ and $\left\|\boldsymbol{U}_k^T\boldsymbol{z} + \boldsymbol{e}_k(\boldsymbol{h})\right\|_{\boldsymbol{I}} = \left\|\widetilde{\boldsymbol{U}}_k^T\widetilde{\boldsymbol{z}} + \boldsymbol{e}_k(\boldsymbol{h})\right\|_{\boldsymbol{I}}$. By the definition of the pseudoinverse operator, the unique vector of minimum Euclidean norm $\widetilde{\boldsymbol{z}}^*$ $(= \boldsymbol{Q}^{1/2}\boldsymbol{z}^*)$ in $\arg\min_{\widetilde{\boldsymbol{z}}\in\mathcal{H}} \left\|\widetilde{\boldsymbol{U}}_k^T\widetilde{\boldsymbol{z}} + \boldsymbol{e}_k(\boldsymbol{h})\right\|_{\boldsymbol{I}}$ is given by $-\left(\widetilde{\boldsymbol{U}}_k^T\right)^{\dagger}\boldsymbol{e}_k(\boldsymbol{h})$. Hence, we have $\boldsymbol{z}^* = -\boldsymbol{Q}^{-1/2}\left(\widetilde{\boldsymbol{U}}_k^T\right)^{\dagger}\boldsymbol{e}_k(\boldsymbol{h})$, which verifies (6.2.3).

From (6.2.3), we can immediately verify that $\boldsymbol{Q} := \boldsymbol{I}$ in (6.2.1) yields $\boldsymbol{h}_{k+1} := \boldsymbol{h}_k - \lambda_k(\boldsymbol{U}_k^T)^{\dagger}\boldsymbol{e}_k(\boldsymbol{h}_k)$, which coincides with the original formulation of APA [89].

Similarly, $\boldsymbol{Q} := \boldsymbol{A}^{-1}$ in (6.2.1) yields

$$\boldsymbol{h}_{k+1} := \boldsymbol{h}_k - \lambda_k \boldsymbol{A}^{1/2} (\boldsymbol{U}_k^T \boldsymbol{A}^{1/2})^\dagger \boldsymbol{e}_k(\boldsymbol{h}_k). \tag{6.2.4}$$

In the original formulation of ESP, the equations [77, (33) and (34)] are rewritten as $\boldsymbol{U}_k^T \boldsymbol{A} \boldsymbol{U}_k [\beta_1(k), \beta_2(k)]^T = \boldsymbol{e}_k(\boldsymbol{h}_k)$, which is guaranteed to have a solution only under the assumption that $\boldsymbol{U}_k^T \boldsymbol{A} \boldsymbol{U}_k$ is invertible ($\Leftrightarrow \boldsymbol{U}_k$ has full column rank). Under this assumption, it holds that $\left(\widetilde{\boldsymbol{U}}_k^T\right)^\dagger = \widetilde{\boldsymbol{U}}_k \left(\widetilde{\boldsymbol{U}}_k^T \widetilde{\boldsymbol{U}}_k\right)^{-1}$, by which with (6.2.4) we obtain the update recursion $\boldsymbol{h}_{k+1} := \boldsymbol{h}_k - \lambda_k \boldsymbol{A} \boldsymbol{U}_k (\boldsymbol{U}_k^T \boldsymbol{A} \boldsymbol{U}_k)^{-1} \boldsymbol{e}_k(\boldsymbol{h}_k)$; it is readily verified that this recursion is equivalent to the original formulation in [77]. □

B. Proof of Proposition 6.2.9

Although the proof can be given more simply with the aid of [99, Proposition 2.5], we present a complete proof for self-containedness below.

(i) Let $\lambda_k \in (0, 2)$. Then, it is obvious that $\mathrm{Fix}((1-\lambda_k)I + \lambda_k T_{\mathrm{sp}(\Theta_k)}) = \mathrm{Fix}(T_{\mathrm{sp}(\Theta_k)})$, and thus $\mathrm{Fix}(\Phi_k) = \mathrm{Fix}(T) \cap \mathrm{Fix}(T_{\mathrm{sp}(\Theta_k)})$ by $\mathrm{Fix}(T) \cap \mathrm{Fix}(T_{\mathrm{sp}(\Theta_k)}) \neq \emptyset$ [127]. Since $T_{\mathrm{sp}(\Theta_k)}$ is firmly (i.e., 1-attracting) nonexpansive [127], $(1-\lambda_k)I + \lambda_k T_{\mathrm{sp}(\Theta_k)}$ is $\dfrac{2-\lambda_k}{\lambda_k}$-attracting quasi-nonexpansive. Hence, Φ_k is η_{Φ_k}-attracting quasi-nonexpansive with $\eta_{\Phi_k} = \dfrac{\eta(2-\lambda_k)}{\eta\lambda_k + 2 - \lambda_k}$, which follows that

$$\begin{aligned} \|\boldsymbol{h}_{k+1} - \boldsymbol{h}^*\|^2 = \|\Phi_k(\boldsymbol{h}_k) - \boldsymbol{h}^*\|^2 &\leq \|\boldsymbol{h}_k - \boldsymbol{h}^*\|^2 - \eta_{\Phi_k} \|\boldsymbol{h}_k - \Phi_k(\boldsymbol{h}_k)\|^2 \\ &\leq \|\boldsymbol{h}_k - \boldsymbol{h}^*\|^2, \qquad \forall \boldsymbol{h}^* \in \mathrm{Fix}(\Phi_k). \end{aligned} \tag{6.2.5}$$

Moreover, by $\boldsymbol{h}_k \notin \mathrm{Fix}(\Phi_k) \Leftrightarrow \boldsymbol{h}_k - \Phi_k(\boldsymbol{h}_k) \neq \boldsymbol{0}$,

$$\boldsymbol{h}_k \notin \mathrm{Fix}(\Phi_k) \Rightarrow \|\boldsymbol{h}_{k+1} - \boldsymbol{h}^*\| < \|\boldsymbol{h}_k - \boldsymbol{h}^*\|, \ \forall \boldsymbol{h}^* \in \mathrm{Fix}(\Phi_k). \tag{6.2.6}$$

(ii) Let $\lambda_k = 0$. Then $T_{\mathrm{sp}(\Theta_k)} = I$, and $\Phi_k = T$ is η-attracting nonexpansive with

$\mathrm{Fix}(\Phi_k) = \mathrm{Fix}(T) \cap \mathrm{Fix}(T_{\mathrm{sp}(\Theta_k)}) = \mathrm{Fix}(T)$, which follows that

$$\begin{aligned}\|\boldsymbol{h}_{k+1} - \boldsymbol{h}^*\|^2 = \|\Phi_k(\boldsymbol{h}_k) - \boldsymbol{h}^*\|^2 &\leq \|\boldsymbol{h}_k - \boldsymbol{h}^*\|^2 - \eta \|\boldsymbol{h}_k - \Phi_k(\boldsymbol{h}_k)\|^2 \\ &\leq \|\boldsymbol{h}_k - \boldsymbol{h}^*\|^2, \qquad \forall \boldsymbol{h}^* \in \mathrm{Fix}(\Phi_k). \end{aligned} \tag{6.2.7}$$

Moreover, by $\boldsymbol{h}_k \notin \mathrm{Fix}(\Phi_k) \Leftrightarrow \boldsymbol{h}_k - \Phi_k(\boldsymbol{h}_k) \neq \boldsymbol{0}$,

$$\boldsymbol{h}_k \notin \mathrm{Fix}(\Phi_k) \Rightarrow \|\boldsymbol{h}_{k+1} - \boldsymbol{h}^*\| < \|\boldsymbol{h}_k - \boldsymbol{h}^*\|, \ \forall \boldsymbol{h}^* \in \mathrm{Fix}(\Phi_k). \tag{6.2.8}$$

(iii) Let $\lambda_k = 2$. Then, $\Phi_k = T(2T_{\mathrm{sp}(\Theta_k)} - I)$. Since $T_{\mathrm{sp}(\Theta_k)}$ is firmly quasi-nonexpansive, $2T_{\mathrm{sp}(\Theta_k)} - I$ is quasi-nonexpansive (see Fact 2.0.1), and it is obvious that $\mathrm{Fix}(2T_{\mathrm{sp}(\Theta_k)} - I) = \mathrm{Fix}(T_{\mathrm{sp}(\Theta_k)})$. Hence, Φ_k is quasi-nonexpansive with $\mathrm{Fix}(\Phi_k) = \mathrm{Fix}(T) \cap \mathrm{Fix}(T_{\mathrm{sp}(\Theta_k)})$, which follows that

$$\|\boldsymbol{h}_{k+1} - \boldsymbol{h}^*\|^2 = \|\Phi_k(\boldsymbol{h}_k) - \boldsymbol{h}^*\|^2 \leq \|\boldsymbol{h}_k - \boldsymbol{h}^*\|^2, \ \forall \boldsymbol{h}^* \in \mathrm{Fix}(\Phi_k). \tag{6.2.9}$$

(6.2.5), (6.2.7), and (6.2.9) verify Proposition 6.2.9-(a), and (6.2.6) and (6.2.8) verify Proposition 6.2.9-(b). □

C. Proof of Proposition 6.2.12

PAPA is essentially given as

$$\boldsymbol{h}_{k+1} := \boldsymbol{h}_k - \lambda_k \boldsymbol{G}_k \boldsymbol{U}_k (\boldsymbol{U}_k^T \boldsymbol{G}_k \boldsymbol{U}_k)^{-1} \boldsymbol{e}_k(\boldsymbol{h}_k), \tag{6.2.10}$$

although, in the original formulation of PAPA in [48], a regularization parameter is introduced in order to force the matrix $\boldsymbol{U}_k^T \boldsymbol{G}_k \boldsymbol{U}_k$ to be invertible. In the same way as in Appendix 6.2.3, we can show that (6.2.2) for $\boldsymbol{Q}_k := \boldsymbol{G}_k^{-1}$ is reduced to (6.2.10) under the assumption that $\boldsymbol{U}_k^T \boldsymbol{G}_k \boldsymbol{U}_k$ is invertible, which completes the proof. □

D. Proof of Proposition 6.2.13

The proposition is proved in the same way as the proof of Proposition 6.2.9, because the time-variability of $\boldsymbol{Q}_k$ and T_k makes no difference in the proof. □

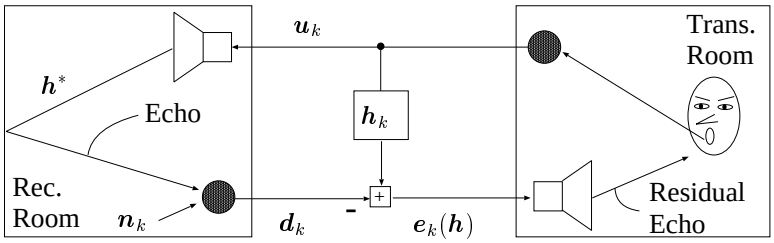


Figure 6.3.2: Acoustic echo canceling scheme.

6.3 Efficient Acoustic Echo Cancellation by Effective Metric

6.3.1 Acoustic Echo Canceling Problem

A basic model of AEC system [15, 56] is illustrated in Fig. 6.3.2. Let $k \in \mathbb{N}$ denote the time index and $N \in \mathbb{N}^*$ the length of echo canceler $\boldsymbol{h}_k$. For notational simplicity, we let the length of the estimandum $\boldsymbol{h}^*$, i.e., echo impulse response, be also N. With a sequence of input signals $(u_k)_{k\in\mathbb{N}} \subset \mathbb{R}$, let $(\boldsymbol{u}_k)_{k\in\mathbb{N}} \subset \mathcal{H}$ be a sequence of input vectors defined as $\boldsymbol{u}_k := [u_k, u_{k-1}, \cdots, u_{k-N+1}]^T$. For $r \in \mathbb{N}^*$, define $\boldsymbol{U}_k := [\boldsymbol{u}_k, \boldsymbol{u}_{k-1}, \cdots, \boldsymbol{u}_{k-r+1}] \in \mathbb{R}^{N\times r}$ (usually $r \ll N$). Also define the noise vector as $\boldsymbol{n}_k := [n_k, n_{k-1}, \cdots, n_{k-r+1}]^T \in \mathbb{R}^r$, $\forall k \in \mathbb{N}$, where $(n_k)_{k\in\mathbb{N}}$ is a sequence of additive noise process. We introduce the linear model for the data process $(\boldsymbol{d}_k)_{k\in\mathbb{N}} \subset \mathbb{R}^r$: $\boldsymbol{d}_k := \boldsymbol{U}_k^T\boldsymbol{h}^* + \boldsymbol{n}_k$. The goal of the echo cancellation is to remove (or cancel) the echo part $\boldsymbol{U}_k^T\boldsymbol{h}^*$ in $\boldsymbol{d}_k$ by subtracting the output of adaptive (linear) filter $\boldsymbol{h}_k \in \mathcal{H}$, $k \in \mathbb{N}$, as $\boldsymbol{d}_k - \boldsymbol{U}_k^T\boldsymbol{h}_k$. Since $\boldsymbol{h}_k \approx \boldsymbol{h}^*$ implies successful echo cancellation, the problem can be interpreted as *system identification* (i.e., identify an unknown system $\boldsymbol{h}^*$ by means of input-output relations), which is also known as adaptive filtering.

6.3.2 Proposed Acoustic Echo Canceling Algorithm

In this section, we present an efficient AEC algorithm based on (i) parallel subgradient projection onto data-dependent sets with a metric simpler than $\mathrm{d}_{\boldsymbol{A}^{-1}}$ and (ii) projections onto two constraint sets that are 'spheres' in half-length subvector spaces with a special metric.

It is well-known that room impulse responses decay exponentially *on average* under the so-called diffused sound field assumption [69]. Moreover, it has experimentally been shown in [77, 78] that averages of an impulse response and its variations in a room are well-modeled by exponentially decaying shapes with the same exponential ratio; this ratio can be measured in advance since it is almost invariable under fixed acoustic conditions of the room (e.g., size, absorption coefficient etc.). However, real impulse responses do *not exactly* decay exponentially, while there *almost always* exists a notable difference between the first and second halves of impulse responses. Thus the following simple matrices can be good alternatives of the matrix $\boldsymbol{A}$ (see Proposition 6.2.4).

Example 6.3.1 (Simple metric design)**.** *For notational simplicity, we assume that the filter length N is divisible by 2 (or 4). Simple metrics are then defined by the following N-by-N diagonal matrices:*

$$\begin{aligned}\boldsymbol{B} &:= \mathrm{diag}\left(\boldsymbol{I}_{N/2}, \gamma^{N/2}\boldsymbol{I}_{N/2}\right) \succ 0,\\ \boldsymbol{C} &:= \mathrm{diag}\left(\boldsymbol{I}_{N/4}, \gamma^{N/4}\boldsymbol{I}_{N/4}, \gamma^{N/2}\boldsymbol{I}_{N/4}, \gamma^{3N/4}\boldsymbol{I}_{N/4}\right) \succ 0.\end{aligned}$$

Here $\boldsymbol{I}_n$ denotes the n-by-n identity matrix, and $\gamma \in (0, 1]$ is the exponential ratio introduced in Proposition 6.2.4 *(For the design of γ, see Sec.* 6.4.2*).*

The simple structures of $\boldsymbol{B}$ and $\boldsymbol{C}$ bring an advantage in robustness against model-mismatch as well as computational complexity (Note that this type of discrete steps has also been presented in [77, §3.3] as practical modification of the matrix $\boldsymbol{A}$ for multiple DSP implementation in order mainly to reduce the computational costs). For instance, the computation of $\boldsymbol{a}^T\boldsymbol{B}\boldsymbol{a}$ requires $N + 1$ multiplications. In other words, only one extra multiplication, compared with $\|\boldsymbol{a}\|^2$, is required. This computational efficiency comes from the following relation: $\boldsymbol{a}^T\boldsymbol{B}\boldsymbol{a} = \boldsymbol{a}_{(\mathrm{e})}^T\boldsymbol{a}_{(\mathrm{e})} + \gamma^{N/2}\boldsymbol{a}_{(\mathrm{t})}^T\boldsymbol{a}_{(\mathrm{t})}$, $\forall \boldsymbol{a} = [\boldsymbol{a}_{(\mathrm{e})}^T, \boldsymbol{a}_{(\mathrm{t})}^T]^T \in \mathcal{H}$.

The next step is to define data-dependent closed convex sets that contain the estimandum $\boldsymbol{h}^*$ with high reliability. We define the stochastic property set as follows [129]:

$$C_k := C_k(\rho) := \left\{\boldsymbol{h} \in \mathcal{H} : g_k(\boldsymbol{h}) := \|\boldsymbol{e}_k(\boldsymbol{h})\|_{\boldsymbol{I}}^2 - \rho \le 0\right\},\ \forall k \in \mathbb{N},$$

where $\boldsymbol{e}_k : \mathcal{H} \to \mathbb{R}^r$, $\boldsymbol{h} \mapsto \boldsymbol{U}_k^T\boldsymbol{h} - \boldsymbol{d}_k$, is the error (or residual) function and $\rho \ge 0$

a parameter governing the membership probability that $\boldsymbol{h}^* \in C_k(\rho)$ (For unified notation, we use $\|\cdot\|_{\boldsymbol{I}}$ rather than the common notation $\|\cdot\|_2$). Noise statistics are involved in the design of ρ [129, Ex. 1]. Since the computational cost for the direct projection onto $C_k(\rho)$ is prohibitive in general, we introduce an outer-approximating half-space $H_k^-(\boldsymbol{h})$ s.t. $P^{(\boldsymbol{Q})}_{H_k^-(\boldsymbol{h})}(\boldsymbol{h}) \cong P^{(\boldsymbol{Q})}_{C_k(\rho)}(\boldsymbol{h})$. Simple design of such a half-space is given below.

Example 6.3.2 (Design of outer-approximating half-space).

(a) *(Standard subgradient outer approximation) A commonly used half-spaces* [129, 134, 135, 138] *is* $H_k^-(\boldsymbol{h}) := \Pi_{k,1}^-(\boldsymbol{h}) := \{\boldsymbol{x} \in \mathcal{H} : \langle \boldsymbol{x} - \boldsymbol{h}, \nabla_{(\boldsymbol{Q})} g_k(\boldsymbol{h}) \rangle_{\boldsymbol{Q}} + g_k(\boldsymbol{h}) \leq 0\} \supset C_k(\rho)$, *where* $\partial_{(\boldsymbol{Q})} g_k(\boldsymbol{h}) \ni \nabla_{(\boldsymbol{Q})} g_k(\boldsymbol{h}) = 2\boldsymbol{Q}^{-1}\boldsymbol{U}_k \boldsymbol{e}_k(\boldsymbol{h})$ *(see the definition of subdifferential in Chapter 2). Note that* $\partial_{(\boldsymbol{Q})} g_k(\boldsymbol{h}) = \{\nabla_{(\boldsymbol{Q})} g_k(\boldsymbol{h})\}$ *in this (differentiable) case. We stress now that we are considering* $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\boldsymbol{Q}})$. *The projection onto* $\Pi_{k,1}^-(\boldsymbol{h})$ *has the following simple closed-form expression:*

$$P^{(\boldsymbol{Q})}_{\Pi_{k,1}^-(\boldsymbol{h})}(\boldsymbol{h}) = \begin{cases} \boldsymbol{h} - \dfrac{g_k(\boldsymbol{h})}{\left\| \nabla_{(\boldsymbol{Q})} g_k(\boldsymbol{h}) \right\|_{\boldsymbol{Q}}^2} \nabla_{(\boldsymbol{Q})} g_k(\boldsymbol{h}) & \text{if } \boldsymbol{h} \notin \Pi_{k,1}^-(\boldsymbol{h}), \\ \boldsymbol{h} & \text{otherwise.} \end{cases} \tag{6.3.1}$$

The computation of $P^{(\boldsymbol{Q})}_{\Pi_{k,1}^-(\boldsymbol{h})}(\boldsymbol{h})$ *requires only* $O(N)$ *complexity.*

(b) *(Deep outer approximation* [87]*) For* $\xi \in [-\rho, \inf_{\boldsymbol{y} \in \mathcal{H}} g_k(\boldsymbol{y})]$, *define a closed half-space* $H_k^-(\boldsymbol{h}) := \Pi_{k,2}^-(\xi, \boldsymbol{h})$ *as*

$$\Pi_{k,2}^-(\xi, \boldsymbol{h}) := \begin{cases} \Big\{ \boldsymbol{x} \in \mathcal{H} : \langle \boldsymbol{x} - \boldsymbol{h}, \nabla_{(\boldsymbol{Q})} g_k(\boldsymbol{h}) \rangle_{\boldsymbol{Q}} \\ \qquad +2 \left[g_k(\boldsymbol{h}) - \xi - \sqrt{(g_k(\boldsymbol{h}) - \xi)(-\xi)} \right] \leq 0 \Big\} & \text{if } \boldsymbol{h} \notin C_k(\rho), \\ \mathcal{H} & \text{otherwise.} \end{cases}$$

Practically, it is reasonable to use $\Pi_{k,2}^-(-\rho, \boldsymbol{h})$ *(see Remark* 6.3.3-(b) *below).*

The projection onto $\Pi^{-}_{k,2}(-\rho, \boldsymbol{h})$ *has the following simple closed-form expression:*

$$P^{(\boldsymbol{Q})}_{\Pi^{-}_{k,2}(-\rho,\boldsymbol{h})}(\boldsymbol{h}) = \begin{cases} \boldsymbol{h} - \dfrac{2\left[g_k(\boldsymbol{h}) + \rho - \sqrt{\rho[g_k(\boldsymbol{h}) + \rho]}\right]}{\left\|\nabla_{(\boldsymbol{Q})} g_k(\boldsymbol{h})\right\|^2_{\boldsymbol{Q}}} \nabla_{(\boldsymbol{Q})} g_k(\boldsymbol{h}) & \text{if } \boldsymbol{h} \notin \Pi^{-}_{k,2}(-\rho, \boldsymbol{h}), \\ \boldsymbol{h} & \text{otherwise.} \end{cases} \tag{6.3.2}$$

Remark 6.3.3 (On subgradient outer approximation in Example 6.3.2).

(a) *The use of* $\Pi^{-}_{k,1}(\boldsymbol{h})$ *for* $r = 1$, $\rho = 0$, *and* $\boldsymbol{Q} = \boldsymbol{A}^{-1}$ *reproduces ESP with the step size within* $[0, 1]$, *while the use of* $\Pi^{-}_{k,2}(0, \boldsymbol{h})$ *for* $r = 1$ *and* $\boldsymbol{Q} = \boldsymbol{A}^{-1}$ *reproduces ESP with the step size within* $[0, 2]$.

(b) *The closed half-spaces presented in Example* 6.3.2 *satisfy (i)* $C_k(\rho) \subset H^{-}_k(\boldsymbol{h}_k)$, *and (ii)* $\boldsymbol{h}_k \notin C_k(\rho) \Rightarrow \boldsymbol{h}_k \notin H^{-}_k(\boldsymbol{h}_k)$ [129, Lemma 2], [87, Theorem 4], *which implies that the boundary hyperplane of* $H^{-}_k(\boldsymbol{h}_k)$ *separates* $C_k(\rho)$ *from* $\boldsymbol{h}_k$ *if* $\boldsymbol{h}_k \notin C_k(\rho)$.

(c) *If* $\boldsymbol{h} \notin C_k(\rho)$ *(*$k \in \mathbb{N}$*), then* $\Pi^{-}_{k,2}(\xi, \boldsymbol{h}) \subsetneq \Pi^{-}_{k,1}(\boldsymbol{h})$, $\forall \xi \in [\rho, \inf_{\boldsymbol{y}\in\mathcal{H}} g_k(\boldsymbol{y})]$. *If in addition* $\xi^{(k)}_{\min} := \min_{\boldsymbol{y}\in\mathcal{H}} g_k(\boldsymbol{y})$ *exists, then* $\Pi^{-}_{k,2}(\xi^{(k)}_{\min}, \boldsymbol{h}) \cap C_k(\rho) \neq \emptyset$ [87, Theorem 5]. *This means with Remark* 6.3.3-(a) *that the boundary hyperplane of* $\Pi^{-}_{k,2}(\xi^{(k)}_{\min}, \boldsymbol{h})$ *supports* $C_k(\rho)$, *thus* $\Pi^{-}_{k,2}(\xi^{(k)}_{\min}, \boldsymbol{h})$ *is a tightest outer approximation among all half-spaces obtained by translating the standard subgradient outer approximation* $\Pi^{-}_{k,1}(\boldsymbol{h})$. *Since it is often the case that* $N \gg r$, *we can mostly assume* $\xi^{(k)}_{\min} = -\rho$. *Note that even if* $\inf_{\boldsymbol{y}\in\mathcal{H}} g_k(\boldsymbol{y}) > -\rho$, *the half-space* $\Pi^{-}_{k,2}(-\rho, \boldsymbol{h})$ *is always a proper subset of the standard subgradient outer approximation* $\Pi^{-}_{k,1}(\boldsymbol{h})$.

The final step is to design the strongly attracting nonexpansive mapping $T : \mathcal{H} \to \mathcal{H}$. We define $T := P^{(\boldsymbol{Q})}_{K_{\mathrm{e}}} P^{(\boldsymbol{Q})}_{K_{\mathrm{t}}}$ with the constraint sets K_{e} and K_{t} defined as below [Note: In this case, T is 1/2-attracting nonexpansive with $\mathrm{Fix}(T) = K_{\mathrm{e}} \cap K_{\mathrm{t}}$ (see Example 6.2.7-(b))].

Example 6.3.4 (Design of constraint sets). *We denote the early and tail parts (i.e., the first and second halves) of any vector* $\boldsymbol{x} \in \mathcal{H}$ *as* $\boldsymbol{x}_{(\mathrm{e})} \in \mathbb{R}^{N/2}$ *and* $\boldsymbol{x}_{(\mathrm{t})} \in$

$\mathbb{R}^{N/2}$, *respectively; i.e.,* $\boldsymbol{x} =: [\boldsymbol{x}_{(\mathrm{e})}^T \boldsymbol{x}_{(\mathrm{t})}^T]^T$. *We then introduce the following two constraint sets that respectively restrict the energy of early and tail parts of* $\boldsymbol{h}_k$*:*

$$K_{\mathrm{e}} := \left\{ \begin{bmatrix} \boldsymbol{h}_{(\mathrm{e})} \\ \boldsymbol{h}_{(\mathrm{t})} \end{bmatrix} \in \mathcal{H} : \left\| \begin{bmatrix} \boldsymbol{h}_{(\mathrm{e})} \\ \boldsymbol{0} \end{bmatrix} \right\|_{\boldsymbol{Q}}^2 \leq \varepsilon_{\mathrm{e}} \right\},$$
$$K_{\mathrm{t}} := \left\{ \begin{bmatrix} \boldsymbol{h}_{(\mathrm{e})} \\ \boldsymbol{h}_{(\mathrm{t})} \end{bmatrix} \in \mathcal{H} : \left\| \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{h}_{(\mathrm{t})} \end{bmatrix} \right\|_{\boldsymbol{Q}}^2 \leq \varepsilon_{\mathrm{t}} \right\}.$$

Here $\varepsilon_{\mathrm{e}}, \varepsilon_{\mathrm{t}} > 0$ *should be designed by using an estimate of* α_0 *and the exponential ratio* γ *(see Proposition* 6.2.4*).*

For an intuitive understanding, we define

$$\widehat{K}_{\mathrm{e}} := \left\{ \boldsymbol{h}_{(\mathrm{e})} \in \mathbb{R}^{N/2} : \left\| \boldsymbol{h}_{(\mathrm{e})} \right\|_{\boldsymbol{Q}_{\mathrm{u.l.}}}^2 = \boldsymbol{h}_{(\mathrm{e})}^T \boldsymbol{Q}_{\mathrm{u.l.}} \boldsymbol{h}_{(\mathrm{e})} \leq \varepsilon_{\mathrm{e}} \right\},$$
$$\boldsymbol{Q}_{\mathrm{u.l.}} \in \mathbb{R}^{N/2 \times N/2} : \text{ upper-left submatrix of } \boldsymbol{Q}.$$

Noting that no constraint is imposed on $\boldsymbol{h}_{(\mathrm{t})}$ in $K_{\mathrm{e}} (\subset \mathcal{H} = \mathbb{R}^N)$, K_{e} imposes the same constraint as $\widehat{K}_{\mathrm{e}} (\subset \mathbb{R}^{N/2})$ on $\boldsymbol{h}_{(\mathrm{e})}$. Since $\boldsymbol{Q} \succ 0$ implies $\boldsymbol{Q}_{\mathrm{u.l.}} \succ 0$, it is seen that $\widehat{K}_{\mathrm{e}}$ is an ellipsoid in general while it is a sphere with the special metric $\mathrm{d}_{\boldsymbol{Q}_{\mathrm{u.l.}}}$. Suppose, for example, that the metric $\mathrm{d}_{\boldsymbol{C}^{-1}}$ is employed. In this case, with the Euclidean metric, the 'ellipsoid' has a larger radius in the first half of $\boldsymbol{h}_{(\mathrm{e})}$ than the second half, which agrees with the decaying feature of echo impulse responses. Remind now that we are considering the real Hilbert space $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\boldsymbol{Q}})$. Since $\widehat{K}_{\mathrm{e}}$ has a simple 'sphere' structure with the special metric, the projection onto K_{e} is simply given as follows:

$$\forall \boldsymbol{h} = \begin{bmatrix} \boldsymbol{h}_{(\mathrm{e})} \\ \boldsymbol{h}_{(\mathrm{t})} \end{bmatrix} \in \mathcal{H}, \; P_{K_{\mathrm{e}}}^{(\boldsymbol{Q})}(\boldsymbol{h}) = \begin{cases} \begin{bmatrix} \dfrac{\sqrt{\varepsilon_{\mathrm{e}}}}{\phi_{\mathrm{e}}(\boldsymbol{h})} \boldsymbol{h}_{(\mathrm{e})} \\ \boldsymbol{h}_{(\mathrm{t})} \end{bmatrix} & \text{if } \boldsymbol{h} \notin K_{\mathrm{e}}, \\ \boldsymbol{h} & \text{otherwise}, \end{cases}$$

where $\phi_{\mathrm{e}}(\boldsymbol{h}) := \left\| [\boldsymbol{h}_{(\mathrm{e})}^T \boldsymbol{0}^T]^T \right\|_{\boldsymbol{Q}}$. Similarly, we can obtain the projection onto K_{t} as

follows:

$$\forall \boldsymbol{h} = \begin{bmatrix} \boldsymbol{h}_{(\mathrm{e})} \\ \boldsymbol{h}_{(\mathrm{t})} \end{bmatrix} \in \mathcal{H},\ P^{(\boldsymbol{Q})}_{K_{\mathrm{t}}}(\boldsymbol{h}) = \begin{cases} \begin{bmatrix} \boldsymbol{h}_{(\mathrm{e})} \\ \frac{\sqrt{\varepsilon_{\mathrm{t}}}}{\phi_{\mathrm{t}}(\boldsymbol{h})}\boldsymbol{h}_{(\mathrm{t})} \end{bmatrix} & \text{if } \boldsymbol{h} \notin K_{\mathrm{t}}, \\ \boldsymbol{h} & \text{otherwise,} \end{cases}$$

where $\phi_{\mathrm{t}}(\boldsymbol{h}) := \left\| [\boldsymbol{0}^T \boldsymbol{h}_{(\mathrm{t})}^T]^T \right\|_{\boldsymbol{Q}}$.

Given $q \in \mathbb{N}^*$, define the control sequence $\mathcal{I}_k := \left\{ \iota_1^{(k)}, \iota_2^{(k)}, \cdots, \iota_q^{(k)} \right\}$, $\forall k \in \mathbb{N}$. The control sequence indicates the closed half-spaces to be processed at time k. The proposed AEC algorithm is given as follows.

Algorithm 6.3.5 (Adaptive Quadratic-Metric Parallel Subgradient Projection (AQ-PSP) Algorithm). *For an arbitrary initial echo canceler $\boldsymbol{h}_0 \in \mathcal{H}$, generate a sequence of echo canceling vectors $(\boldsymbol{h}_k)_{k\in\mathbb{N}} \subset \mathcal{H}$ as*

$$\boldsymbol{h}_{k+1} := P^{(\boldsymbol{Q})}_{K_{\mathrm{e}}} P^{(\boldsymbol{Q})}_{K_{\mathrm{t}}} \left\{ \boldsymbol{h}_k + \lambda_k \mathcal{M}_k \left[\sum_{\iota \in \mathcal{I}_k} w_\iota^{(k)} P^{(\boldsymbol{Q})}_{H_\iota^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right] \right\},\ \forall k \in \mathbb{N},$$

where $\lambda_k \in [0, 2]$ is the step size and

$$\mathcal{M}_k := \begin{cases} \dfrac{\sum_{\iota \in \mathcal{I}_k} w_\iota^{(k)} \left\| P^{(\boldsymbol{Q})}_{H_\iota^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\|_{\boldsymbol{Q}}^2}{\left\| \sum_{\iota \in \mathcal{I}_k} w_\iota^{(k)} P^{(\boldsymbol{Q})}_{H_\iota^-(\boldsymbol{h}_k)}(\boldsymbol{h}_k) - \boldsymbol{h}_k \right\|_{\boldsymbol{Q}}^2} & \text{if } \boldsymbol{h}_k \notin \bigcap_{\iota \in \mathcal{I}_k} H_\iota^-(\boldsymbol{h}_k), \\ 1 & \text{otherwise.} \end{cases}$$

For the design of the half-spaces $H_\iota^-(\boldsymbol{h}_k)$, see Example 6.3.2. Figure 6.3.3 illustrates the behavior of AQ-PSP, where we set $q = 3$ and $\lambda_k \approx 1$, $\forall k \in \mathbb{N}$, and omit the constraint sets K_{e} and K_{t} for visual clarity. The three dotted-arrows express the projections, with an 'appropriate' metric $\mathrm{d}_{\boldsymbol{Q}}$, of $\boldsymbol{h}_k$ onto H_k^-, H_{k-1}^-, and H_{k-2}^-, respectively. The weights $w_\iota^{(k)}$ finally determine the direction of update by taking a point from the shaded-triangle, and the update vector is scaled by λ_k and $\mathcal{M}_k$ (For an efficient design of the weights, see Chapter 4).

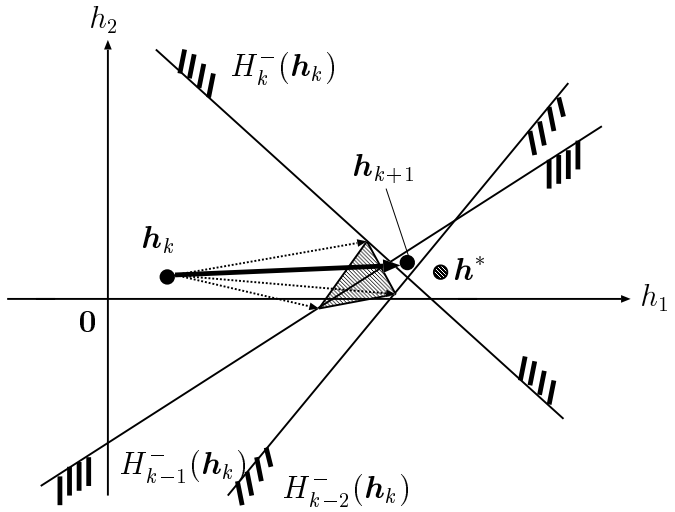


Figure 6.3.3: A geometric interpretation of the proposed APQP algorithm for $q = 3$.

6.4 Numerical Examples

6.4.1 Effects of Different Metrics for Two Extreme Impulse Responses

We examine the effects of different metrics $\mathrm{d}_{\boldsymbol{A}^{-1}}$, $\mathrm{d}_{\boldsymbol{B}^{-1}}$, $\mathrm{d}_{\boldsymbol{I}}$, and $\mathrm{d}_{\boldsymbol{G}_k^{-1}}$ (see Propositions 6.2.4 and 6.2.12, and Example 6.3.1) in simple system identification problems for $N = 256$. The USASI signal (see Sec. 4.5) is used for the input $\boldsymbol{u}_k$. The noise $\boldsymbol{n}_k$ is white with Signal to Noise Ratio (SNR) $:= 10\log_{10}(E\{z_k^2\}/E\{n_k^2\}) = 10$ dB, where $z_k := \boldsymbol{u}_k^T\boldsymbol{h}^*$. We evaluate the achievement level of identification by the system mismatch defined as $10\log_{10}(\|\boldsymbol{h}^* - \boldsymbol{h}_k\|_{\boldsymbol{I}}^2/\|\boldsymbol{h}^*\|_{\boldsymbol{I}}^2)$ at kth iteration. For simplicity, we employ the NLMS-based algorithms: ESP for $r = 1$ and PNLMS.

In the first simulation, we model the estimandum $\boldsymbol{h}^*(=: [h_1^*, h_2^*, \cdots, h_N^*]^T)$ as $h_i^* = 0.4\gamma^{i-1}$, $\forall i = 1, 2, \cdots, N$, with $\gamma = 0.95775$. For ESP, we set (I) $\lambda_k = 0.03$, $\forall k \in \mathbb{N}$, with $\mathrm{d}_{\boldsymbol{A}^{-1}}$; (II) $\lambda_k = 0.1$, $\forall k \in \mathbb{N}$, with $\mathrm{d}_{\boldsymbol{B}^{-1}}$; and (III) $\lambda_k = 0.2$, $\forall k \in \mathbb{N}$, with $\mathrm{d}_{\boldsymbol{I}}$ (which is nothing but NLMS). For ESP-(I) and ESP-(II), the true value of γ is employed in $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively. For PNLMS, we set $\lambda_k = 0.1, \forall k \in \mathbb{N}$, with $\mathrm{d}_{\boldsymbol{G}_k^{-1}}$ proposed in [33, 41] for $\sigma = 5/N$ and $\delta = \min\{0.01, 0.0001/\sigma\}$ by following the recommendation in [33, 41] (see Proposition 6.2.12). The step size for each algorithm is tuned so that all the algorithms attain performance comparable with

each other in initial convergence speed. The results are drawn in Fig. 6.4.1. As expected, ESP-(I) exhibits the best steady-state performance among the employed algorithms.

In the second simulation, we model $\boldsymbol{h}^*$ as $h_i^* = 0.4$, $\forall i = 1, \cdots, N/2$, and $h_i^* = 0.4\gamma^{N/2}$, $\forall i = N/2+1, \cdots, N$, with $\gamma = 0.95775$. For ESP, we set (I) $\lambda_k = 0.5$, $\forall k \in \mathbb{N}$, with $\mathrm{d}_{\boldsymbol{A}^{-1}}$; (II) $\lambda_k = 0.05$, $\forall k \in \mathbb{N}$, with $\mathrm{d}_{\boldsymbol{B}^{-1}}$; and (III) $\lambda_k = 0.2$, $\forall k \in \mathbb{N}$, with $\mathrm{d}_{\boldsymbol{I}}$ (the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are the same as in the first simulation). For PNLMS, we set $\lambda_k = 0.2$, $\forall k \in \mathbb{N}$, with the same metric $\mathrm{d}_{\boldsymbol{G}_k^{-1}}$ as in the first simulation. The results are drawn in Fig. 6.4.1. Contrary to the results in Fig. 6.4.1, due to the special structure of the estimandum $\boldsymbol{h}^*$, ESP-(I) exhibits the worst steady-state performance among the employed algorithms, while ESP-(II) exhibits the best steady-state performance since the metric $\mathrm{d}_{\boldsymbol{B}^{-1}}$ is fit for the structure.

The results of those two simulations suggest that, if the estimandum has a special structure, an appropriate design of metric can drastically improve the performance of adaptive filter. Next we verify the efficacy of the proposed algorithms in the AEC problem.

6.4.2 Evaluation of the Proposed, ESP, and PAPA Algorithms in the Acoustic Echo Cancellation Problem

We evaluate the performance of AQ-PSP (Algorithm 6.3.5) and its variable-metric version named *Adaptive Variable-metric PSP (AV-PSP)* compared with ESP and PAPA in the AEC problem under the following conditions (Note that AV-PSP is a realization of Algorithm 6.2.10). The input signal $\boldsymbol{u}_k$ is English-native-male's speech recorded at sampling rate 8 kHz [see Fig. 6.4.2]. The noise $\boldsymbol{n}_k$ is white with SNR = 10 dB. A real impulse response[4] $\boldsymbol{h}^* \in \mathbb{R}^N$ recorded in a small room [see Fig. 6.4.2] is used with $N = 1024$. Although in this chapter we concentrate on the case in which the length of $\boldsymbol{h}^*$ coincides with N, a case in which the length of $\boldsymbol{h}^*$ is four times greater than N has been investigated in simulations in [141].

[4]The impulse response is available at http://www.echochamber.ch/responses/960/rooms.zip; the name of the file is "1960small_room.wav". Although the frequency of the original is in fact 44.1 kHz, we convert it by the matlab command 'resample'.

Table 6.4.1: Steady-state performance of the proposed algorithms for $q = 8$ in ERLE and system mismatch.

Algorithm	AV-PSP	AQ-PSP $[\boldsymbol{A}, \gamma(0.2)]$	AQ-PSP $[\boldsymbol{A}, \gamma(0.4)]$	AQ-PSP $[\boldsymbol{B}, \gamma(0.2)]$	AQ-PSP $[\boldsymbol{B}, \gamma(0.4)]$	AQ-PSP $[\boldsymbol{I}]$
ERLE [dB]	20.5	19.3	17.5	17.3	16.5	15.2
System Mismatch [dB]	-17.3	-16.8	-15.9	-15.2	-14.9	-13.8

To measure the level of echo cancellation as well as that of identification of echo impulse response, we adopt, in addition to system mismatch, the Echo Return Loss Enhancement (ERLE) [15]. To obtain smooth ERLE curves, after calculating instantaneous ERLE at kth iteration as $\mathrm{ERLE}^{(0)}_{\mathrm{tmp}}(k) := 10\log_{10}[z_k^2/(z_k - \boldsymbol{u}_k^T\boldsymbol{h}_k)^2]$, we pass the vector $\mathrm{ERLE}^{(0)}_{\mathrm{tmp}}$ through a smoothing filter three times. Namely, $\mathrm{ERLE}(k) := \mathrm{ERLE}^{(3)}_{\mathrm{tmp}}(k)$ with $\mathrm{ERLE}^{(i+1)}_{\mathrm{tmp}}(k) := \frac{1}{k+\ell-\psi(k-\ell)+1}\left(\sum_{j=\psi(k-\ell)}^{k+\ell}\mathrm{ERLE}^{(i)}_{\mathrm{tmp}}(j)\right)$, for $i = 0, 1, 2$, where $\ell = 5000$ and $\psi(n) := \max\{n, 0\}$ for any integer n. For numerical stability against poor excitation of the speech input signals, we use certain regularization and threshold for all the algorithms. We remark here that the step size for each algorithm is selected in each experiment so that all the algorithms attain performance comparable with each other in initial convergence speed. Discussion about the results is separately given in Sec. 6.4.3.

A. AV-PSP versus AQ-PSP

Figure 6.4.6 draws a comparison among the proposed algorithms: (I) AV-PSP, (II) AQ-PSP for $\boldsymbol{Q} = \boldsymbol{A}^{-1}$, (III) AQ-PSP for $\boldsymbol{Q} = \boldsymbol{B}^{-1}$, and (IV) AQ-PSP for $\boldsymbol{Q} = \boldsymbol{I}$ (which is nothing but adaptive-PSP [129]). It is known that, under the diffuse sound field assumption, the ensemble average E_n $(n = 1, 2, \cdots, N)$ of the squared room impulse responses decays exponentially (see [26]); i.e., $E_n := E\{(h_n^*)^2\} = E_1 \exp[-n\log 10^6/(T_{60}F_s)]$. Here F_s [Hz] is the sampling frequency and T_{60} [sec.] is the time interval in which the reverberant sound energy drops down by 60 dB [69]. Hence, for a given estimate of T_{60}, say $\widehat{T}_{60}$, the exponential factor γ should be designed as $\gamma = \gamma(\widehat{T}_{60}) := \exp\left[-\log 10^6/(2\widehat{T}_{60}F_s)\right]$. In Fig. 6.4.6, $\gamma(0.2) = 0.99569$ is employed for Proposed (II) and (III).

For all algorithms, we set $r = 1$, $q = 8$, $\rho = \rho_3(= 0)$ (ρ_3: the peak value of

Table 6.4.2: Steady-state performance of the PAPA and ESP algorithms for $r = 2$ in ERLE and system mismatch.

Algorithm	PAPA	ESP $[\boldsymbol{A}, \gamma(0.2)]$	ESP $[\boldsymbol{B}, \gamma(0.2)]$	ESP $[\boldsymbol{I}]$
ERLE [dB]	18.0	16.0	14.0	12.4
System Mismatch [dB]	-14.3	-13.2	-11.2	-9.9

Table 6.4.3: Steady-state performance of the PAPA and ESP algorithms for $r = 8$ in ERLE and system mismatch.

Algorithm	PAPA	ESP $[\boldsymbol{A}, \gamma(0.2)]$	ESP $[\boldsymbol{B}, \gamma(0.2)]$	ESP $[\boldsymbol{I}]$
ERLE [dB]	18.6	17.1	14.4	12.6
System Mismatch [dB]	-15.0	-13.3	-10.6	-9.0

the probability density function of the random variable $\xi := \|\boldsymbol{n}_k\|_{\boldsymbol{I}}^2$ [129]), and $w_i^{(k)} = 1/q$, $\forall i = 1, 2, \cdots, q$, $\forall k \in \mathbb{N}$. For the outer-approximating half-space $H_k^-(\boldsymbol{h})$, moreover, we employ $\Pi_{k,1}^-(\boldsymbol{h})$ in Example 6.3.2-(a) [Note that, since we employ $r = 1$ and $\rho = 0$, the use of $\Pi_{k,1}^-(\boldsymbol{h})$ with λ_k gives the same results as the use of $\Pi_{k,2}^-(\boldsymbol{h})$ with $\lambda_k/2$; cf. Remark 6.3.3-(a)]. For AV-PSP [Proposed-(I)], we set $\lambda_k = 0.4$, $\forall k \in \mathbb{N}$, with $\mathrm{d}_{\boldsymbol{G}_k^{-1}}$ designed in the same way as PNLMS in Sec. 6.4.1. For AQ-PSP, we set $\lambda_k = 0.4$, $\forall k \in \mathbb{N}$, for Proposed-(II); $\lambda_k = 1.0$, $\forall k \in \mathbb{N}$, for Proposed-(III); and $\lambda_k = 1.0$, $\forall k \in \mathbb{N}$, for Proposed-(IV). To examine the pure effect of the newly introduced metric, we omit the projections onto the constraint sets K_{e} and K_{t}, which is equivalent to assigning very large values to ε_{e} and ε_{t}.

Figure 6.4.7 draws the estimation $\widehat{T}_{60}$ versus the steady-state performance for the AQ-PSP based algorithms. The parameters employed in each algorithm are exactly the same as in Fig. 6.4.6 except for $\gamma(\widehat{T}_{60})$. ERLE and system mismatch in Fig. 6.4.7 and Tables 6.4.1–6.4.3 express the values averaged uniformly over the last 3.2×10^5 and 1.0×10^5 samples (40 sec. and 12.5 sec.), respectively. We observe that the best performance of AQ-PSP for both $\boldsymbol{Q} = \boldsymbol{A}^{-1}$ and $\boldsymbol{Q} = \boldsymbol{B}^{-1}$ is achieved by $\widehat{T}_{60} = 0.2$. The exponential curves in Fig. 6.4.2 draw the diagonal elements of $\boldsymbol{A}$ for $\widehat{T}_{60} = 0.2$ and 0.4, respectively. It is seen that $\gamma(0.2) = 0.99569$ is more likely than $\gamma(0.4) = 0.99784$ for the envelope of the impulse response, which agrees with the above observation. Note by the definition of $\gamma(\widehat{T}_{60})$ that $\gamma(\widehat{T}_{60}) \in (0, 1] \Leftrightarrow \widehat{T}_{60} \in (0, 62.5]$. Since γ increases up to 1 as $\widehat{T}_{60}$ increases within (0,62.5], it is expected that the performance of AQ-PSP for $\boldsymbol{Q} = \boldsymbol{A}^{-1}$ and

$\boldsymbol{Q} = \boldsymbol{B}^{-1}$ approaches the level achieved by AQ-PSP with $\boldsymbol{Q} = \boldsymbol{I}$, which is verified by the results in Fig. 6.4.7. Table 6.4.1 shows the steady-state performance of the proposed algorithms employed in Figs. 6.4.6 and 6.4.7.

B. PAPA versus ESP

Figures 6.4.8 and 6.4.9 draw comparisons of ESP [77] and PAPA [10, 48, 64, 120] for $r = 2$ and $r = 8$, respectively. In Fig. 6.4.8, we set, for ESP, (I) $\lambda_k = 0.1$, $\forall k \in \mathbb{N}$, with $\boldsymbol{A}$ and $\gamma(0.2)$; (II) $\lambda_k = 0.2$, $\forall k \in \mathbb{N}$, with $\boldsymbol{B}$ and $\gamma(0.2)$; and (III) $\lambda_k = 0.2$, $\forall k \in \mathbb{N}$, with $\boldsymbol{I}$ (which is nothing but APA), and, for PAPA, $\lambda_k = 0.2$, $\forall k \in \mathbb{N}$, with $\boldsymbol{G}_k$ the same as in Sec. 6.4.1. In Fig. 6.4.9, we set, for ESP, (I) $\lambda_k = 0.02$, $\forall k \in \mathbb{N}$; (II) $\lambda_k = 0.05$, $\forall k \in \mathbb{N}$; and (III) $\lambda_k = 0.05$, $\forall k \in \mathbb{N}$, and, for PAPA, $\lambda_k = 0.05$, $\forall k \in \mathbb{N}$ (the matrices employed are exactly the same as in Fig. 6.4.8). Tables 6.4.2 and 6.4.3 show the steady-state performance of ESP and PAPA corresponding to Figs. 6.4.8 and 6.4.9, respectively.

C. Proposed versus ESP and PAPA

Figure 6.4.10 compares best performance, in our extensive simulations, of the AV-PSP, AQ-PSP, PAPA, and ESP algorithms, respectively. AV-PSP is Proposed-(I) in Fig. 6.4.6; AQ-PSP is Proposed-(II) in Fig. 6.4.6; PAPA is the one in Fig. 6.4.9 ($r = 8$); and ESP is ESP-(I) in Fig. 6.4.9 ($r = 8$). It is seen that the proposed algorithms significantly outperform the other methods both in ERLE and system mismatch.

6.4.3 Discussion

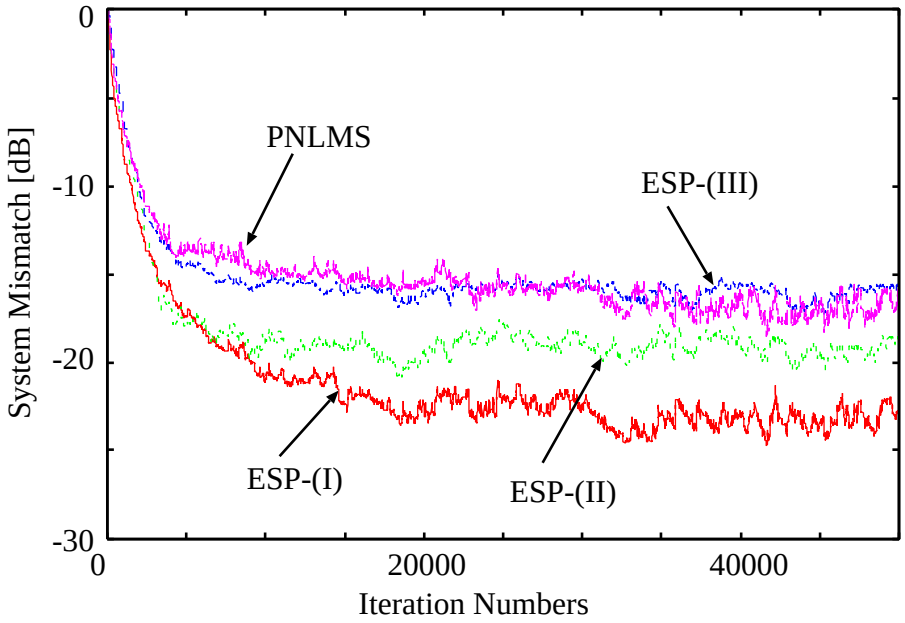
We discuss the steady-state performance according to the results obtained in Sec. 6.4.2. Table 6.4.1 shows that AV-PSP gains, compared with adaptive-PSP (i.e., AQ-PSP with $\boldsymbol{I}$), more than 5 dB in ERLE and 3.5 dB in system mismatch at the cost of $2N$ extra multiplications and computation to design $\boldsymbol{G}_k$. It also shows that AQ-PSP with $\boldsymbol{A}$ for an appropriate $\gamma(0.2)$ gives close performance to AV-PSP without the cost of computation to design $\boldsymbol{G}_k$, while for $\gamma(0.4)$ its performance degrades by 1.8 dB in ERLE compared with the one for $\gamma(0.2)$. Moreover, it is seen that AQ-PSP with $\boldsymbol{B}$ for $\gamma(0.2)$ gains, compared with adaptive-PSP, more than 2 dB in ERLE, while for $\gamma(0.4)$ its performance degrades by 0.8 dB,

which is relatively small compared with AQ-PSP with $\boldsymbol{A}$. This observation and Fig. 6.4.7 imply that the use of the metric $\mathrm{d}_{\boldsymbol{B}^{-1}}$ is more robust than $\mathrm{d}_{\boldsymbol{A}^{-1}}$ against mismatch in the estimate of exponential factor γ. The above arguments suggest that $\mathrm{d}_{\boldsymbol{A}^{-1}}$ or $\mathrm{d}_{\boldsymbol{B}^{-1}}$ should be a reasonable choice in relatively static environments (e.g., in a conference room), in which it is easy to obtain a good estimate of γ, while $\mathrm{d}_{\boldsymbol{G}_k^{-1}}$ or $\mathrm{d}_{\boldsymbol{B}^{-1}}$ should be reasonable in dynamic environments (e.g., in mobile applications); see also [64]. It should be remarked that the use of $\mathrm{d}_{\boldsymbol{B}^{-1}}$ attains significant improvements with almost the same computational complexity as the Euclidean metric $\mathrm{d}_{\boldsymbol{I}}$. Similar relations are also seen in Tables 6.4.2 and 6.4.3 for ESP and PAPA. Finally, Fig. 6.4.10 with the aid of Tables 6.4.1 and 6.4.3 shows that AV-PSP gains (i) 1.9 dB in ERLE and 2.3 dB in system mismatch compared with PAPA for $r = 8$, and (ii) 3.4 dB in ERLE and 4 dB compared with ESP for $r = 8$.

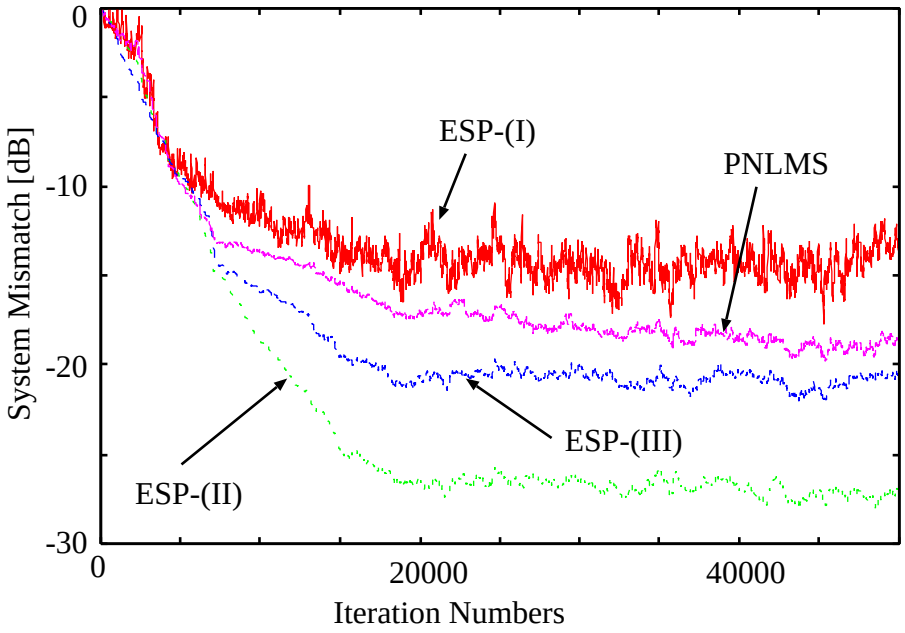
6.5 Conclusion

We have proposed a family of efficient adaptive filtering algorithms based on parallel quadratic-metric projection. The quadratic metrics involved have included both constant and variable ones in time. Our extensive simulations have verified that the proposed algorithms significantly improve the performance while keeping linear computational complexity when we employ an efficient metric. In particular, AV-PSP (a realization of APVP) well performs practically, although its convergence analysis requires further investigation.

The study in this chapter has given light on a concealed path to improve the performance of adaptive filtering algorithms. Concretely speaking, the study has verified that the performance of projection-based adaptive filtering algorithms can be drastically improved by designing the metric appropriately according to a priori or a posteriori information about characteristics of estimandum. We believe that the proposed algorithms shall contribute in a variety of applications such as multiple access interference suppression in CDMA/MIMO systems.

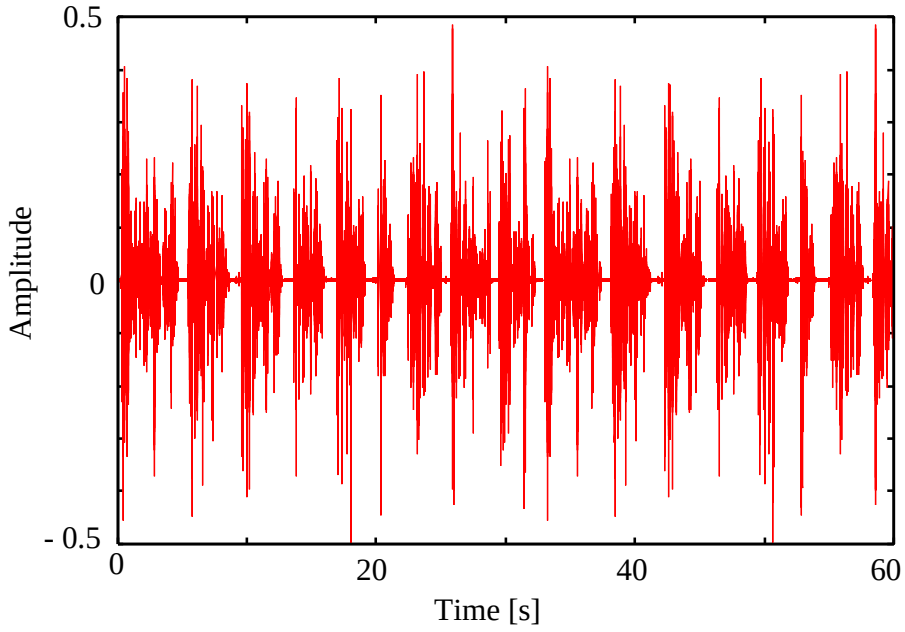


(a)

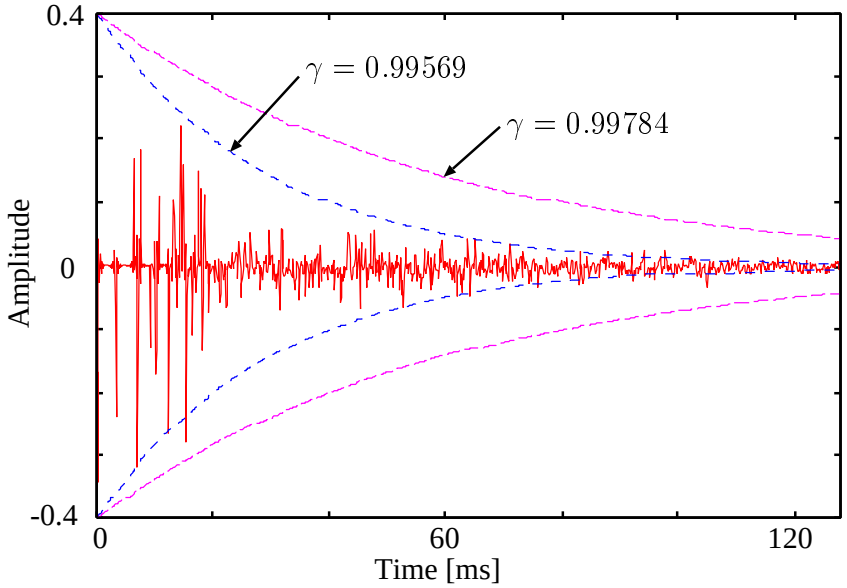


(b)

Figure 6.4.4: System mismatch curves for (a) an exponentially decaying impulse response and (b) an impulse response which is flat in the first and second halves respectively. For ESP, we set $r = 1$ and use the matrices (I) $\boldsymbol{A}$, (II) $\boldsymbol{B}$, and (III) $\boldsymbol{I}$.

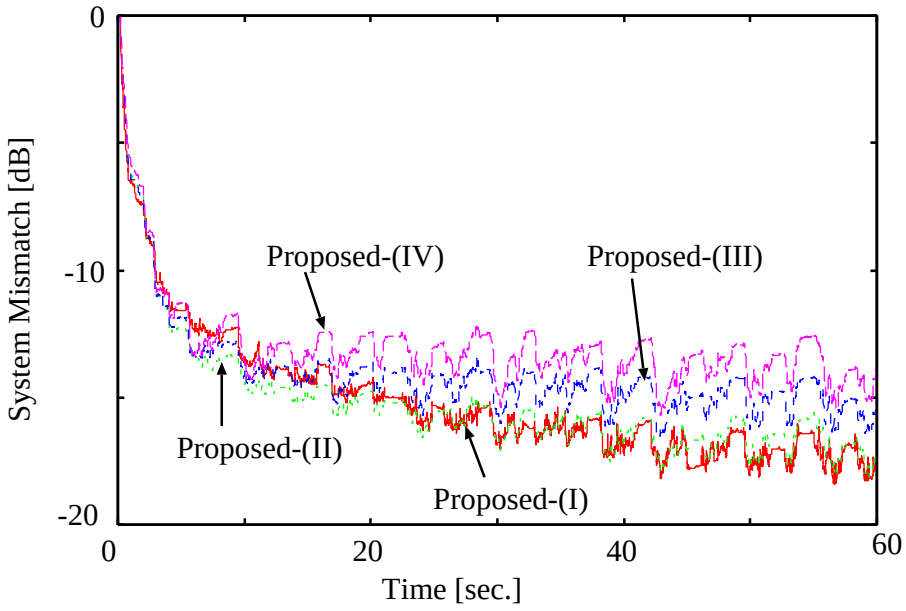


(a)

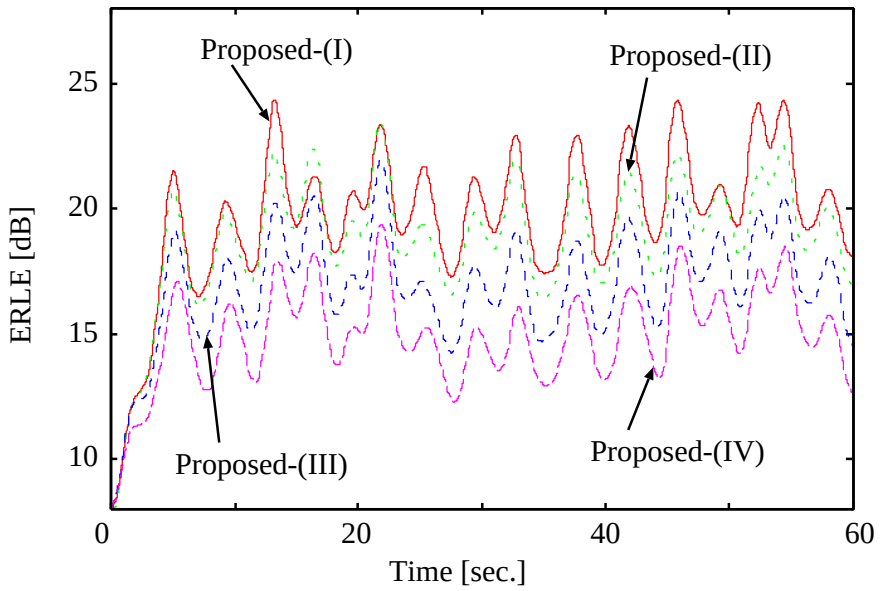


(b)

Figure 6.4.5: (a) The input signal (male's speech) and (b) the recorded room impulse response and the exponential curves given by the diagonal elements of the matrix $\boldsymbol{A}$ (γ is the exponential factor).

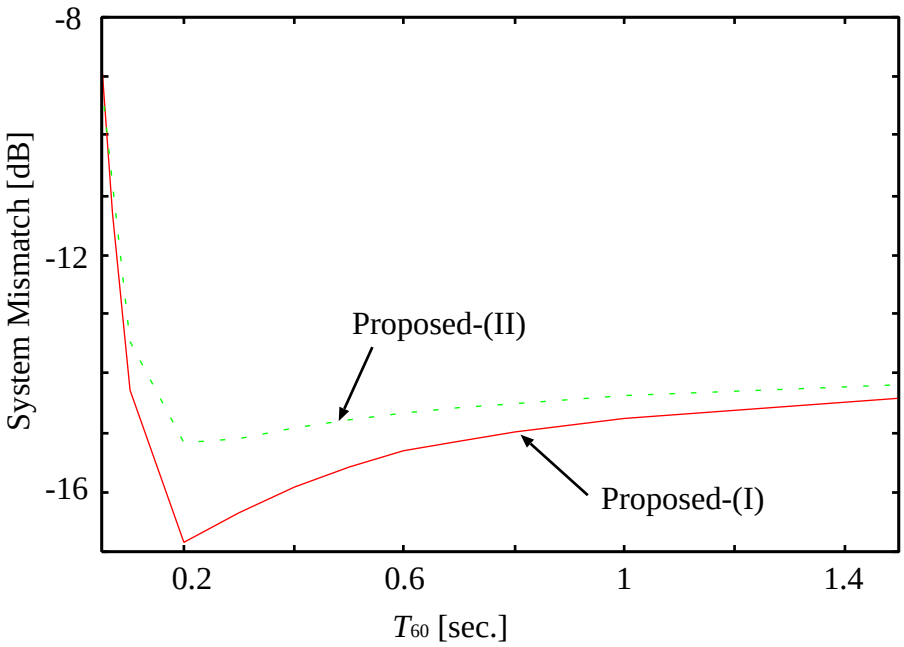


(a)

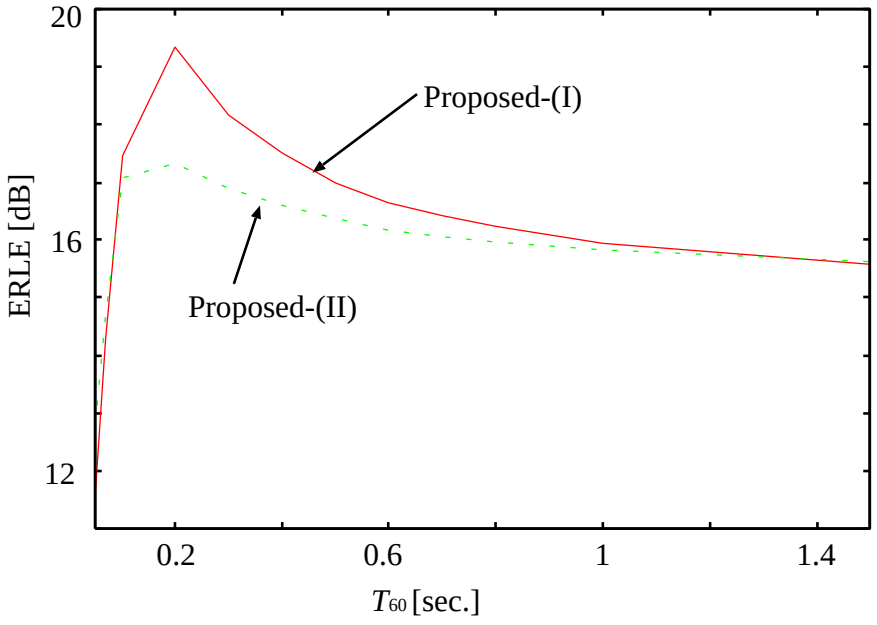


(b)

Figure 6.4.6: A comparison among the proposed algorithms for $q = 8$ and $r = 1$, and $\gamma = 0.99569$. The employed algorithms are (I) AV-PSP, and AQ-PSP for (II) $\boldsymbol{Q} = \boldsymbol{A}^{-1}$, (III) $\boldsymbol{Q} = \boldsymbol{B}^{-1}$, and (IV) $\boldsymbol{Q} = \boldsymbol{I}$.



(a)



(b)

Figure 6.4.7: Effects of the estimation of T_{60} ranging within [0.05,1.5] on the steady-state performance in (a) system mismatch and (b) ERLE.

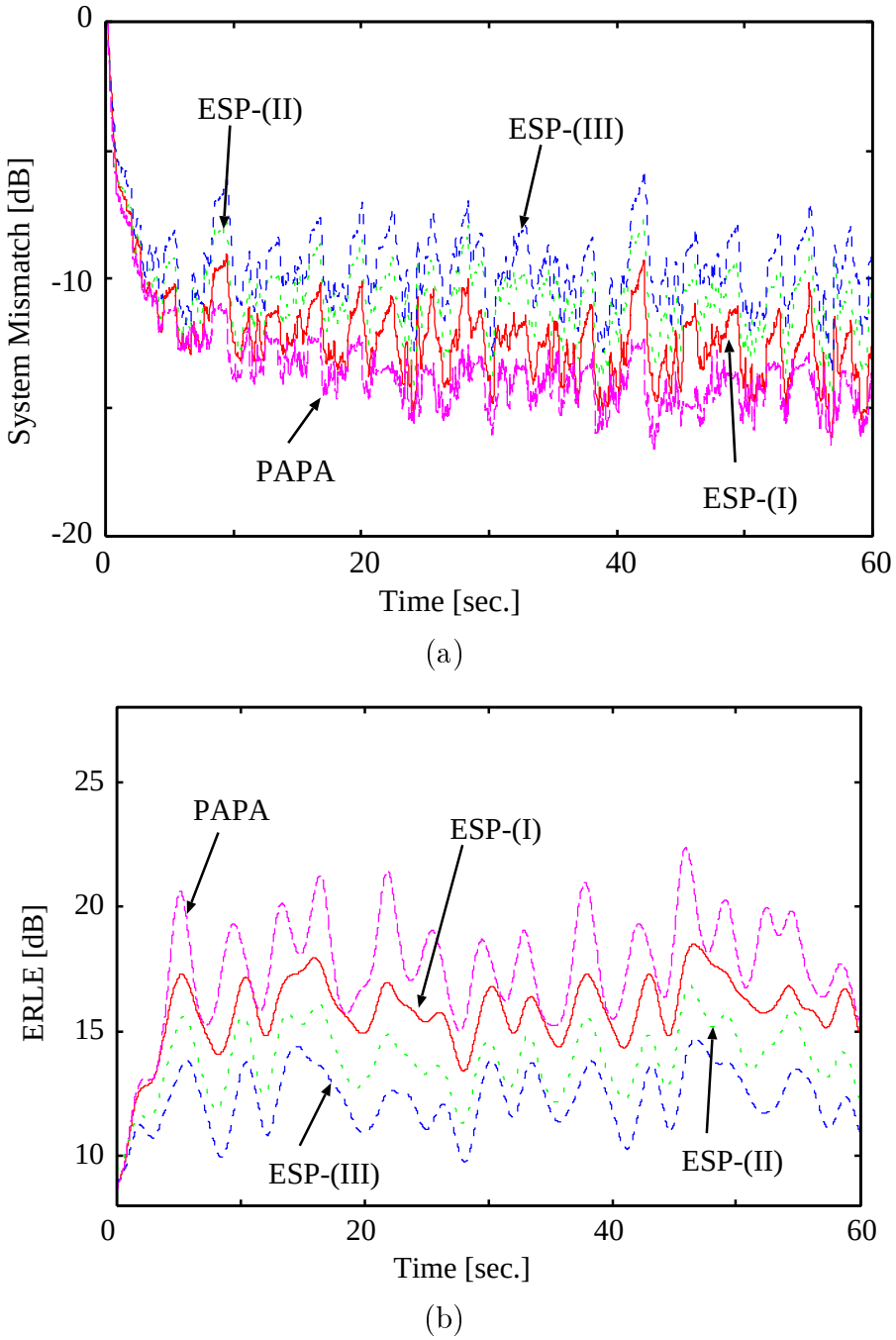
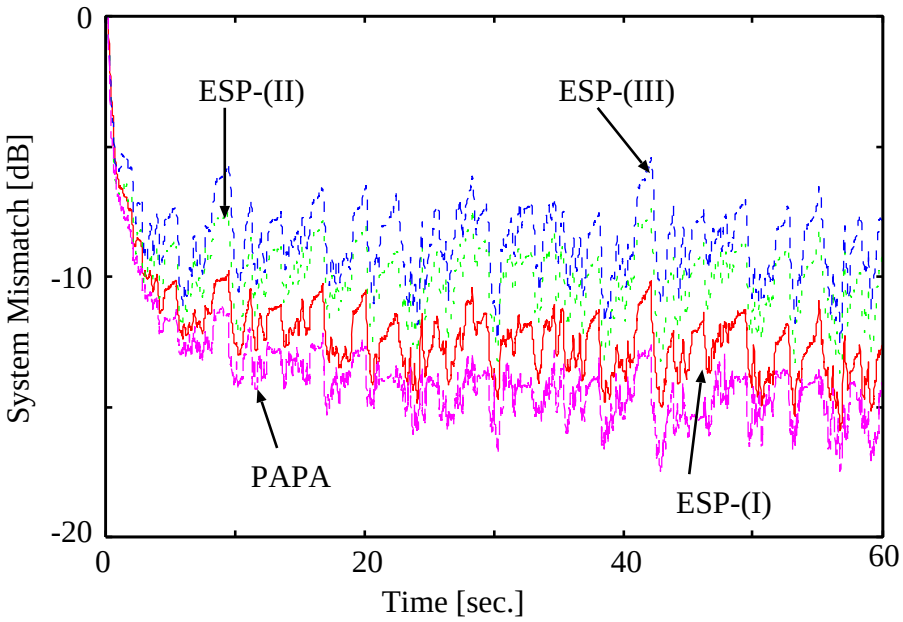
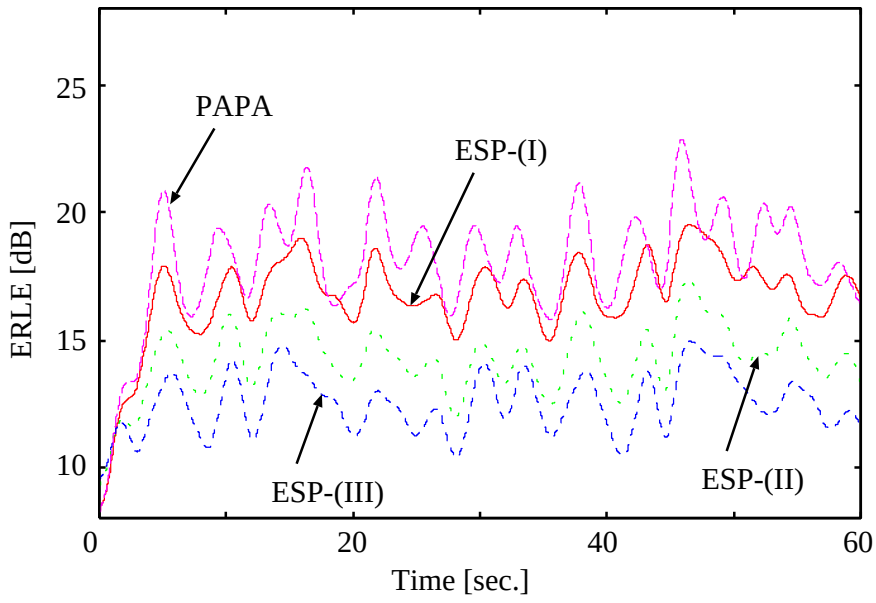


Figure 6.4.8: A comparison between the PAPA and ESP algorithms for $r = 2$. For ESP, we use the matrices (I) $\boldsymbol{A}$, (II) $\boldsymbol{B}$, and (III) $\boldsymbol{I}$.

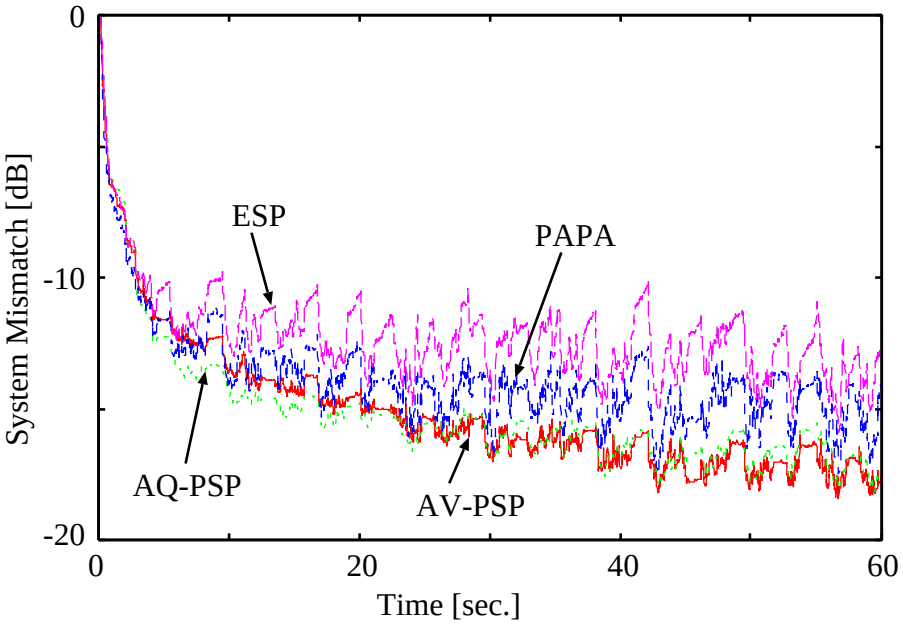


(a)

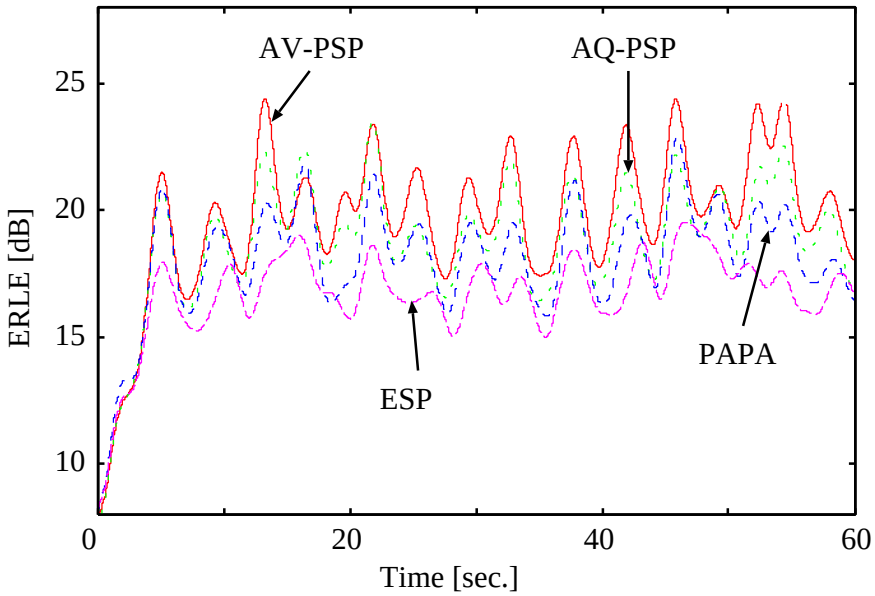


(b)

Figure 6.4.9: A comparison between the PAPA and ESP algorithms for $r = 8$. For ESP, we use the matrices (I) $\boldsymbol{A}$, (II) $\boldsymbol{B}$, and (III) $\boldsymbol{I}$.



(a)



(b)

Figure 6.4.10: A comparison of the proposed algorithms with PAPA and ESP. For the proposed algorithms, we use (I) AV-PSP and (II) AQ-PSP for $q = 8$ and $r = 1$, and $\gamma = 0.99569$. For PAPA, we set $r = 8$. For ESP, we set $r = 8$ with the matrix $\boldsymbol{A}$.

Chapter 7

General Conclusion

Motivated by the growing demand for an adaptive signal processing technique well-performing even in nonstationary environments, this study has developed efficient adaptive filtering algorithms and has proved their efficacy in applications to acoustic and communication systems. The proposed algorithms are based on the *Adaptive Projected Subgradient Method (APSM)*, which has been introduced in Chapter 2. The main body of this thesis has been constructed by four chapters: Chapters 3–6.

Chapter 3 has presented *Adaptive Parallel Constrained Projection (A-PCP) method*, a family of efficient linearly-constrained adaptive filtering algorithms based on *embedded constraint* and *parallel structure*. Two efficient blind adaptive algorithms belonging to A-PCP have been proposed for Multiple Access Interference (MAI) suppression in DS/CDMA wireless communication systems. The simulation results have demonstrated that the proposed algorithms attain approximately 10 times faster convergence speed than the conventional SAGP and CNLMS algorithms and, simultaneously, BER performance close to the MOE linear filter.

Chapter 4 has presented *Pairwise Optimal Weight Realization* (POWER), an efficient adaptive weighting technique to bring out the potential of the adaptive-PSP algorithm aggressively while keeping its computational efficiency. The POWER technique employs an efficient formula to compute projection onto the intersection of two closed half-spaces for an efficient approximation of an ideal direction of update. Important properties of POWER, including optimality in the sense of (i) pairwise and (ii) maximum minimization, have been presented.

The proposed algorithm enjoys fast and stable convergence and good steady-state performance while keeping linear complexity.

Chapter 5 has presented a class of efficient adaptive filtering algorithms for Stereophonic Acoustic Echo Cancellation (SAEC) problem based on two key ideas. The first idea is to utilize, based on the adaptive-PSP algorithm, information provided by preprocessing. The second idea is to employ POWER in an efficient manner for further acceleration with keeping linear computational complexity. In fact, the POWER technique has turned out to exert far-reaching effects in the SAEC problem. The proposed technique has exhibited excellent convergence and tracking behavior after a change of the echo paths in the extensive simulations.

Chapter 6 has presented a family of very flexible adaptive algorithms based on quadratic-metric. First, two adaptive algorithms, in which the metric is constant in time, have been presented: *Adaptive Parallel Quadratic-metric Projection (APQP) algorithm* and *Adaptive Parallel Min-max Quadratic-metric Projection (APMQP) algorithm*. Then, an adaptive algorithm, in which the metric is variable in time, has been presented: *Adaptive Parallel Variable-metric Projection (APVP) algorithm*. The proposed algorithms (APQP/APMQP/APVP) has the valuable monotone property. By employing an efficient metric, the overall computational complexity of the proposed algorithms is kept linear w.r.t. the filter length. The efficacy of the proposed algorithms has been verified in the acoustic echo canceling application.

The consequence of this study proves the efficacy of APSM in real-world applications, since all the proposed algorithms are naturally derived by APSM. All the algorithms developed in this study moreover have the *inherently parallel structure*, thus enjoying *fault-tolerance nature*, as mentioned clearly in Chapter 4; those algorithms which are just parallelizable somehow is *never* equipped with such nature. The fruit provided by this study will contribute to developments in a wide range of engineering applications including acoustic and communication systems.

Bibliography

[1] A. E. Albert and L. S. Gardner, Jr. *Stochastic approximation and nonlinear regression.* Cambridge MA: MIT Press, 1967.

[2] M. Ali. Stereophonic acoustic echo cancellation system using time-varying all-pass filtering for signal decorrelation. In *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 3689–3692, 1998.

[3] H. H. Bauschke. *Projection algorithms and monotone operators.* PhD thesis, Simon Fraser University, Aug. 1996.

[4] H. H. Bauschke and J. M. Borwein. On projection algorithms for solving convex feasibility problems. *SIAM Review*, 38(3):367–426, Sept. 1996.

[5] H. H. Bauschke and P. L. Combettes. A weak-to-strong convergence principle for Fejér monotone methods in Hilbert spaces. *Mathematics of Operations Research*, 26(2):248–264, May 2001.

[6] A. Ben-Israel and T. N. E. Greville. *Generalized inverse: Theory and applications.* Wiley, New York, 1974.

[7] A. Benallal and A. Gilloire. A new method to stabilize fast RLS algorithms based on a first-order model of the propagation of numerical errors. In *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 1373–1376, 1988.

[8] J. Benesty, F. Amand, A. Gilloire, and Y. Grenier. Adaptive filtering algorithms for stereophonic acoustic echo cancellation. In *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 3099–3102, 1995.

[9] J. Benesty and S. L. Gay. An improved PNLMS algorithm. In *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 1881–1884, 2002.

[10] J. Benesty, T. Gänsler, D. R. Morgan, M. M. Sondhi, and S. L. Gay. *Advances in Network and Acoustic Echo Cancellation*. Berlin: Springer-Verlag, 2001.

[11] J. Benesty and Y. Huang, editors. *Adaptive Signal Processing —Applications to real-world problems*. Berlin: Springer, 2003.

[12] J. Benesty, D. R. Morgan, J. L. Hall, and M. M. Sondhi. Stereophonic acoustic echo cancellation using nonlinear transformations and comb filtering. In *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 3673–3676, 1998.

[13] J. Benesty, D.R. Morgan, and M. M. Sondhi. A better understanding and an improved solution to the specific problems of stereophonic acoustic echo cancellation. *IEEE Transactions on Speech Audio Processing*, 6(3):156–165, 1998.

[14] J. Botto and G. V. Moustakides. Stabilizing the fast Kalman algorithms. *IEEE Transactions on Acoustics, Speech, Signal Processing*, ASSP-37(9):1342–1348, Sept. 1989.

[15] C. Breining, P. Dreiseitel, E. Hänsler, A. Mader, B. Nitsch, H. Puder, T. Schertler, G. Schmidt, and J. Tilp. Acoustic echo control — an application of very-high-order adaptive filters. *IEEE Signal Processing Magazine*, 16(4):42–69, July 1999.

[16] H. Buchner, J. Benesty, and W. Kellermann. Generalized multichannel frequency-domain adaptive filtering: efficient realization and application to hands-free speech communication. *Signal Processing*, 85:549–570, 2005.

[17] D. Butnariu, Y. Censor, and S. Reich, editors. *Inherently parallel algorithms in feasibility and optimization and their applications*. New York: Elsevier, 2001.

[18] G. Carayannis, D. Manolakis, and N. Kalouptsidis. A fast sequential algorithm for least-squares filtering and prediction. *IEEE Transactions on Acoustics, Speech, Signal Processing*, ASSP-31(6):1394–1402, Dec. 1983.

[19] B. D. Carlson. Covariance matrix estimation errors and diagonal loading in adaptive arrays. *IEEE Transactions on Aerospace and Electronic Systems*, 24(4):397–401, July 1988.

[20] R. L. G. Cavalcante, I. Yamada, and K. Sakaniwa. A fast blind MAI reduction based on adaptive projected subgradient method. *IEICE Transactions on Fundamentals*, E87-A(8):1973–1980, Aug. 2004.

[21] R. L. G. Cavalcante, M. Yukawa, and I. Yamada. Set-theoretic DS/CDMA receivers for fading channels by adaptive projected subgradient method. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, SP06-1, 2005.

[22] Y. Censor and S. A. Zenios. *Parallel optimization: Theory, algorithm, and optimization*. New York: Oxford University Press, 1997.

[23] J. Cioffi and T. Kailath. Fast recursive LS transversal filters for adaptive processing. *IEEE Transactions on Acoustics, Speech, Signal Processing*, ASSP-32(2):304–337, Apr. 1984.

[24] P. L. Combettes. The foundations of set theoretic estimation. *Proceedings of the IEEE*, 81(2):182–208, Feb. 1993.

[25] P. L. Combettes. Convex set theoretic image recovery by extrapolated iterations of parallel subgradient projections. *IEEE Transactions on Image Processing*, 6(4):493–506, Apr. 1997.

[26] L. Couvreur and C. Couvreur. Blind model selection for automatic speech recognition in reverberant environments. *Journal of VLSI Signal Processing*, 36(2–3):189–203, Mar. 2004.

[27] M. L. R. de Campos, J. Antonio, and J. A. Apolinário Jr. The constrained affine projection algorithm —Development and convergence issues. In *Proceedings of the First IEEE Balkan Conference on Signal Processing, Communications, Circuits and Systems*, 2000.

[28] M. L. R. de Campos, S. Werner, and J. A. Apolinário Jr. Constrained adaptive algorithms employing householder transformation. *IEEE Transactions on Signal Processing*, 50(9):2187–2195, Sept. 2002.

[29] H. Deng and M. Doroslovački. Improving convergence of the PNLMS algorithm for sparse impulse response identification. *IEEE Signal Processing Letters*, 12(3):181–184, Mar. 2005.

[30] F. Deutsch. *Best Approximation in Inner Product Spaces*. Springer, 2001.

[31] P. S. R. Diniz and S. Werner. Set-membership binormalized data-reusing LMS algorithms. *IEEE Transactions on Signal Processing*, 51(1):124–134, Jan. 2003.

[32] M. Doroslovački and H. Deng. On convergence of proportionate-type NLMS adaptive algorithms. In *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 105–108, 2006.

[33] D. L. Duttweiler. Proportionate normalized least-mean-squares adaptation in echo cancelers. *IEEE Transactions on Speech Audio Processing*, 8(5):508–518, Sept. 2000.

[34] P. Eneroth, S. L. Gay, T. Gänsler, and J. Benesty. A real-time implementation of a stereophonic acoustic echo canceler. *IEEE Transactions on Speech Audio Processing*, 9(5):513–523, 2001.

[35] E. Eweda. Comparison of RLS, LMS, and sign algorithms for tracking randomly time-varying channels. *IEEE Transactions on Signal Processing*, 42(11):2937–2944, Nov. 1994.

[36] T. R. Fortescue, L. S. Kershenbaum, and B. E. Ydstie. Implementation of self-tuning regulators with variable forgetting factors. *Automatica*, 17:831–835, 1981.

[37] T. Fujii and S. Shimada. A note on multi-channel echo cancellers. CS84-178, IEICE, Jan. 1985. in Japanese.

[38] K. Fukawa and H. Suzuki. Orthogonalizing matched filter (OMF) detection for DS-CDMA mobile radio systems. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, pages 385–389, 1994.

[39] K. Fukawa and H. Suzuki. Orthogonalizing matched filtering (OMF) detector for DS-CDMA mobile communication systems. *IEEE Transactions on Vehicular Technology*, 48(1):188–197, Jan. 1999.

[40] S. L. Gay. Dynamically regularized fast RLS with application to echo cancellation. In *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 957–960, 1996.

[41] S. L. Gay. An efficient fast converging adaptive filter for network echo cancellation. In *Proceedings of the Asilomar Conference on Signals, Systems, and Computers*, pages 394–398, 1998.

[42] S. L. Gay and J. Benesty, editors. *Acoustic signal processing for telecommunication*. Boston: Kluwer, 2000.

[43] S. L. Gay and S. Tavathia. The fast affine projection algorithm. In *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 3023–3026, 1995.

[44] T. Gänsler and J. Benesty. Stereophonic acoustic echo cancellation and two-channel adaptive filtering: an overview. *International Journal of Adaptive Control and Signal Processing*, 14(6):565–586, 2000.

[45] T. Gänsler and J. Benesty. Multichannel acoustic echo cancellation: what's new? In *Proceedings of the International Workshop on Acoustic Echo and Noise Control (IWAENC)*, 2001.

[46] T. Gänsler and J. Benesty. New insights into the stereophonic acoustic echo cancellation problem and an adaptive nonlinearity solution. *IEEE Transactions on Speech Audio Processing*, 10(5):257–267, July 2002.

[47] T. Gänsler and P. Eneroth. Influence of audio coding on stereophonic acoustic cancellation. In *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 3649–3652, 1998.

[48] T. Gänsler, S. L. Gay, M. M. Sondhi, and J. Benesty. Double-talk robust fast converging algorithms for network echo cancellation. *IEEE Trans. Speech Audio Processing*, 8(6):656–663, Nov. 2000.

[49] A. Gilloire and V. Turbin. Using auditory properties to improve the behavior of stereophonic acoustic echo cancellers. In *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 3681–3683, 1998.

[50] G. O. Glentis, K. Berberidis, and S. Theodoridis. A unified view —Efficient least squares adaptive algorithms for FIR transversal filtering. *IEEE Signal Processing Magazine*, 16(4):13–41, July 1999.

[51] S. Gollamudi, S. Nagaraj, S. Kapoor, and Y. H. Huang. Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size. *IEEE Signal Processing Letters*, 5(5):111–114, May 1998.

[52] L. Guo, A. Ekpenyong, and Y. H. Huang. Frequency-domain adaptive filtering —*A* set-membership approach. In *Proceedings of the Asilomar Conference on Signals, Systems, and Computers*, pages 2073–2077, 2003.

[53] B. Hassibi and T. Kailath. H^∞ bounds for least-squares estimators. *IEEE Transactions on Automatic Control*, 46(2):309–314, Feb. 2001.

[54] B. Hassibi, A. H. Sayed, and T. Kailath. H^∞ optimality of the LMS algorithm. *IEEE Transactions on Signal Processing*, 44(2):267–280, Feb. 1996.

[55] B. Hassibi, A. H. Sayed, and T. Kailath. *Infinite-quadratic estimation and control: A unified approach to* H^2 *and* H^∞ *Theories*. SIAM, Philadelphia, 1999.

[56] S. Haykin. *Adaptive Filter Theory*. New Jersey: Prentice Hall, 4th edition, 2002.

[57] S. Haykin, A. H. Sayed, J. R. Zeidler, P. Yee, and P. C. Wei. Adaptive tracking of linear time-variant systems by extended RLS algorithms. *IEEE Transactions on Signal Processing*, 45(5):1118–1128, May 1997.

[58] T. Hinamoto and S. Maekawa. Extended theory of learning identification. *Transactions of the Institute of Electrical Engineers of Japan*, 95(10):227–234, 1975 (in Japanese).

[59] A. Hirano, K. Nakayama, D. Someda, and M. Tanaka. Stereophonic acoustic echo canceller without pre-processing. In *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 145–148, 2004.

[60] A. Hirano, K. Nakayama, and K. Watanabe. Convergence analysis of stereophonic echo canceller with pre-processing–*R*elation between pre-processing and convergence–. In *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 861–864, 1999.

[61] J. B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms I —Fundamentals*. Springer-Verlag, Germany, 1993.

[62] M. Honig, U. Madhow, and S. Verdu. Blind adaptive multiuser detection. *IEEE Transactions on Information Theory*, 41(4):944–960, July 1995.

[63] M. Honig and M. K. Tsatsanis. Adaptive techniques for multiuser CDMA receivers. *IEEE Signal Processing Magazine*, 17(3):49–61, May 2000.

[64] O. Hoshuyama, R. A. Goubran, and A. Sugiyama. A generalized proportionate variable step-size algorithm for fast changing acoustic environments. In *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2004.

[65] K. Ikeda and R. Sakamoto. Convergence analyses of stereo acoustic echo cancelers with preprocessing. *IEEE Transactions on Signal Processing*, 51(5):1324–1334, May 2003.

[66] Y. Joncour and A. Sugiyama. A stereo echo canceler with pre-processing for correct echo path identification. In *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 3677–3680, 1998.

[67] J. A. Apolinário Jr., S. Werner, P. S. R. Diniz, and T. I. Laakso. Constrained normalized adaptive filters for CDMA mobile communications. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, volume IV, pages 2053–2056, 1998.

[68] A. W. H. Khong and P. A. Naylor. Reducing inter-channel coherence in stereophonic acoustic echo cancellation using partial update adaptive filters. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pages 405–408, 2004.

[69] H. Kuttruff. *Room acoustics*. Elsevier, 4th edition, 2000.

[70] J. T. Lai, A. Y. Wu, and C. C. Yeh. A novel multipath matrix algorithm for exact room response identification in stereo echo cancellation. In *Proceedings of the IEEE Workshop on Signal Processing Systems (SIPS)*, pages 236–240, 2003.

[71] S. H. Leung and C. F. So. Gradient-based variable forgetting factor RLS algorithm in time-varying environments. *IEEE Transactions on Signal Processing*, 53(8):3141–3150, Aug. 2005.

[72] L. Ljung, M. Morf, and D. Falconer. Fast calculations of gain matrices for recursive estimation schemes. *International Journal of Control*, 27:1–19, Jan. 1978.

[73] D. G. Luenberger. *Optimization by Vector Space Methods*. New York: Wiley, 1969.

[74] R. Lupas and S. Verdu. Linear multiuser detectors in synchronous code-decision multiple-access channels. *IEEE Transactions on Information Theory*, 35(1):123–136, Jan. 1989.

[75] R. Lupas and S. Verdu. Near-far resistance of multiuser detectors in asynchronous channels. *IEEE Transactions on Communications*, 38(4):496–508, Apr. 1990.

[76] U. Madhow and M. L. Honig. $MMSE$ interference suppression for direct-sequence spread-spectrum $CDMA$. *IEEE Transactions on Communications*, 42(12):3178–3188, Dec. 1994.

[77] S. Makino and Y. Kaneda. Exponentially weighted step-size projection algorithm for acoustic echo cancellers. *IEICE Transactions on Fundamentals*, E75-A(11):1500–1508, Nov. 1992.

[78] S. Makino, Y. Kaneda, and N. Koizumi. Exponentially weighted stepsize NLMS adaptive filter based on the statistics of a room impulse response. *IEEE Transactions on Speech Audio Processing*, 1(1):101–108, Jan. 1993.

[79] S. L. Miller. An adaptive direct-sequence code division multiple-access receiver for multiuser interference rejection. *IEEE Transactions on Communications*, 43(2/3/4):1746–1754, 1995.

[80] D. R. Morgan, J. L. Hall, and J. Benesty. Investigation of several types of nonlinearities for use in stereo acoustic echo cancellation. *IEEE Transactions on Speech Audio Processing*, 9(6):686–696, Sept. 2001.

[81] D. R. Morgan and S. G. Kratzer. On a class of computationally efficient, rapidly converging, generalized NLMS algorithms. *IEEE Signal Processing Letters*, 3(8):245–247, Aug. 1996.

[82] G. V. Moustakides. Correcting the instability due to finite precision of the fast Kalman identification algorithms. *Signal Processing*, 18:33–42, 1989.

[83] G. V. Moustakides and S. Theodoridis. Fast Newton transversal filters —A new class of adaptive estimation algorithms. *IEEE Transactions on Signal Processing*, 39(10):2184–2193, Oct. 1991.

[84] J. Nagumo and J. Noda. A learning method for system identification. *IEEE Transactions on Automatic Control*, 12(3):282–287, June 1967.

[85] M. Z. Nashed. *Generalized Inverse and Applications*. Academic Press, New York, 1976.

[86] M. Nekuii and M. Atarodi. A fast converging algorithm for network echo cancellation. *IEEE Signal Processing Letters*, 11(4):427–430, Apr. 2004.

[87] N. Ogura and I. Yamada. A deep outer approximating half space of the level set of certain quadratic functions. *Journal of Nonlinear Convex Analysis*, 6(1):187–201, 2005.

[88] I. Oppermann, P. van Rooyen, and R. Kohno. Guest editorial spread spectrum for global communication II. *IEEE Journal on Selected Areas in Communications*, 18(1):1–5, Jan. 2000.

[89] K. Ozeki and T. Umeda. An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties. *Electronics and Communications in Japan*, 67-A(5):19–27, 1984.

[90] S. C. Park and J. F. Doherty. Generalized projection algorithm for blind interference suppression in $DS/CDMA$ communications. *IEEE Transactions on Circuits and Systems–II*, 44(6):453–460, June 1997.

[91] R. L. Plackett. Some theorems in least-squares. *Biometrika*, 37:149, 1950.

[92] H. V. Poor and X. Wang. Code-aided interference suppression for $DS/CDMA$ communications —part II: Parallel blind adaptive implementations. *IEEE Transactions on Communications*, 45(9):1112–1122, Sep. 1997.

[93] M. Rupp. A family of adaptive filter algorithms with decorrelating properties. *IEEE Transactions on Signal Processing*, 46(3):771–775, Mar. 1998.

[94] H. Sakai and T. Sakaguchi. Performance analysis of a new adaptive algorithm for blind multiuser detection. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, volume I, pages 379–382, 2002.

[95] S. G. Sankaran and A. A. L. Beex. Convergence behavior of affine projection algorithms. *IEEE Transactions on Signal Processing*, 48(4):1086–1096, Apr. 2000.

[96] S. G. Sankaran and A. A. (Louis) Beex. Stereophonic echo cancellation using NLMS with orthogonal correction factors. In *Proceedings of the International Workshop on Acoustic Echo and Noise Control (IWAENC)*, pages 40–43, 1999.

[97] A. H. Sayed. *Fundamentals of adaptive filtering*. New Jersey: Wiley, 2003.

[98] S. Shimauchi, Y. Haneda, S. Makino, and Y. Kaneda. New configuration for a stereo echo canceler with nonlinear pre-processing. In *Proceedings of*

the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), pages 3685–3688, 1998.

[99] K. Slavakis, I. Yamada, and N. Ogura. Adaptive projected subgradient method over the fixed point set of strongly attracting non-expansive mappings. *Numerical Functional Analysis and Optimization*. accepted.

[100] K. Slavakis, I. Yamada, N. Ogura, and M. Yukawa. Adaptive projected subgradient method and set theoretic adaptive filtering with multiple convex constraints. In *Proceedings of the Asilomar Conference on Signals, Systems, and Computers*, pages 960–964, 2004.

[101] K. Slavakis, M. Yukawa, and I. Yamada. Efficient robust Capon beamforming by the Adaptive Projected Subgradient Method. submitted for publication.

[102] K. Slavakis, M. Yukawa, and I. Yamada. Robust Capon beamforming by the Adaptive Projected Subgradient Method. In *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 1005–1008, 2006.

[103] D. Slock. The block underdetermined covariance (BUC) fast transversal filter (FTF) algorithm for adaptive filtering. In *Proceedings of the Asilomar Conference on Signals, Systems, and Computers*, pages 550–554, 1992.

[104] D. T. M. Slock and T. Kailath. Fast transversal filters with data sequence weighting. *IEEE Transactions on Acoustics and Speech Signal Processing*, 33(3):346–359, Mar. 1989.

[105] D. T. M. Slock and T. Kailath. Numerically stable fast transversal filters for recursive least squares adaptive filtering. *IEEE Transactions on Signal Processing*, 39(1):92–114, Jan. 1991.

[106] C. F. So and S. H. Leung. Variable forgetting factor RLS algorithm based on dynamic equation of gradient of mean square error. *Electronics*, 37(3):202–203, Feb. 2001.

[107] M. M. Sondhi and D. R. Morgan. Acoustic echo cancellation for stereophonic teleconferencing. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, May. 1991.

[108] M. M. Sondhi, D. R. Morgan, and J .L. Hall. Stereophonic acoustic echo cancellation —An overview of fundamental problem. *IEEE Signal Processing Letters*, 2(8):148–151, Aug. 1995.

[109] S. Song, J.-S. Lim, S. J. Baek, and K.-M. Sung. Variable forgetting factor linear least squares algorithm for frequency selective fading channel estimation. *IEEE Transactions on Vehicular Technology*, 51(3):613–616, May 2002.

[110] H. Stark and Y. Yang. *Vector Space Projections – A Numerical Approach to Signal and Image Processing, Neural Nets and Optics*. New York: Wiley, 1998.

[111] A. Sugiyama, A. Hirano, and K. Nakayama. Acoustic echo cancellation for conference systems. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pages 17–20, 2004.

[112] A. Sugiyama, Y. Joncour, and A. Hirano. A stereo echo canceler with correct echo-path identification based on an input-sliding technique. *IEEE Transactions on Signal Processing*, 49(11):2577–2587, 2001.

[113] M. Tanaka, S. Makino, and J. Kojima. A block exact fast affine projection algorithm. *IEEE Transactions on Speech and Audio Processing*, 7(1):79–86, 1999.

[114] N. Tikhonov. On solving incorrectly posed problems and method of regularization. *Doklady Akademii Nauk USSR*, 151:501–504, 1963.

[115] B. Toplis and S. Pasupathy. Tracking improvements in fast RLS algorithms using a variable forgetting factor. *IEEE Transactions on Acoustics and Speech Signal Processing*, 36(2):206–227, Feb. 1988.

[116] I. J. Umoh and T. Ohunfunmi. Lower bounds for the MSE convergence of APA. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 3482–3485, 2006.

[117] V. V. Vasin and A. L. Ageev. *Ill-posed problems with a priori information.* VSP, 1995.

[118] S. Verdu. Minimum probability of error for asynchronous Gaussian multiple-access channels. *IEEE Transactions on Information Theory*, 32(1):85–96, Jan. 1986.

[119] X. Wang and H. V. Poor. Blind multiuser detection: A subspace approach. *IEEE Transactions on Information Theory*, 44(2):677–690, Mar. 1998.

[120] S. Werner, J. A. Apolinário Jr., P. S. R. Diniz, and T. I. Laakso. A set-membership approach to normalized proportionate adaptation algorithms. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2005.

[121] S. Werner and P. S. R. Diniz. Set-membership affine projection algorithm. *IEEE Signal Processing Letters*, 8(8):231–235, Aug. 2001.

[122] B. Widrow and M. E. Hoff, Jr. Adaptive switching circuits. *IRE WESCON Convention Record,* part 4, pages 96–104, 1960.

[123] I. Yamada. The hybrid steepest descent method for the variational inequality problem over the intersection of fixed point sets of nonexpansive mappings. In D. Butnariu, Y. Censor, and S. Reich, editors, *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications*, pages 473–504. North-Holland, Amsterdam, 2001.

[124] I. Yamada. Adaptive projected subgradient method: A unified view for projection based adaptive algorithms. *Journal of IEICE*, 86(8):654–658, 2003. in Japanese.

[125] I. Yamada and N. Ogura. Adaptive projected subgradient method and its applications to set theoretic adaptive filtering. In *Proceedings of the Asilomar Conference on Signals, Systems, and Computers*, pages 600–606, 2003.

[126] I. Yamada and N. Ogura. Adaptive projected subgradient method for asymptotic minimization of sequence of nonnegative convex functions. *Numerical Functional Analysis and Optimization*, 25(7&8):593–617, 2004.

[127] I. Yamada and N. Ogura. Hybrid steepest descent method for variational inequality problem over the fixed point set of certain quasi-nonexpansive mappings. *Numerical Functional Analysis and Optimization*, 25(7&8):619–655, 2004.

[128] I. Yamada, N. Ogura, and M. Yukawa. Adaptive projected subgradient method and its acceleration techniques. In *Proceedings of the International Federation of Automatic Control (IFAC) Workshop on Adaptation and Learning in Control and Signal Processing (ALCOSP)*, pages 639–644, 2004.

[129] I. Yamada, K. Slavakis, and K. Yamada. An efficient robust adaptive filtering algorithm based on parallel subgradient projection techniques. *IEEE Transactions on Signal Processing*, 50(5):1091–1101, May 2002.

[130] M. Yukawa, R. L. G. Cavalcante, and I. Yamada. Efficient blind DS/CDMA receivers by embedded constraint adaptive parallel projection techniques. In *Proceedings of the Nineteenth Signal Processing Symposium*, Yatsugatake, Japan, Nov. 2004.

[131] M. Yukawa, R. L. G. Cavalcante, and I. Yamada. Efficient adaptive blind MAI suppression in DS/CDMA by embedded constraint parallel projection techniques. In *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 929–932, 2005.

[132] M. Yukawa, R. L. G. Cavalcante, and I. Yamada. Efficient blind MAI suppression in DS/CDMA systems by embedded constraint parallel projection techniques. *IEICE Transactions on Fundamentals*, E88-A(8):2062–2071, Aug. 2005.

[133] M. Yukawa, N. Murakoshi, and I. Yamada. An efficient fast stereo echo canceler by pairwise optimal weight realization technique. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2005.

[134] M. Yukawa, N. Murakoshi, and I. Yamada. Efficient fast stereo acoustic echo cancellation based on pairwise optimal weight realization technique. *EURASIP Journal on Applied Signal Processing*, 2006, 2006. to appear.

[135] M. Yukawa and I. Yamada. Pairwise optimal weight realization — Acceleration technique for set-theoretic adaptive parallel subgradient projection algorithm. *IEEE Transactions on Signal Processing*. accepted.

[136] M. Yukawa and I. Yamada. Adaptive parallel subgradient projection techniques with input sliding technique for stereophonic acoustic echo cancellation. In *Proceedings of the International Workshop on Acoustic Echo and Noise Control (IWAENC)*, pages 55–58, 2003.

[137] M. Yukawa and I. Yamada. Acceleration of adaptive parallel projection algorithms by pairwise optimal weight realization. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pages 713–716, 2004.

[138] M. Yukawa and I. Yamada. Efficient adaptive stereo echo canceling schemes based on simultaneous use of multiple state data. *IEICE Transactions on Fundamentals*, E87-A(8):1949–1957, Aug. 2004.

[139] M. Yukawa and I. Yamada. A note on inflation parameter for adaptive parallel subgradient projection algorithms —An optimal design in case of single projection. Technical report, IEICE, Jan. 2005. SIP2004-104.

[140] M. Yukawa and I. Yamada. Adaptive beamforming by constrained parallel projection in the presence of spatially-correlated interferences. In *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 1009–1012, May 2006.

[141] M. Yukawa and I. Yamada. General-metric parallel subgradient projection algorithm and its application to acoustic echo cancellation. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Sept. 2006. accepted.

Publications Related to the Dissertation

Articles in Journals

- Masahiro Yukawa and Isao Yamada, "Efficient adaptive stereo echo canceling schemes based on simultaneous use of multiple state data," *IEICE Transactions on Fundamentals*, vol. E87-A, no. 8, pp. 1949–1957, Aug. 2004.

- Masahiro Yukawa, Renato L. G. Cavalcante and Isao Yamada, "Efficient blind MAI suppression in DS/CDMA systems by embedded constraint parallel projection techniques," *IEICE Transactions on Fundamentals*, vol. E88-A, no. 8, pp. 2062–2071, Aug. 2005.

- Masahiro Yukawa, Noriaki Murakoshi and Isao Yamada, "Efficient fast stereo acoustic echo cancellation based on pairwise optimal weight realization technique," *EURASIP Journal on Applied Signal Processing*, vol. 2006, 2006, to appear.

- Masahiro Yukawa and Isao Yamada, "Pairwise optimal weight realization —Acceleration technique for set-theoretic adaptive parallel subgradient projection algorithm," *IEEE Transactions on Signal Processing*, accepted.

- Masahiro Yukawa, Konstantinos Slavakis and Isao Yamada, "Adaptive Parallel Quadratic-Metric Projection Algorithms," submitted for publication.

[Other publications not related to the dissertation]

- Noriyuki Takahashi, Masahiro Yukawa and Isao Yamada, "An efficient power control for infeasible downlink situations," *IEICE Transactions on Fundamentals*, vol. E89-A, no. 8, pp. 2107–2118, Aug. 2006.

- Konstantinos Slavakis, Masahiro Yukawa and Isao Yamada, "Efficient robust Capon beamforming by the Adaptive Projected Subgradient Method," submitted for publication.

Peer-Reviewed Articles in International Conference Proceedings

- I. Yamada and M. Yukawa. An efficient robust stereophonic acoustic echo canceler based on parallel subgradient projection techniques. in *Proceedings of ICSP*, E-8, Beijing: China, Aug. 2002.

- M. Yukawa and I. Yamada. Adaptive parallel subgradient projection techniques with input sliding technique for stereophonic acoustic echo cancellation. in *Proceedings of IWAENC*, pages 55–58, Kyoto: Japan, Sept. 2003.

- I. Yamada, N. Ogura and M. Yukawa. Adaptive Projected Subgradient Method and its Acceleration Techniques. in *Proceedings of IFAC Workshop on ALCOSP*, Yokohama: Japan, Aug. 2004 (invited paper).

- M. Yukawa and I. Yamada, "Acceleration of adaptive parallel projection algorithms by pairwise optimal weight realization," in *Proceedings of EUSIPCO*, pp. 713–716, Vienna: Austria, Sept. 2004.

- K. Slavakis, I. Yamada, N. Ogura and M. Yukawa, "Adaptive projected subgradient method and set theoretic adaptive filtering with multiple convex constraints," in *Proceedings of Asilomar Conference on Signals, Systems, and Computers*, pp. 960–964, Nov. 2004.

- M. Yukawa, R. L. G. Cavalcante and I. Yamada, "Efficient adaptive blind MAI suppression in DS/CDMA by embedded constraint parallel projection

techniques," in *Proceedings of IEEE ICASSP*, pp. 929–932, Philadelphia: PA, Mar. 2005.

- M. Yukawa, N. Murakoshi and I. Yamada, "An efficient fast stereo echo canceler by pairwise optimal weight realization technique," in *Proceedings of EUSIPCO*, ThuAmPO2 (Topics in Audio Processing), Antalya: Turkey, Sept. 2005.

- M. Yukawa, K. Slavakis and I. Yamada, "Stereo echo canceler by Adaptive Projected Subgradient Method with multiple room-acoustics information," in *Proceedings of IWAENC*, S03-15, pp. 185–188, Eindhoven: Netherlands, Sept. 2005.

- R. L. G. Cavalcante, M. Yukawa and I. Yamada, "Set-theoretic DS/CDMA receivers for fading channels by Adaptive Projected Subgradient Method," in *Proceedings of IEEE GLOBECOM*, SP06-1, St. Louis: MO, Nov.–Dec. 2005.

- M. Yukawa and I. Yamada, "Adaptive beamforming by constrained parallel projection in the presence of spatially-correlated interferences," in *Proceedings of IEEE ICASSP*, pp. 1009–1012, Toulouse: France, May 2006.

- K. Slavakis, M. Yukawa and I. Yamada, "Robust Capon beamforming by the adaptive projected subgradient method," in *Proceedings of IEEE ICASSP*, pp. 1005–1008, Toulouse: France, May 2006.

- N. Takahashi, M. Yukawa and I. Yamada, "An efficient heuristic approach to infeasible downlink power control problem," in *Proceedings of IEEE ICASSP*, pp. 765–768, Toulouse: France, May 2006.

- I. Yamada, K. Slavakis, M. Yukawa and R. L. G. Cavalcante, "Adaptive Projected Subgradient Method and its applications to robust signal processing," in *Proceedings of IEEE ISCAS*, pp. 269–272, Island of Kos: Greece, May 2006, (invited paper).

- M. Yukawa and I. Yamada, "An effective metric in the adaptive parallel subgradient projection algorithm for acoustic echo cancellation," *EUSIPCO*, Florence: Italy, Sept. 2006, to appear.

Articles in Domestic Conferences

- M. Yukawa and I. Yamada "An efficient robust adaptive multi-channel acoustic echo canceler based on 2-stage parallel subgradient projection techniques," in *Proceedings of IEICE DSP Symposium*, C1-2, Hakodate, Nov. 2002.

- M. Yukawa and I. Yamada "Robust 2-Stage Adaptive Parallel Subgradient Projection Techniques for Multi-Channel Acoustic Echo Cancellation," in *Proceedings of SITA*, vol. I, pp. 271–274, Ikaho, Dec. 2002.

- M. Yukawa and I. Yamada "Two-Stage Adaptive Parallel Subgradient Projection Techniques for Multi-Channel Acoustic Echo Cancellation," IEICE Technical Meeting, Tokyo University of Agriculture and Technology, Dec. 2002.

- M. Yukawa and I. Yamada, "Efficient fast adaptive stereo echo canceling schemes based on simultaneous use of multiple state data," in *Proceedings of IEICE DSP Symposium*, B3-5, Ise-Shima, Nov. 2003.

- M. Yukawa and I. Yamada, "An efficient speed-up weighting technique for adaptive parallel projection algorithm," in *Proceedings of Spring Conference of IEICE*, Tokyo, Mar. 2004.

- M. Yukawa and I. Yamada, "Acceleration technique for adaptive parallel subgradient projection algorithm and its properties," in *Proceedings of IEICE SIP Symposium*, B1-1, Yatsugatake, Nov. 2004.

- N. Murakoshi, M. Yukawa and I. Yamada, "An efficient fast stereophonic acoustic echo canceler by pairwise optimal weight realization technique," in *Proceedings of IEICE SIP Symposium*, B2-3, Yatsugatake, Nov. 2004.

- M. Yukawa, R. L. G. Cavalcante and I. Yamada, "Efficient blind DS/CDMA receivers by embedded constraint adaptive parallel projection techniques," in *Proceedings of IEICE SIP Symposium*, C3-3, Yatsugatake, Nov. 2004.

- R. L. G. Cavalcante, M. Yukawa and I. Yamada, "An efficient receiver

for multipath BPSK-DS/CDMA systems by adaptive projected subgradient method," in *Proceedings of IEICE SIP Symposium*, B5-1, Yatsugatake, Nov. 2004.

- M. Yukawa and I. Yamada, "A note on inflation parameter for adaptive parallel subgradient projection algorithms —An optimal design in case of single projection," Technical Report of IEICE, SIP2004-104, vol. 104, no. 558, Jan. 2005.

- M. Yukawa and I. Yamada, "On optimality of POWER weighting technique for adaptive filtering," Technical Report of IEICE, SIP2005-7, vol. 105, no. 29, pp. 37–42, Apr. 2005.

- K. Slavakis, M. Yukawa and I. Yamada, "Efficient robust adaptive beamforming by the Adaptive Projected Subgradient Method —A set-theoretic time-varying approach over multiple a priori constraints," Technical Report of IEICE, SIP2005-52, vol. 105, no. 174, pp. 13–18, July 2005.

- M. Yukawa, K. Slavakis and I. Yamada, "Stereo echo canceler by Adaptive Projected Subgradient Method with multiple a priori information," in *Proceedings of IEICE Society Conference*, A-4-10, p. 79, Sapporo, Sept. 2005.

- M. Yukawa and I. Yamada, "An embedded-constraint adaptive beamformer in the presence of spatially-correlated interferences —convergence speed of constrained and projected NLMS algorithms," in *Proceedings of IEICE SIP Symposium*, B3-2, Kochi, Nov. 2005.

- N. Takahashi, M. Yukawa and I. Yamada, "An efficient heuristic adaptive power control for downlink transmission in infeasible scenarios," in *Proceedings of IEICE SIP Symposium*, A2-2, Kochi, Nov. 2005.

- R. L. G. Cavalcante, M. Yukawa and I. Yamada, "Blind and nonblind DS/CDMA receivers by Adaptive Projected Subgradient Method," in *Proceedings of SITA*, 36. CDMA (I), vol. II, pp. 685–688, Okinawa, Nov. 2005.

- M. Yukawa and I. Yamada, "An effective acoustic echo canceling algorithm by the Adaptive Projected Subgradient Method with a special metric," Technical Report of IEICE, SIP2005-156, vol. 105, no. 633, pp. 81–86, Mar. 2006.