

RAFFAELLO CLASER

**LOW-COMPLEXITY APPROXIMATIONS FOR
THE KALMAN FILTER**

São Paulo
2021

RAFFAELLO CLASER

**LOW-COMPLEXITY APPROXIMATIONS FOR
THE KALMAN FILTER**

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do título de Doutor em Ciências.

São Paulo
2021

RAFFAELLO CLASER

**LOW-COMPLEXITY APPROXIMATIONS FOR
THE KALMAN FILTER**

Versão Corrigida

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do título de Doutor em Ciências.

Área de Concentração:

Engenharia de Sistemas Eletrônicos

Orientador:

Prof. Dr. Vítor Heloiz Nascimento

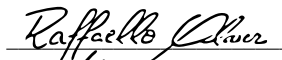
São Paulo
2021

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.


Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 26 de abril de 2021

Assinatura do autor:



Assinatura do orientador:



Catálogo-na-publicação

Claser, Raffaello

Low-complexity approximations for the Kalman filter / R. Claser -- versão corr. -- São Paulo, 2021.

85 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.FILTROS DE KALMAN 2.PROCESSOS ESTOCÁSTICOS
3.PROCESSAMENTO DE SINAIS ADAPTATIVOS I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t.

This dissertation is dedicated to my father João Carlos Claser (in memoriam) and to my mother Giuseppina Gattuso Claser, who always stood by my side and gave me strength to overcome my limits and to complete this thesis.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my supervisor Vítor Heloiz Nascimento for his constant support, guidance, constructive criticism and for the infinite patience over these years.

My thanks to the University of São Paulo for giving me the opportunity to pursue my studies.

A boundless thank you to my mother Giuseppina Gattuso Claser and my sister Carla Claser, who supported me throughout these tough years, for being strict when needed, for nursing me with love and encouraging me to complete my dissertation.

A special thank you to my girlfriend Sirlene Maria Hillesheim for being patient and understanding during this long and arduous journey.

RESUMO

Filtros adaptativos são normalmente empregados em situações em que o ambiente está em constante mudança, de forma que um sistema fixo não possui o desempenho adequado para executar de forma ideal a tarefa desejada. Dentre os exemplos de aplicação, podemos citar: equalização de canal, previsão de dados, cancelamento de eco e assim por diante. Um recurso fundamental dos filtros adaptativos é sua capacidade de rastrear variações nas estatísticas de um sinal em ambientes não estacionários. No entanto, como são normalmente utilizados em aplicações de tempo real, devem ser baseados em algoritmos que requerem menos operações por dado de entrada. O algoritmo *Least Mean Squares* (LMS) representa um dos filtros adaptativos mais simples e fáceis de se aplicar com complexidade linear, enquanto o algoritmo *Recursive Least Squares* (RLS) é conhecido por sua rápida taxa de convergência, mas requer um custo computacional elevado ($\mathcal{O}(M^2)$ para um filtro de tamanho M).

Em cenários variantes no tempo, os esquemas de combinação oferecem recursos de rastreamento aprimorados em relação as componentes de cada filtro. Ao combinar filtros de diferentes famílias, nomeadamente LMS e RLS, é possível tirar vantagem das propriedades de rastreamento de cada filtro e obter uma estrutura com melhor desempenho do que se cada filtro fosse implementado individualmente.

Por outro lado, apesar da alta complexidade computacional ($\mathcal{O}(M^3)$ para modelos gerais de espaço de estados), o filtro de Kalman tem se mostrado a solução ideal para muitas tarefas de rastreamento e previsão de dados, em uma ampla variedade de aplicações, desde navegação até processamento de imagens. Este filtro é ótimo no sentido de minimizar o erro quadrático médio dos parâmetros estimados quando todos os ruídos envolvidos são gaussianos e o vetor de parâmetros a ser estimado muda de acordo com um modelo linear. Ao contrário dos filtros adaptativos, para poder ser projetado, o filtro de Kalman requer conhecimento prévio do modelo matemático do sistema e das características estatísticas dos ruídos envolvidos. Outras versões desse filtro, como o *Extended Kalman Filter* (EKF) e *Unscented Kalman Filter* (UKF), foram desenvolvidas para lidar com modelos não lineares.

Com base neste cenário, o presente trabalho busca comparar o desempenho entre os filtros adaptativos LMS e RLS, bem como sua combinação convexa com a solução ótima obtida via filtro de Kalman sob diferentes modelos autorregressivos de primeira ordem. Além disso, este trabalho também mostra que existem outros modelos para a evolução do vetor de peso ótimo para os quais é possível derivar versões rápidas (ou seja, $\mathcal{O}(M)$) do filtro de Kalman, estendendo o algoritmo RLS-DCD proposto na literatura.

Palavras-Chave – Filtro adaptativo, Filtro de Kalman, Combinação convexa, Processos estocásticos.

ABSTRACT

Adaptive filters are usually employed in situations where the environment is constantly changing, so that a fixed system would not have adequate performance to optimally perform the desired task. Examples include channel equalization, data prediction, echo cancellation and so on. A fundamental feature of adaptive filters is their ability to track variations in the signal statistics for nonstationary environments. However, as they are usually applied in real-time applications, they must be based on algorithms that require a small number of computations per input sample. The least mean squares (LMS) algorithm represents the simplest and most easily applied adaptive filter with linear complexity while the standard recursive least squares (RLS) algorithm is known for its high convergence rate, but requires a higher computational cost ($\mathcal{O}(M^2)$ for a filter of size M).

In time-varying scenarios, combination schemes offer improved tracking capabilities with respect to the component filters. When combining filters from different families, namely LMS and RLS, it is possible to take advantage of the tracking properties from each filter and obtain a structure with better performance than if each filter were implemented individually.

On the other hand, despite the high computational complexity ($\mathcal{O}(M^3)$ for general state-space models), the Kalman filter has long been shown to be the optimal solution to many tracking and data prediction tasks, in a wide variety of applications ranging from navigation to image processing. This filter is optimal in the sense it minimizes the mean square error of the estimated parameters when all noises involved are Gaussian and the parameter vector to be estimated changes according to a linear model. Unlike the adaptive filters, the Kalman filter requires prior knowledge of the mathematical model of the system and the statistical characteristics of noise in order to be designed. Other versions of this filter, such as extended Kalman filter and unscented Kalman filter, were developed in order to deal with nonlinear models.

Based on this scenario, the present work seeks to compare the performance between the adaptive filters LMS and RLS as well as their convex combination with the optimum solution obtained via Kalman filter under different first order autoregressive models. In addition, this work also shows that there exist other models for the evolution of the optimum weight vector for which it is possible to derive fast (i.e., $\mathcal{O}(M)$) versions of the Kalman filter, extending the RLS-DCD algorithm proposed in the literature.

Keywords – Adaptive filter, Kalman filter, Convex combination, Stochastic process.

LIST OF FIGURES

1	General block diagram of an adaptive-filter.	14
2	DCD procedure for minimization.	19
3	Convex combination of two transversal adaptive filters.	26
4	Simulated steady-state EMSE for LMS, RLS, their convex combination and KF, when \mathbf{Q} smoothly changes between \mathbf{R} and \mathbf{R}^{-1}	34
5	Simulated steady-state EMSE for LMS, RLS, their convex combination and KF, when a random perturbation is added to \mathbf{Q}	35
6	Simulated steady-state EMSE curves for μ° -LMS, λ° -RLS, their convex combination and the corresponding Kalman filter when $0 < \theta < 1$	55
7	Tracking comparison between the theoretical steady-state EMSE equations from table 4 and the simulated case for $\min\{\theta_{\min}^{\text{LMS}}, \theta_{\min}^{\text{RLS}}\} < \theta < 1$	55
8	Variation of the optimum parameters for $\min\{\theta_{\min}^{\text{LMS}}, \theta_{\min}^{\text{RLS}}\} < \theta < 1$ considering the general and the approximated equations provided in tables 4 and 5.	56
9	EMSE of μ° -LMS, λ° -RLS, their convex combination and the Kalman filter when \mathbf{Q} smoothly changes between \mathbf{R} and \mathbf{R}^{-1}	57
10	MSD of μ° -LMS, λ° -RLS, their convex combination and the Kalman filter when \mathbf{Q} smoothly changes between \mathbf{R} and \mathbf{R}^{-1}	58
11	Theoretical and simulated steady-state EMSE for μ° -LMS, λ° -RLS, their convex combination and the Kalman filter for $1 \leq M \leq 200$	58
12	Simulated EMSEs learning-curves for $\theta = 0.5$ and considering two different scenarios: a) $\bar{\mathbf{w}}^\circ = \mathbf{0}$ and b) $\bar{\mathbf{w}}^\circ = \mathbf{1}$. To have a better visualization of the result, we used dotted lines to represent the combination and the KF curves.	60
13	Values of θ_{\min} considering the EMSE and NSD (a), and the MSD (b).	61
14	Simulated EMSEs curves for RLS, RLS-DCD, Kalman and Kalman-DCD when $\mathbf{F} = 0.995\mathbf{I}$	70
15	Zoom of the initial iterations in Figure 14.	71

16	Simulated EMSEs curves for RLS, RLS-DCD, Kalman and Kalman-DCD when \mathbf{F} is a 7×7 diagonal matrix composed of $[1, -1, 1, -1, \dots, 1]$	72
17	Zoom of the first iterations in Figure 16.	72
18	Simulated steady-state EMSE for Kalman and Kalman-DCD algorithms for different values of σ_q^2	73

LIST OF TABLES

1	Adaptive Algorithms	15
2	Optimum steady-state EMSEs (ζ°) for LMS, RLS and their combination. . .	33
3	Relation between the Kalman filter and the adaptive filter variables.	51
4	Analytical expressions for the steady-state ζ -EMSE considering LMS, RLS, their convex combination and for the Kalman filter.	52
5	Analytical expressions for the steady-state ζ -EMSE considering LMS, RLS and their convex combination for sufficiently small μ and λ close to 1. . . .	52
6	Analytical expressions for the steady-state ε -MSD considering LMS, RLS, their convex combination and for the Kalman filter.	53
7	Correspondence between Kalman and RLS variables.	65

LIST OF SYMBOLS

We collect here, for ease of reference, a list of the main symbols used throughout the text:

\mathbb{R}	set of real numbers
\mathbf{x}	boldface letter denotes a random scalar or vector variable
\mathbf{X}	boldface capital letter denotes a matrix
$\mathbb{E}\{\cdot\}$	expectation operator
$\text{diag}\{\mathbf{X}\}$	column vector with the diagonal entries of \mathbf{X}
$\text{Tr}\{\mathbf{X}\}$	trace of the matrix \mathbf{X}
\mathbf{I}	identity matrix
μ	step size
λ	forgetting factor
M	filter length
σ_v^2	noise variance
\mathbf{w}_i^o	optimum weight (column vector)
\mathbf{w}_i	weight estimate (column vector)
$\tilde{\mathbf{w}}_i$	weight error vector (column vector)
\mathbf{u}_i	regressor (column vector)
\mathbf{q}_i	random process noise for the adaptive filter (column vector)
$d(i)$	desired variable
$x(i)$	reference signal
$y(i)$	filter output
$e(i)$	output estimation error
$e_a(i)$	a priori estimation error
\mathbf{r}_{du_i}	cross-covariance vector of $d(i)$ and \mathbf{u}_i
\mathbf{R}_i	covariance matrix of the regression data
\mathbf{Q}	covariance matrix of the random process noise for the adaptive filter
\mathbf{P}_i	inverse of an approximation to the covariance matrix \mathbf{R}_i
\mathbf{T}_i	covariance matrix of the random process noise for the Kalman filter
\mathbf{F}_i	transition state matrix
\mathbf{H}_i	measurement matrix
\mathbf{G}_i	control-input model
\mathbf{V}_i	covariance matrix of the random measurement noise for the Kalman filter
\mathbf{t}_i	random process noise for the Kalman filter (column vector)
\mathbf{z}_i	observation signal (column vector)
\mathbf{x}_i	state vector (column vector)
\mathcal{P}_i	error covariance matrix for the Kalman filter
\mathbf{r}_i	residual vector for DCD (column vector)
H	step-size for DCD
Nu	number of updates for DCD
B	number of bits assigned to each coefficient in DCD
$\eta(i)$	mixing factor
$\zeta(i)$	excess mean-square error
$\varepsilon(i)$	mean-square deviation

CONTENTS

1	Introduction	13
1.1	Thesis Overview	13
1.2	Adaptive Filtering	14
1.3	DCD - Dichotomous Coordinate Descent Algorithm	16
1.4	Convex Combination	25
1.5	Kalman Filter	28
1.6	Contributions	29
2	Low-complexity Approximation to the Kalman filter Using Convex Combinations of Adaptive Filters from Different Families	31
2.1	Problem Formulation	32
2.2	Data Model	32
2.3	Simulations	34
2.4	Conclusion	36
3	On the tracking performance of adaptive filters	37
3.1	Introduction	37
3.2	Tracking model	39
3.3	Theoretical steady-state analysis	44
3.3.1	Theoretical steady-state EMSE for LMS	44
3.3.2	Theoretical steady-state EMSE for RLS	48
3.3.3	Theoretical steady-state EMSE for combination between LMS and RLS	49
3.3.4	Theoretical steady-state EMSE for Kalman filter	50
3.4	Simulations	53

3.4.1	DC analysis of the steady-state EMSE	59
3.4.2	Comparison between θ_{\min} and the nonstationarity degree	60
3.5	Conclusion	62
4	Low-complexity approximation to the Kalman filter using the Dichotomous Coordinate Descent Algorithm	63
4.1	Problem Formulation	64
4.2	Kalman-DCD	66
4.3	Simulations	69
4.4	Conclusion	73
5	Future research	75
	References	77
	Appendix A	82
A.1	Diagonal elements of the covariance matrix $\bar{\mathbf{S}}_{\infty}^{(22)}$ for RLS	82
A.2	Diagonal elements of the cross-covariance matrix $\bar{\mathbf{S}}_{\infty}^{(12)}$ for combination between LMS and RLS	83

1 INTRODUCTION

1.1 Thesis Overview

Adaptive filters find application in diverse fields such as communications, control, radar, sonar, acoustics, and speech processing. In most applications, the computational complexity for the adaptation of the filter coefficients must be kept as small as possible, to save expensive real-time resources such as computational power and memory. On the other hand, many applications require the algorithm to be able to track variations on the unknown parameters. Given this scenario, the present thesis describes different techniques to improve the tracking capabilities of adaptive filters, while keeping the complexity of these algorithms as low as possible, either by taking advantage of structure in the data or in the recursive equations, or by using low-complexity minimization algorithms (such as the DCD — Dichotomous Coordinate Descent algorithm).

The following sections present a brief bibliographic review of the main tools (such as DCD, combinations of adaptive filters and Kalman Filtering), that are used later in the text to create new low-complexity algorithms for different types of problems. The next chapters are structured as follows: Chapter 2 describes how the low-complexity convex combination between one LMS and one RLS filters can be used for tracking problems and still obtain performance close to the optimum solution obtained through the Kalman filter; Chapter 3 studies the tracking behavior of combinations of LMS and RLS filters using a more general model for the evolution of the optimum weight vector than the one used in Chapter 2; Chapter 4 discusses under which conditions it is possible to obtain a Kalman filter with linear complexity using the DCD technique; Finally, Chapter 5 presents some topics to be studied in future research.

1.2 Adaptive Filtering

Adaptive filters are required when either the filter specifications are unknown or when they cannot be satisfied by time-invariant filters [1]. These filters are employed in situations in which the environment is constantly changing, so that a fixed system would not have adequate performance. As they are usually applied in real-time applications, they must be based on algorithms that require a small number of computations per input sample. Due to their low cost, reliability, accuracy and flexibility adaptive filters can be very attractive in many applications such as: echo cancellation, system identification, channel equalization, etc.

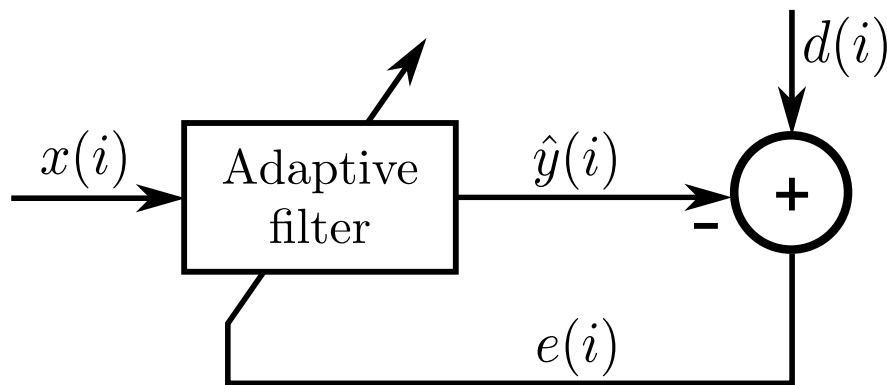


Figure 1: General block diagram of an adaptive-filter.

The general block diagram of an adaptive filtering application is illustrated in Figure 1. In this figure, i is the iteration number, $x(i)$ denotes the reference signal, $\hat{y}(i)$ is the filter output and $d(i)$ defines the desired signal. The error $e(i)$ is calculated as $d(i) - \hat{y}(i)$ and is then used by the adaptation algorithm in order to determine the appropriate update of the filter coefficients. The update law for the filter coefficients is derived through a cost function, which is chosen such that its minimum is achieved when the filter is performing the desired task.

For the class of FIR filters with length M , which will be the case considered in the next chapters, the output of the adaptive filter is given by

$$\hat{y}(i) \triangleq \mathbf{u}_i^T \mathbf{w}_{i-1}, \quad (1.1)$$

where \mathbf{w}_{i-1} is a column vector with the filter coefficients

$$\mathbf{w}_{i-1} = [w_0(i-1) \ w_1(i-1) \ \cdots \ w_{M-1}(i-1)]^T$$

and \mathbf{u}_i is known as the input regressor vector and is frequently a tap-delay line

$$\mathbf{u}_i = [x(i) \ x(i-1) \ \cdots \ x(i-M+1)]^T.$$

For the general block diagram of Figure 1, assume that we have access to several observations of the zero mean random variables $d(i)$ and \mathbf{u}_i , say, $\{d(0), d(1), \dots\}$ and $\{\mathbf{u}_0, \mathbf{u}_1, \dots\}$ with their respective second-order moments $\sigma_d^2(i) = \mathbb{E}\{d^2(i)\}$, $\mathbf{R}_i = \mathbb{E}\{\mathbf{u}_i \mathbf{u}_i^T\}$ and $\mathbf{r}_{du_i} = \mathbb{E}\{d(i) \mathbf{u}_i\}$. Then, in the case of a nonstationary environment in which the optimum solution changes with time, the goal of the adaptive filter is to find the optimal weight $M \times 1$ column vector \mathbf{w}_{i-1}^o that solves the minimization problem

$$\mathbf{w}_{i-1}^o = \underset{\mathbf{w}_{i-1}}{\operatorname{argmin}} \mathbb{E} \{ [d(i) - \mathbf{u}_i^T \mathbf{w}_{i-1}]^2 \}. \quad (1.2)$$

When dealing with adaptive filters there are many types of algorithms, several of each based on two main approaches, namely Steepest-descent and Newton's methods. Since it is hard to know the exact signal statistics (e.g., covariances and cross covariances), which are rarely available in practice, it is necessary to replace the required gradient vectors and Hessian matrices by suitable approximations. Different approximations lead to different algorithms with varied degrees of complexity and different performance properties [2]. In this work, the main algorithms that will be used for comparison and study are: Least Mean Squares — LMS and Recursive Least Squares — RLS. The Kalman filter, which is the optimum filter under certain conditions, will be used as a standard for comparison purposes. For each one of the previous algorithms, except for the Kalman filter which will be discussed later in section 1.5, the optimum solution of (1.2) can be approximated iteratively via the recursions described in Table 1.

Table 1: Adaptive Algorithms

Algorithms	Recursions
LMS	$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}_i [d(i) - \mathbf{u}_i^T \mathbf{w}_{i-1}]$
RLS	$\mathbf{w}_i = \mathbf{w}_{i-1} + \mathbf{P}_i \mathbf{u}_i [d(i) - \mathbf{u}_i^T \mathbf{w}_{i-1}]$, with $\mathbf{P}_i = \lambda^{-1} \left[\mathbf{P}_{i-1} - \frac{\lambda^{-1} \mathbf{P}_{i-1} \mathbf{u}_i \mathbf{u}_i^T \mathbf{P}_{i-1}}{1 + \lambda^{-1} \mathbf{u}_i^T \mathbf{P}_{i-1} \mathbf{u}_i} \right]$

For these recursions, we assumed real-valued signals with the following definitions: $i \geq 0$, with \mathbf{w}_{-1} being an initial guess (usually a zero column vector), μ is a positive step-

size, \mathbf{P}_{-1} = initial guess (usually defined as $\nu^{-1}\mathbf{I}$, where ν is a small positive parameter and \mathbf{I} is the identity matrix), λ is a forgetting factor which lies in the interval $0 \ll \lambda < 1$ and \mathbf{P}_i is the inverse of an approximation $\hat{\mathbf{R}}_i$ to the covariance matrix \mathbf{R}_i given by

$$\hat{\mathbf{R}}_i^{-1} = \left(\lambda^{i+1} \nu \mathbf{I} + \sum_{\ell=0}^i \lambda^{i-\ell} \mathbf{u}_\ell \mathbf{u}_\ell^T \right)^{-1},$$

where $\nu \mathbf{I}$ is an initial condition to guarantee invertibility.

In order to compare the performance of these algorithms, the following measures are commonly used in the literature and will be used in the next chapters:

- Mean-square error (MSE):

$$J(i) \triangleq \mathbb{E} \{e^2(i)\} = \mathbb{E} \{[d(i) - \mathbf{u}_i^T \mathbf{w}_{i-1}]^2\};$$

- Excess mean-square error (EMSE):

$$\zeta(i) \triangleq \mathbb{E} \{e_a^2(i)\} = \mathbb{E} \{(\mathbf{u}_i^T \tilde{\mathbf{w}}_{i-1})^2\},$$

where $e_a(i)$ is the excess a priori error and $\tilde{\mathbf{w}}_{i-1}$ is the weight error vector defined as $\mathbf{w}_{i-1}^o - \mathbf{w}_{i-1}$;

- Mean-square deviation:

$$\varepsilon(i) \triangleq \mathbb{E} \{\|\tilde{\mathbf{w}}_{i-1}\|^2\}.$$

1.3 DCD - Dichotomous Coordinate Descent Algorithm

Many adaptive algorithms require multiplication, division and square root operations, which are complex for hardware implementation since they require a significant chip area and high power consumption [3]. Therefore, it is important to design algorithms that require few multiplication, division and square root operations.

Fast adaptive algorithms based on Newton's method may be unstable in finite precision implementations since they use the inverse of the Hessian matrix, or an approximation to it, to improve the search direction used by the gradient algorithm. As shown in [4], this stability problem can be avoided by using an alternative approach based on solving normal equations efficiently. Although these techniques have good performance to compute the optimal solution, their computational complexity is high, of the order of $\mathcal{O}(M^2)$

operations per sample (M being the filter length). As an example for these techniques, the following algorithms can be highlighted: Euclidean direction search (EDS) [5], stochastic line search algorithm [6] and conjugate gradient [7, 8].

In adaptive filtering, the conventional recursive least squares (RLS) algorithm is known to possess fast convergence, but also to be numerically unstable in finite precision arithmetic and to have a high complexity of $\mathcal{O}(M^2)$ operations [2]. To solve both of these problems, different versions of RLS such as lattice-RLS [9], fast-QR [10, 11] and RLS-DCD [3] algorithms were proposed in the literature. The last technique is based on using the dichotomous coordinate descent (DCD) algorithm to solve the normal equations (see (1.3) below). Although both algorithms, lattice and RLS-DCD, are stable and require a shift structure for the input regressor \mathbf{u}_i , the RLS-DCD algorithm has the advantage of computing explicitly an approximate solution for the weight vector and error signal, while the lattice algorithm computes the error signal exactly using the reflection coefficients, but the filter weights are not provided. The DCD algorithm is designed to be easily implementable in hardware (such as FPGAs, for example), and thus uses additions and comparisons to avoid multiplications, divisions and other operations that are costly to implement in hardware.

Before we give an explanation about the DCD algorithm, let us start with a brief introduction about how the DCD can be applied to the RLS algorithm.

Assume that the RLS weight vector estimates are the solution of the following normal equation:

$$\hat{\mathbf{R}}_i \mathbf{w}_i = \hat{\mathbf{r}}_{du_i}, \quad (1.3)$$

where $\hat{\mathbf{R}}_i$ and $\hat{\mathbf{r}}_{du_i}$ are given by the recursions

$$\hat{\mathbf{R}}_i = \lambda^{i+1} \nu \mathbf{I} + \sum_{j=0}^i \lambda^{i-j} \mathbf{u}_j \mathbf{u}_j^T = \lambda \hat{\mathbf{R}}_{i-1} + \mathbf{u}_i \mathbf{u}_i^T \quad (1.4)$$

and

$$\hat{\mathbf{r}}_{du_i} = \sum_{j=0}^i \lambda^{i-j} d(j) \mathbf{u}_j = \lambda \hat{\mathbf{r}}_{du_{i-1}} + d(i) \mathbf{u}_i, \quad (1.5)$$

with initial conditions $\hat{\mathbf{R}}_{-1} = \nu \mathbf{I}$ and $\hat{\mathbf{r}}_{du_{-1}} = \mathbf{0}$.

The difficulty in solving these expressions is that a general solution for (1.3) involves $\mathcal{O}(M^3)$ operations. This is usually too costly for practical applications, except perhaps for very short filters [12]. The classical RLS algorithm solves this problem partially by using the matrix inversion lemma to compute the inverse of \mathbf{R}_j . This reduces the total

number of computations to $\mathcal{O}(M^2)$. However, because of numerical errors, the resulting algorithm may easily diverge and is difficult to implement in practice.

The RLS-DCD algorithm avoids both of these problems by using some approximations. Assume that we have a good approximation $\hat{\mathbf{w}}_{i-1}$ for \mathbf{w}_{i-1} . Then, it is possible to compute an approximate solution $\hat{\mathbf{w}}_i$ to (1.3), by using a recursive algorithm where $\hat{\mathbf{w}}_{i-1}$ is set as an initial condition and the next iterations are computed just by adding a correction term $\Delta\mathbf{w}_i$ to the current solution, i.e, $\hat{\mathbf{w}}_i = \hat{\mathbf{w}}_{i-1} + \Delta\mathbf{w}_i$.

Since the current solution $\hat{\mathbf{w}}_{i-1}$ is already close to the solution $\hat{\mathbf{w}}_i$, we need only a few iterations of our recursive algorithm to obtain $\hat{\mathbf{w}}_i$. However, this idea only helps if each iteration of the recursive algorithm is very cheap to compute. As shown in [13], in many situations it is possible to obtain a close approximation to the performance of the RLS algorithm by using 1 to 4 updates to approximate the optimal solution \mathbf{w}_i^o with a computational complexity of $\mathcal{O}(M)$ operations.

In general words, DCD is a coordinate descent optimization algorithm modified so as to avoid multiplications and divisions, as we show next. So, let us first consider the problem of finding the solution that minimizes the following one-dimensional quadratic function:

$$\min_{\mathbf{w}} \left\{ f(\mathbf{w}) = \frac{1}{2}a\mathbf{w}^2 - b\mathbf{w} + c \right\}, \quad (1.6)$$

where $a > 0$, b and c are constants.

Assume that we have an approximation for $\hat{\mathbf{w}}(0)$. By choosing an step-size $H > 0$, it is possible to compute:

$$\Delta f_+ = f(\hat{\mathbf{w}}(0) + H) - f(\hat{\mathbf{w}}(0)), \quad \Delta f_- = f(\hat{\mathbf{w}}(0) - H) - f(\hat{\mathbf{w}}(0)). \quad (1.7)$$

If $\Delta f_+ < 0$, update $\hat{\mathbf{w}}(1) = \hat{\mathbf{w}}(0) + H$, and if $\Delta f_- < 0$, update $\hat{\mathbf{w}}(1) = \hat{\mathbf{w}}(0) - H$. If both Δf_+ and Δf_- are positive, choose $\hat{\mathbf{w}}(1) = \hat{\mathbf{w}}(0)$ and reduce the step-size by half: $H \leftarrow H/2$. With the new estimate $\hat{\mathbf{w}}(1)$, the procedure can be repeated until $\hat{\mathbf{w}}(i)$ converges to the minimum of $f(\cdot)$. Figure 2 illustrates this procedure.

The above procedure can be extended to minimize convex functions of several variables as follows. Consider for example the problem of finding the solution \mathbf{w}^o to

$$\min_{\mathbf{w}} \left\{ f(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T \mathbf{R} \mathbf{w} - \mathbf{r}_{du}^T \mathbf{w} + c \right\}, \quad (1.8)$$

where $\mathbf{R} \in \mathbb{R}^{M \times M}$ and $\{\mathbf{w}, \mathbf{r}_{du}\} \in \mathbb{R}^M$.

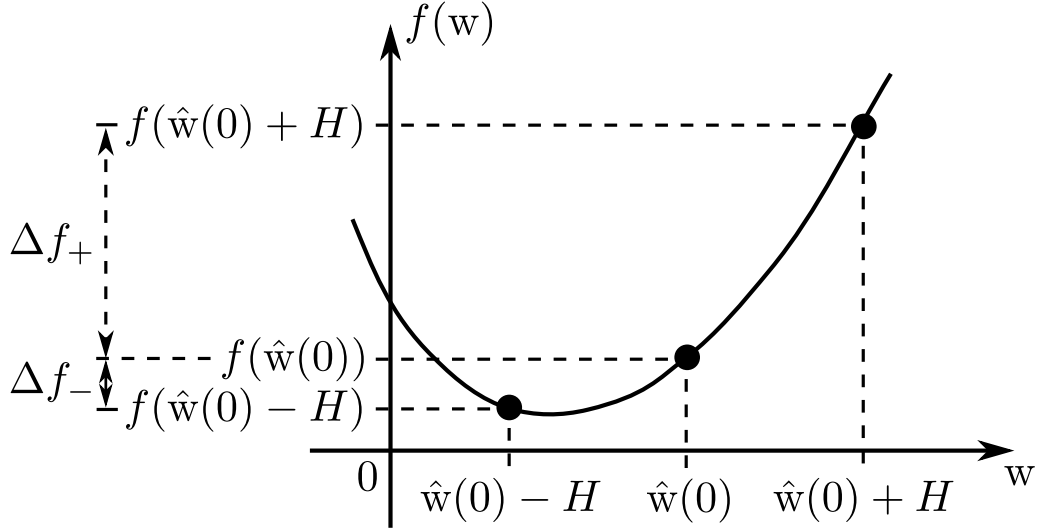


Figure 2: DCD procedure for minimization.

Assume that we have an initial approximation $\hat{\mathbf{w}}_0$ to the solution and we want to find an improved approximation by changing one entry of $\hat{\mathbf{w}}_0$ at a time. In order to differentiate each new approximation, we define:

$$\hat{\mathbf{w}}_i = [\hat{w}_i(1) \ \hat{w}_i(2) \ \cdots \ \hat{w}_i(M)]^T,$$

$$\Delta f_+^{(i-1)}(j) = f(\hat{\mathbf{w}}_{i-1} + H\mathbf{e}_j) - f(\hat{\mathbf{w}}_{i-1}) \quad \text{and}$$

$$\Delta f_-^{(i-1)}(j) = f(\hat{\mathbf{w}}_{i-1} - H\mathbf{e}_j) - f(\hat{\mathbf{w}}_{i-1}),$$

where i denotes the number of the iteration, $j = 1 \dots M$ is the number of the entry and \mathbf{e}_j denotes j column of $M \times M$ identity matrix.

Each entry of $\hat{\mathbf{w}}_i$ will be updated at each iteration according to the following rule:

$$\hat{\mathbf{w}}_i = \begin{cases} \hat{\mathbf{w}}_{i-1} + H\mathbf{e}_j, & \text{if } \Delta f_+^{(i-1)}(j) < 0, \\ \hat{\mathbf{w}}_{i-1} - H\mathbf{e}_j, & \text{if } \Delta f_-^{(i-1)}(j) < 0, \\ \hat{\mathbf{w}}_{i-1}, & \text{otherwise.} \end{cases} \quad (1.9)$$

The change $\Delta f_+^{(i-1)}(j)$ in the cost-function obtained by adding H to each entry is

$$\begin{aligned} \Delta f_+^{(i-1)}(j) &= f(\hat{\mathbf{w}}_{i-1} + H\mathbf{e}_j) - f(\hat{\mathbf{w}}_{i-1}) \\ &= \frac{1}{2}(\hat{\mathbf{w}}_{i-1} + H\mathbf{e}_j)^T \mathbf{R}(\hat{\mathbf{w}}_{i-1} + H\mathbf{e}_j) - \mathbf{r}_{du}^T(\hat{\mathbf{w}}_{i-1} + H\mathbf{e}_j) + c \\ &\quad - \frac{1}{2}\hat{\mathbf{w}}_{i-1}^T \mathbf{R}\hat{\mathbf{w}}_{i-1} + \mathbf{r}_{du}^T \hat{\mathbf{w}}_{i-1} - c \\ &= H\mathbf{e}_j^T \mathbf{R}\hat{\mathbf{w}}_{i-1} + \frac{1}{2}H^2\mathbf{e}_j^T \mathbf{R}\mathbf{e}_j - H\mathbf{r}_{du}^T \mathbf{e}_j. \end{aligned} \quad (1.10)$$

Similarly, the variation $\Delta f_-^{(i-1)}(j)$ obtained from subtracting H from each entry is

$$\begin{aligned}\Delta f_-^{(i-1)}(j) &= f(\hat{\mathbf{w}}_{i-1} - H\mathbf{e}_j) - f(\hat{\mathbf{w}}_{i-1}) \\ &= -H\mathbf{e}_j^T \mathbf{R} \hat{\mathbf{w}}_{i-1} + \frac{1}{2} H^2 \mathbf{e}_j^T \mathbf{R} \mathbf{e}_j + H \mathbf{r}_{du}^T \mathbf{e}_j.\end{aligned}\quad (1.11)$$

If no update is made for $j = 1 \dots M$, before repeating the procedure in the next iteration, we decrease the step H by a factor of two (i.e., $H \leftarrow H/2$).

Implemented as described so far, the algorithm has a high computational cost. The evaluation of $\Delta f_+^{(i-1)}(j)$ and $\Delta f_-^{(i-1)}(j)$ at each step and the corresponding update of all entries of $\hat{\mathbf{w}}_i$ would require more than M^2 multiplications. However, by taking advantage of the fact that only one entry of the vector is changed at each iteration and of some properties of hardware implementations to compute the arithmetic operations, the number of operations can be reduced substantially as described next.

To begin with, let us define the following $M \times 1$ residue vector

$$\mathbf{r}_i = \mathbf{r}_{du} - \mathbf{R} \hat{\mathbf{w}}_i, \quad (1.12)$$

where (1.12) denotes the residue of solving the normal equation $\mathbf{R} \mathbf{w} = \mathbf{r}_{du}$ using $\hat{\mathbf{w}}_i$ as an approximation to the solution. By replacing (1.9) in (1.12), we get the following update rule for \mathbf{r}_i at each iteration:

$$\mathbf{r}_i = \begin{cases} \mathbf{r}_{du} - \mathbf{R}(\hat{\mathbf{w}}_{i-1} + H\mathbf{e}_j) = \mathbf{r}_{i-1} - H\mathbf{R}\mathbf{e}_j, & \text{if } \Delta f_+^{(i-1)}(j) < 0, \\ \mathbf{r}_{du} - \mathbf{R}(\hat{\mathbf{w}}_{i-1} - H\mathbf{e}_j) = \mathbf{r}_{i-1} + H\mathbf{R}\mathbf{e}_j, & \text{if } \Delta f_-^{(i-1)}(j) < 0, \\ \mathbf{r}_{i-1}, & \text{otherwise.} \end{cases} \quad (1.13)$$

This still requires M multiplications since H is multiplying the $j + 1$ column of \mathbf{R} . However, if H is defined as 2^k for some integer k , the computation of \mathbf{r}_i in fixed-point arithmetic requires only M shifts and M additions, which is easier to implement in hardware.

Besides the advantage of hardware implementation, another simplification can also be applied in the update process of $\hat{\mathbf{w}}_i$ and \mathbf{r}_i by avoiding the computation of $\Delta f_+^{(i-1)}(j)$ and $\Delta f_-^{(i-1)}(j)$ for each entry. To see this, let us rewrite (1.10) and (1.11) using (1.12) to obtain

$$\Delta f_+^{(i-1)}(j) = -H\mathbf{e}_j^T \mathbf{r}_{i-1} + \frac{1}{2} H^2 \mathbf{e}_j^T \mathbf{R} \mathbf{e}_j = -H r_{i-1}(j) + \frac{1}{2} H^2 R(j, j) \quad (1.14)$$

and

$$\Delta f_-^{(i-1)}(j) = H\mathbf{e}_j^T \mathbf{r}_{i-1} + \frac{1}{2}H^2\mathbf{e}_j^T \mathbf{R}\mathbf{e}_j = Hr_{i-1}(j) + \frac{1}{2}H^2R(j, j). \quad (1.15)$$

Since H and $R(j, j)$ are positive (\mathbf{R} is positive-definite), the condition $\Delta f_+^{(i-1)}(j) < 0$ only holds if

$$r_{i-1}(j) > \frac{1}{2}HR(j, j).$$

Similarly, the condition for $\Delta f_-^{(i-1)}(j) < 0$ is

$$r_{i-1}(j) < -\frac{1}{2}HR(j, j).$$

Given the previously conditions, we see that the j entry of $\hat{\mathbf{w}}_i$ will be updated if and only if

$$|r_{i-1}(j)| > \frac{1}{2}HR(j, j), \quad (1.16)$$

with updates

$$\hat{\mathbf{w}}_i = \hat{\mathbf{w}}_{i-1} + H \text{sign}\{r_{i-1}(j)\}\mathbf{e}_j \quad (1.17)$$

$$\mathbf{r}_i = \mathbf{r}_{i-1} - H \text{sign}\{r_{i-1}(j)\}\mathbf{R}\mathbf{e}_j. \quad (1.18)$$

As shown in [13], the optimal solution \mathbf{w}^o for (1.8) can be obtained through the DCD procedure described in Algorithm 1 using fixed-point arithmetic for hardware implementation. In this algorithm, the entries of $\hat{\mathbf{w}}_i$ are updated bit by bit considering B bits assigned to each coefficient.

Algorithm 1: DCD algorithm for minimization of function (1.8)

Data:

Initialization:

$$\hat{\mathbf{w}}_{-1} = \mathbf{0}, \mathbf{r}_{-1} = \mathbf{r}_{du}, h = H, \text{cont} = 0 \text{ and } i = 0;$$

```

1 for  $b = 0 : B - 1$  do // Bit by bit loop
2    $flag = 1$  // Update identifier
3   while  $flag = 1$  do // Iteration loop
4      $flag = 0$ 
5     for  $j = 1 : M$  do //  $j$ -entry loop update
6       if  $|r_{i-1}(j)| > \frac{1}{2}hR(j, j)$  then // Condition (1.16)
7          $\hat{\mathbf{w}}_i = \hat{\mathbf{w}}_{i-1} + h\text{sign}\{r(j)\}\mathbf{I}_{1:M,j}$  // See (1.17)
8          $\mathbf{r}_i = \mathbf{r}_{i-1} - h\text{sign}\{r(j)\}\mathbf{R}_{1:M,j}$  // See (1.18)
9          $cont = cont + 1$  // Update counter
10         $flag = 1$ 
11        if  $cont > N_u$  then
12          Stop the algorithm
13         $i = i + 1$ 
14     $h = h/2$ 

```

Given an initial approximation $\hat{\mathbf{w}}_0$, the algorithm cyclically updates each element of $\hat{\mathbf{w}}_i$ using the procedure described before ($j = 1, \dots, M$) from the most significant bits towards the less significant bit. Due to the quantized step-size H , there are iterations at step 6 without update of the solution (requiring a reduction in the step-size by half in step 14, these iterations are called ‘unsuccessful’). On the other hand, in ‘successful’ iterations the solution $\hat{\mathbf{w}}(j)$ and the residual vector \mathbf{r} are updated according to steps 7 and 8. In order to have an efficient algorithm, the number of successful updates N_u must be small.

In Algorithm 1, b is the index of the binary representation of the entries of $\hat{\mathbf{w}}_i$, M is the length of vector $\hat{\mathbf{w}}_i$, H is a power of two step-size, $cont$ is a counter variable of ‘successful’ iterations, N_u is the maximum number of ‘successful’ iterations before the algorithm stops and \mathbf{r}_i is the residue vector. The entries of $\hat{\mathbf{w}}_i$ are updated cyclically from the element $j = 1$ until M . The notations $\mathbf{I}_{1:M,j}$ and $\mathbf{R}_{1:M,j}$ correspond to the elements from rows 1 through M of column j of the identity matrix and \mathbf{R} , respectively. Note that we used normal font for $R(j, j)$ in line 6 because this is a scalar variable that corresponds to the element (j, j) of the matrix \mathbf{R} . The resulting computational complexity of this algorithm is $\mathcal{O}(M(2N_u + B - 1) + N_u)$ [3].

If the number of updates N_u is larger than B , the complexity of the DCD algorithm is approximately upper bounded by $2MN_u$. However, if the number of updates N_u is small, the term MB will dominate in the DCD complexity. In this case, a computationally more efficient variant of the DCD algorithm was proposed in [3] in order to eliminate this term. This algorithm finds a ‘leading’ element (i.e., the element with largest residue) in $\Delta\hat{\mathbf{w}}_i = \mathbf{w}_i - \hat{\mathbf{w}}_{i-1}$ to be updated at each iteration. Algorithm 2 describes the efficient DCD version using the leading element.

Algorithm 2: Leading DCD algorithm

Data:

Initialization:

$$\Delta\hat{\mathbf{w}}_{-1} = \mathbf{0}, \mathbf{r}_{-1} = \mathbf{r}_{du}, h = H, m = 1;$$

```

1 for  $i = 0, \dots, N_u - 1$  do
2    $p = \operatorname{argmax}_{j=1, \dots, M} \{|r_{i-1}(j)|\}$ 
3   while  $|r_{i-1}(p)| \leq \frac{1}{2}hR(p, p)$  do
4      $m = m + 1$ 
5      $h = h/2$ 
6     if  $m > B$  then
7       Algorithm stops.
8    $\Delta\hat{\mathbf{w}}_i = \Delta\hat{\mathbf{w}}_{i-1} + \operatorname{sign}\{r_{i-1}(p)\}h\mathbf{I}_{1:M,p}$ 
9    $\mathbf{r}_i = \mathbf{r}_{i-1} - \operatorname{sign}\{r_{i-1}(p)\}h\mathbf{R}_{1:M,p}$ 

```

As shown in [3], the complexity of the leading DCD algorithm is upper bounded by $(2M + 1)N_u + B$ additions. This corresponds to a worst case scenario when the algorithm makes use of all N_u updates and the condition at step 6 in Algorithm 2 is never satisfied [3].

As we mentioned in the beginning of this section, the RLS-DCD algorithm uses the DCD technique to solve the normal equation given by (1.3) as follows: Assume that \mathbf{w}_i denotes the exact solution of (1.3) obtained via RLS and $\hat{\mathbf{w}}_i$ is the approximate solution obtained via RLS-DCD. Then, by replacing \mathbf{R} and \mathbf{r}_{du} by their respective approximations (1.4) and (1.5) in (1.12), we get the following expression for the residue vector at the time instant $i - 1$

$$\mathbf{r}_{i-1} = \hat{\mathbf{r}}_{du_{i-1}} - \hat{\mathbf{R}}_{i-1} \hat{\mathbf{w}}_{i-1}. \quad (1.19)$$

In order to apply the DCD algorithm to solve for $\hat{\mathbf{w}}_i$, we define the following difference equations:

$$\Delta \mathbf{w}_i = \mathbf{w}_i - \hat{\mathbf{w}}_{i-1}, \quad (1.20a)$$

$$\Delta \hat{\mathbf{R}}_i = \hat{\mathbf{R}}_i - \hat{\mathbf{R}}_{i-1}, \quad (1.20b)$$

$$\Delta \hat{\mathbf{r}}_{du_i} = \hat{\mathbf{r}}_{du_i} - \hat{\mathbf{r}}_{du_{i-1}}. \quad (1.20c)$$

Using the relations of (1.20), the normal equation (1.3) becomes

$$\hat{\mathbf{R}}_i [\hat{\mathbf{w}}_{i-1} + \Delta \mathbf{w}_i] = \hat{\mathbf{r}}_{du_i},$$

and thus

$$\begin{aligned} \hat{\mathbf{R}}_i \Delta \mathbf{w}_i &= \hat{\mathbf{r}}_{du_{i-1}} - \hat{\mathbf{R}}_{i-1} \hat{\mathbf{w}}_{i-1} + \Delta \hat{\mathbf{r}}_{du_i} - \Delta \hat{\mathbf{R}}_i \hat{\mathbf{w}}_{i-1} \\ &= \mathbf{r}_{i-1} + \Delta \hat{\mathbf{r}}_{du_i} - \Delta \hat{\mathbf{R}}_i \hat{\mathbf{w}}_{i-1} \triangleq \boldsymbol{\beta}_i. \end{aligned} \quad (1.21)$$

From equations (1.4) and (1.5), the evaluation of (1.20b) and (1.20c) are given by

$$\Delta \hat{\mathbf{R}}_i = (\lambda - 1) \hat{\mathbf{R}}_{i-1} + \mathbf{u}_i \mathbf{u}_i^T, \quad (1.22a)$$

$$\Delta \hat{\mathbf{r}}_{du_i} = (\lambda - 1) \hat{\mathbf{r}}_{du_{i-1}} + d(i) \mathbf{u}_i. \quad (1.22b)$$

Replacing the results of (1.19) and (1.22) in $\boldsymbol{\beta}_i$, we obtain

$$\begin{aligned} \boldsymbol{\beta}_i &= \hat{\mathbf{r}}_{du_{i-1}} - \hat{\mathbf{R}}_{i-1} \hat{\mathbf{w}}_{i-1} + (\lambda - 1) \hat{\mathbf{r}}_{du_{i-1}} + d(i) \mathbf{u}_i - \left[(\lambda - 1) \hat{\mathbf{R}}_{i-1} + \mathbf{u}_i \mathbf{u}_i^T \right] \hat{\mathbf{w}}_{i-1} \\ &= \lambda \left(\hat{\mathbf{r}}_{du_{i-1}} - \hat{\mathbf{R}}_{i-1} \hat{\mathbf{w}}_{i-1} \right) + \mathbf{u}_i \left[d(i) - \mathbf{u}_i^T \hat{\mathbf{w}}_{i-1} \right] \\ &= \lambda \mathbf{r}_{i-1} + \mathbf{u}_i e(i). \end{aligned} \quad (1.23)$$

Given an approximate solution $\Delta\hat{\mathbf{w}}_i$ to (1.20a), we obtain an updated approximate solution $\hat{\mathbf{w}}_i = \hat{\mathbf{w}}_{i-1} + \Delta\hat{\mathbf{w}}_i$ with the corresponding residue vector at time i

$$\begin{aligned}
\mathbf{r}_i &= \hat{\mathbf{r}}_{du_i} - \hat{\mathbf{R}}_i \hat{\mathbf{w}}_i \\
&= \hat{\mathbf{r}}_{du_{i-1}} + \Delta\hat{\mathbf{r}}_{du_i} - \hat{\mathbf{R}}_i [\hat{\mathbf{w}}_{i-1} + \Delta\hat{\mathbf{w}}_i] \\
&= \hat{\mathbf{r}}_{du_{i-1}} + \Delta\hat{\mathbf{r}}_{du_i} - \hat{\mathbf{R}}_i \Delta\hat{\mathbf{w}}_i - \left[\hat{\mathbf{R}}_{i-1} + \Delta\hat{\mathbf{R}}_i \right] \hat{\mathbf{w}}_{i-1} \\
&= \underbrace{\mathbf{r}_{i-1} + \Delta\hat{\mathbf{r}}_{du_i} - \Delta\hat{\mathbf{R}}_i \hat{\mathbf{w}}_{i-1}}_{= \boldsymbol{\beta}_i \text{ from (1.21)}} - \hat{\mathbf{R}}_i \Delta\hat{\mathbf{w}}_i \\
&= \boldsymbol{\beta}_i - \hat{\mathbf{R}}_i \Delta\hat{\mathbf{w}}_i.
\end{aligned} \tag{1.24}$$

As shown in [13], the DCD algorithm can then be used to compute an approximate solution $\Delta\hat{\mathbf{w}}_i$ to (1.21) and update the residue (1.24) using the RLS-DCD routine described in Algorithm 3.

Algorithm 3: RLS-DCD algorithm

Data:

Initialization:

$$\hat{\mathbf{w}}_{-1} = \mathbf{0}, \mathbf{r}_{-1} = \mathbf{0}, \hat{\mathbf{R}}_{-1} = \mathbf{\Pi} > 0;$$

1 **for** $i = 0, 1, 2 \dots$ **do**

$$\begin{array}{ll}
2 & \left[\begin{array}{l} \hat{\mathbf{R}}_i = \lambda\hat{\mathbf{R}}_{i-1} + \mathbf{u}_i\mathbf{u}_i^T \\ \hat{\mathbf{y}}(i) = \mathbf{u}_i^T \hat{\mathbf{w}}_{i-1} \\ e(i) = d(i) - \hat{\mathbf{y}}(i) \\ \boldsymbol{\beta}_i = \lambda\mathbf{r}_{i-1} + \mathbf{u}_i e(i) \\ \hat{\mathbf{R}}_i \Delta\hat{\mathbf{w}}_i = \boldsymbol{\beta}_i \\ \hat{\mathbf{w}}_i = \hat{\mathbf{w}}_{i-1} + \Delta\hat{\mathbf{w}}_i \end{array} \right. \quad \begin{array}{l} // \text{ See (1.4)} \\ \\ \\ // \text{ See (1.23)} \\ // \text{ Use algorithm 1 or 2 to compute } \Delta\hat{\mathbf{w}}_i \end{array} \\
3 & \\
4 & \\
5 & \\
6 & \\
7 & \end{array}$$

As explained in [13], the resulting complexity of Algorithm 3 can be substantially reduced when \mathbf{u}_i is modeled as a tap-delay line. In this particular case the matrices $\hat{\mathbf{R}}_i$ and $\hat{\mathbf{R}}_{i-1}$ share a common structure and so the computation of $\hat{\mathbf{R}}_i$ in Algorithm 3 can be replaced by an update of only its first column $[\hat{\mathbf{R}}_i]_{1:M,1}$, that is

$$[\hat{\mathbf{R}}_i]_{1:M,1} = \lambda[\hat{\mathbf{R}}_{i-1}]_{1:M,1} + x(i)\mathbf{u}_i. \tag{1.25}$$

Since $\hat{\mathbf{R}}_i$ is symmetric, its first row is equal to the transpose of (1.25). The remaining terms are equal to the top-left $(M-1) \times (M-1)$ block matrix $[\hat{\mathbf{R}}_{i-1}]_{1:M-1,1:M-1}$. Using this procedure, the resulting complexity of the RLS-DCD is $\mathcal{O}(2MN_u + 6M)$.

In Chapter 4 we show a numerical example in which the DCD technique is used to derive a fast (i.e. $\mathcal{O}(M)$) version of the Kalman filter using an specific model for the evolution of the optimum weight vector \mathbf{w}_i^o .

1.4 Convex Combination

When the a priori knowledge about the filtering scenario is limited or imprecise, selecting the most adequate filter and adjusting its parameters becomes a challenging task, and erroneous choices can lead to suboptimal performance. To address this difficulty, one useful approach is to rely on the use of combinations of adaptive structures.

Combinations of adaptive filters have received considerable attention lately, since they decrease the sensitivity of the filter to choices of parameters such as the step-size, forgetting factor or filter length [13]. The idea is to combine the outputs of two (or several) different independently-run adaptive algorithms to achieve better performance than that of a single filter. In general, this approach is more robust than variable parameter schemes [14].

Due to its relative simplicity, since the operating rules of the combination schemes are no more complicated than those of the individual component filters, the convex combination of adaptive filters was the first combined scheme that attracted attention. As shown in [15], it can be proved that the optimum combination is universal, i.e., for stationary inputs the optimum combined estimate is at least as good as the best of the component filters in steady-state.

Several applications for combination of adaptive filters have been proposed, such as: acoustic echo cancellation [16], adaptive line enhancement [17], array beamforming [18], and active noise control [19]. Figure 3 illustrates a convex combination structure between two adaptive filters used to estimate a given desired variable $d(i)$ based on the observation of an input regressor vector \mathbf{u}_i .

As is seen in Figure 3, combination schemes can be divided in the following two concurrent adaptive sections: one section for the adaptation of the individual filters and a second one for the adaptation of the overall combination structure.

As shown in this figure, both adaptive filters have access to the same input and reference signals and produce their individual estimates of the optimum weight vector \mathbf{w}_i^o . Based on each filter output, the combination layer aims to learn dynamically the best weights to optimize the overall performance.

According to Figure 3, the output of the convex combination is given by

$$y(i) = \eta(i)y^{(1)}(i) + [1 - \eta(i)]y^{(2)}(i), \quad (1.26)$$

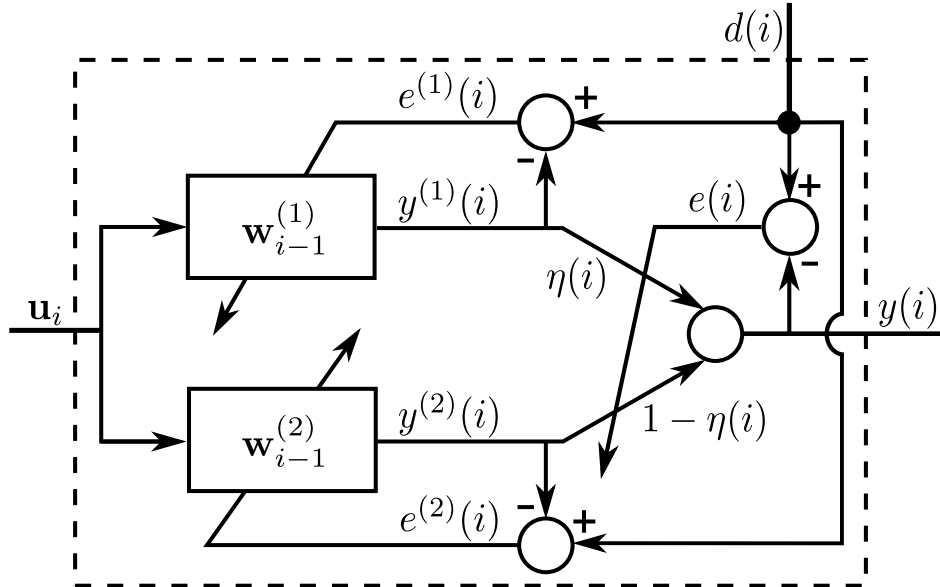


Figure 3: Convex combination of two transversal adaptive filters.

where $\eta(i)$ is the mixing parameter that lies in $[0, 1]$, $y^{(n)}(i) = \mathbf{u}_i^T \mathbf{w}_{i-1}^{(n)}$, for $n = 1, 2$, are the outputs of the transversal filters, \mathbf{u}_i is the input regressor vector and $\mathbf{w}_{i-1}^{(n)}$ are the weight vectors of the component filters. The weight vector and the estimation error of the overall filter are given respectively by

$$\mathbf{w}_i = \eta(i)\mathbf{w}_i^{(1)} + [1 - \eta(i)]\mathbf{w}_i^{(2)}, \quad (1.27)$$

and

$$e(i) = d(i) - y(i) = \eta(i)e^{(1)}(i) + [1 - \eta(i)]e^{(2)}(i). \quad (1.28)$$

In order to reduce the gradient noise when $\eta(i) \approx 0$ or $\eta(i) \approx 1$ and to ensure that $\eta(i)$ will remain between $[0, 1]$, a nonlinear transformation of an auxiliary variable $a(i)$ was proposed in [20]

$$\eta(i) = \frac{\text{sgm}[a(i)] - \text{sgm}[-a_+]}{\text{sgm}[a_+] - \text{sgm}[-a_+]}, \quad (1.29)$$

where $\text{sgm}(a) = \frac{1}{1+e^{-a}}$, $a(i)$ is restricted to an interval $[-a_+, a_+]$ in order to avoid that the adaptation of $a(i)$ (see (1.30) below) slows down too much when $\eta(i)$ is close to 0 or to 1 because of the factor $\text{sgm}[a(i)]\{1 - \text{sgm}[a(i)]\}$ [21]:

$$a(i+1) = a(i) + \frac{\mu_a}{\epsilon + p(i)} e(i)[y^{(1)}(i) - y^{(2)}(i)] \text{sgm}[a(i)]\{1 - \text{sgm}[a(i)]\}. \quad (1.30)$$

Here, ϵ is a small positive number and $p(i)$ is a low-pass filtered estimate of the power of $[y^{(1)}(i) - y^{(2)}(i)]$ defined by

$$p(i) = \gamma p(i-1) + (1 - \gamma)[y^{(1)}(i) - y^{(2)}(i)]^2, \quad (1.31)$$

with $0 \ll \gamma < 1$.

The step-size μ_a usually will be chosen in the interval $[0.01, 1]$. As mentioned in [22], a common choice in practice for a_+ is 4.

It can be shown that the optimum mixing parameter in steady state is given by [15,23]:

$$\eta^\circ = \frac{\zeta^{(2)} - \zeta^{(12)}}{\zeta^{(1)} - 2\zeta^{(12)} + \zeta^{(2)}} \Bigg|_0^1, \quad (1.32)$$

where $\zeta^{(n)}$ is the steady-state excess mean-square error (EMSE) for each component filter, given by

$$\zeta^{(n)} = \lim_{i \rightarrow \infty} \mathbb{E}\{[e_a^{(n)}(i)]^2\}, \quad \text{for } n = 1, 2, \quad (1.33)$$

$e_a^{(n)}(i)$ is the *a priori* error given by

$$e_a^{(n)}(i) = \mathbf{u}_i^T \tilde{\mathbf{w}}_{i-1}^{(n)}, \quad (1.34)$$

with $\tilde{\mathbf{w}}_{i-1}^{(n)} = \mathbf{w}_{i-1}^\circ - \mathbf{w}_{i-1}^{(n)}$ and $\zeta^{(12)}$ is the cross EMSE between both filters in the convex combination which is obtained according to

$$\zeta^{(12)} = \lim_{i \rightarrow \infty} \mathbb{E}\{e_a^{(1)}(i)e_a^{(2)}(i)\}. \quad (1.35)$$

In equations (1.33) and (1.35), the *a priori* error $e_a(i)$ of the overall combination structure can be written as a function of the *a priori* errors of the component filters [14], i.e.,

$$e_a(i) = \eta(i)e_a^{(1)}(i) + [1 - \eta(i)]e_a^{(2)}(i). \quad (1.36)$$

As mentioned in [22], the optimum EMSE for the convex combination between two adaptive filters, using the optimum mixing parameter η° , is given by

$$\zeta^{\text{COMB}} = \zeta^{(12)} + \frac{\Delta\zeta^{(1)}\Delta\zeta^{(2)}}{\Delta\zeta^{(1)} + \Delta\zeta^{(2)}} \quad (1.37a)$$

$$= \zeta^{(1)} - [1 - \eta^\circ]\Delta\zeta^{(1)} \quad (1.37b)$$

$$= \zeta^{(2)} - \eta^\circ\Delta\zeta^{(2)}, \quad (1.37c)$$

where $\Delta\zeta^{(n)} = \zeta^{(n)} - \zeta^{(12)}$, for $n = 1, 2$.

According to (1.37), when the optimum mixing parameter is either 1 or 0, the convex combination behaves just like the best filter component. However, in the case where $0 < \eta^\circ < 1$ and the cross-EMSE $\zeta^{(12)} < \min\{\zeta^{(1)}, \zeta^{(2)}\}$, it can be seen that the combination outperforms both component filters simultaneously.

In chapters 2 and 3 we show numerical examples using the convex combination between LMS and RLS filters under two different models, one for each chapter, for the evolution of the optimum weight vector \mathbf{w}_i^o , and compare the resulting performance with the optimum solution obtained via Kalman filter.

1.5 Kalman Filter

The Kalman filter (KF) is the Bayesian solution to the problem of sequentially estimating the states of a dynamical system in which the state evolution and measurement processes are both linear and Gaussian [24]. This filter has long been used to derive approximations to the optimal solution to many tracking and data prediction tasks, such as: linear estimation [25], target tracking [26] and navigation systems [27].

This filter is optimum in the sense that it minimizes the mean square error of the estimated parameters when all noises involved are Gaussian and the parameter vector to be estimated changes according to a linear model [28].

Thus, assume that we want to estimate a vector \mathbf{x}_i that changes in time according to the model described by the following set of state-space equations:

$$\begin{aligned}\mathbf{x}_i &= \mathbf{F}_i \mathbf{x}_{i-1} + \mathbf{G}_i \mathbf{t}_i, \\ \mathbf{z}_i &= \mathbf{H}_i \mathbf{x}_{i-1} + \mathbf{v}_i,\end{aligned}\tag{1.38}$$

where \mathbf{x}_i is the unknown $M \times 1$ state vector, \mathbf{F}_i is the $M \times M$ state-transition matrix, \mathbf{G}_i is an $M \times N$ matrix, \mathbf{t}_i is an $N \times 1$ real Gaussian random state noise vector with zero mean and covariance matrix \mathbf{T}_i , \mathbf{z}_i is the $D \times 1$ observation vector, \mathbf{H}_i is the $D \times M$ measurement matrix, and \mathbf{v}_i is a $D \times 1$ real Gaussian random measurement noise vector with zero mean and covariance matrix \mathbf{V}_i . Matrices \mathbf{F}_i , \mathbf{G}_i , \mathbf{H}_i , \mathbf{T}_i and \mathbf{V}_i are assumed to be real and known.

The covariances and cross-covariances of the random processes \mathbf{t}_i and \mathbf{v}_i are denoted by:

$$\mathbb{E} \left\{ \begin{bmatrix} \mathbf{t}_i \\ \mathbf{v}_i \end{bmatrix} \begin{bmatrix} \mathbf{t}_j \\ \mathbf{v}_j \end{bmatrix}^T \right\} = \begin{bmatrix} \mathbf{T}_i & \mathbf{S}_i \\ \mathbf{S}_i^T & \mathbf{V}_i \end{bmatrix} \delta_{ij},\tag{1.39}$$

where

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j. \end{cases}$$

The initial state \mathbf{x}_{-1} is assumed to have zero mean, covariance matrix $\mathbf{\Pi}_{-1}$ and to be uncorrelated with \mathbf{t}_i and \mathbf{v}_i , i.e:

$$\mathbb{E}\{\mathbf{x}_{-1}\mathbf{x}_{-1}^T\} = \mathbf{\Pi}_{-1} > 0, \quad \mathbb{E}\{\mathbf{t}_i\mathbf{x}_{-1}^T\} = \mathbf{0} \quad \text{and} \quad \mathbb{E}\{\mathbf{v}_i\mathbf{x}_{-1}^T\} = \mathbf{0}, \quad \text{for } i \geq 0. \quad (1.40)$$

Given observations \mathbf{z}_i that satisfy the state-space model described in (1.38), an estimate $\hat{\mathbf{x}}_i$ for \mathbf{x}_i can be recursively computed by using the following set of KF equations [2]:

$$\mathbf{\Omega}_i = \mathbf{V}_i + \mathbf{H}_i\mathcal{P}_{i-1}\mathbf{H}_i^T, \quad (1.41a)$$

$$\mathbf{K}_i = (\mathbf{F}_i\mathcal{P}_{i-1}\mathbf{H}_i^T + \mathbf{G}_i\mathcal{S}_i)\mathbf{\Omega}_i^{-1}, \quad (1.41b)$$

$$\mathbf{e}_i = \mathbf{z}_i - \mathbf{H}_i\hat{\mathbf{x}}_{i-1}, \quad (1.41c)$$

$$\hat{\mathbf{x}}_i = \mathbf{F}_i\hat{\mathbf{x}}_{i-1} + \mathbf{K}_i\mathbf{e}_i, \quad (1.41d)$$

$$\mathcal{P}_i = \mathbf{F}_i\mathcal{P}_{i-1}\mathbf{F}_i^T + \mathbf{G}_i\mathbf{T}_i\mathbf{G}_i^T - \mathbf{K}_i\mathbf{\Omega}_i\mathbf{K}_i^T, \quad (1.41e)$$

where $\hat{\mathbf{x}}_i$ is the optimal estimate considering the observations $\{z(0), z(1) \dots, z(i-1), z(i)\}$ and \mathcal{P}_i is the $M \times M$ error covariance matrix, i.e.,

$$\mathcal{P}_i = \mathbb{E}\{(\mathbf{x}_i - \hat{\mathbf{x}}_i)(\mathbf{x}_i - \hat{\mathbf{x}}_i)^T\}, \quad (1.42)$$

$\mathbb{E}\{\cdot\}$ denotes expectation, \mathbf{K}_i is the $M \times D$ Kalman gain, $\mathcal{S}_i = \mathbb{E}\{\mathbf{t}_i\mathbf{v}_i^T\}$ and \mathbf{e}_i is a $D \times 1$ error vector.

In chapters 2 and 3 we show numerical examples using the Kalman filter equations under different state-space models, one for each chapter, and compare the resulting performance with the LMS and RLS algorithms, as well as with their convex combination. In chapter 4 we present a numerical example in which the DCD technique is used to derive a fast (i.e. $\mathcal{O}(M)$) version of the Kalman filter using a specific model for the evolution of the state vector \mathbf{x}_i .

1.6 Contributions

The main contributions of the present dissertation are:

- Chapter 2: Comparison of tracking performance between the convex combination of LMS and RLS filters with the optimal solution obtained via Kalman filter. We show that combination schemes may have a tracking performance close to that of a Kalman filter, but with lower computational complexity (linear in the filter length

instead of quadratic — in the case of the example shown in chapter 2 — or cubic, for general Kalman models).

- Chapter 3: Analysis of the fastest speed of change of the optimum parameter vector that an adaptive filter can track and comparison between the theoretical steady-state EMSE and MSD of the convex combination of LMS and RLS filters with the optimal solution obtained via Kalman filter for a more general state-space model than in Chapter 2. We show that, as long as the pole of the AR model used in chapter 3 is greater than a minimum value, the performance of the low-cost approach using convex combination is close to the optimal case obtained via Kalman filter. In addition, we present a theoretical mean-square analysis for the Kalman filter with a random measurement matrix.
- Chapter 4: Derivation of a low-cost recursion for the covariance matrix \mathcal{P}_i in the Kalman filter, using the Dichotomous Coordinate Descent (DCD) technique. We show that it is possible to reduce the complexity of the Kalman filter from $\mathcal{O}(M^2)$ to $\mathcal{O}(M)$ for a certain class of diagonal matrices \mathbf{F}_i .

This research gave rise to 4 publications, which are:

1. Raffaello Claser, Vítor H. Nascimento, and Yuriy V Zakharov. A low-complexity RLS-DCD algorithm for Volterra system identification. In 2016 24th European Signal Processing Conference (EUSIPCO), pages 6–10. IEEE, 2016. Since this article is outside the scope of this dissertation, this work was not included in the final text.
2. Raffaello Claser and Vítor H Nascimento. Low-complexity approximation to the Kalman filter using convex combinations of adaptive filters from different families. In 2017 25th European Signal Processing Conference (EUSIPCO), pages 2630–2633. IEEE, 2017.
3. Raffaello Claser, Vítor H. Nascimento, and Yuriy V Zakharov. Low-complexity approximation to the Kalman filter using the Dichotomous Coordinate Descent Algorithm. In Asilomar Conference on Signals, Systems, and Computers, 2018.
4. Raffaello Claser and Vítor H Nascimento. On the tracking performance of adaptive filters. Submitted to the IEEE Transactions on Signal Processing.

2 LOW-COMPLEXITY APPROXIMATION TO THE KALMAN FILTER USING CONVEX COMBINATIONS OF ADAPTIVE FILTERS FROM DIFFERENT FAMILIES

When choosing an adaptive algorithm for a given application, one of the important points to be considered is the algorithm's ability to track variations in the parameter vector one wishes to estimate [14]. The Kalman filter (KF) has long been shown to be the optimal solution to many tracking and data prediction tasks [28], in the sense it minimizes the mean square error of the estimated parameters when all noises involved are Gaussian and the parameter vector to be estimated changes following a linear model [28] (see chapter 1, section 1.5).

When a combination of two adaptive filters of the same family is used, for example two least mean-squares (LMS) with different step sizes, or two recursive least-squares (RLS) with different forgetting factors, the resulting performance will never be better than the performance of a single filter using optimum settings for a given nonstationary condition [22].

On the other hand, as was shown in [14, 22], when combining filters from different families, namely LMS and RLS, it is possible to take advantage of the tracking properties from each filter and obtain a structure with better performance than that which would be possible with each filter, even if both were implemented using the optimum values of step-size and forgetting factor. Combinations of Kalman filters were also proposed using different update rules as proposed in [29] and [30].

Assuming we want to estimate a vector with M parameters, the computational complexity for a convex combination between one LMS and one RLS can be implemented with $\mathcal{O}(M)$ operations when the regressor vector is a tap delay line (if lattice or Dichotomous Coordinate Descent - DCD algorithms are used) while the Kalman filter requires $\mathcal{O}(M^2)$ operations (for a first-order random walk state-space model, see (2.2) below), or $\mathcal{O}(M^{2.376})$ [31] for a general state-space model. This chapter describes how close the combination

scheme can get to the optimal excess mean square error (EMSE) obtained via Kalman filter. We show that the performance gain obtained with the Kalman filter is not very large, less than 1dB, even when we do not have exact knowledge of the true covariance matrix of the noise process.

2.1 Problem Formulation

The Kalman filter is the optimal linear least-mean-squares (l.l.m.s.) solution to the problem of sequentially estimating the states of a dynamical system in which the state evolution and measurement processes are both linear and Gaussian [24]. Therefore, for the KF equations described in (1.41), in general one needs $\mathcal{O}(M^3)$ operations to compute the Kalman gain \mathbf{K}_i and the covariance matrix \mathcal{P}_i . Depending on the application, this computational cost may be prohibitive. We propose here to use a combination of an LMS and an RLS instead, and show that, in the case of the model usually employed to study tracking of adaptive filters, the optimum solution obtained with the KF is only slightly better than the result obtained with the combination.

2.2 Data Model

In the sequel we adopt the following assumptions.

- $d(i)$ and \mathbf{u}_i are related according to the following linear regression model

$$d(i) = \mathbf{u}_i^T \mathbf{w}_{i-1}^o + v(i), \quad (2.1)$$

where $v(i)$ is i.i.d. noise, independent of \mathbf{u}_i and with variance σ_v^2 .

- $\mathbf{E}\{\mathbf{u}_i\} = 0$ and $\mathbf{E}\{d(i)\} = 0$.
- \mathbf{w}_i^o changes according to a first-order random walk

$$\mathbf{w}_i^o = \mathbf{w}_{i-1}^o + \mathbf{q}_i, \quad (2.2)$$

where \mathbf{q}_i is independent of \mathbf{u}_i , with autocovariance $\mathbf{Q} = \mathbf{E}\{\mathbf{q}_i \mathbf{q}_i^T\}$.

Note that, due to its simplicity, this is the most usual model used in the literature to study tracking properties of adaptive filters [2]. By comparing the state-space model described in (1.38) with the random-walk described in (2.2) and the linear regression

model described in (2.1), the Kalman model corresponding to (2.2) and (2.1) is such that: \mathbf{x}_i corresponds to \mathbf{w}_i^o , \mathbf{H}_i corresponds to \mathbf{u}_i^T , \mathbf{t}_i corresponds to \mathbf{q}_i , \mathbf{v}_i corresponds to $v(i)$, \mathbf{z}_i corresponds to $d(i)$ and the matrices \mathbf{F}_i and \mathbf{G}_i are equal to \mathbf{I} , the $M \times M$ identity matrix. Under these conditions, the number of matrix multiplications in equations (1.41b) and (1.41e) is reduced, resulting in an $\mathcal{O}(M^2)$ computational complexity for the KF.

As will be shown in section 2.3, once the matrix \mathbf{Q} is known, the excess mean-square error (EMSE) obtained via the KF can be closely approximated by the combination of two adaptive filters (namely LMS and RLS) and their respective parameters, step-size μ and forgetting factor λ , which are chosen optimally according to the equations [2]:

$$\mu^o = \sqrt{\frac{\text{Tr}\{\mathbf{Q}\}}{\sigma_v^2 \text{Tr}\{\mathbf{R}\}}} \quad \text{and} \quad \lambda^o = 1 - \sqrt{\frac{\text{Tr}\{\mathbf{RQ}\}}{\sigma_v^2 M}}, \quad (2.3)$$

where μ^o and λ^o are the optimum tracking parameters.

The advantage is that the combination can be implemented with $\mathcal{O}(M)$ complexity, while the Kalman filter for model (2.2) requires computational complexity $\mathcal{O}(M^2)$. In addition, [15] proposes a method for estimating the optimal step sizes online, which can be useful in cases in which one does not have knowledge of the true value of \mathbf{Q} .

During their operation, adaptive filters normally go from a convergence phase, where the expected error decreases, to a steady-state regime in which the error variance tends towards some asymptotic value [32]. Table 2 presents the theoretical optimum steady-state EMSE expressions for each adaptive filter, and their combination [22].

Table 2: Optimum steady-state EMSEs (ζ^o) for LMS, RLS and their combination.

Alg.	ζ^o
μ^o -LMS	$\sqrt{\sigma_v^2 \text{Tr}\{\mathbf{R}\} \text{Tr}\{\mathbf{Q}\}}$
λ^o -RLS	$\sqrt{\sigma_v^2 M \text{Tr}\{\mathbf{QR}\}}$
Combination	$\frac{\zeta^{(1)}\zeta^{(2)} - (\zeta^{(12)})^2}{\zeta^{(1)} - 2\zeta^{(12)} + \zeta^{(2)}}$

where $\zeta^{(1)} = \zeta_{\text{LMS}}^o$, $\zeta^{(2)} = \zeta_{\text{RLS}}^o$ and $\zeta^{(12)}$ is given by [22]:

$$\zeta^{(12)} = \mu^o \lambda^o \sigma_v^2 \text{Tr}\{\mathbf{\Sigma}\} + \text{Tr}\{\mathbf{Q}\mathbf{\Sigma}\}, \quad (2.4)$$

with $\mathbf{\Sigma} = (\lambda^o \mathbf{I} + \mu^o \mathbf{R})^{-1} \mathbf{R}$.

2.3 Simulations

As shown in [22], when the LMS and RLS filters are combined, an interesting result is obtained. Assuming the tracking model (2.2) and considering the optimum LMS and RLS filters with adaptation parameters given by expressions described in (2.3) [32], LMS will outperform RLS if \mathbf{Q} is proportional to the autocorrelation matrix of the input signal, \mathbf{R} , and the opposite will occur when $\mathbf{Q} \propto \mathbf{R}^{-1}$.

Consider an example where \mathbf{Q} is a mixture of \mathbf{R} and \mathbf{R}^{-1} given by [32]:

$$\mathbf{Q} = 10^{-5} \left\{ \alpha \frac{\mathbf{R}}{\text{Tr}\{\mathbf{R}\}} + (1 - \alpha) \frac{\mathbf{R}^{-1}}{\text{Tr}\{\mathbf{R}^{-1}\}} \right\}, \quad (2.5)$$

where $\alpha \in (0, 1)$.

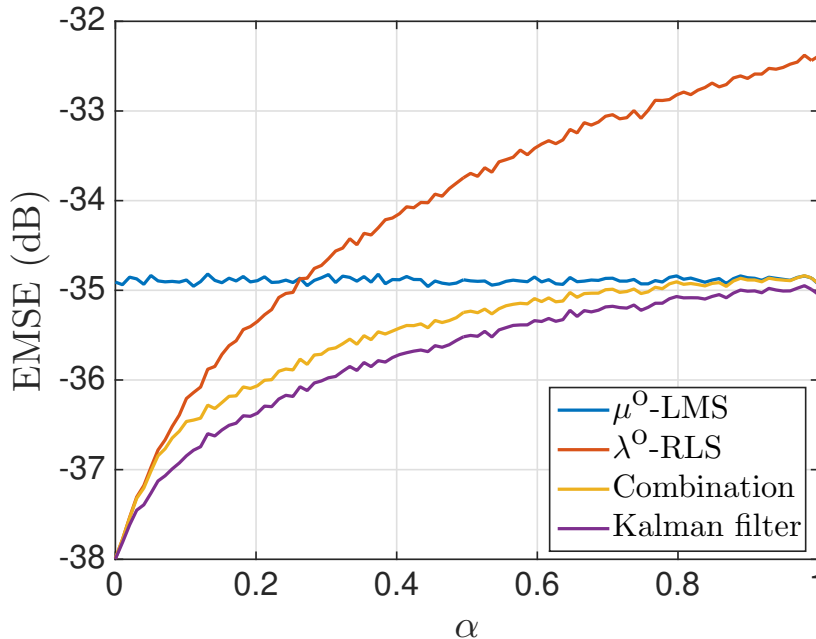


Figure 4: Simulated steady-state EMSE for LMS, RLS, their convex combination and KF, when \mathbf{Q} smoothly changes between \mathbf{R} and \mathbf{R}^{-1} .

As can be seen in Figure 4, the simulated steady-state EMSE that can be achieved by combining both types of filters (LMS and RLS) with optimum settings, is within less than 1dB from the optimum EMSE obtained via KF. However, the computational cost is reduced from $\mathcal{O}(M^2)$ to $\mathcal{O}(M)$ operations (if a lattice or DCD implementation is used for RLS). Other settings considered for this simulation were: \mathbf{w}_i^o was initialized with random values in the interval $[-1, 1]$, $M = 7$, $\sigma_v^2 = 10^{-2}$, \mathbf{R} a Toeplitz matrix with first row given by

$$\frac{1}{7} \begin{bmatrix} 1 & 0.8 & 0.8^2 & \dots & 0.8^6 \end{bmatrix},$$

$\mathbf{F}_i = \mathbf{G}_i = \mathbf{I}$, $\mathbf{H}_i = \mathbf{u}_i^T$, $\mathbf{z}_i = d(i)$ and $\mathbf{V}_i = \sigma_v^2$. Regarding the adjustment for the combinations, we used convex combinations with fixed step-size $\mu_a = 0.25$ and the auxiliary variable $a(i)$ restricted to the interval $[-4, 4]$. For this experiment, 500 simulations with 40.000 iterations each were performed in order to obtain ensemble average EMSE curves for each filter. In this example, μ and λ were chosen optimally according to (2.3).

Even if an $M \times M$ positive-definite random perturbation (with spectral norm 10% of the original \mathbf{Q}) is added to the covariance matrix \mathbf{Q} for each parameter α (according to Algorithm 4), the combination still has an EMSE close to that of the KF (see Figure 5). For this simulation we kept the same settings used to obtain Figure 4.

Algorithm 4: Covariance matrix \mathbf{Q} with random perturbation

```

1 for each  $\alpha$  do
2    $\mathbf{Q} =$  Equation (2.5);
3    $\{\mu^\circ, \lambda^\circ\} =$  Equations (2.3);
4    $\Delta = \text{randn}(M) * 10^{-7};$  // Random perturbation
5    $\mathbf{Q}_p = \mathbf{Q} + \Delta;$ 
6    $[\mathbf{U}', \mathbf{S}', \mathbf{V}'] = \text{svd}(\mathbf{Q}_p);$  // Singular Value Decomp. of  $\mathbf{Q}_p$ 
7   for each realization  $L$  do
8      $\mathbf{w}_0^\circ = \text{randn}(M, 1);$ 
9     for each iteration  $i$  do
10       $\mathbf{q}_i = \mathbf{U}' * \sqrt{\mathbf{S}'} * \text{randn}(M, 1);$ 
11       $\mathbf{w}_i^\circ = \mathbf{w}_{i-1}^\circ + \mathbf{q}_i;$ 
12      Run LMS, RLS, Combination and KF;

```

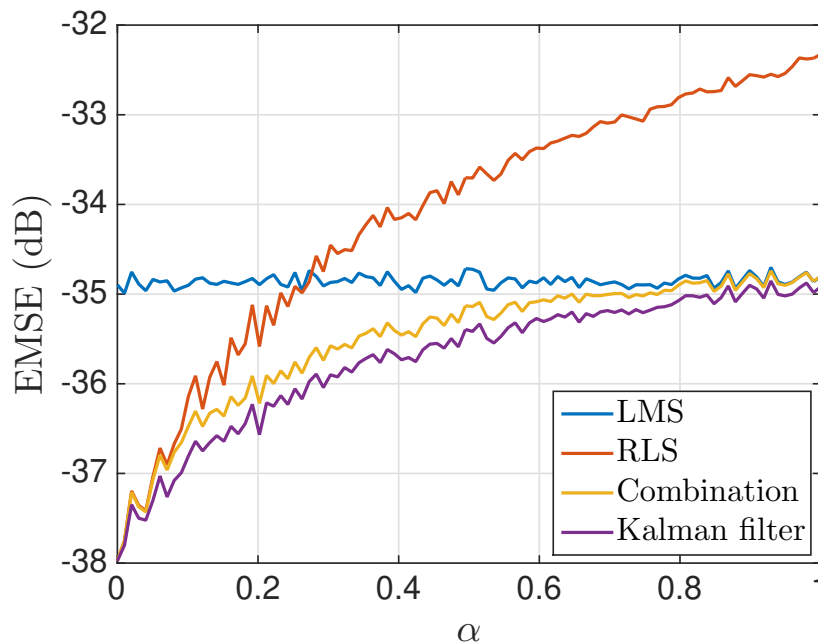


Figure 5: Simulated steady-state EMSE for LMS, RLS, their convex combination and KF, when a random perturbation is added to \mathbf{Q} .

Note that the step-size and forgetting factor were still chosen according to (2.3), but using the nominal value for \mathbf{Q} , so the filters are not operating in the optimal condition anymore. The KF was also designed using the nominal value of \mathbf{Q} (without perturbation).

The analysis presented in this chapter assumed, as commonly used in the literature due to its simplicity, a first order random walk model for \mathbf{w}_i^o . We extend this result to a more general model in the next chapter. By introducing a gain factor in the model of \mathbf{w}_i^o , we are able to study the performance of the filter under faster or slower variations in the optimum parameter vector.

2.4 Conclusion

Combination approaches are an effective way to improve the performance of adaptive filters. In this chapter, proposed originally in [33], we have studied the tracking performance of combinations of LMS and RLS filters and compared the resulting EMSE with the optimal case obtained via Kalman filter.

As it was shown, by using a convex combination between an LMS and an RLS filters, it is possible to achieve a steady-state EMSE performance close to the optimal case obtained via Kalman filter, when a non-stationary environment is considered. The advantage arises from the fact that the combination can be implemented with $\mathcal{O}(M)$ complexity, while it takes at least $\mathcal{O}(M^2)$ operations to compute the corresponding Kalman filter. Similar performance was still obtained, even without precise knowledge of the true value of \mathbf{Q} .

3 ON THE TRACKING PERFORMANCE OF ADAPTIVE FILTERS

3.1 Introduction

When the *a priori* knowledge about the filtering scenario is limited or imprecise, selecting the most adequate filter and adjusting its parameters becomes a challenging task, and erroneous choices can lead to a considerable loss in performance. The Kalman filter (KF) has long been shown to be the optimal solution to many tracking and data prediction tasks [28], in a wide variety of applications ranging from navigation [34, 35] to image processing [36, 37]. This filter is optimal in the sense it minimizes the mean square error of the estimated parameters when all noises involved are Gaussian and the parameter vector to be estimated changes according to a linear model. Thus, consider a state-space description of the form (1.38).

As we can see in equations (1.41b) and (1.41e), in general one needs $\mathcal{O}(M^3)$ operations to compute the Kalman gain and the covariance matrix \mathcal{P}_i or $\mathcal{O}(M^2)$ operations for a first-order random walk state-space model of the form

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{t}_i. \quad (3.1)$$

Depending on the application, this computational cost may be prohibitive. Variations on the Kalman recursions were proposed to reduce the computational complexity. The Schmidt-Kalman filter [38, Ch. 9] and the Chandrasekhar-Kailath-Morf-Sidhu (CKMS) filter [39, Ch. 11 and 13] are able to reduce the total number of arithmetical operations, but with complexity still $\mathcal{O}(M^2)$. Although the hardware technology for embedded systems is quite powerful, a KF with reduced complexity is important to deal with real-time systems that require high sampling rates and low latencies [40]. In the case of active noise control (ANC) [40] and multi-channel linear-prediction (MCLP) for blind speech dereverberation [41], the KF tends to outperform most adaptive algorithms in terms of convergence speed and robustness. By enforcing a band-matrix structure for the covari-

ance matrix \mathcal{P}_i in the case of [40] and block-diagonal matrix in the case of [41], the authors developed low complexity, i.e $\mathcal{O}(M)$, approximations for the KF.

As was shown in chapter 2, through the convex combination between one LMS and one RLS filters (implemented with lattice [42] or Dichotomous Coordinate Descent — DCD [3] algorithms), it is possible to estimate the vector \mathbf{x}_i modeled according to (3.1) with $\mathcal{O}(M)$ operations and with an excess mean-square error (EMSE) $\mathbb{E}\{\|\mathbf{H}_i(\mathbf{x}_{i-1} - \hat{\mathbf{x}}_{i-1})\|^2\}$ less than 1dB from the solution obtained via KF. However, since the covariance matrix of \mathbf{x}_i goes to infinity as $i \rightarrow \infty$, this model is unstable and does not reflect most practical situations. Given this scenario, this chapter studies the tracking behavior of combinations of LMS and RLS filters using a more general model than (3.1) for the evolution of the optimum parameter vector \mathbf{x}_i . Our goal is to describe how close the combination scheme can get to the optimal excess mean-square error (EMSE) and mean-square deviation (MSD) $\mathbb{E}\{\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2\}$ obtained via Kalman filter while keeping the complexity linear in the filter length. In doing so we are able to derive how fast the parameter vector can change so that an adaptive filter can still track the variations.

The contributions of this chapter are:

- Derivation of a theoretical recursion to estimate the covariance matrix \mathcal{P}_i and the corresponding steady-state EMSE and MSD of the Kalman filter considering the data model used in adaptive filtering, in which \mathbf{H}_i is random (an alternate model for the steady-state MSD performance of the Kalman filter was given recently in [43]);
- The derivation of the fastest speed of change of the optimum parameter vector (in a sense to be described later in the text) that an adaptive filter can track. This result enables us to answer when a model-free adaptive filter can be used, and in which situations a model-dependent Kalman filter is necessary. We compare our bound with the nonstationarity degree (NSD) from [44];
- The proposal of convex combinations of adaptive filters as a low-cost approximation to the Kalman filter, and theoretical expressions to quantify the quality of the approximation for the proposed model;
- Derivation of a theoretical model for RLS and the convex combination of one LMS and one RLS filters under the autorregressive model (3.3) for the evolution of the optimum weight vector.

This chapter is organized as follows: in Section 3.2 we derive a general expression for the covariance matrix (3.8) and cross-covariance matrix (3.10) considering algorithms of the general class given by (3.5) and in Section 3.3 we derive the EMSE and MSD for the LMS and RLS algorithms, as well as their respective combination and for the Kalman filter, considering the proposed model (3.3). Section 3.4 compares the performance of each algorithm under different conditions, and finally, section 3.5 concludes the chapter.

3.2 Tracking model

Consider a nonstationary data model relating the random sequences $d(i)$ and \mathbf{u}_i through a linear model of the form

$$d(i) = \mathbf{u}_i^T \mathbf{w}_{i-1}^o + v(i), \quad (3.2)$$

where $v(i)$ is a zero-mean random variable with constant variance $\sigma_v^2 = \mathbb{E}\{v(i)\}^2$ and uncorrelated with \mathbf{u}_i . The weight vector \mathbf{w}_i^o is assumed to evolve according to [45]

$$\mathbf{w}_i^o = \theta \mathbf{w}_{i-1}^o + \sqrt{1 - \theta^2} \mathbf{q}_i, \quad (3.3)$$

where θ is a scalar variable in the range $0 < \theta \leq 1$ and \mathbf{q}_i is a stationary random perturbation independent of the (zero mean) initial conditions $\{\mathbf{w}_{-1}^o, \mathbf{w}_{-1}\}$, of \mathbf{u}_j for all j and of $d(j)$ for all $j < i$. In order to keep the power of the output signal \mathbf{w}_i^o independent of θ and be able to study the EMSE behavior of each filter according to the variation of θ , the model (3.3) was defined in such a way that the power of \mathbf{w}_i^o in the limit is independent of θ , i.e., $\lim_{i \rightarrow \infty} \mathbb{E}\|\mathbf{w}_i^o\|^2 = \mathbb{E}\|\mathbf{q}_i\|^2$. The following assumptions are also considered for this model:

- The sequence $\{\mathbf{u}_i\}$ is real, zero-mean and such that \mathbf{u}_i is independent of \mathbf{u}_j for $i \neq j$ (i.e., the sequence is i.i.d. — independent and identically distributed).
- The noise sequence $v(i)$ is i.i.d. (independent and identically distributed).
- The autocorrelation matrix $\mathbf{R} = \mathbb{E}\{\mathbf{u}_i \mathbf{u}_i^T\}$ is positive-definite ($\mathbf{R} > 0$).
- The random sequences \mathbf{u}_i and $v(i)$ are jointly Gaussian.
- The sequence \mathbf{q}_i is i.i.d., zero-mean and with positive-definite autocorrelation matrix equal to

$$\mathbb{E}\{\mathbf{q}_i \mathbf{q}_i^T\} = \mathbf{Q}. \quad (3.4)$$

- The regressor vector \mathbf{u}_i is independent of the weight-error vector $\tilde{\mathbf{w}}_{i-1}$. This condition is an approximation, part of the widely used independence assumptions in adaptive filter theory [46, 47].

We focus on the convex combination of two algorithms of the following general class [13]:

$$\mathbf{w}_i^{(n)} = \mathbf{w}_{i-1}^{(n)} + \rho^{(n)} \mathbf{M}_i^{(n)} \mathbf{u}_i e^{(n)}(i), \quad (3.5)$$

where the superscript n is associated to the first ($n = 1$) or second ($n = 2$) filter of the combination, $\mathbf{w}_i^{(n)}$ represents the length- M coefficient vector, $\rho^{(n)}$ is a step-size (which is equal to $\mu^{(n)}$ for LMS or 1 for RLS), \mathbf{u}_i is the input regressor vector, $e^{(n)}(i)$ is the estimation error given by $d(i) - \mathbf{u}_i^T \mathbf{w}_{i-1}^{(n)}$ and $\mathbf{M}_i^{(n)}$ is an M -by- M symmetric nonsingular matrix equal to the identity matrix for LMS or, in the RLS case, equal to \mathbf{P}_i

$$\mathbf{P}_i = \mathbf{R}_i^{-1} = \left(\lambda^{i+1} \nu \mathbf{I} + \sum_{\ell=0}^i \lambda^{i-\ell} \mathbf{u}_\ell \mathbf{u}_\ell^T \right)^{-1}, \quad (3.6)$$

where \mathbf{I} denotes the $M \times M$ identity matrix, λ is the forgetting factor and $\nu \mathbf{I}$ is an initial condition to guarantee invertibility.

It is common in the literature to evaluate the steady-state MSD (ε) and EMSE (ζ) for one adaptive filter as

$$\varepsilon = \lim_{i \rightarrow \infty} \text{Tr}\{\mathbf{S}_i\}, \quad \zeta = \lim_{i \rightarrow \infty} \text{Tr}\{\mathbf{R}\mathbf{S}_i\}, \quad (3.7)$$

where $\text{Tr}\{\mathbf{A}\}$ stands for the trace of matrix \mathbf{A} and

$$\mathbf{S}_i = \mathbb{E}\{\tilde{\mathbf{w}}_i \tilde{\mathbf{w}}_i^T\}. \quad (3.8)$$

Recall from chapter 2 that there are three possible situations that can occur in steady-state for the combination: the combination will closely follow LMS or RLS if one of these filters significantly outperforms the other, or, if both component filters have similar performance, the combination may actually outperform both of them [14].

Tracking analyses of convex combinations of the algorithms of the form (3.5) depend on analytical expressions of the cross-MSD and cross-EMSE, defined respectively as $\varepsilon^{(12)}$ and $\zeta^{(12)}$. Using independence assumptions, such expressions can be obtained through the evaluation of

$$\varepsilon^{(12)} = \lim_{i \rightarrow \infty} \text{Tr}\{\mathbf{S}_i^{(12)}\}, \quad \zeta^{(12)} = \lim_{i \rightarrow \infty} \text{Tr}\{\mathbf{R}\mathbf{S}_i^{(12)}\}, \quad (3.9)$$

where

$$\mathbf{S}_i^{(12)} = \mathbb{E}\{\tilde{\mathbf{w}}_i^{(1)}(\tilde{\mathbf{w}}_i^{(2)})^T\}. \quad (3.10)$$

Using the MSD of each filter and the cross-MSD, it is possible to modify (1.32) to obtain the value of the mixing parameter $\eta_{\text{MSD}}^{\circ}$ that minimizes the MSD of the combination. Unfortunately, estimating $\eta_{\text{MSD}}^{\circ}$ online is unrealizable since it requires knowledge of \mathbf{w}_i° to compute the MSD and cross-MSD. So we define the combination MSD using η° from (1.32), which is optimum for the MSE and EMSE and suboptimal for the MSD. Subtracting (1.27) from \mathbf{w}_i° , it can be shown that

$$\varepsilon^{\text{COMB}} = \eta^{\circ 2} \varepsilon^{(1)} + (1 - \eta^{\circ})^2 \varepsilon^{(2)} + 2\eta^{\circ}(1 - \eta^{\circ})\varepsilon^{(12)}. \quad (3.11)$$

The main focus of the following analysis is on the tracking behavior of the adaptive filter in steady-state or, in other words, after initial convergence of the coefficients $\mathbf{w}_i^{(n)}$. Although the optimum weights are time-varying, under model (3.3), the MSD and EMSE approach a steady-state value as seen in section 3.3.

We start the analysis by subtracting both sides of (3.5) from \mathbf{w}_i° , to get

$$\tilde{\mathbf{w}}_i^{(n)} = \mathbf{w}_i^{\circ} - \mathbf{w}_{i-1}^{(n)} - \rho^{(n)} \mathbf{M}_i^{(n)} \mathbf{u}_i e^{(n)}(i). \quad (3.12)$$

Using the linear regression model of (3.2) for the desired response $d(i)$ and the *a priori* error signal $e_a^{(n)}(i)$ given by (1.34), the error signal defined as $e^{(n)}(i) = d(i) - \mathbf{u}_i^T \mathbf{w}_{i-1}^{(n)}$ with $n = 1, 2$, can be rewritten as:

$$e_i^{(n)} = e_a^{(n)}(i) + v(i). \quad (3.13)$$

Replacing the model (3.3) and the error signal $e_i^{(n)}$ given by (3.13) into equation (3.12), we arrive at:

$$\tilde{\mathbf{w}}_i^{(n)} = \theta \mathbf{w}_{i-1}^{\circ} + \sqrt{1 - \theta^2} \mathbf{q}_i - \mathbf{w}_{i-1}^{(n)} - \rho^{(n)} \mathbf{M}_i^{(n)} \mathbf{u}_i [e_a^{(n)}(i) + v(i)]. \quad (3.14)$$

By adding and subtracting $(1 - \theta)\mathbf{w}_{i-1}^{\circ}$ in (3.14) and replacing $e_a^{(n)}(i)$ by (1.34), we get:

$$\tilde{\mathbf{w}}_i^{(n)} = \left[\mathbf{I} - \rho^{(n)} \mathbf{M}_i^{(n)} \mathbf{u}_i \mathbf{u}_i^T \right] \tilde{\mathbf{w}}_{i-1}^{(n)} - \rho^{(n)} \mathbf{M}_i^{(n)} \mathbf{u}_i v(i) - (1 - \theta) \mathbf{w}_{i-1}^{\circ} + \sqrt{1 - \theta^2} \mathbf{q}_i. \quad (3.15)$$

In order to compute the covariance matrices (3.8) and (3.10), multiply (3.15) with $n = \ell$ by its transpose with $n = m$ and take the expectation of both sides. Assuming that

\mathbf{q}_i is independent of the initial conditions and of \mathbf{u}_i , after some algebra, we get

$$\begin{aligned}
\mathbb{E}\{\tilde{\mathbf{w}}_i^{(\ell)}(\tilde{\mathbf{w}}_i^{(m)})^T\} &= \mathbb{E}\{\tilde{\mathbf{w}}_{i-1}^{(\ell)}(\tilde{\mathbf{w}}_{i-1}^{(m)})^T\} \\
&\quad - \rho^{(\ell)}\mathbb{E}\{\mathbf{M}_i^{(\ell)}\mathbf{u}_i\mathbf{u}_i^T\tilde{\mathbf{w}}_{i-1}^{(\ell)}(\tilde{\mathbf{w}}_{i-1}^{(m)})^T\} - \rho^{(m)}\mathbb{E}\{\tilde{\mathbf{w}}_{i-1}^{(\ell)}(\tilde{\mathbf{w}}_{i-1}^{(m)})^T\mathbf{u}_i\mathbf{u}_i^T\mathbf{M}_i^{(m)}\} \\
&\quad - (1-\theta)\mathbb{E}\{\tilde{\mathbf{w}}_{i-1}^{(\ell)}(\mathbf{w}_{i-1}^o)^T\} + \rho^{(\ell)}\rho^{(m)}\mathbb{E}\{\mathbf{M}_i^{(\ell)}\mathbf{u}_i\mathbf{u}_i^T\tilde{\mathbf{w}}_{i-1}^{(\ell)}(\tilde{\mathbf{w}}_{i-1}^{(m)})^T\mathbf{u}_i\mathbf{u}_i^T\mathbf{M}_i^{(m)}\} \\
&\quad - (1-\theta)\mathbb{E}\{\mathbf{w}_{i-1}^o(\tilde{\mathbf{w}}_{i-1}^{(m)})^T\} + (1-\theta)\rho^{(\ell)}\mathbb{E}\{\mathbf{M}_i^{(\ell)}\mathbf{u}_i\mathbf{u}_i^T\tilde{\mathbf{w}}_{i-1}^{(\ell)}(\mathbf{w}_{i-1}^o)^T\} \\
&\quad + \rho^{(\ell)}\rho^{(m)}\sigma_v^2\mathbb{E}\{\mathbf{M}_i^{(\ell)}\mathbf{u}_i\mathbf{u}_i^T\mathbf{M}_i^{(m)}\} + (1-\theta)\rho^{(m)}\mathbb{E}\{\mathbf{w}_{i-1}^o(\tilde{\mathbf{w}}_{i-1}^{(m)})^T\mathbf{u}_i\mathbf{u}_i^T\mathbf{M}_i^{(m)}\} \\
&\quad + (1-\theta)^2\mathbb{E}\{\mathbf{w}_{i-1}^o(\mathbf{w}_{i-1}^o)^T\} + (1-\theta^2)\mathbb{E}\{\mathbf{q}_i\mathbf{q}_i^T\}.
\end{aligned} \tag{3.16}$$

To simplify equation (3.16), we considered the following assumptions commonly used in the literature as mentioned in [13]:

- **Assumption 1:** The regressor vector \mathbf{u}_i is independent of the weight-vector $\tilde{\mathbf{w}}_{i-1}^{(n)}$ for $n = \{\ell, m\}$;
- **Assumption 2:** Matrix $\mathbf{M}_i^{(n)}$ varies slowly in relation to $\tilde{\mathbf{w}}_{i-1}^{(n)}$. Thus, when $\mathbf{M}_i^{(n)}$ appears inside the expectations of (3.16), we simply replace it by its mean $\bar{\mathbf{M}}^{(n)}$. For LMS this assumption is not necessary, since $\mathbf{M}_i^{(n)} = \mathbf{I}$. For RLS, considering large enough i , we can approximate $\mathbb{E}\{\mathbf{M}_i^{(n)}\}$ by $\mathbb{E}\{\mathbf{P}_i\} \approx \bar{\mathbf{P}} \triangleq (1-\lambda)\mathbf{R}^{-1}$.
- **Assumption 3:** According to [13] and relation (3.10), the value of $\mathbb{E}\{\mathbf{M}_i^{(\ell)}\mathbf{u}_i\mathbf{u}_i^T\tilde{\mathbf{w}}_{i-1}^{(\ell)}(\tilde{\mathbf{w}}_{i-1}^{(m)})^T\mathbf{u}_i\mathbf{u}_i^T\mathbf{M}_i^{(m)}\}$ can be approximated by $\bar{\mathbf{M}}^{(\ell)}\left\{\mathbf{R}\text{Tr}\{\mathbf{R}\mathbf{S}_{i-1}^{(\ell m)}\} + 2\mathbf{R}\mathbf{S}_{i-1}^{(\ell m)}\mathbf{R}\right\}\bar{\mathbf{M}}^{(m)}$ when the input regressor vector \mathbf{u}_i is Gaussian and real.

By considering the above assumptions and the relation (3.10), equation (3.16) simplifies to:

$$\begin{aligned}
\mathbf{S}_i^{(\ell m)} &\approx \mathbf{S}_{i-1}^{(\ell m)} - \rho^{(m)}\mathbf{S}_{i-1}^{(\ell m)}\mathbf{R}\bar{\mathbf{M}}^{(m)} - \rho^{(\ell)}\bar{\mathbf{M}}^{(\ell)}\mathbf{R}\mathbf{S}_{i-1}^{(\ell m)} \\
&\quad + \rho^{(\ell)}\rho^{(m)}\bar{\mathbf{M}}^{(\ell)}\left\{\mathbf{R}\text{Tr}\{\mathbf{R}\mathbf{S}_{i-1}^{(\ell m)}\} + 2\mathbf{R}\mathbf{S}_{i-1}^{(\ell m)}\mathbf{R}\right\}\bar{\mathbf{M}}^{(m)} \\
&\quad + \rho^{(\ell)}\rho^{(m)}\sigma_v^2\bar{\mathbf{M}}^{(\ell)}\mathbf{R}\bar{\mathbf{M}}^{(m)} \\
&\quad + (1-\theta)\left[\rho^{(\ell)}\bar{\mathbf{M}}^{(\ell)}\mathbf{R} - \mathbf{I}\right]\mathbb{E}\{\tilde{\mathbf{w}}_{i-1}^{(\ell)}(\mathbf{w}_{i-1}^o)^T\} \\
&\quad + (1-\theta)\mathbb{E}\{\mathbf{w}_{i-1}^o(\tilde{\mathbf{w}}_{i-1}^{(m)})^T\}\left[\rho^{(m)}\mathbf{R}\bar{\mathbf{M}}^{(m)} - \mathbf{I}\right] \\
&\quad + (1-\theta)^2\mathbb{E}\{\mathbf{w}_{i-1}^o(\mathbf{w}_{i-1}^o)^T\} + (1-\theta^2)\mathbf{Q}.
\end{aligned} \tag{3.17}$$

As required in (3.17), the terms $\mathbb{E}\{\mathbf{w}_{i-1}^o(\mathbf{w}_{i-1}^o)^T\}$ and $\mathbb{E}\{\tilde{\mathbf{w}}_{i-1}^{(n)}(\mathbf{w}_{i-1}^o)^T\}$ can be obtained

as follows:

$$\begin{aligned}
\mathbb{E}\{\mathbf{w}_{i-1}^{\circ}(\mathbf{w}_{i-1}^{\circ})^T\} &= \mathbb{E}\{(\theta\mathbf{w}_{i-2}^{\circ} + \sqrt{1-\theta^2}\mathbf{q}_{i-1})(\theta\mathbf{w}_{i-2}^{\circ} + \sqrt{1-\theta^2}\mathbf{q}_{i-1})^T\} \\
&= \theta^2\mathbb{E}\{\mathbf{w}_{i-2}^{\circ}(\mathbf{w}_{i-2}^{\circ})^T\} + (1-\theta^2)\mathbb{E}\{\mathbf{q}_{i-1}\mathbf{q}_{i-1}^T\} \\
&= \theta^2\mathbb{E}\{\mathbf{w}_{i-2}^{\circ}(\mathbf{w}_{i-2}^{\circ})^T\} + (1-\theta^2)\mathbf{Q}.
\end{aligned} \tag{3.18}$$

Assuming that the adaptive filter is in steady-state, i.e.,

$$\mathbb{E}\{\mathbf{w}_{i-1}^{\circ}(\mathbf{w}_{i-1}^{\circ})^T\} = \mathbb{E}\{\mathbf{w}_{i-2}^{\circ}(\mathbf{w}_{i-2}^{\circ})^T\}, \quad \text{as } i \rightarrow \infty, \tag{3.19}$$

equation (3.18) will converge to:

$$\lim_{i \rightarrow \infty} \mathbb{E}\{\mathbf{w}_{i-1}^{\circ}(\mathbf{w}_{i-1}^{\circ})^T\} = \mathbf{Q}. \tag{3.20}$$

The term $\mathbb{E}\{\tilde{\mathbf{w}}_{i-1}^{(n)}(\mathbf{w}_{i-1}^{\circ})^T\}$ can be obtained by taking the expectation of the product between equation (3.15) and the model (3.3), both at time $i-1$, i.e.,

$$\begin{aligned}
\mathbb{E}\{\tilde{\mathbf{w}}_{i-1}^{(n)}(\mathbf{w}_{i-1}^{\circ})^T\} &= \mathbb{E}\{\tilde{\mathbf{w}}_{i-1}^{(n)}(\theta\mathbf{w}_{i-2}^{\circ} + \sqrt{1-\theta^2}\mathbf{q}_{i-1})^T\} \\
&= \theta\mathbb{E}\{\tilde{\mathbf{w}}_{i-2}^{(n)}(\mathbf{w}_{i-2}^{\circ})^T\} - \rho^{(n)}\theta\mathbb{E}\{\mathbf{M}_{i-1}^{(n)}\mathbf{u}_{i-1}\mathbf{u}_{i-1}^T\tilde{\mathbf{w}}_{i-2}^{(n)}(\mathbf{w}_{i-2}^{\circ})^T\} \\
&\quad - \rho^{(n)}\theta\mathbb{E}\{v(i-1)\mathbf{M}_{i-1}^{(n)}\mathbf{u}_{i-1}(\mathbf{w}_{i-2}^{\circ})^T\} \\
&\quad - \theta(1-\theta)\mathbb{E}\{\mathbf{w}_{i-2}^{\circ}(\mathbf{w}_{i-2}^{\circ})^T\} + \sqrt{1-\theta^2}\left[\theta\mathbb{E}\{\mathbf{q}_{i-1}(\mathbf{w}_{i-2}^{\circ})^T\} + \mathbb{E}\{\tilde{\mathbf{w}}_{i-2}^{(n)}\mathbf{q}_{i-1}^T\}\right] \\
&\quad - \rho^{(n)}\sqrt{1-\theta^2}\mathbb{E}\{\mathbf{M}_{i-1}^{(n)}\mathbf{u}_{i-1}\mathbf{u}_{i-1}^T\tilde{\mathbf{w}}_{i-2}^{(n)}\mathbf{q}_{i-1}^T\} \\
&\quad - \rho^{(n)}\sqrt{1-\theta^2}\mathbb{E}\{v(i-1)\mathbf{M}_{i-1}^{(n)}\mathbf{u}_{i-1}\mathbf{q}_{i-1}^T\} \\
&\quad - (1-\theta)\sqrt{1-\theta^2}\mathbb{E}\{\mathbf{w}_{i-2}^{\circ}\mathbf{q}_{i-1}^T\} + (1-\theta^2)\mathbb{E}\{\mathbf{q}_{i-1}\mathbf{q}_{i-1}^T\}.
\end{aligned}$$

Since \mathbf{q}_{i-1} and $v(i-1)$ are zero-mean and \mathbf{u}_{i-1} is independent of $\tilde{\mathbf{w}}_{i-2}^{(n)}$ and \mathbf{w}_{i-2}° , then

$$\begin{aligned}
\mathbb{E}\{\tilde{\mathbf{w}}_{i-1}^{(n)}(\mathbf{w}_{i-1}^{\circ})^T\} &= \theta\mathbb{E}\{\tilde{\mathbf{w}}_{i-2}^{(n)}(\mathbf{w}_{i-2}^{\circ})^T\} \\
&\quad - \rho^{(n)}\theta\bar{\mathbf{M}}^{(n)}\mathbf{R}\mathbb{E}\{\tilde{\mathbf{w}}_{i-2}^{(n)}(\mathbf{w}_{i-2}^{\circ})^T\} - \theta(1-\theta)\mathbb{E}\{\mathbf{w}_{i-2}^{\circ}(\mathbf{w}_{i-2}^{\circ})^T\} + (1-\theta^2)\mathbf{Q}.
\end{aligned} \tag{3.21}$$

Assuming that the filter is operating in steady-state, i.e.,

$$\mathbb{E}\{\tilde{\mathbf{w}}_{i-1}^{(n)}(\mathbf{w}_{i-1}^{\circ})^T\} = \mathbb{E}\{\tilde{\mathbf{w}}_{i-2}^{(n)}(\mathbf{w}_{i-2}^{\circ})^T\}, \quad \text{as } i \rightarrow \infty \tag{3.22}$$

considering that the eigenvalues of $\theta(\mathbf{I} - \rho^{(n)}\bar{\mathbf{M}}^{(n)}\mathbf{R})$ lie between $(-1,1)$ and using the result

of (3.20), equation (3.21) will converge to

$$\lim_{i \rightarrow \infty} \mathbb{E}\{\tilde{\mathbf{w}}_{i-1}^{(n)}(\mathbf{w}_{i-1}^o)^T\} = (1 - \theta)\Psi^{-1}\mathbf{Q}, \quad (3.23)$$

where $\Psi = (1 - \theta)\mathbf{I} + \rho^{(n)}\theta\bar{\mathbf{M}}^{(n)}\mathbf{R}$.

3.3 Theoretical steady-state analysis

Based on the previous results obtained from model (3.3), the next subsections 3.3.1 to 3.3.4 present the theoretical steady-state EMSE for four different situations, namely: one individual LMS filter, one individual RLS filter, their convex combination and finally for the Kalman filter.

3.3.1 Theoretical steady-state EMSE for LMS

To compute the EMSE for an individual LMS filter, substitute $\ell = m = 1$ in (3.17), considering $\bar{\mathbf{M}}^{(1)} = \mathbf{I}$ and $\rho^{(1)} = \mu$ and assuming that the filter is operating in steady-state, i.e. $i \rightarrow \infty$, we get the following recursion for $\mathbf{S}_\infty^{(11)}$ after replacing equations (3.20) and (3.23) in (3.17):

$$\begin{aligned} \mathbf{S}_\infty^{(11)} &\approx \mathbf{S}_\infty^{(11)} - \mu[\mathbf{S}_\infty^{(11)}\mathbf{R} + \mathbf{R}\mathbf{S}_\infty^{(11)}] + \mu^2 \{ \mathbf{R}\text{Tr}\{\mathbf{R}\mathbf{S}_\infty^{(11)}\} + 2\mathbf{R}\mathbf{S}_\infty^{(11)}\mathbf{R} \} + \mu^2\sigma_v^2\mathbf{R} \\ &\quad + (1 - \theta)^2[\mu\mathbf{R} - \mathbf{I}][(1 - \theta)\mathbf{I} + \mu\theta\mathbf{R}]^{-1}\mathbf{Q} + (1 - \theta)^2\mathbf{Q}[(1 - \theta)\mathbf{I} + \mu\theta\mathbf{R}]^{-1}[\mu\mathbf{R} - \mathbf{I}] \\ &\quad + 2(1 - \theta)\mathbf{Q}. \end{aligned} \quad (3.24)$$

To simplify (3.24) and get an easier expression to deal with, we multiply and divide by θ the second line of (3.24) and reorganize the terms to obtain

$$\begin{aligned} \mathbf{S}_\infty^{(11)} &\approx \mathbf{S}_\infty^{(11)} - \mu[\mathbf{S}_\infty^{(11)}\mathbf{R} + \mathbf{R}\mathbf{S}_\infty^{(11)}] + \mu^2 \{ \mathbf{R}\text{Tr}\{\mathbf{R}\mathbf{S}_\infty^{(11)}\} + 2\mathbf{R}\mathbf{S}_\infty^{(11)}\mathbf{R} \} + \mu^2\sigma_v^2\mathbf{R} \\ &\quad + \theta^{-1}(1 - \theta)^2[\mu\theta\mathbf{R} - \theta\mathbf{I}][\mathbf{I} + \mu\theta\mathbf{R} - \theta\mathbf{I}]^{-1}\mathbf{Q} \\ &\quad + \theta^{-1}(1 - \theta)^2\mathbf{Q}[\mathbf{I} + \mu\theta\mathbf{R} - \theta\mathbf{I}]^{-1}[\mu\theta\mathbf{R} - \theta\mathbf{I}] \\ &\quad + 2(1 - \theta)\mathbf{Q}. \end{aligned} \quad (3.25)$$

By using the push-through identity property $\mathbf{A}(\mathbf{I} + \mathbf{A})^{-1} = (\mathbf{I} + \mathbf{A})^{-1}\mathbf{A} = \mathbf{I} - (\mathbf{I} + \mathbf{A})^{-1}$, where $\mathbf{A} = \mu\theta\mathbf{R} - \theta\mathbf{I}$, after some algebra we obtain the following simplified version of

equation (3.24)

$$\begin{aligned}
\mathbf{S}_\infty^{(11)} &\approx \mathbf{S}_\infty^{(11)} - \mu[\mathbf{S}_\infty^{(11)}\mathbf{R} + \mathbf{R}\mathbf{S}_\infty^{(11)}] + \mu^2 \{ \mathbf{R}\text{Tr}\{\mathbf{R}\mathbf{S}_\infty^{(11)}\} + 2\mathbf{R}\mathbf{S}_\infty^{(11)}\mathbf{R} \} \\
&\quad + \mu^2\sigma_v^2\mathbf{R} - \theta^{-1}(1-\theta)^2[(1-\theta)\mathbf{I} + \mu\theta\mathbf{R}]^{-1}\mathbf{Q} - \theta^{-1}(1-\theta)^2\mathbf{Q}[(1-\theta)\mathbf{I} + \mu\theta\mathbf{R}]^{-1} \\
&\quad + 2\theta^{-1}(1-\theta)\mathbf{Q}.
\end{aligned} \tag{3.26}$$

Based on (3.26), instead of computing the steady-state EMSE for LMS in a direct way, we follow the same steps of [13] and define the rotated matrix given by $\bar{\mathbf{S}}_\infty^{(11)} = \mathbf{U}^T\mathbf{S}_\infty^{(11)}\mathbf{U}$, where \mathbf{U} is an orthogonal matrix that diagonalizes \mathbf{R} , that is

$$\mathbf{U}^T\mathbf{R}\mathbf{U} = \text{diag}(\lambda_i) \triangleq \mathbf{\Lambda} \tag{3.27}$$

where $\text{diag}(\lambda_i)$ is a diagonal matrix formed with the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_M$ of \mathbf{R} .

A recursion for $\bar{\mathbf{S}}_\infty^{(11)}$ can be obtained by multiplying (3.26) from the left by \mathbf{U}^T and from the right by \mathbf{U} . Defining the rotated matrix $\bar{\mathbf{Q}} = \mathbf{U}^T\mathbf{Q}\mathbf{U}$ and recalling that $\mathbf{R} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ and $\mathbf{I} = \mathbf{U}\mathbf{U}^T = \mathbf{U}^T\mathbf{U}$, we get after simplifications

$$\begin{aligned}
\bar{\mathbf{S}}_\infty^{(11)} &\approx \bar{\mathbf{S}}_\infty^{(11)} - \mu\{ \bar{\mathbf{S}}_\infty^{(11)}\mathbf{\Lambda} + \mathbf{\Lambda}\bar{\mathbf{S}}_\infty^{(11)} \} + \mu^2\{ \mathbf{\Lambda}\text{Tr}\{\mathbf{\Lambda}\bar{\mathbf{S}}_\infty^{(11)}\} + 2\mathbf{\Lambda}\bar{\mathbf{S}}_\infty^{(11)}\mathbf{\Lambda} \} \\
&\quad + \mu^2\sigma_v^2\mathbf{\Lambda} - \theta^{-1}(1-\theta)^2[(1-\theta)\mathbf{I} + \mu\theta\mathbf{\Lambda}]^{-1}\bar{\mathbf{Q}} - \theta^{-1}(1-\theta)^2\bar{\mathbf{Q}}[(1-\theta)\mathbf{I} + \mu\theta\mathbf{\Lambda}]^{-1} \\
&\quad + 2\theta^{-1}(1-\theta)\bar{\mathbf{Q}}.
\end{aligned} \tag{3.28}$$

Using the rotated matrix $\bar{\mathbf{S}}_\infty^{(11)}$, the steady-state EMSE can be computed as $\text{Tr}\{\mathbf{\Lambda}\bar{\mathbf{S}}_\infty^{(11)}\}$ and so, it depends only on the diagonal entries of $\bar{\mathbf{S}}_\infty^{(11)}$. We can work therefore only with these diagonal entries and define the vectors

$$\bar{\mathbf{s}}_\infty^{(11)} = \text{diag}\{\bar{\mathbf{S}}_\infty^{(11)}\} \text{ and } \boldsymbol{\ell} = \text{diag}\{\mathbf{\Lambda}\}, \tag{3.29}$$

where $\text{diag}\{\mathbf{A}\}$ represents a column vector with the diagonal elements of \mathbf{A} .

Given this, by applying the diagonal operator to both sides of equation (3.28) and simplifying we obtain

$$\bar{\mathbf{s}}_\infty^{(11)} = [2\mu\mathbf{\Lambda} - \mu^2\boldsymbol{\ell}\boldsymbol{\ell}^T - 2\mu^2\mathbf{\Lambda}^2]^{-1} \left\{ \mu^2\sigma_v^2\boldsymbol{\ell} + \frac{2(1-\theta)}{\theta} \left\{ \mathbf{I} - \left[\mathbf{I} + \frac{\mu\theta}{1-\theta}\mathbf{\Lambda} \right]^{-1} \right\} \text{diag}\{\bar{\mathbf{Q}}\} \right\}. \tag{3.30}$$

The steady-state EMSE and MSD can now be computed as

$$\zeta^{\text{LMS}} = \boldsymbol{\ell}^T \bar{\mathbf{s}}_{\infty}^{(11)}, \quad \varepsilon^{\text{LMS}} = \mathbb{1}^T \bar{\mathbf{s}}_{\infty}^{(11)}, \quad (3.31)$$

where $\mathbb{1} = [1 \ 1 \ \dots \ 1]^T$.

Due to the complexity of (3.30), we present next an approximation valid for sufficiently small μ in order to better describe the qualitative behavior of the LMS filter. We use the analysis below to show that there is a minimum value for θ , $\theta_{\min}^{\text{LMS}}$, below which the filter is no longer able to track variations in \mathbf{w}_i° . Assuming that the term $\mu^2 \left\{ \mathbf{R} \text{Tr}\{\mathbf{R}\mathbf{S}_{\infty}^{(11)}\} + 2\mathbf{R}\mathbf{S}_{\infty}^{(11)}\mathbf{R} \right\}$ can be neglected with respect to the three first terms on the right-hand side of (3.26) and applying the trace operator to both sides of this equation, ζ^{LMS} reduces to

$$\begin{aligned} \zeta^{\text{LMS}} = \text{Tr}\{\mathbf{R}\mathbf{S}_{\infty}^{(11)}\} \approx & \frac{1}{2\mu} \left\{ \mu^2 \sigma_v^2 \text{Tr}\{\mathbf{R}\} - 2\theta^{-1}(1-\theta)^2 \text{Tr}\{[(1-\theta)\mathbf{I} + \mu\theta\mathbf{R}]^{-1}\mathbf{Q}\} \right. \\ & \left. + 2\theta^{-1}(1-\theta) \text{Tr}\{\mathbf{Q}\} \right\}, \end{aligned} \quad (3.32)$$

where we used the property $\text{Tr}\{\mathbf{A}\mathbf{B}\} = \text{Tr}\{\mathbf{B}\mathbf{A}\}$.

Applying the matrix inversion lemma [2] to the term $[(1-\theta)\mathbf{I} + \mu\theta\mathbf{R}]^{-1}$ in (3.32), after some algebra, we obtain the following approximate expression

$$\zeta^{\text{LMS}} \approx \frac{\mu\sigma_v^2 \text{Tr}\{\mathbf{R}\}}{2} + (1-\theta) \text{Tr}\{[(1-\theta)\mathbf{I} + \mu\theta\mathbf{R}]^{-1}\mathbf{R}\mathbf{Q}\}. \quad (3.33)$$

The μ° that minimizes (3.33) can be obtained by setting the first derivative of ζ^{LMS} equal to zero. When $\theta = 1$, ζ^{LMS} reduces to a linear function of μ given by

$$\zeta^{\text{LMS}} \approx \left. \frac{\mu\sigma_v^2 \text{Tr}\{\mathbf{R}\}}{2} \right|_{\theta=1} \quad (3.34)$$

and so, the optimum step-size that minimizes (3.34) is $\mu^{\circ} = 0$ with the corresponding minimum EMSE $\zeta_{\text{LMS}}^{\circ} = 0$. The reader may find this result puzzling, since the tracking model usually seen in the literature includes a term proportional to μ^{-1} , resulting in a positive optimum step size [2, 13]. Our result stems from the factor $\sqrt{1-\theta^2}$ in our model (3.3), which we included to keep the power of \mathbf{w}_i° independent of θ . Note that for $\theta = 1$, the optimum step size is $\mu^{\circ} = 0$ because model (3.3) implies that $\mathbb{E}\{\mathbf{w}_i^{\circ}\} = \mathbf{0}$, and the initial condition in our model also satisfies $\mathbb{E}\{\mathbf{w}_{-1}\} = \mathbf{0}$.

In order to compute the first derivative of (3.33) for $0 \leq \theta < 1$, we rewrite the terms of (3.33) as a sum of scalars. To do this, we use the same decomposition for matrix \mathbf{R}

described in (3.27) and the same matrix $\bar{\mathbf{Q}}$ used in (3.28) to obtain

$$\begin{aligned}\zeta^{\text{LMS}} &\approx \frac{\mu\sigma_v^2\text{Tr}\{\mathbf{\Lambda}\}}{2} + (1-\theta)\text{Tr}\{[(1-\theta)\mathbf{I} + \mu\theta\mathbf{\Lambda}]^{-1}\mathbf{\Lambda}\bar{\mathbf{Q}}\} \\ &= \sum_{i=1}^M \frac{\mu\sigma_v^2\lambda_{ii}}{2} + \sum_{i=1}^M \frac{(1-\theta)\lambda_{ii}\bar{q}_{ii}}{1-\theta + \mu\theta\lambda_{ii}},\end{aligned}\quad (3.35)$$

where \bar{q}_{ii} are the diagonal elements of $\bar{\mathbf{Q}}$.

By taking the first derivative of (3.35) and setting it equal to 0, after simplifications, we find the following expression to compute the optimum step-size $\mu = \mu^\circ$ that minimizes the steady-state EMSE for LMS

$$\sum_{i=1}^M \left\{ \frac{\theta(1-\theta)\lambda_{ii}^2\bar{q}_{ii}}{(1-\theta + \mu\theta\lambda_{ii})^2} - \frac{\sigma_v^2\lambda_{ii}}{2} \right\} = 0. \quad (3.36)$$

Since in general a closed solution for μ in (3.36) is not possible (if the λ_{ii} are distinct, (3.36) would reduce to finding the roots of a $2M$ -degree polynomial in μ), we seek solutions for $\theta \in [0, 1[$ numerically. Given this, let us start by naming $f(\mu)$ as the left-hand side of (3.36), i.e.,

$$f(\mu) = \sum_{i=1}^M \left\{ \frac{\theta(1-\theta)\lambda_{ii}^2\bar{q}_{ii}}{(1-\theta + \mu\theta\lambda_{ii})^2} - \frac{\sigma_v^2\lambda_{ii}}{2} \right\}.$$

Since we are seeking conditions under which $f(\mu) = 0$ has a positive solution, we may start our analysis by finding the corresponding boundaries of $f(\mu)$ when $\mu \rightarrow 0$ and $\mu \rightarrow \infty$, which are:

$$\lim_{\mu \rightarrow 0} f(\mu) = \frac{\theta\text{Tr}\{\mathbf{\Lambda}^2\bar{\mathbf{Q}}\}}{1-\theta} - \frac{\sigma_v^2\text{Tr}\{\mathbf{\Lambda}\}}{2} \quad \text{and} \quad \lim_{\mu \rightarrow \infty} f(\mu) = -\sum_{i=1}^M \frac{\sigma_v^2\lambda_{ii}}{2} = -\frac{\sigma_v^2\text{Tr}\{\mathbf{\Lambda}\}}{2}.$$

Since the lower bound $\lim_{\mu \rightarrow \infty} f(\mu)$ is always negative (\mathbf{R} was assumed to be a positive-definite matrix), the upper bound $\lim_{\mu \rightarrow 0} f(\mu)$ must be greater than or equal to 0 to ensure a solution for $f(\mu) = 0$. In other words,

$$\frac{\theta\text{Tr}\{\mathbf{\Lambda}^2\bar{\mathbf{Q}}\}}{1-\theta} - \frac{\sigma_v^2\text{Tr}\{\mathbf{\Lambda}\}}{2} \geq 0.$$

By rewriting $\text{Tr}\{\mathbf{\Lambda}^2\bar{\mathbf{Q}}\}$ as $\text{Tr}\{\mathbf{R}^2\mathbf{Q}\}$ and $\text{Tr}\{\mathbf{\Lambda}\}$ as $\text{Tr}\{\mathbf{R}\}$, and reorganizing the terms, we get

$$\frac{\theta}{1-\theta} \geq \frac{\sigma_v^2\text{Tr}\{\mathbf{R}\}}{2\text{Tr}\{\mathbf{R}^2\mathbf{Q}\}}. \quad (3.37)$$

Multiplying both sides of (3.37) by $1 - \theta$ and solving the inequality for θ , we obtain an expression for the minimum θ of LMS — $\theta_{\min}^{\text{LMS}}$ — that guarantees an intersection point $\mu = \mu^\circ \geq 0$ between $f(\mu)$ and the μ -axis. This expression is given by

$$\theta_{\min}^{\text{LMS}} = \frac{\gamma}{\gamma + 2}, \quad (3.38)$$

where $\gamma = \sigma_v^2 \text{Tr}\{\mathbf{R}\} / \text{Tr}\{\mathbf{R}^2 \mathbf{Q}\}$. For $\theta \leq \theta_{\min}^{\text{LMS}}$, since $\lim_{\mu \rightarrow 0} f(\mu) \leq 0$, it can be shown that the smallest non negative step-size is $\mu^\circ = 0$, otherwise $\mu^\circ > 0$.

This result can be interpreted as follows: since the steady-state power of \mathbf{w}_i° is independent of θ , and (3.3) is a low-pass filter for $0 < \theta < 1$ with a bandwidth that increases as θ decreases, we conclude that for $\theta \leq \theta_{\min}^{\text{LMS}}$, the filter no longer can track the variations of the weight vector. The optimum solution $\mu^\circ = 0$ in this case results from the assumption that $\mathbb{E}\{\mathbf{w}_i^\circ\} = \mathbb{E}\{\mathbf{w}_{-1}\} = \mathbf{0}$. Section 3.4.1 discusses the steady-state EMSE of the filter when $\mathbb{E}\{\mathbf{w}_i^\circ\} \neq \mathbf{0}$ and $\theta \leq \theta_{\min}^{\text{LMS}}$.

The nonstationarity degree (NSD) of [44] also provides a bound for the tracking capabilities of an adaptive filter. Under our notation, a filter is unable to track a time-varying vector if $\text{NSD} = (1 - \theta^2) \text{Tr}\{\mathbf{R}\mathbf{Q}\} / \sigma_v^2 > 1$. We provide a comparison between our result and the NSD in Section 3.4.

3.3.2 Theoretical steady-state EMSE for RLS

Similar to the LMS filter, to compute the steady-state EMSE and MSD for an individual RLS filter, we use $\ell = m = 2$, $\bar{\mathbf{M}}^{(2)} = \bar{\mathbf{P}} = (1 - \lambda)\mathbf{R}^{-1}$, $\rho^{(2)} = 1$ and the same orthogonal transformation \mathbf{U} in (3.17), obtaining

$$\bar{\mathbf{s}}_\infty^{(22)} \approx [2\lambda\mathbf{I} - (1 - \lambda)\boldsymbol{\ell}^{-1}\boldsymbol{\ell}^T]^{-1} \left\{ \sigma_v^2(1 - \lambda)\boldsymbol{\ell}^{-1} + \frac{2(1 - \theta)}{1 - \theta\lambda} \text{diag}\{\bar{\mathbf{Q}}\} \right\}, \quad (3.39)$$

where $\boldsymbol{\ell}^{-1}$ denotes element-wise inversion. The RLS steady-state EMSE and MSD are then

$$\zeta^{\text{RLS}} = \boldsymbol{\ell}^T \bar{\mathbf{s}}_\infty^{(22)}, \quad \varepsilon^{\text{RLS}} = \mathbf{1}^T \bar{\mathbf{s}}_\infty^{(22)}. \quad (3.40)$$

The algebraic development used to obtain (3.39) is described in section A.1 of Appendix A.

Similarly to the qualitative analysis of the LMS filter developed in Sec. 3.3.1, for λ close to 1, the term $(1 - \lambda)\boldsymbol{\ell}^{-1}\boldsymbol{\ell}^T$ can be disregarded with respect to $2\lambda\mathbf{I}$ in (3.39) and so

the EMSE simplifies to

$$\zeta^{\text{RLS}} \approx \frac{\sigma_v^2(1-\lambda)M}{2} + \frac{(1-\theta)\text{Tr}\{\mathbf{RQ}\}}{1-\theta\lambda}. \quad (3.41)$$

The λ° that minimizes (3.41) can be obtained by setting the first derivative of ζ^{RLS} equal to zero. When $\theta = 1$, ζ^{RLS} reduces to a linear function of $(1-\lambda)$ given by

$$\zeta^{\text{RLS}} \approx \left. \frac{\sigma_v^2(1-\lambda)M}{2} \right|_{\theta=1} \quad (3.42)$$

and so, the optimum forgetting factor that minimizes (3.42) is $\lambda^\circ = 1$ with the corresponding minimum EMSE $\zeta_{\text{RLS}}^\circ = 0$ (see the footnote in the previous section).

When $0 \leq \theta < 1$, the first derivative of (3.41) is equal to

$$\frac{\partial \zeta^{\text{RLS}}}{\partial \lambda} \approx \frac{-\sigma_v^2 M}{2} + \frac{\theta(1-\theta)\text{Tr}\{\mathbf{RQ}\}}{(1-\theta\lambda)^2}. \quad (3.43)$$

Setting (3.43) equal to 0 and solving for λ leads to the following expression for λ° :

$$\lambda^\circ \approx \min \left\{ \frac{1-\sqrt{\Gamma}}{\theta}, 1 \right\}, \quad (3.44)$$

where $\Gamma = 2\theta(1-\theta)\text{Tr}\{\mathbf{RQ}\}/(\sigma_v^2 M)$.

After simplification, the minimum θ for the RLS filter — $\theta_{\min}^{\text{RLS}}$ — that guarantees $\lambda^\circ < 1$ in (3.44) is given by

$$\theta_{\min}^{\text{RLS}} = \frac{1}{2\alpha^2 + 1}, \quad (3.45)$$

where $\alpha = \sqrt{\text{Tr}\{\mathbf{RQ}\}/(\sigma_v^2 M)}$. The comments at the end of the previous section also apply here.

Remark: More precise values for θ_{\min} than (3.38) and (3.45) can be obtained numerically directly by minimizing the full expressions for ζ^{LMS} (3.31) and ζ^{RLS} (3.40). We discuss this further in Section 3.4.

3.3.3 Theoretical steady-state EMSE for combination between LMS and RLS

To compute the steady-state EMSE of the convex combination between LMS and RLS filters, we assume the LMS as $\ell = 1$ and the RLS as $m = 2$, with the corresponding parameters $\rho^{(1)} = \mu$, $\bar{\mathbf{M}}^{(1)} = \mathbf{I}$, $\rho^{(2)} = 1$ and $\bar{\mathbf{M}}^{(2)} = \bar{\mathbf{P}} = (1-\lambda)\mathbf{R}^{-1}$. Following the same steps as for the LMS and RLS filters, we obtain the diagonal $\bar{\mathbf{s}}_\infty^{(12)}$ of the steady-state

transformed cross-covariance matrix $\mathbf{U}^T \mathbf{S}_\infty^{(12)} \mathbf{U}$ as

$$\begin{aligned} \bar{\mathbf{s}}_\infty^{(12)} &= \left\{ (1-\lambda)\mathbf{I} + \mu\mathbf{\Lambda} - \mu(1-\lambda)(\mathbb{1}\boldsymbol{\ell}^T + 2\mathbf{\Lambda}) \right\}^{-1} \\ &\times \left\{ \mu(1-\lambda)\sigma_v^2\mathbb{1} - \frac{(1-\theta)^2}{\theta}[(1-\theta)\mathbf{I} + \mu\theta\mathbf{\Lambda}]^{-1}\text{diag}\{\bar{\mathbf{Q}}\} \right. \\ &\left. + \frac{(1-\theta)(1-2\theta\lambda + \theta)}{\theta(1-\theta\lambda)}\text{diag}\{\bar{\mathbf{Q}}\} \right\}, \end{aligned} \quad (3.46)$$

and again

$$\zeta^{(12)} = \boldsymbol{\ell}^T \bar{\mathbf{s}}_\infty^{(12)}, \quad \varepsilon^{(12)} = \mathbb{1}^T \bar{\mathbf{s}}_\infty^{(12)}. \quad (3.47)$$

The algebraic development used to obtain (3.46) is described in section A.2 of Appendix A.

For sufficiently small μ and λ close to 1, the term $\mu(1-\lambda)(\mathbb{1}\boldsymbol{\ell}^T + 2\mathbf{\Lambda})$ can be neglected with respect to the three first terms on the right-hand side of (3.46), and the following approximation is obtained for the cross-EMSE:

$$\begin{aligned} \zeta^{(12)} = \text{Tr}\{\mathbf{R}\mathbf{S}_\infty^{(12)}\} &\approx \mu\sigma_v^2(1-\lambda)\text{Tr}\{\mathbf{\Gamma}\} - \frac{(1-\theta)^2\text{Tr}\{\mathbf{\Gamma}[(1-\theta)\mathbf{I} + \mu\theta\mathbf{R}]^{-1}\mathbf{Q}\}}{\theta} \\ &+ \frac{(1-\theta)(1-2\theta\lambda + \theta)\text{Tr}\{\mathbf{\Gamma}\mathbf{Q}\}}{\theta(1-\theta\lambda)}, \end{aligned} \quad (3.48)$$

where $\mathbf{\Gamma} = \mathbf{R}[\mu\mathbf{R} + (1-\lambda)\mathbf{I}]^{-1}$.

3.3.4 Theoretical steady-state EMSE for Kalman filter

The straightforward approach to develop a theoretical model for the steady-state EMSE of the Kalman filter would be to use the nonstationary data model described in (3.2). However, this results in a recursion for the covariance matrix that is costly to solve. Instead, we define a modified data model (see (3.49) below) that can be put in a DARE (Discrete-Time Algebraic Riccati Equation) format and therefore can be solved efficiently using iterative procedures (see [48, 49]).

Given this, assume that instead of the linear model for $d(i)$ presented in (3.2), we have access to the following nonstationary data model

$$\mathbf{u}_i d(i) = \mathbf{u}_i \mathbf{u}_i^T \mathbf{w}_{i-1}^o + \mathbf{u}_i v(i), \quad (3.49)$$

where the optimum solution \mathbf{w}_i^o is still given by (3.3).

Comparing the state-space model (1.38) with (3.3)–(3.49), the Kalman model corre-

sponding to the nonstationary data model described in (3.3) and (3.49) is presented in Table 3.

Table 3: Relation between the Kalman filter and the adaptive filter variables.

KF	-	AF	KF	-	AF
\mathbf{x}_i	\leftrightarrow	\mathbf{w}_i^o	$\hat{\mathbf{x}}_i$	\leftrightarrow	\mathbf{w}_i
\mathbf{z}_i	\leftrightarrow	$\mathbf{u}_i d(i)$	\mathbf{H}_i	\leftrightarrow	$\mathbf{u}_i \mathbf{u}_i^T$
\mathbf{F}_i	\leftrightarrow	$\theta \mathbf{I}$	\mathbf{G}_i	\leftrightarrow	$\sqrt{1 - \theta^2} \mathbf{I}$
\mathbf{v}_i	\leftrightarrow	$\mathbf{u}_i v(i)$	\mathbf{t}_i	\leftrightarrow	\mathbf{q}_i
\mathbf{V}_i	\leftrightarrow	$\sigma_v^2 \mathbf{R}$	\mathbf{T}_i	\leftrightarrow	\mathbf{Q}

Since the Kalman model assumes \mathbf{H}_i to be deterministic and \mathbf{u}_i is random, using the equivalences from Table 3 in (1.41a), (1.41b) and (1.41e), and defining $\tilde{\mathbf{w}}_i = \mathbf{w}_i^o - \mathbf{w}_i$, we can write

$$\mathbf{\Omega}_i = \sigma_v^2 \mathbf{R} + \mathbf{u}_i \mathbf{u}_i^T \mathcal{P}_{i-1} \mathbf{u}_i \mathbf{u}_i^T, \quad (3.50a)$$

$$\mathbf{K}_i = \theta \mathcal{P}_{i-1} \mathbf{u}_i \mathbf{u}_i^T (\sigma_v^2 \mathbf{R} + \mathbf{u}_i \mathbf{u}_i^T \mathcal{P}_{i-1} \mathbf{u}_i \mathbf{u}_i^T)^{-1}, \quad (3.50b)$$

$$\begin{aligned} \mathbb{E}\{\tilde{\mathbf{w}}_i \tilde{\mathbf{w}}_i^T | \mathbf{u}_0, \dots, \mathbf{u}_i\} &= \mathcal{P}_i = \theta^2 \mathcal{P}_{i-1} + (1 - \theta^2) \mathbf{Q} - \mathbf{K}_i \mathbf{\Omega}_i \mathbf{K}_i^T \\ &= \theta^2 \mathcal{P}_{i-1} + (1 - \theta^2) \mathbf{Q} \\ &\quad - \theta^2 \mathcal{P}_{i-1} \mathbf{u}_i \mathbf{u}_i^T [\sigma_v^2 \mathbf{R} + \mathbf{u}_i \mathbf{u}_i^T \mathcal{P}_{i-1} \mathbf{u}_i \mathbf{u}_i^T]^{-1} \mathbf{u}_i \mathbf{u}_i^T \mathcal{P}_{i-1}, \end{aligned} \quad (3.50c)$$

where the cross-covariance matrix $\mathcal{S}_i = \mathbf{0}$ since $v(i)$ and \mathbf{q}_i are independent and zero mean. Now take the expectation with respect to $\mathbf{u}_0, \dots, \mathbf{u}_i$ in (3.50c), assuming that $\mathbb{E}\{\mathbf{u}_i \mathbf{u}_i^T [\sigma_v^2 \mathbf{R} + \mathbf{u}_i \mathbf{u}_i^T \mathcal{P}_{i-1} \mathbf{u}_i \mathbf{u}_i^T]^{-1} \mathbf{u}_i \mathbf{u}_i^T\} \approx \mathbf{R} [\sigma_v^2 \mathbf{R} + \mathbf{R} \bar{\mathcal{P}}_{i-1} \mathbf{R}]^{-1} \mathbf{R}$ (an approximation that tends to be better for long filters [13]), obtaining the Riccati recursion in DARE format

$$\bar{\mathcal{P}}_i = \mathbb{E}\{\mathcal{P}_i\} \approx \theta^2 \bar{\mathcal{P}}_{i-1} + (1 - \theta^2) \mathbf{Q} - \theta^2 \bar{\mathcal{P}}_{i-1} \mathbf{R} [\sigma_v^2 \mathbf{R} + \mathbf{R} \bar{\mathcal{P}}_{i-1} \mathbf{R}]^{-1} \mathbf{R} \bar{\mathcal{P}}_{i-1}. \quad (3.51)$$

The steady-state EMSE and MSD for the Kalman filter are obtained using

$$\zeta^{\text{KAL}} = \text{Tr}\{\mathbf{R} \bar{\mathcal{P}}_\infty\}, \quad \varepsilon^{\text{KAL}} = \text{Tr}\{\bar{\mathcal{P}}_\infty\}, \quad (3.52)$$

where $\bar{\mathcal{P}}_\infty = \lim_{i \rightarrow \infty} \bar{\mathcal{P}}_{i-1}$ is the solution of the discrete-time algebraic Riccati equation obtained by substituting $\bar{\mathcal{P}}_i$ and $\bar{\mathcal{P}}_{i-1}$ by $\bar{\mathcal{P}}_\infty$ in (3.51).

The analytical expressions of the steady-state ζ -EMSE for LMS, RLS, their convex combination and for the Kalman filter are summarized in Table 4 for the general case $\mu > 0$ and $0 \ll \lambda < 1$ and in Table 5 for sufficiently small μ and λ close to 1. The results for the MSD are summarized in Table 6.

Table 4: Analytical expressions for the steady-state ζ -EMSE considering LMS, RLS, their convex combination and for the Kalman filter.

ζ -EMSE
$\zeta^{\text{LMS}} = \boldsymbol{\ell}^T [2\mu\boldsymbol{\Lambda} - \mu^2\boldsymbol{\ell}\boldsymbol{\ell}^T - 2\mu^2\boldsymbol{\Lambda}^2]^{-1} \left\{ \mu^2\sigma_v^2\boldsymbol{\ell} + \frac{2(1-\theta)}{\theta} \left\{ \mathbf{I} - \left[\mathbf{I} + \frac{\mu\theta}{1-\theta}\boldsymbol{\Lambda} \right]^{-1} \right\} \text{diag}\{\bar{\mathbf{Q}}\} \right\},$ <p style="text-align: center;">with $\boldsymbol{\Lambda} = \mathbf{U}^T\mathbf{R}\mathbf{U}$, $\bar{\mathbf{Q}} = \mathbf{U}^T\mathbf{Q}\mathbf{U}$ and $\boldsymbol{\ell} = \text{diag}\{\boldsymbol{\Lambda}\}$.</p>
$\zeta^{\text{RLS}} = \boldsymbol{\ell}^T [2\lambda\mathbf{I} - (1-\lambda)\boldsymbol{\ell}^{-1}\boldsymbol{\ell}^T]^{-1} \left\{ \sigma_v^2(1-\lambda)\boldsymbol{\ell}^{-1} + \frac{2(1-\theta)}{1-\theta\lambda} \text{diag}\{\bar{\mathbf{Q}}\} \right\},$ <p style="text-align: center;">with $\bar{\mathbf{Q}} = \mathbf{U}^T\mathbf{Q}\mathbf{U}$, $\boldsymbol{\ell} = \text{diag}\{\boldsymbol{\Lambda}\}$ and $\boldsymbol{\ell}^{-1} = \text{diag}\{\boldsymbol{\Lambda}^{-1}\}$.</p>
$\zeta^{\text{COMB}} = \frac{\zeta^{\text{LMS}}\zeta^{\text{RLS}} - (\zeta^{(12)})^2}{\zeta^{\text{LMS}} - 2\zeta^{(12)} + \zeta^{\text{RLS}}}, \text{ where}$ $\zeta^{(12)} = \boldsymbol{\ell}^T \left\{ (1-\lambda)\mathbf{I} + \mu\boldsymbol{\Lambda} - \mu(1-\lambda)(\mathbb{1}\boldsymbol{\ell}^T + 2\boldsymbol{\Lambda}) \right\}^{-1} \left\{ \begin{aligned} &+ \mu(1-\lambda)\sigma_v^2\mathbb{1} - \frac{(1-\theta)^2}{\theta} [(1-\theta)\mathbf{I} + \mu\theta\boldsymbol{\Lambda}]^{-1} \text{diag}\{\bar{\mathbf{Q}}\} \\ &+ \frac{(1-\theta)(1-2\theta\lambda + \theta)}{\theta(1-\theta\lambda)} \text{diag}\{\bar{\mathbf{Q}}\} \end{aligned} \right\},$ <p style="text-align: center;">with $\boldsymbol{\Lambda} = \mathbf{U}^T\mathbf{R}\mathbf{U}$, $\bar{\mathbf{Q}} = \mathbf{U}^T\mathbf{Q}\mathbf{U}$ and $\boldsymbol{\ell} = \text{diag}\{\boldsymbol{\Lambda}\}$.</p>
$\zeta^{\text{KAL}} = \text{Tr}\{\mathbf{R}\bar{\mathcal{P}}_\infty\}, \text{ where } \bar{\mathcal{P}}_\infty \text{ is the solution of}$ $\bar{\mathcal{P}}_\infty \approx \theta^2\bar{\mathcal{P}}_\infty + (1-\theta^2)\mathbf{Q} - \theta^2\bar{\mathcal{P}}_\infty\mathbf{R} [\sigma_v^2\mathbf{R} + \mathbf{R}\bar{\mathcal{P}}_\infty\mathbf{R}]^{-1} \mathbf{R}\bar{\mathcal{P}}_\infty.$

Table 5: Analytical expressions for the steady-state ζ -EMSE considering LMS, RLS and their convex combination for sufficiently small μ and λ close to 1.

ζ -EMSE
$\zeta^{\text{LMS}} \approx \frac{\mu\sigma_v^2\text{Tr}\{\mathbf{R}\}}{2} + (1-\theta)\text{Tr}\{[(1-\theta)\mathbf{I} + \mu\theta\mathbf{R}]^{-1}\mathbf{R}\mathbf{Q}\}$
$\zeta^{\text{RLS}} \approx \frac{\sigma_v^2(1-\lambda)M}{2} + \frac{(1-\theta)\text{Tr}\{\mathbf{R}\mathbf{Q}\}}{1-\theta\lambda}$
$\zeta^{\text{COMB}} = \frac{\zeta^{\text{LMS}}\zeta^{\text{RLS}} - (\zeta^{(12)})^2}{\zeta^{\text{LMS}} - 2\zeta^{(12)} + \zeta^{\text{RLS}}}, \text{ where}$ $\zeta^{(12)} \approx \mu\sigma_v^2(1-\lambda)\text{Tr}\{\boldsymbol{\Gamma}\} - \frac{(1-\theta)^2\text{Tr}\{\boldsymbol{\Gamma}[(1-\theta)\mathbf{I} + \mu\theta\mathbf{R}]^{-1}\mathbf{Q}\}}{\theta} + \frac{(1-\theta)(1-2\theta\lambda + \theta)\text{Tr}\{\boldsymbol{\Gamma}\mathbf{Q}\}}{\theta(1-\theta\lambda)},$ <p style="text-align: center;">with $\boldsymbol{\Gamma} = \mathbf{R}[\mu\mathbf{R} + (1-\lambda)\mathbf{I}]^{-1}$.</p>

Table 6: Analytical expressions for the steady-state ε -MSD considering LMS, RLS, their convex combination and for the Kalman filter.

ε -MSD
$\varepsilon^{\text{LMS}} = \mathbb{1}^T [2\mu\mathbf{\Lambda} - \mu^2\boldsymbol{\ell}\boldsymbol{\ell}^T - 2\mu^2\mathbf{\Lambda}^2]^{-1} \left\{ \mu^2\sigma_v^2\boldsymbol{\ell} + \frac{2(1-\theta)}{\theta} \left\{ \mathbf{I} - \left[\mathbf{I} + \frac{\mu\theta}{1-\theta}\mathbf{\Lambda} \right]^{-1} \right\} \text{diag}\{\bar{\mathbf{Q}}\} \right\},$ <p style="text-align: center;">with $\mathbf{\Lambda} = \mathbf{U}^T\mathbf{R}\mathbf{U}$, $\bar{\mathbf{Q}} = \mathbf{U}^T\mathbf{Q}\mathbf{U}$ and $\boldsymbol{\ell} = \text{diag}\{\mathbf{\Lambda}\}$.</p>
$\varepsilon^{\text{RLS}} = \mathbb{1}^T [2\lambda\mathbf{I} - (1-\lambda)\boldsymbol{\ell}^{-1}\boldsymbol{\ell}^T]^{-1} \left\{ \sigma_v^2(1-\lambda)\boldsymbol{\ell}^{-1} + \frac{2(1-\theta)}{1-\theta\lambda} \text{diag}\{\bar{\mathbf{Q}}\} \right\},$ <p style="text-align: center;">with $\bar{\mathbf{Q}} = \mathbf{U}^T\mathbf{Q}\mathbf{U}$, $\boldsymbol{\ell} = \text{diag}\{\mathbf{\Lambda}\}$ and $\boldsymbol{\ell}^{-1} = \text{diag}\{\mathbf{\Lambda}^{-1}\}$.</p>
$\varepsilon^{\text{COMB}} = \eta^{\circ 2}\varepsilon^{(1)} + (1-\eta^{\circ})^2\varepsilon^{(2)} + 2\eta^{\circ}(1-\eta^{\circ})\varepsilon^{(12)}, \text{ where } \eta^{\circ} = \frac{\zeta^{(2)} - \zeta^{(12)}}{\zeta^{(1)} - 2\zeta^{(12)} + \zeta^{(2)}},$ $\varepsilon^{(12)} = \mathbb{1}^T \left\{ (1-\lambda)\mathbf{I} + \mu\mathbf{\Lambda} - \mu(1-\lambda)(\mathbb{1}\boldsymbol{\ell}^T + 2\mathbf{\Lambda}) \right\}^{-1} \left\{ \begin{aligned} & \mu(1-\lambda)\sigma_v^2\mathbb{1} - \frac{(1-\theta)^2}{\theta}[(1-\theta)\mathbf{I} + \mu\theta\mathbf{\Lambda}]^{-1}\text{diag}\{\bar{\mathbf{Q}}\} \\ & + \frac{(1-\theta)(1-2\theta\lambda + \theta)}{\theta(1-\theta\lambda)}\text{diag}\{\bar{\mathbf{Q}}\} \end{aligned} \right\},$ <p style="text-align: center;">with $\mathbf{\Lambda} = \mathbf{U}^T\mathbf{R}\mathbf{U}$, $\bar{\mathbf{Q}} = \mathbf{U}^T\mathbf{Q}\mathbf{U}$ and $\boldsymbol{\ell} = \text{diag}\{\mathbf{\Lambda}\}$.</p>
$\varepsilon^{\text{KAL}} = \text{Tr}\{\bar{\mathcal{P}}_{\infty}\}, \text{ where } \bar{\mathcal{P}}_{\infty} \text{ is the solution of}$ $\bar{\mathcal{P}}_{\infty} \approx \theta^2\bar{\mathcal{P}}_{\infty} + (1-\theta^2)\mathbf{Q} - \theta^2\bar{\mathcal{P}}_{\infty}\mathbf{R} [\sigma_v^2\mathbf{R} + \mathbf{R}\bar{\mathcal{P}}_{\infty}\mathbf{R}]^{-1} \mathbf{R}\bar{\mathcal{P}}_{\infty}.$

3.4 Simulations

In order to verify the tracking behavior of the proposed model, we consider a system identification application to compare the performance of the well known LMS and RLS filters and their convex combination with the corresponding optimum solution obtained via Kalman filter.

The unknown plant \mathbf{w}_i° , of length $M = 7$, was initialized with random values in the interval $[-1, 1]$. The solution is then changed at each iteration according to the random-walk model (3.3). Following [14], the covariance matrix for \mathbf{q}_i is given by

$$\mathbf{Q} = \delta \left[\beta \frac{\mathbf{R}}{\text{Tr}\{\mathbf{R}\}} + (1-\beta) \frac{\mathbf{R}^{-1}}{\text{Tr}\{\mathbf{R}^{-1}\}} \right], \quad (3.53)$$

where constant δ has been selected to be $\delta = 5 \cdot 10^{-2}$, so that $\text{Tr}\{\mathbf{Q}\} = \delta$, and $\beta \in [0, 1]$ is

a control parameter that allows to trade off between a situation with $\mathbf{Q} \propto \mathbf{R}$ (for $\beta = 1$, this is the situation in which LMS outperforms RLS according to [50]) and $\mathbf{Q} \propto \mathbf{R}^{-1}$ ($\beta = 0$, the case in which RLS outperforms LMS).

The regressor \mathbf{u}_i is obtained from a process $u(i)$ as $\mathbf{u}_i = [u(i) \ u(i-1) \ \dots \ u(i-M+1)]^T$, where $u(i)$ is generated with a first-order autoregressive (AR) model, whose transfer function is $\sigma_u \sqrt{(1-b^2)/(1-bz^{-1})}$ with $b = 0.8$. This model is fed with an i.i.d. Gaussian random process with variance $\sigma_u^2 = \frac{1}{7}$, so that $\text{Tr}\{\mathbf{R}\} = 1$. The output additive noise $v(i)$ is i.i.d. Gaussian with zero mean and variance $\sigma_v^2 = 10^{-2}$.

Regarding the adjustment for the combinations, we used convex combinations with fixed step-size $\mu_a = 0.25$ and the auxiliary variable $a(i)$ restricted to the interval $[-4, 4]$, while the optimum step-size and forgetting factor of the constituent filters were numerically obtained through the theoretical steady-state EMSE general equations (3.31) and (3.40) (that is, we find the optimum values without resorting to the approximations for $\mu \approx 0$ and $\lambda \approx 1$).

To begin with, let us start the tracking analysis considering the parameter $\beta = 0.05$ in (3.53). In this case, according to (3.38) and (3.45), the minimum θ that can be used for the LMS and RLS filters in order to have an optimum step-size $\mu^\circ > 0$ and optimum forgetting factor $0 \ll \lambda^\circ < 1$ is $\theta_{\min}^{\text{LMS}} \approx 0.84$ and $\theta_{\min}^{\text{RLS}} \approx 0.92$. The bounds obtained numerically from the general equations (3.31) and (3.40) are $\theta_{\min}^{\text{LMS}} \approx 0.88$ and $\theta_{\min}^{\text{RLS}} \approx 0.94$. Figure 6 shows the steady-state EMSEs estimated from the ensemble-average learning curve obtained from 600 independent runs for 30.000 iterations of the algorithms μ° -LMS, λ° -RLS, their combination and the corresponding Kalman filter when $0 < \theta < 1$. Figure 7 compares the tracking performance between the simulated case and the theoretical steady-state EMSE equations provided in table 4 when θ lies in the range $[\min\{\theta_{\min}^{\text{LMS}}, \theta_{\min}^{\text{RLS}}\}, 1]$.

As can be seen in Figures 6 and 7, for $\theta \approx 1$, the steady-state EMSE achieved by combining both LMS and RLS filters with optimum settings is close to the optimum EMSE obtained via Kalman filter. As θ decreases, since the speed of change of \mathbf{w}_i° is increased, the difference between the combination and the KF curves starts to increase, reaching its maximum value of approximately 0.80 dB for $\theta \approx \theta_{\min}^{\text{LMS}}$. For $\theta < \min\{\theta_{\min}^{\text{LMS}}, \theta_{\min}^{\text{RLS}}\}$, the adaptive filters LMS and RLS as well as their combination, are no longer tracking the variations of \mathbf{w}_i° since $\mu^\circ \approx 0$ and $\lambda^\circ = 1$. For $\theta < 0.2$, the resulting performance for all filters is approximately the same. The values of μ° and λ° as θ changes from $\min\{\theta_{\min}^{\text{LMS}}, \theta_{\min}^{\text{RLS}}\}$ to 1 are shown in Figure 8.

To see the effect of assumptions of small step-sizes and λ close to one, in Figure 8

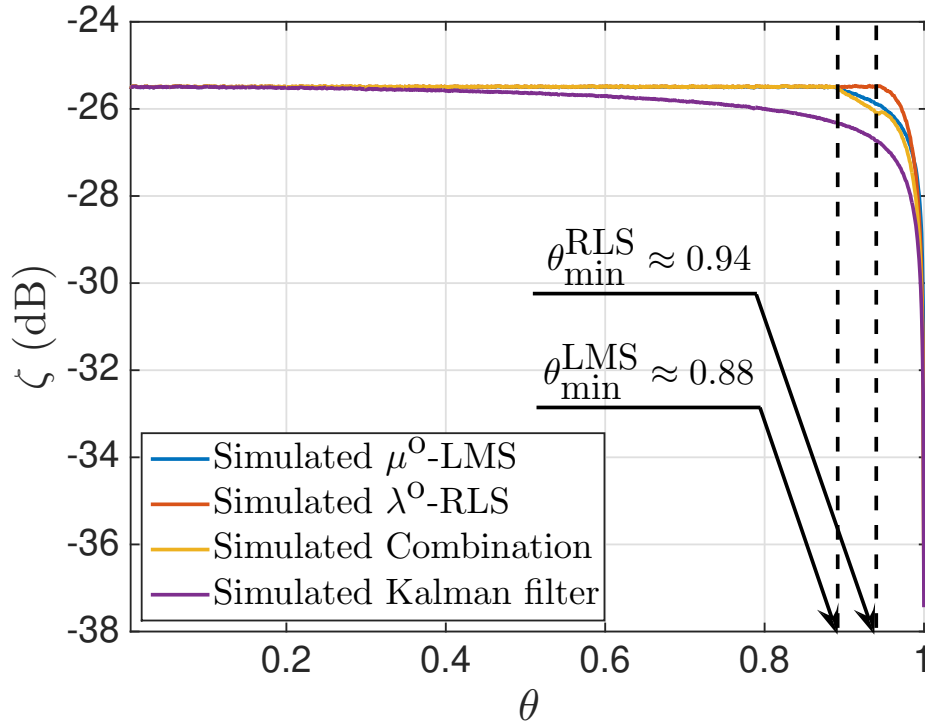


Figure 6: Simulated steady-state EMSE curves for μ° – LMS, λ° – RLS, their convex combination and the corresponding Kalman filter when $0 < \theta < 1$.

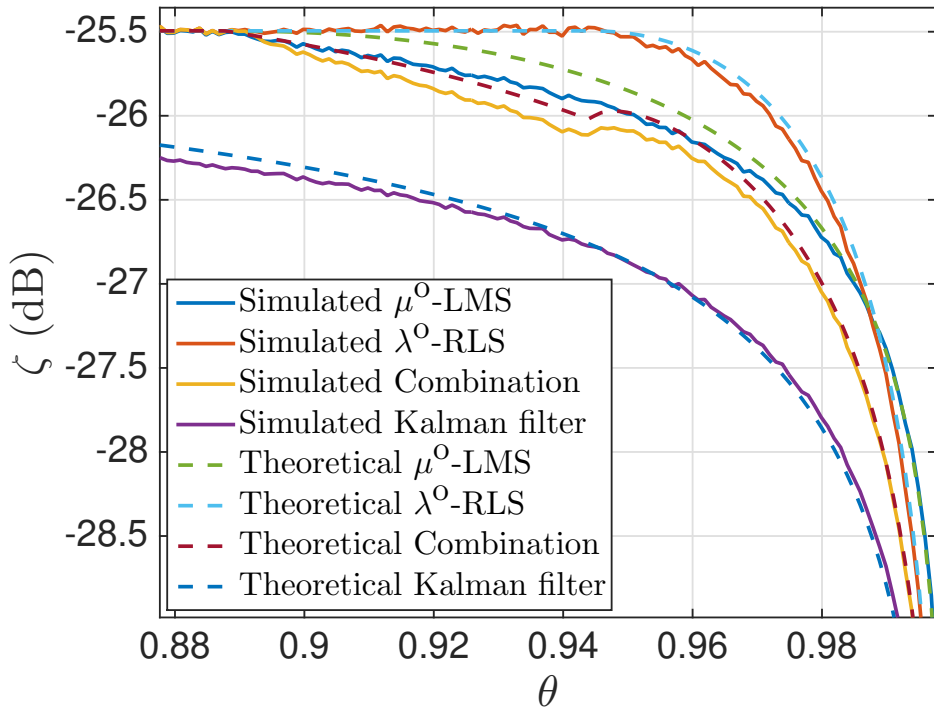


Figure 7: Tracking comparison between the theoretical steady-state EMSE equations from table 4 and the simulated case for $\min\{\theta_{\min}^{\text{LMS}}, \theta_{\min}^{\text{RLS}}\} < \theta < 1$.

we also plot the approximated curves obtained through expressions (3.36) and (3.44). As can be seen in this figure, for $\theta < 1$, the error between the optimum parameters estimated numerically from the general equations provided in table 4 and by using equations (3.36) and (3.44) is at most $\Delta_{\mu^o} \approx 0.05$ for μ^o and $\Delta_{\lambda^o} \approx 0.01$ for λ^o .

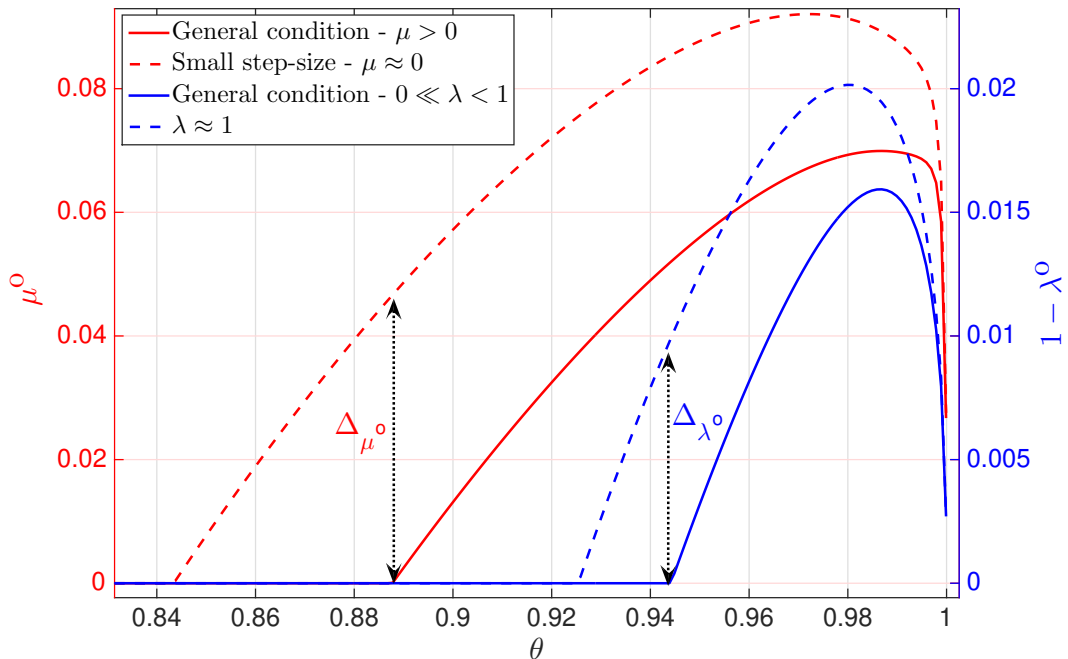


Figure 8: Variation of the optimum parameters for $\min\{\theta_{\min}^{\text{LMS}}, \theta_{\min}^{\text{RLS}}\} < \theta < 1$ considering the general and the approximated equations provided in tables 4 and 5.

By computing the ratio $\theta_{\min}^{\text{LMS}}/\theta_{\min}^{\text{RLS}}$, we are able to compare the θ range allowed for each filter and see if they have the same range ($\theta_{\min}^{\text{LMS}} = \theta_{\min}^{\text{RLS}}$) or if one filter is more restrictive than the other ($\theta_{\min}^{\text{LMS}} > \theta_{\min}^{\text{RLS}}$ or $\theta_{\min}^{\text{LMS}} < \theta_{\min}^{\text{RLS}}$). The ratio between (3.38) and (3.45) is given by

$$\frac{\theta_{\min}^{\text{LMS}}}{\theta_{\min}^{\text{RLS}}} = \frac{\text{Tr}\{\mathbf{R}\}\{2\text{Tr}\{\mathbf{R}\mathbf{Q}\} + \sigma_v^2 M\}}{M\{\sigma_v^2 \text{Tr}\{\mathbf{R}\} + 2\text{Tr}\{\mathbf{R}^2 \mathbf{Q}\}\}}. \quad (3.54)$$

With this expression, we can see that the value of θ_{\min} for LMS can be smaller or larger than that for RLS, depending on the values of \mathbf{R} and \mathbf{Q} . As a simple example, consider $\sigma_v^2 = 0.01$, $\mathbf{R} = \text{diag}(1, 2)$. If $\mathbf{Q} = \text{diag}(0.01, 0.001)$, then $\theta_{\min}^{\text{LMS}}/\theta_{\min}^{\text{RLS}} = 1.14$, while, for $\mathbf{Q} = \text{diag}(0.001, 0.01)$ we have $\theta_{\min}^{\text{LMS}}/\theta_{\min}^{\text{RLS}} = 0.830$.

By using the theoretical steady-state EMSE equations summarized in table 4, Figure 9 compares the tracking behavior between the LMS, RLS, their combination and the Kalman filter when $\beta \in [0, 1]$ and $\theta = 0.99$. For this simulation we kept the same matrices \mathbf{R} and \mathbf{Q} as well as the parameters σ_v^2 and M used to obtain figure 6.

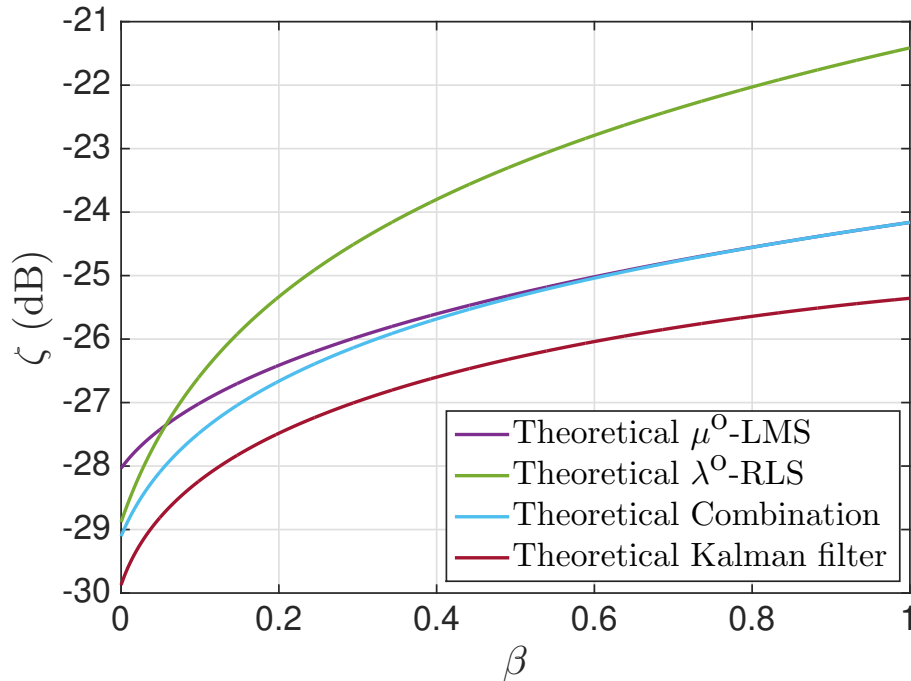


Figure 9: EMSE of μ^o -LMS, λ^o -RLS, their convex combination and the Kalman filter when \mathbf{Q} smoothly changes between \mathbf{R} and \mathbf{R}^{-1} .

As shown in Figure 9, depending on the value of β , the optimal EMSE for RLS filters can be larger or smaller than the optimal EMSE for LMS filters, but always lower bounded by the optimal EMSE provided by the Kalman filter. For the general equations in which $\mu > 0$ and $0 \ll \lambda < 1$, the optimal steady-state EMSE curve for the combination is approximately 1dB larger than the optimal KF. However, compared to the $\mathcal{O}(M^2)$ complexity of the KF, if lattice [42] or DCD [3] implementations are used for RLS, the computational cost of the combination is reduced from $\mathcal{O}(M^2)$ to $\mathcal{O}(M)$.

Figure 10 plots the MSD as a function of β , under the same conditions. Note that we plotted two curves for the combination — one (green) using the value of the mixing parameter that is optimal for the MSD, and the second (yellow) using the optimum mixing parameter for the EMSE, which the filter can estimate in practice. It can be seen that both are close, but the practical curve is slightly worse than the best filter at a few points.

In order to see the influence of the filter length M in the optimal steady-state EMSE of each filter, Figure 11 shows the tracking behavior for the theoretical and simulated μ^o -LMS, λ^o -RLS, their convex combination and for the KF when M changes from 1 to 200. For this simulation we used the general equations provided in Table 4 and compared with the simulated case using the following parameters $\sigma_v^2 = 10^{-2}$, $\text{Tr}\{\mathbf{Q}\} = 5 \cdot 10^{-2}$, $\beta = 0.05$, $\sigma_u^2 = \frac{1}{M}$ to keep $\text{Tr}\{\mathbf{R}\} = 1$ and $\theta = 0.9999$ to cover all possible values of $\theta_{\min}^{\text{LMS}}$ and $\theta_{\min}^{\text{RLS}}$. In addition to the simulated case, we used convex combinations with fixed

step-size $\mu_a = 0.25$, the auxiliary variable $a(i)$ restricted to the interval $[-4, 4]$ and the optimal steady-state EMSEs were estimated from the ensemble-average learning curve obtained from 50 independent runs and 10^6 iterations for each algorithm. Due to the long processing time, only a few points were plotted for the simulated curves.

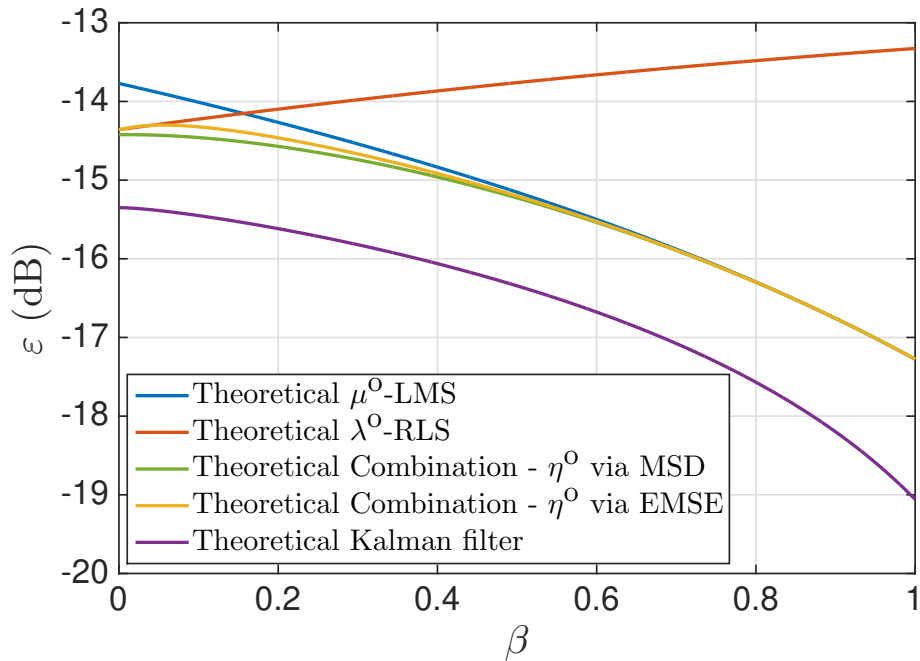


Figure 10: MSD of μ^o -LMS, λ^o -RLS, their convex combination and the Kalman filter when \mathbf{Q} smoothly changes between \mathbf{R} and \mathbf{R}^{-1} .

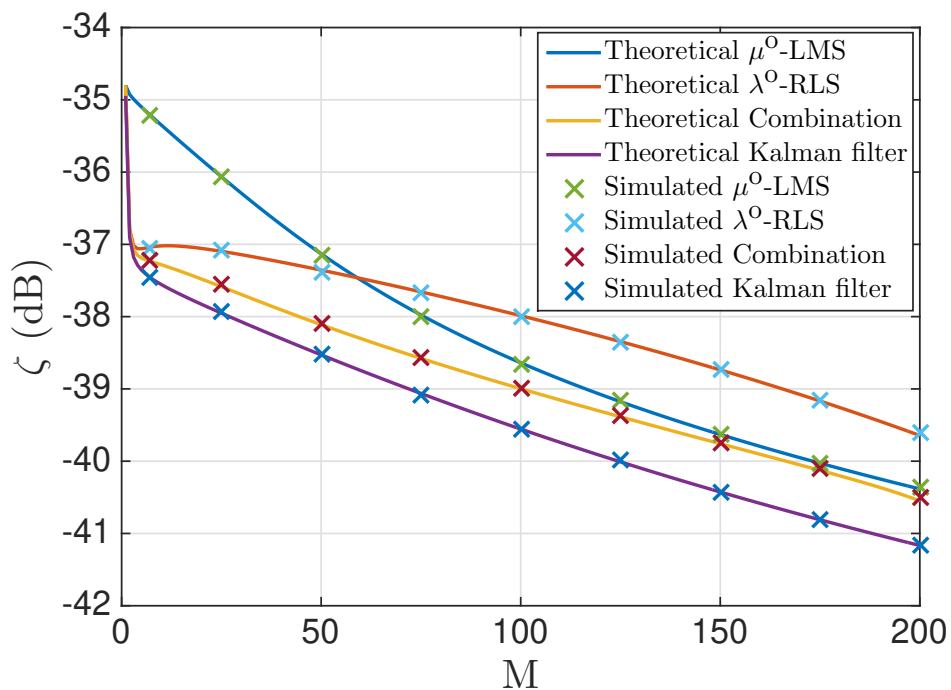


Figure 11: Theoretical and simulated steady-state EMSE for μ^o -LMS, λ^o -RLS, their convex combination and the Kalman filter for $1 \leq M \leq 200$.

As depicted in Figure 11, for the defined range M , the optimal steady-state EMSE for the KF is at most 0.6 dB smaller than the combination when $M > 100$, despite the computational cost for the KF increasing at a rate of $\mathcal{O}(M^2)$ while the combination increases linearly with the filter length.

3.4.1 DC analysis of the steady-state EMSE

All simulations performed until now considered a zero mean plant \mathbf{w}_i^o , whose values are updated at each iteration according to model (3.3), and the parameter θ was always kept greater than $\min\{\theta_{\min}^{\text{LMS}}, \theta_{\min}^{\text{RLS}}\}$ since we were dealing with tracking problems. Let us assume now that $\theta < \min\{\theta_{\min}^{\text{LMS}}, \theta_{\min}^{\text{RLS}}\}$ and the model (3.3) has a DC component added to the optimum vector \mathbf{w}_i^o , i.e.,

$$\mathbf{w}_i^o = \theta \mathbf{w}_{i-1}^o + \sqrt{1 - \theta^2} \mathbf{q}_i + \bar{\mathbf{w}}^o, \quad (3.55)$$

where $\bar{\mathbf{w}}^o = \mathbb{E}\{\mathbf{w}_i^o\}$ is an $M \times 1$ vector.

The simulation in Figure 12 compares the performance of the Kalman filter, RLS with $\lambda = 1$ and LMS with decreasing μ (see below) considering both a zero and a nonzero DC component. We can see that the adaptive filters, although not able to track the variations of \mathbf{w}_i^o (since $\mu^o = 0$ and $\lambda^o = 1$), are still able to converge to the DC component $\bar{\mathbf{w}}^o$, reaching the same optimum steady-state EMSE regardless of the value assumed for $\bar{\mathbf{w}}^o$. For this simulation we plot the EMSEs estimated from the ensemble-average learning curve of 600 independent runs for 3000 iterations of the algorithms LMS, RLS, their convex combination and the KF considering $\theta = 0.5$ and the same matrices \mathbf{R} and \mathbf{Q} as well as the parameters σ_v^2 , β and M used to obtain figure 6.

The following assumptions had to be made in order to ensure convergence to the optimum solution for both LMS algorithm and for the KF when $\bar{\mathbf{w}}^o > \mathbf{0}$:

- LMS: the required $\mu^o = 0$ when $\theta < \theta_{\min}^{\text{LMS}}$ can not be used, since the filter will not be able to adapt and converge to the mean value of \mathbf{w}_i^o . For this reason, we assumed fixed step-size $\mu = 10^{-8}$ when $\bar{\mathbf{w}}^o = \mathbf{0}$ and variable step-size $\mu(i) = 1/i$ for nonzero $\bar{\mathbf{w}}^o$ to ensure equivalence to the RLS case with $\lambda = 1$ [51]. To speed up the convergence rate for nonzero $\bar{\mathbf{w}}^o$, we used $\mu(i) = 1/\sqrt{i}$ for $i < 1000$.
- KF: we added the *a priori* information of the DC value to the estimate $\hat{\mathbf{x}}_i$ since the KF equations assumes that \mathbf{x}_i is zero mean.

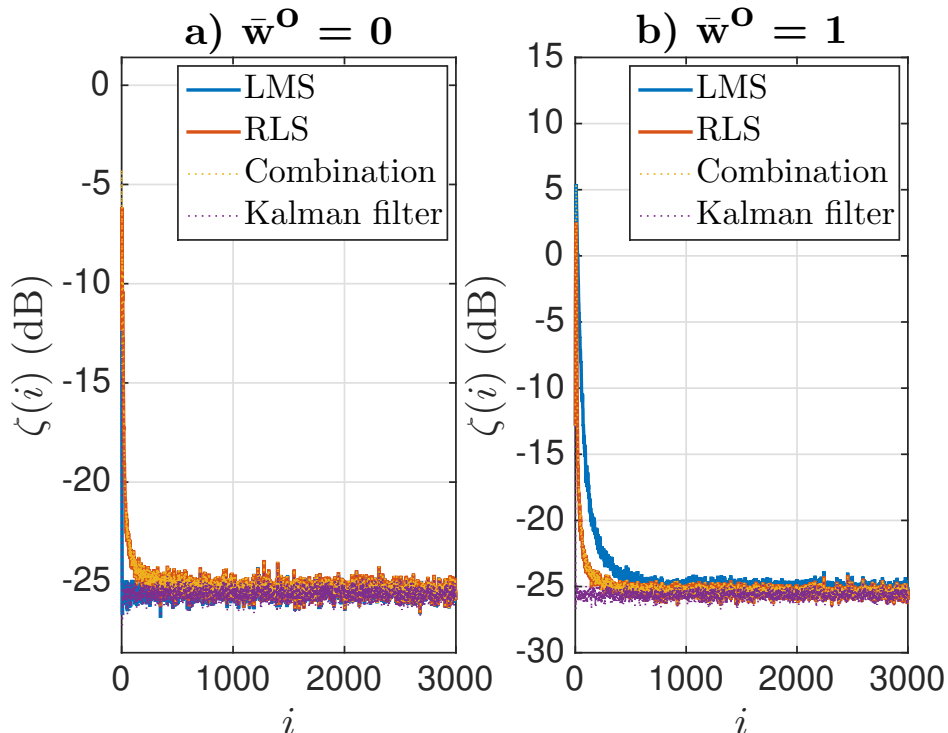


Figure 12: Simulated EMSEs learning-curves for $\theta = 0.5$ and considering two different scenarios: a) $\bar{\mathbf{w}}^o = \mathbf{0}$ and b) $\bar{\mathbf{w}}^o = \mathbf{1}$. To have a better visualization of the result, we used dotted lines to represent the combination and the KF curves.

3.4.2 Comparison between θ_{\min} and the nonstationarity degree

The nonstationarity degree (NSD), defined in [44], is another measure of when an adaptive filter is able to track a time-varying weight vector. It compares the performance (measured by the steady-state EMSE) of an adaptive filter to estimate $\mathbf{u}_i^T \mathbf{w}_{i-1}^o$ with the estimate provided by $d(i)$ itself. Let $\mathcal{F} = \{\mathbf{w}_{i-1} \in \mathbb{R}^M : \mathbf{w}_{i-1} \text{ depends only on } d(j), \mathbf{u}_j \text{ for } j < i\}$ be the set of possible a-priori estimates. The NSD is defined as

$$\text{NSD} = \frac{\min_{\mathbf{w}_{i-1} \in \mathcal{F}} \mathbb{E} \{ [\mathbf{u}_i^T (\mathbf{w}_{i-1} - \mathbf{w}_{i-1}^o)]^2 \}}{\sigma_v^2}. \quad (3.56)$$

If $\text{NSD} > 1$, [44] argues that an adaptive filter is not able to track \mathbf{w}_i^o . Under our model for \mathbf{w}_i^o (3.3), the NSD is given by

$$\text{NSD} = (1 - \theta^2) \frac{\text{Tr}\{\mathbf{RQ}\}}{\sigma_v^2}. \quad (3.57)$$

Imposing the condition $\text{NSD} = 1$, we obtain

$$\theta_{\min}^{\text{NSD}} = \sqrt{\max \left\{ 0, 1 - \frac{\sigma_v^2}{\text{Tr}\{\mathbf{RQ}\}} \right\}}. \quad (3.58)$$

If $\theta < \theta_{\min}^{\text{NSD}}$, all a-priori estimates would have an EMSE larger than σ_v^2 , and therefore if the goal of the adaptive filter were to estimate $\mathbf{u}_i^T \mathbf{w}_{i-1}^o$, no adaptive filter would do better than simply using $d(i)$ as an estimate (note that $\theta_{\min}^{\text{NSD}}$ tends to 1 as $\sigma_v^2 \rightarrow 0$).

The minimum values for θ defined in this chapter complement the NSD in two ways: (a) we can also define a condition for tracking based on the MSD, while the NSD considers only the EMSE; (b) our estimates for θ_{\min} take into account the DC part of the weight vector, as described in Section 3.4.1 — that is, a situation in which an adaptive filter is useful to estimate the DC part of the weight vector, but not to track variations around that DC value.

Figure 13 compares the values of θ_{\min} as derived in this chapter, for both EMSE and MSD, with $\theta_{\min}^{\text{NSD}}$, considering the same case as in Figure 9, but with $\sigma_v^2 = 10^{-3}$. It can be seen on the left panel that the values of θ_{\min} obtained in this chapter and through the NSD are quite different: the condition we use to define when a filter is no longer able to track is different than the condition used for the NSD.

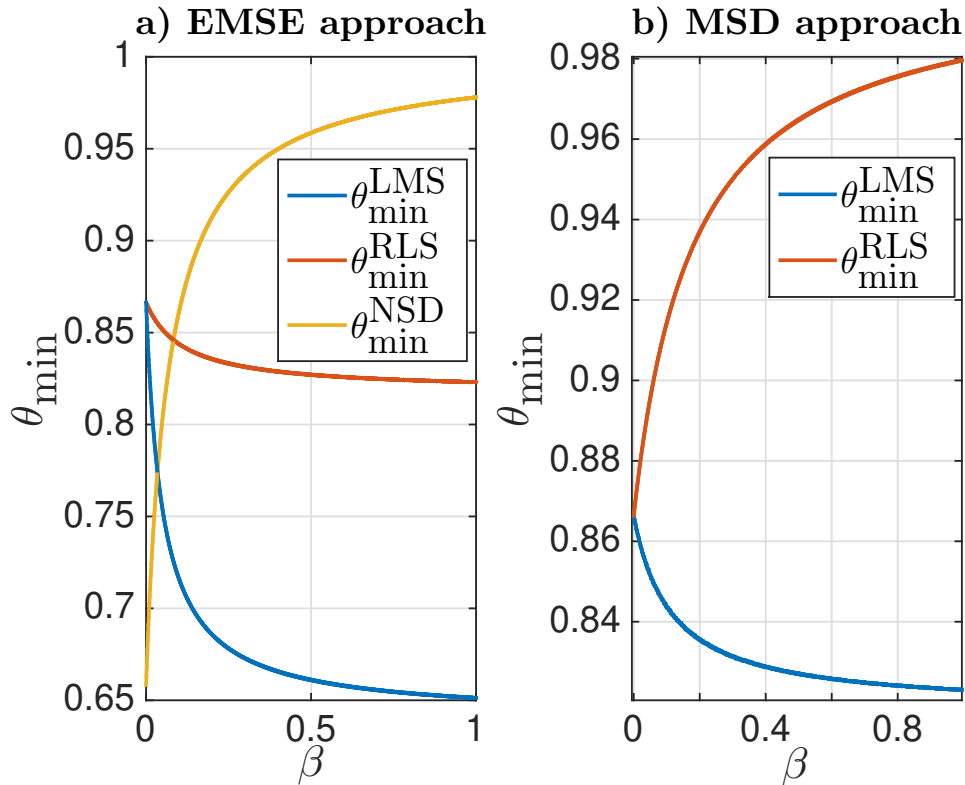


Figure 13: Values of θ_{\min} considering the EMSE and NSD (a), and the MSD (b).

3.5 Conclusion

Combination approaches are an effective way to improve the performance of adaptive filters. In this chapter we have studied the tracking performance of combinations between LMS and RLS filters and compared the resulting EMSE with the optimal case obtained via Kalman filter.

We have shown that, assuming a first order AR model with finite autocovariance matrix to describe the time variation of the unknown parameter vector, it is possible to achieve through convex combination between LMS and RLS filters a steady-state EMSE and MSD performance close to the optimal case obtained via Kalman filter, as long as the pole of the AR model is greater than a minimum value. This remains true even if the unknown parameter vector has a DC component. The minimum value for the pole of the AR model provides a model for how fast a plant can vary so that an adaptive filter can still track it.

The advantage of our approach using combinations arises from the fact that the combination can be implemented with complexity $\mathcal{O}(M)$, while it takes at least $\mathcal{O}(M^2)$ operations to compute the corresponding Kalman filter for the model considered here.

4 LOW-COMPLEXITY APPROXIMATION TO THE KALMAN FILTER USING THE DICHOTOMOUS COORDINATE DESCENT ALGORITHM

The Kalman filter (KF) is a simple and elegant algorithm formulated almost 60 years ago [52], as an optimal recursive Bayesian estimator for a class of linear Gaussian problems [53]. It is used in areas as diverse as aeronautics, signal processing, and futures trading [28], [54]. On the other hand, we have also the class of adaptive filters, which does not require any information about the model being estimated. For a filter of size M , the Recursive Least Squares (RLS) algorithm for example, converges significantly faster than other filters such as the least mean squares (LMS) algorithm [2, 47], and requires $\mathcal{O}(M^2)$ operations per iteration to compute the solution, while the Kalman filter requires $\mathcal{O}(M^{2.376})$ [31] operations per iteration. Given that the $\mathcal{O}(M^2)$ RLS algorithm was shown to be equivalent to the Kalman filter for a particular state-space model [2, Sec. 31.2], [55], and that it is possible to reduce the complexity of RLS to $\mathcal{O}(M)$ operations per iteration by using lattice or Dichotomous Coordinate Descent (DCD) techniques [2, 3, 47], in this chapter we show that there exist other models for the evolution of the optimum weight vector for which it is possible to derive fast (i.e., $\mathcal{O}(M)$) versions of the Kalman filter, extending the RLS-DCD algorithm of [3].

In the next section we describe briefly the connection between RLS and the Kalman filter. In Section 4.2, we describe new conditions on the state transition matrix for the Kalman model that allow the development of $\mathcal{O}(M)$ versions of the Kalman filter. Comparisons between the new low-complexity algorithms and both the Kalman filter and the standard RLS algorithm are provided in Section 4.3, and Section 4.4 concludes the chapter.

4.1 Problem Formulation

Consider a state-space model (see (1.38)) of the form:

$$\mathbf{x}_i = \mathbf{F}_i \mathbf{x}_{i-1} + \mathbf{G}_i \mathbf{t}_i, \quad (4.1a)$$

$$z(i) = \mathbf{h}_i^T \mathbf{x}_{i-1} + \varphi(i), \quad (4.1b)$$

where \mathbf{x}_i is the $M \times 1$ state vector, \mathbf{F}_i is the $M \times M$ state-transition matrix, \mathbf{G}_i is an $M \times N$ matrix, \mathbf{t}_i is an $N \times 1$ Gaussian random process noise vector with zero mean and covariance matrix \mathbf{T}_i . We assume here for simplicity that the observation variable $z(i)$ is scalar, \mathbf{h}_i^T is the $1 \times M$ measurement vector, and $\varphi(i)$ is Gaussian measurement noise with zero mean and variance $\sigma^2(i)$.

In this case, given observations $z(i)$ that satisfy the state-space model described in (4.1), an estimate $\hat{\mathbf{x}}_i$ of \mathbf{x}_i can be recursively computed by using the following set of Kalman filter equations [2]:

$$\beta(i) = \sigma^2(i) + \mathbf{h}_i^T \mathcal{P}_{i-1} \mathbf{h}_i, \quad (4.2a)$$

$$\mathbf{k}_i = (\mathbf{F}_i \mathcal{P}_{i-1} \mathbf{h}_i + \mathbf{G}_i \mathcal{S}_i) \beta^{-1}(i), \quad (4.2b)$$

$$\nu(i) = z(i) - \mathbf{h}_i^T \hat{\mathbf{x}}_{i-1}, \quad (4.2c)$$

$$\hat{\mathbf{x}}_i = \mathbf{F}_i \hat{\mathbf{x}}_{i-1} + \mathbf{k}_i \nu(i), \quad (4.2d)$$

$$\mathcal{P}_i = \mathbf{F}_i \mathcal{P}_{i-1} \mathbf{F}_i^T + \mathbf{G}_i \mathbf{T}_i \mathbf{G}_i^T - \beta(i) \mathbf{k}_i \mathbf{k}_i^T, \quad (4.2e)$$

where \mathcal{P}_{i-1} is the $M \times M$ error covariance matrix defined as $\mathbb{E}\{(\mathbf{x}_{i-1} - \hat{\mathbf{x}}_{i-1})(\mathbf{x}_{i-1} - \hat{\mathbf{x}}_{i-1})^T\}$, \mathbf{k}_i is the $M \times 1$ Kalman gain, $\mathcal{S}_i = \mathbb{E}\{\mathbf{t}_i \varphi(i)\}$ is the cross-covariance vector between the noise processes and $\nu(i)$ is a scalar.

As we can see in equations (4.2b) and (4.2e), one needs $\mathcal{O}(M^3)$ operations to compute directly the covariance matrix \mathcal{P}_i (due to the matrix-matrix product $\mathbf{F}_i \mathcal{P}_{i-1} \mathbf{F}_i^T$). There are faster options, for example, the $\mathcal{O}(M^{2.376})$ algorithm of [31], and, for constant \mathbf{F} , \mathbf{G} , \mathbf{H} , σ^2 and \mathbf{T} , the $\mathcal{O}(M^2)$ array forms [2, p. 618]. However, even with these efficient algorithms, this computational cost may be prohibitive for applications in which M is large (such as echo cancellation).

The RLS algorithm, on the other hand, assumes no model for the evolution of the unknown vectors \mathbf{w}_i^o that it tries to estimate [1, 2, 47]. The only assumption is that \mathbf{w}_i^o is

related to available observations $d(i)$ and \mathbf{u}_i as follows

$$d(i) = \mathbf{u}_i^T \mathbf{w}_{i-1}^o + v(i), \quad (4.3)$$

where \mathbf{u}_i is an $M \times 1$ regressor vector, $d(i)$ is a scalar, and $v(i)$ is an unknown scalar measurement noise with zero mean and variance σ_v^2 uncorrelated to \mathbf{u}_i . The RLS recursions are given by

$$\gamma(i) = \frac{1}{1 + \lambda^{-1} \mathbf{u}_i^T \mathbf{P}_{i-1} \mathbf{u}_i}, \quad (4.4a)$$

$$\mathbf{g}_i = \lambda^{-1} \mathbf{P}_{i-1} \mathbf{u}_i \gamma(i), \quad (4.4b)$$

$$e(i) = d(i) - \mathbf{u}_i^T \mathbf{w}_{i-1}, \quad (4.4c)$$

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mathbf{g}_i e(i), \quad (4.4d)$$

$$\mathbf{P}_i = \lambda^{-1} \mathbf{P}_{i-1} - \mathbf{g}_i \mathbf{g}_i^T / \gamma(i), \quad (4.4e)$$

where $0 \ll \lambda < 1$ is a forgetting factor.

Given this, it can be verified by direct substitution that the RLS recursions (4.4) can be obtained from the Kalman recursions (4.2) with the correspondence of Table 7 and for the particular state-space model with [2, p. 513], [55]

$$\mathbf{F}_i = \mathbf{I}, \quad \mathbf{G}_i = \mathbf{0}, \quad \mathbf{h}_i^T = \mathbf{u}_i^T, \quad \sigma^2(i) = \lambda^i, \quad (4.5)$$

where \mathbf{I} is the identity matrix. Table 7 lists the relations between Kalman and RLS variables.

Description	Kalman	RLS	Description
Measurement	$z(i)$	$d(i)$	Reference Signal
State estimator	$\hat{\mathbf{x}}_i$	\mathbf{w}_i	Weight estimate
Error covariance	\mathcal{P}_i	$\lambda^i \mathbf{P}_i$	(Input cov. matrix) ⁻¹
Innovations	$\nu(i)$	$e(i)$	A priori error
Meas. noise variance	$\sigma^2(i)$	σ_v^2	Noise variance

Table 7: Correspondence between Kalman and RLS variables.

As shown in [2, p. 506], there are other ways to derive the RLS recursions from the KF equations. However, the formulation used next helps to understand and visualize the connection between the RLS and the KF variables.

Although a direct implementation of (4.4) requires $\mathcal{O}(M^2)$ operations, several different alternatives with $\mathcal{O}(M)$ complexity are available when the regressor vector \mathbf{u}_i follows a

delay-line structure, i.e., if

$$\mathbf{u}_i = \begin{bmatrix} u(i) & u(i-1) & \dots & u(i-M+1) \end{bmatrix}^T, \quad (4.6)$$

where $u(i)$ is a scalar sequence. RLS algorithms with linear complexity include the Fast Transversal Filter (FTF) [2, 56], lattice-RLS filters [2, 57], and the RLS-DCD algorithm [3, 47]. Since the RLS-DCD algorithm is numerically stable (unlike FTF [2]), and provides \mathbf{w}_i (unlike lattice filters, which do not compute \mathbf{w}_i explicitly), in the remainder of this chapter we will concentrate on the RLS-DCD algorithm, and show that there are other, more general state-space models for which linear-complexity versions of the Kalman filter can be obtained, extending the RLS-DCD algorithm to a Kalman-DCD algorithm.

4.2 Kalman-DCD

Instead of solving directly the time and measurement-update equations (4.2) of the Kalman filter, we follow [58] and use a different way to compute the state-vector $\hat{\mathbf{x}}_i$ by using an alternative form of the filter which does not require the explicit calculation of equations (4.2a), (4.2b) and (4.2e) in the way they appear in the standard Kalman filter algorithm, which is

$$\hat{\mathbf{x}}_{i|i} = \mathbf{F}_i(\hat{\mathbf{x}}_{i-1|i-1} + \boldsymbol{\alpha}_i), \quad (4.7)$$

where the notation $\hat{\mathbf{x}}_{i|i}$ is used to define the estimate of \mathbf{x}_i given the observations $\{z(0), z(1), \dots, z(i-1), z(i)\}$ and $\boldsymbol{\alpha}_i$ is the solution to the following linear equation:

$$\mathcal{P}_{i|i}^{-1} \boldsymbol{\alpha}_i = (1/\sigma^2(i)) \mathbf{h}_i \nu(i). \quad (4.8)$$

The difficulty in solving (4.8) is that a general solution involves a number of operations of the order of $\mathcal{O}(M^{2.376})$ operations for a filter with M coefficients as mentioned in [31]. This is usually too costly for practical applications, except perhaps for very short filters. We show next that this complexity can be reduced to $\mathcal{O}(M)$ operations for a certain class of matrices \mathbf{F}_i , \mathbf{G}_i and \mathbf{h}_i , by using DCD to solve (4.8), and by further generalizing the recursion for \mathcal{P}_i in (4.2e).

In order to obtain this simplified recursion, we use instead of (4.2), the following relation which holds when the required inverses exist [58]:

$$\mathcal{P}_{i|i}^{-1} = \mathcal{P}_{i|i-1}^{-1} + (1/\sigma^2(i)) \mathbf{h}_i \mathbf{h}_i^T, \quad (4.9)$$

where matrix $\mathcal{P}_{i|i-1}$ is defined as:

$$\mathcal{P}_{i|i-1} = \mathbf{F}_i \mathcal{P}_{i-1|i-1} \mathbf{F}_i^T + \mathbf{G}_i \mathbf{T}_i \mathbf{G}_i^T. \quad (4.10)$$

By replacing equation (4.10) in (4.9) and restricting our state-space model to a problem where \mathbf{F}_i is a constant diagonal matrix given by \mathbf{F} and the random process noise is zero (which means $\mathbf{T}_i = \mathbf{0}$), we obtain the following expression:

$$\mathcal{P}_{i|i}^{-1} = \mathbf{F}^{-T} \mathcal{P}_{i-1|i-1}^{-1} \mathbf{F}^{-1} + (1/\sigma^2(i)) \mathbf{h}_i \mathbf{h}_i^T, \text{ for } i \geq 0. \quad (4.11)$$

By setting $\mathcal{P}_{-1|-1}^{-1} = \epsilon \mathbf{I}$ (with $\epsilon \approx 0$) as the initial condition for the covariance matrix $\mathcal{P}_{i|i}^{-1}$ in equation (4.11) and assuming that the variables \mathbf{F}^{-1} and \mathbf{h}_i^T are defined as:

$$\mathbf{F}^{-1} = \begin{bmatrix} f_{11} & 0 & 0 & \dots & 0 \\ 0 & f_{22} & 0 & \dots & 0 \\ 0 & 0 & f_{33} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & f_{MM} \end{bmatrix} \text{ and}$$

$$\mathbf{h}_i^T = \begin{bmatrix} x(i) & x(i-1) & x(i-2) & \dots & x(i-M+1) \end{bmatrix},$$

where $x(i)$ is a scalar signal, we show below that if the elements of \mathbf{F}^{-1} obey the relations

$$f_{11}^2 = f_{22}^2 = f_{33}^2 = \dots = f_{MM}^2, \quad (4.12)$$

$$f_{\ell\ell} f_{mm} = f_{(\ell+1)(\ell+1)} f_{(m+1)(m+1)}, \quad (4.13)$$

for $1 \leq \ell \leq M-2$, $2 \leq m \leq M-1$ and $\ell \neq m$, the matrices $\mathcal{P}_{i|i}^{-1}$ and $\mathcal{P}_{i-1|i-1}^{-1}$ share a common structure as i increases. More specifically, we claim that for a tapped-delay line measurement vector \mathbf{h}_i^T and if ϵ is sufficiently small so that the term related to the initial condition can be disregarded, the computation of $\mathcal{P}_{i|i}^{-1}$ follows the expression

$$[\mathcal{P}_{i|i}^{-1}]_{\ell,m} = \frac{1}{\sigma^2(i)} \sum_{j=0}^i f_{\ell\ell}^j f_{mm}^j x(i-j-\ell+1) x(i-j-m+1), \quad (4.14)$$

where the notation $[\cdot]_{\ell,m}$ indicates the line ' ℓ ' and the column ' m ' for $1 \leq \ell \leq M$, $1 \leq m \leq M$. Using the conditions described in (4.12) and (4.13), the computation of (4.14) can be replaced by an update of only its first column (denoted by $[\mathcal{P}_{i|i}^{-1}]_{\ell,1}$), that is,

$$[\mathcal{P}_{i|i}^{-1}]_{\ell,1} = \frac{1}{\sigma^2(i)} \sum_{j=0}^i f_{\ell\ell}^j f_{11}^j x(i-j-\ell+1) x(i-j), \text{ for } i \geq 0. \quad (4.15)$$

Instead of (4.15), the first column of $\mathcal{P}_{i|i}^{-1}$ can be efficiently computed using the recursion described next.

Let us first split (4.15) into the following two terms

$$[\mathcal{P}_{i|i}^{-1}]_{\ell,1} = \frac{1}{\sigma^2(i)}x(i-\ell+1)x(i) + \frac{1}{\sigma^2(i)}\sum_{j=1}^i f_{\ell\ell}^j f_{11}^j x(i-j-\ell+1)x(i-j). \quad (4.16)$$

By defining $j = k + 1$ for $0 \leq k \leq i - 1$, we get

$$\begin{aligned} [\mathcal{P}_{i|i}^{-1}]_{\ell,1} &= \frac{1}{\sigma^2(i)}x(i-\ell+1)x(i) + \frac{1}{\sigma^2(i)}\sum_{k=0}^{i-1} f_{\ell\ell}^{k+1} f_{11}^{k+1} x(i-k-\ell)x(i-k-1) \\ &= \frac{1}{\sigma^2(i)}x(i-\ell+1)x(i) + f_{\ell\ell} f_{11} \underbrace{\left[\frac{1}{\sigma^2(i)}\sum_{k=0}^{i-1} f_{\ell\ell}^k f_{11}^k x(i-k-\ell)x(i-k-1) \right]}_{=[\mathcal{P}_{i-1|i-1}^{-1}]_{\ell,1}}, \end{aligned} \quad (4.17)$$

and thus

$$[\mathcal{P}_{i|i}^{-1}]_{\ell,1} = \frac{1}{\sigma^2(i)}x(i-\ell+1)x(i) + f_{\ell\ell} f_{11} [\mathcal{P}_{i-1|i-1}^{-1}]_{\ell,1}, \text{ for } i \geq 0. \quad (4.18)$$

Since $\mathcal{P}_{i|i}^{-1}$ is symmetric, its first row is the transpose of its first column, and does not need to be evaluated again. As proved next, the other terms of $\mathcal{P}_{i|i}^{-1}$ are just a copy of the values computed in the previous iteration (and so the whole matrix can be obtained using $\mathcal{O}(M)$ operations).

The structure we want to guarantee is such that, for $1 \leq \ell \leq M-1$ and $1 \leq m \leq M-1$, the element (ℓ, m) of $\mathcal{P}_{i|i}^{-1}$ at time $i-1$ is the same as the element $(\ell+1, m+1)$ at time i . In other words, $[\mathcal{P}_{i-1|i-1}^{-1}]_{\ell,m} = [\mathcal{P}_{i|i}^{-1}]_{\ell+1,m+1}$.

In fact, by expanding the terms of the right hand side of (4.14) at time $i-1$ and supposing that $\sigma^2(i) = \sigma^2$, we obtain:

$$\begin{aligned} [\mathcal{P}_{i-1|i-1}^{-1}]_{\ell,m} &= \frac{1}{\sigma^2}[x(i-\ell)x(i-m) + \\ &\quad + f_{\ell\ell} f_{mm} x(i-\ell-1)x(i-m-1) + \\ &\quad + f_{\ell\ell}^2 f_{mm}^2 x(i-\ell-2)x(i-m-2) + \\ &\quad + \dots + f_{\ell\ell}^{i-1} f_{mm}^{i-1} x(1-\ell)x(1-m)]. \end{aligned} \quad (4.19)$$

By using relation (4.14) to compute $[\mathcal{P}_{i|i}^{-1}]_{\ell+1,m+1}$ for $\sigma^2(i) = \sigma^2$, we get:

$$\begin{aligned}
[\mathcal{P}_{i|i}^{-1}]_{\ell+1,m+1} &= \frac{1}{\sigma^2} [x(i-\ell)x(i-m) + \\
&\quad + f_{\ell+1\ell+1}f_{m+1m+1}x(i-\ell-1)x(i-m-1) + \\
&\quad + f_{\ell+1\ell+1}^2f_{m+1m+1}^2x(i-\ell-2)x(i-m-2) + \\
&\quad + \dots + f_{\ell+1\ell+1}^{i-1}f_{m+1m+1}^{i-1}x(1-\ell)x(1-m) \\
&\quad + \underbrace{f_{\ell+1\ell+1}^{i-\ell}f_{m+1m+1}^{i-\ell}}_{=0}x(-\ell)x(-m)].
\end{aligned} \tag{4.20}$$

Under the conditions (4.12) and (4.13) and considering $x(i) = 0$ for $i < 0$, equations (4.19) and (4.20) are the same for $1 \leq \ell, m \leq M-1$. Using the symmetry of $\mathcal{P}_{i|i}^{-1}$, for any instant i , the following pattern holds:

$$\mathcal{P}_{i|i}^{-1} = \begin{bmatrix} [\mathcal{P}_{i|i}^{-1}]_{1,1} & [\mathcal{P}_{i|i}^{-1}]_{2:M,1}^T \\ [\mathcal{P}_{i|i}^{-1}]_{2:M,1} & [\mathcal{P}_{i-1|i-1}^{-1}]_{(1:M-1,1:M-1)} \end{bmatrix},$$

where we use the subscript $(1 : M-1, 1 : M-1)$ to indicate the top-left $(M-1) \times (M-1)$ block matrix of $\mathcal{P}_{i|i}^{-1}$.

By using this update structure to compute $\mathcal{P}_{i|i}^{-1}$ and the DCD technique to compute α_i in (4.8), it is possible to reduce the complexity of the Kalman filter to $\mathcal{O}(M)$.

The next section shows through simulations, how close the performance of the Kalman-DCD algorithm can get to that of the standard Kalman filter.

4.3 Simulations

We use the standard Kalman filter and RLS algorithms, and their respective reduced complexity versions Kalman-DCD and RLS-DCD to track a vector $\mathbf{w}_i^o \in \mathbb{R}^7$ satisfying

$$\mathbf{w}_i^o = \mathbf{F}\mathbf{w}_{i-1}^o, \tag{4.21}$$

where \mathbf{w}_0^o is drawn from a zero mean Gaussian distribution with covariance matrix equal to the identity, $\mathbf{F} = \theta\mathbf{I}$ is a 7×7 diagonal matrix with $\theta = 0.995$. The observed signals are $d(i)$ and \mathbf{u}_i , related by (4.3), with $\sigma_v^2 = 0.01$, and \mathbf{u}_i is a tap-delay line in which $u(i)$ is generated from a first-order Auto-Regressive (AR) process with parameter 0.8, so that the covariance matrix \mathbf{R} of \mathbf{u}_i is a Toeplitz matrix with first row given by $\frac{1}{7}[1, 0.8, 0.8^2, \dots, 0.8^6]$. The algorithms are implemented with the following set of parameters:

- Standard RLS: $\lambda = 0.995$, $\mathbf{P}_0 = \mathbf{I}$.
- Standard Kalman filter: $\mathbf{F} = \theta \mathbf{I}$ with $\theta = 0.995$, $\mathbf{G}_i = \mathbf{0}$, $\mathbf{h}_i^T = \mathbf{u}_i^T$, $\sigma^2 = 0.01$, $z(i) = d(i)$ and $\mathcal{P}_{-1} = \mathbf{I}$.
- RLS-DCD and Kalman-DCD: same as for RLS and the Kalman filters, but additionally with the number of updates $N_u = 10$, step-size $H = 64$, number of bits $B = 24$ and $\mathcal{P}_{-1|-1}^{-1} = \mathbf{I}$ (for Kalman-DCD).

Under these conditions, we obtain, after running 100 simulations to compute the ensemble average Excess Mean Squared Error (EMSE) for each algorithm, the plots shown in Figure 14 where the EMSE is defined as $\mathbb{E}\{\|\mathbf{u}_i^T(\mathbf{w}_{i-1}^o - \mathbf{w}_{i-1})\|^2\}$ for the RLS and RLS-DCD algorithms, and $\mathbb{E}\{\|\mathbf{h}_i^T(\mathbf{w}_{i-1}^o - \hat{\mathbf{x}}_{i-1})\|^2\}$ for the Kalman and Kalman-DCD algorithms. As we can see in Figure 14, when \mathbf{F} is multiple of the identity matrix (a condition that satisfies (4.12) and (4.13)), the Kalman-DCD approximates closely the EMSE of the standard Kalman filter even using a small number of updates ($N_u = 10$). The performance of the RLS algorithm is reasonable, but with an EMSE saturating at around -37.5 dB, while the Kalman and Kalman-DCD EMSEs decrease to zero. Note that the performance of the Kalman-DCD algorithm is very close to that of the Kalman filter, even though we used $\epsilon = 1$ in the initial condition for $\mathcal{P}_{-1|-1}^{-1}$. Figure 15 shows a zoom of the first iterations of the algorithms.

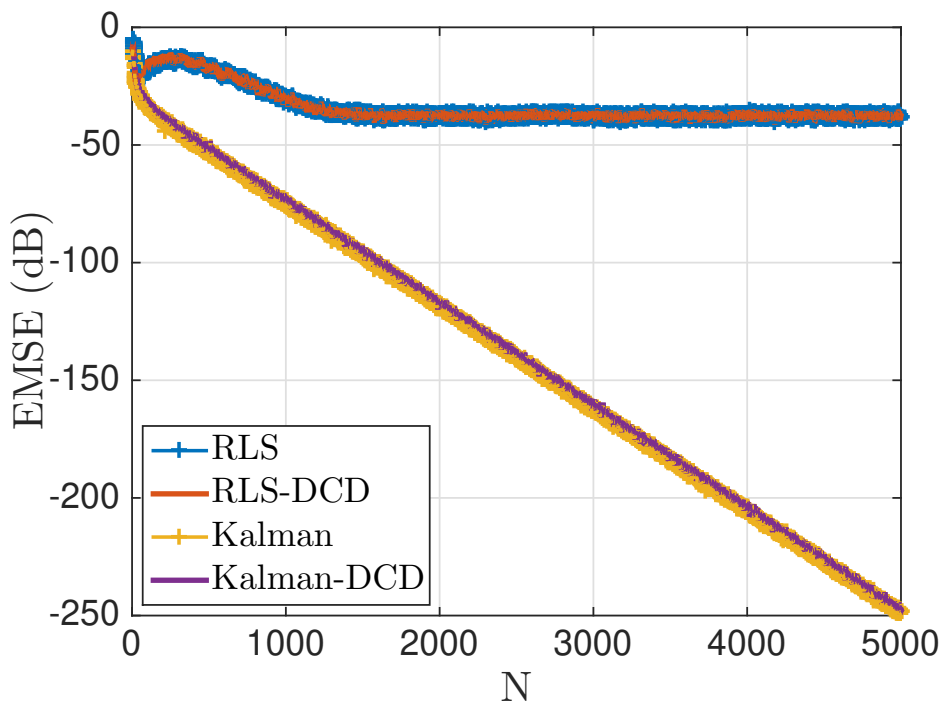


Figure 14: Simulated EMSEs curves for RLS, RLS-DCD, Kalman and Kalman-DCD when $\mathbf{F} = 0.995\mathbf{I}$.

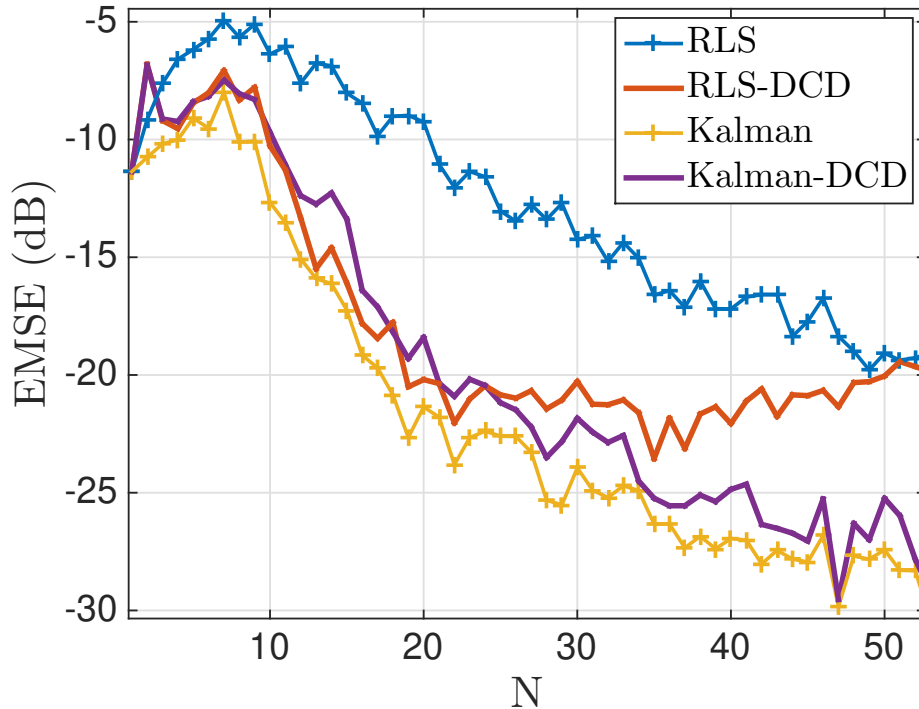


Figure 15: Zoom of the initial iterations in Figure 14.

Initially the EMSE of the Kalman-DCD algorithm is up to 5dB larger than that of the full Kalman filter (due to the approximations used in solving (4.8) by the DCD algorithm, and by using a nonzero value of ϵ for the initial condition of $\mathcal{P}_{-1|-1}^{-1}$). The algorithm however quickly converges to about 1dB of the Kalman filter.

By keeping the same set of parameters described before, and changing the matrix \mathbf{F} to another condition that also satisfies (4.12) and (4.13), in this case a diagonal matrix with elements $[1, -1, 1, -1, \dots, 1]$ (alternate sign numbers but with same module), we obtain the curves shown in Figure 16, which show that the Kalman-DCD algorithm still follows closely the performance of the standard Kalman filter, but with a smaller computational complexity. The EMSEs of RLS and RLS-DCD now saturate at -2.9 dB. A zoom of the first iterations is shown in Figure 17.

Again, the Kalman-DCD algorithm quickly converges to a close approximation of the full Kalman filter (after about 200 iterations the difference between the algorithms is negligible).

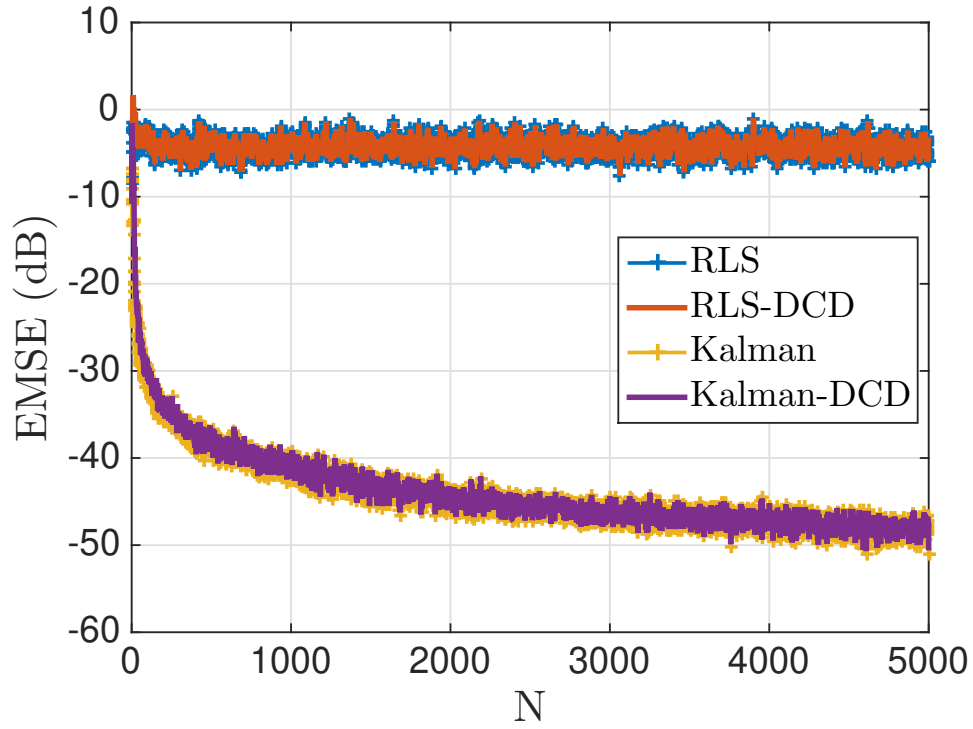


Figure 16: Simulated EMSEs curves for RLS, RLS-DCD, Kalman and Kalman-DCD when \mathbf{F} is a 7×7 diagonal matrix composed of $[1, -1, 1, -1, \dots, 1]$.

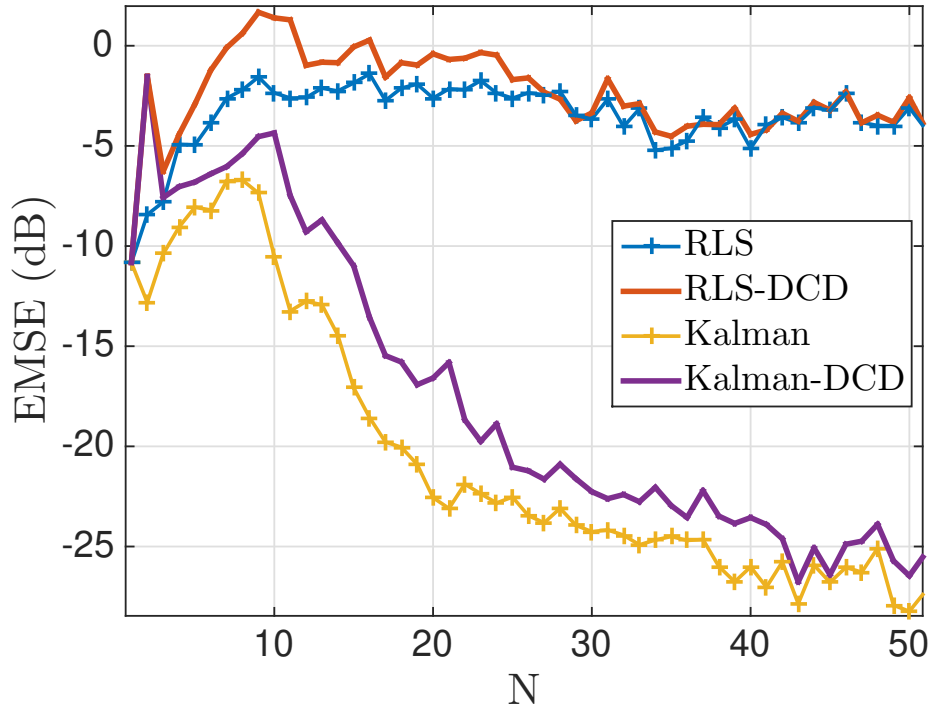


Figure 17: Zoom of the first iterations in Figure 16.

In order to see the performance of the Kalman-DCD under a nonstationary environment, we add to model (4.21) a random process noise \mathbf{q}_i with covariance matrix $\mathbf{Q} = \sigma_q^2 \mathbf{I}$.

Figure 18 shows the steady-state EMSEs estimated from the ensemble-average learning curve obtained from 100 independent runs for 30.000 iterations of the algorithms Kalman and Kalman-DCD when $10^{-10} \leq \sigma_q^2 \leq 5 \cdot 10^{-6}$. For this simulation we used $\mathbf{G} = \mathbf{I}$ and we kept the same matrices \mathbf{F} and \mathbf{R} , as well as the parameters M , σ^2 , N_u , H and B used to obtain Figure 14.

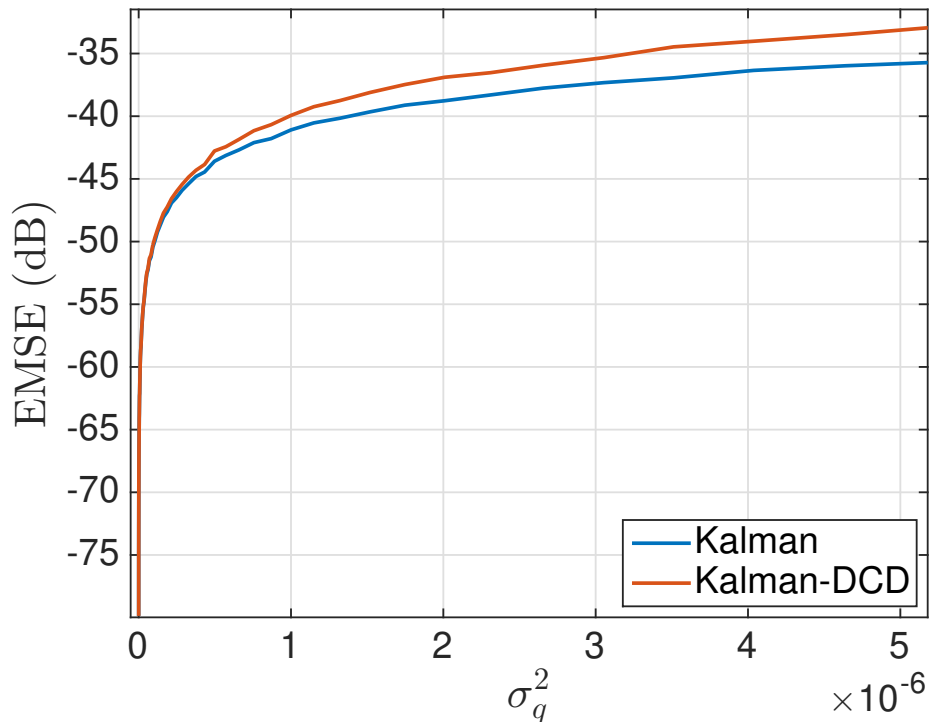


Figure 18: Simulated steady-state EMSE for Kalman and Kalman-DCD algorithms for different values of σ_q^2 .

As we can see in Figure 18, the steady-state EMSE curve obtained via Kalman-DCD is less than 1dB from the curve obtained via Kalman filter for $\sigma_q^2 < 1 \cdot 10^{-6}$. For σ_q^2 as small as 10^{-7} , the difference between the curves is negligible.

4.4 Conclusion

In this chapter we have shown that it is possible to derive low-complexity approximations to the Kalman filter by extending the RLS-DCD algorithm to approximate a larger class of Kalman models.

As it was shown, by using the DCD technique and taking advantage of the structure of the covariance matrix $\mathcal{P}_{i|i}^{-1}$, it is possible to reduce the complexity of the Kalman filter from $\mathcal{O}(M^2)$ to $\mathcal{O}(M)$ for a certain class of diagonal matrices \mathbf{F}_i . The procedure via DCD

has also shown similar behavior to the Kalman filter even for a first-order (AR) model with small process noise.

5 FUTURE RESEARCH

Adaptive filters are at the core of many signal processing applications, ranging from array beamforming, echo cancellation, channel equalization, target localization and tracking. In the case of dynamical systems with additive Gaussian noises such as navigation and control of vehicles, the Kalman filter provides optimal estimates for the unknown parameter. In this case, to choose the best algorithm for a given application, one of the important points to be considered is the algorithm's ability to track variations in the statistics of the signals of interest.

Based on this scenario, we compared in chapters 2 and 3 the steady-state performance of the convex combination of two transversal adaptive filters, namely LMS and RLS, with the optimal solution obtained via Kalman filter.

For a non-stationary environment such as the one used in chapter 2, we have shown that it is possible to achieve an EMSE less than 1dB from the KF by using a convex combination between an LMS and an RLS filters and still keep the complexity linear with the filter length (i.e $\mathcal{O}(M)$) if lattice or DCD algorithms are used.

Similar behavior was also obtained in chapter 3 for a first order AR model with finite autocovariance matrix to describe the time variation of the unknown parameter vector \mathbf{w}_i^o . As long as the parameter θ is kept greater than a minimum value, the convex combination is able to track the variations in the statistics of the signals of interest even if the unknown parameter vector has a DC component. Similarly to chapter 2, the resulting complexity of the convex combination can be kept linear with the filter length if lattice or DCD algorithms are used.

Finally in chapter 4, we derived a new DCD based Kalman filter under a model with zero process noise. As it was shown, if the state-transition matrix \mathbf{F}_i obeys some conditions, the inverse of the covariance matrix $\mathcal{P}_{i|i}$ can be efficiently updated and the resulting complexity can be kept $\mathcal{O}(M)$ if the DCD technique is used to estimate the state vector \mathbf{x}_i . For a first-order (AR) model with a small process noise, the difference between

the Kalman-DCD and the original Kalman filter is negligible.

Based on the analysis of chapters 2 to 4, we propose the following topics to be explored in future research:

- Generalize the random-walk model by using $\mathbf{w}_i^o = \mathbf{A}\mathbf{w}_{i-1}^o + \mathbf{q}_i$ for general class of matrices \mathbf{A} in the convex combination between adaptive filters;
- Test the performance of the convex combination between the Kalman-DCD and other adaptive filters to track more general models;
- Develop a Kalman-DCD procedure for nonzero process noise;
- Compare the steady-state analyses provided in chapter 3 with the energy conservation method.

REFERENCES

- 1 DINIZ, P. S. R. *Adaptive filtering: algorithms and practical implementations*. 3rd. ed. [S.l.]: Springer, 2008.
- 2 SAYED, A. H. *Adaptive filters*. [S.l.]: John Wiley & Sons, 2011.
- 3 ZAKHAROV, Y. V.; WHITE, G. P.; LIU, J. Low-complexity RLS algorithms using dichotomous coordinate descent iterations. *IEEE Transactions on Signal Processing*, IEEE, v. 56, n. 7, p. 3150–3161, 2008.
- 4 HUSØY, J. H. Adaptive filters viewed as iterative linear equation solvers. In: SPRINGER. *International Conference on Numerical Analysis and Its Applications*. [S.l.], 2004. p. 320–327.
- 5 BOSE, T.; XU, G.-F. The euclidean direction search algorithm in adaptive filtering. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, The Institute of Electronics, Information and Communication Engineers, v. 85, n. 3, p. 532–539, 2002.
- 6 DAVILA, C. E. Line search algorithms for adaptive filtering. *IEEE transactions on signal processing*, IEEE, v. 41, n. 7, p. 2490–2494, 1993.
- 7 BORAY, G. K.; SRINATH, M. D. Conjugate gradient techniques for adaptive filtering. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, IEEE, v. 39, n. 1, p. 1–10, 1992.
- 8 CHANG, P. S.; WILLSON, A. N. Analysis of conjugate gradient algorithms for adaptive filtering. *IEEE Transactions on Signal Processing*, IEEE, v. 48, n. 2, p. 409–418, 2000.
- 9 FRIEDLANDER, B. Lattice filters for adaptive processing. *Proceedings of the IEEE*, IEEE, v. 70, n. 8, p. 829–867, 1982.
- 10 PROUDLER, I.; MCWHIRTER, J.; SHEPHERD, T. Computationally efficient QR decomposition approach to least squares adaptive filtering. In: IET. *IEE Proceedings F (Radar and Signal Processing)*. [S.l.], 1991. v. 138, n. 4, p. 341–353.
- 11 SAYED, A. H.; KAILATH, T. A state-space approach to adaptive filtering. In: IEEE. *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*. [S.l.], 1993. v. 3, p. 559–562.
- 12 TSAKIRIS, M. C.; LOPES, C. G.; NASCIMENTO, V. H. An array recursive least-squares algorithm with generic nonfading regularization matrix. *IEEE Signal Processing Letters*, IEEE, v. 17, n. 12, p. 1001–1004, 2010.
- 13 NASCIMENTO, V. H.; SILVA, M. T. *Adaptive filters*. [S.l.]: Chennai: Academic Press, 2014. 619–761 p.

- 14 SILVA, M. T.; NASCIMENTO, V. H. Improving the tracking capability of adaptive filters via convex combination. *IEEE Transactions on Signal Processing*, IEEE, v. 56, n. 7, p. 3137–3149, 2008.
- 15 ARENAS-GARCIA, J.; FIGUEIRAS-VIDAL, A. R.; SAYED, A. H. Mean-square performance of a convex combination of two adaptive filters. *IEEE transactions on signal processing*, IEEE, v. 54, n. 3, p. 1078–1090, 2006.
- 16 ARENAS-GARCÍA, J.; FIGUEIRAS-VIDAL, A. R. Adaptive combination of proportionate filters for sparse echo cancellation. *IEEE Transactions on Audio, Speech, and Language Processing*, IEEE, v. 17, n. 6, p. 1087–1098, 2009.
- 17 TRUMP, T. A combination of two NLMS filters in an adaptive line enhancer. In: IEEE. *Digital Signal Processing (DSP), 2011 17th International Conference on*. [S.l.], 2011. p. 1–6.
- 18 COMMINIELLO, D. et al. Combined adaptive beamforming schemes for nonstationary interfering noise reduction. *Signal Processing*, Elsevier, v. 93, n. 12, p. 3306–3318, 2013.
- 19 GEORGE, N. V.; GONZALEZ, A. Convex combination of nonlinear adaptive filters for active noise control. *Applied Acoustics*, Elsevier, v. 76, p. 157–161, 2014.
- 20 LÁZARO-GREDILLA, M. et al. Adaptively biasing the weights of adaptive filters. *IEEE Transactions on Signal Processing*, IEEE, v. 58, n. 7, p. 3890–3895, 2010.
- 21 AZPICUETA-RUIZ, L. A.; FIGUEIRAS-VIDAL, A. R.; ARENAS-GARCIA, J. A normalized adaptation scheme for the convex combination of two adaptive filters. In: IEEE. *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. [S.l.], 2008. p. 3301–3304.
- 22 NASCIMENTO, V. H. et al. On the tracking performance of combinations of least mean squares and recursive least squares adaptive filters. In: IEEE. *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. [S.l.], 2010. p. 3710–3713.
- 23 CANDIDO, R.; SILVA, M. T.; NASCIMENTO, V. H. Affine combinations of adaptive filters. In: IEEE. *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*. [S.l.], 2008. p. 236–240.
- 24 KOVVALI, N.; BANAVAR, M.; SPANIAS, A. An introduction to Kalman filtering with Matlab examples. *Synthesis Lectures on Signal Processing*, Morgan & Claypool Publishers, v. 6, n. 2, p. 1–81, 2013.
- 25 ANDERSON, B. D.; MOORE, J. B. Optimal filtering. *Englewood Cliffs*, v. 21, p. 22–95, 1979.
- 26 BLACKMAN, S. S. Multiple-target tracking with radar applications. *Dedham, MA, Artech House, Inc., 1986, 463 p.*, 1986.
- 27 GREWAL, M. S.; HENDERSON, V. D.; MIYASAKO, R. S. Application of Kalman filtering to the calibration and alignment of inertial navigation systems. *IEEE Transactions on automatic control*, IEEE, v. 36, n. 1, p. 3–13, 1991.

- 28 ZARCHAN, P. *Progress In Astronautics and Aeronautics: Fundamentals of Kalman Filtering: A Practical Approach*. [S.l.]: Aiaa, 2005. v. 208.
- 29 NIEDZWIECKI, M. Identification of nonstationary stochastic systems using parallel estimation schemes. *IEEE Transactions on Automatic Control*, IEEE, v. 35, n. 3, p. 329–334, 1990.
- 30 NIEDZWIECKI, M. Multiple-model approach to finite memory adaptive filtering. *IEEE Transactions on signal processing*, IEEE, v. 40, n. 2, p. 470–473, 1992.
- 31 MONTELLA, C. The Kalman filter and related algorithms: A literature review. *Research Gate*, 2011.
- 32 ARENAS-GARCIA, J. et al. Combinations of adaptive filters: performance and convergence properties. *IEEE Signal Processing Magazine*, IEEE, v. 33, n. 1, p. 120–140, 2016.
- 33 CLASER, R.; NASCIMENTO, V. H. Low-complexity approximation to the Kalman filter using convex combinations of adaptive filters from different families. In: IEEE. *Signal Processing Conference (EUSIPCO), 2017 25th European*. [S.l.], 2017. p. 2630–2633.
- 34 BARCZYK, M.; LYNCH, A. F. Invariant observer design for a helicopter UAV aided inertial navigation system. *IEEE Transactions on Control Systems Technology*, IEEE, v. 21, n. 3, p. 791–806, 2012.
- 35 LIM, C. H.; LIM, T. S.; KOO, V. C. Design and development of a real-time GPS-aided SINU system. *International Journal of Advanced Robotic Systems*, SAGE Publications Sage UK: London, England, v. 9, n. 5, p. 194, 2012.
- 36 SALTI, S.; LANZA, A.; STEFANO, L. D. Synergistic change detection and tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, IEEE, v. 25, n. 4, p. 609–622, 2014.
- 37 BRESSON, G. et al. Real-time monocular slam with low memory requirements. *IEEE Transactions on Intelligent Transportation Systems*, IEEE, v. 16, n. 4, p. 1827–1839, 2015.
- 38 GREWAL, M. S.; ANDREWS, A. P. *Kalman filtering: theory and practice using MATLAB*. [S.l.]: Wiley, 2011.
- 39 KAILATH, T.; SAYED, A. H.; HASSIBI, B. *Linear Estimation*. NJ: Prentice Hall, 2000.
- 40 FABRY, J. et al. Active noise control with reduced-complexity Kalman filter. In: IEEE. *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*. [S.l.], 2018. p. 166–170.
- 41 DIETZEN, T. et al. Low-complexity Kalman filter for multi-channel linear-prediction-based blind speech dereverberation. In: IEEE. *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. [S.l.], 2017. p. 284–288.
- 42 SAYED, A. H. *Fundamentals of adaptive filtering*. [S.l.]: John Wiley & Sons, 2003.

- 43 FABRY, J.; KÜHL, S.; JAX, P. On the steady state performance of the Kalman filter applied to acoustical systems. *IEEE Signal Processing Letters*, v. 27, p. 1854–1858, 2020. ISSN 1558-2361.
- 44 MACCHI, O.; TURKI, M. The non stationarity degree: Can an adaptive filter be worse than no processing? *IFAC Proceedings Volumes*, Elsevier, v. 25, n. 14, p. 517–522, 1992.
- 45 SOLO, V.; KONG, X. *Adaptive Signal Processing Algorithms*. Englewood Cliffs, NJ: Prentice Hall, 1995.
- 46 HAYKIN, S. S. *Adaptive filter theory*. [S.l.]: Pearson Education India, 2008.
- 47 NASCIMENTO, V. H.; SILVA, M. T. M. Adaptive filters. In: CHELLAPPA, R.; THEODORIDIS, S. (Ed.). *Academic Press Library in Signal Processing*. [S.l.]: Chennai: Academic Press, 2014. v. 1, Signal Processing Theory and Machine Learning, p. 619–761. ISBN 978-0-12-396502-8.
- 48 LU, L.-Z.; LIN, W.-W. An iterative algorithm for the solution of the discrete-time algebraic Riccati equation. *Linear Algebra and Its Applications*, Elsevier, v. 188, p. 465–488, 1993.
- 49 FENG, Y.; ANDERSON, B. D.; CHEN, W. Solving discrete algebraic Riccati equations: A new recursive method. In: IEEE. *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. [S.l.], 2009. p. 1720–1724.
- 50 EWEDA, E. Comparison of RLS, LMS, and sign algorithms for tracking randomly time-varying channels. *IEEE Transactions on Signal Processing*, IEEE, v. 42, n. 11, p. 2937–2944, 1994.
- 51 KUSHNER, H. J.; YIN, G. G. Applications to learning, state dependent noise, and queueing. In: *Stochastic Approximation Algorithms and Applications*. [S.l.]: Springer, 1997. p. 25–46.
- 52 KALMAN, R. E. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, American Society of Mechanical Engineers, v. 82, n. 1, p. 35–45, 1960.
- 53 RISTIC, B.; ARULAMPALAM, S.; GORDON, N. *Beyond the Kalman filter: Particle filters for tracking applications*. [S.l.]: Artech house, 2003.
- 54 MANOLIU, M.; TOMPAIDIS, S. Energy futures prices: term structure models with Kalman filter estimation. *Applied mathematical finance*, Taylor & Francis, v. 9, n. 1, p. 21–43, 2002.
- 55 SAYED, A. H.; KAILATH, T. A state-space approach to adaptive RLS filtering. *IEEE Signal Process. Mag.*, v. 11, n. 3, p. 18–60, jul. 1994.
- 56 SLOCK, D. T. M.; KAILATH, T. Numerically stable fast transversal filters for recursive least squares adaptive filtering. *IEEE Signal Process. Lett.*, v. 39, n. 1, p. 92–114, 1991.

- 57 MIRANDA, M.; GERKEN, M.; SILVA, M. D. Efficient implementation of error-feedback LSL algorithm. *Electronics Letters*, v. 35, n. 16, p. 1308–1309, 1999.
- 58 SAYED, A. H. A framework for state-space estimation with uncertain models. *IEEE Transactions on Automatic Control*, IEEE, v. 46, n. 7, p. 998–1013, 2001.

APPENDIX A

A.1 Diagonal elements of the covariance matrix $\bar{\mathbf{S}}_\infty^{(22)}$ for RLS

Similar to the LMS filter, we begin the algebraic development of the covariance matrix $\bar{\mathbf{S}}_\infty^{(22)} = \mathbf{U}^T \mathbf{S}_\infty^{(22)} \mathbf{U}$ for the RLS filter by computing the covariance matrix $\mathbf{S}_\infty^{(22)}$. Replacing equations (3.20) and (3.23) in (3.17) and considering: $\ell = m = 2$, $\bar{\mathbf{M}}^{(2)} = \bar{\mathbf{P}} = (1 - \lambda) \mathbf{R}^{-1}$, $\rho^{(2)} = 1$ and assuming that the filter is operating in steady-state, i.e. $i \rightarrow \infty$, we get the following recursion for $\mathbf{S}_\infty^{(22)}$:

$$\begin{aligned} \mathbf{S}_\infty^{(22)} &\approx \mathbf{S}_\infty^{(22)} - 2\lambda(1 - \lambda)\mathbf{S}_\infty^{(22)} + (1 - \lambda)^2 \text{Tr}\{\mathbf{R}\mathbf{S}_\infty^{(22)}\}\mathbf{R}^{-1} \\ &\quad + \sigma_v^2(1 - \lambda)^2 \mathbf{R}^{-1} + \frac{2(1 - \theta)(1 - \lambda)}{1 - \theta\lambda} \mathbf{Q}. \end{aligned} \quad (\text{A.1})$$

A recursion for $\bar{\mathbf{S}}_\infty^{(22)}$ can be obtained by multiplying (A.1) from the left by \mathbf{U}^T and from the right by \mathbf{U} . Defining the rotated matrix $\bar{\mathbf{Q}} = \mathbf{U}^T \mathbf{Q} \mathbf{U}$ and recalling that $\mathbf{R} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T$ and $\mathbf{I} = \mathbf{U} \mathbf{U}^T = \mathbf{U}^T \mathbf{U}$, we get after simplifications

$$\begin{aligned} \bar{\mathbf{S}}_\infty^{(22)} &\approx \bar{\mathbf{S}}_\infty^{(22)} - 2\lambda(1 - \lambda)\bar{\mathbf{S}}_\infty^{(22)} + (1 - \lambda)^2 \text{Tr}\{\mathbf{\Lambda}\bar{\mathbf{S}}_\infty^{(22)}\}\mathbf{\Lambda}^{-1} \\ &\quad + \sigma_v^2(1 - \lambda)^2 \mathbf{\Lambda}^{-1} + \frac{2(1 - \theta)(1 - \lambda)}{1 - \theta\lambda} \bar{\mathbf{Q}}. \end{aligned} \quad (\text{A.2})$$

Using the rotated matrix $\bar{\mathbf{S}}_\infty^{(22)}$, the steady-state EMSE can be computed as $\text{Tr}\{\mathbf{\Lambda}\bar{\mathbf{S}}_\infty^{(22)}\}$ and so, it depends only on the diagonal entries of $\bar{\mathbf{S}}_\infty^{(22)}$. We can work therefore only with these diagonal entries and define the vectors

$$\bar{\mathbf{s}}_\infty^{(22)} = \text{diag}\{\bar{\mathbf{S}}_\infty^{(22)}\} \text{ and } \boldsymbol{\ell} = \text{diag}\{\mathbf{\Lambda}\}, \quad (\text{A.3})$$

where $\text{diag}\{\mathbf{A}\}$ represents a column vector with the diagonal elements of \mathbf{A} .

By applying the diagonal operator to both sides of equation (A.2) and using the

definitions from (A.3), we obtain

$$\begin{aligned}\bar{\mathbf{s}}_{\infty}^{(22)} &\approx \bar{\mathbf{s}}_{\infty}^{(22)} - 2\lambda(1-\lambda)\bar{\mathbf{s}}_{\infty}^{(22)} + (1-\lambda)^2\boldsymbol{\ell}^{-1}\boldsymbol{\ell}^T\bar{\mathbf{s}}_{\infty}^{(22)} \\ &+ \sigma_v^2(1-\lambda)^2\boldsymbol{\ell}^{-1} + \frac{2(1-\theta)(1-\lambda)}{1-\theta\lambda}\text{diag}\{\bar{\mathbf{Q}}\},\end{aligned}\quad (\text{A.4})$$

where $\boldsymbol{\ell}^{-1}$ denotes the element-wise inversion.

Multiplying both sides of (A.4) by $(1-\lambda)^{-1}$ and simplifying it leads to equation (3.39), repeated below

$$\bar{\mathbf{s}}_{\infty}^{(22)} \approx [2\lambda\mathbf{I} - (1-\lambda)\boldsymbol{\ell}^{-1}\boldsymbol{\ell}^T]^{-1} \left\{ \sigma_v^2(1-\lambda)\boldsymbol{\ell}^{-1} + \frac{2(1-\theta)}{1-\theta\lambda}\text{diag}\{\bar{\mathbf{Q}}\} \right\}.$$

A.2 Diagonal elements of the cross-covariance matrix $\bar{\mathbf{S}}_{\infty}^{(12)}$ for combination between LMS and RLS

To compute the transformed cross-covariance matrix $\bar{\mathbf{S}}_{\infty}^{(12)} = \mathbf{U}^T\mathbf{S}_{\infty}^{(12)}\mathbf{U}$ for the combination between LMS and RLS filters, we followed similar algebraic development used in the previous section A.1. Replacing equations (3.20) and (3.23) in (3.17) and considering: LMS as $\ell = 1$ and the RLS as $m = 2$, with the corresponding parameters $\rho^{(1)} = \mu$, $\bar{\mathbf{M}}^{(1)} = \mathbf{I}$, $\rho^{(2)} = 1$ and $\bar{\mathbf{M}}^{(2)} = \bar{\mathbf{P}} = (1-\lambda)\mathbf{R}^{-1}$, and assuming that the filter is operating in steady-state, i.e. $i \rightarrow \infty$, we get the following recursion for $\mathbf{S}_{\infty}^{(12)}$:

$$\begin{aligned}\mathbf{S}_{\infty}^{(12)} &\approx \mathbf{S}_{\infty}^{(12)} - (1-\lambda)\mathbf{S}_{\infty}^{(12)} - \mu\mathbf{R}\mathbf{S}_{\infty}^{(12)} \\ &+ \mu(1-\lambda)\text{Tr}\{\mathbf{R}\mathbf{S}_{\infty}^{(12)}\}\mathbf{I} + 2\mu(1-\lambda)\mathbf{R}\mathbf{S}_{\infty}^{(12)} + \mu(1-\lambda)\sigma_v^2\mathbf{I} \\ &+ (1-\theta)^2[\mu\mathbf{R} - \mathbf{I}][\mu\mathbf{R} - \mathbf{I}]^{-1}\mathbf{Q} \\ &+ \frac{(1-\theta)(2-\lambda-\theta\lambda)}{(1-\theta\lambda)}\mathbf{Q}.\end{aligned}\quad (\text{A.5})$$

To simplify (A.5) and get an easier expression to deal with, we multiply and divide by θ the third line of (A.5) and reorganize the terms to obtain

$$\begin{aligned}\mathbf{S}_{\infty}^{(12)} &\approx \mathbf{S}_{\infty}^{(12)} - (1-\lambda)\mathbf{S}_{\infty}^{(12)} - \mu\mathbf{R}\mathbf{S}_{\infty}^{(12)} \\ &+ \mu(1-\lambda)\text{Tr}\{\mathbf{R}\mathbf{S}_{\infty}^{(12)}\}\mathbf{I} + 2\mu(1-\lambda)\mathbf{R}\mathbf{S}_{\infty}^{(12)} + \mu(1-\lambda)\sigma_v^2\mathbf{I} \\ &+ \theta^{-1}(1-\theta)^2[\mu\theta\mathbf{R} - \theta\mathbf{I}][\mathbf{I} + \mu\theta\mathbf{R} - \theta\mathbf{I}]^{-1}\mathbf{Q} \\ &+ \frac{(1-\theta)(2-\lambda-\theta\lambda)}{(1-\theta\lambda)}\mathbf{Q}.\end{aligned}\quad (\text{A.6})$$

By using the push-through identity property $\mathbf{A}(\mathbf{I}+\mathbf{A})^{-1} = (\mathbf{I}+\mathbf{A})^{-1}\mathbf{A} = \mathbf{I} - (\mathbf{I}+\mathbf{A})^{-1}$,

where $\mathbf{A} = \mu\theta\mathbf{R} - \theta\mathbf{I}$, after some algebra we obtain the following simplified version of equation (A.6)

$$\begin{aligned}\mathbf{S}_\infty^{(12)} &\approx \mathbf{S}_\infty^{(12)} - (1 - \lambda)\mathbf{S}_\infty^{(12)} - \mu\mathbf{R}\mathbf{S}_\infty^{(12)} \\ &+ \mu(1 - \lambda)\text{Tr}\{\mathbf{R}\mathbf{S}_\infty^{(12)}\}\mathbf{I} + 2\mu(1 - \lambda)\mathbf{R}\mathbf{S}_\infty^{(12)} + \mu(1 - \lambda)\sigma_v^2\mathbf{I} \\ &- \theta^{-1}(1 - \theta)^2 [(1 - \theta)\mathbf{I} + \mu\theta\mathbf{R}]^{-1} \mathbf{Q} \\ &+ \frac{(1 - \theta)(1 - 2\theta\lambda - \theta)}{\theta(1 - \theta\lambda)}\mathbf{Q}.\end{aligned}\tag{A.7}$$

By multiplying (A.7) from the left by \mathbf{U}^T and from the right by \mathbf{U} and using the same rotated matrix $\bar{\mathbf{Q}}$ and diagonal matrix $\mathbf{\Lambda}$, we get the following recursion for $\bar{\mathbf{S}}_\infty^{(12)}$ after simplifications

$$\begin{aligned}\bar{\mathbf{S}}_\infty^{(12)} &\approx \bar{\mathbf{S}}_\infty^{(12)} - (1 - \lambda)\bar{\mathbf{S}}_\infty^{(12)} - \mu\mathbf{\Lambda}\bar{\mathbf{S}}_\infty^{(12)} \\ &+ \mu(1 - \lambda)\text{Tr}\{\mathbf{\Lambda}\bar{\mathbf{S}}_\infty^{(12)}\}\mathbf{I} + 2\mu(1 - \lambda)\mathbf{\Lambda}\bar{\mathbf{S}}_\infty^{(12)} + \mu(1 - \lambda)\sigma_v^2\mathbf{I} \\ &- \theta^{-1}(1 - \theta)^2 [(1 - \theta)\mathbf{I} + \mu\theta\mathbf{\Lambda}]^{-1} \bar{\mathbf{Q}} \\ &+ \frac{(1 - \theta)(1 - 2\theta\lambda - \theta)}{\theta(1 - \theta\lambda)}\bar{\mathbf{Q}}.\end{aligned}\tag{A.8}$$

Using the rotated matrix $\bar{\mathbf{S}}_\infty^{(12)}$, the steady-state EMSE can be computed as $\text{Tr}\{\mathbf{\Lambda}\bar{\mathbf{S}}_\infty^{(12)}\}$ and so, it depends only on the diagonal entries of $\bar{\mathbf{S}}_\infty^{(12)}$. We can work therefore only with these diagonal entries and define the vector

$$\bar{\mathbf{s}}_\infty^{(12)} = \text{diag}\{\bar{\mathbf{S}}_\infty^{(12)}\}.\tag{A.9}$$

By applying the diagonal operator to both sides of equation (A.8) and using the definition from (A.9), we obtain

$$\begin{aligned}\bar{\mathbf{s}}_\infty^{(12)} &\approx \bar{\mathbf{s}}_\infty^{(12)} - (1 - \lambda)\bar{\mathbf{s}}_\infty^{(12)} - \mu\mathbf{\Lambda}\bar{\mathbf{s}}_\infty^{(12)} \\ &+ \mu(1 - \lambda)\mathbb{1}\ell^T\bar{\mathbf{s}}_\infty^{(12)} + 2\mu(1 - \lambda)\mathbf{\Lambda}\bar{\mathbf{s}}_\infty^{(12)} + \mu(1 - \lambda)\sigma_v^2\mathbb{1} \\ &- \theta^{-1}(1 - \theta)^2 [(1 - \theta)\mathbf{I} + \mu\theta\mathbf{\Lambda}]^{-1} \text{diag}\{\bar{\mathbf{Q}}\} \\ &+ \frac{(1 - \theta)(1 - 2\theta\lambda - \theta)}{\theta(1 - \theta\lambda)}\text{diag}\{\bar{\mathbf{Q}}\},\end{aligned}\tag{A.10}$$

where $\mathbb{1} = [1 \ 1 \ \dots \ 1]^T$.

Reorganizing the terms and simplifying equation (A.10) it leads to equation (3.46),

repeated below

$$\begin{aligned} \bar{\mathbf{s}}_{\infty}^{(12)} = & \left\{ (1 - \lambda)\mathbf{I} + \mu\mathbf{\Lambda} - \mu(1 - \lambda) (\mathbb{1}\ell^T + 2\mathbf{\Lambda}) \right\}^{-1} \\ & \times \left\{ \mu(1 - \lambda)\sigma_v^2\mathbb{1} - \frac{(1 - \theta)^2}{\theta} [(1 - \theta)\mathbf{I} + \mu\theta\mathbf{\Lambda}]^{-1} \text{diag}\{\bar{\mathbf{Q}}\} \right. \\ & \left. + \frac{(1 - \theta)(1 - 2\theta\lambda + \theta)}{\theta(1 - \theta\lambda)} \text{diag}\{\bar{\mathbf{Q}}\} \right\}. \end{aligned}$$