MANOLIS TSAKIRIS

ON THE REGULARIZATION OF THE RECURSIVE LEAST-SQUARES ALGORITHM

São Paulo 2010

MANOLIS TSAKIRIS

ON THE REGULARIZATION OF THE RECURSIVE LEAST-SQUARES ALGORITHM

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Engenharia Elétrica.

Área de Concentração: Sistemas Eletrônicos

Orientadores: Prof. Dr. Cássio Guimarães Lopes

São Paulo 2010

FICHA CATALOGRÁFICA

Tsakiris, Manolis

On the Regularization of the Recursive Least-Squares Algorithm. São Paulo, 2010.

54 p.

Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1. Filtros Elétricos Adaptativos. 2. Mínimos Quadrados. 3. Robustez. I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos. II. t.

I dedicate this dissertation to all my family and friends.

ACKNOWLEDGMENTS

I am very thankful to my advisor, Prof. Cássio Guimarães Lopes, for all his help and support, which made this work possible.

I wish to thank Prof. Vítor Nascimento, Prof. Paulo Sérgio da Silva and Prof. Felipe Pait for their inspiring lectures and for their help during the process.

For all their suport and articulations, I am indebted to Prof. Luiz Baccalá, to Prof. José Roberto Piqueira and to Prof. Miguel Ramirez.

I am also thankful to the Comissão de Pós-Graduação, as well as the office staff, namely Cláudia Marcelino, Marcelo and Simone for the uncountable times we needed and obtained prompt support.

I could not forget to thank my friend Matheus Moraes for being one of the brightest persons I have ever met. I dedicate this thesis to him.

Last, but not least, I thank FAPESP for the financial support.

ABSTRACT

This thesis is concerned with the issue of the regularization of the Recursive Least-Squares (RLS) algorithm. In the first part of the thesis, a novel regularized exponentially weighted array RLS algorithm is developed, which circumvents the problem of fading regularization that is inherent to the standard regularized exponentially weighted RLS formulation, while allowing the employment of generic time-varying regularization matrices. The standard equations are directly perturbed via a chosen regularization matrix; then the resulting recursions are extended to the array form. The price paid is an increase in computational complexity, which becomes cubic. The superiority of the algorithm with respect to alternative algorithms is demonstrated via simulations in the context of adaptive beamforming, in which low filter orders are employed, so that complexity is not an issue. In the second part of the thesis, an alternative criterion is motivated and proposed for the dynamical regulation of regularization in the context of the standard RLS algorithm. The regularization is implicitely achieved via dithering of the input signal. The proposed criterion is of general applicability and aims at achieving a balance between the accuracy of the numerical solution of a perturbed linear system of equations and its distance from the analytical solution of the original system, for a given computational precision. Simulations show that the proposed criterion can be effectively used for the compensation of large condition numbers, small finite precisions and uncessary large values of the regularization.

RESUMO

Esta tese trata da regularização do algoritmo dos mínimos-quadrados recursivo (Recursive Least-Squares - RLS). Na primeira parte do trabalho, um novo algoritmo array com matriz de regularização genérica e com ponderação dos dados exponencialmente decrescente no tempo é apresentado. O algoritmo é regularizado via perturbação direta da inversa da matriz de autocorrelação (P_i) por uma matriz genérica. Posteriormente, as equações recursivas são colocadas na forma array através de transformações unitárias. O preço a ser pago é o aumento na complexidade computacional, que passa a ser de ordem cúbica. A robustez do algoritmo resultante é demonstrada via simulações quando comparado com algoritmos alternativos existentes na literatura no contexto de beamforming adaptativo, no qual geralmente filtros com ordem pequena são empregados, e complexidade computacional deixa de ser fator relevante. Na segunda parte do trabalho, um critério alternativo é motivado e proposto para ajuste dinâmico da regularização do algoritmo RLS convencional. A regularização é implementada pela adição de ruído branco no sinal de entrada (dithering), cuja variância é controlada por um algoritmo simples que explora o critério proposto. O novo critério pode ser aplicado a diversas situações; procura-se alcançar um balanço entre a precisão numérica da solução de um sistema linear de equações perturbado e sua distância da solução do sistema original não-perturbado, para uma dada precisão. As simulações mostram que tal critério pode ser efetivamente empregado para compensação de números de condicionamento (CN) elevados, baixa precisão numérica, bem como valores de regularização excessivamente elevados.

CONTENTS

List of Figures			1
1	Intr	oduction	2
	1.1	A brief note on adaptive filters	2
	1.2	Notation	4
2	Reg	gularization in the Recursive Least-Squares Algorithm	6
	2.1	The Least-Squares Criterion	6
	2.2	The RLS Algorithm	9
	2.3	The Exponentially-Weighted RLS Algorithm	10
	2.4	Regularization Issues in RLS	12
	2.5	Existing Approaches	14
	2.6	The Need for Extensions	15
	2.7	The Contributions of the Thesis	17
3	An	Array RLS Algorithm with Generic Regularization Matrix	18
	3.1	Introduction	18
	3.2	Elements of Array-Forms Theory	19
	3.3	Square-Root RLS	21
	3.4	Perturbing the Standard Square-Root RLS	23

	3.5	Completing the Recursion	25
	3.6	Proposed Algorithm	27
	3.7	A Comment on the Computational Complexity	28
	3.8	Simulations	28
4	An	Alternative Criterion to Regularize RLS Problems	34
	4.1	Introduction	34
	4.2	The Image Function	35
	4.3	Contrived Example	38
	4.4	Image Function Analysis	39
	4.5	Application to RLS	41
	4.6	Proposed Algorithm	43
	4.7	Results	44
5	Cor	clusions	50
R	References		

LIST OF FIGURES

1	Example 1: SINR $@$ B = 32 bits	30
2	Example 1: CN $@$ B = 32 bits	31
3	Example 2: SINR $@$ B = 13 bits	32
4	Example 2: CN $@$ B = 13 bits	33
5	The J function for an M ${\bf x}$ M system solved @ B bits $\ . \ . \ .$.	36
6	Adaptive prediction of a sinusoid: Varying precision	46
7	Adaptive prediction of a sinusoid: Varying CN	47
8	Adaptive prediction of a sinusoid: Large perturbations $\ldots \ldots$	48
9	The IRLS algorithm handling speech signals @ B = 12 bits	49

1 INTRODUCTION

1.1 A brief note on adaptive filters

Adaptive filters are used in digital signal processing and control applications, when the exact filtering specifications are unknown prior to operation-time or they are time-varying due to the nature of the application. As an example, in mobile communications the communication channels are highly non-stationary since they depend on the location of the mobile users and are determined by the given physical and electromagnetic environment. Consequently, a system that can adapt to the variations of the channel is required at the receiver, in order to achieve acceptable levels of the Bit Error Rate (BER). The applications where adaptive filters are employed today span the entire range of modern technology with some of the most representative being channel equalization, beamforming, target tracking, echo cancellation, speech enhancement, active noise control and cardiac/brain signal processing.

Adaptive filters are discrete-time filters, whose coefficients are allowed to vary with time, so that some time-variant objective is achieved (e.g. tracking of a fastfading mobile communication channel). The variation of the filter coefficients is performed by an adaptive algorithm, which is designed to minimize some cost function. There exists a large variety of different cost functions, ranging from relatively simple to highly sophisticated ones. Different cost functions correspond to different adaptive algorithms and hence to different adaptive filters, with varying degrees of capabilities. An adaptive filter can have a Finite Impulse Response (FIR) or an Infinite Impulse Response (IIR) and can be implemented in various structures similar to the structures appearing in the implementation of the standard time-invariant filters, e.g. transversal or lattice structures. However, adaptive filters are much more difficult systems to analyze than time-invariant filters, since by design their coefficients are non-linear, time-variant functions of the generally stochastic signals pertaining to their operation. Finally, an adaptive filter can be used in various configurations in an overall system, giving rise to different functions, with most popular configurations being those of system identification, system inversion and signal prediction.

The appearence of the area of adaptive filtering can be traced back to 1950; however it was not until the landmark work of Widrow and Hoff in 1960 [1] and the birth of the famous Least Mean Square (LMS) algorithm, that adaptive filtering was established as a distinct area of scientific research and engineering application. Since then, a large amount of literature has been developed (see for example [2], [3], [4] and [5]) and with the aid of the power of modern computing systems a significant part of it has been incorporated into the industrial and technological design. Today, adaptive filtering is considered a mature research area, however still exhibiting important and challenging open issues to be addressed, including the analysis and further insight into the behaviour of the various adaptive schemes and the development of fast adaptive algorithms, which are robust to noise and numerical round-off errors, when implemented in finite precision.

1.2 Notation

The notation adopted in this thesis is as follows:

- Matrices are denoted by upper-case, italic characters, e.g. R.
- Vectors are denoted by lower-case, italic characters, e.g. w.
- Scalar quantities can be represented either by upper-case or lower-case italic characters, and their distinction from matrices or vectors will be clear from the context.
- The normalized, discrete-time index is denoted by *i*.
- Time-varying vectors and matrices are denoted by placing *i* as a subscript,
 e.g. w_i, R_i.
- The indexing of the entries of a vector starts from zero.
- Discrete-time scalar functions (signals) are denoted by lower-case, italic characters followed by *i* included in parentheses, e.g. u(i), d(i). The only exception is when denoting the entries of a time-varying vector, e.g. w_i(0) is the first entry of the vector w_i.
- All vectors are assumed to be column-vectors, with one exception: this is the so-called *regressor vector*, which contains the *M* most recent samples of the input process of a filter with *M* coefficients. This convention avoids the use of the transpose operator, when representing the convolution sum of a filter as the inner product of the regressor vector and the impulse response vector.
- The conjugate-traspose of a matrix R is denoted as R^* .

The definition of a quantity is explicitly denoted by using the symbol '[△].
 The symbol '=' denotes that the quantity on its left can be proved to be equal to the quantity on its right.

2 REGULARIZATION IN THE RECURSIVE LEAST-SQUARES ALGORITHM

2.1 The Least-Squares Criterion

The problem that a conventional adaptive filter aims at solving, can be described as follows.

Consider a given complex data collection $\{u_j, d(j)\}_{j=0}^i$, where u_j is the $1 \times M$ regressor

$$u_j \stackrel{\Delta}{=} [u(j), u(j-1), \cdots, u(j-1+M)]$$
 (2.1)

at time instant j corresponding to the signal u and d(j) is the j^{th} sample of the signal d. Then an adaptive algorithm aims at determining at iteration i a linear transformation

$$w_i : \mathbb{C}^{1 \times M} \to \mathbb{C} \tag{2.2}$$

that maps u_i to a scalar $\hat{d}(i) \stackrel{\Delta}{=} u_i w_i$, such that $\hat{d}(i)$ is an estimate of d(i). The determination of w_i is performed via the minimization of some criterion, which can usually be thought of as some function of the error signal $e \stackrel{\Delta}{=} d - \hat{d}$. As an example, the celebrated Least-Mean-Squares (LMS) algorithm [1], employs the criterion

$$J_{LMS}(i) \stackrel{\Delta}{=} |e(i)|^2 \tag{2.3}$$

which is an instantaneous approximation of the well-known Mean-Square-Error

(MSE)

$$J_{MSE} \stackrel{\Delta}{=} E \left| e(i) \right|^2 \tag{2.4}$$

where E denotes the mathematical expectation operator.

On the other hand, the Recursive-Least-Squares (RLS) algorithm is based on the following (deterministic) least squares criterion

$$J_{LS}(i) \stackrel{\Delta}{=} \sum_{j=0}^{i} |d(j) - u_j w|^2$$
(2.5)

from which the algorithm bears its name. This type of criterion, consisting of the sum of squared errors, was firstly used by Carl Friedrich Gauss in 1801 in his successful attempt to relocate the asteroid Ceres, whose orbit the astronomers of the time had lost track of [6]. Ten years later, Gauss published his famous least-squares theory, whose modern geometric interpretation and linear algebraic formulation consist one of the most important chapters in quantitative mathematics and the basis for many theories and applications.

Defining the $(i+1) \times M$ matrix H_i and the $(i+1) \times 1$ vector z_i , respectively, as

$$H_i \stackrel{\Delta}{=} col \{u_i, u_{i-1}, \cdots, u_0\}$$
(2.6)

$$z_i \stackrel{\Delta}{=} col \left\{ d(i), d(i-1), \cdots, d(0) \right\}$$
(2.7)

where $col \{\cdot\}$ is the column vector operator, $J_{LS}(i)$ can be written as

$$J_{LS}(i) = \|y_i - H_i w\|_2^2 \tag{2.8}$$

and via standard (modern) least-squares theory the vector w_i that minimizes $J_{LS}(i)$, is given by

$$w_i = H_i^+ z_i \tag{2.9}$$

where H_i^+ denotes the *pseudo-inverse* of H_i [6]. Assuming H_i^+ has full rank, its

explicit expression is given by

$$H_i^+ = \begin{cases} H_i^* (H_i H_i^*)^{-1}, & i+1 \le M \\ (H_i^* H_i)^{-1} H_i^*, & i+1 > M. \end{cases}$$
(2.10)

Observe that if H_i does not have full rank, then the above formulae do not apply, since neither $H_iH_i^*$ nor $H_i^*H_i$ is invertible. In that case, an equivalent problem, albeit one with dimension smaller than M and in particular of dimension $rank(H_i)$, can be formulated and solved, using similar expressions with those given above. This approach, known as *dimensionality reduction* [7], will not be explored any further in the current thesis, since we will assume that the filter order M is fixed.

Alternatively to dimensionality reduction, and interestingly enough, regularization can be used to circumvent issues of invertibility. Inspired by the identity

$$H_i^* (H_i H_i^* + \epsilon I)^{-1} = (H_i^* H_i + \epsilon I)^{-1} H_i^*$$
(2.11)

where I is the identity matrix of suitable dimensions and ϵ is a *small* positive scalar, one could consider posing the regularized least-squares problem

$$\tilde{J}_{LS}(i) = \epsilon \|w\|_2^2 + \|z_i - H_i w\|_2^2$$
(2.12)

where $\epsilon \|w\|_2^2 = w^*(\epsilon I)w$ is the regularization term and ϵI is the underlying positive-definite regularization matrix. The solution that minimizes $\tilde{J}_{LS}(i)$ is given by

$$w_i = (H_i^* H_i + \epsilon I)^{-1} H_i^* z_i.$$
(2.13)

Observe now, that irrespectively of the dimensions or of the rank of H_i , the matrix $(H_i^*H_i + \epsilon I)$ is always invertible. Additionally, the regularized problem now solved can be thought of as a *perturbed* version of the original problem.

In fact, we can consider the more general regularized least-squares problem

$$\min_{w \in \mathbb{C}^{M \times 1}} \left\{ w^* \Pi w + \| z_i - H_i w \|_2^2 \right\}$$
(2.14)

where Π is a positive-definite regularization matrix. The solution is now given by

$$w_i = (H_i^* H_i + \Pi)^{-1} H_i^* z_i.$$
(2.15)

2.2 The RLS Algorithm

The solution $w_i = (H_i^* H_i + \Pi)^{-1} H_i^* z_i$ of the problem (2.14), can be computed by directly solving the *normal equations*

$$(H_i^* H_i + \Pi) w_i = H_i^* z_i \tag{2.16}$$

e.g. either by QR or by Cholesky decomposition, with both methods being of cubic complexity.

However, it turns out that the displacement structure [8] of the coefficient matrix $(H_i^*H_i + \Pi)$ is such that its displacement rank is equal to 1. As a direct consequence of this fact, w_i can be updated recursively (for increasing *i*) at quadratic complexity by the following recursions:

$$\gamma(i) = \left(1 + \lambda^{-1} u_i P_{i-1} u_i^*\right)^{-1}$$
(2.17)

$$g_i = \lambda^{-1} \gamma(i) P_{i-1} u_i^*$$
 (2.18)

$$e(i) = d(i) - u_i w_{i-1}$$
 (2.19)

$$w_i = w_{i-1} + g_i e(i) (2.20)$$

$$P_i = \lambda^{-1} P_{i-1} - g_i g_i^* / \gamma(i).$$
(2.21)

with initialization $w_{-1} = 0_{M \times 1}$ and $P_{-1} = (\Pi)^{-1}$ and

$$P_i \stackrel{\Delta}{=} \left(H_i^* H_i + \Pi\right)^{-1}. \tag{2.22}$$

Recursions (2.17)-(2.21) constitute the conventional RLS algorithm and the quantities $\gamma(i)$ and g_i are known as the *conversion factor* and *Kalman gain vector* respectively [2].

2.3 The Exponentially-Weighted RLS Algorithm

In adaptive filtering applications, the underlying data collection $\{u_j, d(j)\}_{j=0}^i$ might come from non-stationary signals. In such cases it is of interest to deemphasize the effect of older data in the computed solution, while weighting more the recent data, thus enhancing the tracking abilities of the adaptive filter.

This weighting of the data can be achieved by introducing a so-called *forget*ting factor

$$0 < \lambda < 1 \tag{2.23}$$

thus rendering the exponentially-weighted least-squares problem

$$\min_{w \in \mathbb{C}^{M \times 1}} \left\{ \sum_{j=0}^{i} \lambda^{i-j} \left| d(j) - u_j w \right|^2 \right\}$$
(2.24)

rewritten in matrix notation as

$$\min_{w \in \mathbb{C}^{M \times 1}} \left\{ \left\| \left(\Lambda_i^{1/2} z_i \right) - \left(\Lambda_i^{1/2} H_i \right) w \right\|_2^2 \right\}$$
(2.25)

with

$$\Lambda_i^{1/2} \stackrel{\Delta}{=} col\left\{1, \lambda^{1/2}, \cdots, \lambda^{i/2}\right\}.$$
(2.26)

This problem fits the standard least-squares formulation as in (2.8), albeit now the *data matrix* is $\Lambda_i^{1/2}H_i$ instead of H_i and its solution is given by the linear system of equations

$$H_i^* \Lambda_i H_i w_i = H_i^* \Lambda_i z_i. \tag{2.27}$$

Again, the coefficient matrix $H_i^* \Lambda_i H_i$ might not be invertible, thus possibly leading to an inconsistent system of linear equations. In cases where a filter of fixed order M is of interest (and thus dimensionality reduction can not be applied), regularization can be employed as before, i.e. we can consider the regularized exponentially-weighted least-squares problem

$$\min_{w \in \mathbb{C}^{M \times 1}} \left\{ w^* \Pi w + \left\| \left(\Lambda_i^{1/2} z_i \right) - \left(\Lambda_i^{1/2} H_i \right) w \right\|_2^2 \right\}$$
(2.28)

whose solution is

$$w_i = (H_i^* \Lambda_i H_i + \Pi)^{-1} H_i^* \Lambda_i z_i.$$
(2.29)

Now, towards avoiding the cubic complexity that is required for a direct computation of w_i from the above formula, we can consider weighting the regularization matrix itself, i.e. replacing Π by $\lambda^{i+1}\Pi$, thus rendering the problem

$$\min_{w \in \mathbb{C}^{M \times 1}} \left\{ w^* \left(\lambda^{i+1} \Pi \right) w + \left\| \left(\Lambda_i^{1/2} z_i \right) - \left(\Lambda_i^{1/2} H_i \right) w \right\|_2^2 \right\}$$
(2.30)

whose solution can be updated for increasing i at quadratic complexity via the recursions

$$\gamma(i) = \left(1 + \lambda^{-1} u_i P_{i-1} u_i^*\right)^{-1}$$
(2.31)

$$g_i = \lambda^{-1} \gamma(i) P_{i-1} u_i^*$$
 (2.32)

$$e(i) = d(i) - u_i w_{i-1}$$
 (2.33)

$$w_i = w_{i-1} + g_i e(i) (2.34)$$

$$P_i = \lambda^{-1} P_{i-1} - g_i g_i^* / \gamma(i).$$
(2.35)

where in order to avoid an explosion of notation we redefine P_i as

$$P_i \stackrel{\Delta}{=} R_i^{-1} \tag{2.36}$$

$$R_i \stackrel{\Delta}{=} \bar{R}_i + \lambda^{i+1} \Pi \tag{2.37}$$

$$\bar{R}_i \stackrel{\Delta}{=} \sum_{j=0}^{j} \lambda^{i-j} u_j^* u_j. \tag{2.38}$$

Equations (2.31)-(2.35) constitute the conventional exponentially-weighted RLS algorithm, to which we will be referring simply as RLS.

2.4 Regularization Issues in RLS

We will now take a closer look into the interplay between computational complexity and regularization in RLS.

Let us consider for a moment that no data weighting takes place, i.e. that $\lambda = 1$. Then

$$R_i = \bar{R}_i + \Pi. \tag{2.39}$$

But observe that

$$\bar{R}_i = \bar{R}_{i-1} + u_i^* u_i \tag{2.40}$$

and hence

$$R_i = \bar{R}_{i-1} + u_i^* u_i + \Pi \tag{2.41}$$

$$= (\bar{R}_{i-1} + \Pi) + u_i^* u_i \tag{2.42}$$

$$= R_{i-1} + u_i^* u_i \tag{2.43}$$

and consequently R_i is related to R_{i-1} via a rank-1 update, since $rank(u_i^*u_i) = 1$. It is this very fact that allows, via the *matrix inversion lemma* [2], for the inverse of R_i , i.e. for P_i , to be related to P_{i-1} via a rank-1 update as well, rendering the overall RLS complexity of quadratic order.

Now, assume that it is of interest to employ time-varying regularization matrices, i.e. we replace Π by Π_i . Then the corresponding regularized sample autocorrelation matrix becomes

$$R_i = \bar{R}_i + \Pi_i \tag{2.44}$$

and towards relating R_i to R_{i-1} , we proceed as follows:

$$R_i = \bar{R}_i + \Pi_i \tag{2.45}$$

$$= \bar{R}_{i-1} + u_i^* u_i + \Pi_i \tag{2.46}$$

$$= \left(\bar{R}_{i-1} + \Pi_{i-1}\right) + \left(u_i^* u_i + \Pi_i - \Pi_{i+1}\right)$$
(2.47)

$$= R_{i-1} + (u_i^* u_i + \Pi_i - \Pi_{i+1})$$
(2.48)

and since Π_i might be a completely different matrix from Π_{i+1} , we see that in general R_i is now related to R_{i-1} via a full-rank update, which implies that the mapping $P_{i-1} \mapsto P_i$ can only be performed via cubic complexity, thus rendering an RLS algorithm of cubic complexity.

Now, let us consider that λ is not necessarily equal to 1, while the regularization matrix is not weighted but fixed in time. Then

$$\bar{R}_i = \lambda \bar{R}_{i-1} + u_i^* u_i \tag{2.49}$$

and so

$$R_i = \bar{R}_i + \Pi \tag{2.50}$$

$$= \lambda \bar{R}_{i-1} + u_i^* u_i + \Pi \tag{2.51}$$

$$= \lambda \left(\bar{R}_{i-1} + \Pi \right) + \left(u_i^* u_i + \Pi - \lambda \Pi \right)$$
(2.52)

$$= \lambda R_{i-1} + (u_i^* u_i + \Pi - \lambda \Pi).$$
 (2.53)

But then $(u_i^*u_i + \Pi - \lambda \Pi)$ is in general a full-rank matrix and consequently, as before, this renders an RLS algorithm of cubic complexity. On the contrary, if we select $\lambda^{i+1}\Pi$ as our regularization matrix, then we see that

$$R_i = \bar{R}_i + \lambda^{i+1} \Pi \tag{2.54}$$

$$= \lambda R_{i-1} + u_i^* u_i \tag{2.55}$$

and so quadratic complexity can be achieved. The price to be paid however, is

that

$$\lambda^{i+1} \Pi \to 0_{M \times M} \tag{2.56}$$

as $i \to \infty$, i.e. in the long run there is no regularization! Obviously, this is an undesirable property, since the propagated matrix P_i might as well correspond to the inverse of a matrix that is not invertible, thus leading to unpredictable algorithmic behaviour.

2.5 Existing Approaches

The literature on the regularization of RLS is rather limited, particularly because, as we saw in the previous section, generic time-variant non-fading regularization leads to cubic complexity, which had been prohibitive for practical applications for many years, until the recent advent of fast multi-core platforms [9], that allow for processing powers of the order of Tera-FLOPS i.e., 10¹² floatingpoint operations per second.

What has traditionally been the standard practice towards preventing P_i from becoming unbounded, is to inflict regularization implicitely, via a process referred to as *dithering* [10]. Dithering consists of injecting uncorrelated white noise of small variance into the input signal u, thus making its sample autocorrelation matrix positive-definite in the mean-square sense. However, this technique has the obvious disadvantage of direct performance degradation of the adaptive filter and should be used with care.

Several *semi-direct* regularization approaches have been proposed, see e.g. [11] or [12] and references therein, which attempt to indirectly decrease the probability of R_i becoming non-invertible, while keeping the computational complexity close to quadratic. However, these methods do not guarantee the invertibility of R_i and they usually incorporate a number of design parameters to be selected

a-priori, a selection that for practical applications might be far from straightforward. The value of these methods seems to lie more in the insights that their analytical perspective offers, rather than in the establishment of a robust RLS algorithm that is reliable for hard application scenarios.

More recently, in [13] an interesting regularization structure was proposed, which consists of a regularization matrix that is updated in time via rank-1 matrices. This structure allows for *inflation* or *deflation* of the regularization matrix and can guarantee *on the average* the positive definiteness of R_i with a complexity of $O(2M^2)$. However, strictly speaking, R_i is still positive-semidefinite, i.e. its smallest eigenvalue is not prevented from becoming zero.

Finally, a non-fading regularization RLS algorithm has appeared in the literature, referred to as *leaky-RLS* [14], which corresponds to the problem

$$\min_{w \in \mathbb{C}^{M \times 1}} \left\{ \epsilon \|w\|_2^2 + \sum_{j=0}^i \lambda^{i-j} |d(j) - u_j w|^2 \right\}.$$
 (2.57)

This algorithm is of cubic complexity, while in [14] it is argued that a quadratic implementation is also possible. However, an explicit listing of this implementation has not yet been given.

2.6 The Need for Extensions

The standard RLS algorithm (regularization matrix equal to $\lambda^{i+1}\Pi$) and particularly recursion (2.35) for P_i , is known to be highly sensitive to finite precision effects [15]. In particular, although P_i is in theory a symmetric positive-definite matrix, in practical implementations it might loose both its symmetry and its positive definiteness. Despite the fact that preventing loss of symmetry is a relatively simple task (one can only propagate the upper triangular part of P_i), preventing loss of positive definiteness is very difficult. Consequently, the various known regularization approaches that are applied directly on P_i can still not prevent the algorithm from easily becoming unstable in finite precision, as is the case e.g. with [14].

On the other hand, there exist numerically robust RLS variants that are, from a theoretical point of view, equivalent to recursions (2.31)-(2.35). These variants are known as array forms and the most important representatives are the Square-Root RLS (SR-RLS) and the QR-RLS [2], [16]. The array forms, are based on the propagation via unitary transformations of some matrix squareroot of P_i thus achieving two things: 1) the finite precision round-off errors are bounded, since the processing is performed via unitary transformations and 2) the underlying autocorrelation matrix or its inverse (the information matrix) are always at least positive-semidefinite. However, the array forms still suffer from the fading regularization phenomenon (since they are equivalent to RLS) and they can still face numerical problems. Remeding (even partially) this problem for array forms has not yet been considered in the literature.

In addition, the significance of generic time-varying regularization has not been emphasized in the RLS literature. For example, in [14], the regularization matrix is fixed over time and equal to ϵI . However, there exist applications, where employing time-varying regularization of the sample autocorrelation matrix is of critical importance, such as in adaptive beamforming [17], [18]. Moreover, algorithms supporting time-varying regularization can admit more sophisticated numerical protection, as is highlighted in [19].

Finally, an important open problem in the literature, is, in algorithms that support regularization, how should the regularization parameters be selected? Although in the adaptive filtering literature, several results exist for selecting the amount of regularization in the Affine Projection Algorithm (APA) or in the NLMS algorithm, see e.g. [19], [20], [21], [22], no similar results seem to exist for

2.7 The Contributions of the Thesis

This thesis attempts to fill some of the gaps in the RLS regularization literature, as mentioned in the previous section. In particular

- 1. in chapter 3 a novel array RLS algorithm is presented, that is equipped with both a forgetting factor and time-varying regularization matrix with generic structure. In this way the problem of fading regularization, inherent to the standard exponentially-weighted RLS is circumvented, while the proposed algorithm improves on the leaky-RLS of [14] in the sense that 1) all computations are performed via unitary transformations and 2) the regularization matrix can be selected arbitrarily by the designer according to any desired rule (while for leaky-RLS it is strictly equal to ϵI). Finally, the presented algorithm is robust to small values of the forgetting factor, as is required in tracking applications [23].
- 2. in chapter 4 a novel criterion is motivated and presented for the dynamical regulation of regularization in the context of the standard RLS algorithm, implicitely achieved via dithering of the input signal. The proposed criterion is of general applicability and aims at achieving a balance between the accuracy of the numerical solution of a perturbed linear system of equations and its distance from the analytical solution of the original system, for a given computational precision. Extensive simulations show that the proposed criterion can be effectively used for the compensation of large condition numbers, small finite precisions and uncessary large values of the regularization.

3 AN ARRAY RLS ALGORITHM WITH GENERIC REGULARIZATION MATRIX

3.1 Introduction

Towards solving a regularized least-squares problem with exponential data weighting that allows for time-varying non-fading regularization of arbitrary matrix structure, one can choose to solve the following normal equations

$$\tilde{R}_i w_i = y_i \tag{3.1}$$

where

$$\tilde{R}_i \stackrel{\Delta}{=} \bar{R}_i + \Pi_i \tag{3.2}$$

$$\bar{R}_i \stackrel{\Delta}{=} \sum_{j=0}^{i} \lambda^{i-j} u_j^* u_j \tag{3.3}$$

$$y_i \stackrel{\Delta}{=} \sum_{j=0}^{i} \lambda^{i-j} d(j) u_j^* \tag{3.4}$$

and Π_i is an arbitrary regularization matrix. Moreover, the quantities \bar{R}_i and y_i can recursively be computed as follows [2]:

$$\bar{R}_i = \lambda \bar{R}_{i-1} + u_i^* u_i \tag{3.5}$$

$$y_i = \lambda y_{i-1} + d(i)u_i^*.$$
 (3.6)

However, this approach is problematic when precision is not high or the data are not persistently exciting, since

- the backward (forward) substitution process during the direct solution of (3.1) might face numerical difficulties
- 2. recursion (3.5) is quite sensitive in quantization effects in the sense that \bar{R}_i can easily become indefinite.

In this work we adopt a novel approach that alleviates the numerical difficulties associated with (3.1)-(3.5). The method is based on the propagation via unitary transformations of the square-root factors \tilde{L}_i , \tilde{S}_i of \tilde{R}_i , $\tilde{P}_i \triangleq \tilde{R}_i^{-1}$, respectively; \tilde{L}_i having considerably smaller condition number than \tilde{R}_i . Towards this end, we revisit results from array-form theory and list SR-RLS for ease of reference (Secs. 3.2/3.3), while in Secs. 3.4 and 3.5 we explicitly regularize the SR-RLS structural equations and complete the resulting set of equations, arriving at an array RLS algorithm allowing for time-varying regularization matrix of arbitrary structure.

3.2 Elements of Array-Forms Theory

We summarize the matrix theoretical basis for deriving RLS array forms ([2], pp. 561–579) afresh in the two following lemmas. Let¹

$$R = L L^* \tag{3.7}$$

be the LTCD of the positive definite matrix R, where L is the LTCF of R. Then the UTCD of $P \triangleq R^{-1}$ is given by

$$P = (L^{-1})^* (L^{-1}) \tag{3.8}$$

where $(L^{-1})^*$ is the UTCF of *P*.

¹All Cholesky factors are assumed to have positive diagonal elements and are thus unique [24]. For short, we also use: LTCF (UTCF) for Lower (Upper) Triangular Cholesky Factor and LTCD (UTCD) for Lower (Upper) Triangular Cholesky Decomposition.

Proof. Inverting both sides of $R = LL^*$ yields

$$P = (L^{-1})^* (L^{-1}).$$
(3.9)

Since L is lower triangular, so will be its inverse L^{-1} . Moreover, the eigenvalues of L^{-1} will be the reciprocals of the eigenvalues of L. But the eigenvalues of a triangular matrix are the elements of its diagonal, which implies that the diagonal elements of L^{-1} will be positive and consequently $(L^{-1})^*$ must be the UTCF of P.

Let A, B, D and E be given matrices of suitable dimensions and let the matrix

$$\left[\begin{array}{cc}A & B\end{array}\right] \tag{3.10}$$

have full row rank. Assume that it is desirable to determine a lower (upper) triangular matrix C with positive diagonal elements and a matrix F such that

$$CC^* = AA^* + BB^* \tag{3.11}$$

$$FC^* = DA^* + EB^*.$$
 (3.12)

Then if the *pre-array*

$$\left[\begin{array}{cc} A & B \\ D & E \end{array}\right] \tag{3.13}$$

is transformed via a composition of unitary transformations (and it is always possible to find such transformations), say Θ , into a *post-array*

$$\begin{bmatrix} A & B \\ D & E \end{bmatrix} \Theta = \begin{bmatrix} X & 0 \\ Y & Z \end{bmatrix}$$
(3.14)

where Θ is unitary and the block A has been lower (upper) triangularized yielding the block X with positive diagonal elements, while the block B has been annihilated, it will hold that

$$X = C \tag{3.15}$$

$$Y = F. (3.16)$$

In the simplified case where given A and B only a lower (upper) triangular matrix C with positive diagonal elements is sought so that (3.11) is satisfied, then annihilating the B block of the pre-array (3.10) and lower (upper) triangularizing with positive diagonal elements the block A via unitary transformations, will result into the block A of the pre-array being replaced by C in the post-array.

Proof. First note that

$$AA^* + BB^* \tag{3.17}$$

is positive-definite and determining C is equivalent to determining its unique LTCF (UTCF). Moreover, since C is invertible, F is uniquely determined by (3.12). Now, multiplying from the right each side of (3.14) by its transpose and equating the upper leftmost and lower leftmost blocks of the resulting equation, yields X = C and Y = F. For the existence of Θ see [24].

3.3 Square-Root RLS

We are now ready to state the Square-Root RLS (also known as inverse-QR) recursions. [SR-RLS] Consider a complex data collection $\{u_j, d(j)\}_{j=0}^i$, where u_j is $1 \times M$ and d(j) scalar, $0 < \lambda < 1$ is a forgetting factor and Π a positive definite regularization matrix. Then the solution of the regularized exponentially-weighted least-squares problem

$$\min_{w \in \mathbb{C}^{M \times 1}} \left\{ \lambda^{i+1} \, w^* \, \Pi \, w + \sum_{j=0}^i \, \lambda^{i-j} \, |d(j) - u_j \, w|^2 \right\}$$
(3.18)

can be updated recursively for increasing *i* as follows. Begin with $S_{-1} = (\Sigma^{-1})^*$, where Σ is the LTCF of Π and $w_i = 0_{M \times 1}$ and iterate for $i \ge 0$:

1. Form the pre-array

$$\begin{bmatrix} 1 & \lambda^{-1/2} u_i S_{i-1} \\ 0_{M \times 1} & \lambda^{-1/2} S_{i-1} \end{bmatrix}$$
(3.19)

and transform it from the right via unitary transformations to obtain the post-array

$$\begin{bmatrix} \gamma(i)^{-1/2} & 0_{1 \times M} \\ g_i \gamma(i)^{-1/2} & S_i \end{bmatrix}$$
(3.20)

where the block S_i is required to be upper triangular with positive diagonal elements. Then S_i is the UTCF of P_i .

 Perform the weight vector update using elements of the post-array as follows:

$$w_i = w_{i-1} + \left[g_i \gamma(i)^{-1/2}\right] \left[\gamma(i)^{-1/2}\right]^{-1} e(i).$$
 (3.21)

Proof. Note that (2.31) and (2.32) can be written exactly in the form of (3.11) and (3.12), respectively, with the mappings

$$C \mapsto \gamma(i)^{-1/2} \qquad F \mapsto g_i \gamma(i)^{-1/2}$$

$$A \mapsto 1 \qquad B \mapsto \lambda^{-1/2} u_i S_{i-1}$$

$$D \mapsto 0_{M \times 1} \qquad E \mapsto \lambda^{-1/2} S_{i-1}$$
(3.22)

where S_{i-1} is the UTCF of P_{i-1} . Then observe that the matrix

$$\left[\begin{array}{cc}1 & \lambda^{-1/2}u_i S_{i-1}\end{array}\right] \tag{3.23}$$

has always full row rank and hence Lemma 3.2 implies that if the pre-array (3.19) is transformed from the right via a composition of unitary transformations, say

equivalent to a unitary transformation Θ_i , such that

$$\begin{bmatrix} 1 & \lambda^{-1/2} u_i S_{i-1} \\ 0_{M \times 1} & \lambda^{-1/2} S_{i-1} \end{bmatrix} \Theta_i = \begin{bmatrix} x(i) & 0_{1 \times M} \\ y_i & Z_i \end{bmatrix}$$
(3.24)

where x(i) > 0, it will hold that

$$x(i) = \gamma(i)^{-1/2} (3.25)$$

$$y_i = g_i \gamma(i)^{-1/2}.$$
 (3.26)

Now, we can further require that Θ_i is such that the block Z_i is upper triangular with positive diagonal elements. Multiplying each side of (3.24) by its transpose, equating the resulting lower rightmost blocks and invoking equation (2.35), yields

$$Z_i = S_i \tag{3.27}$$

which must be equal to the UTCF of P_i due to the uniqueness of the Cholesky factor.

3.4 Perturbing the Standard Square-Root RLS

Now we are ready to apply the core concept of this work: *perturb* the structural equations of SR-RLS, so that the Cholesky factor that is propagated corresponds to the fully regularized matrix \tilde{R}_i (see (3.1)), in contrast to R_i (see (2.37)) as is the case with the standard SR-RLS.

We proceed by defining the *perturbed conversion factor* as

$$\tilde{\gamma}(i) \stackrel{\Delta}{=} \left(1 + \lambda^{-1} u_i \, \tilde{P}_{i-1} u_i^* \right)^{-1} \tag{3.28}$$

where

$$\tilde{P}_i \stackrel{\Delta}{=} \tilde{R}_i^{-1} \tag{3.29}$$

with \tilde{R}_i defined as in (3.1) (Compare with (2.31)). Now, define the *perturbed*

Kalman gain vector as

$$\tilde{g}_i \stackrel{\Delta}{=} \lambda^{-1} \,\tilde{\gamma}(i) \,\tilde{P}_{i-1} \,u_i^*. \tag{3.30}$$

The fundamental observation now is that (3.28) and (3.30) can be written in the form of (3.11) and (3.12) respectively, where

$$C \mapsto \tilde{\gamma}(i)^{-1/2} \qquad F \mapsto \tilde{g}_i \, \tilde{\gamma}(i)^{-1/2}$$

$$A \mapsto 1 \qquad B \mapsto \lambda^{-1/2} \, u_i \, \tilde{S}_{i-1} \qquad (3.31)$$

$$D \mapsto 0_{M \times 1} \qquad E \mapsto \lambda^{-1/2} \, \tilde{S}_{i-1}$$

with \tilde{S}_i being the UTCF of \tilde{P}_i . Since

$$\left[\begin{array}{cc}1 & \lambda^{-1/2}u_i\,\tilde{S}_{i-1}\end{array}\right] \tag{3.32}$$

has always full rank, Lemma 3.2 implies that for any unitary transformation $\tilde{\Theta}_i^{(1)}$, such that

$$\begin{bmatrix} 1 & \lambda^{-1/2} u_i \tilde{S}_{i-1} \\ 0_{M \times 1} & \lambda^{-1/2} \tilde{S}_{i-1} \end{bmatrix} \tilde{\Theta}_i^{(1)} = \begin{bmatrix} \tilde{x}(i) & 0_{1 \times M} \\ \\ \tilde{y}_i & \tilde{Z}_i^{(1)} \end{bmatrix}$$
(3.33)

where $\tilde{x}(i) > 0$, it holds that

$$\tilde{x}(i) = \tilde{\gamma}(i)^{-1/2} \tag{3.34}$$

$$\tilde{y}_i = \tilde{g}_i \,\tilde{\gamma}(i)^{-1/2}. \tag{3.35}$$

This implies that the weight vector update can be performed exactly as in (3.21), albeit using the more robust quantities \tilde{g}_i and $\tilde{\gamma}(i)$.

Contrary however to the standard SR-RLS (Theorem 3.3), further upper triangularization with positive diagonal elements of the block $\tilde{Z}_i^{(1)}$ will not yield \tilde{S}_i . The reason is that \tilde{R}_i is not a rank-1 update of \tilde{R}_{i-1} anymore, as with R_i , therefore a recursion for \tilde{P}_i in the form of (2.35) cannot be obtained.

3.5 Completing the Recursion

We now need to find a way to generate \tilde{S}_i via unitary transformations. Letting \tilde{L}_i and Σ_i be the LTCFs of \tilde{R}_i and Π_i , respectively, and

$$\bar{R}_i = \bar{L}_i \bar{L}_i^* \tag{3.36}$$

with \bar{L}_i being a lower triangular with non-negative diagonal elements square-root factor of \bar{R}_i (\bar{R}_i might be singular), (3.1) becomes:

$$\tilde{L}_i \tilde{L}_i^* = \bar{L}_i \bar{L}_i^* + \Sigma_i \Sigma_i^*.$$
(3.37)

In addition, Lemma 3.2 ensures that the quantity \tilde{S}_i , to be determined, must satisfy the equation

$$\tilde{S}_i \, \tilde{L}_i^* = I_M \tag{3.38}$$

where I_M is the $M \times M$ identity matrix, which can be equivalently written in the more suggesting form

$$\tilde{S}_i \, \tilde{L}_i^* = 0_{M \times M} \, \bar{L}_i^* + (\Sigma_i^*)^{-1} \, \Sigma_i^* \, .$$
(3.39)

But now (3.37) and (3.39) are exactly in the form of (3.11) and (3.12), respectively, where

$$C \mapsto \tilde{L}_i \quad , \quad F \mapsto \tilde{S}_i \tag{3.40}$$

$$A \mapsto \bar{L}_i \quad , \quad B \mapsto \Sigma_i$$
 (3.41)

$$D \mapsto 0_{M \times M}$$
, $E \mapsto (\Sigma_i^*)^{-1}$. (3.42)

Then Lemma 3.2 implies that for any unitary $\tilde{\Theta}_i^{(2)}$, such that

$$\begin{bmatrix} \bar{L}_i & \Sigma_i \\ 0_{M \times M} & (\Sigma_i^*)^{-1} \end{bmatrix} \tilde{\Theta}_i^{(2)} = \begin{bmatrix} \tilde{X}_i & 0_{M \times M} \\ \tilde{Y}_i & \tilde{Z}_i^{(2)} \end{bmatrix}$$
(3.43)

where \tilde{X}_i is lower triangular with positive diagonal elements, it holds that

$$\tilde{X}_i = \tilde{L}_i \tag{3.44}$$

and

$$\tilde{Y}_i = \tilde{S}_i. \tag{3.45}$$

Due to the potential singularity of \bar{R}_i , \bar{L}_i might be non-unique. However, the important point is to be able to map some \bar{L}_{i-1} to some \bar{L}_i , and this is always possible and a well-conditioned problem. To see this, assume that a lower-triangular square-root factor of \bar{R}_{i-1} with non-negative diagonal elements, denoted \bar{L}_{i-1} , is available. Then observe that if we zero out via a unitary transformation $\bar{\Theta}_i$ the $M \times 1$ rightmost block of the pre-array

$$\left[\begin{array}{cc}\lambda^{1/2}\,\bar{L}_{i-1} & u_i^*\end{array}\right] \tag{3.46}$$

i.e.

$$\begin{bmatrix} \lambda^{1/2} \bar{L}_{i-1} & u_i^* \end{bmatrix} \bar{\Theta}_i = \begin{bmatrix} \bar{X}_i & 0_{M \times 1} \end{bmatrix}$$
(3.47)

while forcing \bar{X}_i to be lower triangular with non-negative diagonal elements (such a $\bar{\Theta}_i$ always exists), then multiplying each side by its transpose yields

$$\bar{R}_{i} = \lambda \bar{L}_{i-1} \bar{L}_{i-1}^{*} + u_{i}^{*} u_{i}$$
(3.48)

$$= \bar{X}_i \bar{X}_i^* \tag{3.49}$$

therefore

$$\bar{L}_i \stackrel{\Delta}{=} \bar{X}_i \tag{3.50}$$

thus leading to a recursive computation of a square-root factor \bar{L}_i for \bar{R}_i . This square-root factor \bar{L}_i is equal to the LTCF of \bar{R}_i , in case the latter is nonsingular.

3.6 Proposed Algorithm

Non-Fading Regularization Array RLS (NFRA-RLS): Select an initial positive definite regularization matrix Π_{-1} with LTCF Σ_{-1} , set $\tilde{S}_{-1} = (\Sigma_{-1}^*)^{-1}$, $\bar{L}_{-1} = 0_{M \times M}$ and iterate for $0 \le i \le N$:

1. Perform via any unitary $\tilde{\Theta}_i^{(1)}$ the mapping

$$\begin{bmatrix} 1 & \lambda^{-1/2} u_i \, \tilde{S}_{i-1} \\ 0_{M \times 1} & \lambda^{-1/2} \, \tilde{S}_{i-1} \end{bmatrix} \tilde{\Theta}_i^{(1)} = \begin{bmatrix} \tilde{\gamma}(i)^{-1/2} & 0_{1 \times M} \\ \tilde{g}_i \, \tilde{\gamma}(i)^{-1/2} & \tilde{Z}_i^{(1)} \end{bmatrix}$$
(3.51)

where $\tilde{\gamma}(i)^{-1/2}$ must be positive, while $\tilde{Z}_i^{(1)}$ is of no interest.

2. Perform the weight vector update as follows:

$$w_{i} = w_{i-1} + \frac{\left[\tilde{g}_{i}\tilde{\gamma}(i)^{-1/2}\right]}{\left[\tilde{\gamma}(i)^{-1/2}\right]} [d(i) - u_{i}w_{i-1}].$$
(3.52)

3. Perform via any unitary $\overline{\Theta}_i$ the mapping

$$\begin{bmatrix} \lambda^{1/2} \bar{L}_{i-1} & u_i^* \end{bmatrix} \bar{\Theta}_i = \begin{bmatrix} \bar{L}_i & 0_{M \times 1} \end{bmatrix}$$
(3.53)

where the block \bar{L}_i must be lower triangular with real-valued non-negative diagonal elements.

- 4. Determine via some rule Π_i and its LTCF Σ_i .
- 5. Perform via any unitary $\tilde{\Theta}_i^{(2)}$ the mapping

$$\begin{bmatrix} \bar{L}_i & \Sigma_i \\ 0_{M \times M} & (\Sigma_i^*)^{-1} \end{bmatrix} \tilde{\Theta}_i^{(2)} = \begin{bmatrix} \tilde{L}_i & 0_{M \times M} \\ \tilde{S}_i & \tilde{Z}_i^{(2)} \end{bmatrix}$$
(3.54)

where \tilde{L}_i must be lower triangular with positive diagonal elements, while $\tilde{Z}_i^{(2)}$ is of no interest. The complexity of this step is $\frac{M^3}{2}$ complex Givens rotations.

3.7 A Comment on the Computational Complexity

Step 5) of the algorithm requires $O\left(\frac{M^2}{2}\right)$ Givens rotations, thus implying cubic complexity if the algorithm is implemented on a sequential processor. However, it is possible to map the algorithm on a systolic array [25] using square-root and division free Givens rotations [26], thus arriving at an efficient implementation. Moreover, note that steps 1) and 2) are independent of 3), 4) and 5) so that they can be carried out in parallel.

Cubic complexity might be prohibitive for some applications. However, cubic complexity is comparable to quadratic complexity (or even to linear complexity) for small filter length, as encountered in some beamforming instances (in [27] a 4element antenna array has been designed for W-CDMA), or in DS-CDMA/OFDM systems, where frequency selective multipath channels are usually modeled as short FIR filters (e.g., with 2 to 5 taps [28]-[29]). Further applications involving small filter length are combination of adaptive filters, where $m \geq 2$ filters are combined via the adaptation of m-1 parameters, [30], and adaptive notch filtering, where cancelling p sinusoids requires the adaptation of p or 2p parameters [31], with both m and p small. In addition, note that cubic complexity is inherent in beamforming, i.e., the standard LSMI/MVDR beamformers are cubic [32] and a cubic RLS beamformer is available [33]. Finally, cubic complexity can also be considered for higher order filters, when robustness is mandatory, due to the advent of fast multicore platforms [9].

3.8 Simulations

We consider a desired sinusoidal signal impinging from 45° on a linear uniform antenna array of M = 4 elements [27] with inter-element spacing d = l/5, where l is the wavelength of the carrier. An orthogonal sinusoidal interference at 26 dB is also present from 140°. The measurement noise is at -27 dB and is Gaussian and white both in time and space. Note that this scenario corresponds to insufficiently persistent data, since the interference dominates.

We compare NFRA-RLS with dithered SR-RLS (DSR-RLS), the normal equations (NE) of (3.1), the standard RLS recursion using the regularization structure of [13] referred to as R1R-RLS and with the Loaded Sample Matrix Inverse (LSMI) beamformer [18]. All RLS variants use $\lambda = 0.995$, while LSMI uses a data window of length $K = \lfloor -1/log\lambda \rfloor$. NFRA-RLS, NE and LSMI use $\Pi_i = \epsilon I_M, \ \epsilon = 1$, while DSR-RLS uses differing variance equal to $\epsilon(1 - \lambda)$ and R1R-RLS parameters $\phi_n = 1$ and $\xi_n^2 = \epsilon(1 - \lambda)M$ (see [13] equation (13)) so that they correspond to effective regularization equal to ϵ . The accuracy of computations is B = 32 bits of floating point precision. The SINR [33] along with the condition number (CN) for all algorithms averaged over 50 experiments are depicted in Fig. 1 and 2 respectively. The optimal SINR of the Minimum Variance Distortionless Response (MVDR) beamformer [2] is also shown. We observe that NE, R1R-RLS and NFRA-RLS perform similarly and close to the optimal, while LSMI exhibits relatively slow convergence and as expected DSR-RLS poor SINR due to dithering. In the sequel we perform the same simulation, albeit with B = 13 Fig. 3. Then we see that the coupling effect of insufficiently persistend data (large CN) and small accuracy result in LSMI, NE and R1R-RLS to become unstable while although stable, DSR-RLS exhibits poor SINR due to dithering. NFRA-RLS is practically not affected by the accuracy reduction. Finally, we consider the same scenario, albeit the interference now moves at a rate of 0.1° per iteration and $\lambda = 0.9$ Fig. 4. Note how NFRA-RLS exhibits robustness to this hard scenario (large CN, small λ , small B), while the other algorithms become unstable.



Figure 1: Desired signal from 25°, 26dB interference from 140°, -27dB measurement noise, B = 32 bits, M = 4, $\epsilon = 1$, $\lambda = 0.995$, 50 experiments.



Figure 2: Desired signal from 25°, 26dB interference from 140°, -27dB measurement noise, B = 32 bits, M = 4, $\epsilon = 1$, $\lambda = 0.995$, 50 experiments.



Figure 3: Desired signal from 25°, 26dB interference from 140°, -27dB measurement noise, B = 13 bits, M = 4, $\epsilon = 1$, $\lambda = 0.995$, 50 experiments.



Figure 4: Desired signal from 25°, 26dB interference moving at 0.1° /iteration, -27dB measurement noise, $\lambda = 0.9$, B = 13 bits, M = 4, $\epsilon = 1$, 50 experiments.

4 AN ALTERNATIVE CRITERION TO REGULARIZE RLS PROBLEMS

4.1 Introduction

In finite precision the solution of linear systems of equations may be greatly affected by the data conditioning, so that regularization is employed in order to enhance numerical stability [34], [35]. Particularly, in the systems that arise in adaptive signal processing the interplay between computational precision (i.e., the number of bits) and the data conditioning is what ultimately governs the overall numerical properties of the adaptive algorithms, and it is not a trivial task to select the "amount" of regularization required in a given scenario.

There exist ways to control regularization [19], [21]. The condition number [36], [7] is a typical measure of how well the required inversion in the system of equations can be performed, and it can be employed to guide the regularization process [19]. However, it does not capture properly how that ultimately affects the solution that the numerical algorithm outputs. In this work we pose an accuracy criterion that accounts simultaneously for the data conditioning as well as for the numerical precision when solving a regularized linear system of equations. It quantifies how far is the numerical solution of the regularized system from the analytical solution of the original system.

Simulations show that the new criterion, the *image function*, captures well variations in data conditioning, and it promptly reacts to the numerical precision

available.

We motivate the new criterion by proposing a pilot algorithm that regularizes the standard recursive least-squares algorithm (RLS) via dithering of the input regressor. The image function is simply fed back into the dithering mechanism, so that the resulting algorithm regulates automatically the amount of regularization (the dither noise variance) in response to changes in the adaptation scenario.

4.2 The Image Function

We start by considering an abstract $M \times M$ linear system of equations

$$Ax = b \tag{4.1}$$

where A is positive-semidefinite. In order to stabilize the numerical process of solving this system in finite precision, we perturb A and b in some way and instead attempt solving the system¹

$$\tilde{A}\tilde{x} = \tilde{b} \tag{4.2}$$

where we assume that the structure of the perturbation is such that the larger the matrix perturbation

$$\left\| A - \tilde{A} \right\| \tag{4.3}$$

is, the smaller the condition number $\chi\left(\tilde{A}\right)$ of \tilde{A} will be. As an example, one choice could be

$$\tilde{A} = A + \epsilon I \tag{4.4}$$

where I is the identity matrix and $\tilde{b} = b$. This corresponds to the well-known regularization structure.

Let $[\tilde{x}]$ be the solution in finite precision for (4.2). We would like to perturb the original system (4.1) just enough so that the following cost function is

¹The use of tilde in this chapter should be distinguished from that of the previous chapter.



Figure 5: The *J* function (4.5) for an $M \times M$ system $(A + \epsilon I) = b$ solved under *B* bits of precision with M = 30 and B = 20. The condition number of the coefficient matrix *A* is $CN(A) \propto 10^3$.

minimized

$$J \stackrel{\Delta}{=} \|x - [\tilde{x}]\|. \tag{4.5}$$

Note that J penalizes (1) large condition numbers, (2) low numerical precisions and (3) large perturbations (see Fig. 5). To see this intuitively, note that [7]

$$\frac{\|\tilde{x} - [\tilde{x}]\|}{\|\tilde{x}\|} \le \chi\left(\tilde{A}\right) f(B) \tag{4.6}$$

where f(B) is a function of the number B of available bits, which in principle obtains large values for small values of B. Now consider the following three scenarios.

To begin with (*large condition numbers*), assume that $\chi\left(\tilde{A}\right)$ is large, while the perturbation and B are moderately valued. Then the quantity $\chi\left(\tilde{A}\right) f(B)$ will be large and hence $\|\tilde{x} - [\tilde{x}]\|$ will (in general) be large as well. But since the perturbation is small, we will have that $\tilde{x} \approx x$ and consequently $J = \|x - [\tilde{x}]\|$ will be large. In the sequel (small precisions), assume that B is small, while $\chi\left(\tilde{A}\right)$ and the perturbation are moderately valued. Then the quantity $\chi\left(\tilde{A}\right) f(B)$ will be large and hence $\|\tilde{x} - [\tilde{x}]\|$ will (in general) be large as well. But since the perturbation is small, we will have that $\tilde{x} \approx x$ and consequently $J = \|x - [\tilde{x}]\|$ will be large.

Finally (*large perturbations*), assume that B is moderately valued, while the perturbation is large, which implies that $\chi\left(\tilde{A}\right)$ will be small but $||x - \tilde{x}||$ will be large. Consequently, $\chi\left(\tilde{A}\right) f(B)$ will be small, so that $[\tilde{x}] \approx \tilde{x}$ and hence $J = ||x - [\tilde{x}]||$ will be large as well.

An important difficulty associated with the computation of J is that it requires knowledge of x, which is the very quantity that we are trying to estimate. To circumvent this issue, note that

$$J \stackrel{(4.1)}{=} \|A^{-1}b - [\tilde{x}]\| \tag{4.7}$$

$$\leq ||A^{-1}|| ||b - A[\tilde{x}]|| \tag{4.8}$$

and since $||A^{-1}||$ is not a function of the perturbation, we can choose instead to minimize the measurable quantity

$$J_{im} \stackrel{\Delta}{=} \|b - A[\tilde{x}]\| \tag{4.9}$$

which is a relative upper bound of J.

 J_{im} will be referred to as the *image function*, since the vector

$$b - A\left[\tilde{x}\right] \tag{4.10}$$

is the image of the *error* vector

$$x - [\tilde{x}] \tag{4.11}$$

under the linear transformation A [37].

4.3 Contrived Example

To further illustrate the power of the concept behind the criterion J we discuss a simple numerical example for the case of regularization.

Assume that we want to solve the trivial linear system of equations

$$9x = 8 \tag{4.12}$$

on a platform that uses four decimal digits and no rounding takes place. The analytical solution of our equation is x = 8/9 = 0.8888... and the numerical solution that will be returned by our solver is [x] = 0.8888, leading to a numerical error

$$\|\mathbf{x} - [\mathbf{x}]\| = 0.8888... - 0.8888$$
 (4.13)

$$= 8.8888... \times 10^{-5}. \tag{4.14}$$

Now, the system (4.12) that we want to solve is perfectly well conditioned, since

$$\chi(9) = 1 \tag{4.15}$$

and there is no uncertainty in our data, i.e. we know with perfect accuracy our coefficient matrix

$$\mathbf{A} = 9 \tag{4.16}$$

and our right hand side vector

$$b = 8.$$
 (4.17)

Consequently, according to existing regularization approaches, there is no need to apply any regularization.

Now, following the ideas presented so far, we choose to solve the *regularized* system

$$(9 + \epsilon_{opt})\tilde{x} = 8 \tag{4.18}$$

instead of (4.12), with

$$\epsilon_{opt} = \arg\left\{\min_{\epsilon} \|\mathbf{x}_i - [\tilde{\mathbf{x}}]\|\right\}$$
(4.19)

$$= \arg\left\{\min_{\epsilon} \left\| 0.8888... - \left\lfloor \frac{8}{9+\epsilon} \right\rfloor \right\|\right\}$$
(4.20)

$$= -0.0002$$
 (4.21)

which yields a numerical error

$$J(\epsilon_{opt}) = \|\mathbf{x}_i - [\tilde{\mathbf{x}}]\|$$
(4.22)

$$= \left\| 0.8888... - \left[\frac{8}{8.9998} \right] \right\|$$
(4.23)

$$= \|0.8888... - 0.8889\| \tag{4.24}$$

$$= 1.1111... \times 10^{-5} \tag{4.25}$$

$$< 8.8888... \times 10^{-5} = J(0). \tag{4.26}$$

This example also shows that as long as we are getting more accurate numerical solutions, negative values of the regularization parameter can be allowed.

4.4 Image Function Analysis

The fact that the vector $(b - A[\tilde{x}])$ is the image of the *error* vector $(x - [\tilde{x}])$ under the linear transformation A sets some important limitations on how well we are doing by minimizing J_{im} given the fact that we would like to be minimizing J.

To further elaborate on this subtle issue, we introduce the eigendecomposition of the symmetric positive-definite matrix A

$$A = U\Lambda U^* \tag{4.27}$$

where U is orthogonal containing in its i^{th} column the i^{th} eigenvector of A and

$$\Lambda \stackrel{\Delta}{=} diag \left\{ \lambda_1, \cdots, \lambda_M \right\} \tag{4.28}$$

contains the corresponding positive eigenvalues, where it is also assumed that

$$\lambda_1 \ge \lambda_2 \ge \dots \ge \lambda_M > 0. \tag{4.29}$$

Now, using the Euclidean norm without loss of generality 2 we have

$$J_{im} = \|b - A[x]\|_2 \tag{4.30}$$

$$= ||A(x - [\tilde{x}])||_{2}$$
(4.31)

$$\stackrel{(4.27)}{=} \|U\Lambda U^* \left(x - [\tilde{x}]\right)\|_2. \tag{4.32}$$

Introducing the rotated error vector

$$v \stackrel{\Delta}{=} U^* \left(x - [\tilde{x}] \right) \tag{4.33}$$

and using the fact that the Euclidean norm is invariant under orthogonal transformations of its argument [6] (4.32) becomes

$$J_{im} = \sqrt{\left\|\Lambda v\right\|_2^2} \tag{4.34}$$

$$= \sqrt{v^* \Lambda^2 v} \tag{4.35}$$

$$= \sqrt{\sum_{j=1}^{M} \lambda_j^2 |v(j)|^2}$$
(4.36)

$$= \lambda_1 \sqrt{|v(1)|^2 + \sum_{j=2}^M \left(\frac{\lambda_j}{\lambda_1}\right)^2 |v(j)|^2}.$$
 (4.37)

Now let us raise momentarily our assumption that A is invertible and let us assume that $\lambda_M = 0$, while the rest of the eigenvalues are positive. Then we readily conclude from (4.37) that the component of the error vector $(x - [\tilde{x}])$ along the one-dimensional subspace spanned by the K^{th} eigenvector of A that corresponds to the zero eigenvalue is actually unobservable via the image function J_{im} . In other words, the component of $(x - [\tilde{x}])$ that belongs to the kernel of Ais annihiliated as we attempt to observe $(x - [\tilde{x}])$ via its image under A.

^{2}All norms in finite dimensional vector spaces are equivalent [6].

The practical implication of this phenomenon is that J_{im} might be obtaining infinitely-small values, while the actual error is huge and there is no way we could know. It is very important to emphasize though, that this is not an inherent disadvantage of J_{im} , rather an inherent problem of the data, which no regularization method could mend (one should consider reducing the dimension of the problem and then re-applying the concept of the image function).

Now, returning to our assumption that A is invertible, observe from (4.37) that the squares of the components of $(x - [\tilde{x}])$ along the one-dimensional eigenspaces of A are summed inside the radical with weights less than unity and observe that the smallest weight is equal to $\left(\frac{\lambda_M}{\lambda_1}\right)^2$, which is actually equal to the inverse of the square of the condition number of A. This implies, that as A becomes more and more ill-conditioned, the component of $(x - [\tilde{x}])$ along the eigenspace that tends to zero (i.e. the space spanned by the eignevectors that correspond to the eigenvalues that tend to zero) will be less and less weighted, i.e. it will tend to be unobservable. Again, this is an inherent problem of the data and not a weakness of J_{im} .

4.5 Application to RLS

We show an application of the previous concepts in adaptive filtering and in particular in the context of the Recursive Least-Squares (RLS) algorithm, by devising a simple feedback mechanism that adjusts the level of dithering [10], [38] in response to an appropritely defined image function.

The regularization matrix of the conventional RLS algorithm³ $\lambda^{i+1} \Pi$ serves to guarantee the invertibility of R_i during the initial stage of the algorithm and it fades towards zero for $i \to \infty$, giving rise to the probability of R_i loosing its invertibility and hence of P_i being a numerically unreliable quantity. To alleviate

³The reader is referred to chapter 2.

this event, it is standard practice to perturb the input signal by adding to it a small quantity of white noise, a process known as *dithering* or *noise injection* [10], [13], [2].

By direct multiplication of both sides of (2.35) from the right by u_i^* and incorporation of (2.32) we obtain the RLS identity

$$g_i = P_i u_i^* \tag{4.38}$$

which can equivalently be written as

$$R_i g_i = u_i^*. \tag{4.39}$$

If we now consider that the input signal is dithered and denote the dithered version of a quantity ξ as $\tilde{\xi}$, then the above identity becomes in the context of Dithered-RLS (DRLS)

$$\ddot{R}_i \tilde{g}_i = \tilde{u}_i^*. \tag{4.40}$$

The challenge now is to determine the amount of dithering so that the numerical Kalman gain vector of the dithered algorithm, denoted as $[\tilde{g}_i]$, to be as close as possible to the Kalman gain vector of the infinite precision undithered RLS, i.e. g_i . But this problem fits exactly the formulation of section 4.2 with the mappings

$$R_i \mapsto A \quad u_i^* \mapsto b \quad g_i \mapsto x. \tag{4.41}$$

Then a suboptimal level of dithering at iteration i can be obtained by minimizing

$$J_{im,RLS}(i) \stackrel{\Delta}{=} \|u_i^* - R_i [\tilde{g}_i]\|$$

$$(4.42)$$

which is a relative upper bound of the function

$$J_{RLS}(i) \stackrel{\Delta}{=} \|g_i - [\tilde{g}_i]\|.$$

$$(4.43)$$

Minimization of $J_{im,RLS}$ is a difficult problem still under investigation. Alternati-

vely, we can compensate for $J_{im,RLS}$ recursively in time by designing the variance of the injected white noise $\delta(i)$ as

$$\delta(i) = \alpha \delta(i-1) + \beta J_{im,RLS}(i) \tag{4.44}$$

where $0 < \alpha < 1$ and $0 < \beta \ll 1$ are scalars that determine the properties of the feedback mechanism. This leads to an RLS algorithm with automatic dithering regulation driven by the image function (4.42), referred to as IRLS (Image-RLS). Note that in order to compute $J_{im,RLS}(i)$ the matrix R_i is required, which however can be easily computed in a recursive manner as

$$R_i = \lambda R_{i-1} + u_i^* u_i. \tag{4.45}$$

The computational complexity of IRLS is $O(M^2)$ similarly to the standard RLS, since its effective additional computational burdern is the matrix-vector multiplication $R_i[\tilde{g}_i]$.

For practical implementations, and since all norms in $\mathbb{C}^{M \times 1}$ are equivalent, we suggest using the *infinity norm* for its simplicity, defined as

$$\|u_{i}^{*} - R_{i}[\tilde{g}_{i}]\|_{\infty} \stackrel{\Delta}{=} \max_{k=1,\cdots M} |u_{i}^{*}(k) - R_{i}(k,:)[\tilde{g}_{i}]|$$
(4.46)

where $u_i^*(k)$ is the k^{th} entry of u_i^* and $R_i(k, :)$ is the k^{th} row of R_i .

4.6 Proposed Algorithm

Summarizing the previous developments, we arrive at the following proposed algorithm:

Image-RLS (IRLS): Select a filter order M, a forgetting factor $0 < \lambda < 1$, an initial positive-definite regularization matrix Π , constants $0 < \alpha < 1$ and $0 < \beta \ll 1$, an initial level of dithering $\delta(0) = \delta_0 \ge 0$, set $w_{-1} = 0_{M \times 1}$, $\tilde{P}_{-1} = \Pi^{-1}$ and iterate for $i \ge 0$: Draw a number $\eta(i)$ from a random distribution of variance $\delta(i)$. The generation of this number must be independent from previous number generations.

$$\tilde{u}(i) = u(i) + \eta(i) \tag{4.47}$$

$$\tilde{\gamma}(i) = \left(1 + \lambda^{-1} \tilde{u}_i \tilde{P}_{i-1} \tilde{u}_i^*\right)^{-1}$$
(4.48)

$$\tilde{g}_i = \lambda^{-1} \tilde{\gamma}(i) \tilde{P}_{i-1} \tilde{u}_i^*$$
(4.49)

$$[\tilde{g}_i]$$
 : numerical version of \tilde{g}_i (4.50)

$$\tilde{e}(i) = d(i) - \tilde{u}_i w_{i-1}$$
 (4.51)

$$w_i = w_{i-1} + \tilde{g}_i \,\tilde{e}(i) \tag{4.52}$$

$$\tilde{P}_i = \lambda^{-1} \tilde{P}_{i-1} - \tilde{g}_i \tilde{g}_i^* / \tilde{\gamma}(i)$$
(4.53)

$$R_i = \lambda R_{i-1} + u_i^* u_i \tag{4.54}$$

$$J_{im,RLS}(i) = \max_{k=1,\dots,M} |u_i^*(k) - R_i(k,:)[\tilde{g}_i]|$$
(4.55)

$$\delta(i+1) = \alpha \delta(i) + \beta J_{im,RLS}(i). \tag{4.56}$$

4.7 Results

We present simulations where IRLS is compared with fixed-dithering RLS (DRLS) algorithms (fixed variance $\delta(i) = \delta$).

In Fig. (6) we demonstrate the ability of IRLS to respond to changes in variations of the underlying computational precision in order to achieve numerical stability. In the first 2000 iterations the precision is B = 32 bits; then in the following 3000 iterations the precision is abruptly switched to B = 18 bits. In the last 2000 iterations the precision switches back to the original B = 32 bits. Note how the image function *senses* the changes in precision, keeping IRLS robust, while the other fixed dithering implementations experience divergence peaks.

In Fig. (7) we demonstrate the ability of IRLS to respond in variations of the condition number (CN) of the sample autocorrelation matrix of the original input signal. In the first 2000 iterations the SNR is 20 dB (moderate CN)⁴; then in the following 3000 iterations the SNR is abruptly switched to 30 dB (large CN). In the last 2000 iterations the SNR switches back to its original value of 20 dB. Observe that as the SNR increases, the Mean-Square-Error of the prediction is decreased; however the numerical difficulty encountered by the algorithms is increased, since R_i is now ill-conditioned. Note how the image functin *senses* the CN variation and promptly increases the dithering level during the interval of ill-conditioning (SNR = 30 dB) to keep IRLS robust, while the other algorithms exhibit explosive divergence peaks.

In Fig. (8) we demonstrate the ability of IRLS to penalize unecessarily large perturbations. We initialize IRLS with a small value of dithering (0.0005) and at iteration i = 2000 we abruptly interfere to the feedback process and make the value of dithering equal to 0.5; then we resume the feedback of the image function. Observe how the image function *senses* unecessarily large values of perturbation (no need for large dithering at 32 bits) and promptly drifts the dithering to a very small value.

Finally, Fig. (9) depicts the IRLS algorithm in a linear prediction scenario for speech signals; note how, for fixed (low) precision, the image function captures the power and conditioning variations of the input signal, allowing the IRLS algorithm to react correspondingly. In both simulations fixed-dithering RLS algorithms fail. Note, particularly, that in the (b) curves the fixed "average" dithering ($\delta = 0.1$, or $\delta_{dB} = -20dB$), although unfair with IRLS (the average suboptimal dithering is an unknown quantity a priori), is outperformed by the proposed scheme.

⁴Note that the autocorrelation matrix of a noiseless sinusoid has rank 1.



Figure 6: Adaptive prediction of a pure sinusoidal tone. In all filters M = 6, SNR = 30 dB, $\lambda = 0.995$. Fixed dithering: (DRLS1) $\delta = 0.0005$ and (DRLS2) $\delta = 0.5$. IRLS is run with $\delta(-1) = 0.0005$, $\alpha = 0.99$ and $\gamma = 0.001$. Curves averaged over 100 independent experiments.



Figure 7: Adaptive prediction of a pure sinusoidal tone. In all filters M = 6, $B = 18 \ \lambda = 0.995$. Fixed dithering: (DRLS1) $\delta = 0.0005$ and (DRLS2) $\delta = 0.5$. IRLS is run with $\delta(-1) = 0.0005$, $\alpha = 0.99$ and $\gamma = 0.001$. Curves averaged over 100 independent experiments.



Figure 8: Adaptive prediction of a pure sinusoidal tone. In all filters M = 6, B = 32, SNR = 30 dB, $\lambda = 0.995$. Fixed dithering: (DRLS1) $\delta = 0.0005$ and (DRLS2) $\delta = 0.5$. IRLS is run with $\delta(-1) = 0.0005$, $\alpha = 0.99$ and $\beta = 0.001$. Curves averaged over 100 independent experiments.



Figure 9: Linear Predictive Coding (LPC) of a speech signal. Simulations employ M = 14, B = 12. Fixed dithering: (a) $\delta = 1$, (b) $\delta = 0.1$, (c) $\delta = 10^{-2}$; (d) IRLS with $\alpha = 0.99$ and $\beta = 0.01$. The forgetting factor is $\lambda = 0.995$ for all cases. In the bottom plot curve (a) was removed for clarity (it presented several peaks as well).

5 CONCLUSIONS

This thesis was concerned with the study of regularization in the RLS algorithm.

In particular, an array RLS algorithm (NFRA-RLS) was developed embedded with forgetting factor and non-fading regularization of arbitrary matrix structure. The algorithm was found to exhibit superior robustness to alternative RLS algorithms for adaptive beamforming as well as to the well-known LSMI beamformer.

Moreover, a dynamically dithered RLS algorithm (IRLS) was developed based on a new criterion of general applicability referred to as the *image function*. The algorithm was shown to be capable of adjusting its level of dithering accordingly to variations of the condition number and the computational precision.

This thesis is concluded by mentioning a few challenges for future research. As far as NFRA-RLS is concerned, it would be very interesting to investigate how the imposition of some structure on the regularization matrix could lead to array algorithms of smaller computational complexity. As far as IRLS is concerned, it would be important to investigate the possibility of deriving stability bounds for the constants α and β that govern the compensation mechanism of the image function. Additionally, we pose the problem of direct minimization of the image function. Finally, the ultimate objective is to exploit the non-fading regularization structure of NFRA-RLS so that to apply the image function for the direct control of the norm of the regularization matrix, a much more effective and reliable approach in comparison to dithering.

REFERÊNCIAS BIBLIOGRÁFICAS

1 WIDROW, B.; HOFF, J. M. E. Adaptive switching circuits. *IREN WESCON Convention Record*, v. 4, p. 96–105, 1960.

2 SAYED, A. H. Adaptive Filters. [S.I.]: Wiley Inter-Science, 2008.

3 HAYKIN, S. Adaptive filter theory. Prentice-Hall, 2001.

4 BELLANGER, M. Adaptive Digital Filters and Signal Analysis. [S.1.]: Marcel Dekker Ltd., 1987.

5 WIDROW, B. Adaptive Signal Processing. [S.l.]: Prentice Hall, 1985.

6 MEYER, C. D. *Matrix Analysis and Applied Linear Algebra*. [S.I.]: Society of Industrial and Applied Mathematics (SIAM), 2000.

7 KENNEY, A. J. L. C. S.; REESE, M. S. Statistical condition estimation for linear systems. *SIAM Journal of Scientific Computing*, v. 29, n. 2, p. 566–583, mar. 1998.

8 KAILATH, T.; SAYED, A. H. Displacement structure: Theory and applications. *SIAM Review*, v. 37, n. 3, p. 297–386, set. 1995.

9 CHEN, Y.-K. C. P. Challenges and opportunities of obtaining performance from multi-core cpus and many-core gpus. In: *Proc. of ICASSP*. Taipei, Taiwan: [s.n.], 2009. p. 613–616.

10 CHIOFFI, J. M.; KAILATH, T. Fast, recursive-least-squares transversal filters for adaptive filtering. *IEEE Trans. on Acoustics, Speech and Signal Processing*, v. 32, n. 2, p. 304–337, abr. 1984.

11 BOTTOMLEY, G. E.; ALEXANDER, S. T. A novel approach for stabilizing recursive least squares filters. *IEEE Trans. Signal Process.*, v. 39, n. 8, p. 1770–1779, ago. 1991.

12 SALGADO, G. G. M.; MIDDLETON, R. Modified least squares algorithm incorporating exponential resetting and forgetting. *International Journal of Control*, v. 47, n. 2, p. 477–491, fev. 1988.

13 GAY, S. Dynamically regularized fast rls with application to echo cancellation. In: *Proc. of IEEE ICASSP*. [S.l.: s.n.], 1996. v. 2, p. 957–960.

14 HORITA K. SUMIYA, H. U.; MITSUISHI, S. A leaky rls algorithm: its optimality and implementation. *IEEE Trans. Signal Process.*, v. 52, n. 10, p. 2924–2932, out. 2004.

15 LIAVAS, P.; REGALIA, P. A. On the numerical stability and accuracy of the conventional recursive least squares algorithm. *IEEE Trans. Signal Process.*, v. 47, n. 1, p. 88–96, Jan 1999.

16 JR., J. A. QRD-RLS Adaptive Filtering. [S.I.]: Springer, 2009.

17 GUERCI, R. Theory and application of covariance matrix tapers for robust adaptive beamforming. *IEEE Trans. Signal Process.*, v. 47, n. 4, p. 977–985, abr. 1999.

18 MA, N.; GOH, J. Efficient method to determine diagonal loading value. In: *Proc. of IEEE ICASSP.* [S.l.: s.n.], 2003. v. 5, p. 341–344.

19 TSAKIRIS, M. C.; LOPES, C. G. A robust affine projection algorithm with feedback compensation of the condition number. In: *Proceedings of ISCAS*. Taipei, Taiwan: [s.n.], 2009. p. 2057–2060.

20 CARINI; SICURANZA, G. L. Optimal regularization parameter for the multichannel filtered-x affine projection algorithm. *IEEE Trans. Signal Process.*, v. 55, n. 10, p. 4882–4895, 2007.

21 REY L. R. VEGA, S. T. H.; BENESTY, J. Variable explicit regularization in affine projection algorithm: robustness issues and optimal choice. *IEEE Trans. Signal Process.*, v. 55, p. 2096–2109, 2007.

22 CHOI, S.; SONG. Affine projection algorithms with adaptive regularization matrix. In: *Proc. of IEEE ICASSP*. Toulouse, France: [s.n.], 2006. v. 3, p. 201–204.

23 AKINO, K. Optimum-weighted rls channel estimation for rapid fading mimo channels. *IEEE Trans. on Wireless Commun.*, v. 7, n. 11, p. 4248–4260, 2008.

24 GOLUB, G. H.; LOAN, C. F. V. *Matrix Computations*. [S.l.]: The Johns Hopkins University Press, 3rd edition, 1986.

25 GENTLEMAN, W. M.; KUNG, H. T. Matrix triangularization by systolic arrays. *Proc. Society of Photo-Optical Instrumentation Engineers (SPIE), Int. Soc. Opt. Eng.*, v. 298, p. 19–26, 1981.

26 GOETZE, J.; SCHWIEGELSHOHN, U. A square root and division free givens rotation for solving least squares problems on systolic arrays. *SIAM J. on Scientific and Statistical Comp.*, v. 12, n. 4, p. 297–386, jul. 1991.

27 YASUI, S. K.; NAKAMURA, T. Adaptive array antenna for w-cdma systems. *Fujitsu Sci. Tech. Journal*, v. 38, n. 2, p. 192–200, 2002.

28 ADACHI; TJHUNG, T. Tapped delay line model for band-limited multipath channel in ds-cdma mobile radio. *Electronic Letters*, v. 37, n. 5, p. 318–319, mar. 2001.

29 HONG S-H HWANG, K.-J. K. E.-K.; WHANG, K.-C. Synchronous transmission technique for the reverse link in ds-cdma terrestrial mobile systems. *IEEE Transactions on Communications*, v. 47, n. 11, p. 1632–1635, nov. 1999.

30 ARENAS-GARCIA, V. G.-V. J.; FIGUEIRAS-VIDAL, A. New algorithms for improved adaptive convex combination of lms transversal filters. *IEEE Trans. Instr. and Measurements*, v. 54, n. 6, p. 2239–2249, Dec 2005.

31 CHO, C.-H. C. N.; LEE, S. Adaptive line enhancement by using an iir lattice notch filter. *IEEE Trans. on Acoustics, Speech and Signal Processing*, v. 37, n. 4, p. 585–589, abr. 1989.

32 HARRIS, M. P. D. F.; VULETIC, D. Real-time qrd-based beamforming on an fpga platform. In: *Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers*. Pacific Grove, EUA: [s.n.], 2006. p. 1200–1204.

33 SONG, J. W.; WANG, H. Robust recursive least squares adaptive beamforming algorithm. In: *Proc. of IEEE Intern. Symp. on Commun. and Inf. Techn.* [S.l.: s.n.], 2004. p. 238–241.

34 HILGERS, J. W. A theory for optimal regularization in the finite dimensional case. *Linear Algebra and Its Applications*, v. 4, n. 2, p. 359–379, dez. 1982.

35 HANSEN, C. *Rank-Deficient and Discrete Ill-Posed Problems*. [S.l.]: Society of Industrial and Applied Mathematics (SIAM), 1998.

36 BENESTY; GAENSLER, T. A recursive estimation of the condition number in the rls algorithm. In: *Proc. of ICASSP*. [S.l.: s.n.], 2005. v. 4, p. 25–28.

37 ROMAN, S. Advanced Linear Algebra. [S.l.]: Springer Third Edition, 2008, 2008.

38 ABOULNASR, T.; MAYYAS, K. A robust variable step-size lms-type algorithm: analysis and simulations. *IEEE Trans. Signal Process.*, v. 45, n. 3, p. 631–639, mar. 1997.