INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI®

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Université d'Ottawa • University of Ottawa

On the Laguerre series expansion and transform

Stephan Quednau

A thesis submitted to the
School of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Master of Applied Science
in Electrical Engineering

Ottawa-Carleton Institute of Electrical and Computer Engineering
School of Information Technology and Engineering
Department of Electrical and Computer Engineering
Faculty of Engineering
University of Ottawa

March, 2000

Abstract

The Laguerre transform is defined and motivated based on the Laguerre series expansion and Laguerre filter.

The Laguerre transform is represented by a unitary transformation matrix of infinite size and is controlled by the Laguerre parameter. When realised as a finite size matrix, the case where the Laguerre parameter approaches zero results in the identity matrix. Otherwise the transform is less and less unitary and the inverse transform is not complete. However, we can choose the size of the matrix such that orthogonality is preserved and use the matrix transform for signal compression. We show that modeling an acoustic duct impulse response using the Laguerre transform has less mean-squared error than the truncated DCT transform for high compression ratios. We also show results using least-square solutions and adaptive filtering where the Laguerre filter gave improved results over classical methods.

Acknowledgements

The following work was carried out at the VIVA Lab, itself part of the School of Information Technology and Engineering at the University of Ottawa, Canada, during the period of May 1999 to spring 2000. The VIVA Lab was formed in 1999 to carry out research in the areas of video processing and coding; image processing and analysis; computer vision; and acoustics, audio and speech processing. It is with this last group that Dr. Martin Bouchard conducts research in areas of signal processing, adaptive filtering and neural networks, applied to speech coding, audio and acoustics.

I am indebted to some very fine individuals who have formed my interest in signal processing. I would like to thank my thesis supervisor, Dr. Martin Bouchard, for his willingness to support my subject of study and confidence in its outcome. I am grateful to Dr. Nicolas D. Georganas and the School of Information Technology and Engineering for their financial support that made this work possible. I give thanks to Dr. John Lodge at the Communications Research Center (CRC) for having given me the opportunity of working in signal processing with a distinguished team of researchers. I owe my programming skills to Mr. Richard J. Young, my immediate supervisor and mentor whilst at the CRC. I thank Dr. Stanley P. Lipshitz at the University of Waterloo for having given me the opportunity of working in digital audio at the undergraduate level. For their supportive role in the realisation of this document, I wish to thank Dr. Eric Dubois for having shared his knowledge of LaTeX and Mr. Keith White for his help with UNIX.

Final thanks are reserved for my immediate family, my wife and our rabbit, for their love and encouragement.

Stephan Quednau

Ottawa, Canada, March 2000

Contents

List of Figures

List of Tables

List of Symbols

j	$\sqrt{-1}$
t	Continuous time variable
n	Discrete-time sample index
k	Dimension of a vector space
K	Number of FIR filter taps, filter order +1
c_k	Spectrum coefficient
w_k	Weight coefficient
ξ	Laguerre parameter
θ	Discount factor
$\hat{x}$	Least-squares approximation of x
$\tilde{x}$	Laguerre transform of x
X	Fourier transform of x
$X(z)$	z-transform of x
$A(z)$	All-pass transfer function
$\boldsymbol{x}$	Column vector x
$\boldsymbol{R}$	Autocorrelation matrix

$\boldsymbol{p}$	Crosscorrelation vector
ϕ	Orthonormal function
$\Phi(z)$	Orthonormal transfer function
$\boldsymbol{\Phi}$	Orthonormal transform
ψ	Laguerre function
$\Psi(z)$	Laguerre transfer function
$\boldsymbol{\Psi}$	Laguerre transform
$\mathcal{M}$	Subspace of a vector space

List of Abbreviations

ANC	Active Noise Cancellation
AP	Allpass
CT	Continuous-Time
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DLT	Discrete Laguerre Transform
DST	Discrete Sine Transform
DT	Discrete-Time
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FTF	Fast Transversal Filter
IIR	Infinite Impulse Response
KLT	Karhunen-Loeve Transform
LMS	Least Mean-Squared Algorithm
MSE	Mean-Squared Error
RLS	Recursive Least-Squares Algorithm

STFT	Short Term Fourier Transform
TDL	Tapped Delay Line
TSR	Transaural Sound Reproduction

Chapter 1

Introduction

1.1 Background

1.1.1 Motivation 1: series expansions

Transforms or linear expansions of signals are fundamental tools in signal analysis. They allow an alternative representation of a signal and provide insight in its characteristics. And when an approximation of a signal is desired, a good series expansion allows a more compact and computationally efficient representation.

A linear series expansion of a signal x in a vector space $\mathcal{S}$, be it finite or infinite dimensional, is written as a linear combination of the set of functions $\{\phi_k\}$

$$x = \sum_k c_k \phi_k \tag{1.1}$$

If the set $\{\phi_k\}$ is orthonormal, then we can sum over less functions and obtain increasingly greater approximation error.

There exists many orthogonal functions but a fundamental series expansion is the canonical function set $\{\delta[n-k]\}$ or $\{\delta(t-nT)\}$ as these functions sample the initial function x.

For example, the z-transform decomposes a sequence $x[n]$ into a linear combination of the sample amplitudes and the canonical basis functions in $l_2(\mathbb{Z})$. The transversal filter allows

such a decomposition as it is composed of unit delays with associated weights at each delay element output. The output is thus a sum of the weighted delayed impulse functions $\delta[n-k]$.

In linear adaptive filtering, the FIR filter is a basic backbone which generates this series expansion. By definition, such a filter has finite impulse response and stability is simple to achieve. However, for long impulse responses, such a filter becomes computationally expensive especially if an adaptation process is performed.

IIR filters provide more compact representation of long FIR filters. However, these have disadvantages such as having local minima in the error surface, the risk of instability during adaptation and slow initial convergence [1]. The slow convergence stems from the lack of orthogonality amongst the internal variables in adaptive IIR filters.

This motivates the use of hybrid structures which could be implemented in an FIR-like structure. It turns out that the Laguerre filter, based on the Laguerre series expansion, is such a structure. It can be viewed as a generalisation of the tapped delay line and has a free design parameter called the Laguerre parameter. The filter can be recursively defined and is thus computationally efficient. When the Laguerre parameter approaches 0, the Laguerre filter degenerates into a classical tapped delay line. For a non-zero parameter value, the impulse responses seen at each filter tap are the orthonormal Laguerre sequences. These become the functions used in the Laguerre series expansion. We will show that, under certain conditions, these provide a more compact signal representation.

Music box paradigm

To illustrate this first motivation, consider the operation of a music box. A rotating cylinder is fixed to a plate. A thin comb-like metal sheet it also fixed to this plate such that the comb edges, called tines, are perpendicular to cylinder surface. Now the cylinder is circularly banded into tracks where pegs denote the activity of the facing tine. The tines may be replaced by bells to yield a *Glockenspiel.*

In signal processing terms this is a signal synthesiser where the tines represent the orthogonal functions and the pegs the Laguerre coefficients. The pegs on the cylinder are an efficient representation as they simply serve to fire the correct tine at the correct time. Time is encoded in both the resonance length of each tine and the rotational speed of the cylinder. If the cylinder stops rotating, the tines continue to resonate for a little while. The amount

of information is determined by the number and characteristics of the tines. If we limit the music piece to a major or minor key, we can avoid using all the 13 semitones in an octave and thus require less than 13 tines per octave.

Compare this system to the omnipresent sampled signal system which is intricately tied to the canonical functions by sampling. Here, to portray a music box, the number of tines is determined by the length or time axis of the entire music sequence and the tines simply indicates the amplitude at that instance. Time is only encoded in the tine firing rate (e.g. the sampling rate). If rotation is halted, the system holds the last amplitude forever (e.g. after the sample-and-hold). Again, the amount of information here depends on the number of tines (samples) and the quantisation of the tine amplitudes. However, this system requires samples proportional to the length of the music piece and not proportional to its frequency range.

1.1.2 Motivation 2: frequency warping

A second motivation to using the Laguerre filter is psychoacoustically motivated. There is evidence that the human hearing and visual systems process information in the logarithmic domain in both amplitude and frequency for a more efficient signal representation [2] [3] [4]. Both the visual and auditory system employ variable resolution such that certain received signals are detected more acutely than others.

In the frequency domain this amounts to dividing the frequency scale using a filter bank such that each filter's bandwidth is proportional to its center frequency. For acoustic signal processing work, the ear is the final receptor and the processing of such a signal should optimally be done to match this receiver. The inner ear has been shown to analyse sound energy non-uniformly across time and frequency. Such a controlled distortion is called warping and a Laguerre filter will realise it.

Signal warping and nonuniform sampling are related issues as they both address the possibility of optimum signal representation for certain receptors.

1.2 Present contribution

In this work we define the Laguerre transform matrix and show examples for its use in lossy signal compression. This contribution is presented in a unified framework to help understand the Laguerre series expansion and Laguerre filter.

The Laguerre transform is represented by a unitary transformation matrix of infinite size and is controlled by the Laguerre parameter ξ. When the Laguerre transform is realised with a finite size matrix, the case $\xi \rightarrow 0$ results in the identity matrix. For $|\xi| > 0$, the unitary property of the transform is weakened and the inverse transform is not complete. However, we can choose the size of the matrix such that orthogonality is preserved and use the matrix transform in signal compression. The resulting compressed signals have non-uniform resolution, with resolution becoming coarser with the length of the signal. We show that modeling an acoustic duct impulse response using the Laguerre transform has less mean-squared error than the truncated DCT transform for high compression ratios.

The motivation of the Laguerre transform stems from results obtained with the Laguerre filter. The Laguerre filter has been used in system identification problems to obtain a more compact model of the plant. This is of interest in modeling acoustic impulse responses which have typically long tails. In this work, we also show system identification examples, both in least-square solutions and adaptive filtering, where the use of Laguerre filter reduced the mean-square error, reduced the number of filter coefficients, and shaped the error frequency response.

1.3 Outline of the thesis

Before beginning, I urge the reader to take a look at the list of symbols as I have tried to strictly adhere to the notation. In summary, time is always denoted by t or n, and dimension is indexed by k. This avoids the confusion when presenting both vector spaces and adaptive filtering using the variable n in both cases.

We begin in Chapter 2 by covering the fundamentals of signal decomposition. We introduce the vector space, its inner product and then formally define a Hilbert space. The projection theorem is presented and its implications for the least-squares error is shown.

In Chapter 3 we present orthogonal polynomials which include the Laguerre polynomials and better known ultraspherical polynomials such as the Legendre, Taylor and Chebychev polynomials. This chapter prepares the reader for the notation used in presenting the Laguerre polynomials and functions.

I have divided the Laguerre polynomials in their continuous and discrete-time versions in Chapter 4 and 5, respectively, in order to maintain the notation used in Chapter 3 on orthogonal polynomials. The progression was found to aid in understanding the Laguerre sequences which, once transformed, define the Laguerre filter.

Chapter 4 begins by defining the generalised Laguerre polynomials to mimic the generalised approach in Chapter 3. We then present the Laguerre polynomials and orthonormal functions. Properties are presented but not in great detail as this will be done in Chapter 5.

Thus Chapter 5 presents the discrete Laguerre sequences. Their generalised versions are not realisable and detract from the material. The reader is thus referred to the cited work. We continue with a derivation of the Laguerre filter as given by [5]. Properties conclude this chapter.

Chapter 6 covers matrices and transforms in discrete-time. Many common transforms, such as the DFT or DCT, are simpler to understand in matrix form. Thus we begin the chapter with matrix properties. We then show that matrix transforms can be written in terms of a synthesis and analysis filter bank. Some common transforms are presented and we motivate the existence of a Laguerre transform with definitions. We complete the chapter with examples of signal compression using the Laguerre transform.

Chapter 7 presents an alternative view of the Laguerre filter where it is used to obtain a non-equally spaced frequency representation. This is also referred to as *frequency warping*. As a main component in the Laguerre filter, the allpass filter and its properties are presented. An example of frequency warping is then given. The dual to frequency warping is time warping or non-uniform signal sampling. We motivate both by defining matrix notation.

Chapter 8 reports least-squares solutions using Laguerre filters. Properties of the mean-squared error surface are discussed.

Adaptive signal processing results are reported in Chapter 9. Important results of previous work are cited. Results using Laguerre LMS and RLS algorithms are presented for a system identification problem.

We conclude in Chapter 10 with the main aspects of the Laguerre series expansion. transform and adaptive filtering results.

The Appendix includes complementary work and definitions used in the text.

Appendix A presents implementations of the allpass filter chain which composes the main structure of a Laguerre filter. These results emphasise the recursive nature of the filter. The computation complexity is discussed and compared to the classical FIR filter.

Appendix B includes the c source code for the allpass chain and the Laguerre filter. The original allpass chain code is credited to Härmä [6] and was corrected and modified to yield a Laguerre filter.

Appendix C presents functions with properties which are used in the text.

Appendix D presents notation and system structure for the system identification and system inverse problems.

Chapter 2

Fundamentals of signal decomposition

Signal decomposition is a structured method of decomposing a signal into orthogonal components which can be linearly combined without affecting each other. The space in which this quality holds is a Hilbert space. In this chapter we will define the objects necessary to define a Hilbert space, such as the concept of a vector space and its inner product. We also address projection and the successive approximation property. The aim of the chapter is to delineate the terms which will be used throughout this work and thus clarify the material.

2.1 Notation

Let $\mathbb{C}, \mathbb{R}, \mathbb{Z}$ and $\mathbb{N}$ denote the sets of complex, real, integer and natural numbers, respectively. Sets of k-tuples $(x_1, \ldots, x_k)$ are denoted with the previous symbols with exponent k.

The superscript * denotes complex conjugation, as in $(a+jb)^* = (a-jb)$ where $j = \sqrt{-1}$ and $a, b \in \mathbb{R}$.

We distinguish between the continuous-time (CT) and discrete-time (DT) notation by using parenthesis and square brackets, respectively, with the following equivalence

$$x[n] = f(t)|_{t=nT} \qquad n \in \mathbb{Z}, T \in \mathbb{R}_+ \tag{2.1}$$

We will denote $f(t)$ a function and $x[n]$ a sequence. In particular $\delta(t)$ denotes the Dirac function, a generalised function defined as a limit of a finite energy function $\varphi(t)$

$$\delta(t) = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \varphi(\frac{t}{\epsilon}), \tag{2.2}$$

and $\delta[n]$ the Kronecker delta sequence which is 1 for $n = 0$ and 0 otherwise.

Other notation will be introduced as required and can be inspected in the List of Symbols.

2.2 Hilbert spaces

The two dimensional vector spaces $\mathbb{R}^2$ and $\mathbb{Z}^2$ are familiar finite dimensional Hilbert spaces as they can be expressed by a Cartesian plane. The infinite dimensional Hilbert spaces $L_2(\mathbb{R})$ and $l_2(\mathbb{Z})$ are less familiar but are fundamental to signal processing. Key aspects such as linear independence, orthonormal bases, completeness, projection and least-squares approximation have to be delineated. Fortunately finite and infinite vector spaces differ only in their proof of completeness and thus we can start with the finite dimensional case and progress to the infinite dimensional case.

Legitimate questions that will guide the reader to the notion of a Hilbert space are:

1. Given a set of linearly independent vectors, do these span the finite k-dimensional space $\mathbb{C}^k$, meaning that every vector in $\mathbb{C}^k$ can be written as a linear combination of these?

2. Are the vectors linearly independent, or can some vectors be written as a linear combination of others?

3. What is a basis, and why should it also be orthogonal?

4. If we remove some spanning vectors in the basis, what are the properties of the best approximation obtainable and what would an orthonormal basis give us?

2.2.1 Vector spaces and inner products

We begin with a formal definition of a vector space.

Vector space

Definition 2.1 *A vector space over the set of complex numbers $\mathbb{C}$ is a set $\mathcal{E}$, together with two operations,*

1. *Vector addition $+ : \mathcal{S} \times \mathcal{S} \rightarrow \mathcal{S}$*
2. *Scalar multiplication $\cdot : \mathbb{C} \times \mathcal{S} \rightarrow \mathcal{S}$*

for which the following properties hold with $x, y, z \in \mathcal{E}, a, b \in \mathbb{C}$:

1. *Commutativity over addition: $x + y = y + x$*
2. *Associativity over addition: $(x + y) + z = x + (y + z)$*
3. *Additive identity: $\exists 0 \in E$, such that $x + 0 = x$*
4. *Additive inverse: $\exists -x \in E$, such that $x + (-x) = 0$*
5. *Commutativity over multiplication: $a \cdot (b \cdot x) = b \cdot (a \cdot x)$*
6. *Multiplicative identity: $1 \cdot x = x$*
7. *Multiplicative zero: $0 \cdot x = 0$*
8. *Distributivity over multiplication: $a \cdot (x + y) = a \cdot x + a \cdot y$*
9. *Distributivity over addition: $(a + b) \cdot x = a \cdot x + b \cdot x$*

Example 2.1 *A familiar example is the set $\mathcal{S}$ of all k-tuples*

$$\mathcal{S} = \{(x_0, x_1, \ldots, x_{k-1}) | x_i \in \mathbb{C}\} = \mathbb{C}^k. \tag{2.3}$$

Example 2.2 *The set $\mathcal{S}$ of infinite dimensional square-summable vectors form the vector space $l_2(\mathbb{Z})$*

$$S = \{(x_0 x_1 \ldots) | x_i \in \mathbb{C}, \sum_{i=0}^{\infty} |x_i|^2 < \infty\}. \tag{2.4}$$

Example 2.3 *The set of real numbers* $\{x\} \in \mathbb{R}$ *forms a vector space. In fact, here the elements are both scalars and vectors [7].*

Example 2.4 *The set of all matrices of a given size forms a vector space [7].*

Example 2.5 *The set of all random variables defined on the same sample space forms a vector space.*

Subspace

A subset $\mathcal{M}$ of $\mathcal{E}$, $\mathcal{M} \subset \mathcal{E}$, is called a subspace of $\mathcal{E}$ if we can perform the two defining operations of a vector space in it:

1. Vector addition: for all $x, y \in \mathcal{M}, x + y$ is also in $\mathcal{M}$.
2. Scalar multiplication: for all $x \in \mathcal{M}$ and all $a \in \mathbb{C}, ax$ is also in $\mathcal{M}$.

Since $a = 0 \in \mathbb{C}$, a subspace includes the origin.

Spanning of a subspace

Now suppose we collect k vectors $\mathcal{S} = \{x_1, x_2, \ldots, x_k\} \in \mathcal{E}$. Then the span of $\mathcal{S}$ is a subspace of $\mathcal{E}$ generated by all linear combinations of vectors in $\mathcal{S}$

$$span(\mathcal{S}) = \{\sum_{i=1}^{k} c_i x_i | c_i \in \mathbb{C}\} \tag{2.5}$$

Linear independence

Vectors $x_1 \ldots x_k$ are linearly independent if the only way to get a linear combination equal to zero is if all weights $\{c_i\}$ are zero, i.e., $\sum_{i=1}^{k} c_i x_i = 0$ iff $c_i = 0, \forall i$.

Basis and dimension

A subset $\{x_1 \ldots x_k\}$ of a vector space $\mathcal{E}$ is called a basis for $\mathcal{E}$ if $\{x_1 \ldots x_k\}$ are linearly independent and $\mathcal{E} = span(x_1 \ldots x_k)$. The vector space $\mathcal{S}$ than has *dimension* k. Note that if we supply one extra vector x_{k+1} to the subset which is already a basis, then the new subset still spans $\mathcal{E}$, but the vectors are no longer linearly independent.

Example 2.6 *The space of infinite square-summable sequences $l_2(\mathbb{Z})$ is spanned by the infinite dimensional set $\{\delta[n-k]\}, k \in \mathbb{Z}$. Since they are linearly independent, the set is a basis and has infinite dimension.*

Inner product

We now define the inner product of two vectors.

Definition 2.2 *The inner product on a vector space $\mathcal{E}$ over $\mathbb{C}$ is a mapping of two vectors to a scalar*

$$\langle \cdot, \cdot \rangle : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{C} \tag{2.6}$$

for which the following properties hold:

1. $\langle x, y \rangle = \langle y, x \rangle^*$
2. $\langle ax + by, z \rangle = a\langle x, z \rangle + b\langle y, z \rangle$
3. $\langle x, x \rangle \geq 0$, *with* $\langle x, x \rangle = 0$, *if and only if* $x \equiv 0$

The inner product then allows a measure of length (norm) and distance (metric) to be defined. The norm squared of a vector x is denoted as $\|x\|^2 = \langle x, x \rangle$ and the metric of vectors x_i, x_j as $\|x_i - x_j\|^2$.

The inner product satisfies the Cauchy-Schwartz inequality

$$|\langle x, y \rangle|^2 \leq \langle x, x \rangle \langle y, y \rangle \tag{2.7}$$

with equality when x, y are colinear, $x = ay, a \in \mathbb{C}$.

Two vectors are said to be orthogonal if their inner product is zero $\langle x_i, x_j \rangle = 0$. Furthermore, if the vectors are normalised to have unit norm, they form an orthonormal system if

$$\langle x_i, x_j \rangle = \delta_{ij} \tag{2.8}$$

2.2.2 Hilbert spaces

A vector space associated with a suitable inner product is called an *inner product space.* We can perform geometrical operations in such a space [7]. In order to obtain a Hilbert space, we further need to define the notion of completeness. Completeness implies that every vector $y \in \mathcal{E}$ can be expressed as linear combination of an orthonormal basis for $\mathcal{E}$. This notion will become important when we remove some of the basis vectors and try to approximate any vector with this set.

Complete vector space

First define convergence as the existence of a vector sequence $\{x_k\}$ if $\|x_k - x\| \rightarrow 0$ for $k \rightarrow \infty$ and some $x \in \mathcal{E}$. A Cauchy sequence is then a sequence of vectors $\{x_k\}$ in $\mathcal{E}$ such that $\|x_k - x_m\| \rightarrow 0$ for $k, m \rightarrow \infty$. If every Cauchy sequence converges in $\mathcal{E}$, then $\mathcal{E}$ is said to be complete.

Hilbert spaces

We can then define a Hilbert space.

Definition 2.3 *A complete inner product space is a Hilbert space.*

Furthermore, a Hilbert space is separable if and only if it has a countable orthonormal basis. Examples of Hilbert spaces are given below.

Complex/real spaces

The complex space $\mathbb{C}^k$ is the set of all k-tuples $(x_1, \ldots, x_k)$ with finite length $\|x_i\|^2 \in \mathbb{C}$. The inner product is given as

$$\langle x, y \rangle = \sum_{i=1}^{k} x_i y_i^* \tag{2.9}$$

and norm is

$$\|x\| = \sqrt{\langle x, x \rangle} = \sqrt{\sum_{i=1}^{k} |x_i|^2}. \tag{2.10}$$

Space of square-summable functions

Functions $x(t)$ defined on $\mathbb{R}$ are in the Hilbert space $L_2(\mathbb{R})$ if $|x(t)|^2$ is integrable

$$\sqrt{\int_{t \in \mathbb{R}} |x(t)|^2 \, dt} < \infty \tag{2.11}$$

This is also seen as x having finite energy.

The inner product is given by

$$\langle x, y \rangle = \int_{t \in \mathbb{R}} x(t) y^*(t) \, dt \tag{2.12}$$

and the norm is

$$\|x\| = \sqrt{\langle x, x \rangle} = \sqrt{\int_{t \in \mathbb{R}} |x(t)|^2 \, dt}. \tag{2.13}$$

This space is infinite dimensional.

Space of square-summable sequences

The complex sequences $x[n], n \in \mathbb{Z}$ of finite energy are vectors in the Hilbert space $l_2(\mathbb{Z})$. Here the inner product is

$$\langle x, y \rangle = \sum_{n=-\infty}^{\infty} x[n] y^*[n] \tag{2.14}$$

and the norm is

$$\|x\| = \sqrt{\langle x, x \rangle} = \sqrt{\sum_{n \in \mathbb{Z}} |x[n]|^2}. \tag{2.15}$$

These sequences are almost always assumed in DT signal processing. $l_2(\mathbb{Z})$ is an infinite dimensional space and all sequences in it have finite norm. A possible orthonormal basis for it is $\{\delta[n-k]\}, k \in \mathbb{Z}$.

Space of square-summable functions on $[-\pi, \pi]$

The harmonic sine and cosine series $\{\sin(nt), \cos(nt)\}, n \in \mathbf{N}$ form a complete, orthogonal vector space for square-summable functions on the interval $[-\pi, \pi]$ and, in fact, form a basis for $L_2([-\pi, \pi])$. To show this [8] we write every function $f(t) \in L_2([-\pi, \pi])$ as the sum of its even and odd components

$$f(t) = f_e(t) + f_o(t) \tag{2.16}$$

with $f_e(t) \in \mathcal{E}$, $f_o(t) \in \mathcal{O}$, and

$$f_e(t) = f(t) + f(-t) \tag{2.17}$$

$$f_o(t) = f(t) - f(-t) \tag{2.18}$$

and use the trigonometric Fourier series to express each component

$$f_e(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos(nt) \tag{2.19}$$

$$f_o(t) = \sum_{n=1}^{\infty} b_n \sin(nt) \tag{2.20}$$

where the Fourier coefficients are given as

$$a_0 = \frac{2}{2\pi} \int_0^{\pi} f_e(t)\, dt \tag{2.21}$$

$$a_n = \frac{2}{2\pi} \int_0^{\pi} f_e(t) \cos(nt)\, dt \tag{2.22}$$

$$b_n = \frac{2}{2\pi}\int_0^{\pi} f_o(t)\sin(nt)\,dt \tag{2.23}$$

Note that for $t \in [-\pi, \pi]$,

1. $\langle \cos(mt), \cos(nt) \rangle = \pi\delta_{mn} \quad n, m \in \mathbf{N}$
2. $\langle \sin(mt), \sin(nt) \rangle = \pi\delta_{mn} \quad n, m \in \mathbf{N}$
3. $\langle \cos(mt), \sin(nt) \rangle = 0 \quad \forall n, m \in \mathbf{N}.$

Then an orthonormal basis for $\mathcal{E}$ is $\{\frac{1}{\sqrt{2\pi}}, \frac{\cos(nt)}{\sqrt{\pi}}\}$ and one for $\mathcal{O}$ is $\{\frac{\sin(nt)}{\sqrt{\pi}}\}$ and since the vector spaces $\mathcal{E}$ and $\mathcal{O}$ are orthogonal, their direct sum spans $L_2([-\pi, \pi])$, $\mathcal{E} \oplus \mathcal{O} = L_2([-\pi, \pi])$, with an orthonormal basis [8]

$$\{\frac{1}{\sqrt{2\pi}}, \frac{\cos(nt)}{\sqrt{\pi}}, \frac{\sin(nt)}{\sqrt{\pi}}\}. \tag{2.24}$$

The inner product here is given by

$$\langle x, y \rangle = \frac{1}{2\pi}\int_{-\pi}^{\pi} x(t)y^*(t)\,dt \tag{2.25}$$

2.3 Orthonormal bases

2.3.1 Gram-Schmidt orthogonalisation

Given a set of linear independent vectors $\{x_k\}$ in $\mathcal{E}$, we can construct an orthonormal set $\{y_k\}$ with the same span as $\{x_k\}$ using the Gram-Schmidt orthogonalisation procedure. Starting from any normalised vector in $\mathcal{E}$, we recursively construct an orthonormal vector to the space spanned by the previously orthogonalised vectors. This is done by *projecting* the candidate vector onto this space and normalising the difference between the candidate vector and its projection.

Definition 2.4 *Given the k-dimensional set of linearly independent vectors $\{x_1 x_2 \dots x_k\}$, we construct a orthonormal set of vectors $\{y_1 y_2 \dots y_k\}$, by the Gram-Schmidt orthogonalisation procedure:*

$$y_1 = \frac{x_1}{\|x_1\|} \tag{2.26}$$

$$y_j = \frac{v_j}{\|v_j\|} \qquad j > 2. \tag{2.27}$$

where

$$v_j = x_j - \sum_{i=0}^{j-1} \langle y_i, x_j \rangle y_i. \tag{2.28}$$

2.3.2 Series expansion

We can now show how to express a vector from a Hilbert space using its basis. For a set of vectors $\{x_k\}$ to be an orthonormal basis, the set of vectors has to be orthonormal and complete. In other words we have an orthonormal basis $\{x_k\}$ for $\mathcal{E}$, if for every $y \in \mathcal{E}$, we can express y as a linear combination of an orthonormal basis

$$y = \sum_k c_k x_k \tag{2.29}$$

where $\{c_k\}$ are the series expansion coefficients of y and are given by the projection of y onto each basis vector

$$c_k = \langle x_k, y \rangle. \tag{2.30}$$

For an orthonormal set of vectors $\{x_k\}$, Bessel's inequality holds

$$\|y\|^2 \geq \sum_k |\langle x_k, y \rangle|^2. \tag{2.31}$$

Equality is reached if we have a complete orthonormal set. Then we have an orthonormal basis for $\mathcal{E}$ and the relation is called Parseval's equality [8].

In a finite dimensional space, like $\mathbb{R}^k$, an orthonormal set size k is also complete since k linearly independent vectors span $\mathbb{R}^k$. For an infinite dimensional space, it is not sufficient to have an infinite number of orthogonal functions and completeness has to be proved [8].

2.3.3 Projection and least-squares approximation

Suppose we wish to express a vector y from a Hilbert space $\mathcal{S}$ as a linear combination of the first K orthonormal basis vectors

$$\hat{x} = \sum_{k=1}^{K} \langle x_k, y \rangle x_k. \tag{2.32}$$

Recall that for a separable vector space $\mathcal{S}$ we can create a subspace $\mathcal{M}$ spanned by the first K basis vectors of $\mathcal{S}$ and these K vectors form an orthogonal basis for $\mathcal{M}$.

2.3.4 Projection theorem

A key concept tied to orthonormal vectors, and in fact Hilbert spaces, is projection. We begin with a definition of projection.

Definition 2.5 *In a Hilbert space $\mathcal{S}$, the projection of vector $y \in \mathcal{S}$ on a subspace $\mathcal{M}$ is a vector $\hat{x} \in \mathcal{M}$ which minimises the error $y - \hat{x}$ in the mean-squared sense of the Hilbert space norm*

$$\|y - \hat{x}\|^2 \leq \|y - x\|^2 \qquad \forall x \in \mathcal{M} \tag{2.33}$$

This key concept can be visualised with $\mathcal{S} = \mathbb{R}^2$ and subspace $\mathcal{M}$ of dimension one as in Fig. 2.1. Of all the points x in $\mathcal{M}$, the one that is closest to y is $\hat{x}$ which is just the projection of y onto $\mathcal{M}$.

If we define the error $d = y - \hat{x}$, then d is orthogonal to $\mathcal{M}$ since d cannot be spanned by $\mathcal{M}$. We show that $\hat{x}$ is the best approximation of y in the mean-squared sense by the

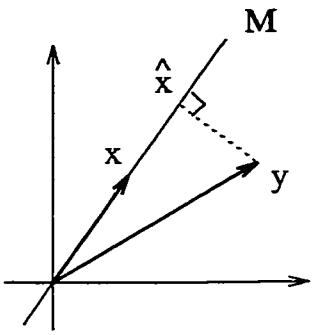


Figure 2.1: Projection $y \in \mathbb{R}^2$ on a subspace $\mathcal{M}$ of dimension one.

proof of the *projection theorem.* The proof is in two parts and is shown as it illustrates this optimality condition.

Theorem 2.1 *$\hat{x}$ is the projection of y on a subspace $\mathcal{M}$ if and only if $\langle y-\hat{x}, x\rangle = 0, \forall x \in \mathcal{M}$*

Proof 2.1 *Part 1: Suppose that $\hat{x} \in \mathcal{M}$ such that $y - \hat{x}$ is orthogonal to any x*

$$\langle y - \hat{x}, x\rangle = 0 \qquad \forall x \in \mathcal{M}. \tag{2.34}$$

Then for any arbitrary element $x \in \mathcal{M}$,

$$\|y - x\|^2 = \|(y - \hat{x}) - (x - \hat{x})\|^2 \tag{2.35}$$

$$= \langle y - \hat{x}, y - \hat{x}\rangle - \underbrace{\langle y - \hat{x}, x - \hat{x}\rangle}_{=0} - \underbrace{\langle x - \hat{x}, y - \hat{x}\rangle}_{=0} + \langle x - \hat{x}, x - \hat{x}\rangle \text{ by (2.34)} \tag{2.36}$$

$$= \|y - \hat{x}\|^2 + \underbrace{\|x - \hat{x}\|^2}_{\geq 0} \tag{2.37}$$

which implies that

$$\|y - x\|^2 \geq \|y - \hat{x}\|^2 \tag{2.38}$$

with equality occurring if $\|x - \hat{x}\|^2 = 0$. *Therefore* $\hat{x}$ *is the projection of* y *on* $\mathcal{M}$. □

Part 2:

Let $\hat{x}$ *be the projection of* y *on* $\mathcal{M}$ *and we want to show that*

$$\langle y - \hat{x}, x \rangle = 0 \qquad \forall x \in \mathcal{M} \tag{2.39}$$

Let $\check{x}$ *be expressed as a series expansion of* K *orthonormal basis vectors*

$$\check{x} = \sum_{i=1}^{K} \langle y, u_i \rangle u_i \tag{2.40}$$

where $\{u_1, u_2, \ldots, u_K\}$ *is an orthonormal basis for* $\mathcal{M}$. *Then the inner product of the error with each basis vector is*

$$\langle y - \check{x}, u_j \rangle \tag{2.41}$$

$$= \langle y, u_j \rangle - \sum_{i=1}^{K} \langle y, u_i \rangle \underbrace{\langle u_i, u_j \rangle}_{\delta_{ij}} \tag{2.42}$$

$$= \langle y, u_j \rangle - \langle y, u_j \rangle \tag{2.43}$$

$$= 0 \tag{2.44}$$

Therefore

$$\langle y - \check{x}, x \rangle = 0 \qquad \forall x \in \mathcal{M} \tag{2.45}$$

and by part 1, $\hat{x} = \check{x}$ *since it is unique.* □

Now given $\hat{x}^{(k)}$, the best approximation using the first k orthonormal vectors $\{x_1, \ldots, x_k\}$, then the approximation $\hat{x}^{(k+1)}$ becomes, by (2.32)

$$\hat{x}^{(k+1)} = \hat{x}^{(k)} + \langle x_{k+1}, y \rangle x_{k+1} \tag{2.46}$$

Thus the previously calculated k coefficients do not change. This property is called the *successive approximation* property of orthonormal expansions.

2.4 Chapter summary

In this chapter we have seen that a Hilbert space is a complete inner product space. The vectors that span it can have finite or infinite dimensions but require a finite norm. We have seen that an orthonormal expansion has the quality of successive approximation. This means that the minimum mean-squared error is attained for every linear combination of some of the basis vectors. We have also seen that the notions of norm and orthogonality can be extended to functions which are just orthogonal polynomials on a certain interval. The Chebychev, Legendre and Laguerre orthogonal polynomials belong to the class of classical orthogonal polynomials. It is this class of polynomials which is presented next.

Chapter 3

Orthogonal polynomials

The Laguerre functions are obtained by orthogonalising the Laguerre polynomials which, in turn, are derived from a set of functions or vectors. To familiarise the reader with these polynomials, a formal definition and examples of classical orthogonal polynomials are given.

3.1 Function approximation

Orthogonal polynomials are used in approximation of functions. Suppose we wish to approximate e^x. The Taylor series expansion about $x_0 = 0$ of e^x is an infinite sum

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = c_0 + c_1 x + c_2 x^2 + \dots \tag{3.1}$$

and we usually calculate only the first K terms. Now the set of functions $\{x^k\}, k = 0, 1, \dots$ are linearly independent on $\mathbb{R}_+$. These become the Legendre polynomials and are obtained by orthogonalising $\{x^k\}$ using the Gram-Schmidt orthogonalisation procedure. Once they are orthogonal, the coefficients $\{c_k\}$ do not depend on K by the successive approximation property of orthogonal expansions. We can thus simply truncate the series and obtain a monotonically decreasing error as $K \to \infty$.

3.2 Orthogonal polynomials

We begin with a formal definition of orthogonal polynomials [9].

Definition 3.1 *Let the set of orthogonal polynomials of degree k in x, $\{\pi_k(x)\}$, be orthonormal under a positive integrable window function $w(x)$ on the interval $[a, b]$*

$$\int_a^b w(x)\pi_k(x)\pi_l(x)dx = C_k\delta_{kl}. \tag{3.2}$$

Then a series expansion of $f(x)$, written as

$$f(x) = \sum_{k=0}^{\infty} c_k\pi_k(x) \tag{3.3}$$

with coefficients

$$c_k = \frac{1}{C_k}\int_a^b w(x)f(x)\pi_k(x)dx \tag{3.4}$$

minimises the norm-2 error

$$\int_a^b w(x)(f(x) - f_K(x))^2dx \tag{3.5}$$

where $f_K(x)$ is the K-th order series expansion of $f(x)$.

The window function plays a key role in orthogonal polynomials. In fact, the window determines the linear independence of the orthogonal polynomials, whose span includes all the powers of $x^k, k \in \mathbf{N}$. The orthogonal polynomials form a complete set as the following theorem states [9].

Theorem 3.1 *Let $\{\phi_k(x)\}$ be a sequence of orthogonal polynomials determined by the positive integrable weight function $w(x)$. Assume they are orthogonal over the finite interval $[a, b]$*

$$\int_a^b w(x)\phi_k(x)\phi_l(x)\,dx = C_k\delta_{kl} \tag{3.6}$$

where C_k is a normalisation constant. Then $\{\phi_k(x)\}$ is complete with respect to the continuous functions over $[a, b]$.

It can be shown that, for $w(x) > 0, x \in [a, b]$, the polynomial $\pi_k(x)$ has k real zeros in $[a, b]$. Thus the polynomial oscillates between $[a, b]$ [9]. This behaviour is used in approximating functions and characterises orthogonal polynomials in general. We will present some specific orthogonal polynomials following an overview of classical orthogonal polynomials.

3.3 Classical orthogonal polynomials

The classical orthogonal polynomials can be classified as follows [10]:

1. The Jacobi polynomials, $p_k^{(\alpha,\beta)}(x)$ with $a = -1, b = 1, w(x) = (1-x)^\alpha(1-x)^\beta, \alpha, \beta > -1$. For $\alpha = \beta$, we have the ultraspherical polynomials, amongst which we mention
   (a) The Chebychev polynomials of the first kind, $t_k(x) = \cos(k\theta), x = \cos(\theta), \alpha = \beta = -1/2$;
   (b) The Chebychev polynomials of the second kind, $u_k(x) = \frac{\sin(k+1)\theta}{\sin(\theta)}, x = \cos(\theta), \alpha = \beta = 1/2$;
   (c) The Legendre polynomials, $p_k(x), \alpha = \beta = 0$.
2. The Laguerre polynomials, $l_k^{(\alpha)}$, with $a = 0, b = \infty, w(x) = x^\alpha e^{-x}, \alpha > -1$;
3. The Hermite polynomials, $h_k(x)$, with $a = -\infty, b = \infty, w(x) = e^{-x^2}$.

3.4 Ultraspherical polynomials

The Taylor, Legendre and Chebychev orthogonal polynomials belong to the more general class of orthogonal polynomials called the ultraspherical polynomials [9]

$$p_k^{(\alpha)}(x) = C_k w^{-1}(x) \frac{d^k}{dx^k}(1-x^2)^{k+\alpha} \tag{3.7}$$

where $w(x) = (1-x^2)^\alpha$ and $\alpha \in [-1, \infty)$ the *order of generalisation.* They are orthogonal under this window function $w(x)$ in the interval $[-1, 1]$.

It can be shown that the amplitude of the oscillations of $\pi_k(x)$ in $[-1, 1]$ for $-\frac{1}{2} < \alpha < \infty$ increases from $x = 0$ to $x = \pm 1$ whereas the amplitudes decrease for $-1 \leq \alpha < -\frac{1}{2}$. The

case for equiripple, $\alpha = -\frac{1}{2}$, are the Chebychev polynomials. We present these and two other ultraspherical polynomials next.

Note that a monic polynomial is a polynomial of degree k, $\sum_{i=0}^{k} c_i x^i$, where $c_k = 1$.

3.4.1 Legendre polynomials

For $\alpha = 0$ the Legendre polynomials are obtained with $w(x) = 1$ and $C_k = \frac{(-1)^k}{2^k k!}$:

$$p_k(x) = \frac{(-1)^k}{2^k k!} \frac{d^k}{dx^k}(x^2 - 1)^k \tag{3.8}$$

The Legendre polynomials are orthogonal in the interval $[-1, 1]$

$$\int_{-1}^{1} p_k(x) p_l(x) dx = \frac{2}{2k+1} \delta_{kl}. \tag{3.9}$$

and satisfy the recurrent relation [9]

$$p_{k+1}(x) = (\frac{2k+1}{k+1}) x p_k(x) - (\frac{k}{k+1}) p_{k-1}(x) \tag{3.10}$$

A standard way to obtain the Legendre polynomials is to apply the Gram-Schmidt orthogonalisation procedure to the set of linearly independent vectors $\{x^k\}, k \in \mathbb{N}$ on the interval $[-1, 1]$ [8]. The first 4 Legendre functions are

$$p_0(x) = 1 \tag{3.11}$$

$$p_1(x) = x \tag{3.12}$$

$$p_2(x) = \frac{3}{2} x^2 - \frac{1}{2} \tag{3.13}$$

$$p_3(x) = \frac{5}{2}x^3 - \frac{3}{2}x \tag{3.14}$$

The Legendre polynomials have the property that, once they are multiplied by c_k such that x_k in unity, they have the smallest norm-2 of all monic polynomials over $[-1, 1]$ [9].

The first four Legendre polynomials are depicted in Fig. 3.1 (a).

3.4.2 Taylor series

For the limit case where $\alpha \longrightarrow \infty$ we have the Taylor series about x_0 with

$$p_k(x) = (x - x_0)^k \tag{3.15}$$

with expansion coefficients

$$c_k(x) = \frac{1}{k!}\frac{d^k}{dx^k} f(x)|_{x=x_0} \tag{3.16}$$

It can be shown that the Taylor polynomials yield the worst infinite norm for $f(x)$ [9] as the approximation represents extrapolation of $f(x)$ about $x = x_0$.

3.4.3 Chebychev polynomials

The Chebychev polynomials of the first kind are obtained by setting $\alpha = -\frac{1}{2}, C_k = \frac{(-1)^k 2^k k!}{(2k)!}$ [9]. They are given in their simplest form as

$$t_k(x) = cos(k\theta), \qquad x = cos(\theta) \tag{3.17}$$

and can be constructed from the recursive relations

$$t_{k+1}(x) - 2xt_k(x) + t_{k-1}(x) = 0 \tag{3.18}$$

with

$$t_0(x) = 1 \tag{3.19}$$

$$t_1(x) = x \tag{3.20}$$

$$t_2(x) = 2x^2 - 1 \tag{3.21}$$

Thus $t_3(x) = 4x^3 - 3x$.

Chebychev's theorem states that, of all monic functions on $[-1, 1]$, the Chebychev polynomials, $t_k(x)$, have the smallest infinite norm of all monic polynomials

$$||2^{-(k-1)}t_k(x)||^\infty = 2^{-(k-1)} \tag{3.22}$$

with

$$||f||^\infty = \sup_{-1 \le x \le 1} |f(x)| \tag{3.23}$$

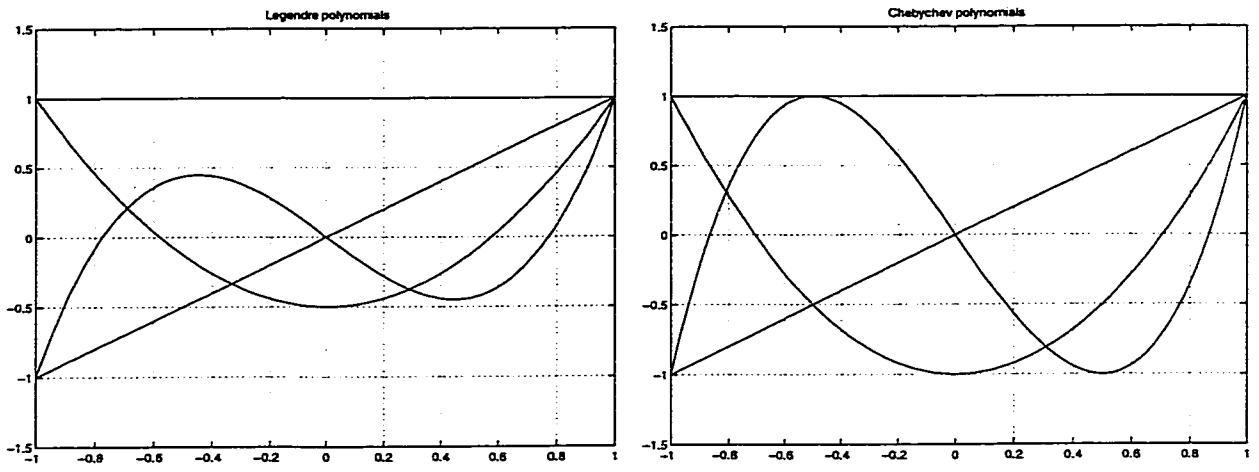
and thus have the largest leading coefficient, namely, $2^{-(k-1)}$ [9].

A plot of the first four Chebychev polynomials is shown in Fig. 3.1 (b).

3.5 Summary and overview of Laguerre polynomials

Orthogonal polynomials are associated with a positive integrable normalising weight function and are orthogonal over a certain defined interval. The degree k of an orthogonal polynomial is equal to the number of roots or zero-crossings on this interval. The Legendre and Chebychev orthogonal polynomials were given as examples.

We now turn to the Laguerre orthogonal polynomials and functions. The Laguerre polynomials exist in different forms depending on the number of free parameters and whether they are written in continuous-time (CT) or discrete-time (DT). As the polynomials need



(a) (b)

Figure 3.1: Ultraspherical polynomials: (a) Legendre (b) Chebychev.

to be orthogonalised in order to be used in a series expansion, we will refer to them here as functions.

In CT we have the generalised Laguerre functions which have two free parameters: the Laguerre parameter σ and the order of generalisation α. The generalised Laguerre functions have a rational Laplace transform only for even values of generalisation order [11]. For $\sigma = 0$ we obtain the Laguerre functions.

In DT the generalised functions are called the Meixner functions with Laguerre parameter ξ and generalisation order α. These do not have a rational z-transform. However there exists Meixner-like functions which have a rational z-transform. Again, their degeneralised form, with $\alpha = 0$, are called the discrete Laguerre functions or simply the Laguerre sequences.

We treat the CT and DT Laguerre functions in separate chapters as the development progresses naturally from orthogonal polynomials, to CT functions and further to DT sequences. Also more emphasis is placed on the DT case as we aim to present the Laguerre filter based on its z-transform expression.

Chapter 4

Laguerre polynomials and functions

The orthogonal polynomials and their related notation will now serve to introduce the Laguerre polynomials and their CT functions. We begin with the generalised Laguerre polynomials and functions ($\alpha > -1$), actually discovered by Sonine [12] and then their degeneralised versions ($\alpha = 0$) called the Laguerre polynomials and functions. We present an overview of the Laguerre function properties. These will be presented in detail for the DT versions in chapter 5, as simulations will be performed using a sampled system representation.

4.1 Laguerre polynomials

Analysis and geometry were areas of research of Edmond Nicolas Laguerre (1834-1886). His work in geometry was important at the time but has been overtaken. Laguerre studied approximation methods and introduced the Laguerre polynomials which are solutions to the Laguerre differential equations [13].

The continuous-time (CT) Laguerre functions are obtained by orthonormalising the Laguerre polynomials. These polynomials are obtained by orthogonalising the set of functions $\{t^k e^{-pt}\}, k \in \mathbb{N}, p > 0$ [14] [15]. Note that for $p \to 0$ the functions become $\{t^k\}$. We first present the generalised Laguerre functions.

4.2 Generalised Laguerre functions

The Laguerre series expansion is given as $f(x) = \sum_{k\in\mathbb{N}} c_k l_k^{(\alpha)}(x), x \geq 0$ where $l_k^{(\alpha)}(x)$ are the generalised Laguerre polynomials, also called the Laguerre-Sonine polynomials [10] [16] [17] [12] which appear in different equivalent forms

$$l_k^{(\alpha)}(x) = \frac{x^{-\alpha}e^x}{k!}\frac{d^k}{dx^k}(x^{k+\alpha}e^{-x}) \tag{4.1}$$

$$= \sum_{i=0}^{k} \frac{\Gamma(k+\alpha+1)}{\Gamma(i+\alpha+1)}\frac{(-x)^i}{i!(k-i)!} \tag{4.2}$$

$$= \sum_{i=0}^{k} (-1)^i \binom{k+\alpha}{k-i}\frac{x^i}{i!} \tag{4.3}$$

Polynomials $l_k^{(\alpha)}(x)$ are orthogonal under a window $w^{(\alpha)}(x) = x^\alpha e^{-x}$

$$\int_0^\infty w^{(\alpha)}(x) l_k^{(\alpha)}(x) l_m^{(\alpha)}(x) dx = \delta_{km}\frac{\Gamma(k+\alpha+1)}{k!} \tag{4.4}$$

In fact, the generalised Laguerre polynomials are an orthogonal basis for all finite norm functions $f(x)$ on $L_2(\mathbb{R}_+)$ as the following theorem states [12]

Theorem 4.1 *Suppose $f : (0,\infty) \mapsto \mathbb{R}$ is piecewise smooth, and that*

$$\int_0^\infty w(x) f^2(x) dx < \infty \tag{4.5}$$

with window $w(x) = x^\alpha e^{-x}$.

Then for all x and possibly at discontinuities

$$\frac{1}{2}(f(x_-) + f(x_+)) = \sum_{k=0}^{\infty} c_k l_k^{(\alpha)}(x) \qquad 0 < x < \infty \tag{4.6}$$

with series coefficients

$$c_k = \frac{k!}{\Gamma(k+\alpha+1)} \int_0^\infty w(x) f(x) l_k^{(\alpha)}(x) dx \tag{4.7}$$

To obtain the generalised orthogonal Laguerre functions, we perform the following changes.

1. We absorb the window [18] into the polynomial such that the inner product can now be taken with a unity window function, with functions $\sqrt{w^{(\alpha)}} l_k^{(\alpha)}$
2. We multiply the functions by the inverse square-root of the orthogonality constant (prefix of integral in 4.7) to make the functions orthonormal.
3. We let $x = \sigma t$ where $\sigma > 0$ is a scale factor that scales the time axis. Then $\frac{dx}{dt} = \sigma$ and the right-hand side of the inner product in becomes $\frac{1}{\sigma}\delta_{km}$. Thus again we normalise the functions by multiplying by $\sqrt{\sigma}$.

We then obtain the more common normalised form of the generalised Laguerre functions [16]

$$\lambda_k^{(\alpha)}(\sigma, t) = \sqrt{\frac{\sigma k!}{\Gamma(k+\alpha+1)}} \sum_{i=0}^{k} (-1)^i \binom{k+\alpha}{k-i} \frac{(\sigma t)^{i+\alpha/2}}{i!} e^{-\sigma t/2} \tag{4.8}$$

with

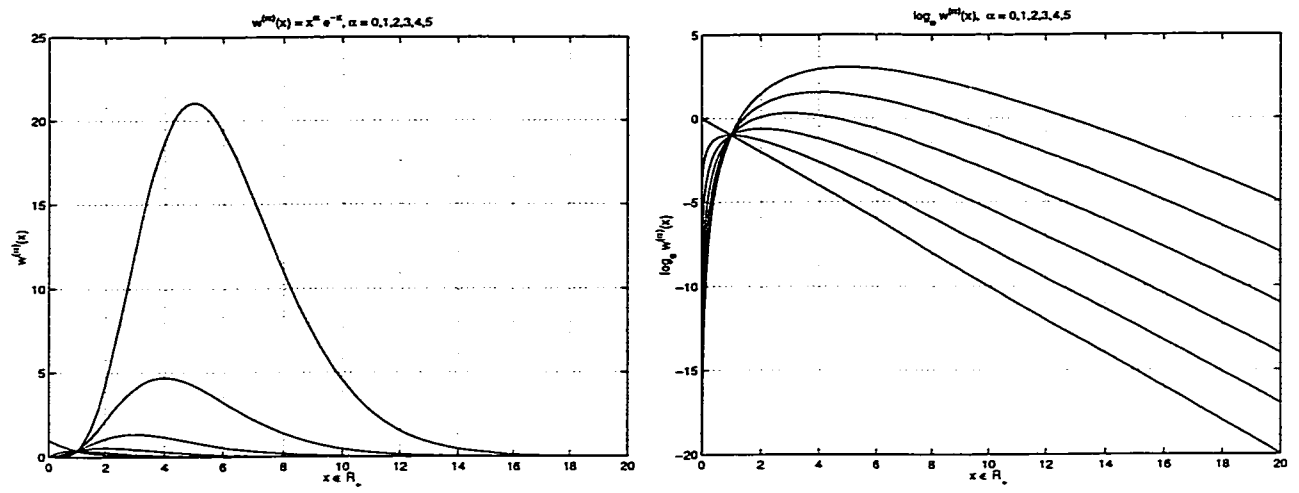
$$\langle \lambda_k(\sigma, t), \lambda_m(\sigma, t) \rangle = \delta_{km} \tag{4.9}$$

Some authors further use $\sigma = 2p$ [19] [14] where $p > 0$ is the Laguerre pole.

4.2.1 Window function

The orthogonalisation window and its logarithm are shown for orders $\alpha = 0, 1, 2, 3, 4, 5$ in Figs. 4.1 (a) and (b), respectively.

Also note that for $\alpha = \pm 1/2$, the Laguerre functions can be written as Hermite functions and vice-versa [10].



(a) (b)

Figure 4.1: Orthogonalisation window (a) $w^{(\alpha)}(x)$, (b) $\ln w^{(\alpha)}(x)$: $\alpha = 0$ (lowest) and $\alpha = 5$ (highest).

For $\alpha = 0$ we obtain the Laguerre functions. Increasing the value of α from 0 results in more slowly starting weight functions under which the generalised Laguerre functions were generated. This leads to more slowly starting generalised Laguerre functions. The order parameter α thus permits a shift in the center of energy. This is equivalent to saying that increasing α emphasises the low-frequency components at the expense of high-frequency components [16].

4.2.2 Laplace transform

The generalised Laguerre functions have a rational Laplace transform only for even values of α [17]. Since we are interested in a DT realisation, we will expand on the z-transform of Laguerre sequences in Chapter 6.

4.3 Laguerre functions

With $\alpha = 0$ we obtain the Laguerre polynomials $l_k(x)$ with the following equivalences [20] [18] [17] (cf. Appendix C)

$$l_k(x) = \frac{1}{k!} e^x \frac{d^k}{dx^k}(x^k e^{-x}) \tag{4.10}$$

$$= \sum_{i=0}^{k} \binom{k}{i} \frac{(-x)^i}{i!} \tag{4.11}$$

which are orthonormal under a window e^{-x}

$$\int_0^\infty e^{-x} l_k(x) l_m(x) dx = \delta_{km} \tag{4.12}$$

since $\Gamma(k+1) = k\Gamma(k) = \ldots = k!$ (cf. Appendix C).

The first 4 Laguerre polynomials are

$$l_0(x) = 1 \tag{4.13}$$

$$l_1(x) = 1 - x \tag{4.14}$$

$$l_2(x) = 1 - 2x + \frac{1}{2}x^2 \tag{4.15}$$

$$l_3(x) = 1 - 3x + \frac{3}{2}x^2 - \frac{1}{6}x^3 \tag{4.16}$$

The first 4 windowed Laguerre polynomials $e^{-x/2} l_k(x)$ are shown in Fig. 4.2. Note that the kth Laguerre functions, $e^{-x/2} l_k(x)$, has k roots on the interval $(0, \infty)$.

Similarly to the CT generalised Laguerre functions, we absorb the window and introduce a time scale to obtain the Laguerre functions $\lambda_k(\sigma, t)$

$$\lambda_k(\sigma, t) = e^{-x/2} l_k(x) \mid_{x=\sigma t} \tag{4.17}$$

$$= e^{-\sigma t/2} \sum_{i=0}^{k} \binom{k}{i} \frac{(-\sigma t)^i}{i!} \tag{4.18}$$

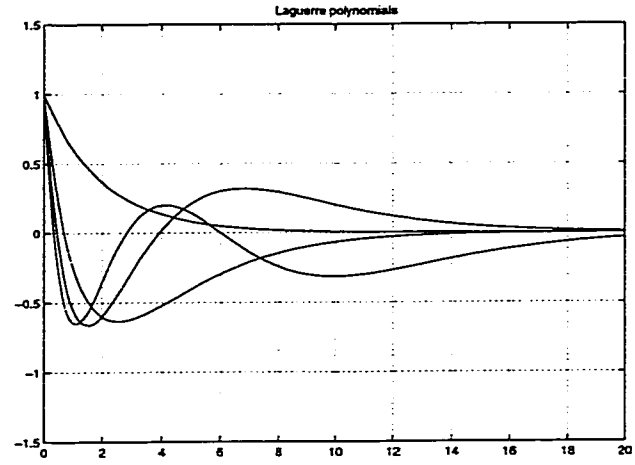


Figure 4.2: Laguerre polynomials $e^{-x/2}l_k(x), k = 0, 1, 2, 3$. $e^{-x/2}l_k(x)$ has k roots on $(0, \infty)$.

with

$$\langle \lambda_k(\sigma, t), \lambda_m(\sigma, t) \rangle = \frac{1}{\sigma}\delta_{km} \tag{4.19}$$

Thus the orthonormalised Laguerre functions become [14] [21].

$$\lambda_k(\sigma, t) = \sqrt{\sigma}e^{-\sigma t/2} \sum_{i=0}^{k} \binom{k}{i} \frac{(-\sigma t)^i}{i!} \tag{4.20}$$

Again, some authors substitute $\sigma = 2p$.

4.3.1 Laplace transform

The CT Laguerre functions have a simple Laplace transform [22] [14]

$$L_k(s) = \frac{\sqrt{2p}}{s+p}\left(\frac{s-p}{s+p}\right)^k \qquad k \in \mathbb{N} \tag{4.21}$$

and $p = \sigma/2$. Again, as we are interested in its DT equivalent, we postpone the discussion on the Laguerre filter realisation to Chapter 5.

4.3.2 Truncated Laguerre expansions

The Laguerre functions form an orthonormal basis in $L_2(\mathbb{R}_+)$ and thus any square integrable function $f(t)$ can be written as a Laguerre series as

$$f(t) = \sum_{k=0}^{\infty} c_k(\sigma)\lambda_k(\sigma, t) \tag{4.22}$$

where the Laguerre coefficients are determined by the projections of $f(t)$ onto the Laguerre functions

$$c_k(\sigma) = \langle f(t), \lambda_k(\sigma, t)\rangle. \tag{4.23}$$

When the series in (4.22) is truncated to K terms, the truncation error is $e(t) = f(t) - f_K(t)$ and the integrated square error is

$$E_K(\sigma) = \langle e(t), e(t)\rangle = \sum_{k=K}^{\infty} c_k^2(\sigma) \tag{4.24}$$

since the Laguerre series is an orthogonal decomposition.

There exists no analytic bound on the error introduced by truncating the series to K terms [12]. However, it was shown by Clowes [21] that the derivative of the integrated squared error with respect to the scale factor σ is zero when the last or subsequent Laguerre coefficient is zero

$$c_{K-1}(\sigma) = 0 \qquad \text{or} \qquad c_K(\sigma) = 0. \tag{4.25}$$

Note that for this value of σ, not only a minimum but a maximum or saddle point may exist. Thus this condition may be aided by a gradient or higher-derivative test [10]. Finding the minimum of the integrated squared error surface is thus a non-linear minimisation problem. As is typically the case in such problems, a good initial estimate of σ is crucial in obtaining a minimum solution.

4.3.3 Scale factor estimation

It turns out that an estimate for the time scale σ is given by [19] [16]

$$\hat{\sigma} = 2\sqrt{\frac{m_2}{m_1}} \tag{4.26}$$

where m_2, m_1 are the following moments of $f(t)$

$$m_2 = \langle \frac{df(t)}{dt}, t\frac{df(t)}{dt} \rangle \tag{4.27}$$

$$m_1 = \langle f(t), tf(t) \rangle \tag{4.28}$$

4.4 Chapter summary

We have presented the CT generalised Laguerre functions and the degenerated Laguerre functions with single parameter σ. The functions are orthonormal on a semi-infinite interval. The series expansion has the successive approximation property and can thus be truncated without recalculating the Laguerre coefficients. Our aim was to prepare the reader for the derivation of the Laguerre filter which follows from its DT representation. We have also only briefly discussed the mean-squared error surface and the scale factor estimation as these will be presented in more detail for the DT Laguerre functions.

Chapter 5

Laguerre sequences and functions

In this chapter we define the discrete-time (DT) Laguerre functions, or simply the Laguerre sequences. The more general polynomials in DT are called the Meixner polynomials and will not be presented. Once orthonormalised, their z-transform defines the Laguerre filter. The Laguerre filter can then be used to obtain the series expansion coefficients of a given sequence x. It can also be used to generate the Laguerre basis vectors which will, in turn, be used as elements in the Laguerre transform matrix defined in chapter 6.

5.1 Meixner functions

In DT, the generalised functions appear in a different form and are called the Meixner functions or sequences. They are however non-realisable since their z-transforms are non-rational. Den Brinker [11] defines Meixner-like functions that have a rational z-transform. We will focus on their degeneralised form which are the Laguerre sequences and which lead to the realisation of the Laguerre filter.

5.2 Discrete-time Laguerre polynomials

Similarly to the CT case, the DT Laguerre polynomials are obtained by orthogonalising the set of Laguerre sequences [14] $x_k[n] = n^k \xi^n, k \in \mathbf{N}, |\xi| < 1$.

The DT Laguerre difference polynomials of degree k [5] [11] [23]:

$$f_k[n] = \theta^{-n}\Delta^k[\binom{n}{k}\theta^n], \qquad n \in \mathbb{N} \tag{5.1}$$

where Δ is the forward difference operator, $\theta = e^{-\lambda}$ and $\lambda \in \mathbb{R}_+$. Taking the kth order difference it can be shown that (5.1) has the explicit form

$$f_k[n] = \theta^k \sum_{i=0}^{k} (\frac{\theta - 1}{\theta})^i \binom{k}{i}\binom{n}{i} \tag{5.2}$$

The set $\{f_k\}$ is a complete set in $l_2(\mathbb{N})$ and satisfies the orthogonality property on a semi-infinite interval under a window θ^n

$$\sum_{n=0}^{\infty} \theta^n f_k[n] f_l[n] = \frac{1}{\upsilon_k^2}\delta[k-l] \tag{5.3}$$

with

$$\upsilon_k = \sqrt{\frac{1-\theta}{\theta^k}}. \tag{5.4}$$

The real parameter $\theta \in (0,1)$ acts as an exponential window and is called the *discount factor*. From (5.2) we can rewrite these in their degenerated and normalised Meixner form which are called the Laguerre sequences [5] [23]

$$\psi_k[n] = (-1)^k \upsilon_k \xi^n f_k[n] \tag{5.5}$$

where $\xi = \pm\sqrt{\theta}$ is called the Laguerre parameter.

Their explicit form is given as [14]

$$\psi_k[n] = \sqrt{1-\xi^2} \sum_{i=0}^{k} (-1)^{i+k} C_i^k C_k^{n+k-i} \xi^{n+k-2i} \tag{5.6}$$

with C_k^n the binomial coefficient (cf. Appendix C). The Laguerre sequences are orthonormal over $l_2(\mathbb{N})$ [5]

$$\sum_{n=0}^{\infty} \psi_k[n]\psi_l[n] = \delta[k-l] \qquad k \in \mathbf{N} \tag{5.7}$$

and, in fact, form a basis for $l_2(\mathbf{N})$.

5.3 Laguerre filter

5.3.1 History of Laguerre filters

Much of the early work using Laguerre filters was in network and filter synthesis.

It was Lee [18] which on advice of Wiener himself used the Fourier transform of the Laguerre functions in his work on synthesis of electrical networks. Interestingly, Golay, inventor of the Golay code, has also written about the subject [24].

5.3.2 Derivation

By taking the z-transform of (5.5), a practical alternative definition of the Laguerre series expansion is obtained. We have [5]

$$\Psi_k(z) = Z\{\psi_k[n]\} = (-1)^k \upsilon_k Z\{\xi^{-n}\Delta^k \binom{n}{k} \theta^n\} \tag{5.8}$$

Now the z-transform of the binomial coefficient $\binom{n}{k}$ is (cf. Appendix C)

$$Z\{\binom{n}{k}\} = \frac{z}{(z-1)^{k+1}} \tag{5.9}$$

and using the shifting property, $Z\{\theta^n x\} = X(z/\theta)$, we have the transform of the product

$$Z\{\binom{n}{k}\theta^n\} = \frac{z/\theta}{(z/\theta - 1)^{k+1}}. \tag{5.10}$$

Since the z-transform of the kth order difference of a sequence with zero initial condition is $Z\{\theta^k x[n]\} = (z-1)^k X(z)$, we re-apply the shift property and obtain

$$Z\{\xi^{-n}\Delta^k\xi\binom{n}{k}\theta^n\} = Z\{(1/\sqrt{\theta})^n\Delta^k(\binom{n}{k}\theta^n)\} = (-1)^k(1-\sqrt{\theta}z)\frac{z/\sqrt{\theta}}{(z/\sqrt{\theta}-1)^{k+1}}. \quad (5.11)$$

Substituting v_k as well as the Laguerre parameter $\xi = \pm\sqrt{\theta}$ we obtain the Laguerre network equations [11] [5] [15]

$$\Psi_k(z) = \sqrt{1-\xi^2}\frac{z}{z-\xi}(\frac{1-\xi z}{z-\xi})^k \quad (5.12)$$

$$= \sqrt{1-\xi^2}\Psi_0(z)\Psi^k(z) \quad (5.13)$$

where $\Psi_0(z)$ is an IIR LP filter with pole at ξ

$$\Psi_0(z) = \frac{z}{z-\xi} \quad (5.14)$$

and $\Psi(z)$ is an all-pass [AP] filter with parameter ξ

$$\Psi(z) = \frac{1-\xi z}{z-\xi} \quad (5.15)$$

The Laguerre filter can thus be realised by taking a delay line with K taps, as in Fig. 5.1, replacing all the delay elements by first-order AP filters and prepending a single-pole LP filter. Fig. 5.2 shows the filter realisation [14] [5] [25]. Note that for $\xi = 0$, the Laguerre filter simplifies to a K-tap delay line.

The Laguerre sequences can be generated by the network shown in Fig. 5.3 with an impulse at its input.

Fig. 5.4 shows the first 8 Laguerre sequences or basis functions with $\xi = 0.5$ truncated to length 32. These are just the impulse responses of each Laguerre filter tap.

5.4 Laguerre spectrum

The set of Laguerre sequences $\{\psi_k[n]\}$ form a complete orthonormal set on $l_2(\mathbb{N})$ (5.7) and thus any real sequence $x[n] \in l_2(\mathbb{N})$ can be expressed as

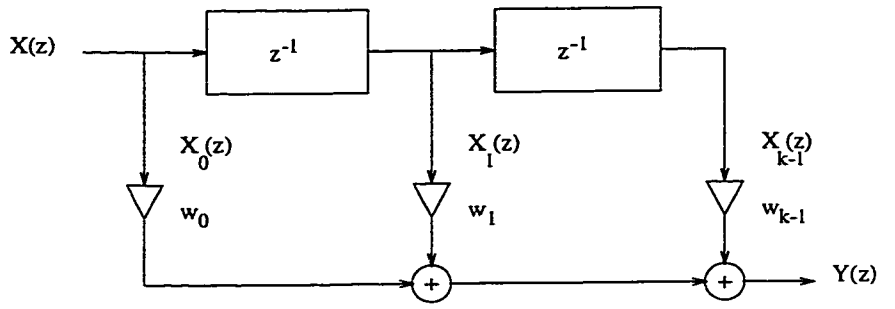


Figure 5.1: Classical FIR filter.

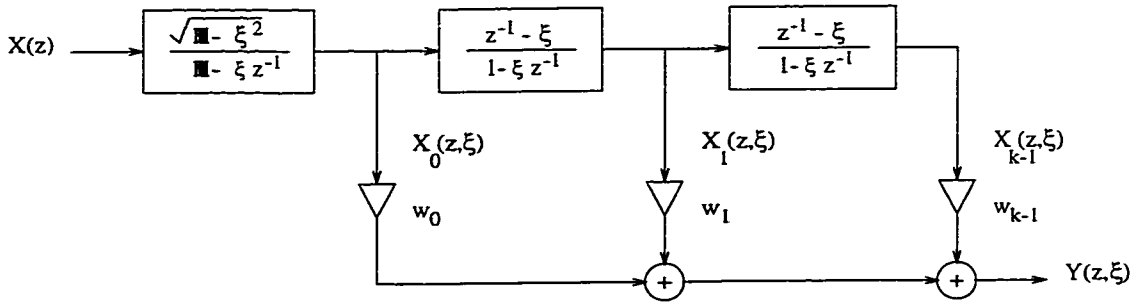


Figure 5.2: Laguerre filter.

$$x[n] = \sum_{k=0}^{\infty} c_k \psi_k[n] \tag{5.16}$$

with the set $c_k, k \in \mathbb{N}$,

$$c_k = \langle x[n], \psi_k[n] \rangle = \sum_{n=0}^{\infty} x[n] \psi_k[n] \tag{5.17}$$

the Laguerre spectrum of $x[n]$. The coefficients $\{c_k\}$ can also be computed using the z-transform version of Parseval's theorem to (5.17) [14]:

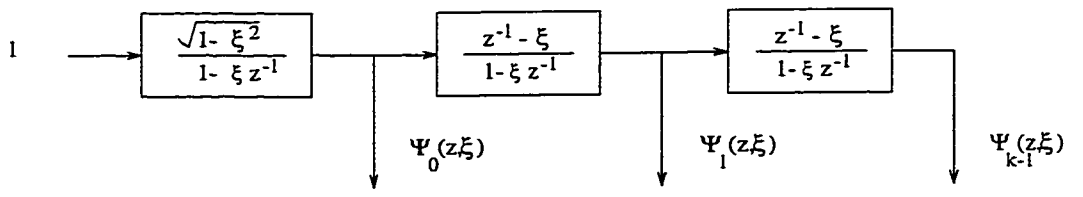


Figure 5.3: Network for generating Laguerre sequences with input $X(z) = 1$.

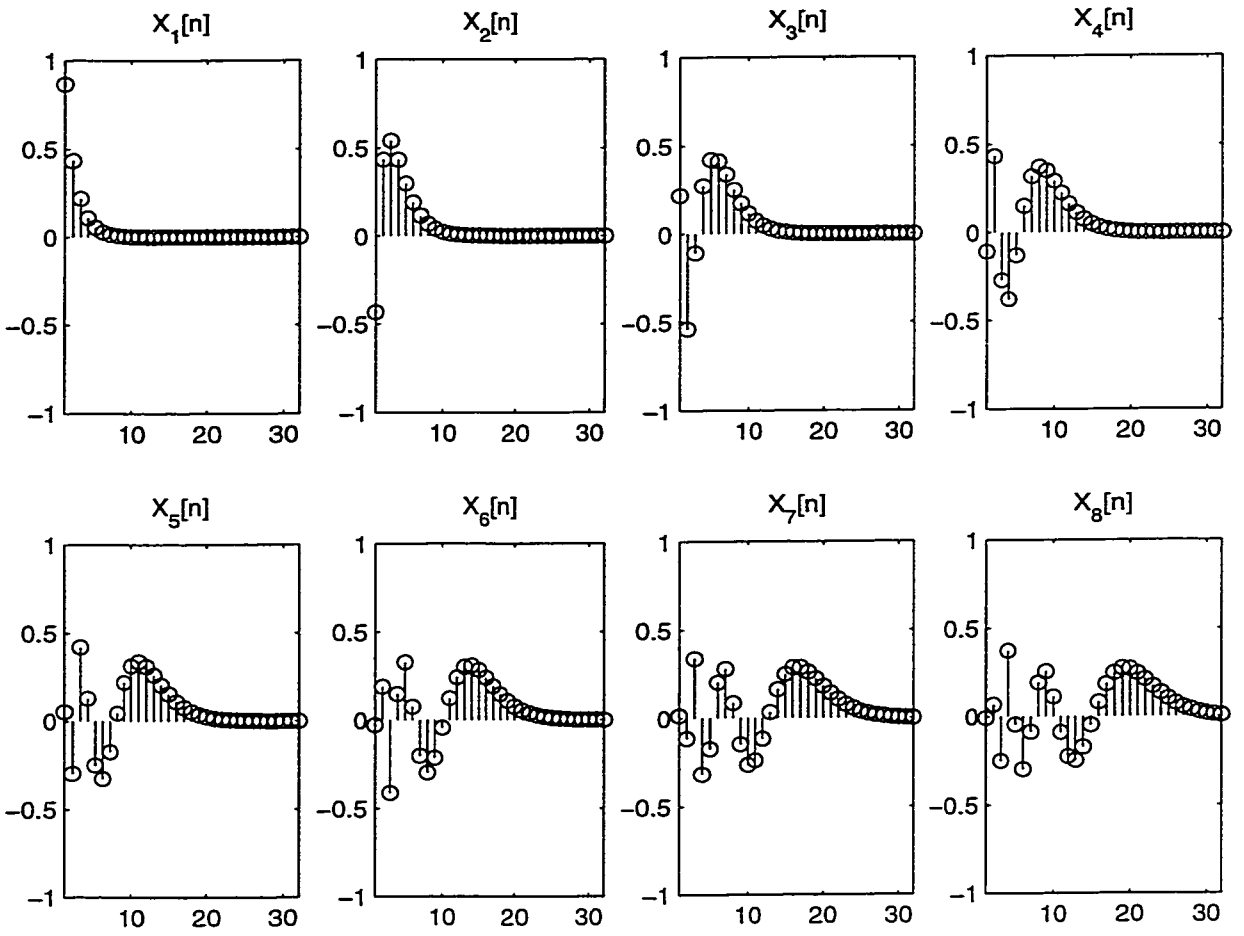


Figure 5.4: Basis functions $\psi_k[n], k = 0, 1, \ldots, 7$ of a 8 tap Laguerre filter, $\xi = 0.5$.

$$c_k = \langle X(z), \Psi_k(z) \rangle = \frac{1}{2\pi j} \oint_\Gamma X(z) \Psi_k(z^{-1}) \frac{dz}{z} \tag{5.18}$$

where Γ is a circle of radius $1 < r < \frac{1}{|\xi|}$ [14]. We note that $X(z)$ is guaranteed to be analytic outside the closed unit disc as it is square-summable in $l_2(\mathbb{N})$.

5.5 Recursive Laguerre identities

The following properties hold for the Laguerre functions $\{\psi_k[n]\}$ [23]:

1. The first component is a constant

$$\psi_0[0] = \sqrt{1-\xi^2} \tag{5.19}$$

2. The first basis vector is a geometric series with common ratio ξ

$$\psi_0[n+1] = \xi\psi_0[n] \tag{5.20}$$

3. The first elements of the basis vectors form a geometric series with common ratio $-\xi$

$$\psi_{k+1}[0] = -\xi\psi_k[0] \tag{5.21}$$

4. It may be shown [23] that the following recursive relation holds:

$$\xi\psi_k[n] + \psi_{k+1}[n] = \psi_k[n-1] + \xi\psi_{k+1}[n-1] \tag{5.22}$$

This relation plays a key role in the realisation of the Laguerre transform matrix (cf. Chapter 6) and the Laguerre filter (cf. Appendix A).

5. Further, by explicitly writing out (5.22) for the first values of k, we obtain

$$\sum_{r=0}^{j} \binom{j}{r} \xi^{j-r}\psi_{k+r}[n+j] = \sum_{r=0}^{j} \binom{j}{r} \xi^{r}\psi_{k+r}[n] \tag{5.23}$$

which states that when a sum of $j+1$ Laguerre coefficients, weighted by coefficients ξ^r and by the binomial coefficient, is delayed by j, then the coefficients become ξ^{j-r}.

5.6 System approximation by Laguerre functions

5.6.1 Energy

Since the basis functions are orthonormal, we can calculate the energy of a partial Laguerre series as

$$E_K = \langle x_K, x_K \rangle = \sum_{k=0}^{K} c_k^2 \tag{5.24}$$

and get a measure of the approximation by forming the ratio $E_K/E, \leq 1$ where $E = \langle x, x \rangle$.

5.6.2 Autocorrelation

A Laguerre autoregressive model of order K, AR(K), is given as

$$H(z) = \frac{1}{\sum_{k=0}^{K} w_k^* \Psi^k(z,\xi)} \tag{5.25}$$

$$= \frac{1}{1 - \sum_{k=1}^{K} w_k^* \Psi^k(z,\xi)} \tag{5.26}$$

where $\Psi^k(z,\xi)$ is the transfer functions of the k-th tap of a Laguerre filter.

Let $X(z)$ be a wide-sense stationary signal with mean 0 and variance σ_x^2 excite the Laguerre filter. Then the *Yule-Walker* equations are given as

$$\boldsymbol{R}(\xi)\boldsymbol{w}(\xi) = \boldsymbol{p}(\xi) \tag{5.27}$$

where the elements of $\boldsymbol{R}(\xi)$ are

$$r_{ij}(\xi) = \langle \Psi_i(z,\xi)X(z), \Psi_j(z,\xi)X(z) \rangle, \qquad 0 \le i,j \le K-1 \tag{5.28}$$

and those of $\boldsymbol{p}(\xi)$ are

$$p_i(\xi) = \langle \Psi_i(z,\xi)X(z), \Psi_i(z,\xi)X(z) \rangle, \qquad 1 \le i \le K \tag{5.29}$$

Furthermore it can be shown that $\boldsymbol{R}(\xi)$ is Toeplitz [14] by the dependence on $i-j$ of each element $r_{i,j}(\xi)$ written with the z-transform version of Parseval's theorem

$$r_{i,j}(\xi) = (1-\xi^2)\langle \Psi_0(z)\Psi_i(z)X(z), \Psi_0(z)\Psi_j(z)X(z) \rangle \tag{5.30}$$

$$= \frac{1-\xi^2}{2\pi j} \oint_{\Gamma} \Psi_0(z)\Psi_i(z)X(z)\Psi_0(z^{-1})\Psi_j(z^{-1})X(z^{-1})\,\frac{dz}{z} \tag{5.31}$$

$$= \frac{1-\xi^2}{2\pi j} \oint_\Gamma \frac{1}{z-\xi} \left(\frac{1-\xi z}{z-\xi}\right)^i \frac{1}{1-z\xi} \left(\frac{z-\xi}{1-z\xi}\right)^j X(z)X(z^{-1})\, dz \tag{5.32}$$

$$= \frac{1-\xi^2}{2\pi j} \oint_\Gamma \frac{(1-\xi z)^{i-j-1}}{(z-\xi)^{i-j+1}} X(z)X(z^{-1})\, dz \tag{5.33}$$

5.6.3 Optimal pole

The optimal truncated Laguerre expansion is the one that minimises the approximation error or equivalently maximises the Kth order sum in (5.24):

$$\max_\xi \{E_K\} \tag{5.34}$$

The derivative of E_K with respect to ξ is

$$\frac{dE_K}{d\xi} = 2 \sum_{k=0}^{K} c_k \frac{dc_k}{d\xi} \tag{5.35}$$

which will unfortunately solve for all extrema and saddle points of the error function E_K.

A remarkable [26] property of the Laguerre function is that its derivative with respect to its pole is itself a Laguerre series [27]:

$$\frac{d}{d\xi} L_k(n) = \frac{-k}{1-\xi^2} L_{k-1}(n) + \frac{k+1}{1-\xi^2} L_{k+1}(n) \qquad k \geq 1 \tag{5.36}$$

with $dL_0(n)/d\xi = L_1(n)/(1-\xi^2)$.

The derivative can thus easily be generated by adding an extra section to the Laguerre filter. Using (5.36) and the Laguerre spectrum (5.17), we can write the derivative of the spectrum coefficients in a similar way:

$$\frac{dc_k}{d\xi} = \frac{-kc_{k-1} + (k+1)c_{k+1}}{1-\xi^2} \tag{5.37}$$

Using (5.37) we rewrite (5.35) as a telescopic series where sequent $c_k \frac{dc_k}{d\xi}$ cancel and thus

$$\frac{dE_K}{d\xi} = \frac{2(K+1)}{1-\xi^2} c_K(\xi) c_{K+1}(\xi) \tag{5.38}$$

and the optimality statement becomes

$$c_K(\xi) = 0 \quad \text{or} \quad c_{K+1}(\xi) = 0 \tag{5.39}$$

5.6.4 Estimation of optimal pole

By comparing the Laguerre generating functions and sequences $t^i e^{-pt}$ and $n^i \xi^n$, we deduce that $\xi = e^{-p}$ with $\sigma = 2p$ and thus we rewrite (4.26) as

$$\hat{\xi} = e^{-\hat{\sigma}/2} \tag{5.40}$$

$$= e^{-\sqrt{\frac{\mu_2}{\mu_1}}} \tag{5.41}$$

where μ_2, μ_1 are the following moments of $f[n]$

$$\mu_2 = \langle (f[n+1] - f[n]), n(f[n+1] - f[n]) \rangle \tag{5.42}$$

$$\mu_1 = \langle f[n], nf[n] \rangle \tag{5.43}$$

5.7 Chapter summary

The Laguerre filter is realised by substituting the unit-delay element in a tapped delay line by a first-order allpass filter with parameter $|\xi| < 1$ and by prepending the structure by a normalising single-pole IIR filter. The impulse response of each tap k is the kth Laguerre

orthonormal sequence $\psi_k[n]$. Note that if the number of taps k is increased to infinity, the resulting K Laguerre sequences form an orthonormal basis for $l_2(\mathbb{N})$. This requirement follows from the definition in (5.7). For a practical filter, K is finite and thus the K Laguerre sequences will still be orthonormal but not a basis for $l_2(\mathbb{N})$. Also note that as $\xi \longrightarrow 0$, for any finite number of taps K, the Laguerre orthonormal sequences become the canonical basis $\{\delta[n-k]\}$. Thus the Laguerre sequences produced by a Laguerre filter become bases for two independent limiting cases. We investigate this behaviour by motivating the existence of the Laguerre transform in matrix form and by looking at a few transform examples.

Chapter 6

Matrices and DT transforms

We have seen that a vector can be expressed as an orthogonal series expansion of basis vectors. As will be shown, a finite series expansion can be represented by a transformation matrix having the basis vectors as columns. In this chapter we first present matrix properties which will be used to characterise commonly-known matrix transforms. We then show that the forward and inverse transforms can be realised using a filter bank. This result motivates the definition of the Laguerre transform based on its filter realisation. After having presented the more classical transforms, such as the Fourier transform, we motivate the existence and properties of the Laguerre transform. We then show examples of the Laguerre transform.

6.1 Matrix properties

6.1.1 Eigenvectors

The roots of the kth order characteristic polynomial of a $k \times k$ matrix $\boldsymbol{A}$

$$\det(\lambda \boldsymbol{I} - \boldsymbol{A}) = 0 \tag{6.1}$$

are called the eigenvalues $\{\lambda\}$. By the fundamental theorem of algebra, there exists at least one possibly complex root λ. Thus there may be 1 to k roots.

Associated with each eigenvalue λ_i is an eigenvector $\boldsymbol{q}_i$. The eigenvector of a matrix $\boldsymbol{q}$ is a vector $\boldsymbol{q}$ such that $\boldsymbol{Aq}$ points in the same (or opposite) direction of $\boldsymbol{q}$ [7]:

$$\boldsymbol{A}\boldsymbol{q} = \lambda \boldsymbol{q} \tag{6.2}$$

Given λ_i, we can solve for eigenvector $\boldsymbol{q}_i$:

$$(\lambda_i \boldsymbol{I} - \boldsymbol{A})\boldsymbol{q}_i = \boldsymbol{0} \tag{6.3}$$

6.1.2 Positive definite matrices

A $k \times k$ matrix $\boldsymbol{A}$ is called positive definite if the scalar

$$\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} = \sum_{i=1}^{k}\sum_{j=1}^{n} a_{ij}x_i x_j > 0 \tag{6.4}$$

for all non-zero $\boldsymbol{x}$.

For a $k \times k$ symmetric positive definite matrix $\boldsymbol{A}$ [7],

1. The eigenvalues are real and non-negative, and can be ranked $\lambda_i \leq \lambda_j, 1 \leq i \leq j \leq k$.
2. The corresponding eigenvectors are mutually orthogonal and form an k dimensional basis. They can further be normalised such that $\langle \boldsymbol{q}_i, \boldsymbol{q}_j \rangle = \delta_{ij}$
3. The matrix $\boldsymbol{A}$ can then be constructed as a weighted sum of outer products $\boldsymbol{A} = \sum_{i=1}^{k} \lambda_i \boldsymbol{q}_i \boldsymbol{q}_i^{*T}$

Thus we write

$$\boldsymbol{A} = \boldsymbol{Q}\Lambda\boldsymbol{Q}^{*T} \tag{6.5}$$

where $\boldsymbol{Q}$ contains the eigenvectors $\boldsymbol{q}$ as columns and say that matrix $\boldsymbol{Q}$ diagonalises matrix $\boldsymbol{A}$.

Symmetric positive-definite matrices describe covariance matrices. Their eigenvalues are real non-negative and distinct and can furthermore be ranked. All covariance and correlation matrices are symmetric [7].

6.1.3 Toeplitz matrix

A $k \times k$ Toeplitz matrix has elements t_{ij} uniquely specified by the value $i - j$.

$$\boldsymbol{T} = \begin{bmatrix} t_0 & t_1 & \dots & t_{k-1} \\ t_{-1} & t_0 & \dots & t_{k-2} \\ \vdots & \vdots & & \vdots \\ t_{-k+1} & t_{-k+2} & \dots & t_0 \end{bmatrix} \tag{6.6}$$

A special case of a Toeplitz matrix is *circulant* matrix where each row is a circular right-shift of its previous row. Thus its elements c_i are elements of the k-dimensional set $\{c_0 c_1 \dots c_{k-1}\}$

$$\boldsymbol{C} = \begin{bmatrix} c_0 & c_1 & \dots & c_{k-1} \\ c_{k-1} & c_0 & \dots & c_{k-2} \\ \vdots & \vdots & & \vdots \\ c_1 & c_2 & \dots & c_0 \end{bmatrix} \tag{6.7}$$

Note that both matrices have constant diagonals and are not necessarily symmetric.

6.2 Transforms

6.2.1 Unitary transforms

An orthonormal basis satisfies completeness and orthonormality. A matrix satisfying both properties is called *unitary*. A unitary transform matrix has the property that its inverse is its Hermitian (its conjugate transpose) $\boldsymbol{U}^{-1} = \boldsymbol{U}^{*T}$, or equivalently

$$\boldsymbol{U}\boldsymbol{U}^{*T} = \boldsymbol{I} \tag{6.8}$$

with the converse also being true,

$$\boldsymbol{U}^{*T}\boldsymbol{U} = \boldsymbol{I}. \tag{6.9}$$

In fact, the two above equations are direct expressions for orthonormality and completeness in matrix form. Suppose the basis we are using has dimension N. Then consider the non-square matrix formed by the first K basis vectors $\{\phi_k\}$ placed as columns in the $N \times K$ matrix $\boldsymbol{U}^{*T}$ with $K \leq N$. Then we can state orthonormality as

$$\left[\boldsymbol{U}\boldsymbol{U}^{*T}\right]_{kl} = \sum_{n=1}^{N} \phi_k[n]\phi_l[n] = \delta[k-l] \quad 0 \leq k, l < K \tag{6.10}$$

and completeness as

$$\left[\boldsymbol{U}^{*T}\boldsymbol{U}\right]_{nm} = \sum_{k=1}^{K} \phi_k[n]\phi_k[m] = \delta[n-m] \quad 0 \leq n, m < N \tag{6.11}$$

Note that the orthogonality identity matrix has size $K \times K$ whereas the completeness identity matrix has size $N \times N$.

Properties

In addition to the definition in (6.8), unitary matrices have the following properties [28]

1. A unitary matrix preserves the $l_2(\mathbb{Z})$ norm [8]. This is equivalent to stating that it conserves energy as

$$\|X\|^2 = \|\boldsymbol{U}\boldsymbol{x}\|^2 = \boldsymbol{x}^{*T}\boldsymbol{U}^{*T}\boldsymbol{U}\boldsymbol{x} = \|x\|^2 \tag{6.12}$$

   A non-unitary transform is said to be *lossy*.

2. Certain unitary transforms, such as the KLT and DCT, pack a large fraction of the average energy of a signal into relatively few components. This is exploited in signal compression as we then have an alternative signal representation requirering less coefficients.

3. The determinant and eigenvalues of a unitary matrix have unit magnitude [8].

Example 6.1 *The Givens planar rotation is given by*

$$\boldsymbol{U}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{6.13}$$

and is unitary since

$$\boldsymbol{U}_\theta \boldsymbol{U}_\theta^{*T} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.14}$$

and also $\boldsymbol{U}_\theta^{*T} \boldsymbol{U}_\theta = \boldsymbol{I}$.

6.3 Filter bank notation

For K dimensional vector spaces, we can express a series expansion in (2.29) and (2.30) as premultiplication by a transform matrix. For the orthogonality condition, if we define the synthesis filter bank $\{g_k\}$

$$g_k[n] = \phi_k[n], \qquad k \in \{0, 1, \ldots K-1\} \tag{6.15}$$

and analysis filter bank $\{h_k\}$

$$h_k[n] = g_k[-n], \qquad k \in \{0, 1, \ldots K-1\} \tag{6.16}$$

we can write the cascade of analysis and synthesis transforms in matrix form:

$$\boldsymbol{X} = \boldsymbol{\phi}_a \boldsymbol{x} \quad \text{and} \quad \boldsymbol{x} = \boldsymbol{\phi}_s \boldsymbol{X} \tag{6.17}$$

where

$$\boldsymbol{\phi}_a = \begin{bmatrix} \phi_0[-n]^{*T} \\ \phi_1[-n]^{*T} \\ \vdots \\ \phi_{K-1}[-n]^{*T} \end{bmatrix} \tag{6.18}$$

$$\boldsymbol{\varphi}_s = \begin{bmatrix} \phi_0[n] & \phi_1[n] & \cdots & \phi_{K-1}[n] \end{bmatrix} \tag{6.19}$$

and $\phi_k[n] = [\phi_k[0]\phi_k[1] \ldots \phi_k[N-1]]^T$ is a $N \times 1$ column vector.

Thus by placing the basis vectors as columns in a matrix we have a synthesis matrix. The analysis matrix is then its inverse and may be simple to calculated depending on whether the synthesis matrix is unitary, real and/or symmetric.

Using the analysis and synthesis matrices, we can rewrite the definitions for orthonormality and completeness. The matrix elements of each become

$$\left[\Phi_a \Phi_s\right]_{kl} = \sum_{n=1}^{N} \phi_k[n]\phi_l[n] = \delta[k-l] \quad 0 \leq k, l < K \tag{6.20}$$

$$\left[\Phi_s \Phi_a\right]_{nm} = \sum_{k=1}^{K} \phi_k[n]\phi_k[m] = \delta[n-m] \quad 0 \leq n, m < N \tag{6.21}$$

6.3.1 How to choose a series expansion

To choose a series expansion, certain qualities are important. We would like to write our signal as a sum as in (2.29). One requirement is that the set $\{\varphi_k[n]\}$ be a basis of orthonormal functions. By the successive approximation property of orthogonal expansions, this will allow us to truncate the series when a certain amount of error is allowed. The set $\{\varphi_k[n]\}$ should also be structured from a computation and practical point of view [8].

We will consider the following unitary transforms for which we mention their properties:

1. Generally complex, non-symmetric transforms. From (6.8), these matrices have the property that $\boldsymbol{A}\boldsymbol{A}^{*T} = \boldsymbol{I}$. The KL and DFT are such transforms.

2. Real, symmetric transforms. If $\boldsymbol{A}$ is symmetric, $\boldsymbol{A} = \boldsymbol{A}^T$ and $\boldsymbol{A}^{-1} = \boldsymbol{A}$. Thus $\boldsymbol{A}^{-1} = \boldsymbol{A} = \boldsymbol{A}^T = \boldsymbol{A}^*$. The DST and identity matrix belong to this group.

3. Real, non-symmetric transforms. If $\boldsymbol{A}$ is real then $\boldsymbol{A} = \boldsymbol{A}^*$ and $\boldsymbol{A}^{-1} = \boldsymbol{A}^T$. The DCT and Haar transforms belong to this group.

4. The Laguerre transform with parameter ξ. In this case, the transform matrix is real and thus $\boldsymbol{A} = \boldsymbol{A}^*$. We will show that symmetry is achieved with $\xi = 0$ and otherwise the matrix is symmetric if we take the element-by-element absolute value. We will show that orthogonality and completeness are affected by ξ and the matrix size.

6.3.2 DT Karhunen-Loeve transform

For a random input signal this transform is optimal as its transform matrix diagonalises the input autocorrelation matrix $\boldsymbol{R}$. This occurs since the column vectors in the KLT matrix are the eigenvectors of $\boldsymbol{R}$. Thus, by (6.5), the diagonalised matrix contains the ordered eigenvalues of $\boldsymbol{R}$.

6.3.3 DT cosine/sine transform

The knth element of the $N \times N$ DCT matrix is given as

$$\phi_k[n] = \begin{cases} \frac{1}{\sqrt{N}} & k = 0, 0 \leq n \leq N-1 \\ \sqrt{\frac{2}{N}} \cos \frac{\pi(2n+1)k}{2N} & 1 \leq k \leq N-1, 0 \leq n \leq N-1 \end{cases} \tag{6.22}$$

For a Markov-1 process with correlation coefficient $|\rho| > 0.95$, the DCT approaches the same energy packing property as the KL transform. Its major advantage is that it can be realised using the FFT and is thus a fast transform. The DST approaches the KLT when $|\rho| < 0.05$ [28].

6.3.4 Discrete-time Fourier transform

The discrete-time (DT) Fourier transform

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{j\omega n} \, d\omega \tag{6.23}$$

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n} \tag{6.24}$$

produces a 2π-periodic series. Because the synthesis equation involves integration, it has less practical use in DSP. Instead the DT Fourier series is used, and the transform pair is then called the Discrete Fourier Transform (DFT) pair

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X_k W_N^{-nk} \tag{6.25}$$

$$X_k = \sum_{n=0}^{N-1} x[n] W_N^{nk} \tag{6.26}$$

where $x[n]$ is either N-periodic or of finite length N and $\{W_N^n\} = \{e^{-j2\pi n/N}\}$ are the N roots of unity. The unitary basis functions $\{\phi_k\}$ are given as

$$\phi_k[n] = \frac{1}{\sqrt{N}} e^{j2\pi kn/N} \tag{6.27}$$

The Fourier basis vectors of the unitary DFT are eigenvectors of any circulant matrix [28]. Thus the DFT diagonalises circulant matrices.

The synthesis and analysis filters and their output equations are

$$g_k[n] = \phi_k[n] = e^{-j2\pi kn/N} \tag{6.28}$$

$$h_k[n] = \phi_k[-n] = e^{j2\pi kn/N} \tag{6.29}$$

$$x[n] = \sum_{k=0}^{N-1} X_k e^{j2\pi kn/N} \tag{6.30}$$

where

$$X_k = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} \tag{6.31}$$

For $N = 4$, the synthesis matrix is

$$\boldsymbol{\phi}_s = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}. \tag{6.32}$$

This DFT transform matrix is a symmetric matrix $\boldsymbol{\Phi} = \boldsymbol{\Phi}^T$ since we can invert the order of n and k without changing the matrix. From the DFT definition pair (6.26) and (6.25), we see that the analysis and synthesis matrices are also related by $\boldsymbol{\Phi}_a^* = \boldsymbol{\Phi}_s$. Thus the transform is unitary

$$\boldsymbol{\phi}_s \boldsymbol{\phi}_a = \boldsymbol{\phi}_a^{*T} \boldsymbol{\phi}_a = \boldsymbol{I} \tag{6.33}$$

The DFT has the disadvantage in that it trades time resolution for frequency resolution. Its basis functions are everlasting exponentials and thus have no time locality. To remedy this, an overcomplete expansion[8], the Short-Time Fourier Transform (STFT), provides some time locality in that the expansion is done on a short time segment.

Another disadvantage is that the frequency tiling is uniform. This may not be optimum for certain applications where greater frequency resolution is required in certain frequency bands. Its main advantage is that it can be realised using the Fast Fourier Transform (FFT) and is thus a *fast* algorithm.

6.3.5 Discrete-time identity transform

If we choose the canonical basis $\phi_k[n] = \delta[n-k], 0 \leq, n, k, \leq N-1$, the transform matrices becomes $\boldsymbol{\phi_a} = \boldsymbol{\phi_s} = \boldsymbol{I}$. This transform is symmetric, real, and orthogonal. The canonical functions form an orthonormal basis on $\mathbb{Z}^N$.

The synthesis and analysis filters and their output equations are

$$g_k[n] = \phi_k[n] = \delta[n-k] \tag{6.34}$$

$$h_k[n] = \phi_k[-n] = \delta[k-n] \tag{6.35}$$

$$x[n] = \sum_{k=0}^{N-1} X_k \delta[k-n] = X_k \tag{6.36}$$

where

$$X_k = \sum_{n=0}^{N-1} x[n]\delta[n-k] = x[n] \tag{6.37}$$

Note that the analysis filter is non-causal. Although this case seems trivial, it corresponds to the degenerate form ($\xi = 0$) of the Laguerre transform.

6.4 Laguerre transform

The $K \times N$ Laguerre transform matrix $\boldsymbol{L} \doteq \boldsymbol{L}(\xi)$ with parameter $\xi, |\xi| < 1$. We denote this matrix as the analysis matrix $\boldsymbol{\Psi}_a = \boldsymbol{L}$ and present some of its properties.

6.4.1 Construction

Let $\psi_k[n]$ be the n, kth element of the synthesis matrix $\boldsymbol{L}^T$, then using (5.19), (5.20), (5.21) and (5.22), the Laguerre transform matrix is constructed as follows:

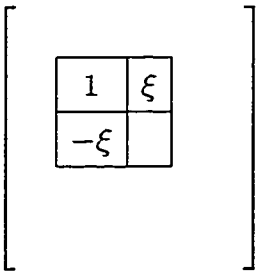
$$\psi_0[0] = \sqrt{1-\xi^2} \tag{6.38}$$

$$\psi_0[1] = \xi\psi_0[0] \tag{6.39}$$

$$\psi_1[0] = -\xi\psi_0[0] \tag{6.40}$$

$$\psi_{k+1}[n] = \psi_k[n-1] + \xi\psi_{k+1}[n-1] - \xi\psi_k[n] \tag{6.41}$$

The recursive relation may be visualised as a local operator affecting neighbouring matrix elements of $\boldsymbol{L}^T$. The operator looks like an inverted L-shaped 3-point window moving in the k (increasing column index) direction, thus filling the matrix $\boldsymbol{L}^T$ one row at time:



Note that if $N = K$ and $\xi = 0$, we obtain the identity matrix. We can thus obtain the Laguerre basis vectors using the recursive relations given here or record the outputs of a Laguerre filter with an impulse as input.

6.4.2 Orthogonality and completeness

The $K \times N$ Laguerre matrix $\boldsymbol{L}$ a non-symmetric, real transform. Recall the Laguerre sequences are an orthonormal basis on a semi-infinite interval. In matrix form this would require $N \to \infty$. Thus for finite N, the Laguerre transform is not orthonormal. However, as long as $K < N$, we can choose to weaken completeness but maintain orthonormality of the matrix $\boldsymbol{L}$. Consider the following cases with $\xi \neq 0$:

1. For $K < N$, orthonormality (6.10) is satisfied but completeness (6.11) isn't. This is the case when we want less coefficients in the transform (Laguerre) domain. The completeness condition matrix then looks like

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J} \end{bmatrix} \tag{6.42}$$

with $\det \boldsymbol{J} < 1$.

In this case we are less concerned with completeness than with the ability to truncate the (orthonormal) series expansion. Weak completeness will affect synthesis since synthesis increases the space dimensions from K to N. Examples will be shown.

2. For $K > N$, orthonormality isn't satisfied but completeness is. This condition is not useful as the successive approximation condition does not hold.

3. For $K = N$, neither condition is satisfied except for $\xi \to 0$ and this yields the identity matrix.

6.4.3 Inverse

In chapter 7, we show that the inverse Laguerre transform is obtained by negating ξ. However, since unitariness requires both orthonormality and completeness, the inverse Laguerre transform matrix may not always exist. This means that, although we can compute the inverse Laguerre mapping using the recursive definitions, the finite size inverse Laguerre matrix deviates from the true (unitary) inverse. The consequences of this will be shown in the examples.

When the matrix is close to being unitary, with $\xi \to 0$ and/or $K, N \to \infty$, the transform is invertible as

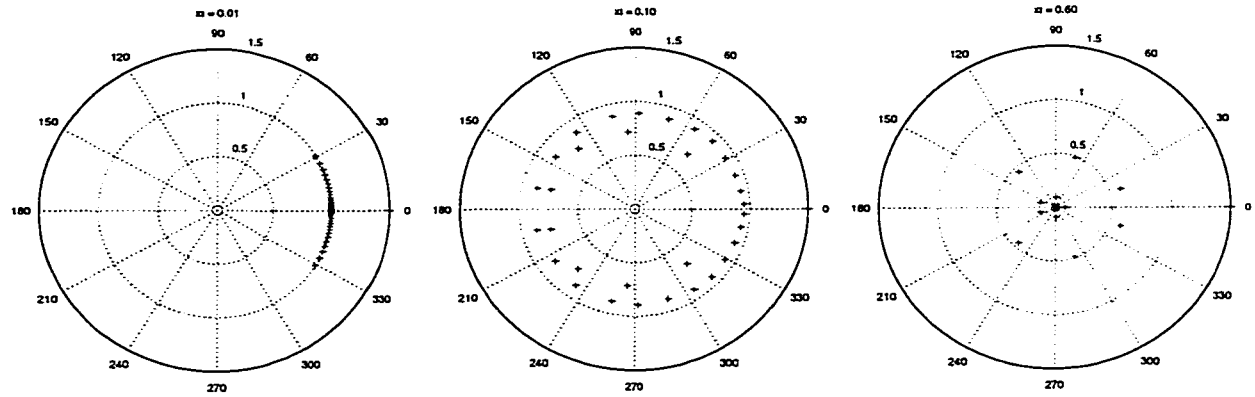
$$\boldsymbol{L}^{-1}(\xi) = \boldsymbol{L}^{T}(\xi) = \boldsymbol{L}(-\xi) \tag{6.43}$$

6.4.4 Absolutely symmetric

Since inverting ξ results in the transpose, and by the construction equations (5.19), (5.20), (5.21) and (5.22), taking the element-by-element absolute value of the matrix elements makes the matrix symmetric.

6.4.5 Eigenvalues

It has been observed that the $K = N$ square matrix $\boldsymbol{L}^T$ has K complex conjugate eigenvalues $\lambda_k \leq 1$ which migrate from $z = 1$ at $\xi = 0$, circulate along the inside of the unit circle and then spiral on unique $K/2$ paths toward $z = 0$ as $\xi \rightarrow \pm 1$. The eigenvalues of the 32×32 matrix $\boldsymbol{L}^T$ for $\xi = 0.01, 0.1$ and 0.6 are shown in Fig. 6.1 (a), (b) and (c), respectively.



(a) (b) (c)

Figure 6.1: Eigenvalues of $\boldsymbol{L}^T$: (a) $\xi = 0.01$, (b) $\xi = 0.1$, (c) $\xi = 0.6$.

6.4.6 Limit cases

The $N = K$ square matrix $\boldsymbol{L}$ is a real, orthonormal, absolutely-symmetric matrix with parameter $\xi, |\xi| < 1$. For $\xi = 0, \boldsymbol{L} \equiv \boldsymbol{I}$. For $\xi \rightarrow \pm 1$, the elements become equal in magnitude but have sign $(-1)^i$ in the ith column or row, respectively:

$$\left[\boldsymbol{L}\right]_{ij} = \begin{cases} (-1)^j\zeta & \xi \to 1 \\ (-1)^i\zeta & \xi \to -1 \end{cases} \qquad \zeta \to 0, 0 \le i, j < N-1 \tag{6.44}$$

6.5 Laguerre transform examples

We present Laguerre transform examples. The first shows its weak unitariness and the second example shows signal compression results.

6.5.1 Laguerre transform matrix

The 4×4 Laguerre matrix $\boldsymbol{L}(\xi)$ with $\xi = 0.1$ is given by

$$\boldsymbol{L}(0.1) = \begin{bmatrix} 0.9950 & 0.0995 & 0.0099 & 0.0010 \\ -0.0995 & 0.9751 & 0.1960 & 0.0295 \\ 0.0099 & -0.1960 & 0.9359 & 0.2867 \\ -0.0010 & 0.0295 & -0.2867 & 0.8786 \end{bmatrix} \tag{6.45}$$

The Laguerre matrix with parameter $\xi = -0.1$ is $\boldsymbol{L}(-0.1) = \boldsymbol{L}^T(0.1)$ and the product $\boldsymbol{L}(0.1)\boldsymbol{L}^T(0.1)$ is

$$\boldsymbol{L}(0.1)\boldsymbol{L}^T(0.1) = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & -0.0002 & -0.0015 \\ 0.0000 & -0.0002 & 0.9966 & -0.0222 \\ 0.0000 & -0.0015 & -0.0222 & 0.8549 \end{bmatrix} \tag{6.46}$$

We note that transposing the Laguerre matrix with parameter ξ results in the Laguerre transform matrix with parameter $-\xi$ and that the Laguerre matrix is less and less unitary as $|\xi| \to 1$.

6.5.2 Laguerre signal compression

To illustrate the use of the Laguerre transform, the effects of series truncation, orthogonality of the coefficients, and MSE, we present a 1-D and 2-D example of signal compression using

the Laguerre transform. We compare these results to those obtained using the DCT.

Recall that the Laguerre functions are orthonormal on a semi-infinite interval. When we truncate the series, which we must to have a matrix form, we limit the the minimum MSE to a finite value. As $\xi \to 0$, the series becomes complete and the minimum MSE goes to zero. We then have the identity transform.

In the following examples, we let $\xi = 0.1$ and $\xi = 0.6$ to illustrate the the minimum MSE limitation and the deterioration of the inverse transform.

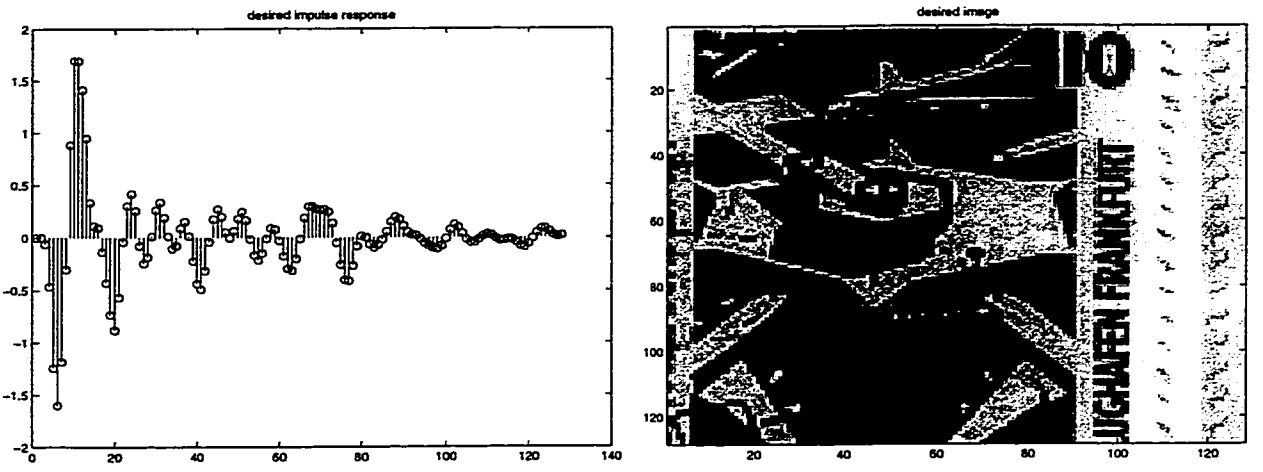
Impulse response

Suppose a given impulse response $\boldsymbol{x}$, a column vector length N as shown in Fig. 6.2 (a). The length K Laguerre coefficients are then obtained by the analysis transform $\tilde{\boldsymbol{x}} = \boldsymbol{L}(\xi)\boldsymbol{x}$ where $\boldsymbol{L}(\xi)$ is the $K \times N$ Laguerre transform matrix. Since $N > K$, the expansion is orthonormal and the coefficient values do not depend on K and thus we have only one plot for the Laguerre coefficients. These are shown in Fig. 6.3 (a) for $\xi = 0.1$ and (b) for $\xi = 0.6$. The series expansion is then the synthesis transform $\hat{\boldsymbol{x}} = \boldsymbol{L}^T(\xi)\tilde{\boldsymbol{x}}$. The reconstructed approximations $\hat{\boldsymbol{x}}$ are shown in Fig. 6.4. Note that $\boldsymbol{L}$ is not square to illustrate the effect of weak completeness. We also show decomposition results using the DCT with the DCT coefficients shown in Fig. 6.5 (a), and the truncated series expansions of length $K = 32$ and $K = 64$ in Fig. 6.5 (b) and (c), respectively. The resulting MSE normalised to the energy of the input signal $\boldsymbol{x}$ are tabulated in Table 6.1.

Comparing the series expansions of $\xi = 0.1$ and $\xi = 0.6$ in Fig. 6.4, we note that, for small K, the higher value of ξ results in more reconstructed signal beyond K. From Table 6.1, we note that for $K = 16, 32$, the Laguerre transform has less MSE than the DCT. Note also that for $K = N = 128$, the MSE for $\xi = 0.6$ is the greatest since the inverse transform used in synthesis deviates the most from being unitary. This effect will be clearly seen in the image compression examples.

Image compression

Suppose a given 128×128 image to be compressed $\boldsymbol{x}$ depicted in Fig. 6.2 (b). It is a portion of a German 10 *Pfennig* stamp showing aeroplanes parked at a hexagonal terminal. We



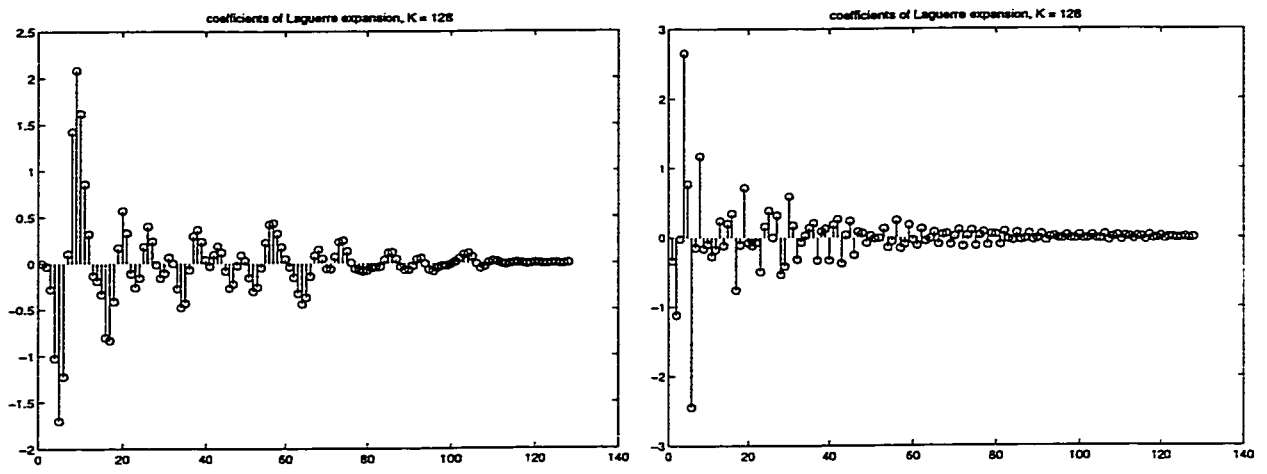
(a) (b)

Figure 6.2: Given signals $\boldsymbol{x}$: (a) Impulse response length 128; (b) 128 × 128 image.

Table 6.1: Mean-squared error [dB] for 1-D Laguerre and DCT compression.

Comp.	K	$\xi = 0.1$	$\xi = 0.6$	DCT
1:8	16	-6.8	-7.5	-1.9
1:4	32	-9.1	-12.9	-8.1
1:2	64	-16.5	-21.0	-27.0
1:1	128	-51.2	-36.1	-298.8

will apply the Laguerre transform to each of the columns of $\boldsymbol{x}$. We will show the Laguerre coefficients and resulting series expansion side by side to better portray the transform. The Laguerre coefficients and resulting expansions are shown for $\xi = 0.1$ in Figs. 6.6, 6.7 and 6.8; and those for $\xi = 0.6$ in Figs. 6.9, 6.10 and 6.11. To correctly compare these results to the DCT examples, we perform a DCT transform on each of the columns of $\boldsymbol{x}$ since the Laguerre transform is not *separable* as is the DCT. The results using the DCT are shown in Fig. 6.12 and Fig. 6.13 with the DCT coefficients in (a) and K term truncated series expansions in (b). Note that, typically, the DCT is performed on image subblocks, say 16 × 16, and the column transform approach was used solely to obtain a comparable MSE expression. The resulting MSE normalised to the energy of the input signal $\boldsymbol{x}$ are tabulated in Table 6.2.



(a) (b)

Figure 6.3: Laguerre coefficients, $K = 128$: (a) $\xi = 0.1$, (b) $\xi = 0.6$.

First we note again the effect of ξ on the ability of reconstructing beyond the series length K. For $\xi = 0.1$ in Fig. 6.6 and 6.7, we note that there is not much reconstruction beyond K. This contrasts with the results obtained with $\xi = 0.6$ in Figs. 6.9 and 6.10, where the reconstructed image has more than just K rows.

When $K = 128$, the reconstructed image with $\xi = 0.6$ is less sharp near the end (lower rows of the image) than with $\xi = 0.1$. This can be seen by the amount of blurring on the letters *UGHAFEN*.

The DCT reconstructed images are uniformly blurred, whereas the Laguerre transformed ones all resolve the tail of the top plane and the valuation value of *10*. If the image has more detail near one edge, the Laguerre transform on the correctly oriented image has superior performance to the DCT. However, from Table 6.2, the length $K = 128$ Laguerre reconstructions cannot reach very low MSE because of the error in computing the inverse (synthesis) transform.

To summarise the results, for a given $\xi > 0$, the Laguerre transform should be used with the intent of keeping less coefficients in the transform domain than in the non-transformed domain ($K < N$). In this way, the orthogonality condition is maintained at the expense of completeness. The unitary property is thus weakened and the inverse is not complete. As $\xi \rightarrow 0$, completeness is strengthened and the inverse matrix has less error. Once N and K

Table 6.2: Mean-squared error [dB] for 2-D Laguerre and DCT compression.

Comp.	K	$\xi = 0.1$	$\xi = 0.6$	DCT
1:4	32	-1.5	-13.0	-15.8
1:2	64	-4.5	-17.8	-20.4
1:1	128	-29.5	-20.7	-300.7

are fixed, the value of ξ should be adjusted to yield low MSE and/or low synthesis error, depending on the application.

6.6 Chapter summary

In this chapter we have established the notation for the Laguerre transform and motivated its use by examples. The Laguerre transform matrix need not be square and thus one of the two conditions for unitariness, orthogonality and completeness, has to be compromised. For lossy signal compression, we are more interested in orthonormality than completeness as we accept a certain amount of distortion at the synthesis stage.

The non-uniform resolution seen in the image compression example stems from the frequency warping effect of the Laguerre filter. It is this effect which we will investigate next.

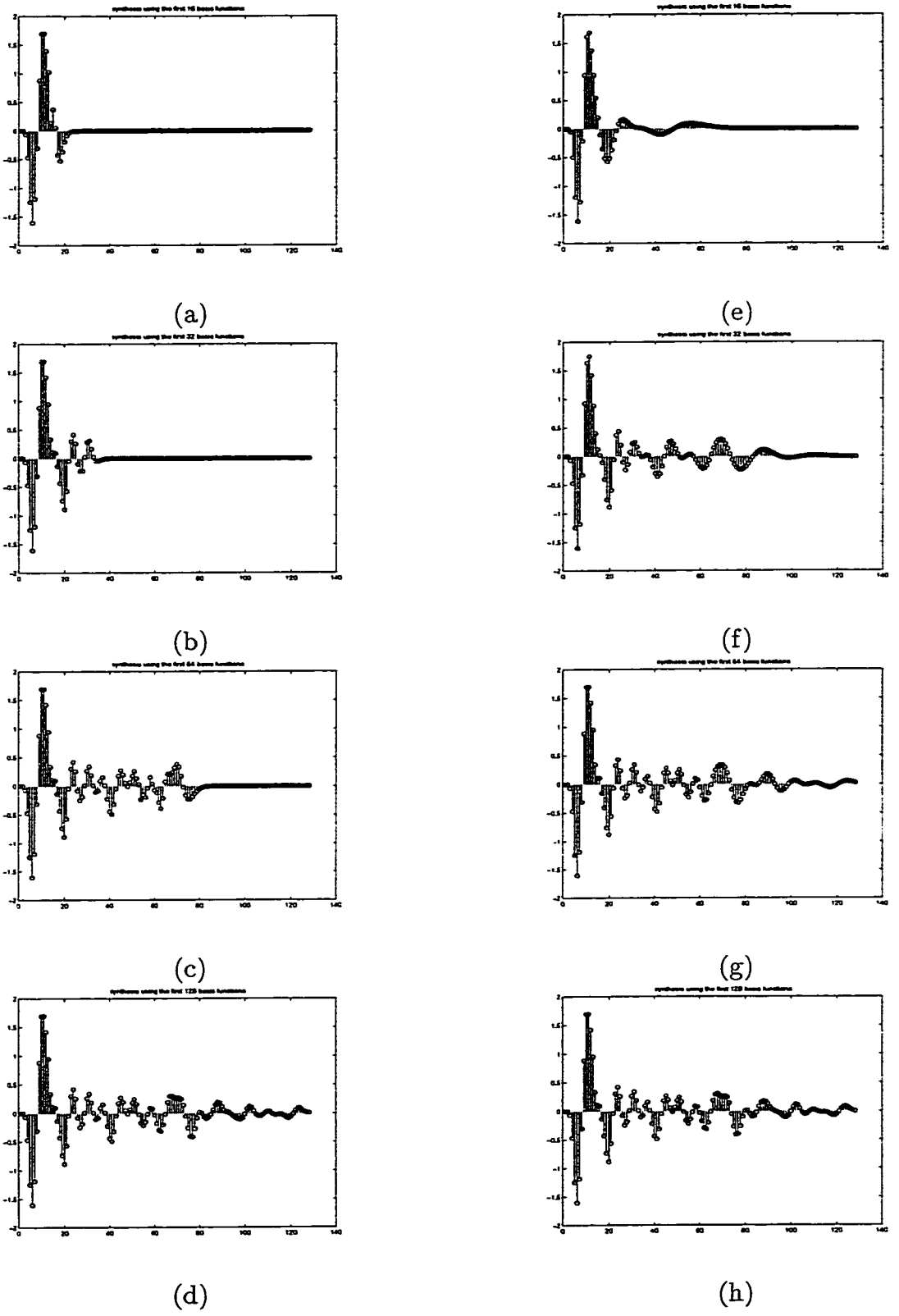
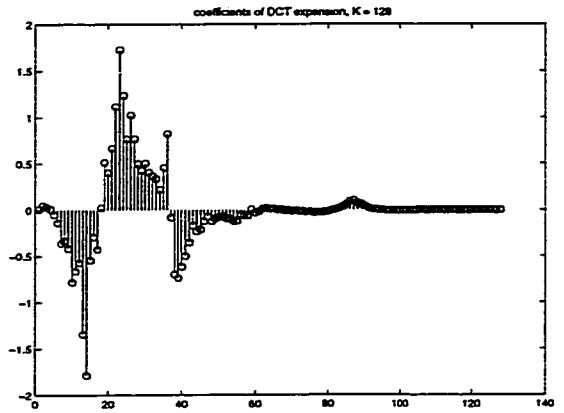
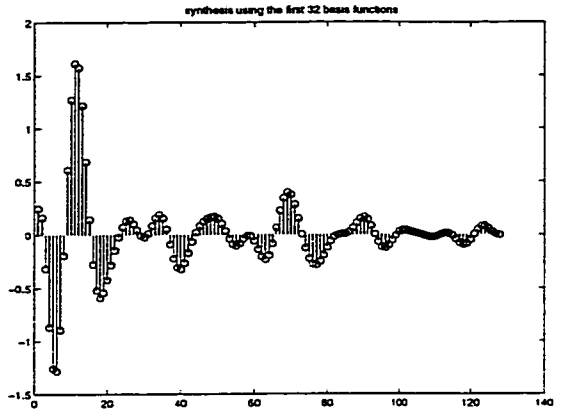


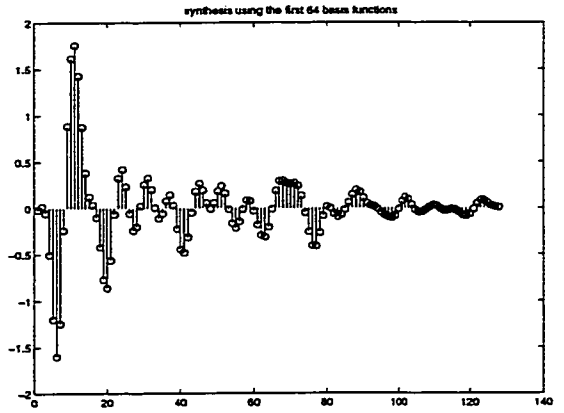
Figure 6.4: K-term Laguerre series expansion $\hat{\boldsymbol{x}}$: Left-hand column: $\xi = 0.1$, Right-hand column: $\xi = 0.6$, (a),(e): $K = 16$, (b),(f): $K = 32$, (c),(g): $K = 64$, (d),(h): $K = 128$.



(a)

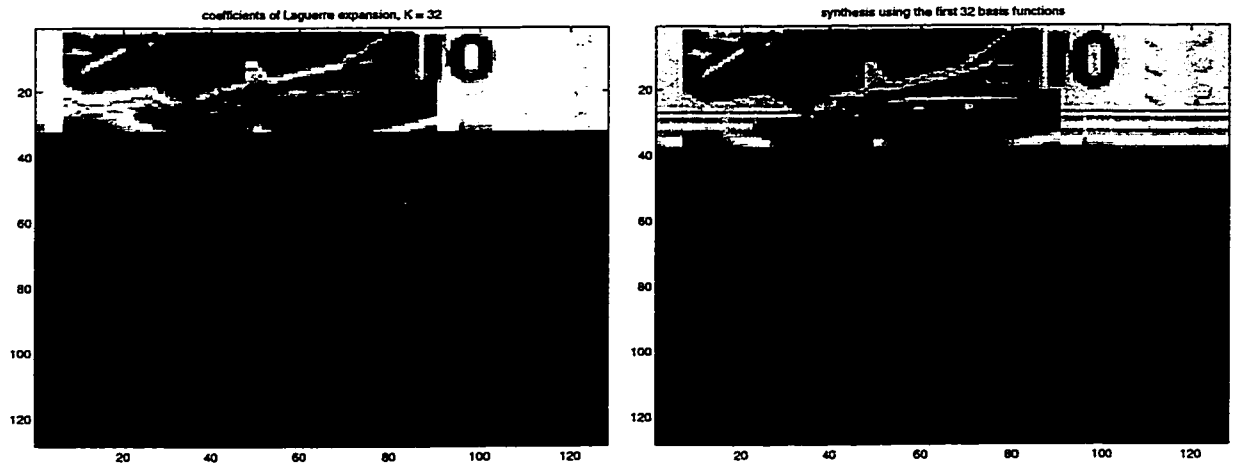


(b)



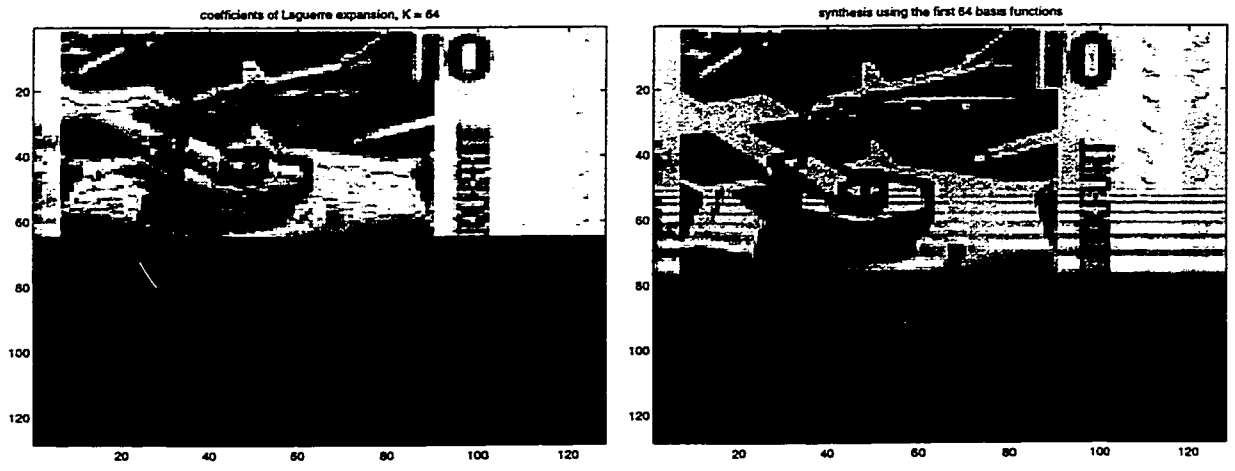
(c)

Figure 6.5: (a) DCT coefficients, $K = 128$; (b) 32-term DCT series expansion $\hat{\boldsymbol{x}}$; (c) 64-term DCT series expansion $\hat{\boldsymbol{x}}$.



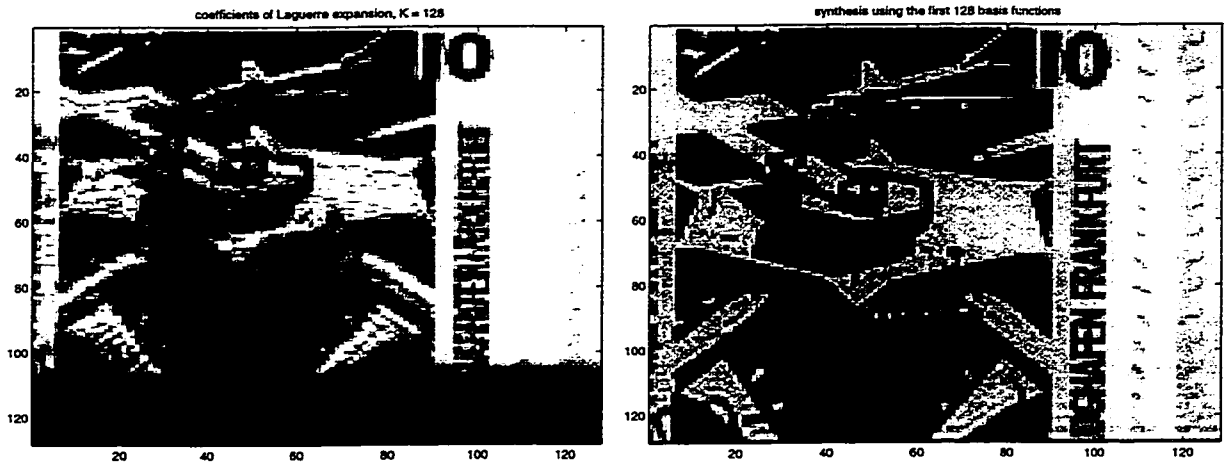
(a) (b)

Figure 6.6: For $\xi = 0.1, K = 32$: (a) First K Laguerre coefficients $\tilde{\boldsymbol{x}}$; (b) K-term Laguerre series expansion $\hat{\boldsymbol{x}}$.



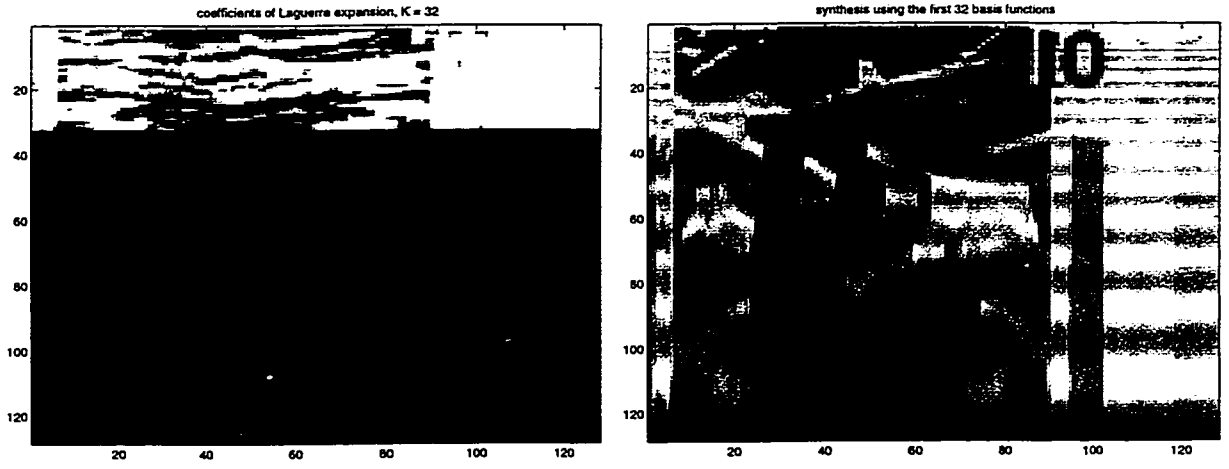
(a) (b)

Figure 6.7: For $\xi = 0.1, K = 64$: (a) First K Laguerre coefficients $\tilde{\boldsymbol{x}}$; (b) K-term Laguerre series expansion $\hat{\boldsymbol{x}}$.



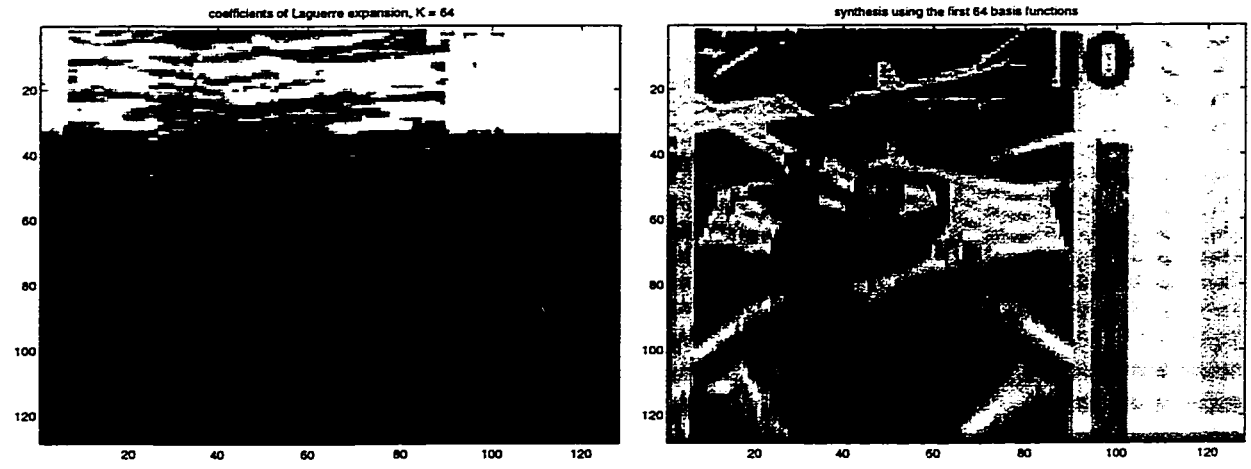
(a) (b)

Figure 6.8: For $\xi = 0.1, K = 128$: (a) First K Laguerre coefficients $\tilde{\boldsymbol{x}}$; (b) K-term Laguerre series expansion $\hat{\boldsymbol{x}}$.



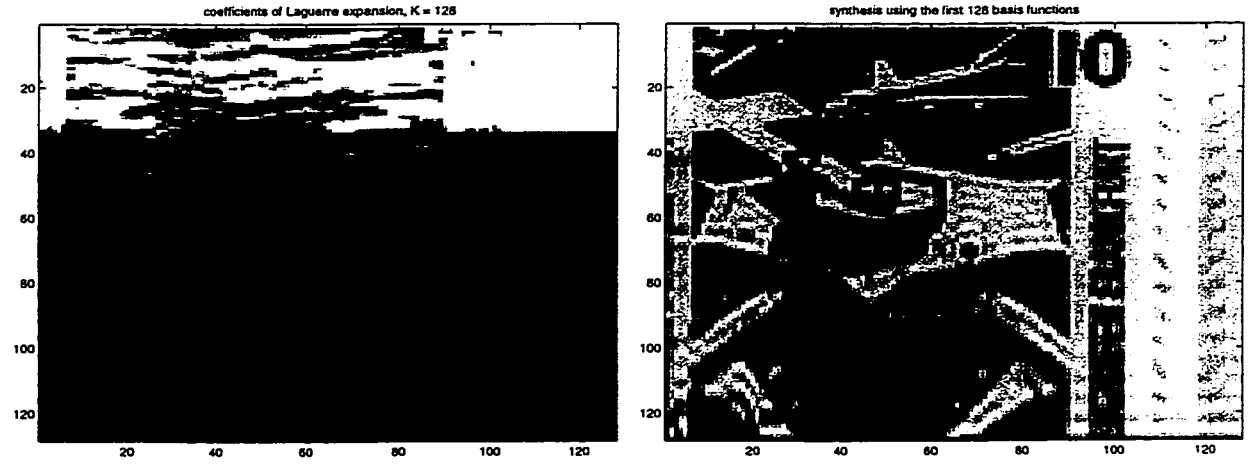
(a) (b)

Figure 6.9: For $\xi = 0.6, K = 32$: (a) First K Laguerre coefficients $\tilde{\boldsymbol{x}}$; (b) K-term Laguerre series expansion $\hat{\boldsymbol{x}}$.



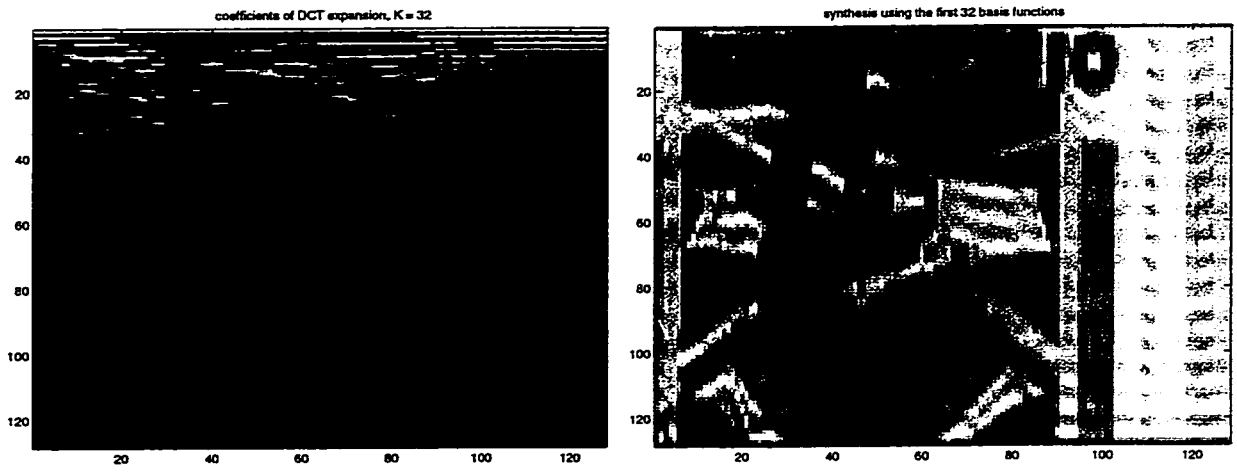
(a) (b)

Figure 6.10: For $\xi = 0.6, K = 64$: (a) First K Laguerre coefficients $\tilde{\boldsymbol{x}}$; (b) K-term Laguerre series expansion $\hat{\boldsymbol{x}}$.



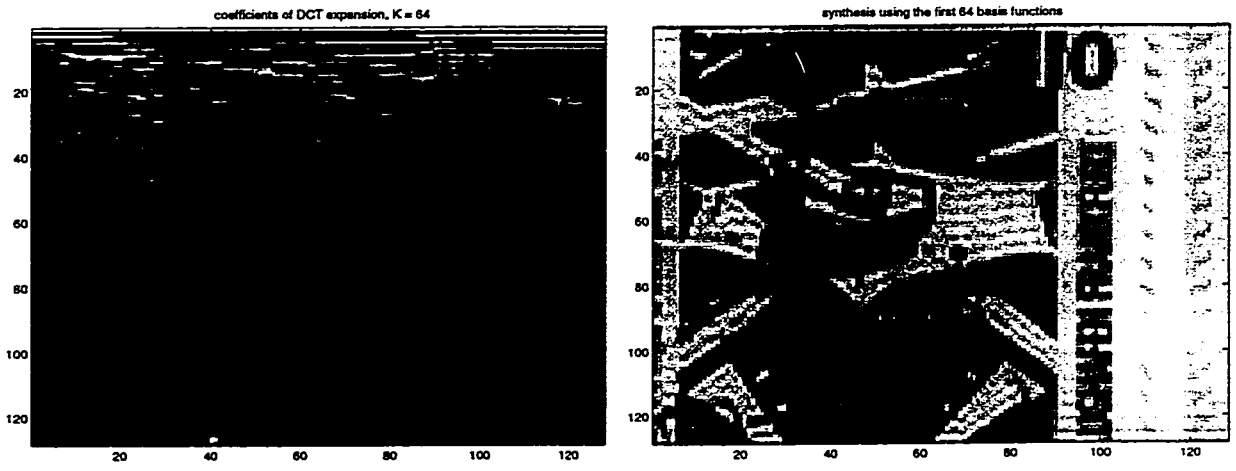
(a) (b)

Figure 6.11: For $\xi = 0.6, K = 128$: (a) First K Laguerre coefficients $\tilde{\boldsymbol{x}}$; (b) K-term Laguerre series expansion $\hat{\boldsymbol{x}}$.



(a) (b)

Figure 6.12: (a) 32 DCT coefficients $\tilde{\boldsymbol{x}}$; (b) 32-term DCT series expansion $\hat{\boldsymbol{x}}$.



(a) (b)

Figure 6.13: (a) First 64 DCT coefficients $\tilde{\boldsymbol{x}}$; (b) 64-term DCT series expansion $\hat{\boldsymbol{x}}$.

Chapter 7

Time and frequency warping

The Laguerre filter has been used to obtain a controlled distortion of the frequency response. The goal was to obtain unequal bandwidth spectral analysis in conjunction with the FFT [25]. Applications included linear prediction on a warped frequency axis [29] and loudspeaker equalisation [30]. After having discussed the frequency warping effect of a Laguerre filtered sequence, the notation for frequency warping is given followed by an example.

The dual to frequency warping is time warping. Its realisation is plagued by the boundary effects of the DFT. However the notation can be established and possible applications are given.

7.1 Frequency warping

In speech coding, a motivation to use frequency warped signal is that formant bandwidth increases with frequency, and that the frequency resolution of the ear at high frequency is weaker than at low frequencies [29]. The same argument may be made for loudspeaker equalisation [30].

The DFT of a signal produces equally spaced samples around the unit circle (6.26). Thus the frequency analysis consists of equidistantly spaced samples in frequency (also called bins). Suppose we require bandwidth resolution proportional to frequency, that is, we want high resolution at low frequencies and low resolution at high frequencies. This effect has been labeled *warping* of the frequency axis but is in fact just a Laguerre transformed signal. It

turns out that there is a relation between the scale used by the auditory system and the Laguerre parameter ξ.

7.1.1 Psychoacoustics

The critical band scale divides the frequency scale from 20 Hz to 20 kHz into 24 *critical bands*. The unit of width of one critical band is the Bark. Bark bandwidths are about 100 Hz below 1 kHz and increase almost logarithmically thereafter reaching a width of 3 kHz at 20 kHz. The mapping from frequency f [kHz] to Bark index [6] is given by

$$z = 13\tan^{-1}(0.76f/7) + 3.5\tan^{-1}(f/7.5)^2. \tag{7.1}$$

Smith and Abel derived an expression for ξ which best approximates the psychoacoustic Bark scale (Zwicker) as a function of sampling frequency f_s [6]:

$$\xi \approx 1.0211(2/\pi\tan^{-1}(0.076f_s))^{1/2} - 0.19877. \tag{7.2}$$

With $f_s = 44.1$ kHz, we have that $\xi = 0.72$ [6]. With $f_s = 10$ kHz, $\xi = 0.47$ [29].

7.1.2 Unequal bandwidth spectral analysis

Braccini and Oppenheim [31] [25] define frequency warping as a transformation which relates an original, x, and a transformed sequence $\tilde{x}$, such that the Fourier transform of the transformed sequence has unequal bandwidth resolution. Denoting X the DFT of x, the Fourier transforms of both sequences are related by [25]

$$X(e^{j\omega}) = \tilde{X}(e^{j\tilde{\omega}}) \tag{7.3}$$

where the function $\tilde{\omega} = \tilde{\omega}(\omega)$ performs a controlled distortion of the frequency axis called *warping*. It turns out that the Laguerre filter realises this transform and the governing equations are given as [29] [25]

$$\tilde{z}^{-1} = \frac{z^{-1} - \xi}{1 - \xi z^{-1}} \qquad 0 < \xi < 1 \tag{7.4}$$

$$\tilde{\omega} = \omega + 2\tan^{-1}\left(\frac{\xi \sin(\omega)}{1 - \xi\cos(\omega)}\right) \tag{7.5}$$

$$\frac{d\tilde{\omega}}{d\omega} = \frac{1 - \xi^2}{1 + \xi^2 - 2\xi\cos\omega} \tag{7.6}$$

where $z = e^{j\omega}$ and $\omega \in [0, 2\pi]$.

7.1.3 Allpass filter

Equation (7.4) represents the transfer function of a first-order allpass (AP) filter and we show this by letting $\tilde{z} = A(z)$. We will examine some of the properties of the first-order AP filter.

A first-order AP filter is a stable IIR filter with impulse response

$$a[n, \xi] = \begin{cases} -\xi & \text{if } n = 0 \\ (1 - \xi^2)\xi^{n-1} & \text{otherwise} \end{cases} \tag{7.7}$$

and is a *minimum phase* filter (cf. Appendix D).

The AP filter is a unitary filter in that it conserves energy. Let $x[n]$ be the input and $y[n]$ be the output of the AP filter. Then

$$\|y\|^2 = \frac{1}{2\pi}\|Y(e^{jw})\|^2 = \frac{1}{2\pi}\|A(e^{jw})X(e^{jw})\|^2 = \frac{1}{2\pi}\|X(e^{jw})\|^2 = \|x\|^2. \tag{7.8}$$

7.1.4 Laguerre filter

Oppenheim and Braccini, and Strube both use the Laguerre filter to realise the frequency transform but omit to mention the name Laguerre. The filter is exactly as shown in Fig. 5.3. A plot of (7.5) is shown for $\xi = -1/2, 0, 1/4, 1/2$ in Fig. 7.1 (a).

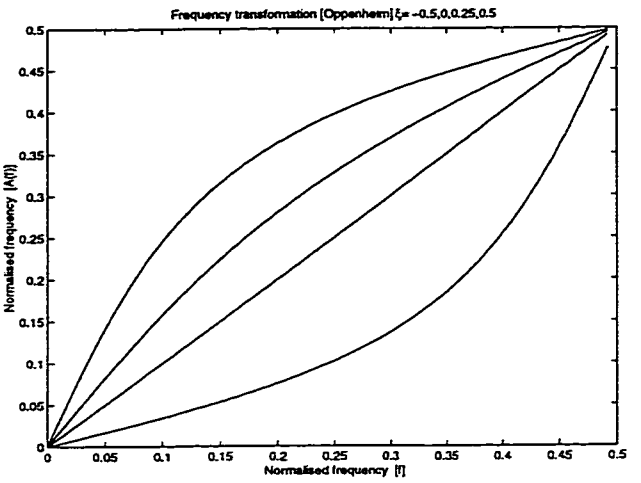


Figure 7.1: Frequency transformation in (7.5) for $\xi = -1/2, 0, 1/4, 1/2$.

Note that $\tilde{\omega}$ is an odd function of ξ. Thus the inverse mapping is realised by negating ξ. This is mentioned by both Strube an Oppenheim. As can be seen from Fig. 7.1, $\xi > 0$ skews the spectrum towards π rad whereas $\xi < 0$ skews the spectrum towards 0 rad.

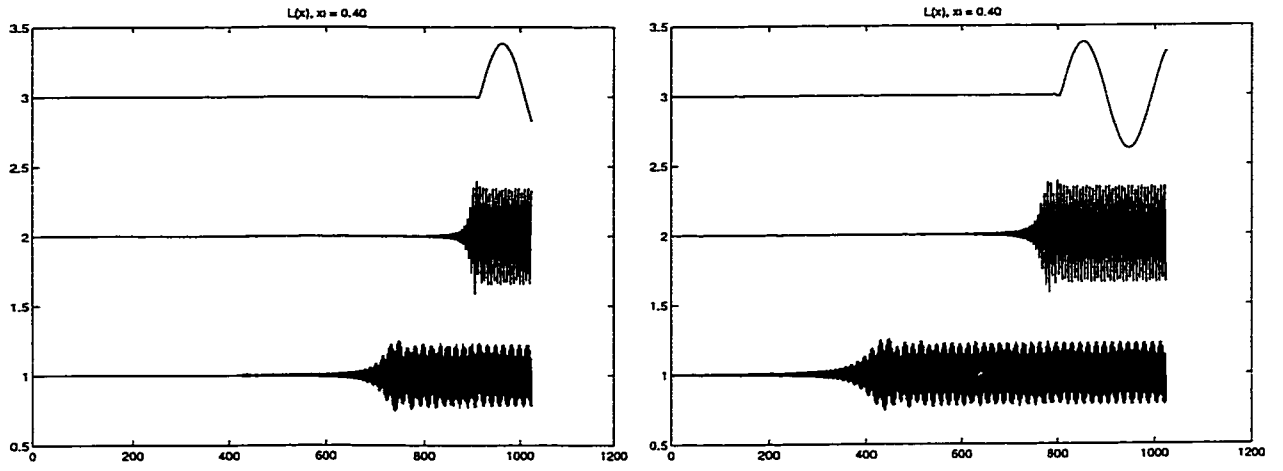
7.1.5 Example

In order to visualise the frequency warping effect, a DFT of the vector length K formed by the K taps of the Laguerre filter (refer to Fig. 5.3) is taken. This implies the implicit use of a rectangular window on the samples. The spectral window applied to the transformed sequence maintains a constant bandwidth on the linear frequency scale ω. This results in spectral smearing viewed with respect to the warped frequency axis $\tilde{\omega}$. In fact, from (7.5) with $\xi > 0$, the bandwidth of the window increases with frequency just as the spacing of the frequency bins increases with frequency. Therefore, the result obtained by taking a DFT results in increased spectral smearing with increasing frequency[31].

In this example, the input signals were sine waves of length 1024 with frequencies 100 Hz, 300 Hz and 10 kHz, at f_s=44.1 kHz. These signals were passed through a $K = 1024$ tap Laguerre filter and the length K output vector, formed by the filter taps, as in Fig. 5.3, was recorded after the 1024th sample entered the filter. We also show the output vector after the 256th and 512th sample entered the Laguerre filter to study the filter transient behaviour.

The output of the Laguerre filter with $\xi = 0.4$ after the 256th and 512th input sample are

shown in Fig. 7.2 (a) and (b). For $\xi = -0.4$ the same outputs are shown in Fig. 7.3 (a) and (b). With the 1024 samples having entered the Laguerre filter, the time and corresponding frequency transform of the input vector and the Laguerre filter output vector are shown in Figs. 7.4, 7.5 and 7.6.

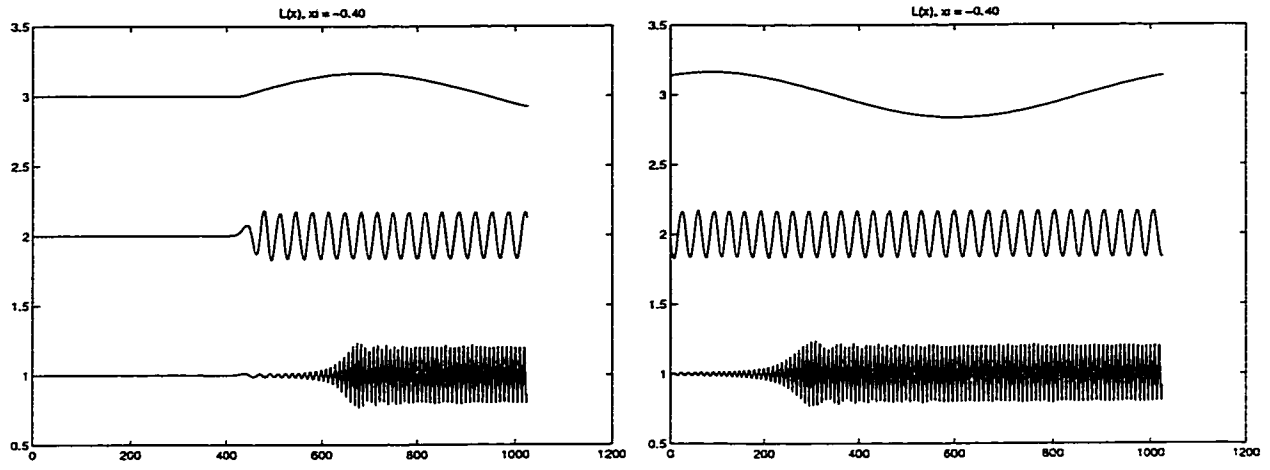


(a) (b)

Figure 7.2: Laguerre output vector after (a) 256 and (b) 512 samples input, $\xi = 0.4$.

We discuss the time and frequency plots separately. From the time waveforms in Figs. 7.2, 7.3 and 7.4 (a), and both values of ξ, we notice that there is some peaking visible for the higher frequency signals. For $\xi = 0.4$, we notice that the 100 Hz tone has propagated less through the filter than the higher frequency tones. The input signals appear to have been delayed depending on the input signal frequency. That is, the higher frequency signals appear in time before the lower frequency signals. From Fig. 7.5 (a) the 100 Hz signal still has the same number of cycles after the 1024th input sample was input to the filter. This implies that the signal frequency with respect to the sampling rate is higher. This is perceived as an increase in pitch. In fact, the pitch change depends on the signal frequency and this has been verified by processing speech and music signals.

For $\xi = -0.4$, after the 256th input entered the filter, we see the initial zero crossing of the 100 Hz tone propagate to the left together with the 300 Hz tone. Note that after 256 input samples there is no output on the first 425 taps. The time signals also appear to oscillate less. It suggests that not all of the signal has traversed the Laguerre filter. This



(a) (b)

Figure 7.3: Laguerre output vector after (a) 256 and (b) 512 samples input, $\xi = -0.4$.

has been verified to result in a reduction in pitch. Again, it has been verified that the pitch reduction is a function of the original signal frequency.

The frequency responses in Figs. 7.4 (b), 7.5 (b) and 7.6 (b) depict the frequency warping effect. For $\xi = 0.4$ the signals appear to have been skewed to π rads whereas for $\xi = -0.4$ the signals appear to have been skewed towards 0 rads.

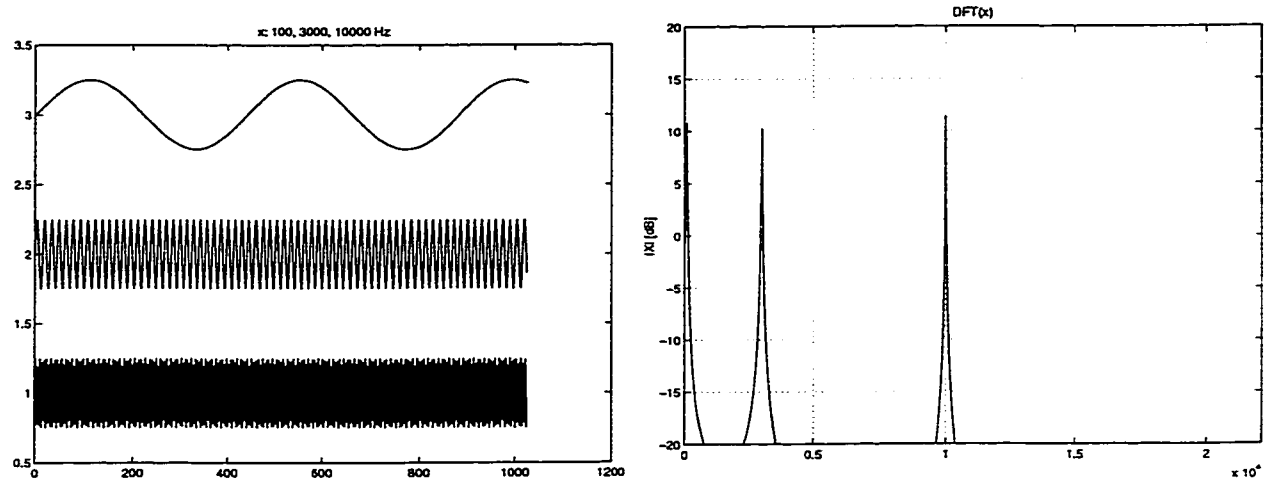
7.1.6 Matrix notation

To realise frequency warping with extended resolution in the low frequencies and reduction of resolution in the high frequencies, we transform a vector in the time domain to the Laguerre domain using the Laguerre transform. Given a column vector $\boldsymbol{x} = \{x[n]\}, n = 0, 1, \ldots, N-1$, and denoting the Laguerre domain representation as $\tilde{\boldsymbol{x}}$,

1. Pass the vector $\boldsymbol{x}$ through a Laguerre filter with parameter $\xi > 0$ or

2. Premultiply by the analysis matrix $\boldsymbol{\Psi}_a(\xi) = \boldsymbol{L}(\xi)$

$$\tilde{\boldsymbol{x}} = \boldsymbol{L}(\xi)\boldsymbol{x} \tag{7.9}$$



(a) (b)

Figure 7.4: Input signals: (a) time and (b) frequency response

To return to the time domain from the Laguerre domain,

1. Pass the vector $\boldsymbol{x}$ through a Laguerre filter with parameter $-\xi$ or
2. Premultiply by the synthesis matrix $\boldsymbol{\Psi}_s(\xi) = \boldsymbol{L}^T(\xi)$

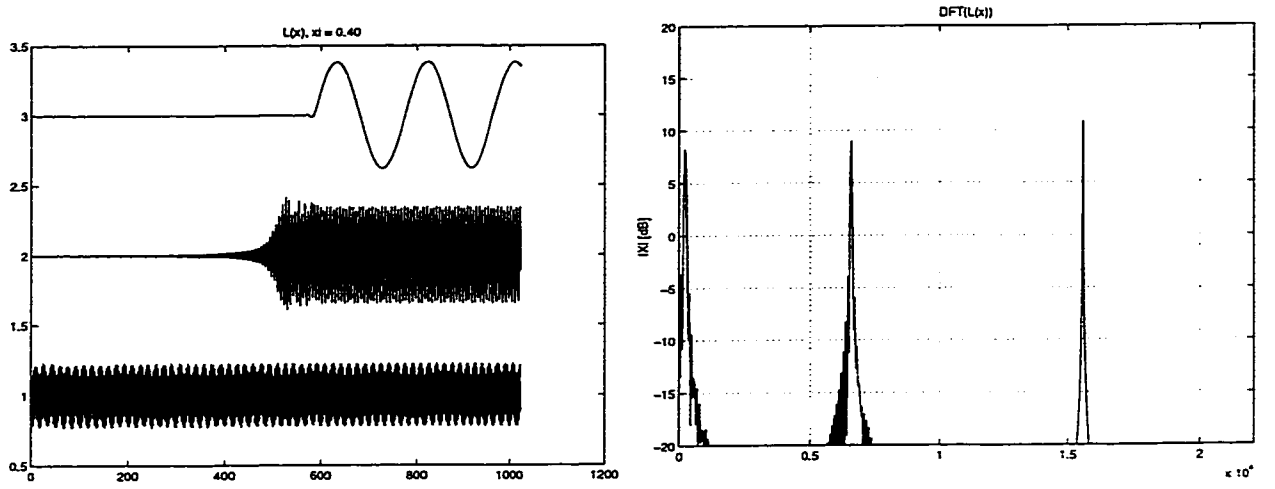
$$\boldsymbol{x} = \boldsymbol{L}^T(\xi)\tilde{\boldsymbol{x}} \tag{7.10}$$

Recall from (6.43) that the inverse of the Laguerre transform matrix with parameter $\xi > 0$ is obtained by either computing the Laguerre transform matrix with parameter $-\xi$ or by taking its transpose.

7.2 Time warping

The dual to warping in the frequency domain is warping in the time domain. This can be accomplished by transforming a signal in the Fourier domain by the Laguerre transform. The result can be described as a *nonuniform resampling* of a signal.

Work on a discrete theory of irregular sampling has been published by Gröchenig [32], with applications published by Feichtinger and his student Strohmer [33]. These are members



(a) (b)

Figure 7.5: Laguerre output signals, $\xi = 0.4$: (a) time and (b) frequency response

of the Numerical Harmonic Analysis Group, Department of Mathematics, University of Vienna, Austria. We also mention contributions by Mark [34] and Steenart [35].

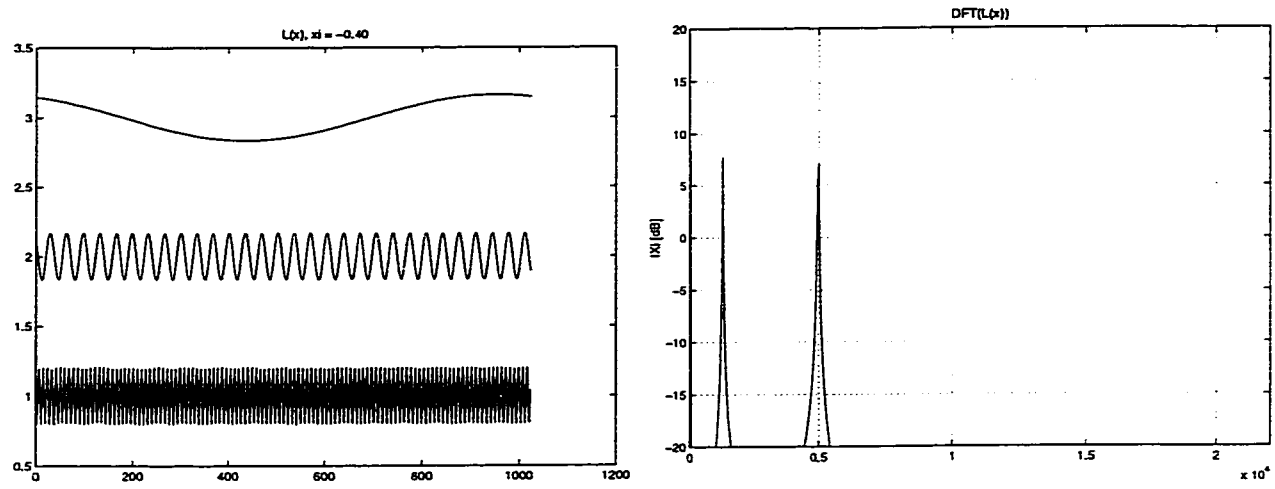
The main applications of nonuniform sampling are in reconstruction of damaged or noisy signals such as in video and speech. Time warping would skew the signal in time in order to, say, have rarefication of samples at one end of a block and concentration of samples at the other.

The transform does not lend itself to signals with time variance. Thus a signal of finite length would be processed as a single block. We motivate the idea of time warping by giving the matrix notation.

7.2.1 Matrix notation

To realise time warping with extended resolution at the beginning and reduction of resolution at the end of a time sequence, we transform a vector in the frequency domain to the Laguerre domain using the Laguerre transform. Given a forward Fourier transform of a column vector $\boldsymbol{X} = DFT(\boldsymbol{x}) = \{X[n]\}, n = 0, 1, \ldots, N-1$, and denoting the Fourier transformed Laguerre transformed representation as $\tilde{\boldsymbol{X}}$,

1. Pass the vector $\boldsymbol{X}$ through a Laguerre filter with parameter $\xi > 0$ or



(a) (b)

Figure 7.6: Laguerre output signals, $\xi = -0.4$: (a) time and (b) frequency response

2. Premultiply by the analysis matrix $\boldsymbol{\Psi}_a(\xi) = \boldsymbol{L}(\xi)$

$$\tilde{\boldsymbol{X}} = \boldsymbol{L}(\xi)\boldsymbol{X} \tag{7.11}$$

To obtain the inverse effect, that is of extended resolution at the tail of the sequence, or to return from the Laguerre domain,

1. Pass the vector $\tilde{\boldsymbol{X}}$ through a Laguerre filter with parameter $-\xi$ or

2. Premultiply by the synthesis matrix $\boldsymbol{\Psi}_s(\xi) = \boldsymbol{L}^T(\xi)$

$$\boldsymbol{X} = \boldsymbol{L}^T(\xi)\tilde{\boldsymbol{X}} \tag{7.12}$$

To complete the time warping, we apply the IDFT to $\boldsymbol{X}$ to obtain the time vector $\boldsymbol{x}$, $\boldsymbol{x} = IDFT(\boldsymbol{X})$.

7.3 Chapter summary

The Laguerre filter skews the frequency response depending on the Laguerre parameter ξ. This effect may be useful in adaptive filtering where the adaptive filter would be designed in the Laguerre domain. For example, if greater resolution in the low frequencies and less precision in the high frequencies is desired, a Laguerre adaptive filter could realise this. In the following two chapters, we examine the Laguerre filter first as a least-squares solution and then as an adaptive filter.

Chapter 8

Least-squares results

In this chapter we investigate the minimum MSE surface when using a Laguerre filter. The least-squares algorithm is used as it approaches the Wiener solution without a stochastic model. We review past results and investigate the error surface obtained from a ventilation duct impulse response. Such an impulse response is typical of acoustic impulse responses as the impulse decays exponentially to zero and is relatively long.

8.1 Least-squares algorithm

The least-squares algorithm assumes no stochastic model and is data driven. The estimate of the desired response is modeled by a K tap FIR filter $\{w_k\}$

$$\hat{d}[n] = \sum_{k=0}^{K-1} w_k^* u[n-k] \tag{8.1}$$

where $\{u[i]\}, \{d[i]\}, i = 1, \ldots, N$ are observed samples.

The least-squares criterion is to minimise the mean-squared error based on actual error observations over a window length $i_2 - i_1 + 1 = M$

$$\varepsilon(w) = \sum_{i_1}^{i_2} |e[n]|^2 \tag{8.2}$$

where $e[n] = d[n] - \sum_{k=0}^{K-1} w_k^* u[n-k]$ and, for example, $i_1 = K, i_2 = N$ which implies $M = N - K + 1$.

Define the M-tuple $\boldsymbol{b} = \{d[i_1]d[i_1+1]\ldots d[i_2]\}$ and the K column vectors of length M which we will denote as the observation windows

$$\phi_k = [u[i_1-k]u[i_1-k+1]\ldots u[i_2-k]]^T \qquad k = 0, \ldots, K-1 \tag{8.3}$$

Then let $\mathcal{S}$ be spanned by these K observation windows: $\mathcal{S} = span(\phi_0, \phi_1, \ldots, \phi_{K-1})$.

Now assume the observation windows are linearly independent. Then, by the projection theorem, the projection of $\boldsymbol{b}$ onto $\mathcal{S}$ is the least-squares estimate of $\boldsymbol{b}$ and the error $\boldsymbol{b} - \hat{\boldsymbol{b}}$ will be orthogonal to $\mathcal{S}$.

We then form the deterministic autocorrelation and crosscorrelation matrices Φ and Θ,

$$\Phi = \begin{bmatrix} \langle \phi_0, \phi_0 \rangle & \cdots & \langle \phi_0, \phi_{K-1} \rangle \\ \vdots & & \vdots \\ \langle \phi_{K-1}, \phi_0 \rangle & \cdots & \langle \phi_{K-1}, \phi_{K-1} \rangle \end{bmatrix} \tag{8.4}$$

$$\Theta = \begin{bmatrix} \langle \phi_0, \boldsymbol{b} \rangle \\ \vdots \\ \langle \phi_{K-1}, \boldsymbol{b} \rangle \end{bmatrix} \tag{8.5}$$

and solve $\Phi \hat{w} = \Theta$.

To compute the deterministic correlation matrices from the observed samples, we construct the $M \times K$ matrix $\boldsymbol{A}$ as:

$$\boldsymbol{A} = \begin{bmatrix} \Phi_0 & \Phi_1 & \ldots & \Phi_{K-1} \end{bmatrix} = \begin{bmatrix} u[N-M+1] & u[N-M] & \ldots & u[1] \\ u[N-M+2] & u[N-M+1] & \ldots & u[2] \\ \vdots & \vdots & & \vdots \\ u[N] & u[N-1] & \ldots & u[M] \end{bmatrix} \tag{8.6}$$

and write $\Phi = \boldsymbol{A}^{*T}\boldsymbol{A}$, $\Theta = \boldsymbol{A}^{*T}\boldsymbol{d}$ where $\boldsymbol{d}$ is a $M \times 1$ vector $[d^*[i_1]\ldots d^*[i_2]]^T$. The *deterministic normal equations* are then written as

$$\boldsymbol{A}^{*T}\boldsymbol{A}\boldsymbol{w} = \boldsymbol{A}^{*T}\boldsymbol{d} \tag{8.7}$$

and the expression for the minimum mean-squared error then becomes

$$\varepsilon_{\min} = \|\boldsymbol{b} - \hat{\boldsymbol{b}}\|^2 \tag{8.8}$$

$$= \|\boldsymbol{b}\|^2 - \langle \boldsymbol{b}, \hat{\boldsymbol{b}} \rangle \tag{8.9}$$

$$= \|\boldsymbol{b}\|^2 - \Theta^{*T}\boldsymbol{w} \tag{8.10}$$

8.1.1 Constructing matrix A

If the samples are stored consecutively in memory, then we can construct $\boldsymbol{A}$ by taking length M vectors and placing them as columns. However, if the outputs are obtained from the taps of a filter based on a TDL (e.g. the Laguerre filter), the matrix $\boldsymbol{A}$ has to be formed by placing the length K tap vector output at time n in row n. If the input to the filter was white, then the K columns would be linearly independent.

8.2 Least-squares Laguerre

In two papers by Oliveira e Silva [26] [14], the optimal pole position problem is considered. Using a least-squares estimate of a system identification problem, as depicted in Fig. D.1, the mean-squared error (MSE) was plotted against ξ.

In both experiments, white Gaussian noise of mean 0 and variance 1 was filtered by a filter $X(z)$ and then passed through a plant $H(z)$.

8.2.1 System impulse responses

The system transfer functions, and their respective poles ane zeros, are shown in Table 8.1 and 8.2. The first system is composed of two LP filters and the second one of two HP filters.

Table 8.1: Low-pass system transfer functions.

	transf. func.	zeros	poles
$X(z)$	$\frac{1}{1-0.5z^{-1}}$	$z = 0$	$z = 0.5$
$H(z)$	$\frac{1-1.2z^{-1}+z^{-2}}{(1+0.7z^{-1})(1-1.8z^{-1}+0.9z^{-2})}$	$z = 0.6 \pm j0.8$	$z = -0.7, 0.9 \pm j0.3$

Table 8.2: High-pass system transfer functions.

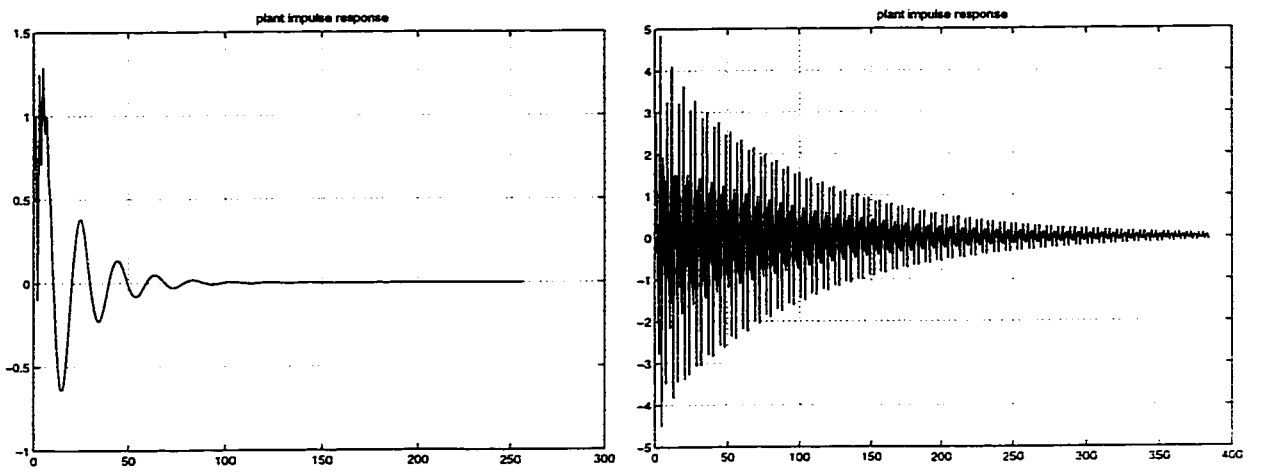
	transf. func.	zeros	poles
$X(z)$	$\frac{2+0.2z^{-1}}{1+0.6z^{-1}+0.13z^{-2}}$	$z = -0.1$	$z = -0.3 \pm j0.2$
$H(z)$	$\frac{3+6.4z^{-1}+2.8z^{-2}}{(1+0.9z^{-1})(1+1.4z^{-1}+0.98z^{-2})}$	$z = -1.5188, -0.6145$	$z = -0.9, -0.7 \pm j0.7$

The impulse responses are shown in Fig 8.1 (a) and (b).

For each transfer function system, the results were reproduced. Furthermore higher filter orders were simulated in order to see if numerical problems would occur and degrade the optimality condition on the weights. The authors used filter order, which for FIR filters is just the number of taps $K - 1$, in their results and thus our results reflect this. For the LP case, filters of orders 1 to 6, and 7 to 12 were simulated. For the HP case, orders 0 to 8, and 9 to 17 were simulated.

Figs 8.2 and Figs 8.3 show the mean-squared error plotted against the Laguerre parameter. Note that for $\xi < 0$ in the LP system, and for $\xi > 0$ for the HP system, the curves approach 0dB and are thus not shown. The reason for the sign difference between the LP and HP systems has not been published as these two examples come from separate papers. However, from the effects of frequency warping, it may be argued that, for the LP system, $\xi > 0$ increased low frequency resolution by skewing the frequency response towards π. This spread the system energy and thus improved the minimum MSE. Conversely, for the HP system, $\xi < 0$ increased resolution of the high frequences where much of the energy was already present and thus improved the minimum MSE.

We note that consecutive MSE curves touch where both have local extrema. This condition occurs when both the length K and $K + 1$ filters have their last coefficient equal to



(a) (b)

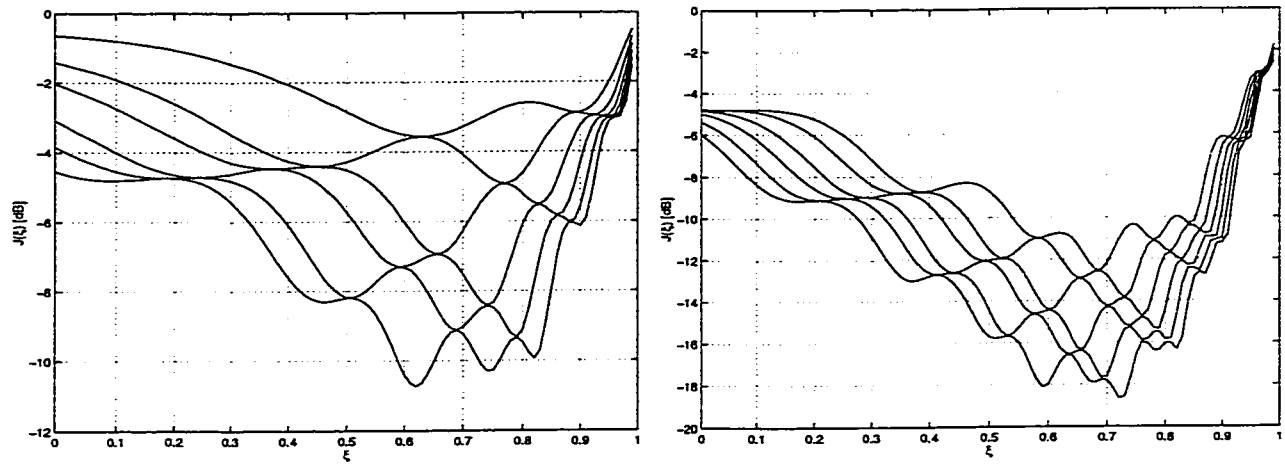
Figure 8.1: Impulse responses $H(z)$ (a) Lowpass plant. (b) Highpass plant.

zero and was stated in (5.39). It should not be inferred however that these extrema always appear in a maximum/minimum pair [26]. For the LP transfer function using (5.40), the Laguerre parameter was estimated to $\hat{\xi} = 0.66$.

8.2.2 Measured impulse response

The same experiment was performed using a duct impulse response as the plant. This impulse reponse was taken from a ventilation duct. The duct impulse response has length 256 and is shown in Fig 8.4(a). As with most acoustic impulse responses, the response decays to zero over a long time interval. The application here would be in modeling such an impulse response with a shorter filter. The minimum MSE was then calculated from the least-squares solution for filter lengths of 16, 32, 64 and 128 for positive values of ξ. The results are shown in Fig 8.4(b).

We note that the curves are flatter than in the previous example and only for higher filter lengths does warping reduce the minimum MSE. The optimal pole estimate using (5.40) gave $\hat{\xi} = 0.33$. Referring to Fig 8.4(b), for $M = 128$ and $\xi = 0.3$, the minimum MSE is about -40 dB and this is 10 dB better than with $\xi = 0$. Thus the Laguerre filter achieves a minimum MSE of -40 dB with half the number of coefficients in the plant impulse response.



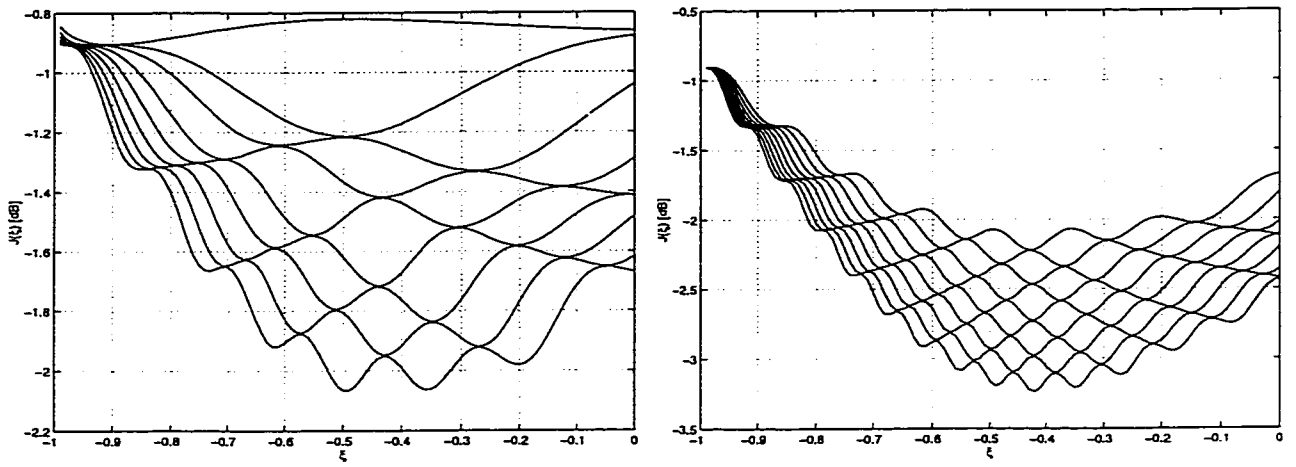
(a) (b)

Figure 8.2: Mean-squared error versus ξ for LP system. (a) Orders 1 to 6. (b) Orders 7 to 12.

8.3 Chapter summary

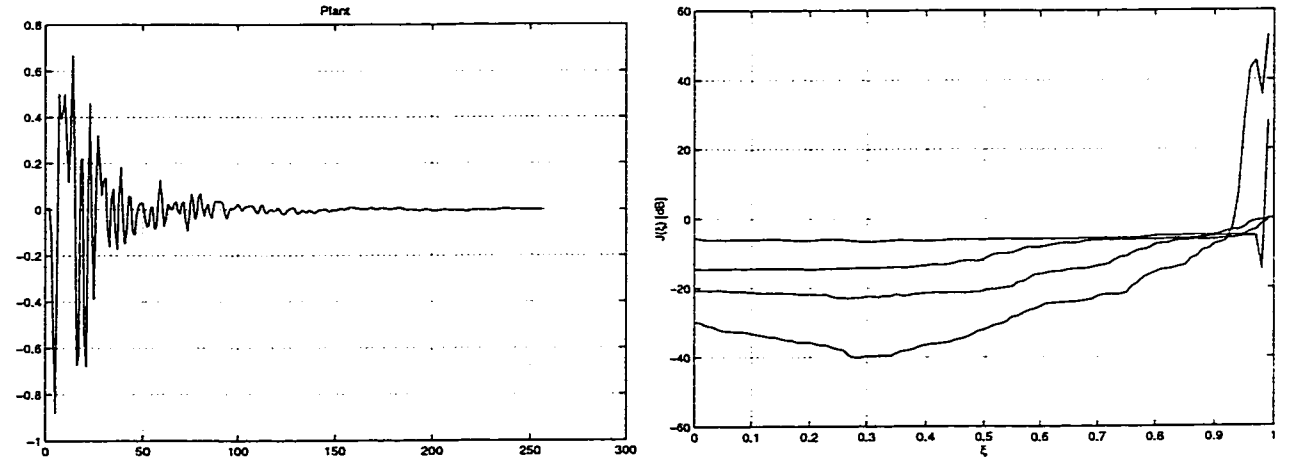
The least-squares solution was used to obtain the minimum MSE resulting from the use of a Laguerre filter. This is possible since the stochastic Laguerre autocorrelation matrix in (5.33) is Toeplitz. For simple transfer functions, the minimum MSE surface as a function of ξ exhibits local minima and finding the minimum becomes a non-linear problem. The frequency response of the plant can be used to estimate the sign of ξ. This follows from the frequency warping effect of the Laguerre filter. A positive value for ξ characterises low-pass systems and a negative value of ξ characterises high-pass systems.

We now look at the Laguerre filter used in conjunction with the LMS and RLS algorithms.



(a) (b)

Figure 8.3: Mean-squared error versus ξ for HP system. (a) Orders 0 to 8. (b) Orders 9 to 17.



(a) (b)

Figure 8.4: (a) Duct impulse response, length 256, (b) Mean-squared error versus ξ for filter lengths 16, 32, 64, 128.

Chapter 9

Adaptive Laguerre filtering

Several authors have published results using adaptive Laguerre filters. We first review the stochastic gradient algorithm (LMS) and the recursive least-squares (RLS) when the input samples $u[n]$ are replaced by the Laguerre filter outputs. Since the Laguerre transform is almost unitary, we use the algorithms unchanged except for using the Laguerre output instead of TDL outputs.

We review the main conclusions from previously published results. We then present an example using the Laguerre LMS and RLS in a system identification setting.

9.1 Adaptive algorithms

9.1.1 LMS

Also called the stochastic gradient and steepest descent algorithm, the LMS algorithm minimises the mean-squared error by finding a sequence of weights which converge to the Wiener solution $\boldsymbol{w}_0$.

In practice the weights are updated based on the error at time n, $e[n] = d[n] - \hat{d}[n]$ and convergence is measured by averaging the scalars $e[n]$ with an exponential window.

The estimate of the desired response is modeled by a length FIR filter with K taps $\{w_k\}$

$$\hat{d}[n] = \sum_{k=0}^{K-1} w_k[n]^* \tilde{u}[n-k] = \boldsymbol{w}^{*T}[n]\tilde{\boldsymbol{u}}[n] \tag{9.1}$$

with $\boldsymbol{w}[n] = [w_0[n]w_1[n]\dots w_k[n]]^T$ the weight column vector at time n and $\tilde{\boldsymbol{u}}[n] = [\tilde{u}[n]\tilde{u}[n-1]\dots\tilde{u}[n-K+1]]^T$ the K outputs of a Laguerre delay line at time n, also a column vector. To emphasise the orthogonal transform, we use the notation $\tilde{\boldsymbol{u}}[n] = \boldsymbol{\Psi}\boldsymbol{u}[n]$.

From here onwards, the development of the LMS-Laguerre follows the same development as the LMS with no new simplifications. We thus have

$$e[n] = d[n] - \boldsymbol{w}^{*T}\boldsymbol{\Psi}\boldsymbol{u}[n] \tag{9.2}$$

$$|e[n]|^2 = (d[n] - \boldsymbol{w}^{*T}\boldsymbol{\Psi}\boldsymbol{u}[n])(d^*[n] - \boldsymbol{u}^{*T}[n]\boldsymbol{\Psi}^{*T}\boldsymbol{w}[n]) \tag{9.3}$$

After multiplying out, we find the same form for the gradient of the squared error with respect to the weight vector

$$\nabla_{\boldsymbol{w}}|e[n]|^2 = -2\boldsymbol{\Psi}\boldsymbol{u}[n]d^*[n] + 2\boldsymbol{\Psi}\boldsymbol{u}[n]\boldsymbol{u}^{*T}[n]\boldsymbol{\Psi}^{*T}\boldsymbol{w}[n] \tag{9.4}$$

The weight update is then written as

$$\boldsymbol{w}[n+1] = \boldsymbol{w}[n] + \mu(\boldsymbol{\Psi}\boldsymbol{u}[n]d^*[n] - \boldsymbol{\Psi}\boldsymbol{u}[n]\boldsymbol{u}^{*T}[n]\boldsymbol{\Psi}^{*T}\boldsymbol{w}[n]) \tag{9.5}$$

$$\boldsymbol{w}[n+1] = \boldsymbol{w}[n] + \mu\boldsymbol{\Psi}\boldsymbol{u}[n](d^*[n] - \underbrace{\boldsymbol{u}^{*T}[n]\boldsymbol{\Psi}^{*T}\boldsymbol{w}[n]}_{\tilde{d}^*[n]}) \tag{9.6}$$

$$\boldsymbol{w}[n+1] = \boldsymbol{w}[n] + \mu\boldsymbol{\Psi}\boldsymbol{u}[n]e^*[n] \tag{9.7}$$

Since the Laguerre transform (and not the matrix) is unitary and thus does not change the average power $R_{uu}(0)$, the step size μ was set just like in the classical LMS to 1/4 of its maximum value [36]

$$\mu = 1/4\frac{2}{MR_{uu}(0)} \tag{9.8}$$

Recall the weight error of a gradient-driven adaptive filter has two parts: the weight error due to the proximity to the optimum Wiener solution and the error due to the fluctuations caused by the choice of the step size. The step-size is thus chosen as a compromise between these two conflicting requirements.

9.1.2 RLS

As we have seen for the LMS case by emphasising the Laguerre transform of input vector $\boldsymbol{u}[n]$, the algorithm remains unchanged.

By performing the same emphasis of the transform of $\boldsymbol{u}$, the RLS algorithm also remains unchanged. However we may ask about the effect of using a transform on the condition number of the transformed deterministic autocorrelation matrix. Ill-conditioning is related to the amount of pivoting performing during Gauss-Jordan elimination to compute the inverse of a matrix. Ill-conditioning is thus related to the size of the matrix. If we can find a reduced order system representation, which is determined by the length of the weight vector $\boldsymbol{w}$, the deterministic autocorrelation matrix should be less ill-conditioned than its non-transformed version.

For the forgoing examples, the initial gain of the initial deterministic correlation matrix was 1e-6 and the exponential weighting factor [36] was 0.999 since the process was stationary.

Remark

Note that the RLS algorithm has complexity proportional to K^2. By using transversal filters, the Fast-RLS computes the least-squares algorithm with a complexity proportional to K. It is properly initialised by forming the forward and backward prediction filters one sample at a time, such that after the first K iterations both filters are full for the first time. Such a reinitialisation should also be done when the rescue variable becomes out of bounds after a delay of $2K$. The Fast-RLS is derived using recursive time and order updates [36]. This requires that, at each iteration, the input vectors $\boldsymbol{u}[n]$ be related by a time shift of 1 sample. As the outputs are now taken from the Laguerre filter taps, $\tilde{\boldsymbol{u}}$, this relation is only true for $\xi \rightarrow 0$. Thus the Fast-RLS is not applicable to a Laguerre filter.

9.2 Previous results

Results have been published using the Laguerre filter in adaptive echo cancellation and system identification. We mention some important results before proceeding with an example.

9.2.1 Transversal filters

Falconer and Davidson [37] investigated the use of a Laguerre filter appended to a classical FIR filter to perform echo cancellation. They decided on this two-part approach based on the echo impulse response. This impulse response had rapid time variations initially but the tail smoothly decayed towards zero. In order to approximate the tail, a long FIR filter would have been required whereas a few IIR sections would have sufficed. Their main results were as follows:

1. Using a recursive definition of the Laguerre functions was numerically more stable than the functions obtained directly form Gram-Schmidt orthogonalisation.

2. The optimal value of ξ seemed to scale linearly with symbol rate.

3. The optimal value of ξ prevailed within a certain range (about $\pm 10\%$) and thus ξ appears robust once nearly optimal.

4. Inaccuracies in the gain of the first section, $\sqrt{1-\xi^2}$, which involves a square, a subtraction and a square-root operation, will cause the norms to differ from unity and will require compensation in the step-size. The rate of convergence will thus be affected.

9.2.2 Laguerre Gradient Lattice

It should be noted that Fejzo and Lev-Ari [1] have described a Gradient Adaptive Laguerre Lattice (GALL) algorithm. Compared to the normal GAL algorithm, the GALL showed good convergence using only one-tenth of the normal lattice order for a system identification problem.

9.3 System identification results

9.3.1 LMS and RLS

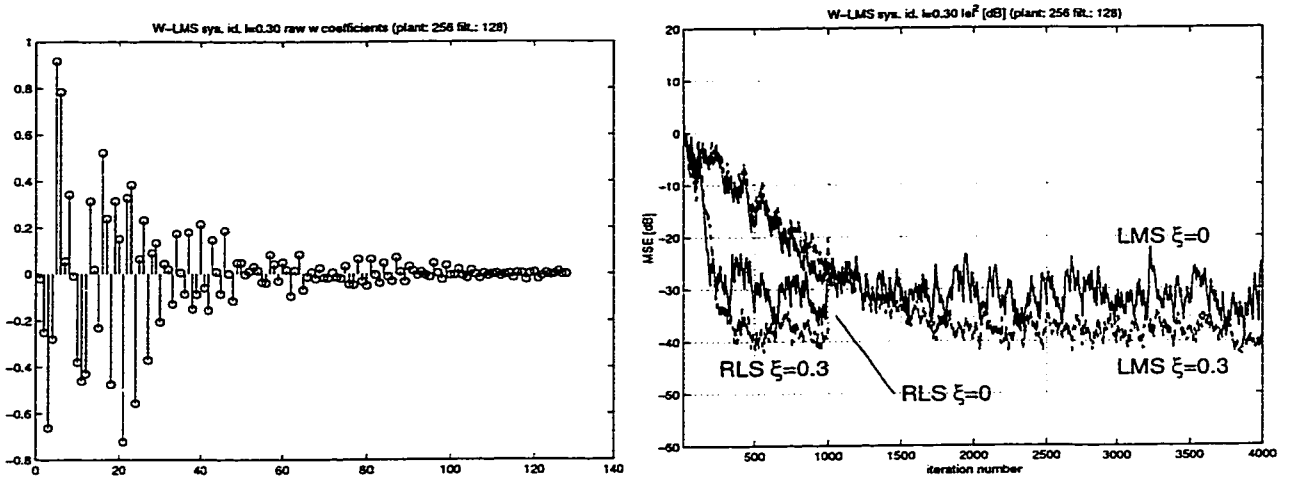
The duct impulse response shown in Fig. 8.4(a) was used as the plant in the adaptive system identification simulation as shown in Fig. D.1. White Gaussian noise of mean 0 and variance 1 was filtered by the LP filter $X(z)$ shown in Table 8.1 and its output denoted as x. This filter was used to shape the white Gaussian noise to keep the same system as in the least-squares solution. The choice of ξ was decided *a priori* by the minimum MSE results in Fig. 8.4(b) and thus set to $\xi = 0.3$. In this experiment, we compared the MSE results for $\xi = 0$ and $\xi = 0.3$ for both the RLS and LMS algorithms with various filter lengths K. The number of iterations were 4000 and 1000, for the LMS and RLS algorithms, respectively.

The plots shown in Figs. 9.1 and 9.2. refer only to the results obtained with $K = 128$. The $K = 128$ Laguerre filter coefficients are shown in Fig. 9.1. We have not shown the adaptive weights for $\xi = 0$ as these would be visually identical to the first 128 coefficients of the plant shown in Fig. 8.4 (a).

The estimated MSE, averaged using an exponential window with coefficient 0.9, for the $K = 128$ LMS and RLS algorithms for both $\xi = 0.3$ and $\xi = 0$ are shown in Fig. 9.1 (b). The lower MSE curves correspond to $\xi = 0.3$. We note the step-size dependent convergence of the LMS and the rapid convergence of the RLS. The MSE with $\xi = 0.3$, -40 dB, matches the result predicted using the least-squares method.

To illustrate the adaptive weights designed with $\xi = 0.3$ and $K = 128$, Fig. 9.2 (a) shows the frequency response of the plant impulse response, and the adaptive weights in both the Laguerre and non-Laguerre domain found by the RLS algorithm with $\xi = 0.3$. The adaptive weights for $\xi = 0.3$ are seen to have been designed on a frequency axis skewed towards π. In order to obtain the Laguerre coefficients in the non-Laguerre domain, we premultiply them by the 256×128 synthesis matrix and this performs the (lossy) inverse transformation. The resulting coefficients are then seen to match the plant frequency response except at high frequencies where less coefficients are available due to warping.

Fig. 9.2 (b) depicts the frequency response of the difference of the plant and the adaptive weights for $\xi = 0$ (solid line) and $\xi = 0.3$ (dashed line) again with $K = 128$. At frequencies below 0.2 f_s there is greater attenuation with the warped filter. Inversely, above 0.3 f_s it



(a) (b)

Figure 9.1: For $K = 128$, (a) Laguerre coefficients, $\xi = 0.3$. (b) MSE vs. iteration number: LMS, RLS.

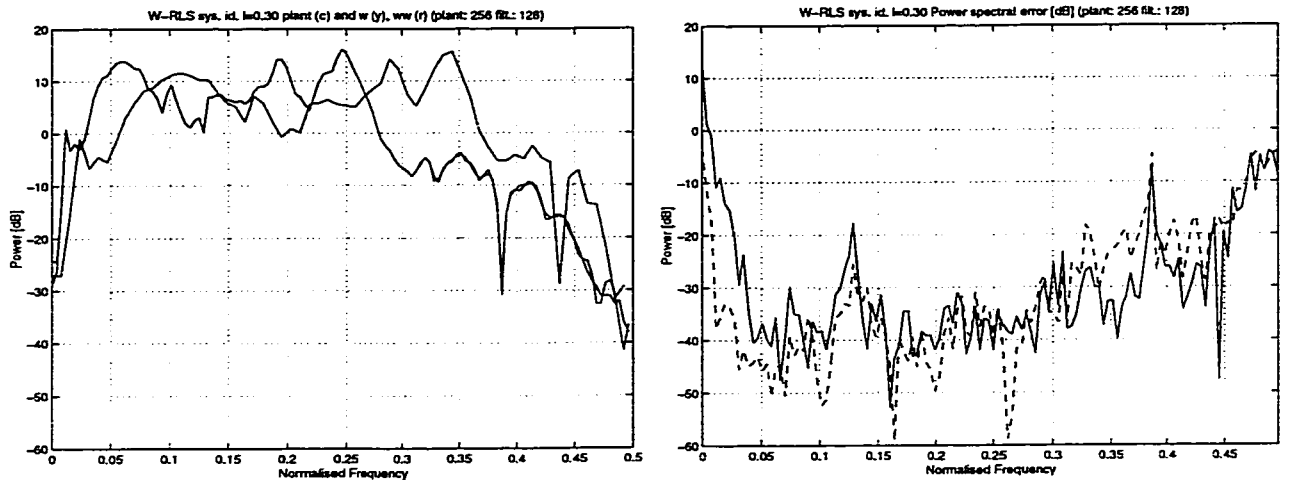
is the non-warped or $\xi = 0$ filter which achieves less error. Residual noise-shaping results have been published using a noise-shaping filter in an FX-LMS algorithm for system inverse problems [38]. However, the effect of the filter actually deteriorates convergence in spectra with low amplitudes. This is due to the fact that the LMS is a gradient-based algorithm and the gradient depends on the most energetic components in the reference signal. The Laguerre filter, on the other hand, forces convergence using less coefficients for high frequency signals (for $\xi > 0$).

The MSE averaged over the last 500 estimates and normalised by the energy of the desired signal $d[n]$ for $\xi = 0$ and $\xi = 0.3$ are shown in Table 9.1

9.3.2 RLS with pole tracking

Belt and den Brinker [39] show the results of an RLS adaptive Laguerre filter with simultaneous adaptation of the Laguerre parameter. In their results, they varied a plant pole once the optimal weights for an initial solution ξ_0 were found.

In our case we would like the algorithm to find the optimal pole once started from some initial pole value and its associated optimal weight solution.



(a) (b)

Figure 9.2: For $K = 128$, (a) Fourier transform of plant impulse response, Laguerre and transformed Laguerre coefficients. (b) Fourier transform of RLS approximation error: $\xi = 0$ (solid line), 0.3 (dashed line).

Pole tracking algorithm

Assuming the weights are optimal enough and almost satisfy the normal equations given in (5.27), the Laguerre parameter ξ is updated as follows. Lengthen the length K Laguerre filter by one section and denote its output by $\tilde{u}_{K+1}[n]$. Let $V[n]$ be an estimate of the $K+1$th gradient, $E\{e[n]\tilde{u}_{K+1}[n]\}$. Then update $\xi[n]$ as

Table 9.1: Mean-squared error [dB] for system identification.

K	LMS $\xi = 0$	LMS $\xi = 0.3$	RLS $\xi = 0$	RLS $\xi = 0.3$
32	-14.4	-13.7	-13.9	-14.1
64	-20.8	-22.0	-20.6	-22.6
128	-30.7	-38.3	-29.0	-40.0
256	-45.1	-40.2	-143.4	-57.9

$$V[n+1] = \lambda V[n] + (1-\lambda)e[n]\tilde{u}_{K+1}[n] \tag{9.9}$$

$$\xi[n+1] = \xi[n] - \Delta\xi[n]sign(-V[n]w_K[n]) \tag{9.10}$$

where

$$\Delta\xi[n] = \mu_\xi(1-\xi^2[n])^{3/2} \tag{9.11}$$

A few words about the derivation. A change in the value of $\xi[n]$ will generate a transient at all outputs of the Laguerre filter. By assuming the value of $\xi[n]$ is close to its optimal value and fluctuates whitely with $E\{\xi[n]\} = \xi$, the idea is to make the power of the output at the last filter tap smaller than the variance of $\xi[n]$. A *sign* algorithm is used to have constant step-size before multiplication by $\Delta\xi[n]$. It is also mentioned that the assumption that $\xi[n]$ fluctuates whitely is more true for low values of λ. Thus, and just like in all gradient algorithms, a larger value of λ will give a better estimate of $E\{e[n]\tilde{u}_{K+1}[n]\}$ and thus allow faster convergence towards the optimum ξ with more variations about the optimum. Conversely, a smaller value of λ will converge more slowly but with less fluctuations about the optimum. Also, by the assumption that the weights almost satisfy the normal equations, the RLS adaptive filter and the pole tracking LMS must operate on different time-scales [39]. Thus the parameter μ_ξ must be chosen sufficiently small to ensure that the RLS can find the optimal weights $\boldsymbol{w}$ before the pole is permitted to move.

Results

The following simulation results were done with an adaptive filter length $K = 128$. First, $\xi = 0.4$ was chosen and the RLS was used to find the optimal filter in 500 iterations. Without restarting, the pole tracking algorithm was then enabled.

The adaptation parameters for the pole tracking algorithm were set to : $\lambda = 0.9, \mu = 1 \times 10^{-4}, V[0] = 0.1$. For the RLS algorithm, the initial gain of the initial deterministic correlation matrix was unchanged at 1e-6 and the exponential weighting factor reduced to 0.99.

The results for the $K = 128$ filter and 500+3500 iterations are shown in Figs. 9.3 (a) and (b). Fig. 9.3 (a) depicts the pole migration versus iteration number. We see that the pole approaches the optimum value but over a few thousand iterations. This result is caused by the slow pole migration required to keep the RLS solution near optimal. Fig. 9.3 (b) shows the MSE curves versus iteration number for initial parameter values $\xi[0] = 0.4$. The MSE averaged over the last 500 iterations was -23.5 dB.

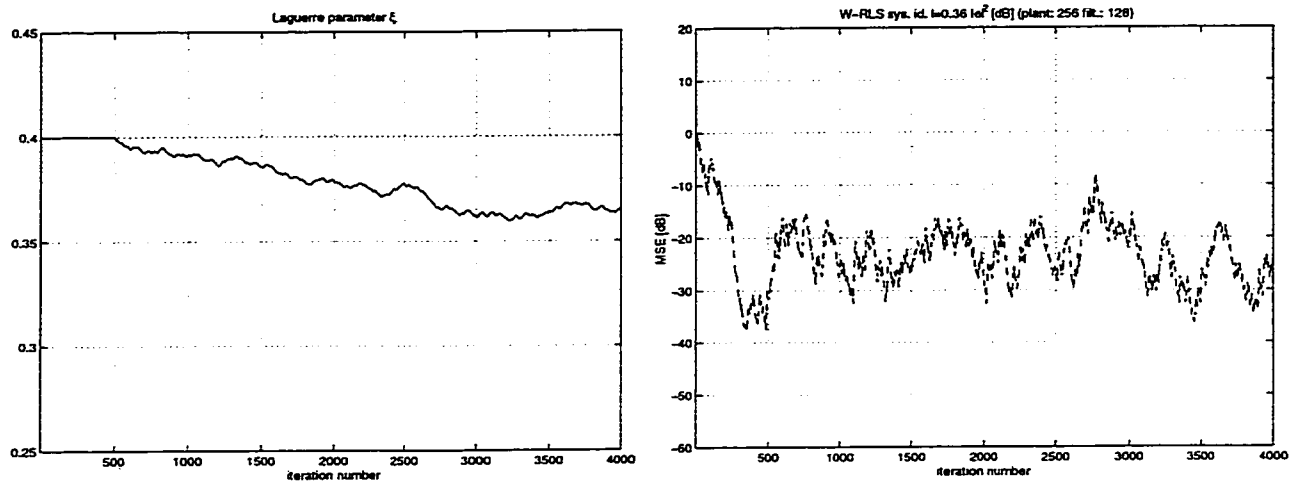
In order to get an idea of the adaptation noise produced by the pole tracking algorithm, the same simulation was repeated with the pole at the optimum value of $\xi = 0.3$ with 500+1500 iterations. The pole migration is shown in Fig. 9.4 (a) and the MSE versus iteration number in (b). The averaged MSE in this case was -23.9 dB. By comparing the MSE figures in Fig. 9.3 (b) and Fig. 9.4 (b), and the similar MSE values, we conclude that, in both cases, the adaptation noise hides the optimum MSE. By virtue of the LMS algorithm used, decreasing μ_ξ and increasing λ would result in less adaptation noise but even longer convergence time.

In conclusion, the pole tracking algorithm is not satisfactory as it was derived based on the assumption that the normal equations be nearly satisfied. As the pole movement hinders reaching the optimal solution, the resulting adaptation noise is high. A better algorithm should be investigated.

9.4 Chapter summary

The Laguerre filter can be used in adaptive filtering using the LMS and RLS algorithms. Choosing the Laguerre parameter ξ *a priori*, we have shown that the MSE in a system identification problem can be reduced by 10 dB for a filter length 128 where the plant impulse response had length 256. The number of iterations required to reach convergence was about the same as for a classical adaptive FIR filter. Furthermore the error spectrum was shaped depending on the Laguerre parameter. This may be desirable in acoustics where the ear is the final receptor as we can shape the noise to match the ear's non-uniform sensitivity to loudness.

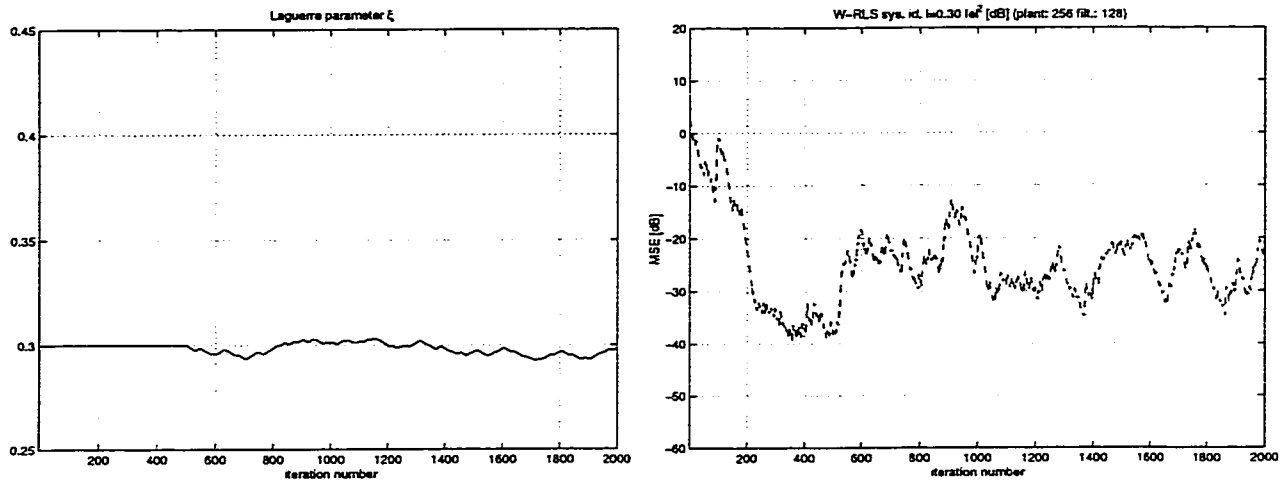
When no estimate of the Laguerre parameter is available, an LMS pole-tracking algorithm, operating simultaneously with an RLS adaptive filter, was shown to converge to the



(a) (b)

Figure 9.3: RLS $K = 128$, $\xi[0] = 0.4$, (a) Laguerre coefficient $\xi[n]$ vs. iteration number n. (b) MSE vs. iteration number.

optimum albeit plagued with slow LMS convergence.



(a) (b)

Figure 9.4: RLS $K = 128$, $\xi[0] = 0.3$, (a) Laguerre coefficient $\xi[n]$ vs. iteration number n. (b) MSE vs. iteration number.

Chapter 10

Conclusion

10.1 Main results

We have presented the Laguerre transform and shown its application in signal compression. The Laguerre transform is controlled by the Laguerre parameter ξ. For $\xi \to 0$, the Laguerre transform approaches the identity transform or matrix. For $|\xi| > 0$, this matrix is unitary only if it is infinite in size. When implemented as a finite size matrix, the transform becomes less and less unitary and the inverse transform is not complete. However, by choosing the size of the matrix such that orthogonality is preserved, an orthogonal series expansion is obtained. Using an acoustic duct impulse response, the Laguerre transform was shown to provide a more compact representation with less mean-squared error than the truncated DCT for high compression ratios. A particularity of a signal having been compressed and subsequently decompressed using the Laguerre transform is that it has non-uniform resolution, with resolution becoming coarser with the length of the signal. This effect was shown to be the main cause for outperforming the DCT at high compression ratios.

10.2 Secondary results

Laguerre functions provide an orthonormal series expansion on the semi-infinite Hilbert space of square-integrable functions, $L_2(\mathbb{R}_+)$, and are orthogonal under an exponential window function on this interval. They thus inherit the property of successive approximation of

orthonormal series expansions.

The Laguerre generating polynomials belong to the class of classic orthogonal polynomials which include ultraspherical polynomials such as the Chebychev and Legendre orthogonal polynomials.

The Laguerre polynomials exist in different forms depending on the number of free parameters and whether they are written in continuous-time (CT) or discrete-time (DT).

In CT, the generalised Laguerre functions have two free parameters: The Laguerre parameter σ and the order of generalisation α. They have a rational Laplace transform only for even values of generalisation order. For $\sigma = 0$, we obtain the Laguerre functions.

In DT, the generalised functions are called the Meixner functions with Laguerre parameter ξ and generalisation order α. These do not have a rational z-transform. However there exists Meixner-like functions which have a rational z-transform. Their degeneralised form are called the discrete Laguerre function or simply the Laguerre sequences.

The Laguerre filter is obtained by the z-transform of the DT Laguerre sequences and is realised by substituting the unit-delay element in a tapped delay line by a first-order allpass filter with parameter $|\xi| < 1$ and by prepending the structure by a normalising single-pole IIR filter. Structures and c code were given for the realisation of the Laguerre filter. The impulse response of each tap k is the kth Laguerre orthonormal sequence $\psi_k[n]$. Note that if the number of taps K is increased to infinity, the resulting Laguerre sequences form a basis for $l_2(\mathbb{N})$. Also note that as $\xi \to 0$, for any finite number of taps K, the Laguerre orthonormal sequences become the canonical basis $\{\delta[n-k]\}$.

In the frequency domain, the Laguerre filter realises a non-uniform frequency representation called frequency warping. The amount of warping is controlled by the Laguerre parameter ξ. This effect skews the frequency response towards π rads for $\xi > 0$ or towards 0 rads for $\xi < 0$. Such an effect is useful in designing filters with variable precision in the frequency response.

Using the least-squares algorithm, the minimum MSE surface of a Laguerre filter was shown to have local minima which vary as a function of ξ. For one duct impulse response of length 256, choosing the optimal value of ξ improved the minimum mean-squared error by 10 dB of a classical FIR adaptive filter of length 128. This result was shown in Fig 8.4(b).

These improved performance results were reproduced with a Laguerre adaptive filter

using the LMS and RLS algorithms. The Laguerre filter required about the same number of iterations to reach convergence as a classical FIR filter. In addition, the error spectrum was shown to be shaped depending on ξ and this might be an attractive feature for work in acoustics and active noise control. A good choice for the Laguerre parameter ξ was found vital to reduce the mean-squared error below that of a classical tapped delay line.

Depending on the frequency response of the plant modeled, the sign of the Laguerre parameter ξ can be infered. Given the plant impulse response, an estimate of ξ can be obtained. Furthermore, in adaptive filtering using RLS, an LMS Laguerre pole tracking algorithm was shown to converge to the optimum. However, since it is a gradient-based algorithm, the time to convergence and convergence rate are inversely affected by the LMS step-size.

10.3 Future work

The following topics expand on the present work.

1. The single Laguerre pole may not be sufficient to characterise a given system. If we allow a distinct (possibly-complex) pole in each filter section we have a Kautz-1 filter. The use of this filter in system modeling has been investigated by [40]. The selection of poles is done using an energy criterion and these poles have to be ordered in order to pack the most energy modeling poles into the first coefficients. Results and adaptive algorithms for Kautz-1 filters in adaptive filtering would need to be investigated. If N poles per section are allowed then we have a Kautz-N filter [41] and results for adaptive filtering (say $N = 2$) have yet to be published [27].

2. The Laguerre filter is a degeneralised version of the more general Laguerre functions. The filter is then characterised by the Laguerre pole ξ and by the order of generalisation α. In discrete-time, these functions, called the Meixner functions, do not have a rational z-transform. However Den Brinker et al. [11] [17] [16] have defined Meixner-like functions which have a rational z-transform. Good initial values for ξ and α, given the impulse response, can be calculated and adaptive filtering results have been published. It remains to be seen how these algorithms behave with real acoustic impulse responses

which require complex models. A pole tracking algorithm could be presented. The implementation complexity of such a filter also needs to be determined.

Appendix A

Allpass chain realisation

A.1 Structures

Härmä [6] shows 4 AP-chain implementations. The Direct forms I and II are classical IIR-structures and are shown in Figs. A.1 and A.2. The A- and B- structures, (Figs. A.3 and A.4), were the ones actually used in the simulations as Direct form realisations are strongly affected by coefficient quantisation [8].

The form II and the B-structure have only one memory element per AP filter; form I and the A-structure have two. Implementing an AP filter however always requires 2 memory elements. For example, in the form II, the output of the second sum has to be stored temporarily, as it will be required for the computation of the following section.

A.2 Algorithm

The output of each AP section follows from the z-transform of the AP filter $A(z)$ (7.4) [1]:

$$Y(z) = z^{-1}X(z) + \xi(z^{-1}Y(z) - X(z)) \tag{A.1}$$

The A-structure was used for the functions longchain.c and laguerre.c (cf. Appendix B). Each output was calculated by propagating each input sample through all $K-1$ sections. Each section was traversed using the following exchange algorithm [6]:

```
tmp = p_out[i] + xi*p_out[i+1] - xi*x;
p_out[i] = x;
x = tmp;
```

Note that this algorithm reverses the order of the samples. This can be seen by setting $\xi = 0$. In other words, the vector output of the algorithm has the samples ordered as a physical right-sided delay line, with the first sample propagating to the far right and the last sample stationed at the far left.

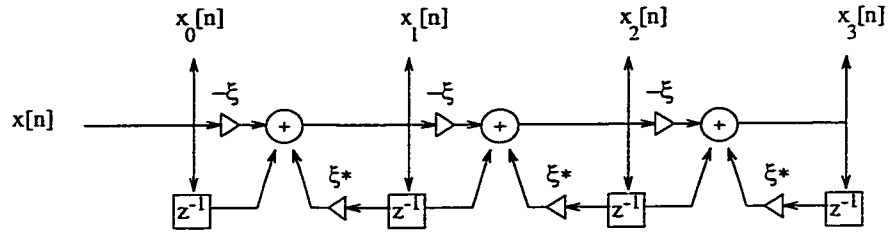


Figure A.1: Allpass chain: Direct form I.

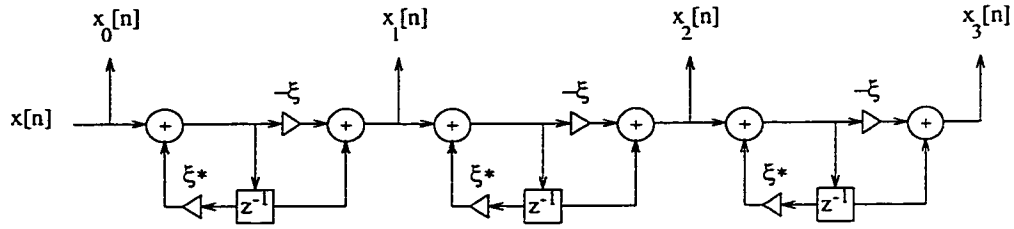


Figure A.2: Allpass chain: Direct form II.

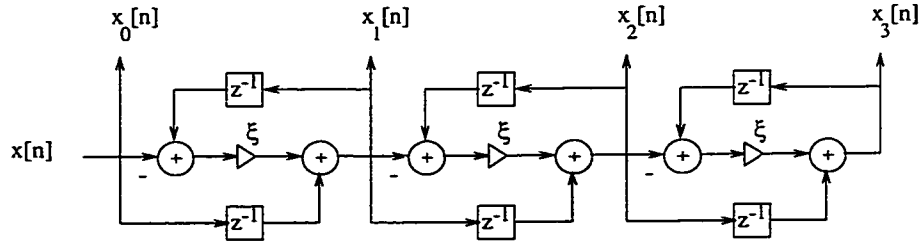


Figure A.3: Allpass chain: A-structure (longchain.c, laguerre.c).

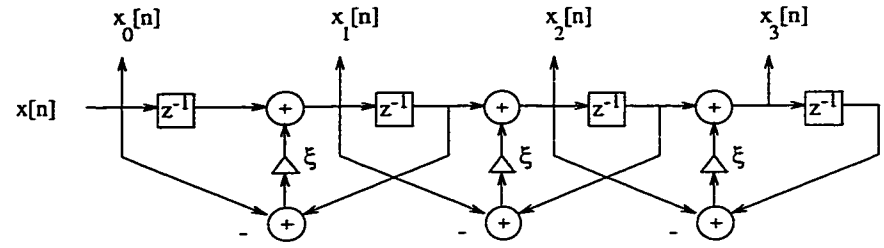


Figure A.4: Allpass chain: B-structure.

A.3 Implementation complexity

The complexity of implementing a Laguerre filter depends on whether we wish to also perform an inner product with a vector of weights or not. The inner product with a weight vector is required for filtering but not for a Laguerre series decomposition. We assume that all samples are non-complex and ignore machine cycles required for function overhead.

Consider the case without the inner product. We refer to the 3 lines of c code given above. For the first line of code we require one multiplication, 1 negation and 2 accumulates. A DSP like the AD2181 can perform 2 fetches and a multiply-accumulate per machine cycle. Since we need 2 accumulates, the first line of code requires 2 cycles. Lines 2 and 3 are a simple exchange and can be performed in a single cycle. Thus for a single input and K output taps we require $3K$ cycles. Compare this to a Tapped Delay Line (TDL) which represents the samples in vector form and are thus immediately available and require no machine cycles to compute.

Now consider the additional step of performing an inner product of the tap outputs with a weight vector. Note that the ith output stored in x is available at the start of the loop. Since the exchange portion (lines 2 and 3) has a free multiply/accumulate operation, we can embed the inner product here. Thus still $3K$ cycles are required. For a TDL we require K cycles for a length K FIR filter.

A.4 Signal-to-noise ratio analysis

Braccini and Oppenheim analysed round-off noise for a Laguerre filter implemented in fixed point arithmetic [25]. When the input was scaled at each stage to avoid clipping the signal-

to-noise ratio was relatively insensitive to the number of sections K. When clipping was allowed by modeling the input by white Gaussian noise, the results were still very good (-70 dB for 16 bit arithmetic and 6×10^{-5} saturation probability).

Appendix B

Source code

B.1 longchain.c

```
#include <stdio.h>
#include <math.h>

/*********************************************************************/
/* LONGCHAIN.C                                                       */
/* This function uses a long chain of first order                    */
/* allpass elements to warp a signal                                 */
/* USAGE: y = longchain(x,n,lam)                                     */
/* where y : output of the filter                                    */
/*       x : an input signal                                         */
/*       n : # of all-pass sections,                                 */
/*          (n+1) <= length(x) is length of produced warped signal   */
/*       lam : the warping parameter                                 */
/*       mem : memory, same len as output                            */
/*                                                                   */
/*                                                                   */
/* in:                        +-------+    +-------+     +-------+   */
/* [x(N), x(N-1),.. x(0)] -+->| A(z) |-+->| A(z) |-..- | A(z) |-+-  */
/*                         |  +-------+ |  +-------+     +-------+ |  */
```

```
/*                          |          |                          | */
/* out:                   y(0)       y(1)                      y(N) */
/*                                                                  */
/*                                                                  */
/* Aki Hrm and Matti Karjalainen, Helsinki U. of Technology,      */
/* Laboratory of Acoustics and Audio Signal Processing             */
/*                                                                  */
/* Stephan Quednau                                                  */
/* S.I.T.E, University of Ottawa, Canada                            */
/*                                                                  */
/* This file is a part of the Warping toolbox available at          */
/* http://www.acoustics.hut.fi/software/warp/                       */
/*******************************************************************/

void trans(double *p_in, unsigned int N, double lam,
      unsigned int Nmax_in, double *p_out)
{
   double x, tmp;
   unsigned int w,e;

   for(w=0; w<Nmax_in; w++)
   {
      x = *p_in++;

      for(e=0; e<N; e++)
      {
         /* The difference equation */
         /* Each input sample propagates through */
         /* all the all-pass sections */

         /* Note: with lam = 0, this loop time reverses x */
         /* example, with x = [1 2 3 4], N = 3 */
         /* w = 0: p_out = [1 0 0 0]           */
```

```
            /* w = 1: p_out = [2 1 0 0]          */
            /* w = 2: p_out = [3 2 1 0]          */
            /* w = 3: p_out = [4 3 2 1]          */

            /* A-structure */
            tmp = p_out[e] + lam*p_out[e+1] - lam*x;
            p_out[e] = x;
            x = tmp;
        }
        /* e = N, the last (N+1)th output */
        p_out[N] = x;
    }
}
```

B.2 laguerre.c

```
#include <stdio.h>
#include <math.h>

/*****************************************************************/
/* LAGUERRE.C                                                    */
/* This function implements the Laguerre filter                  */
/* as shown by King and Paraskevopoulos                          */
/* USAGE: y = laguerre(x,n,lam)                                  */
/* where y : output of the filter                                */
/*       x : an input signal                                     */
/*       n : # of all-pass sections,                             */
/*          (n+1) <= length(x) is length of produced warped signal */
/*       lam : the warping parameter                             */
/*       mem : memory, same len as output                        */
/*                                                               */
/*                                                               */
/*   +------------+    +---------------------+    +------+      */
```

```
/* ->| 1/(1-az-1) |-+->| (1-a^2)z-1/(1-az-1) |-+->| A(z) |-+        */
/*   +------------+ |  +----------------------+ |  +------+ |        */
/*                  |                           |    ...    |        */
/* out:            y(0)                        y(1)        y(N)      */
/*                                                                    */
/*                                                                    */
/* Aki Hrm and Matti Karjalainen, Helsinki U. of Technology,      */
/* Laboratory of Acoustics and Audio Signal Processing               */
/*                                                                    */
/* Stephan Quednau                                                    */
/* S.I.T.E, University of Ottawa, Canada                              */
/*                                                                    */
/* This file is a part of the Warping toolbox available at           */
/* http://www.acoustics.hut.fi/software/warp/                         */
/*********************************************************************/
void trans(double *p_in, unsigned int N, double lam,
      unsigned int Nmax_in, double *p_out)
{
   double x, tmp;
   unsigned int w,e;
   double b;

   b = sqrt(1-lam*lam);

   for(w=0; w<Nmax_in; w++)
   {
      /* e=0, the LP single pole (IIR) section */
      /* with gain b at the input */
      x = lam*p_out[0] + b*(*p_in++);

      for(e=0; e<N; e++)
      {
         /* The difference equation */
```

```
            /* Each input sample propagates through all */
            /* the all-pass sections */

            /* Note: with lam = 0, this loop time reverses x */
            /* example, with x = [1 2 3 4], N = 3 */
            /* w = 0: p_out = [1 0 0 0]          */
            /* w = 1: p_out = [2 1 0 0]          */
            /* w = 2: p_out = [3 2 1 0]          */
            /* w = 3: p_out = [4 3 2 1]          */

            /* A-structure */
            tmp = p_out[e] + lam*p_out[e+1] - lam*x;
            p_out[e] = x;
            x = tmp;
        }
        /* e = N, the last (N+1)th output */
        p_out[N] = x;
    }
}
```

Appendix C

Other functions and properties

C.1 Gamma function

The Euler integral of the second kind [10] is also called the Gamma function

$$\Gamma(z) = \int_0^\infty e^{-t} t^{z-1}\, dt \tag{C.1}$$

with $Re(z) > 0$. The following properties hold:

$$\Gamma(z+1) = z\Gamma(z) \tag{C.2}$$

$$\Gamma(z)\Gamma(1-z) = \pi / sin(\pi z) \tag{C.3}$$

$$\prod_{i=0}^{n-1} \Gamma(z + \frac{i}{n}) = n^{1/2-nz}(2\pi)^{(n-1)/2}\Gamma(nz) \qquad n \in \mathbb{N} \tag{C.4}$$

The following holds as well

$$\frac{1}{\Gamma(z)} = \frac{1}{2\pi j} \int_{-\infty}^{0} e^{t} t^{-z}\, dt \tag{C.5}$$

C.2 Binomial coefficient

The binomial coefficient is given as

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}. \tag{C.6}$$

with $0! = 1$.

The following identity holds

$$\binom{n-1}{k-1} + \binom{n-1}{k} = \binom{n}{k} \tag{C.7}$$

The binomial theorem states that

$$(x+y)^n = \sum_{i=0}^{n} \binom{n}{i} x^i y^{n-i} \tag{C.8}$$

holds for x, y in any ring $\mathcal{R}$ for $n \geq 1$. The binomial coefficients $\binom{n}{k}$ can be obtained directly by reading the nth row of the Pascal triangle

$$\begin{array}{ccccccccccc}
 & & & & & 1 & & & & & \\
 & & & & 1 & & 1 & & & & \\
 & & & 1 & & 2 & & 1 & & & \\
 & & 1 & & 3 & & 3 & & 1 & & \\
 & 1 & & 4 & & 6 & & 4 & & 1 & \\
 & & & & & \vdots & & & & &
\end{array}$$

Appendix D

System structures

D.1 System identification

We define the following signals

1. $x[n]$, the input (reference) signal at time n
2. $y[n]$, the adaptive filter output (actuator) signal at time n
3. $d[n]$, the desired (primary (ANC) or target (TSR)) signal at time n
4. $\hat{d}[n]$, the estimate of desired signal at time n
5. $e[n]$, the error (sensor) signal at time n
6. $h[n]$, the plant impulse response
7. $w_k[n]$, the kth coefficient of the adaptive filter at time n

and the following transfer functions

1. $X(z)$, the reference filter transfer function
2. $W(z)$, the adaptive filter transfer function
3. $H(z)$, the plant transfer function

A system identification problem is shown in Fig. D.1. An adaptive filter is set into a parallel path to the plant and the error is used to steer the adaptive filter.

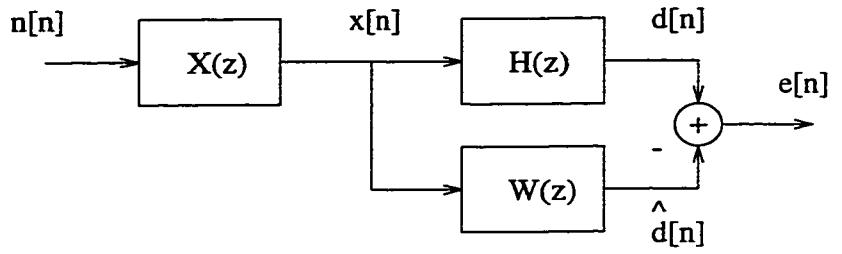


Figure D.1: System identification using adaptive filtering.

D.2 System inverse

If we let $y[n]$ be the adaptive filter output (actuator) signal at time n and D a delay, then the system inverse or deconvolution problem can be shown as in Fig. D.2. An adaptive filter appended to the plant and the plant output is compared to a delayed version of the reference signal. The delay models the combined plant and adaptive filter delays and makes the combined impulse response causal. Again the error is used to steer the adaptive filter.

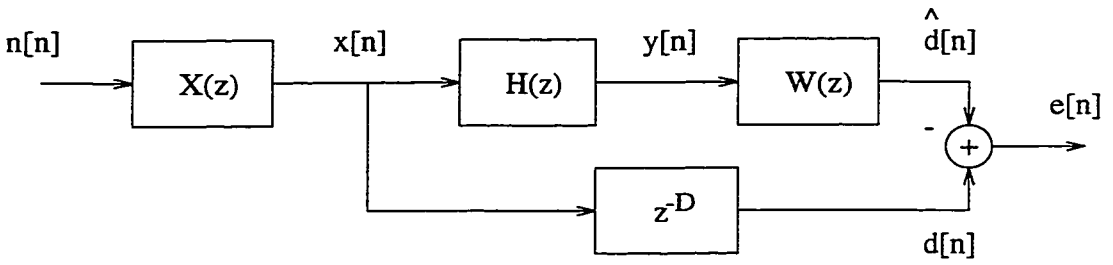


Figure D.2: Solving a system inverse problem using adaptive filtering.

Minimum phase systems

A stable system

$$H(z) = \frac{B(z)}{A(z)} \tag{D.1}$$

is said to be minimum phase if all the zeros of $H(z)$ lie inside the unit circle $|z| \leq 1$. The stability is guaranteed if all the poles of $H(z)$ are inside the unit circle. Note that the

minimum phase property of $H(z)$ implies a stable inverse $H(z)^{-1}$ and the stability property of $H(z)$ implies the minimum phase of $H(z)^{-1}$ [42].

Now, the invertibility of a linear time-invariant system is related to the phase of the system [42]. Indeed, we may have minimum, maximum and mixed phase realisations of transfer functions with identical amplitude responses.

If a stable inverse filter is non-causal, we can make it causal by prepending sufficient delay to the filter.

Bibliography

[1] Z. Fejzo and H. Lev-Ari, "Adaptive Laguerre-lattice filters," *IEEE Trans. Signal Process.*, vol. 45, pp. 3006–3016, Dec. 1997.

[2] L. Kinsler and A. R. Frey, *Fundamentals of Acoustics.* New York: John Wiley and Sons, second ed., 1962.

[3] R. McEachern, "Hearing it like it is: Audio signal processing the way the ear does it," *DSP Appl.*, vol. 3, pp. 35–47, Feb. 1994.

[4] R. McEachern, "Hearing it like it is: Sound de-modulation via parallel filter banks," *DSP Multimedia Tech.*, vol. 3, pp. 23–37, May 1994.

[5] R. E. King and P. N. Paraskevoloupos, "Digital Laguerre filters," *Circuit Theory Appl.*, vol. 5, pp. 81–91, 1977.

[6] A. Härmä, "Perceptual aspects and warped techniques in audio coding," Master's thesis, Helsinki university of technology, 1997.

[7] D. F. Mix, *Random signal processing.* New Jersey: Prentice-Hall, 1995.

[8] M. Vetterli and J. Kovačević, *Wavelets and subband coding.* New Jersey: Prentice-Hall, 1995.

[9] M. A. Snyder, *Chebyshev methods in numerical approximation.* New Jersey: Prentice-Hall, 1966.

[10] G. Szegő, *Orthogonal polynomials*, vol. 23. New York: American Mathematical Society, 1975.

[11] A. den Brinker, "Meixner-like functions having a rational z-transform," *Int. J. Circuit Theory Appl.*, vol. 23, pp. 237–246, 1995.

[12] D. Dufresne, "Laguerre series for asian and other options." Research Papers Series, Center for Actuarial Studies, 1999.

[13] J. J. O'Connor and E. F. Robertson. http://turnbull.dcs.st-and.ac.uk/history/Mathematicians/Laguerre.html.

[14] T. O. e Silva, "On the determination of the optimal pole position of Laguerre filters," *IEEE Trans. Signal Process.*, vol. 43, pp. 2079–2587, Sept. 1995.

[15] P. Broome, "Discrete orthonormal sequences," *J. ACM*, vol. 12, pp. 151–168, Apr. 1965.

[16] H. Belt and A. den Brinker, "Optimal parametrization of truncated generalised Laguerre series," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, (Munich, Germany), pp. 3805–3808, Apr. 1997.

[17] H. J. W. Belt and A. C. den Brinker, "Optimality condition for truncated generalized Laguerre networks," *Int. J. Circuit Theory Appl.*, vol. 23, pp. 227–235, 1995.

[18] Y. Lee, "Synthesis of electrical networks by means of the fourier transforms of Laguerre functions," *J. Math. Phys.*, vol. 11, pp. 83–113, 1932.

[19] T. Parks, "Choice of time scale in Laguerre approximations using signal measurements," *IEEE Trans. Autom. Contr.*, vol. 16, pp. 511–513, Oct. 1971.

[20] S. Winkler, "The approximation problem of network synthesis," *IRE Trans. Circuit Theory*, vol. CT-1, pp. 5–20, Sept. 1954.

[21] G. Clowes, "Choice of the time-scaling factor for linear system approximations using orthonormal Laguerre functions," *IEEE Trans. Autom. Contr.*, vol. 10, pp. 487–489, Oct. 1965.

[22] W. H. Kautz, "Transient synthesis in the time domain," *IRE Trans. Circuit Theory*, vol. CT-1, pp. 29–39, Sept. 1954.

[23] H. A. Nurges, "Laguerre models in problems of approximation and identification." *Autom. Rem. Contr.*, vol. 48, pp. 346–352, Mar. 1987.

[24] M. Golay, "The direct method of filter and delay line synthesis," *Proc. IRE*, vol. 42, pp. 585–588, Mar. 1954.

[25] C. Braccini and A. Oppenheim, "Unequal bandwidth spectral analysis using frequency warping," *IEEE Trans. Acoust. Speech Signal Process.*, vol. AASP-22, pp. 236–244, Aug. 1974.

[26] T. O. e Silva, "Optimality conditions for truncated Laguerre networks," *IEEE Trans. Signal Process.*, vol. 42, pp. 2528–2530, Sept. 1994.

[27] H. Belt, A. den Brinker, and T. O. e Silva, "Optimality conditions for truncated kautz series," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 43, pp. 117–122, Feb. 1996.

[28] A. K. Jain, ed., *Fundamentals of Digital Image Processing.* Prentice Hall, 1989.

[29] H. Strube, "Linear prediction on a warped frequency scale," *J. Acoust. Soc. Am.*, vol. 68, pp. 1071–1076, Oct. 1980.

[30] M. Karjalainen, E. Piirilä, A. Järvinen, and J. Huopaniemi, "Comparison of loudspeaker equalization methods based on DSP techniques," *J. Audio Eng. Soc.*, vol. 47, pp. 14–31, jan/feb 1999.

[31] A. Oppenheim, D. Johnson, and K. Steiglitz, "Computation of spectra with unequal resolution using the fft," *Proc. IEEE*, vol. 59, pp. 299–301, Feb. 1971.

[32] K. H. Grőchenig, "A discrete theory of irregular sampling," *Lin. Alg. and Appl.*, vol. 193, pp. 129–150, 1993.

[33] H. G. Feichtinger and K. H. Grőchenig, "Theory and practice of irregular sampling," in *Wavelets: mathematics and applications* (J. Benedetto and M. Frazier, eds.), pp. 305–363, CRC Press, 1993.

[34] J. W. Mark and T. D. Todd, "A nonuniform sampling approach to data compression," *IEEE Trans. Commun.*, vol. 29, pp. 24–32, Jan. 1981.

[35] M. S. Sabri and W. Steenart, "An approach to band-limited signal extrapolation: the extrapolation matrix," *IEEE Trans. Commun.*, pp. 74–78, Feb. 1987.

[36] S. Haykin, *Adaptive filter theory.* New Jersey: Prentice-Hall, 1986.

[37] G. Davidson and D. Falconer, "Reduced complexity echo cancellation using orthonormal functions," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 20–28, Jan. 1991.

[38] S. M. Kuo and J. Tsai, "Residual noise shaping technique for active noise control systems," *J. Acoust. Soc. Am.*, vol. 95, pp. 1665–1668, Mar. 1994.

[39] H. Belt and A. den Brinker, "Laguerre filters with adaptive pole optimization," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, (Atlanta, GA), pp. 37–40, May 1996.

[40] A. den Brinker and H. Belt, "Model reduction by orthogonalized exponentials," in *First Euro. Conf. Signal Anal. Pred.*, (Prague, Czech Republic), pp. 121–124, June 1997.

[41] H. Belt and H. Butterweck, "Cascaded all-pass sections for LMS adaptive filtering," in *VIII Euro. Signal Process. Conf., EUSIPCO*, vol. 2, (Trieste, Italy), pp. 1219–1222, Sept. 1996.

[42] J. G. Proakis and D. G. Manloakis, *Introduction to digital signal processing.* New York: Macmillan Publishing Company, 1988.