CHALMERS |  GÖTEBORGS UNIVERSITET

Digital Filter Design Using Semidefinite Programming

Examensarbete för kandidatexamen i matematik vid Göteborgs universitet
Kandidatarbete inom civilingenjörsutbildningen vid Chalmers

Jimmy Johansson
Fabian Samuelsson
Moa Samuelsson

Institutionen för matematiska vetenskaper
Chalmers tekniska högskola
Göteborgs universitet
Göteborg 2014

Digital Filter Design Using Semidefinite Programming

Examensarbete för kandidatexamen i matematik vid Göteborgs universitet
Fabian Samuelsson

Examensarbete för kandidatexamen i tillämpad matematik inom matematikprogrammet vid Göteborgs universitet
Jimmy Johansson

Kandidatarbete i matematik inom civilingenjörsprogrammet Teknisk matematik vid Chalmers
Moa Samuelsson

Handledare: Kin Cheong Sou
Examinator: Maria Roginskaya

Institutionen för matematiska vetenskaper
Chalmers tekniska högskola
Göteborgs universitet
Göteborg 2014

Abstract

This thesis explores an optimization based approach to the design problem of digital filters. We show how a digital filter in the form of a discrete linear time-invariant causal system can be characterized by a non-negative trigonometric polynomial, which in turn can be represented by a positive semidefinite matrix known as Gram matrix representation. This allows us to utilize the framework of linear conic optimization, especially semidefinite programming to obtain filters based on given specifications and optimal with respect to some property of the filter. The optimization is carried out with respect to minimizing the stopband energy as well as the passband ripple. We cover both FIR and IIR filters. The model is implemented in MATLAB using the modelling language CVX and solved using SeDuMi.

Sammanfattning

I den här rapporten presenteras en optimeringsbaserad metod för design av digitala filter. Vi visar hur filter som beskrivs av diskreta linjära tidsinvarianta kausala system kan representeras som icke-negativa trigonometriska polynom, vilka i sin tur kan representeras av positiva semidefinita matriser, så kallade Grammatriser. Det här möjliggör användadet av linjär konoptimering, speciellt semidefinit programmering för att ta fram filter baserade på givna specifikationer som är optimala med avseende på någon egenskap hos filtret. Optimeringen utförs med avseende på att minimera energin i stoppbandet samt att optimera för ett så platt passband som möjligt. Vi behandlar både FIR och IIR filter. Optimeringsmodellen implementeras i MATLAB med hjälp av modelleringsspråket CVX och löses med hjälp av SeDuMi.

Preface

We would like to thank our supervisor Kin Cheong Sou for his supervision and valuable comments throughout this project.

During this project, a journal as well as a time log have been kept describing the progress of the group as well as the individual contributions.

The following lists the individual contributions of each group member:

Jimmy Johansson:
Sections 1, 2, 3.1, 3.3, 4, 5. CVX implementation. Ti*k*Z figures.

Fabian Samuelsson:
Sections 1, 4.1, 4.2.2, 6, 7.4, 8. Spectral factorization algorithms. 2D filters.

Moa Samuelsson:
Sections 1, 3.1, 3.2, 7. MATLAB implementation. MATLAB figures. 2D filters.

List of Notation

$\mathbb{N}$	the natural numbers, $\mathbb{N} = \{1, 2, 3, \dots\}$
$\mathbb{Z}$	the integers, $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$
$\mathbb{R}$	the real numbers
$\mathbb{C}$	the complex numbers
$\bar{z}$	complex conjugate of z
$\operatorname{Re} z$	real part of z
$\operatorname{Im} z$	imaginary part of z
$\arg z$	argument of z, i.e φ if $z = r\mathrm{e}^{\mathrm{i}\varphi}$
$\log x$	the natural logarithm of x
$\deg P$	the degree of the polynomial P
$\|v\|_2$	the Euclidian norm of the vector v
$\|x\|_\infty$	the supremum norm of the signal x
Θ_k^n	$n \times n$ elementary Toeplitz matrix
$\operatorname{Toep}(a_0, a_1, \dots, a_n)$	symmetric Toeplitz matrix with diagonals $a_0, a_1, \dots, a_n$
$\mathbb{S}^n$	space of symmetric $n \times n$ matrices
$\operatorname{tr} A$	trace of the matrix A
$\det A$	determinant of the matrix A
A^T	transpose of the matrix A
A^H	hermitian transpose of the matrix A, $A^H = \overline{A^T}$
$A \succeq 0$	the symmetric matrix A is positive semidefinite
δ_n	Kronecker's delta
$\delta(x)$	Dirac delta function
$\mathcal{F}(f)$	Fourier transform of the function f
$\mathcal{H}(f)$	Hilbert transform of the function f

Abbreviations

LTI	Linear time-invariant
FIR	Finite impulse response
IIR	Infinite impulse response
BIBO	Bounded input bounded output
ROC	Region of convergence
DFT	Discrete fourier transform
FFT	Fast fourier transform
IFFT	Inverse fast fourier transform

Contents

1 Introduction

1.1 Background

Digital filters are important in signal processing applications including for example averaging, denoising and anti-aliasing. The idea behind a filter is to let signals of certain frequencies pass unaffected and suppress signals of unwanted frequencies. One example of a digital filter is a low-pass filter, that attenuates signals of high frequencies. Figure 1 shows an example of a signal with high frequency noise and the signal after it is filterated using a low-pass filter.

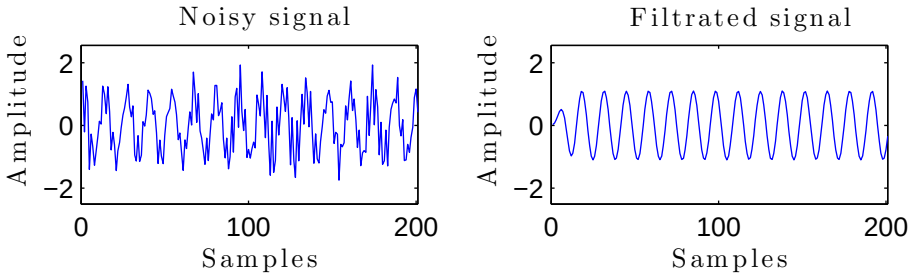

Figure 1: Example of noisy signal that is filtrated.

This can for example be utilized for removing high frequency noise in audio and video signals. The ideal low-pass filter would be designed such that it rejects all signals above a certain frequency and leaves signals with other frequency components unaffected. The range of rejected frequencies, $[\omega_s, \pi]$, is referred to as the stopband while the range of unaffected frequencies, $[0, \omega_p]$, is called the passband. However, an ideal low-pass filter is not physically realizable as we shall see, and therefore there is a need for design techniques where the filter is designed to perform as close as possible to given specifications. A desired specification is not always possible, hence trade-offs between different properties of the filter has to be considered.

In this work we shall consider digital filters that are special cases of discrete linear time-invariant causal systems. Mathematically, a discrete linear time-invariant causal system can be described by its transfer function

$$H(z) = \sum_{k=0}^{n} h_k z^{-k}, \quad h_k \in \mathbb{R}, \quad k = 0, 1, \ldots, n, \tag{1.1}$$

where z is a complex number. Evaluated on the unit circle, i.e. $z = \mathrm{e}^{\mathrm{i}\omega}$, $\omega \in [-\pi, \pi]$, the modulus of the transfer function, $|H(\mathrm{e}^{\mathrm{i}\omega})|$, determines how the amplitude of signals with different frequencies are attenuated. Therefore $|H(\mathrm{e}^{\mathrm{i}\omega})|$ is called the amplitude response of the system [1]. Figure 2 shows the amplitude response of an ideal filter together with a filter designed such that the deviation of the amplitude response is bounded by ε_p and ε_s in the passband and stopband respectively.

1.2 Aim

The aim of this project is to develop an optimization based algorithm for determining the coefficients, $h_0, h_1, \ldots, h_n$, of the transfer function, H, for a digital low-pass filter based on given specifications. We formulate this optimization problem as:

$$\begin{aligned} \text{minimize} \quad & f(H) \\ \text{subject to} \quad & \big||H(\mathrm{e}^{\mathrm{i}\omega})| - 1\big| \leq \varepsilon_p, \quad \forall \omega \in [0, \omega_p], \\ & |H(\mathrm{e}^{\mathrm{i}\omega})| \leq 1 + \varepsilon_p, \quad \forall \omega \in [\omega_p, \omega_s], \\ & |H(\mathrm{e}^{\mathrm{i}\omega})| \leq \varepsilon_s, \quad \forall \omega \in [\omega_s, \pi], \end{aligned} \tag{1.2}$$

where $f(H)$ denotes some quantitative property of H that we want to minimize.

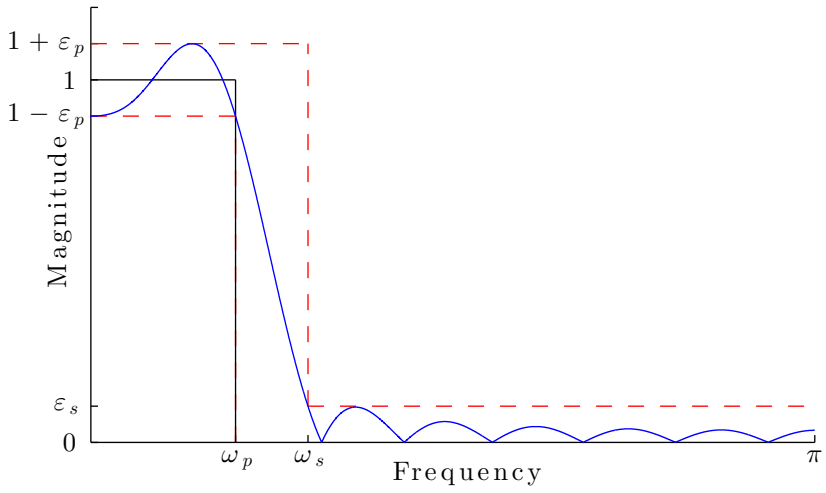

Figure 2: Example of an ideal (black) and designed (blue) low-pass filter with frequency response specifications (red).

1.3 Method

The algorithm is implemented in MATLAB as a function where design specifications are provided by the user and the computed filter coefficients are returned if possible, i.e., the corresponding problem is feasible and can be solved in an efficient manner.

The application of semidefinite programming to digital filters is due to a characterisation of the transfer function in terms of what is known as trigonometric polynomials as described in Section 3. We will show that the square of the frequency response of a finite impulse response digital filter can be expressed as a positive trigonometric polynomial. A positive trigonometric polynomial can in turn be characterized by a semidefinite matrix called the Gram matrix [2]. The semidefinite program is based on finding optimal Gram matrices such that the constraints in (1.2) are satisfied.

1.4 Basic definitions

Definition 1.1. The space of real symmetric $n \times n$ matrices will be referred to as $\mathbb{S}^n$. Given a symmetric matrix, $Q \in \mathbb{S}^n$, we say that Q is positive semidefinite if $v^T Q v \geq 0$ for all $v \in \mathbb{R}^n$. The notation $Q \succeq 0$ will be used to denote positive semidefinite matrices.

In this work we shall frequently encounter a generalization of the polynomials known as Laurent polynomials.

Definition 1.2. An expression in the form

$$a_{-m} z^{-m} + \cdots + a_{-1} z^{-1} + a_0 + a_1 z + \cdots + a_n z^n,$$

with indeterminate $z \in \mathbb{C}$ and where $a_{-m}, \ldots, a_{-1}, a_0, a_1, \ldots, a_n$ are constant complex coefficients, is called a *Laurent polynomial* [3].

If A is a Laurent polynomial, then $P(z) = z^m A(z)$ is a polynomial of degree $n + m$. Note that A and P share the same zeros on $\mathbb{C} \setminus \{0\}$. In this work we shall encounter two important cases of Laurent polynomials known as causal polynomials and trigonometric polynomials.

1.5 Contents

This work is divided into the following parts.

Section 2 presents the necessary theory to understand the frequency description of discrete-time signals and systems. The outline of is largely inspired by [4, pp. 26-36], but it has been modified to cover discrete-time signal and systems.

Section 3 covers the theory of trigonometric polynomials and is based on [2]. It will be shown that the amplitude response of a filter can be represented by a trigonometric polynomial and that a transfer function can be obtained through a process known as spectral factorization.

In Section 4, basic mathematical optimization theory is presented and semidefinite programming is introduced in the context of linear cone programming.

Section 5 presents the optimization model for the filter design problem utilizing the theory established from the previous sections. The optimization is carried out with respect to the stopband energy as described in [2], but we will also cover ripple minimizaion in the passband.

Section 6 covers the numerical methods on spectral factorization needed for obtaining the filter coefficients from the amplitude response, using theory presented in [5] and [6].

In Section 7, the MATLAB implementation of the results from Section 5 is presented together with the results from some filters designed using the algorithm.

2 Discrete-time signals and systems

In this section, we present the mathematical treatment on discrete-time signals and systems. Notions such as the frequency content of a signal shall be made precise, and we will derive the transfer function from the fundamental properties of linear systems and see how it determines the output of the system.

2.1 Discrete-time signals

Definition 2.1. A *discrete-time signal* can be regarded as a function $\mathbb{Z} \to \mathbb{C}$. The value of a signal x at time n will be referred to as x_n.

In practice, a signal is in general real-valued, but allowing complex values allows us to utilize the relation between the complex exponentials and the trigonometric functions. One would also expect a signal to have a beginning, i.e. it takes the value 0 for all times less than some given time. As a simple convention we will define this time as $n = 0$. For reasons that will become apparent later, we shall refer to such signals as *causal*. A visual representation of two signals is shown in Figure 3.

For signals that are bounded, i.e. $x_n \leq M$ for all n and some constant M, we define the supremum norm,

$$\|x\|_\infty = \max_n |x_n|.$$

Equipped with the supremum norm, the space of bounded signals becomes a normed space.

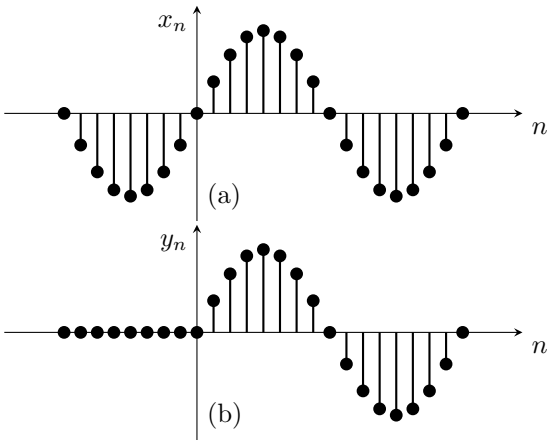

Figure 3: An example of a signal, x, (a), and its causal counterpart, y, (b).

2.2 Linear time-invariant systems

The mathematical formulation of filters is made through operators called *linear systems*.

Definition 2.2. A *linear system*, L, is a linear operator on the space of signals, that to each signal x, called the input, maps a signal $L(x)$, called the output. Linearity means that for arbitrary signals x and y and scalars α and β,

$$L(\alpha x + \beta y) = \alpha L(x) + \beta L(y).$$

This property is also known as the superposition principle.

An important class of linear systems are those that, given a bounded signal, produces a bounded output. This property is known as *stability*.[†]

[†]Stability is commonly referred to as *bounded input bounded output stability* (BIBO) in the literature [1].

Definition 2.3. A linear system, L, is said to be *stable* if there exists a constant C such that for an arbitrary bounded signal x,

$$\|L(x)\|_\infty \leq C\|x\|_\infty.$$

Stability is crucial in both practice and theory. For example, one expects the output of a filter to be bounded given any bounded input. For theoretical purposes, stability will be used as it can be shown that stability of a system, L, is equivalent to L being continuous.

Theorem 2.4. *A linear system, L, is stable if and only if L is a continuous operator.*

Proof. Suppose that L is stable and let (x^k) be a sequence of bounded signals such that $x^k \to x$, $k \to \infty$ for some bounded signal x, i.e. $\|x^k - x\|_\infty \to 0$, $k \to \infty$. From the linearity and the fact that L is stable, it follows that there exists a constant C such that

$$\|L(x^k) - L(x)\|_\infty = \|L(x^k - x)\|_\infty \leq C\|x^k - x\|_\infty,$$

hence $L(x^k) \to L(x)$, $k \to \infty$, i.e. L is continuous. We shall not make direct use of the reverse implication, but a proof can be found in e.g. [7, p. 27]. □

Given a stable system, L, continuity gives that the linearity can be extended to infinite linear combinations of signals $x^1, x^2, x^3, \ldots$:

$$L\left(\sum_{k=0}^{\infty} \alpha_k x^k\right) = L\left(\lim_{n\to\infty} \sum_{k=0}^{n} \alpha_k x^k\right) = \lim_{n\to\infty} L\left(\sum_{k=0}^{n} \alpha_k x^k\right) = \sum_{k=0}^{\infty} \alpha_k L(x^k).$$

For convenience, the notation x_n for the signal x will frequently be used. This has the advantage that a signal, x, shifted in time by k can be expressed as x_{n-k}. Consequently the output of such a signal will be written as $L(x_{n-k})$.[†]

Definition 2.5. Let $y_n = L(x_n)$. L is said to be *time-invariant* if $y_{n-k} = L(x_{n-k})$ for all k and signals x.

In words, time-invariance states that the properties of the system does not change with time.

Definition 2.6. The unit impulse, δ_n, is the signal that takes the value 1 for $n = 0$ and zero otherwise. The *impulse response*, h, of L is defined as $h_n = L(\delta_n)$. An example of a causal impulse response is illustrated in Figure 4.

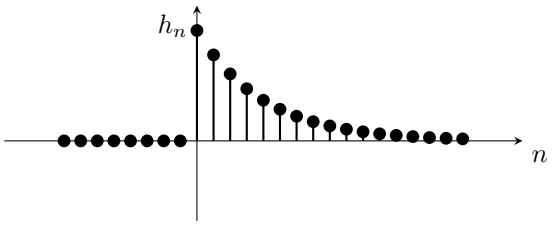

Figure 4: Example of a causal impulse response, h.

We will show that a linear time-invariant (LTI) system is completely determined by its impulse response. In other words, if the impulse response of a system is known, then the output, y, for every input, x, can be determined. First we observe that the value of a signal, x, at time n is given by

$$x_n = \sum_{k=-\infty}^{\infty} x_k \delta_{n-k}$$

[†]Compare with the common notation of $f(x)$ for the function f. The function f shifted by x_0 can then be expressed as $f(x - x_0)$.

since $\delta_{n-k} = 0$ except for $k = n$. This can be interpreted as x being a linear combination of shifted unit impulses with the values of x as weights. Assuming that L is stable, the output, y, at time n is then given by

$$y_n = L(x_n) = L\left(\sum_{k=-\infty}^{\infty} x_k \delta_{n-k}\right) = \sum_{k=-\infty}^{\infty} x_k L(\delta_{n-k}) = \sum_{k=-\infty}^{\infty} x_k h_{n-k}.$$

The third equality follows from the linearity and continuity of L and the last from the fact that L is time-invariant. The last expression is known as the convolution of x and h and is usually denoted by $x * h$. Note that convolution is commutative, i.e. $x * h = h * x$. We formulate the preceding result in the following theorem.

Theorem 2.7. *If L is an LTI system, then*

$$L(x) = h * x \tag{2.1}$$

for all inputs x.

The impulse response is closely related to what is called the transfer function of an LTI system, which we will define next.

Definition 2.8. Let x be a signal. The Z-transform of x, denoted by X, is defined as

$$X(z) = \sum_{k=-\infty}^{\infty} x_k z^{-k},$$

for complex numbers, z.

Definition 2.9. Let L be an LTI system with impulse response h. The Z-transform of h, denoted by H, is called the *transfer function* of L and is given by

$$H(z) = \sum_{k=-\infty}^{\infty} h_k z^{-k}.$$

The transfer function arises naturally in what is called the eigenfunction property for LTI systems.

Theorem 2.10. *Let L be an LTI system with transfer function H. Signals of the form z^n, $z \in \mathbb{C}$, are eigenfunctions of L with eigenvalue $H(z)$.*

Proof. Using the signal z^n as input gives

$$L(z^n) = \sum_{k=-\infty}^{\infty} z^{n-k} h_k = z^n \sum_{k=-\infty}^{\infty} h_k z^{-k} = H(z) z^n.$$

□

An important special case of z^n are the complex exponentials $\mathrm{e}^{\mathrm{i}\omega n}$, which can be used to model sinusoidal signals. For example, the output of the signal $\sin \omega n$ is given by

$$\begin{aligned} L(\sin \omega n) &= \operatorname{Im} L(\mathrm{e}^{\mathrm{i}\omega n}) = \operatorname{Im} H(\mathrm{e}^{\mathrm{i}\omega}) \mathrm{e}^{\mathrm{i}\omega n} = \operatorname{Im} |H(\mathrm{e}^{\mathrm{i}\omega})| \mathrm{e}^{\mathrm{i}(\omega n + \phi)} \\ &= |H(\mathrm{e}^{\mathrm{i}\omega})| \sin(\omega n + \phi) \quad \text{where} \quad \phi = \arg H(\mathrm{e}^{\mathrm{i}\omega}). \end{aligned} \tag{2.2}$$

We see that the output of a sinusoidal signal is another sinusoidal signal of equal frequency but with different phase and amplitude differing by the factor $|H(\mathrm{e}^{\mathrm{i}\omega})|$. In other words, the modulus of the transfer function evaluated on the unit circle determines how the amplitude of sinusoidal signals of different frequencies are affected. Viewed as a function of ω, $H(\mathrm{e}^{\mathrm{i}\omega})$ is called the *frequency response* of L, while $|H(\mathrm{e}^{\mathrm{i}\omega})|$ is known as the *amplitude response* and the phase characteristics $\arg H(\mathrm{e}^{\mathrm{i}\omega})$ is known as the *phase response* [8]. For a signal consisting of a linear combination of sinusoidal signals, the superposition principle gives that an LTI

system will act on each separate signal amplifying or suppressing the amplitude of each signal according to the value of the amplitude response.

In equation (2.2) it was implied that the signal had been active since $-\infty$. It turns out that this idea can be used even if this is not the case. What remains is to show how an LTI system acts on arbitrary signals, which in general are not sinusoidal or even periodic. The idea lies in decomposing the input into eigenfunctions of the system and utilize the linearity of the system. A formal exposition of this possibility is based on the following theorem.

Theorem 2.11. (Inverse Z-transform) *An arbitrary signal, x, can be expressed as*

$$x_n = \frac{1}{2\pi \mathrm{i}} \oint_C X(z) z^{n-1} \, \mathrm{d}z$$

where X is the Z-transform of x and C is a closed curve that encircles the origin and is contained within the region of convergence of X.

Proof. Since X is an analytic function, the result follows immediately from a generalization of Cauchy's integral formula. □

If the region of convergence includes the unit circle, change of variables gives

$$x_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\mathrm{e}^{\mathrm{i}\omega}) \mathrm{e}^{\mathrm{i}\omega n} \, \mathrm{d}\omega. \tag{2.3}$$

We see that an arbitrary signal can be represented as an infinite superposition of complex exponentials with frequencies ranging from $-\pi$ to π with $X(\mathrm{e}^{\mathrm{i}\omega})$ determining the amplitude of each frequency component. We may say that $X(\mathrm{e}^{\mathrm{i}\omega})$ describes the *frequency content* of the signal x. We now show how this is used in generalizing (2.2) to arbitrary signals. Let x be the input for a linear system with transfer function H. The Z-transform of the relation in equation (2.1) gives

$$\begin{aligned} Y(z) &= \sum_{n=-\infty}^{\infty} y_n z^{-n} = \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x_k h_{n-k} z^{-n} = \sum_{k=-\infty}^{\infty} x_k z^{-k} \sum_{n=-\infty}^{\infty} h_{n-k} z^{-(n-k)} \\ &= \sum_{k=-\infty}^{\infty} x_k z^{-k} \sum_{n=-\infty}^{\infty} h_n z^{-n} = H(z) X(z), \end{aligned}$$

hence convolution of signals correspond to multiplication of the transforms. Using this and expressing y as in equation (2.3) gives

$$y_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} Y(\mathrm{e}^{\mathrm{i}\omega}) \mathrm{e}^{\mathrm{i}\omega n} \, \mathrm{d}\omega = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(\mathrm{e}^{\mathrm{i}\omega}) X(\mathrm{e}^{\mathrm{i}\omega}) \mathrm{e}^{\mathrm{i}\omega n} \, \mathrm{d}\omega,$$

hence we see that, just as in (2.2), the transfer function determines the effect on each frequency.

For many purposes, e.g. real-time systems, it is essential that the output of a system at time n does not depend on future values of the input. This motivates the following definition.

Definition 2.12. An LTI system L is said to be *causal* if the output, y_n, only depends on the input values x_k for $k \leq n$.

The following theorem provides a necessary and sufficient condition for an LTI system to be causal in term of its impulse response.

Theorem 2.13. *An LTI system, L, is causal if and only if its impulse response is causal, i.e. $h_n = 0$ for all $n < 0$.*

Proof. According to equation (2.1), the value of the output, y, at time n is given by

$$y_n = \sum_{k=-\infty}^{\infty} h_k x_{n-k}.$$

It follows that y_n is independent of x_k, $k > n$, if and only if $h_n = 0$ for $n < 0$. □

Given a causal signal, x, and a causal system with impulse response h, the following formula gives the value of the output, y, at time n:

$$y_n = \sum_{k=0}^{n} h_k x_{n-k}. \tag{2.4}$$

Example 2.14. The ideal low-pass filter rejects all signals above a certain frequency. Denoting this frequency as ω_s, the transfer function is given by

$$H(\mathrm{e}^{\mathrm{i}\omega}) = \begin{cases} 1, & |\omega| < \omega_s, \\ 0, & \omega_s \leq |\omega| \leq \pi. \end{cases}$$

Using (2.3), the impulse response is given by

$$h_n = \frac{1}{2\pi} \int_{-\omega_s}^{\omega_s} \mathrm{e}^{\mathrm{i}\omega n} \, \mathrm{d}\omega = \frac{\sin \omega_s n}{\pi n}.$$

Since $h_n = 0$ does not hold for all $n < 0$, the system is not causal, hence a causal ideal low-pass filter does not exist.

LTI systems can be categorized into *finite impulse response* (FIR) and *infinite impulse response* (IIR) systems, the difference being that the impulse response for the FIR system has finitely many non-zero values. In practice, FIR and IIR system have their own advantages and disadvantages as we shall see.

2.2.1 Constant coefficient difference equation systems

Given an FIR system, the output can be computed using (2.4), i.e. the constant coefficient difference equation

$$y_k = h_0 x_k + h_1 x_{k-1} + \cdots + h_n x_{k-n}.$$

It is easily verified that any constant coefficient difference equation of this form constitute a FIR system and that the impulse response is given by the coefficients $h_0, h_1, \ldots, h_n$. The transfer function is given by the Laurent polynomial

$$H(z) = \sum_{k=0}^{n} h_k z^{-k},$$

hence we shall refer to Laurent polynomials of this kind as *causal polynomials*.

Example 2.15. A simple example of a FIR low-pass filter is given by

$$y_k = \frac{x_k + x_{k-1}}{2}.$$

The amplitude response is given by

$$|H(\mathrm{e}^{\mathrm{i}\omega})| = \sqrt{\frac{1}{2} + \frac{1}{2}\cos\omega}$$

and is illustrated in Figure 5. Although very simple, it is clear that the filter tends to suppress signals of higher frequencies.

For IIR filters, computing the output using (2.4) is not practical as the number of operations increases for each time step. It may not even be possible as there might not exist a closed form expression for h. In this work, we will work with systems of the form

$$b_0 y_k + b_1 y_{k-1} + \cdots + b_m y_{k-m} = a_0 x_k + a_1 x_{k-1} + \cdots + a_l x_{k-l}$$

with initial conditions $y_n = 0$ for $n < 0$. The transfer function can be computed by taking the Z-transform of both sides:

$$(b_0 + b_1 z^{-1} + \cdots + b_m z^{-m}) Y(z) = (a_0 + a_1 z^{-1} + \cdots + a_l z^{-l}) X(z),$$

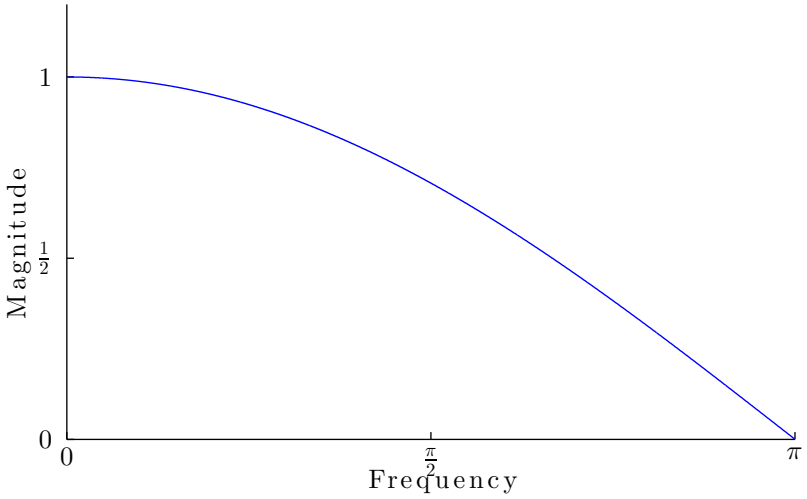

Figure 5: Amplitude response for the filter $y_k = (x_k + x_{k-1})/2$.

thus we end up with the rational transfer function

$$H(z) = \frac{\sum_{i=0}^{l} a_i z^{-i}}{\sum_{j=0}^{m} b_j z^{-j}}.$$

IIR filters have the advantage over FIR filters that they can be designed give the same magnitude performance as FIR filters but with fewer parameters [2, p. 211]. However, the issue of ensuring stability arises when designing IIR filters, something that is not present in the FIR case.

2.2.2 Stability

We continue the theory on the stability concept first presented in Section 2.1.

Theorem 2.16. *A causal LTI system, L, is stable if and only if its impulse response, h, satisfies*

$$\sum_{k=0}^{\infty} |h_k| < \infty. \tag{2.5}$$

Proof. Assume that h satisfies (2.5) and let x be a bounded input with $\|x\|_\infty = C$ for some constant C.

$$|y_n| \leq \sum_{k=0}^{n} |h_k||x_{n-k}| \leq C \sum_{k=0}^{\infty} |h_k|$$

for all n, hence y is bounded. To prove the converse statement, we assume that L is stable. Let

$$x_n = \begin{cases} \frac{h_{-n}}{|h_{-n}|}, & h_{-n} \neq 0, \\ 0, & h_{-n} = 0. \end{cases}$$

Then

$$y_0 = \sum_{k=0}^{\infty} h_k x_{-k} = \sum_{k=0}^{\infty} |h_k| \leq \|y\|_\infty$$

since y is bounded, hence (2.5) holds. □

As FIR filters have impulse responses with finitely many non-zero values, it immediately follows that FIR filters are inherently stable.

Stability can also be expressed in terms of the transfer function.

Theorem 2.17. *An LTI system is stable if and only if its transfer function converges absolutely on the unit circle.*

Proof. Let H be the transfer function of an LTI system. If H converges absolutely on the unit circle, (2.5) is satisfied since

$$\sum_{k=0}^{\infty} |h_k| = \sum_{k=0}^{\infty} |h_k \mathrm{e}^{-\mathrm{i}\omega n}|.$$

□

Perhaps the most useful sufficient criterion for stability is given by the following theorem.

Theorem 2.18. *An LTI system is stable if all poles of its transfer function are contained inside the unit circle.*

Proof. Let L be an LTI system with transfer function H and let z_0 be the pole with the largest magnitude. H is absolutely convergent for all z with $|z| > |z_0|$, hence the region of convergence includes the unit circle and L is stable. An illustration of the region of convergence (ROC) is displayed in Figure 6. □

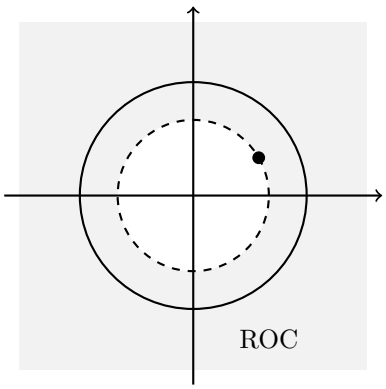

Figure 6: The transfer function of an LTI system is absolutely convergent for all z outside the circle determined by its outermost pole.

2.2.3 Energy

An important quantity concerning signals is energy. Basically, the energy of a signal is proportional to the physical concept of energy determined by the application of the signal.

Definition 2.19. The energy of a signal x is defined as

$$E = \sum_{k=0}^{\infty} |x_k|^2$$

provided the sum exists.

It turns out that there is a close relation between the energy of a signal and its frequency content.

Theorem 2.20. *Let x be a signal and X its Z-transform. Then*

$$E = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(\mathrm{e}^{\mathrm{i}\omega})|^2 \, \mathrm{d}\omega.$$

Proof. Using the relation (2.3), it results that

$$\begin{aligned}\sum_{k=0}^{\infty}|x_k|^2 &= \sum_{k=0}^{\infty}\frac{1}{4\pi^2}\int_{-\pi}^{\pi}X(\mathrm{e}^{\mathrm{i}\omega})\mathrm{e}^{\mathrm{i}\omega k}\,\mathrm{d}\omega\int_{-\pi}^{\pi}\overline{X(\mathrm{e}^{\mathrm{i}\omega'})}\mathrm{e}^{-\mathrm{i}\omega' k}\,\mathrm{d}\omega'\\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi}X(\mathrm{e}^{\mathrm{i}\omega})\int_{-\pi}^{\pi}\overline{X(\mathrm{e}^{\mathrm{i}\omega'})}\frac{1}{2\pi}\sum_{k=0}^{\infty}\mathrm{e}^{\mathrm{i}(\omega-\omega')k}\mathrm{d}\omega'\mathrm{d}\omega\\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi}X(\mathrm{e}^{\mathrm{i}\omega})\int_{-\pi}^{\pi}\overline{X(\mathrm{e}^{\mathrm{i}\omega'})}\delta(\omega-\omega')\mathrm{d}\omega'\mathrm{d}\omega\\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi}|X(\mathrm{e}^{\mathrm{i}\omega})|^2\mathrm{d}\omega.\end{aligned}$$

□

Given an LTI system with transfer function H, we define the stopband energy, E_s, as

$$E_s = \frac{1}{\pi}\int_{\omega_s}^{\pi}|H(\mathrm{e}^{\mathrm{i}\omega})|^2\,\mathrm{d}\omega. \tag{2.6}$$

The stopband energy is a common measure of the performance of a low-pass filter [2], which will be utilized in the optimization applications.

3 Trigonometric polynomials

Consider the transfer function, H, for a causal system,

$$H(z) = \sum_{k=0}^{n} h_k z^{-k}, \quad h_k \in \mathbb{R}, \quad k = 0, 1, \ldots, n,$$

and define a function, R, as $R(z) = H(z)H(z^{-1})$. Since the square of the amplitude response is given by

$$|H(\mathrm{e}^{\mathrm{i}\omega})|^2 = H(\mathrm{e}^{\mathrm{i}\omega})\overline{H(\mathrm{e}^{\mathrm{i}\omega})} = H(\mathrm{e}^{\mathrm{i}\omega})H(\mathrm{e}^{-\mathrm{i}\omega}),$$

$R(z) = |H(z)|^2$ on the unit circle, i.e. when $z = \mathrm{e}^{\mathrm{i}\omega}$, $\omega \in [-\pi, \pi]$. The expression for R is given by

$$R(z) = \sum_{k=-n}^{n} r_k z^{-k},$$

where $r_{-k} = r_k$ and

$$r_k = \sum_{m=k}^{n} h_{m-k} h_m, \quad k = 0, 1. \ldots, n. \tag{3.1}$$

R is known as a *trigonometric polynomial* [2], which will be the topic of this section. The main theorem will be the Riesz-Fejér spectral factorization theorem, which states that a trigonometric polynomial that is non-negative on the unit circle can be factorized as $R(z) = H(z)H(z^{-1})$, where H is a causal polynomial. In terms of linear systems, it results that, given an amplitude response, there exists a corresponding transfer function, which can be obtained through spectral factorization.

For purposes related to optimization, we shall cover a representation of non-negative trigonometric polynomials known as the *Gram matrix representation* as well as establish conditions for trigonometric polynomials that are non-negative on intervals.

3.1 Trigonometric polynomials

We will begin with the basic concepts of trigonometric polynomials. For completeness we will take a more general approach and not restrict ourselves with the case of the coefficients, r_k, $k = 0, 1, \ldots, n$, being real.

Definition 3.1. A *trigonometric polynomial* of degree n is defined as

$$R(z) = \sum_{k=-n}^{n} r_k z^{-k} \quad r_{-k} = \bar{r}_k, \quad r_k \in \mathbb{C}, \quad k = 0, 1, \ldots, n, \quad z \in \mathbb{C}. \tag{3.2}$$

Since $r_{-k} = \bar{r}_k$, we can write R in (3.2) as

$$R(z) = r_0 + \sum_{k=1}^{n} (r_k z^{-k} + \bar{r}_k z^k). \tag{3.3}$$

On the unit circle, we obtain

$$\begin{aligned} R(\mathrm{e}^{\mathrm{i}\omega}) &= r_0 + \sum_{k=1}^{n} |r_k| (\mathrm{e}^{\mathrm{i}\varphi_k} \mathrm{e}^{-\mathrm{i}k\omega} + \mathrm{e}^{-\mathrm{i}\varphi_k} \mathrm{e}^{\mathrm{i}k\omega}) \\ &= r_0 + 2 \sum_{k=1}^{n} |r_k| \underbrace{\frac{\mathrm{e}^{\mathrm{i}(k\omega - \varphi_k)} + \mathrm{e}^{-\mathrm{i}(k\omega - \varphi_k)}}{2}}_{=\cos(k\omega - \varphi_k)} \\ &= r_0 + 2 \sum_{k=1}^{n} a_k \cos k\omega + b_k \sin k\omega, \end{aligned}$$

thus explaining the name trigonometric polynomial. When the coefficients, r_k, $k = 0, 1, \ldots, n$, are real, the trigonometric polynomial, R, will only consist of cosine terms with $a_k = r_k$.

Before we proceed with the Riesz-Fejér spectral factorization theorem, we shall make two useful observations. First, we note that

$$\overline{R(1/\bar{z})} = \sum_{k=-n}^{n} \overline{r_k \bar{z}^k} = \sum_{k=-n}^{n} \bar{r}_k z^k = \sum_{k=-n}^{n} r_{-k} z^k = \sum_{k=-n}^{n} r_k z^{-k} = R(z), \tag{3.4}$$

hence if z is a zero of R, then so is $1/\bar{z}$. The relationship between $1/\bar{z}$ and z is illustrated in Figure 7. We shall refer to $1/\bar{z}$ as the unit circle mirror of z.

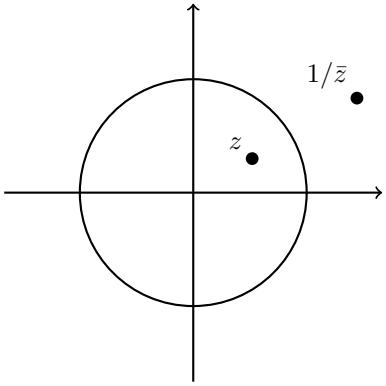

Figure 7: z and its unit circle mirror $1/\bar{z}$.

Secondly, consider the causal polynomial

$$H(z) = \sum_{k=0}^{n} h_k z^{-k}, \tag{3.5}$$

and define

$$\bar{H}(z) = \sum_{k=0}^{n} \bar{h}_k z^{-k}. \tag{3.6}$$

Then, evaluated on the unit circle,

$$\bar{H}(\mathrm{e}^{-\mathrm{i}\omega}) = \sum_{k=0}^{n} \bar{h}_k \mathrm{e}^{\mathrm{i}k\omega} = \sum_{k=0}^{n} \overline{h_k \mathrm{e}^{-\mathrm{i}k\omega}} = \overline{H(\mathrm{e}^{\mathrm{i}\omega})}. \tag{3.7}$$

Theorem 3.2. (Riesz-Fejér spectral factorization theorem) *A trigonometric trigonometric polynomial, R, defined as in* (3.2) *is non-negative on the unit circle if and only if R can be expressed as*

$$R(z) = H(z)\bar{H}(z^{-1}), \tag{3.8}$$

where H, called the spectral factor of R, is a causal polynomial, (3.5)*, and $\bar{H}$ is defined as in* (3.6)*.*

Proof. ($\Leftarrow$) We have that

$$R(z) = H(z)\bar{H}(z^{-1}),$$

and by equation (3.7) we get that

$$R(z) = H(z)\overline{H(z)} = |H(z)|^2,$$

hence R is non-negative on the unit circle.

($\Rightarrow$) Observe that $z^n R(z)$ is a polynomial of degree $2n$, which can be factored into a product of $2n$ monomials. Recall from equation (3.4), that if z is a zero of R, then so is $1/\bar{z}$. Therefore we propose that R can be factorized as

$$R(z) = cz^{-n} \prod_{k=1}^{n} (z - z_k)(z - 1/\bar{z}_k) \tag{3.9}$$

for some constant c. However, since $1/\bar{z} = z$ for z on the unit circle, equation (3.9) is valid if and only if the zeros on the unit circle have even multiplicities. To show that this is the case, assume that $z_0 = \mathrm{e}^{\mathrm{i}\omega_0}$ is a zero on the unit circle of multiplicity m and express R as a power series about z_0:

$$R(z) = \sum_{k=m}^{\infty} c_k (z - z_0)^k.$$

It follows that

$$\frac{\mathrm{d}^n}{\mathrm{d}\omega^n} R(\mathrm{e}^{\mathrm{i}\omega}) = \begin{cases} ac_n, & n = m, \\ 0, & n < m, \end{cases}$$

for some constant a. From the taylor expansion for $R(\mathrm{e}^{\mathrm{i}\omega})$ around ω_0 it follows that $R(\mathrm{e}^{\mathrm{i}\omega})$ changes sign about ω_0 unless m is even. This contradicts the non-negativity condition.

Factoring out $-z/\bar{z}_k$, for $k = 1, 2, \ldots, n$, from the second parenthesis in (3.9), it results that

$$R(z) = d \prod_{k=1}^{n} (z - z_k)(z^{-1} - \bar{z}_k) \tag{3.10}$$

where d is given by

$$d = c \prod_{k=1}^{n} \frac{-1}{\bar{z}_k}.$$

By evaluating (3.10) on the unit circle, it follows that $d \geq 0$ since, on the unit circle, R is non-negativea and $(z - z_k)(z^{-1} - \bar{z}_k) = |z - z_k|^2$. By letting

$$H(z) = \sqrt{d} \prod_{k=1}^{n} (z - z_k), \tag{3.11}$$

it results that $R(z) = H(z)\bar{H}(z^{-1})$. □

For a non-negative trigonometric polynomial with real coefficients, we observe that if z is a zero, then so is $\bar{z}$. Hence the spectral factor, H, consists of pairs of the form $(z - z_k)(z - \bar{z}_k)$ and therefore has real coefficients. We have therefore shown that given an amplitude response, there exists a corresponding transfer function given by the spectral factor, H, of $R(z) = |H(z)|^2$. The above proof provides a method of constructing the transfer function corresponding to a given amplitude response, namely through (3.11). The factorization is not unique since H may involve zeros both inside or outside the unit circle. However, for stability purposes, we shall exclusively deal with the factorization involving the zeros inside or on the unit circle. This factorization is known as the *minimum phase system* [8].

3.2 Gram matrix representation

Every trigonometric polynomial R of degree n defined as in (3.2) can be represented as

$$R(z) = \zeta_n^T(z^{-1}) \cdot Q \cdot \zeta_n(z), \tag{3.12}$$

where $\zeta_n^T(z)$ is the canonical basis $\begin{bmatrix} 1 & z & z^2 & \ldots & z^n \end{bmatrix}^T$ and Q is a Hermitian matrix called the *Gram* matrix, see definition 3.13. We denote $\mathcal{G}(R)$ the set of all Gram matrices associated with $R(z)$.

Definition 3.3. A Hermitian matrix Q is a square matrix where elements

$$q_{ij} = \overline{q_{ji}} \quad \forall\, i,j \quad q \in \mathbb{C} \tag{3.13}$$

or

$$Q = \overline{Q^T} \tag{3.14}$$

Example 3.4. Consider the trigonometric polynomial R of degree 2 with real coefficients r_k. Then

$$R(z) = \begin{bmatrix} 1 & z^{-1} & z^{-2} \end{bmatrix} \begin{bmatrix} q_{00} & q_{01} & q_{02} \\ q_{10} & q_{11} & q_{12} \\ q_{20} & q_{21} & q_{22} \end{bmatrix} \begin{bmatrix} 1 \\ z \\ z^2 \end{bmatrix}$$

is equal to

$$q_{20}z^{-2} + q_{10}z^{-1} + q_{21}z^{-1} + q_{00} + q_{11} + q_{22} + q_{01}z + q_{12}z + q_{02}z^2$$

Identify the coefficients to

$$R(z) = r_2z^{-2} + r_1z^{-1} + r_0 + r_1z + r_2z^2,$$

using (3.13), gives

$$\begin{array}{rcl} r_0 & = & q_{00} + q_{11} + q_{22} \\ r_1 & = & q_{01} + q_{12} \\ r_2 & = & q_{02} \end{array}$$

We see that r_k, in this case, is equal to the sum of the k:th diagonal, which will be the case in general.

Definition 3.5. A matrix where each diagonal has constant entries, i.e.

$$\begin{bmatrix} a_0 & a_1 & a_2 & \dots & a_n \\ a_{-1} & a_0 & a_1 & \ddots & \vdots \\ a_{-2} & a_{-1} & a_0 & \ddots & a_2 \\ \vdots & \ddots & \ddots & \ddots & a_1 \\ a_{-n} & \dots & a_{-2} & a_{-1} & a_0 \end{bmatrix}$$

is called a *Toeplitz* matrix. The special case when the elements of the k:th diagonal is 1 and the others are 0 is refered to as the *elementary* Toeplitz matrix and is denoted Θ_k^n.

Example 3.6.

$$\Theta_0^3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \Theta_1^3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \Theta_2^3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Theorem 3.7. *For any trigonometric polynomial $R \in \mathbb{C}$ and some $Q \in \mathcal{G}(R)$,*

$$r_k = tr[\Theta_k Q], \tag{3.15}$$

where Θ_k is the elementary Toeplitz matrix and $tr[\Theta_k Q]$ is the trace of the matrix product.

Proof. We know from (3.12) that

$$R(z) = \zeta^T(z^{-1}) \cdot Q \cdot \zeta(z).$$

Since the trace is invariant for cyclic permutations, i.e. $tr[ABC] = tr[CAB]$ where A,B and C are matrices, we have that

$$tr[\zeta^T(z^{-1}) \cdot Q \cdot \zeta(z)] = tr[\zeta(z) \cdot \zeta^T(z^{-1}) \cdot Q].$$

$$\zeta(z)\cdot\zeta^T(z^{-1}) = \begin{bmatrix}1\\ z\\ \vdots\\ z^n\end{bmatrix}\begin{bmatrix}1 & z^{-1} & \dots & z^{-n}\end{bmatrix} = \begin{bmatrix}1 & z^{-1} & \dots & z^{-n}\\ z & 1 & \ddots & z^{-n+1}\\ \vdots & \ddots & \ddots & \vdots\\ z^n & z^{n-1} & \dots & 1\end{bmatrix}$$
$$= \sum_{k=-n}^{n}\Theta_k z^{-k}. \tag{3.16}$$

We get

$$R(z) = \sum_{k=-n}^{n} tr[\Theta_k Q]z^{-k}$$

hence

$$r_k = tr[\Theta_k Q]$$

□

A causal polynomial can be represented as $H(z) = h^T\zeta(z^{-1})$, where $h = \begin{bmatrix}h_0 & h_1 & \dots & h_n\end{bmatrix}^T$ contains the coefficients of H.

Theorem 3.8. *A trigonometric polynomial R of degree n is non-negative on the unit circle if and only if there exists a positive semidefinite matrix $Q \in \mathcal{G}(R)$ such that $r_k = tr[\Theta_k Q]$.*

Proof. ($\Leftarrow$) If there exists a $Q \succeq 0$, that is $v^TQv \geq 0\ \forall v$ and v is a vector, such that $r_k = tr[\Theta_k Q]$, then we have

$$R(\mathrm{e}^{\mathrm{i}\omega}) = \begin{bmatrix}1 & \mathrm{e}^{-\mathrm{i}\omega} & \dots & \mathrm{e}^{-\mathrm{i}n\omega}\end{bmatrix}\cdot Q\cdot\begin{bmatrix}1\\ \mathrm{e}^{\mathrm{i}\omega}\\ \vdots\\ \mathrm{e}^{\mathrm{i}n\omega}\end{bmatrix} = \zeta^H(\mathrm{e}^{\mathrm{i}\omega})\cdot Q\cdot\zeta(\mathrm{e}^{\mathrm{i}\omega}) \geq 0$$

where ζ^H is the complex transpose of ζ.
($\Rightarrow$) If R is non-negative, then from the spectral factorization (Thm 3.2) we have that

$$R(z) = H(z)\bar{H}(z^{-1}) = h^T\zeta(z^{-1})\cdot h^H\zeta(z) = \zeta(z^{-1})\cdot hh^H\cdot\zeta(z).$$

Hence

$$Q = hh^H \succeq 0$$

is a positive semidefinite Gram matrix of rank 1 associated with R. □

3.3 Non-negativity on intervals

We now turn to trigonometric polynomials with real coefficients that are non-negative on intervals on the unit circle. Although the square of an amplitude response for a causal LTI system is given by a trigonometric polynomial that is non-negative on the unit circle, the need for trigonometric polynomials that are non-negative on intervals are essential for expressing the constraints of the filter design problem as will be shown in Section 5.

First we will prove a general theorem that polynomials that are non-negative on intervals can be expressed as a weighted sum of squares of two polynomials. This theorem is then generalized to include trigonometric polynomials that are non-negative on intervals on the unit circle. The proofs of these results rely on a transformation between intervals of the unit circle and the real axis, which we will discuss first. For $z = \mathrm{e}^{\mathrm{i}\omega}$, $\omega \in [0, \pi]$, define the transformation

$$x = \frac{z + z^{-1}}{2} = \cos\omega. \tag{3.17}$$

Note that (3.17) is a bijective mapping between $[0, \pi]$ and $[-1, 1]$. For non-negative integers k, define

$$T_k(x) = \frac{z^k + z^{-k}}{2}. \tag{3.18}$$

To express T_k in terms of x we observe that

$$z^{k+1} + z^{-(k+1)} = (z + z^{-1})(z^k + z^{-k}) - (z^{k-1} + z^{-(k-1)}).$$

Therefore the recurrence relation

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x), \quad k \in \mathbb{N},$$

holds. Using (3.18), we obtain $T_0(x) = 1$ and $T_1(x) = x$. Inductively it results that T_k, $k = 0, 1, \ldots$ are polynomials of degree k. In the literature, T_k, $k = 0, 1, \ldots$ are known as *Chebyshev polynomials* of the first kind [9]. From the transformation (3.17), we obtain a correspondence between polynomials with real coefficients and trigonometric polynomials. If P is a polynomial with real coefficients, then R defined as $R(z) = P(x)$, $x = (z + z^{-1})/2$, is a trigonometric polynomial. If $P(x) \geq 0$ for $x \in [-1, 1]$, then R is non-negative on the unit circle.

Theorem 3.9. *Let P be a polynomial of degree $2n$, $n \in \mathbb{N}$, such that $P(x) \geq 0$ for all $x \in [a, b]$. Then P can be expressed as*

$$P(x) = F(x)^2 + (x - a)(b - x)G(x)^2,$$

where F and G are polynomials with $\deg F \leq n$ and $\deg G \leq n - 1$.

Proof. First we assume that $[a, b] = [-1, 1]$, hence we want to prove that

$$P(x) = F(x)^2 + (1 - x^2)G(x)^2. \tag{3.19}$$

Using the transformation $x = (z + z^{-1})/2$ we define R as in the previous paragraph, $R(z) = P(x)$. Observe that R is a trigonometric polynomial that is non-negative on the unit circle since P is non-negative on $[-1, 1]$. By Theorem (3.2), there exists a causal polynomial, H, such that $R(z) = H(z)H(z^{-1})$. Relabelling, H can be written as

$$\begin{aligned} H(z) &= \sum_{k=0}^{2n} h_k z^{-k} = z^{-n} \sum_{k=0}^{2n} h_k z^{-(k-n)} = z^{-n} \sum_{k=-n}^{n} c_k z^{-k} \\ &= z^{-n} \sum_{k=-n}^{n} (a_k + b_k) z^{-k} = z^{-n}(A(z) + B(z)), \end{aligned}$$

where A and B are Laurent polynomials with coefficients satisfying $a_{-k} = a_k$ and $b_{-k} = -b_k$. This is well defined since the system

$$\begin{aligned} c_k &= a_k + b_k \\ c_{-k} &= a_k - b_k \end{aligned}$$

is consistent. It follows that $A(z^{-1}) = A(z)$ while $B(z^{-1}) = -B(z)$. R can now be expressed as

$$R(z) = H(z)H(z^{-1}) = A(z)^2 - B(z)^2. \tag{3.20}$$

Returning to x, we obtain, using (3.18),

$$A(z) = a_0 + 2\sum_{k=0}^{n} a_k \frac{z^k + z^{-k}}{2} = a_0 + 2\sum_{k=0}^{n} a_k T_k(x) = F(x),$$

where F is a polynomial with $\deg F \leq n$. For B we factor out $(z^{-1} - z)$ from $(z^{-k} - z^k)$ and note that the quotient is a trigonometric polynomial.

$$\begin{aligned} B(z) &= \sum_{k=1}^{n} b_k (z^{-k} - z^k) \\ &= (z^{-1} - z) \sum_{k=1}^{n} b_k (z^{-k+1} + \cdots + z^{k-1}) \\ &= \frac{z^{-1} - z}{2} G(x), \end{aligned}$$

where G is a polynomial with $\deg G \leq n-1$. Since

$$\left(\frac{z^{-1}-z}{2}\right)^2 = \left(\frac{z^{-1}+z}{2}\right)^2 - 1,$$

we obtain

$$B(z)^2 = (x^2-1)G(x)^2.$$

Returning to x in (3.20), we obtain the expression in (3.19).

For the general case, we use the transformation

$$x = \frac{(b-a)t + a + b}{2},$$

that maps $[-1,1]$ to $[a,b]$. Then

$$\tilde{P}(t) = P\left(\frac{(b-a)t + a + b}{2}\right) \geq 0$$

for $t \in [-1,1]$, hence there exists polynomials $\tilde{F}$ and $\tilde{G}$ such that

$$\tilde{P}(t) = \tilde{F}(t)^2 + (1-t^2)\tilde{G}(t)^2.$$

To express P as a function of x, we use

$$t = \frac{2x-a-b}{b-a},$$

and compute the factor $(1-t^2)$:

$$(1+t)(1-t) = (x-a)(b-x)\frac{4}{(b-a)^2}.$$

After relabelling, it results that

$$P(x) = F(x)^2 + (x-a)(b-x)G(x)^2.$$

□

The corresponding characterization of trigonometric polynomials that are non-negative on intervals of the unit circle is given in the following theorem.

Theorem 3.10. *Let R be a trigonometric polynomial with real coefficients of even degree, n, such that $R(\mathrm{e}^{\mathrm{i}\omega}) \geq 0$ for all $\omega \in [\alpha,\beta] \subseteq [0,\pi]$. Then, on the unit circle, R can be expressed as*

$$R(\mathrm{e}^{\mathrm{i}\omega}) = R_1(\mathrm{e}^{\mathrm{i}\omega}) + (\cos\omega - \cos\alpha)(\cos\beta - \cos\omega)\, R_2(\mathrm{e}^{\mathrm{i}\omega}), \tag{3.21}$$

where R_1 and R_2 are non-negative trigonometric polynomials with $\deg R_1 \leq n$ and $\deg R_2 \leq n-2$.

Proof. Let $z = \mathrm{e}^{\mathrm{i}\omega}$, $\omega \in [0,\pi]$ and define

$$R(z) = P(x)$$

using the transformation (3.17). Since $R(\mathrm{e}^{\mathrm{i}\omega}) \geq 0$ for $\omega \in [\alpha,\beta]$, it follows that $P(x) \geq 0$ for $x \in [\cos\alpha, \cos\beta]$. By the previous theorem, there exists polynomials F and G of degree $n/2$ and $n/2-1$ respectively such that

$$P(x) = F(x)^2 + (x-\cos\alpha)(\cos\beta - x)G(x)^2.$$

Observe that $F(x)^2$ is a polynomial with real coefficients of degree n. Returning to z, using the correspondence between polynomials with real coefficients and trigonometric polynomials, it results that

$$F(x)^2 = F(\cos\omega)^2 = R_1(\mathrm{e}^{\mathrm{i}\omega}),$$

where R_1 is a trigonometric polynomial of degree n. The same argument gives $G(x)^2 = R_2(\mathrm{e}^{\mathrm{i}\omega})$, where R_2 is a trigonometric polynomial of degree $n-2$. □

Finally we extend the Gram matrix representation to include trigonometric polynomials that are non-negative on intervals.

Theorem 3.11. *Let R be a trigonometric polynomial with real coefficients of even degree n such that $R(\mathrm{e}^{\mathrm{i}\omega}) \geq 0$ for all $\omega \in [\alpha, \beta] \subseteq [0, \pi]$. Then there exist positive semidefinite matrices $Q_1 \in \mathbb{S}^{n+1}$ and $Q_2 \in \mathbb{S}^{n-1}$ such that the coefficients satisfy*

$$\begin{aligned} r_k =& \operatorname{tr} \Theta_k^{n+1} Q_1 + \\ & \operatorname{tr} \left(\left(-\left(ab + \frac{1}{2} \right) \Theta_k^{n-1} + \frac{a+b}{2} (\Theta_{k-1}^{n-1} + \Theta_{k+1}^{n-1}) - \frac{1}{4} (\Theta_{k-2}^{n-1} + \Theta_{k+2}^{n-1}) \right) Q_2 \right). \end{aligned}$$

where $a = \cos \alpha$ and $b = \cos \beta$.

Proof. The result follows by considering the Gram matrix representation of the non-negative trigonometric polynomials R_1 and R_2 in (3.21) and taking into account the factor $(\cos \omega - a)(b - \cos \omega)$. For z on the unit circle, we expand the expression as

$$\begin{aligned} & \left(\frac{z + z^{-1}}{2} - a \right) \left(b - \frac{z + z^{-1}}{2} \right) \\ = & -\left(ab + \frac{1}{2} \right) + \frac{a+b}{2} (z + z^{-1}) - \frac{1}{4} (z^2 + z^{-2}). \end{aligned}$$

The last term in (3.21) is a linear combination of functions of the form $z^m R_2(z)$. If the coefficients for R_2 are given by $r_{2,k} = \operatorname{tr} \Theta_k^{n-1} Q$, then the coefficients for $z^m R_2(z)$ are obtained by the shift $r_{2,k} = \operatorname{tr} \Theta_{k-m}^{n-1} Q$. For example, the coefficients for $z R_2(z)$ are given by $r_{2,k} = \operatorname{tr} \Theta_{k-1}^{n-1} Q$. The result follows after adding up all terms. □

We will abbreviate the formula for the coefficients for a trigonometric polynomial that is non-negative on $[\alpha, \beta] \subseteq [0, \pi]$ as

$$r_k = g_k(Q_1, Q_2; \alpha, \beta).$$

We extend the Gram matrix representation in this fashion and write

$$r_k = g_k(Q)$$

for the coefficients of a trigonometric polynomial that is non-negative on the unit circle. Note that g_k is linear with respect to (Q_1, Q_2) and Q respectively.

4 Mathematical optimization

In this section we present the basic theory of mathematical optimization. In particular we shall introduce the concept of cone programming that allows us to solve optimization problems involving positive semidefinite matrices, which we have seen arises in the characterization of non-negative trigonometric polynomials.

4.1 General theory

Mathematical optimization or mathematical programming concerns the study of the problems of the type: find x^*, provided it exists, such that

$$f(x^*) = \min_{x \in S} f(x),$$

where S is a set and f is a real valued function defined on S. If $\inf_{x \in S} f(x) = -\infty$, the problem is said to be *unbounded*. If S is empty, the problem is said to be *infeasible*. Note that x^* does not need to exist even if S is non-empty.

In this work, we will assume that S is a subset of a real vector space, V, and f is a real valued function defined on V. For optimization problems the following notation is often used:

$$\begin{array}{lll} \text{minimize} & f(x) & \\ \text{subject to} & g_i(x) \leq a_i, & i = 1, 2, \ldots, m, \\ & h_j(x) = b_j, & j = 1, 2, \ldots, n, \\ & x \in S, & \end{array} \tag{4.1}$$

where $f : V \to \mathbb{R}$ is called the *objective function*, $g_i(x) : V \to \mathbb{R}$, $i = 1, 2, \ldots, m$ and $h_i(x) : V \to \mathbb{R}$, $i = 1, 2, \ldots, n$ are the inequality and equality *constraint functions* and the constants $a_1, a_2, \ldots, a_m$ and $b_1, b_2, \ldots, b_n$ are the limits of the constrains. If $V = \mathbb{R}^n$ and all f, g_i, $i = 1, 2, \ldots, m$, h_i, $i = 1, 2, \ldots, n$ in (4.1) are linear functions, i.e. satisfying $f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$ for all $x, y \in \mathbb{R}^n$ and $\alpha, \beta \in \mathbb{R}$, and S is a polyhedron, then (4.1) is said to be a *linear program*.

Definition 4.1. The set $S \subseteq V$ is said to be a *convex set* if

$$\lambda x + (1 - \lambda) y \in S$$

holds for all $x, y \in S$ and $\lambda \in (0, 1)$.

The geometric meaning of a convex set is that all points on the line segment connecting two points in the set also lies in the set, as can be seen in Figure 8.

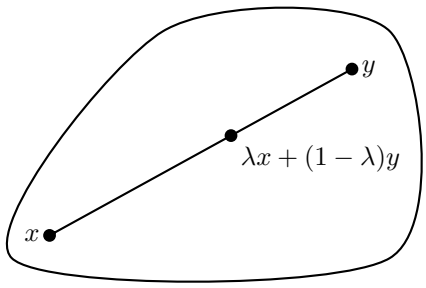

Figure 8: An example of a convex set.

Definition 4.2. Let S be a convex set. $f : S \to \mathbb{R}$ is said to be a *convex function* if

$$f(\lambda x + (1 - \lambda) y) \leq \lambda f(x) + (1 - \lambda) f(y), \quad \forall x, y \in S, \ \forall \lambda \in (0, 1).$$

If all functions f, g_i, $i = 1, \ldots, m$ in (4.1) are convex functions and h_j, $j = 1, 2, \ldots, n$, are linear, then (4.1) is said to be a *convex program*. An essential property of a convex program is that any locally optimal solution is also a globally optimal solution. This is very useful in practice since numerical methods finds locally optimal points.

Proof. Let $x \in S$ be a local minimum and let $y \in S$ be arbitrary. By the convexity and the fact that x is a local minimum, there exists a $\lambda \in (0, 1)$ such that $f(x) \leq f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$. It follows that $f(x) \leq f(y)$, hence x is a global minimum since y is arbitrary. □

4.2 Conic optimization and semidefinite programming

We have seen that a non-negative trigonometric polynomial can be characterized by a positive semidefinite matrix. Optimization of a linear objective function with semidefinite matrices is known as *semidefinite programming*. We will introduce semidefinite programming in a somewhat broader sense known as linear cone optimization, which we will see unifies the notions of some of the most common optimization programs.

Definition 4.3. $K \subseteq V$ is said to be a *cone* if $\lambda x \in K$ holds for all $x \in K$ and $\lambda \geq 0$.

Definition 4.4. Let V, W be vector spaces and $K \subseteq V$ be a cone. Let f be a linear functional defined on V, $A : V \to W$ a linear mapping and $b \in W$. An optimization problem of the form

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && Ax = b, \\ & && x \in K \end{aligned} \tag{4.2}$$

is said to be a *linear cone program*.

If K is convex then (4.2) becomes a convex program. A linear program is a special case of a linear cone program. Namely, if $c^T \in \mathbb{R}^n$ is the matrix for f in the standard basis and $V = \mathbb{R}^n$, $W = \mathbb{R}^m$, $K = \{x \in \mathbb{R}^n : x \geq 0\}$, $A \in \mathbb{R}^{m \times n}$, and $b \in \mathbb{R}^m$, then (4.2) takes the familiar form

$$\begin{aligned} &\text{minimize} && c^T x \\ &\text{subject to} && Ax = b, \\ & && x \geq 0. \end{aligned} \tag{4.3}$$

Semidefinite programming is a special case of convex programming where a linear objective function is optimized over a subset of the cone of positive semidefinite matrices. We will show that a semidefinite program can be put in the form (4.2).

Theorem 4.5. *$K = \{X \in \mathbb{S}^n : X \succeq 0\}$ is a convex cone.*

Proof. Let $X, Y \in K$. For arbitrary $v \in \mathbb{R}^n$ and $\lambda \geq 0$, we have that $v^T \lambda X v \geq 0$ since $v^T X v \geq 0$. This shows that K is a cone. For arbitrary $v \in \mathbb{R}^n$ and $\lambda \in (0, 1)$,

$$v^T(\lambda X + (1 - \lambda)Y)v = v^T \lambda X v + v^T(1 - \lambda)Y v \geq 0$$

since $\lambda \geq 0$ and $(1 - \lambda) \geq 0$, hence K is a convex set. □

A semidefinite program can be expressed as

$$\begin{aligned} &\text{minimize} && \operatorname{tr} CX \\ &\text{subject to} && \operatorname{tr} A_i X = b_i, \quad i = 1, 2, \ldots, m, \\ & && X \succeq 0, \end{aligned}$$

where $X \in \mathbb{S}^n$. Altough slightly different, this is consistent with (4.2) as the trace is a linear function.

We have seen that linear cone programs unifies both linear and semidefinite programs. In this work, we will encounter a third type of optimization problem known as *second order cone programming*.

Definition 4.6. Let $y \in \mathbb{R}^n$ be the decision variable and let $c \in \mathbb{R}^n$ and $A \in \mathbb{R}^{m \times n}$. The constraint

$$\|Ay\|_2 \leq c^T y \tag{4.4}$$

is known as a *second order cone constraint*.

It can easily be shown that second order cone constraints constitute convex cones. In this work we shall encounter an application of second order cone constraints where we wish to minimize the norm of the decision variable x. Such an objective is not linear but by letting $y = \begin{bmatrix} x^T & \gamma \end{bmatrix}^T$ and choosing A and c in (4.4) such that

$$\|x\|_2 \leq \gamma,$$

$\|x\|_2$ is minimized by minimizing the linear objective γ.

4.2.1 Duality

In this section we present the duality theory for linear cone programs. Although we shall not make use of these results directly in the coming sections, it will provide us with some insight on how numerical methods on solving linear cone programs uses duality in order to determine whether an approximate solution is close enough to the optimal solution as well as how to determine when a problem is infeasible.

Let $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and let $K \subseteq \mathbb{R}^n$ be a cone. We will refer to the optimization problem

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax = b, \\ & x \in K \end{aligned} \tag{4.5}$$

as the *primal program*. Note that (4.5) provides a unifying notation for combinations of linear, semidefinite and second order cone programs. For each primal program, there exists what is called a *dual program*. The dual program has some interesting properties as we shall see. The dual program can be derived using the following constructive approach. First we form a *Lagrange function*, L, by moving the constraints to the objective function together with a *Lagrange multiplier* $y \in \mathbb{R}^m$:

$$\begin{aligned} L(x, y) &= c^T x - y^T (Ax - b) \\ &= c^T x - (A^T y)^T x + b^T y \\ &= (c - A^T y)^T x + b^T y. \end{aligned}$$

Next we define what is known as the *dual cone* of K.

Definition 4.7. The *dual cone*, K^*, of K is defined as

$$K^* = \{y \in \mathbb{R}^n : y^T x \geq 0, \ \forall x \in K\}.$$

A visual representation of the relation between a cone and its dual cone is illustrated in Figure 9. Proceeding by minimizing L with respect to x, it results that

$$\inf_{x \in K} L(x, y) = \begin{cases} b^T y, & c - A^T y \in K^*, \\ -\infty, & \text{otherwise.} \end{cases}$$

We arrive at our first important result, known as *weak duality*.

Theorem 4.8. (Weak duality) *If $c - A^T y \in K^*$, then $b^T y \leq c^T x$ for all $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$.*

Proof.

$$c^T x - b^T y = c^T x - (Ax)^T y = x^T c - x^T A^T y = x^T (c - A^T y) \geq 0$$

since $c - A^T y \in K^*$. □

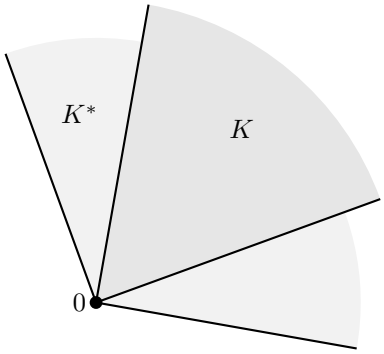

Figure 9: K and its dual cone, K^*.

In words, weak duality tells us that $b^T y$ provides lower bounds on the primal objective function for y such that $c - A^T y \in K^*$. This is of importance in numerical solutions as we know that the primal solution can not be better than $b^T y$. The obvious step is then to compute the lower bound that lies closest to the primal solution. We arrive at the dual program:

$$\begin{aligned} \text{maximize} \quad & b^T y \\ \text{subject to} \quad & A^T y + s = c, \\ & s \in K^*. \end{aligned} \tag{4.6}$$

It would be desirable to conclude that the optimal objective value for the dual program coincides with the primal counterpart, i.e. $b^T y^* = c^T x^*$, a concept known as *strong duality*. Strong duality is in general not the case but there exists a sufficient condition for strong duality known as *Slater's condition* [10, p. 226].

Our second application of duality arises from the question of the existence of solutions to the primal problem. We shall make use of a part of a result known as *Farkas' lemma*.

Theorem 4.9. *If the system $Ax = b$, $x \in K$, is feasible, then the system $b^T y < 0$, $A^T y \in K^*$ is not.*

Proof. Assume that (4.5) is feasible and take $c = 0$. From weak duality, $b^T y \leq 0$ if $-A^T y \in K^*$. Replacing y with $-y$, it follows that $b^T y \geq 0$ if $A^T y \in K^*$. □

Given a program to solve, we wish to either find a feasible solution or otherwise conclude that the program is infeasible. Theorem 4.9 provides us with a way of proving that a program is infeasible by finding one solution to the system $b^T y < 0$, $A^T y \in K^*$. Such a solution is called a *certificate of infeasibility* [10, p. 259].

4.2.2 SeDuMi

SeDuMi is an add-on software for MATLAB that solves symmetric cone programs. SeDuMi uses an primal-dual interior point method, in this the optimization is done for both the primal and dual problem simultaneously. The optimal solution is found by going through the inside of the primal and dual feasible sets. SeDuMi uses a method called *central path* to find a new search direction in each iteration. An initial feasible solution, or a certificate of infeasibility, is found using the self-dual embedding technique. Details on how SeDuMi works are found in [11].

5 Filter optimization

This is the first section devoted to optimization methods for discrete-time filters. Based on the theory presented in the previous sections, we will show how the design of a low-pass filter can be posed as an optimization problem in terms of the transfer function. The characterization of filters in terms of trigonometric polynomials will be used to set up linear matrix constraints involving semidefinite matrices which allows us to utilize the framework of linear cone programs to design filters subject to different constraints. We shall consider both FIR filters and IIR filters with rational transfer functions.

5.1 FIR filters

5.1.1 Magnitude optimization

As shown in Section 2, the transfer function for an FIR filter of order n is given by

$$H(z) = \sum_{k=0}^{n} h_k z^{-k}, \quad h_k \in \mathbb{R}, \quad k = 0, 1, \ldots, n.$$

Suppose that edges ω_p and ω_s defining the passband, $[0, \omega_p]$, and the stopband, $[\omega_s, \pi]$, for a lowpass filter are given. Typical constraints on the amplitude response are then given by $\big||H(\mathrm{e}^{\mathrm{i}\omega})| - 1\big| \leq \varepsilon_p$ for all $\omega \in [0, \omega_p]$ and $|H(\mathrm{e}^{\mathrm{i}\omega})| \leq \varepsilon_s$ for all $\omega \in [\omega_s, \pi]$, i.e. the maximum deviation of the amplitude response with respect to the ideal low-pass filter is bounded by ε_p and ε_s in the passband and stopband respectively. As discussed in Section 2, a typical measure of the performance of a low-pass filter is the suppression of energy in the stopband, E_s, which we will take as our objective function. Formulated as an optimization problem we have:

$$\begin{array}{lll} \text{minimize} & E_s & \\ \text{subject to} & \big||H(\mathrm{e}^{\mathrm{i}\omega})| - 1\big| \leq \varepsilon_p, & \forall \omega \in [0, \omega_p], \\ & |H(\mathrm{e}^{\mathrm{i}\omega})| \leq 1 + \varepsilon_p, & \forall \omega \in [\omega_p, \omega_s], \\ & |H(\mathrm{e}^{\mathrm{i}\omega})| \leq \varepsilon_s, & \forall \omega \in [\omega_s, \pi], \end{array} \tag{5.1}$$

where we have imposed an additional constraint in the transition band for the amplitude response to satisfy $|H(\mathrm{e}^{\mathrm{i}\omega})| \leq 1 + \varepsilon_p$ for all $\omega \in [0, \pi]$. The constraints in (5.1) constitute an allowed region for the amplitude response. An example of this is illustrated in Figure 10.

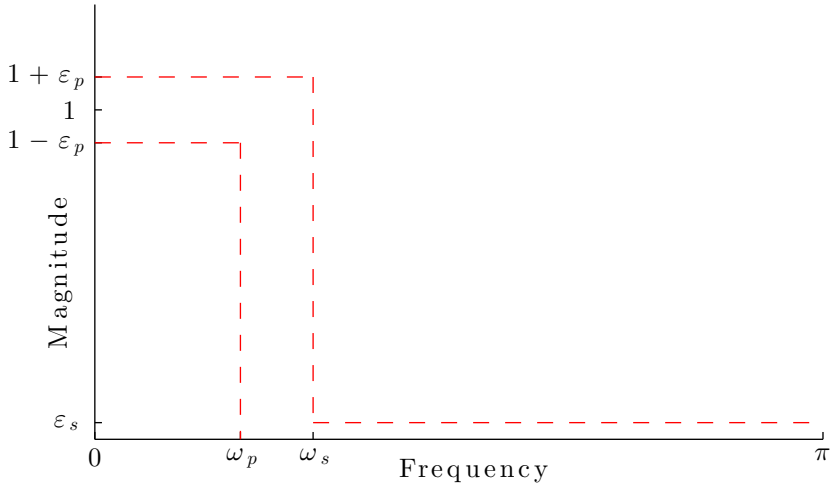

Figure 10: The allowed region for the amplitude response defined by the constraints (5.1).

In order to solve (5.1), the constraints as well as the objective function must be formulated in terms of the filter coefficients. We will use the theory developed in Section 3 to pose (5.1)

as a linear cone program involving semidefinite matrices. First we observe that

$$|H(\mathrm{e}^{\mathrm{i}\omega})|^2 = H(\mathrm{e}^{\mathrm{i}\omega})H(\mathrm{e}^{-\mathrm{i}\omega}) = R(\mathrm{e}^{\mathrm{i}\omega}), \tag{5.2}$$

where R is a trigonometric polynomial. That is, the squared amplitude response is given by a trigonometric polynomial. In terms of R, the inequalities in (5.1) can be expressed as

$$\begin{aligned} (1+\varepsilon_p)^2 - R(\mathrm{e}^{\mathrm{i}\omega}) &\geq 0, \quad \forall \omega \in [0,\pi], \\ R(\mathrm{e}^{\mathrm{i}\omega}) - (1-\varepsilon_p)^2 &\geq 0, \quad \forall \omega \in [0,\omega_p], \\ \varepsilon_s^2 - R(\mathrm{e}^{\mathrm{i}\omega}) &\geq 0, \quad \forall \omega \in [\omega_s,\pi], \\ R(\mathrm{e}^{\mathrm{i}\omega}) &\geq 0, \quad \forall \omega \in [0,\pi]. \end{aligned} \tag{5.3}$$

Observe that all left hand sides are by themselves trigonometric polynomials. In terms of the Gram matrix representation, the non-negativity requirement is equivalent according to Theorem 3.11 to the existence of positive semidefinite matrices satisfying

$$\begin{aligned} &(1+\varepsilon_p)^2\delta_k - r_k = g_k(Q_1), \\ &r_k - (1-\varepsilon_p)^2\delta_k = g_k(Q_2, Q_3; 0, \omega_p), \\ &\varepsilon_s^2\delta_k - r_k = g_k(Q_4, Q_5; \omega_s, \pi), \\ &r_k = g_k(Q_6), \quad k = 0, 1, \ldots, n, \\ &Q_1 \succeq 0, \ldots, Q_6 \succeq 0, \end{aligned}$$

where δ_k denotes Kronecker's delta.

From (2.6), the stopband energy in terms of R is given by

$$E_s = \frac{1}{\pi}\int_{\omega_s}^{\pi} R(\mathrm{e}^{\mathrm{i}\omega})\,\mathrm{d}\omega.$$

Using

$$R(\mathrm{e}^{\mathrm{i}\omega}) = r_0 + 2\sum_{k=1}^{n} r_k \cos k\omega,$$

we obtain

$$E_s = r_0\left(1 - \frac{\omega_s}{\pi}\right) - 2\sum_{k=1}^{n} r_k \frac{\sin k\omega_s}{k\pi}. \tag{5.4}$$

We observe that E_s is linear in terms of the coefficients, r_k. Expressed as a linear cone program, (5.1) takes the form

$$\begin{aligned} \text{minimize} \quad & r_0\left(1 - \frac{\omega_s}{\pi}\right) - 2\sum_{k=1}^{n} r_k \frac{\sin k\omega_s}{k\pi} \\ \text{subject to} \quad & (1+\varepsilon_p)^2\delta_k - r_k = g_k(Q_1), \\ & r_k - (1-\varepsilon_p)^2\delta_k = g_k(Q_2, Q_3; 0, \omega_p), \\ & \varepsilon_s^2\delta_k - r_k = g_k(Q_4, Q_5; \omega_s, \pi), \\ & r_k = g_k(Q_6), \quad k = 0, 1, \ldots, n, \\ & Q_1 \succeq 0, \ldots, Q_6 \succeq 0. \end{aligned} \tag{5.5}$$

By Theorem 3.2, for R satisfying the inequalites, the transfer function can be obtained via spectral factorization. Details for this procedure will be discussed in Section 6.

Apart from minimum energy in the stopband, another desirable characteristic for a low-pass filter would be that it leaves signals within the passband unaltered as much as possible or, in other words, it has a maximally flat amplitude response in the passband. The constraints governing the amplitude response in the passband are given by

$$\begin{aligned} R(\mathrm{e}^{\mathrm{i}\omega}) &\leq (1+\varepsilon_p)^2, \\ R(\mathrm{e}^{\mathrm{i}\omega}) &\geq (1-\varepsilon_p)^2, \end{aligned} \tag{5.6}$$

for $\omega \in [0, \omega_p]$. However, due to the quadratic expressions, we must be able to rewrite the nonlinear constraints in a way that is consistent with the linear cone program as discussed in Section 4. The approach will be based on minimizing the maximum deviation from 1 of the amplitude response in the passband. We will show that this is acheived by minimizing the auxiliary variable γ subject to the constraints

$$\begin{aligned} R(\mathrm{e}^{\mathrm{i}\omega}) &\leq \gamma, \\ R(\mathrm{e}^{\mathrm{i}\omega}) &\geq \gamma - 4\varepsilon, \\ (1+\varepsilon)^2 &\leq \gamma, \end{aligned} \tag{5.7}$$

where ε is another auxiliary descision variable. Note that (5.7) implies $\varepsilon \geq 0$. Let $\alpha \geq 0$ denote the maximum deviation from 1 of the amplitude response in the passband. We will show that the optimal γ is given by $\gamma = (1+\alpha)^2$, hence, by minimizing γ, the maximum deviation will be minimized as well. First we observe that $\gamma = (1+\alpha)^2$ satisfies all three inequalities for example with the choice of $\varepsilon = \alpha$ as this gives

$$\begin{aligned} R(\mathrm{e}^{\mathrm{i}\omega}) &\leq (1+\alpha)^2, \\ R(\mathrm{e}^{\mathrm{i}\omega}) &\geq (1+\alpha)^2 - 4\alpha = (1-\alpha)^2, \end{aligned}$$

which is consistent with the definition of α. To show that this is the optimal γ, we analyse two separate cases. Suppose that the maximum deviation from 1 of the amplitude response in the passband is attained when $|H(\mathrm{e}^{\mathrm{i}\omega})| \geq 1$. Then the first inequality implies that $(1+\alpha)^2 \leq \gamma$, hence we conclude that $\gamma = (1+\alpha)^2$ is optimal. Suppose now instead that the maximum deviation from 1 of the amplitude response in the passband is attained when $|H(\mathrm{e}^{\mathrm{i}\omega})| \leq 1$. Suppose that $\gamma < (1+\alpha)^2$. Then the third inequality gives that $(1+\varepsilon)^2 < (1+\alpha)^2$, hence $\varepsilon < \alpha$. Combined with the second inequality, it follows that $(1-\alpha)^2 \geq \gamma - 4\varepsilon \geq (1+\varepsilon)^2 - 4\varepsilon = (1-\varepsilon)^2 > (1-\alpha)^2$ since $\alpha \leq 1$ in this case. We have reached a contradiction, hence we conclude that $\gamma = (1+\alpha)^2$ is optimal also in this case.

We are still left with a non-linear inequality in (5.7),

$$\gamma - (1+\varepsilon)^2 \geq 0. \tag{5.8}$$

Such constraints can, however, be transformed into a linear matrix inequality. We will show that (5.8) is equivalent to

$$X = \begin{bmatrix} 1 & 1+\varepsilon \\ 1+\varepsilon & \gamma \end{bmatrix} \succeq 0. \tag{5.9}$$

That (5.9) implies (5.8) is obvious since $X \succeq 0$ implies $\det X \geq 0$. The reverse implication is proven by showing that (5.8) implies that X has non-negative eigenvalues. Recall that the eigenvalues, λ, of the 2×2 matrix X are given by the characteristic equation

$$\lambda^2 - \lambda \operatorname{tr} X + \det X = 0.$$

Combining the facts that $\operatorname{tr} X \geq 0$ and $\det X \geq 0$, it follows that the eigenvalues of X are non-negative. With the modified constraints in the passband together with the objective function γ, the linear cone program for minimal ripple in the passband takes the form:

$$\begin{aligned} \text{minimize} \quad & \gamma \\ \text{subject to} \quad & \gamma\delta_k - r_k = g_k(Q_1), \\ & r_k - (\gamma - 4\varepsilon)\delta_k = g_k(Q_2, Q_3; 0, \omega_p), \\ & \varepsilon_s^2\delta_k - r_k = g_k(Q_4, Q_5; \omega_s, \pi), \\ & r_k = g_k(Q_6), \quad k = 0, 1, \ldots, n, \\ & Q_1 \succeq 0, \ldots, Q_6 \succeq 0, \\ & \begin{bmatrix} 1 & 1+\varepsilon \\ 1+\varepsilon & \gamma \end{bmatrix} \succeq 0. \end{aligned}$$

5.1.2 Linear phase filters

Consider the trigonometric polynomial, $\tilde{H}$, with real coefficients,

$$\tilde{H}(z) = \sum_{k=-n}^{n} \tilde{h}_k z^{-k}, \quad \tilde{h}_{-k} = \tilde{h}_k \in \mathbb{R}, \quad k = 0, 1, \ldots, n,$$

and define the transfer function

$$H(z) = z^{-n}\tilde{H}(z) = \sum_{k=0}^{2n} h_k z^{-k}, \tag{5.10}$$

where the vectors of coefficients, h and $\tilde{h}$, are related via

$$h = P\tilde{h}, \quad P = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & \iddots & \vdots & \vdots \\ 0 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}, \tag{5.11}$$

with $P \in \mathbb{R}^{(2n+1)\times(n+1)}$. For ω such that $\tilde{H}(\mathrm{e}^{\mathrm{i}\omega}) \geq 0$, we have that

$$|H(\mathrm{e}^{\mathrm{i}\omega})| = |\mathrm{e}^{-\mathrm{i}n\omega}||\tilde{H}(\mathrm{e}^{\mathrm{i}\omega})| = \tilde{H}(\mathrm{e}^{\mathrm{i}\omega}), \tag{5.12}$$

and

$$\arg H(\mathrm{e}^{\mathrm{i}\omega}) = \arg \mathrm{e}^{-\mathrm{i}n\omega}\tilde{H}(\mathrm{e}^{\mathrm{i}\omega}) = \arg \mathrm{e}^{-\mathrm{i}n\omega} = -n\omega.$$

In other words, the phase response for a filter with transfer function defined as in (5.10) is a linear function over intervals where $\tilde{H}(\mathrm{e}^{\mathrm{i}\omega}) \geq 0$. Such filters are known as *linear phase filters* [8]. Linear phase filters have the advantage of delaying all frequencies by the same amount, thus avoiding phase distortion. To see this, consider again (2.2). The output of the signal $\sin \omega k$ is given by

$$|H(\mathrm{e}^{\mathrm{i}\omega})| \sin(\omega k - \omega n) = |H(\mathrm{e}^{\mathrm{i}\omega})| \sin \omega(k - n),$$

thus each signal, regardless of frequency, is delayed by n (the order of $\tilde{H}$) time steps.

In terms of $\tilde{h}$, the constraints in (5.1) can be expressed as

$$\begin{aligned} &(1 + \varepsilon_p)\delta_k - \tilde{h}_k = g_k(Q_1), \\ &\tilde{h}_k - (1 - \varepsilon_p)\delta_k = g_k(Q_2, Q_3; 0, \omega_p), \\ &\varepsilon_s\delta_k - \tilde{h}_k = g_k(Q_4, Q_5; \omega_s, \pi), \\ &\tilde{h}_k + \varepsilon_s\delta_k = g_k(Q_6, Q_7; \omega_s, \pi), \quad k = 0, 1, \ldots, n, \\ &Q_1 \succeq 0, \ldots, Q_7 \succeq 0. \end{aligned}$$

We only require the filter to have linear phase in the passband, hence the condition $|\tilde{H}(\mathrm{e}^{\mathrm{i}\omega})| \leq \varepsilon_s$ in the stopband is sufficient as given by the last two constraints.

For linear phase filters, ripple minimization becomes especially simple, as the quadratic factors are replaced with linear ones due to (5.12). Taking ε_p as our objective function, the linear cone program takes the form

$$\begin{aligned} \text{minimize} \quad & \varepsilon_p \\ \text{subject to} \quad & (1 + \varepsilon_p)\delta_k - \tilde{h}_k = g_k(Q_1), \\ & \tilde{h}_k - (1 - \varepsilon_p)\delta_k = g_k(Q_2, Q_3; 0, \omega_p), \\ & \varepsilon_s\delta_k - \tilde{h}_k = g_k(Q_4, Q_5; \omega_s, \pi), \\ & \tilde{h}_k + \varepsilon_s\delta_k = g_k(Q_6, Q_7; \omega_s, \pi), \quad k = 0, 1, \ldots, n, \\ & Q_1 \succeq 0, \ldots, Q_7 \succeq 0. \end{aligned}$$

Energy minimization is still possible, but since the optimization is carried out with respect to $\tilde{h}$, we must be able to express the stopband energy, E_s, in terms of $\tilde{h}$. Define a trigonometric polynomial, R, as $R(z) = H(z)H(z^{-1})$, and recall that R is equal to the squared magnitude response on the unit circle. We shall use the formula for the stopband energy in terms of the coeffcients, r_k. Using elementary Toeplitz matrices, the expression for the coefficients (3.1) can be written as the quadratic form $r_k = h^T \Theta_k^{2n+1} h$. Using the expression for the stopband energy in terms of r_k, (5.4), it results that

$$E_s = \sum_{k=-2n}^{2n} c_k r_k = h^T \left(\sum_{k=-2n}^{2n} c_k \Theta_k^{2n+1} \right) h = h^T C h, \tag{5.13}$$

where $C = \mathrm{Toep}(c_0, c_1, \ldots, c_{2n})$ is a symmetric Toeplitz matrix with elements

$$c_k = \begin{cases} 1 - \omega_s/\pi, & k = 0, \\ -\dfrac{\sin k\omega_s}{k\pi}, & k = 1, 2, \ldots, 2n. \end{cases}$$

Finally, from (5.11) we obtain the expression for the stopband energy in terms of $\tilde{h}$. $E_s = \tilde{h}^T \tilde{C} \tilde{h}$, where $\tilde{C} = P^T C P$. The standard way of handling minimization of a quadratic form is to introduce it in a second order cone constraint bounded by an auxiliary variable. From (5.13) and the fact that $E_s \geq 0$, it results that $\tilde{C} \succeq 0$, hence $\tilde{C}$ is diagonalizable with non-negative eigenvalues and therefore there exists a square root $\tilde{C}^{1/2}$. Through the linear transformation $y = \tilde{C}^{1/2}\tilde{h}$ and the second order cone constraint $\|y\|_2 \leq \gamma$, we observe that minimizing the auxiliary variable γ, E_s is minimized as well. The linear cone program for minimum stopband energy for a linear phase filter takes the form

$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & (1+\varepsilon_p)\delta_k - \tilde{h}_k = g_k(Q_1), \\ & \tilde{h}_k - (1-\varepsilon_p)\delta_k = g_k(Q_2, Q_3; 0, \omega_p), \\ & \varepsilon_s \delta_k - \tilde{h}_k = g_k(Q_4, Q_5; \omega_s, \pi), \\ & \tilde{h}_k + \varepsilon_s \delta_k = g_k(Q_6, Q_7; \omega_s, \pi), \quad k = 0, 1, \ldots, n, \\ & y = \tilde{C}^{1/2}\tilde{h}, \\ & \|y\|_2 \leq \gamma, \\ & Q_1 \succeq 0, \ldots, Q_7 \succeq 0. \end{array}$$

Note that the vector of filter coefficients, h, is obtained from the simple transformation (5.11) thus no spectral factorization is required in this case.

5.2 IIR filters

In this section we consider IIR filters with rational transfer functions, i.e.

$$H(z) = \frac{\sum_{i=0}^{l} a_i z^{-i}}{\sum_{j=0}^{m} b_j z^{-j}} = \frac{A(z)}{B(z)}. \tag{5.14}$$

As it can be shown that the stopband energy for (5.14) can not be expressed as a convex function in terms of the filter coefficients [2], we can not use the linear cone program to do energy minimization. However, a feasibility problem for a magnitude constrained filter can be constructed using the same ideas used earlier.

As in Section 5.1.1, we shall work with the square of the amplitude response:

$$|H(\mathrm{e}^{\mathrm{i}\omega})|^2 = \frac{A(\mathrm{e}^{\mathrm{i}\omega})A(\mathrm{e}^{-\mathrm{i}\omega})}{B(\mathrm{e}^{\mathrm{i}\omega})B(\mathrm{e}^{-\mathrm{i}\omega})} = \frac{R_a(\mathrm{e}^{\mathrm{i}\omega})}{R_b(\mathrm{e}^{\mathrm{i}\omega})},$$

where R_a and R_b are trigonometric polynomials. The inequalities in (5.1) can immediately be generalized using (5.3), hence

$$\begin{aligned}
(1+\varepsilon_p)^2 R_b(\mathrm{e}^{\mathrm{i}\omega}) - R_a(\mathrm{e}^{\mathrm{i}\omega}) &\geq 0, \quad \forall \omega \in [0,\pi], \\
R_a(\mathrm{e}^{\mathrm{i}\omega}) - (1-\varepsilon_p)^2 R_b(\mathrm{e}^{\mathrm{i}\omega}) &\geq 0, \quad \forall \omega \in [0,\omega_p], \\
\varepsilon_s^2 R_b(\mathrm{e}^{\mathrm{i}\omega}) - R_a(\mathrm{e}^{\mathrm{i}\omega}) &\geq 0, \quad \forall \omega \in [\omega_s,\pi], \\
R_a(\mathrm{e}^{\mathrm{i}\omega}) \geq 0,\ R_b(\mathrm{e}^{\mathrm{i}\omega}) &\geq 0, \quad \forall \omega \in [0,\pi].
\end{aligned} \tag{5.15}$$

However, the cancellation of the denominator, introduces the trivial solution in (5.15). Since multiplication by a constant in (5.14) does not change the transfer function, this issue is resolved by imposing a normalization constraint:

$$\sum_{j=0}^{m} b_j^2 = 1.$$

From (3.1), it follows that this is equivalent to the simple condition $r_{b,0} = 1$. The linear cone feasibility program can now be expressed as to find coefficients $r_{a,0}, r_{a,1}, \ldots, r_{a,l}$ and $r_{b,0}, r_{b,1}, \ldots, r_{b,m}$ such that

$$\begin{aligned}
&(1+\varepsilon_p)^2 r_{b,k} - r_{a,k} = g_k(Q_1), \\
&r_{a,k} - (1-\varepsilon_p)^2 r_{b,k} = g_k(Q_2, Q_3; 0, \omega_p), \\
&\varepsilon_s^2 r_{b,k} - r_{a,k} = g_k(Q_4, Q_5; \omega_s, \pi), \\
&r_{a,k} = g_k(Q_6), \quad k = 0, 1, \ldots, \max(l,m), \\
&r_{b,0} = 1,\ r_{b,k} = g_k(Q_7), \quad k = 1, 2, \ldots, \max(l,m), \\
&Q_1 \succeq 0, \ldots, Q_7 \succeq 0.
\end{aligned}$$

The filter coefficients $a_0, a_1, \ldots, a_l$ and $b_0, b_1, \ldots, b_m$ are obtained via spectral factorization of R_a and R_b respectively. We require H to be stable, hence the zeros of the spectral factor B must be contained on or inside the unit circle. This is always possible according to Theorem 3.2 but to ensure stability, the zeros must lie strictly inside the unit circle.

6 Spectral factorization

In Section 5.1.1 the optimization was done with respect to the squared frequency response, $|H(e^{i\omega})|^2 = R(e^{i\omega})$. The transfer function H is obtained from R by spectral factorization. The existence of the spectral factors of R is proved by Theorem (3.2). However the task to obtain the spectral factors in practice is not trivial, there exist many methods to do this.

6.1 Method using roots of $R(z)$

The most straightforward method is more or less given by the proof of (3.2):

1. Find the $2n$ zeros of $R(z)$.
2. The zeros are pairs of unit circle mirrors, choose one root from each pair, for minimum phase choose the roots inside or on the unit circle, call them $s_1, \ldots, s_n$.
3. Construct $\tilde{H}(z) = \tilde{h}_0 z^n + \tilde{h}_1 z^{n-1} + \cdots + \tilde{h}_n = \prod_{k=1}^{n}(z - s_k)$.
4. Set $H(z) = a(\tilde{h}_0 + \tilde{h}_1 z^{-1} + \cdots + \tilde{h}_n z^{-n})$, observe that $H(z)$ has the same roots as $\tilde{H}(z)$ and that $H(z)H(z^{-1})$ and $R(z)$ have the same roots for all values of the constant $a \neq 0$.
5. To get $R(z) = H(z)H(z^{-1})$ we set $a = \sqrt{r_0 / \sum_{k=0}^{n} \tilde{h}_k^2}$, since in

$$H(z)H(z^{-1}) = a^2(\tilde{h}_0 + \tilde{h}_1 z^{-1} + \cdots + \tilde{h}_n z^{-n})(\tilde{h}_0 + \tilde{h}_1 z + \cdots + \tilde{h}_n z^n)$$

the constant term is $a^2 \sum_{k=0}^{n} \tilde{h}_k^2$.

In practice this method is stable only for trigonometric polynomials of low order ($n < 20$). For higher orders this method causes errors due to bad performance in the construction of a polynomial from its roots. There exist a number of other methods for spectral factorization that works well for large polynomials as well, in this project a method based on the Hilbert transform is used.

6.2 Kolmogorov's method

The Hilbert transform method, or Kolmogrov's method, uses the Hilbert transform and the (complex) logarithm to compute the frequency response. We assume that R (and H) has no roots on the unit circle i.e $R(e^{i\omega}) \neq 0$ for $\omega \in [0,\pi]$.

Definition 6.1. The Hilbert transform of a function f is defined by

$$\mathcal{H}(f(t)) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{f(x)}{t - x} \mathrm{d}x$$

where the integral is taken as Cauchy principal value [6]. The Hilbert transform can be expressed as the convolution

$$\mathcal{H}(f(t)) = f(t) * \frac{1}{\pi t}.$$

Consider the minimum-phase spectral factor, H, which can be written as

$$H(e^{i\omega}) = |H(e^{i\omega})| e^{i \arg H(e^{i\omega})}.$$

Since H has no zeros on the unit circle we can take the logarithm of both sides

$$\log H(e^{i\omega}) = \log |H(e^{i\omega})| + i \arg H(e^{i\omega}).$$

It can be shown that $\log H(e^{i\omega})$ is an *analytic signal* and that the Hilbert transform can be used to compute the phase [5],

$$\arg H(e^{i\omega}) = -\mathcal{H}(\log |H(e^{i\omega})|).$$

Note that this is only true for a minimum-phase spectral factor. We know that

$$|H(\mathrm{e}^{\mathrm{i}\omega})| = \sqrt{R(\mathrm{e}^{\mathrm{i}\omega})} \iff \log|H(\mathrm{e}^{\mathrm{i}\omega})| = \frac{1}{2}\log R(\mathrm{e}^{\mathrm{i}\omega}).$$

The frequency response of H is given by

$$H(\mathrm{e}^{\mathrm{i}\omega}) = \sqrt{R(\mathrm{e}^{\mathrm{i}\omega})}\mathrm{e}^{-\mathrm{i}\mathcal{H}(\frac{1}{2}\log R(\mathrm{e}^{\mathrm{i}\omega}))}.$$

Theorem 6.2. *(Relation between the Fourier transform and the Hilbert transform)*

$$\mathcal{F}(\mathcal{H}(f(t))) = -\mathrm{i}\,\mathrm{sgn}(t)\mathcal{F}(f(t))$$

where

$$\mathrm{sgn}(t) = \begin{cases} 1, & \text{if } t > 0, \\ 0, & \text{if } t = 0, \\ -1 & \text{if } t < 0. \end{cases}$$

Proof. Hilbert transform of $f(t)$ is the convolution between $f(t)$ and $\frac{1}{\pi t}$ so we can use the convolution Theorem [12]

$$\mathcal{F}(f * g) = \mathcal{F}(f)\mathcal{F}(g).$$

From [13] we know the Fourier transform of $\frac{1}{\pi t}$,

$$\mathcal{F}(\frac{1}{\pi t}) = -\mathrm{i}\,\mathrm{sgn}(t).$$

This gives

$$\mathcal{F}(f(t) * \frac{1}{\pi t}) = \mathcal{F}(f(t))\mathcal{F}(\frac{1}{\pi t}) = -\mathrm{i}\,\mathrm{sgn}(t)\mathcal{F}(f(t)).$$

□

In practice the results above is implemented using the *discrete Fourier transform*, DFT, which is the Z-transform (see Section 2) evaluated on the unit circle sampled at a finite number of equally spaced points. The DFT can be computed using the fast Fourier transform, FFT. The algorithm is described in the following steps.

1. Choose N much larger than n e.g. the smallest power of $2 > 100n$, where n is the order of R. N is the number of points for which we will sample R to be able to use the FFT, take N as a power of 2 to improve performance of FFT.

2. Compute $\{x_n\} = x_0, x_1, \ldots, x_N$ such that

   $$x_l = \tfrac{1}{2}\log R(\mathrm{e}^{\mathrm{i}\omega_l})$$

   for N points on the unit circle, $\omega_l = 2\pi l/N,\ l = 0,1,\ldots,N-1$.

3. Compute the Hilbert transform, by first computing an FFT.

   $$\{X_n\} = FFT(\{x_n\}).$$

   In this transfer domain, using the discrete variant of the relation in Theorem 6.2, the Hilbert transform is computed as

   $$Y_k = \begin{cases} -\mathrm{i}X_k, & \text{if } k = 1,\ldots,\frac{N}{2}-1, \\ 0, & \text{if } k = 0, N/2, \\ \mathrm{i}X_k, & \text{if } k = \frac{N}{2}+1,\ldots,N-1. \end{cases}$$

4. The phase response of the sampled points is obtained by going back to original domain, using the inverse fast Fourier transform IFFT, $\{y_n\} = IFFT(\{Y_n\})$.

5. The frequency response of the spectral factor is given by $H(\mathrm{e}^{\mathrm{i}\omega_l}) = \mathrm{e}^{x_l - \mathrm{i}y_l}$.

6. In the same manner as in Theorem 2.11, we use IFFT to calculate the (approximate) impulse response of $H(z)$, $IFFT(\mathrm{e}^{\{x_n\}-\mathrm{i}\{y_n\}})$, and take the first $n+1$ coefficients as filter coefficients, $h_0, \ldots, h_n$.

7 Implementation

This section gives an example of the MATLAB implementation of the results from Section 5 and results from some filters designed using the algorithm. The complete MATLAB code can be found in the Appendix.

7.1 CVX

CVX is a modelling language implemented in MATLAB to solve convex optimization problems.

Example 7.1. An example of CVX syntax.

```
1. begin_cvx sdp % Semidefinite programming mode
2.    % Decision variables.
3.    variable x(m)
4.    variable Q(n) semidefinite
5.
6.    % Objective function
7.    minimize c*x
8.
9.    % Constraints
10.    A*x == b1
11.    trace(A2*Q) == b2
12. cvx_end
```

7.2 Implementation

The optimization problem described in equation (5.5), magnitude design of low pass FIR filter such that energy in the stop band is minimized, will serve as example for describing the implementation in detail. It follows the structure presented in Example 7.1.

The decision variables are defined as

```
variable r(n+1)
variable Q1(n+1,n+1) semidefinite
variable Q2(n+1,n+1) semidefinite
variable Q3(n-1,n-1) semidefinite
variable Q4(n+1,n+1) semidefinite
variable Q5(n-1,n-1) semidefinite
variable Q6(n+1,n+1) semidefinite
```

where `n` is the degree of the filter. The objective function simply writes

```
minimize c*r
```

where `c` is a vector of elements

$$c_{k+1} = \begin{cases} 1 - \omega_s/\pi, & k = 0, \\ -\dfrac{2\sin k\omega_s}{k\pi}, & k = 1, 2, \ldots, n \end{cases}$$

hence `c*r` is the stopband energy as in equation (5.4).

The constraints are implemented as

```
for k = 0:n
        (1 + ep)^2 * not(k) - r(k+1) == sum(diag(Q1,k));
        r(k+1) - (1 - ep)^2 * not(k) == sum(diag(Q2,k)) + ...
    trace(intervalmatrix(n,k,0,wp)*Q3);
        es^2 * not(k) - r(k+1) == sum(diag(Q4,k)) + ...
    trace(intervalmatrix(n,k,ws,pi)*Q5);
        r(k+1) == sum(diag(Q6,k));
end
```

and corresponds directly to the constraints of (5.5). Note that `not(k)` has the same property as the Kronecker delta. The function `intervalmatrix(n,k,a,b)` returns the matrix

$$\left(-\left(\cos a \cos b + \frac{1}{2} \right) \Theta_k^{n-1} + \frac{\cos a + \cos b}{2} (\Theta_{k-1}^{n-1} + \Theta_{k+1}^{n-1}) - \frac{1}{4} (\Theta_{k-2}^{n-1} + \Theta_{k+2}^{n-1}) \right),$$

see Section 3.3. It in turn uses the function `eltoep(n,k)` that returns an elementary Toeplitz matrix Θ_k^n as in Definition 3.5. The last step is to spectral factorize `r` using the algorithms presented in Section 6. For details on the functions see code in the Appendix. The algorithms that solves for minimizing ripple in the passband, for linear phase filters (referenser) and for IIR filters are similar and can all be found in the Appendix.

Figure 11-15 show the result using each of these algorithms for one specification.

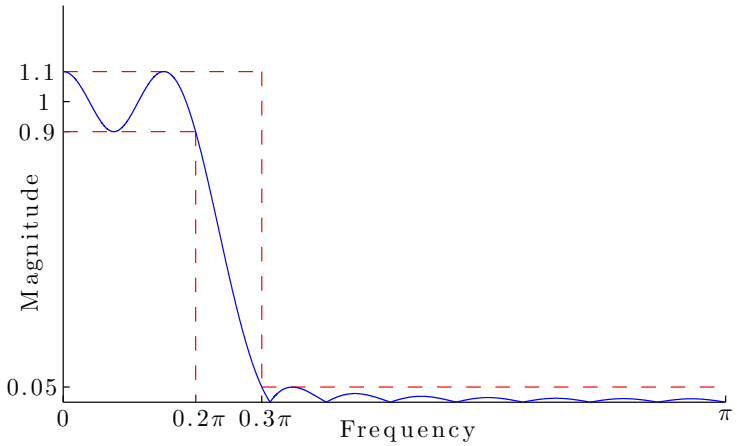

Figure 11: Frequency response of received filter when minimizing the energy in the stopband for $n = 20$, $\omega_p = 0.2$, $\omega_s = 0.3$, $\varepsilon_p = 0.1$, $\varepsilon_s = 0.05$ using FIR magnitude filter design. The stopband energy is $E_s = 6.604 \cdot 10^{-5}$ (implementation in Appendix).

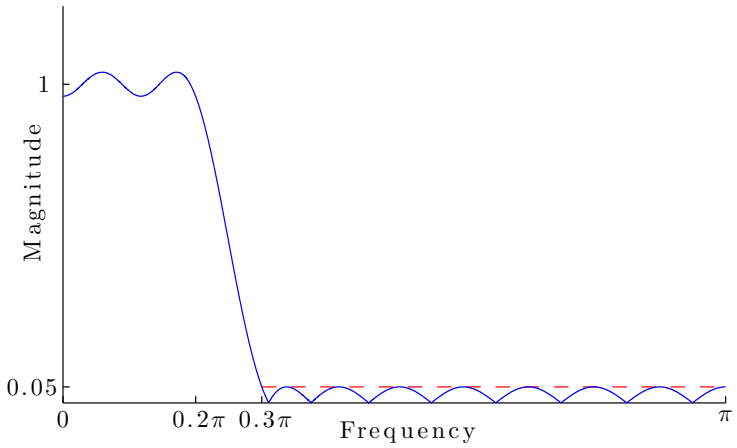

Figure 12: Frequency response of received filter when minimizing the ripple in the passband for $n = 20$, $\omega_p = 0.2$, $\omega_s = 0.3$, $\varepsilon_s = 0.05$ using FIR magnitude filter design. The maximum deviation in the passband is $\varepsilon_p = 0.037$ and the stopband energy is $E_s = 8.7204 \cdot 10^{-4}$ (implementation in Appendix).

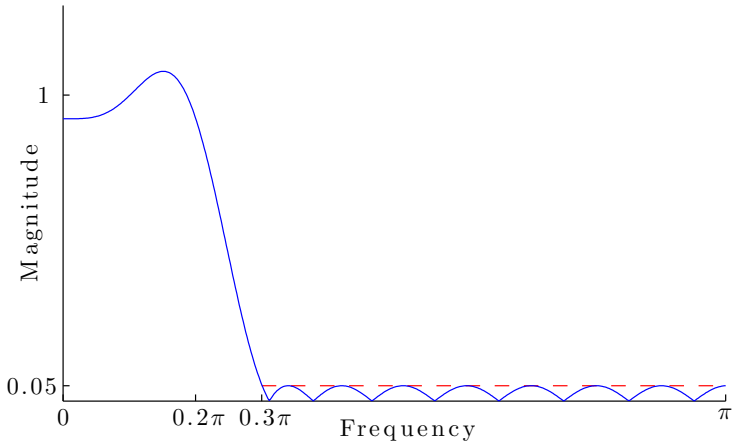

Figure 13: Frequency response of received filter when minimizing the ripple in the passband for $n = 20$, $\omega_p = 0.2$, $\omega_s = 0.3$, $\varepsilon_s = 0.05$ using FIR linear phase filter design. The maximum deviation in the passband is $\varepsilon_p = 0.0775$ and the stopband energy is $E_s = 8.7187 \cdot 10^{-4}$ (implementation in Appendix).

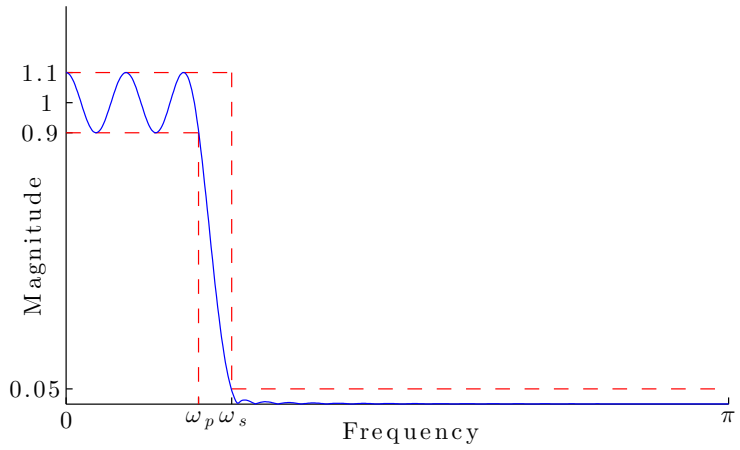

Figure 14: Frequency response of received filter when minimizing the energy in the stopband for $n = 50$, $\omega_p = 0.2$, $\omega_s = 0.25$, $\varepsilon_p = 0.1$, $\varepsilon_s = 0.05$ using FIR linear phase filter design. The stopband energy is $E_s = 8.7651 \cdot 10^{-6}$ (implementation in Appendix).

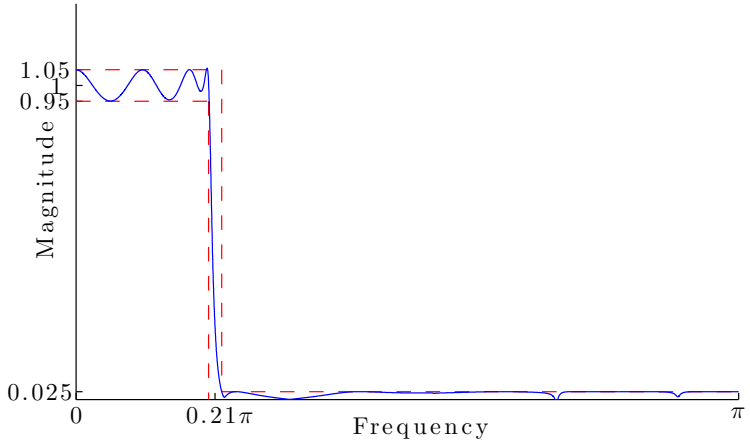

Figure 15: Frequency response of received filter for IIR magnitude filter design with $l = 10$, $m = 20$, $\omega_p = 0.2$, $\omega_s = 0.22$, $\varepsilon_p = 0.05$, $\varepsilon_s = 0.025$ (implementation in Appendix).

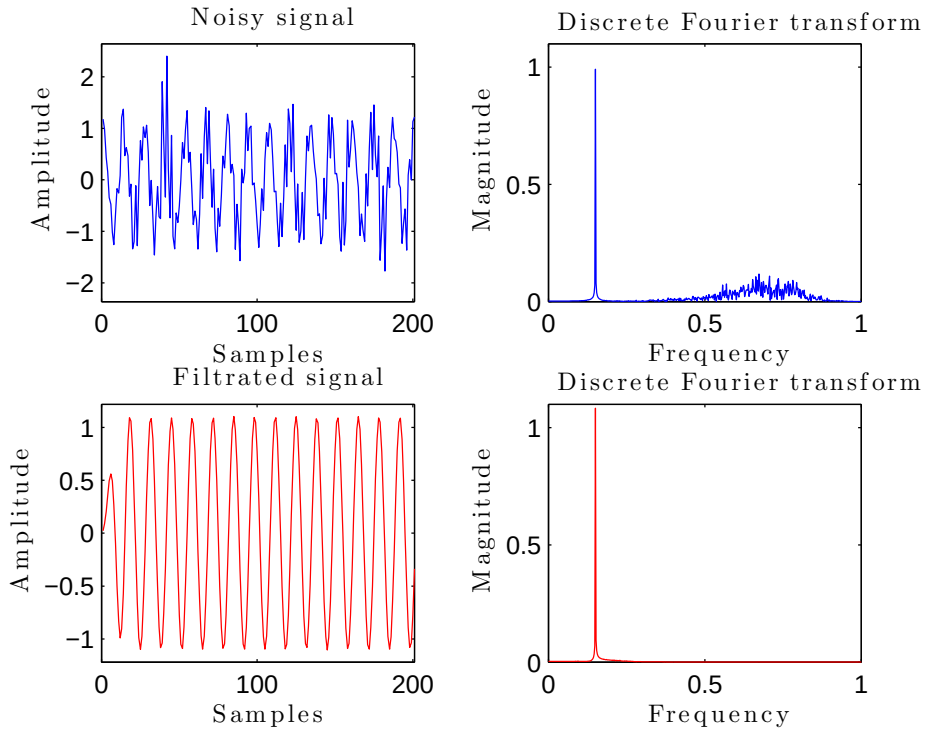

Figure 16: Signal $\cos 0.15\pi n$ with high frequency noise and its frequency contents (blue) that is filtrated by a low-pass filter of degree 20 given as in Figure 11 (red).

7.3 MATLAB function

The optimization algorithms for each filter design are assembled into a MATLAB function `optimalfilter`. The function takes the input arguments:

`n`	degree of the filter
`wp`	upper boundary of the passband, [0,`wp`·π]
`ws`	lower bound of the stopband, [`ws`·π,π]
`ep`	tolerated deviation from 1 in the passband
`es`	tolerated deviation from 0 in the stopband
`options`	stringmap containing user defined options

The variable `options` handle choices between optimization type (magnitude/linear), minimization approach (energy/ripple) and spectral factorization method (hilbert/using roots), where the first mentioned in each parenthesis is default (default/optional). There is also an option to plot the frequency response of the optimized filter together with the boundaries. The function returns two arguments [`h s`] where `h` is a vector with the filter coefficients and `s` contains the status of CVX. The function does not handle exceptions.

7.4 Comparison

Here we show a small comparison with another filter design method, called the Hamming window method. The idea of a windowing method filter is to multiply the impulse response of an ideal filter like the one in Example 2.14 with the an finite length window function in order make an causal FIR filter. So the impulse response of the filter, h_n, can be written as

$$h_n = \hat{h}_n w_n$$

where $\hat{h}_n$ is the ideal filter impulse response and w_n is the window function. The Hamming window function is defined as

$$w_n = \begin{cases} 0.54 - 0.46\cos(\frac{2\pi n}{M}), & \text{if } 0 < n < M, \\ 0, & \text{otherwise.} \end{cases}$$

Here M is the order of the filter. We chose to use Hamming window method in this comparison since it is a common and easy way to design digital FIR filters. In MATLAB it is implemented in the Signal Processesing Toolbox as the function `fir1`. Figure 17 and Table 1 shows an example of a FIR filter, designed to minimize energy in the stopband, compared with a FIR filter designed using Hamming window method. The optimization filters are of squared magnitude design with $\omega_p = 0.4\pi$, $\omega_s = 0.6\pi$ and the tolerances $\varepsilon_s = \varepsilon_p = 0.1$, for the Hamming window method the filters was designed with 0.5π as cutoff frequency.

Table 1: Computation time and stopband energy.

Filter type	Order	Computation time	Stopband energy, E_s
Optimal	10	0.74 s	$3.22 \cdot 10^{-5}$
Hamming window	10	0.0013 s	$2.70 \cdot 10^{-3}$
Optimal	20	1.17 s	$3.01 \cdot 10^{-9}$
Hamming window	20	0.0016 s	$8.66 \cdot 10^{-5}$
Optimal	30	1.94 s	$4.62 \cdot 10^{-11}$
Hamming window	30	0.0017 s	$1.38 \cdot 10^{-6}$

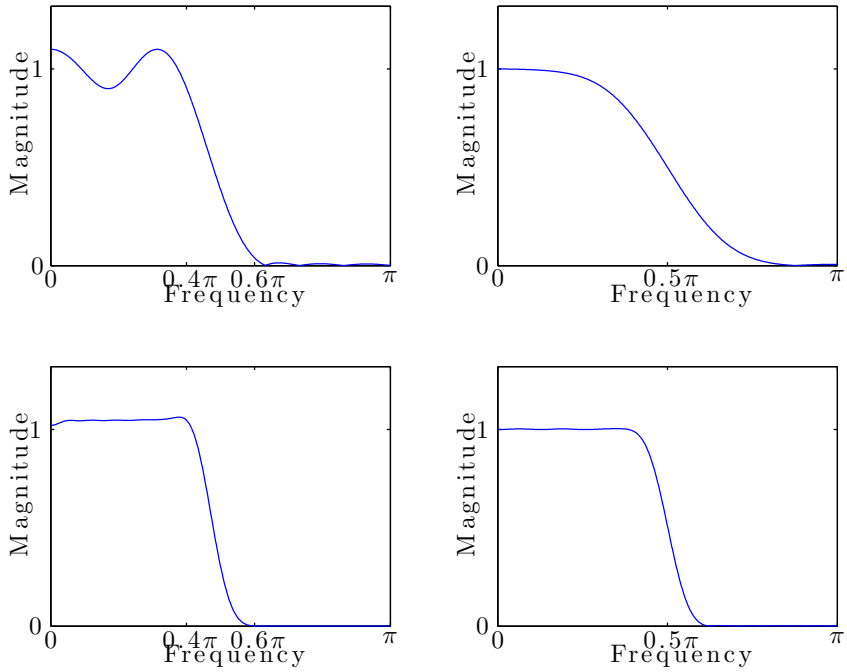

Figure 17: Frequency response of optimal filter (left) compared with Hamming-window filter (right) of order 10(first row) and 30 (second row).

8 Discussion

This work gives an optimization based approach to the design of digital filters. All the theory needed is presented and the algorithm is implemented in MATLAB using the modelling language CVX and solved using SeDuMi. The algorithm was implemented for both FIR and IIR filters. For FIR filters the optimization was carried out with respect to minimizing the stopband energy as well as the passband ripple. Throughout the work only low-pass filters were considered, however the design of high-pass or band-pass filters is completely analogous once the constraints have been altered accordingly.

The produced MATLAB implementation for the different types of filters works well in practice. In Section 7.4 an FIR filter, designed to minimize energy in the stopband, was compared with an existing filter design method, the Hamming window method. It is not possible to draw any general conclusions from this comparison since there exist various other filter design methods, all with different advantages and drawbacks. The main drawback of this optimization based algorithm is the computation time, the benefits is that it is exact, in the sense of constraints, and optimal in some way.

The main focus in the implementation of the algorithm was to make it work and aspects like computation time/efficiency was not prioritised. One way to speed up computation is to use the Gram pair representation, where two smaller semidefinite matrices are used instead of one large, see [2]. One aspect of filters that we have not considered is how the phase response affects the quality of the filtered signal. We have shown how to design linear and minimum phase filters but not discussed in detail why such properties are desirable.

A great amount of work was put into investigating conditions for the existence of solutions to the filter design problem. Necessary conditions for the problem can improve the algorithm, since failure to satisfy a necessary condition implies directly that the problem is infeasible. For example, a necessary condition might be that the order of the filter must be high enough depending on the maximum deviation in the pass- and stopband as well as the width of the transition band. Our main approach was to try to utilize the properties of trigonometric polynomials. We also explored polyhedral relaxations of semidefinite programs in order to find easier problems whose infeasibility would imply infeasibility of the original problem. However no useful results were found. Necessary or sufficient conditions for feasibility is an area that could be researched more.

A natural step for further developing this work would be to design filters for multidimensional signals. We looked into the two-dimensional case, which can be used to filter images for example. Much of the theory can be generalized to the two-dimensional case, but some aspects are different, e.g is it not possible to do spectral factorization and optimize the squared magnitude response [2]. Due to the time frame of this project it was not possible to complete an algorithm for two dimensional signals.

References

[1] Ronald W. Schafer Alan V. Oppenheim. *Discrete-Time Signal Processing*. 3rd ed. Pearson Education, 2010.

[2] Bogdan Dumitrescu. *Positive Trigonometric Polynomials and Signal Processing Applications*. Springer, 2007.

[3] Eric Weisstein. *Laurent Polynomial*. URL: http://mathworld.wolfram.com/LaurentPolynomial.html.

[4] Kjell Holmåker. "Tillämpningar av komplex analys och fourieranalys". Matematiska institutionen, Chalmers & GU.

[5] Julius O. Smith III. *Spectral Audio Signal Processing*. 2011. URL: https://ccrma.stanford.edu/~jos/sasp/Minimum_Phase_Filter_Design.html.

[6] Mathias Johansson. "The Hilbert transform". MA thesis. Växjö University.

[7] Piotr Mikusinski Lokenath Debnath. *Hilbert Spaces with Applications*. Elsevier Academic Press.

[8] Julius O. Smith III. *Introduction to Digital Filters with Audio Applications*. URL: http://ccrma.stanford.edu/~jos/filters/.

[9] Eric Weisstein. *Chebyshev Polynomial of the First Kind*. URL: http://mathworld.wolfram.com/ChebyshevPolynomialoftheFirstKind.html.

[10] Lieven Vandenberghe Stephen Boyd. *Convex Optimization*. Cambridge University Press, 2004.

[11] Jos F. Sturm. *Implementation of Interior Point Methods for Mixed Semidefinite and Second Order Cone Optimization Problems*. Tech. rep. Department of Econometrics, Tilburg University, The Netherlands., 2002.

[12] Eric Weisstein. *Convolution Theorem*. URL: http://mathworld.wolfram.com/ConvolutionTheorem.html.

[13] Eric Weisstein. *Fourier Transform–Inverse Function*. URL: http://mathworld.wolfram.com/FourierTransformInverseFunction.html.

Appendix

FIR magnitude filter design - stopband energy minimization

```
%% FIR magnitude filter design - minimizing energy in stopband
clear
%Degree of filter
n = 20;
%Pass- and stopband limits
wp = 0.2*pi;
ws = 0.3*pi;
%Tolerated deviations in pass- and stopband
ep = 0.1;
es = 0.05;
%Coefficients such that c*r is the energy in the stopband of the filter
c = zeros(1,n+1);
c(1) = (1 - ws/pi);
for k = 1:n
    c(k+1) = -2*sin(k*ws)/(k*pi);
end

%cvx_solver sedumi
cvx_begin sdp

%Decision variables
variable r(n+1)
variable Q1(n+1,n+1) semidefinite
variable Q2(n+1,n+1) semidefinite
variable Q3(n-1,n-1) semidefinite
variable Q4(n+1,n+1) semidefinite
variable Q5(n-1,n-1) semidefinite
variable Q6(n+1,n+1) semidefinite

%Objective function
minimize c*r

%Constraints
for k = 0:n
    %Globally upper boundary
    not(k)*(1 + ep)^2 - r(k+1) == sum(diag(Q1,k));
    %Lower boundary in the passband
    r(k+1) - not(k)*(1 - ep)^2 == sum(diag(Q2,k)) ...
        + trace(intervalmatrix(n,k,0,wp)*Q3);
    %Upper boundary in the stopband
    not(k)*es^2 - r(k+1) == sum(diag(Q4,k)) ...
        + trace(intervalmatrix(n,k,ws,pi)*Q5);
    %Global positivity of R
    r(k+1) == sum(diag(Q6,k));
end
cvx_end

%Spectral factorize r to receive the filter coefficients
h=hfactor(r);
```

FIR magnitude filter design - ripple minimization

```
%Degree of filter
```

```
n = 20;
%Pass- and stopband limits
wp = 0.2*pi;
ws = 0.3*pi;
%Tolerated deviations in stopband
es = 0.05;


cvx_begin sdp
cvx_solver sedumi
%Decision variables
    variable r(n+1)
    variable Q1(n+1,n+1) semidefinite
    variable Q2(n+1,n+1) semidefinite
    variable Q3(n-1,n-1) semidefinite
    variable Q4(n+1,n+1) semidefinite
    variable Q5(n-1,n-1) semidefinite
    variable Q6(n+1,n+1) semidefinite

    variable t
    variable ep
    variable X(2,2) semidefinite

%Objective function
    minimize t

%Constraints
    X(1) == t
    X(2) == 1+ep
    X(3) == 1+ep
    X(4) == 1

    for k = 0:n
        not(k)*t - r(k+1) == sum(diag(Q1,k))
        r(k+1) - not(k)*(t-4*ep) == sum(diag(Q2,k))...
            + trace(intervalmatrix(n,k,0,wp)*Q3)
        not(k)*es^2 - r(k+1) == sum(diag(Q4,k))...
            + trace(intervalmatrix(n,k,ws,pi)*Q5)
        r(k+1) == sum(diag(Q6,k))
    end
cvx_end

%Spectral factorize r to receive the filter coefficients
h = hfactor(r);
```

FIR linear phase filter design - ripple minimization

```
%% Linear phase ripple minimization
clear all
%Degree of filter
n = 20;
ntilde = n/2;
%Pass- and stopband limits
wp = 0.2*pi;
ws = 0.3*pi;
%Tolerated deviation in pass- and stopband
es = 0.05;
```

```
%P matrix to relate h and htilde (h = P*htilde)
I = eye(ntilde+1);
P = [rot90(I);I(2:end,:)];


cvx_begin sdp
cvx_solver sedumi
%Decision variables
variable htilde(ntilde+1)
variable ep
variable Q1(ntilde+1,ntilde+1) semidefinite
variable Q2(ntilde+1,ntilde+1) semidefinite
variable Q3(ntilde-1,ntilde-1) semidefinite
variable Q4(ntilde+1,ntilde+1) semidefinite
variable Q5(ntilde-1,ntilde-1) semidefinite
variable Q6(ntilde+1,ntilde+1) semidefinite
variable Q7(ntilde-1,ntilde-1) semidefinite

%Objective function
minimize ep

%Constraints
    for k = 0:ntilde
        not(k)*(1+ep) - htilde(k+1) == sum(diag(Q1,k))
        htilde(k+1) - not(k)*(1-ep) == sum(diag(Q2,k))...
            + trace(intervalmatrix(ntilde,k,0,wp)*Q3)
        not(k)*es - htilde(k+1) == sum(diag(Q4,k))...
            + trace(intervalmatrix(ntilde,k,ws,pi)*Q5)
        htilde(k+1) + not(k)*es == sum(diag(Q6,k))...
            + trace(intervalmatrix(ntilde,k,ws,pi)*Q7)
    end
cvx_end
%Receive the filter coefficients h from htilde
h = P*htilde;
```

FIR linear phase filter design - stopband energy minimization

```
% Linear phase energy minimization
clear all
%Degree of filter
n = 50;
ntilde = n/2;
%Pass- and stopband limits
wp = 0.2*pi;
ws = 0.25*pi;
%Tolerated deviation in pass- and stopband
ep = 0.1;
es = 0.05;
%Coefficients such that c*r is the energy in the stopband of the filter
c = zeros(1,n+1);
c(1) = (1 - ws/pi);
for k = 1:n
c(k+1) = -sin(k*ws)/(k*pi);
end
%P matrix to relate h and htild e (h = P*htilde)
I=eye(ntilde+1);
```

```
P = [rot90(eye(ntilde+1)); I(2:end,:)];
%Calculate squareroot of ctilde
ctildesqrt = sqrtm(P'*toeplitz(c)*P);

cvx_begin sdp
cvx_solver sedumi
%Decision variables
variable htilde(ntilde+1)
variable aux
variable Y(ntilde+1)
variable Q1(ntilde+1,ntilde+1) semidefinite
variable Q2(ntilde+1,ntilde+1) semidefinite
variable Q3(ntilde-1,ntilde-1) semidefinite
variable Q4(ntilde+1,ntilde+1) semidefinite
variable Q5(ntilde-1,ntilde-1) semidefinite
variable Q6(ntilde+1,ntilde+1) semidefinite
variable Q7(ntilde-1,ntilde-1) semidefinite
% Objective function
minimize aux
% Constraints
for k = 0:ntilde
not(k)*(1 + ep) - htilde(k+1) == sum(diag(Q1,k))
not(k)*(ep - 1) + htilde(k+1) == sum(diag(Q2,k))...
+ trace(intervalmatrix(ntilde,k,0,wp)*Q3)
not(k)*es - htilde(k+1) == sum(diag(Q4,k))...
+ trace(intervalmatrix(ntilde,k,ws,pi)*Q5)
htilde(k+1) + not(k)*es == sum(diag(Q6,k))...
+ trace(intervalmatrix(ntilde,k,ws,pi)*Q7)
end
Y == ctildesqrt*htilde;
norm(Y)-aux <= 0
cvx_end;
%Receive the filter coefficients h from htilde
h = P*htilde;
```

IIR filter design

```
clear
% Numerator order
l = 10;
% Denominator order
m = 20;
% Pass and stopband specification
wp = 0.2*pi;
ws = 0.22*pi;
% Tolerated deviation in pass and stopband
ep = 0.01;
es = 0.05;

n = max(m,l);

cvx_begin sdp
% Decision variables
    variable ra(n+1)
    variable rb(n+1)
    variable Q0(n+1,n+1) semidefinite
    variable Q1(n+1,n+1) semidefinite
```

```
    variable Q2(n-1,n-1) semidefinite
    variable Q3(n+1,n+1) semidefinite
    variable Q4(n-1,n-1) semidefinite
    variable Q5(n+1,n+1) semidefinite
    variable Q6(n+1,n+1) semidefinite

    %Constraints
    for k = 0:n
        (1+ep)^2*rb(k+1) - ra(k+1) == sum(diag(Q0,k))
        ra(k+1) - (1-ep)^2*rb(k+1) == sum(diag(Q1,k))...
            + trace(intervalmatrix(n,k,0,wp)*Q2)
        es^2*rb(k+1) - ra(k+1) == sum(diag(Q3,k))...
            + trace(intervalmatrix(n,k,ws,pi)*Q4)
        ra(k+1) == sum(diag(Q5,k))
        rb(k+1) == sum(diag(Q6,k))
        rb(1) == 1
        if k > l
            ra(k+1) == 0
        end
        if k > m
            rb(k+1) == 0
        end
    end
cvx_end
a=hfactor(ra);
b=hfactor(rb);
```

Functions used by the algorithm

```
function theta = eltoep(n,k)
% ELTOEP Elementray toeplitz matrix.
% theta = ELTOEP(n,k) returns an elementary n-by-n toeplitz matrix with
% ones on the k:th diagonal
theta = zeros(n);
d = ones(n,1);
theta = full(spdiags(d,k,theta));
end

function M = intervalmatrix(n,k,a,b)
% INTERVALMATRIX The interval matrix for rk.
M = (-(cos(a)*cos(b) + 0.5)*eltoep(n-1,k) + ...
0.5*(cos(a) + cos(b))*(eltoep(n-1,k-1) + eltoep(n-1,k+1)) - ...
0.25*(eltoep(n-1,k-2) + eltoep(n-1,k+2)));
end
```

Spectral factorization using roots

```
function h = specfactor(r)
%SPECFACTOR Spectral factorization algorithm
%   Takes the coeffients of R(z) as input and the output is the coefficients of
%   H(z) such that R(z)=H(z)H(z^-1) and H is minimum phase.
n=length(r)-1;
rz=[r(end:-1:2);r];% R(z)*z^n
rootz=roots(rz);

tol =0.0001; %error tolerance to find roots on the unit circle
rin = rootz(abs(rootz)<1-tol); %roots inside the unit circle
```

```
ron = rootz((abs(rootz)<=(1+tol)) & (abs(rootz)>=(1-tol))); %roots on the unit circle
[~,k]=sort(angle(ron)); %sort the roots on the unit circle to just take one of the
%                       pair of roots that actually are the same.
ron = ron(k(1:2:end));
ri=[rin;ron];
htilde=poly(ri); % Creates a polynomial of the picked roots
a=0;
for k=1:n+1 % a is a constant to rescale the coefficients
    a=a+htilde(k)^2;
end
a=sqrt(r(1)/a);
h=real(a*htilde)';

end
```

Hilbert factorization

```
function h=hfactor(rr)
%HFACTOR Spectral factorization using the Hilbert transform algorithm
%   Takes the coeffients of R(z) as input and the output is the coefficients of
%   H(z) such that R(z)=H(z)H(z^-1) and H is minimum phase.
n = length(rr);
mult = 100;
N=2^nextpow2(n*mult); % choose N >> n, power of 2 to improve performance of FFT.
xl = zeros(N,1);
for i=0:N-1
    xl(i+1) = log(response(rr,(2*pi*i/N)))/2; % xl = 1/2 log ( R (e^(i omega_l) ) )
end

X = fft(xl); %fast Fourier transform of xl
XX = [0; X(2:N/2)*-1; 0; X(N/2+2:N)]; % Hilbert transform
XX = XX * 1i;
yl = real( ifft(XX) ); % inverse fast Fourier transform
h = real( ifft (exp(xl-1i*yl) ) ); % inverse fast Fourier transform
                                %to recieve the coefficients of H(z)
h = h(1:n); % return the first n:th coefficients
end

function s=response(r,w)
% Frequency response of R (with coeffients r) for frequency w
s=r(1);
n=length(r)-1;
k=1:n;
s=s+2*cos(k*w)*r(2:end);

end
```

Final program

```
function [h,s] = optimalfilter(n,wp,ws,ep,es,options)
%MYFILTER returns the optimal filter coefficients if possible due to specifications
%n, wp, ws, ep, es, options.
%n - degree of the filter
%wp - sets passband [0,wp*pi]
%ws - sets stopband [ws*pi,pi], ws must be greater than wp
%ep - the maximum tolerated deviation from 1 in the passband
%es - the maximum tolerated deviation from 0 in the stopband
```

```
%options - a string map, choose between 'magnitude'(default) or 'linear' phase,
%choose between minimizing 'energy' in the stopband or 'ripple' in the passband,
%spectral factorize using 'Hilbert' transformation (default) or by
%'spectral', that is method using roots, note that this method only is stable
%for n<20
%choose to 'plot' the result with 'boudaries'

%Define constants
wwp = wp*pi;
wws = ws*pi;
c = zeros(1,n);
c(1) = (1 - wws/pi);
for k = 1:n
    c(k+1) = -2*sin(k*wws)/(k*pi);
end
if ismember('Linear', options) || ismember('linear', options)
    n = n/2;
    c=zeros(1,n);
    c(1) = (1 - wws/pi);
    for k = 1:2*n
    c(k+1) = -sin(k*wws)/(k*pi);
    end
end
cvx_precision('high')
cvx_begin sdp

%Decision variables
variable r(n+1)
variable Q1(n+1,n+1) semidefinite
variable Q2(n+1,n+1) semidefinite
variable Q3(n-1,n-1) semidefinite
variable Q4(n+1,n+1) semidefinite
variable Q5(n-1,n-1) semidefinite
variable Q6(n+1,n+1) semidefinite

if ismember('Linear', options) || ismember('linear', options)

    I = eye(m+1);
    P = [rot90(I);I(2:end,:)];
    if ismember('energy',options) || ismember('Energy', options)

        ctildesqrt = sqrtm(P'*toeplitz(c)*P);

        % Additional decision variable
        variable Q7(n-1,n-1) semidefinite
        variable Y(n+1)
        variable aux

        % Objective function
        minimize aux

        % Constraints
        for k = 0:n
            not(k)*(1 + ep) - r(k+1) == sum(diag(Q1,k))
            not(k)*(ep - 1) + r(k+1) == sum(diag(Q2,k))...
                + trace(intervalmatrix(n,k,0,wwp)*Q3)
            not(k)*es - r(k+1) == sum(diag(Q4,k))...
```

```
                + trace(intervalmatrix(n,k,wws,pi)*Q5)
            r(k+1) + not(k)*es == sum(diag(Q6,k))...
                + trace(intervalmatrix(n,k,ws,pi)*Q7)
        end
        Y == ctildesqrt*r;
        norm(Y)-aux <= 0
    else
        % Additional decision variable
        variable Q7(n-1,n-1) semidefinite
        variable ep

        % Objective function
        minimize ep

        % Constraints
    for k = 0:n
        not(k)*(1+ep) - r(k+1) == sum(diag(Q1,k))
        r(k+1) - not(k)*(1-ep) == sum(diag(Q2,k))...
            + trace(intervalmatrix(n,k,0,wwp)*Q3)
        not(k)*es - r(k+1) == sum(diag(Q4,k))...
            + trace(intervalmatrix(n,k,wws,pi)*Q5)
        r(k+1) + not(k)*es == sum(diag(Q6,k))...
            + trace(intervalmatrix(n,k,wws,pi)*Q7)
    end
    end

        cvx_end
        h = P*r;
        s=cvx_status;

else
    if ismember('ripple',options) || ismember('Ripple',options)
       variable aux
       variable X(2,2) semidefinite
       variable ep

       minimize aux

       X(1) == aux
       X(2) == 1+ep
       X(3) == 1+ep
       X(4) == 1
    for k = 0:n
        not(k)*aux - r(k+1) == sum(diag(Q1,k))
        r(k+1) - not(k)*(aux-4*ep) == sum(diag(Q2,k))...
            + trace(intervalmatrix(n,k,0,wwp)*Q3)
        not(k)*es^2 - r(k+1) == sum(diag(Q4,k))...
            + trace(intervalmatrix(n,k,wws,pi)*Q5)
        r(k+1) == sum(diag(Q6,k))
    end

    else
    % Objective function
    minimize c*r %minimize energy in the stopband

    % Constraints
    for k = 0:n
```

```
        not(k)*(1 + ep)^2 - r(k+1) == sum(diag(Q1,k));
        r(k+1) - not(k)*(1 - ep)^2 == sum(diag(Q2,k))...
            + trace(intervalmatrix(n,k,0,wwp)*Q3);
        not(k)*es^2 - r(k+1) == sum(diag(Q4,k))...
            + trace(intervalmatrix(n,k,wws,pi)*Q5);
        r(k+1) == sum(diag(Q6,k));
    end
    end
    cvx_end ;
    s = cvx_status;

    % choose spectral factorization algorithm
    if ismember('spectral',options) || ismember('Spectral',options)
        h=specfactor(r);
    else
        h=hfactor(r);
    end

end

% Plot solution
if ismember('plot',options)
    figure()
    w = linspace(0,pi,5000);
    y = r(1)*ones(1,5000);

    if ismember('linear',options) || ismember('Linear',options)
        for k = 1:n
            y = y + 2*r(k+1)*cos(k*w);
        end
        plot(w,y)
    else
        for k = 1:n
            y = y + 2*r(k+1)*cos(k*w);
        end
        plot(w,sqrt(y))
    end
    xlabel('Frequency','interpreter','latex')
    ylabel('Magnitude','interpreter','latex')
    axis([0 pi 0 1.2*(1+ep)])

    % Plot boundaries
    if ismember('boundaries',options)
        hold on
        line([wws pi], [es es],'Color',[1 0 0], 'LineStyle','--')
        if ismember('energy',options) || ismember('Energy',options)
            line([wws wws], [1+ep es],'Color',[1 0 0], 'LineStyle','--')
            line([0 wws],[1+ep 1+ep],'Color',[1 0 0], 'LineStyle','--')
            line([0 wwp],[1-ep 1-ep],'Color',[1 0 0], 'LineStyle','--')
            line([wwp wwp],[1-ep 0],'Color',[1 0 0], 'LineStyle','--')
            axis([0 pi 0 1.2*(1+ep)])
        end
    end
end

end
```