On Digital Filter Design with Semidefinite Programming

Michał Przyłuski M.Przyluski@elka.pw.edu.pl

Warsaw University of Technology, Faculty of Electronics and Information Technology, Institute of Control and Computation Engineering

2 July 2016

Semidefinite matrices

Definition

Matrix $A \in \mathbb{R}^{n \times n}$ is *positive semidefinite*, denoted as $A \succcurlyeq 0$, if it is symmetrical and for each $x \in \mathbb{R}^n$: $x^T Ax \geqslant 0$.

Semidefinite matrices

Definition

Matrix $A \in \mathbb{R}^{n\times n}$ is *positive semidefinite*, denoted as $A \succcurlyeq 0$, if it is symmetrical and for each $x \in \mathbb{R}^n$: $x^T Ax \geqslant 0$.

Positive semidefinite cone

Set of positive semidefinite matrices (of size $n \times n$) $S^n_+ = \{X \in S^n : X \succcurlyeq 0\}$ is a cone, called *the cone of positive semidefinite matrices.*

Semidefinite programming features

Semidefinite programming $\subset$ convex optimization.

Semidefinite programming features

Semidefinite programming $\subset$ convex optimization.

- Objective function — linear.
- Feasible set — intersection of the cone of positive semidefinite matrices with an affince space.

Semidefinite programming features

Semidefinite programming $\subset$ convex optimization.

- Objective function — linear.
- Feasible set — intersection of the cone of positive semidefinite matrices with an affince space.

Pros

- all LP problems can be formulated as SDP problems,
- many real-life problems in operations research can be modeled by SDP,
- easily solvable by interior-point methods (special case of conic programming).

Semidefinite programming — general form

Let us define

$$F(x) := F_0 + \sum_{i=1}^{m} x_i F_i,$$

where $F_0, F_1, \ldots, F_m \in S^n$ are symmetric matrices.

Semidefinite programming — general form

Let us define

$$F(x) := F_0 + \sum_{i=1}^{m} x_i F_i,$$

where $F_0, F_1, \ldots, F_m \in S^n$ are symmetric matrices.

Semidefinite Problem

Consider the following:

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{st} \quad & F(x) \succcurlyeq 0. \end{aligned} \qquad (*)$$

Semidefinite programming — general form

Let us define

$$F(x) := F_0 + \sum_{i=1}^{m} x_i F_i,$$

where $F_0, F_1, \ldots, F_m \in S^n$ are symmetric matrices.

Semidefinite Problem

Consider the following:

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{st} \quad & F(x) \succcurlyeq 0. \end{aligned} \qquad (*)$$

Proposition

A symmetrical block-diagonal matrix is positive semi-definite if and only if all the blocks are positive semi-definite.

Convexity

Fact

The problem defined above is convex.

Convexity

Fact

The problem defined above is convex.

Justification

$$\forall x, y \in \mathbb{R}^m : \forall \lambda, \mu \in [0,1], \lambda + \mu = 1 \Rightarrow$$
$$\Rightarrow F(\lambda x + \mu y) = \lambda F(x) + \mu F(y) \succcurlyeq 0.$$

Since $x \mapsto F(x)$ is affine, and $c^T x$ is linear, then it is also a conic linear program.

Were are digital filters?

Digital filters

are essential for:

Were are digital filters?

Digital filters

are essential for:

- telecommunications (television, radiolocation, cellphones),

Were are digital filters?

Digital filters

are essential for:

- telecommunications (television, radiolocation, cellphones),
- data transmission,

Were are digital filters?

Digital filters

are essential for:

- telecommunications (television, radiolocation, cellphones),
- data transmission,
- audio-video data compression,

Were are digital filters?

Digital filters

are essential for:

- telecommunications (television, radiolocation, cellphones),
- data transmission,
- audio-video data compression,
- processing of medical, geological, (...) data

Were are digital filters?

Digital filters

are essential for:

- telecommunications (television, radiolocation, cellphones),
- data transmission,
- audio-video data compression,
- processing of medical, geological, (...) data

Namely — in digital signal processing (DSP).

What are digital filters?

Digital filters

Digital filters with discrete-time are systems processing the input signal $u : \mathbb{Z}_+ \to \mathbb{R}$ to output signal $y : \mathbb{Z}_+ \to \mathbb{R}$.

This is a *very* vague characterization.

What are digital filters?

Digital filters

Digital filters with discrete-time are systems processing the input signal $u : \mathbb{Z}_+ \to \mathbb{R}$ to output signal $y : \mathbb{Z}_+ \to \mathbb{R}$.

This is a *very* vague characterization.

In general...

- A filter can be a Finite Impulse Response (FIR) filter, or
- an Infinite Impulse Response (IIR) filter.

How to describe digital filters?

Impulse response

Impulse response (IR) is a sequence $h : \mathbb{Z}_+ \rightarrow \mathbb{R}$

How to describe digital filters?

Impulse response

Impulse response (IR) is a sequence $h : \mathbb{Z}_+ \to \mathbb{R}$

Remarks

FIR filters can be efficiently described by their *impulse response* — reponse y of the system to a Kronecker δ input (namely $[1\ 0\ 0\ \ldots]$).

How to describe digital filters?

Impulse response

Impulse response (IR) is a sequence $h : \mathbb{Z}_+ \rightarrow \mathbb{R}$

Remarks

FIR filters can be efficiently described by their *impulse response* — reponse y of the system to a Kronecker δ input (namely $[1\ 0\ 0\ \ldots]$).
FIR filters have a finite number of non-zeros in their IR.

How to describe digital filters?

Impulse response

Impulse response (IR) is a sequence $h : \mathbb{Z}_+ \to \mathbb{R}$

Remarks

FIR filters can be efficiently described by their *impulse response* — reponse y of the system to a Kronecker δ input (namely $[1\ 0\ 0\ \ldots]$).
FIR filters have a finite number of non-zeros in their IR.
In case of IIR filters IR is less useful, since it may contain infinitely many non-zeros.

Definitions (FIR)

For an arbitrary input signal u:

$$y(t) = \sum_{\tau=0}^{t} h_{t-\tau} u(\tau).$$

Definitions (FIR)

For an arbitrary input signal u:

$$y(t) = \sum_{\tau=0}^{t} h_{t-\tau} u(\tau).$$

It can be easily extended for input signals defined for all $t \in \mathbb{Z}$:

$$y(t) = \sum_{\tau=-\infty}^{t} h_{t-\tau} u(\tau).$$

Definitions (FIR)

For an arbitrary input signal u:

$$y(t) = \sum_{\tau=0}^{t} h_{t-\tau} u(\tau).$$

It can be easily extended for input signals defined for all $t \in \mathbb{Z}$:

$$y(t) = \sum_{\tau=-\infty}^{t} h_{t-\tau} u(\tau).$$

We will assume now that the IR h — is an absolutely convergent sequence. It means that the filter is stable (bounded input — bounded output).

Definitions (FIR)

For such systems the series

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

is absolutely convergent (for complex $|z| \geqslant 1$).
Such a function $H(z)$ is called *the transfer function*.

Definitions (FIR)

For such systems the series

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

is absolutely convergent (for complex $|z| \geqslant 1$).

Such a function $H(z)$ is called *the transfer function*.

Let $\omega \in [0, 2\pi]$ and consider the system's response to a (complex) input signal $u(t) = e^{j\omega t}$, for $t \in \mathbb{Z}$. It is easy to see that

$$y(t) = \sum_{\tau=-\infty}^{t} h_{t-\tau} e^{j\omega\tau} = \sum_{k=0}^{\infty} h_k e^{j\omega(t-k)} = e^{j\omega t} \sum_{k=0}^{\infty} h_k e^{-j\omega k}.$$

It means that $y(t) = H\left(e^{j\omega}\right) e^{j\omega t}$. And the function assigning ω a complex number $H\left(e^{j\omega}\right)$ is called **the frequency response**.

Finite Impulse Response

Finite Impulse Response filters

are filters, such that their IR has a finite number of non-zeros, namely there exists such n that $h_t = 0$ for $t \geqslant n$.

Finite Impulse Response

Finite Impulse Response filters

are filters, such that their IR has a finite number of non-zeros, namely there exists such n that $h_t = 0$ for $t \geqslant n$.

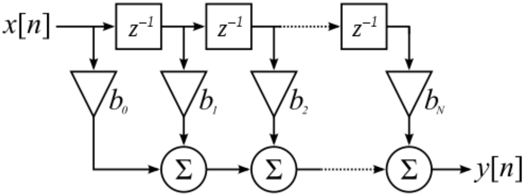
The smallest such n is called *the filter length*.

Finite Impulse Response

Finite Impulse Response filters

are filters, such that their IR has a finite number of non-zeros, namely there exists such n that $h_t = 0$ for $t \geqslant n$.

The smallest such n is called *the filter length*.
The transfer function of such filter is given by $H(z) = \sum_{k=0}^{n-1} h_k z^{-k}$.



Infinite Impulse Response

Infinite Impulse Response filters

are filters, such that their IR can have infinitely many non-zeros.

Infinite Impulse Response

Infinite Impulse Response filters

are filters, such that their IR can have infinitely many non-zeros.

Transfer function

Transfer function of such a filter is given by

$$H(z) = \frac{\sum_{k=0}^{n} a_k z^{-k}}{\sum_{k=0}^{m} b_k z^{-k}}.$$

Infinite Impulse Response

Infinite Impulse Response filters

are filters, such that their IR can have infinitely many non-zeros.

Transfer function

Transfer function of such a filter is given by

$$H(z) = \frac{\sum_{k=0}^{n} a_k z^{-k}}{\sum_{k=0}^{m} b_k z^{-k}}.$$

As you can see the stability is **not** implied.

Infinite Impulse Response

Infinite Impulse Response filters

are filters, such that their IR can have infinitely many non-zeros.

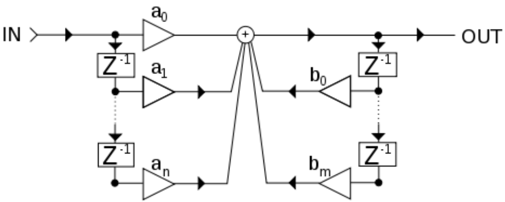
Transfer function

Transfer function of such a filter is given by

$$H(z) = \frac{\sum_{k=0}^{n} a_k z^{-k}}{\sum_{k=0}^{m} b_k z^{-k}}.$$

As you can see the stability is **not** implied.
Pair (n, m) is called *filter length*.

IIR filter schematics



FIR/IIR filters comparison

FIR Filters:

- are always stable,
- require more computational effort.

FIR/IIR filters comparison

FIR Filters:

- are always stable,
- require more computational effort.

Infinite Impulse Response (IIR) Filters:

- an effort has to be made to make them stable,
- provide beter filter parameters for a given filter length,
- there are some valuable results thanks to SOCP methods.

Classical FIR Filter design methods:

Parks-McClennan (1972)

- *"The goal of the algorithm is to minimize the error in the pass and stop bands by utilizing the Chebyshev approximation"* (Wikipedia),

Classical FIR Filter design methods:

Parks-McClennan (1972)

- *"The goal of the algorithm is to minimize the error in the pass and stop bands by utilizing the Chebyshev approximation"* (Wikipedia),
- designed filters are linear-phase (highly sought after feature),

Classical FIR Filter design methods:

Parks-McClennan (1972)

- *"The goal of the algorithm is to minimize the error in the pass and stop bands by utilizing the Chebyshev approximation"* (Wikipedia),
- designed filters are linear-phase (highly sought after feature),
- and equiripple (which amplitude response oscillates uniformly between the tolerance bounds of each band).

Classical FIR Filter design methods:

Parks-McClennan (1972)

- *"The goal of the algorithm is to minimize the error in the pass and stop bands by utilizing the Chebyshev approximation"* (Wikipedia),
- designed filters are linear-phase (highly sought after feature),
- and equiripple (which amplitude response oscillates uniformly between the tolerance bounds of each band).

They are computationally effective, but lack the possibility to add any extra constraints on the filter.

Classical FIR Filter design methods:

Parks-McClennan (1972)

- *"The goal of the algorithm is to minimize the error in the pass and stop bands by utilizing the Chebyshev approximation"* (Wikipedia),
- designed filters are linear-phase (highly sought after feature),
- and equiripple (which amplitude response oscillates uniformly between the tolerance bounds of each band).

They are computationally effective, but lack the possibility to add any extra constraints on the filter.

Linear Programming method (Rabiner)

- slower than Parks-McClennan (big problems),
- allows for some extra constraints during design.

Assumptions

System characteristics

We will now only consider FIR filters. Those systems are:

- linear,
- time-invariant,
- causal.

Let us consider FIR filter's frequency response:

$$H(e^{j\omega}) = \sum_{k=0}^{n-1} h_k e^{-jk\omega}.$$

Let us consider FIR filter's frequency response:

$$H(e^{j\omega}) = \sum_{k=0}^{n-1} h_k e^{-jk\omega}.$$

Denoting

$$c_k(\omega) = \cos(k\omega), \quad s_k(\omega) = \sin(k\omega)$$

we reformulate as

$$H(e^{j\omega}) = \sum_{k=0}^{n-1} h_k (c_k(\omega) - j s_k(\omega)).$$

Let us name $H^{\mathrm{aim}}(\omega)$ — desired frequency response.

Let us name $H^{\mathrm{aim}}(\omega)$ — desired frequency response.

$$\text{minimize} \quad \max_{\omega \in \Omega} |H(e^{j\omega}) - H^{\mathrm{aim}}(\omega)|,$$

where Ω — frequency interval,
and $(h_0, h_1, \ldots, h_{n-1})^T$ — filter coefficients (decision variable).

Let us name $H^{\text{aim}}(\omega)$ — desired frequency response.

$$\text{minimize} \quad \max_{\omega \in \Omega} |H(e^{j\omega}) - H^{\text{aim}}(\omega)|,$$

where Ω — frequency interval,
and $(h_0, h_1, \ldots, h_{n-1})^T$ — filter coefficients (decision variable).

This problem can be easily reformulated as

$$\begin{aligned} \text{minimize} \quad & t \\ \text{st} \quad & |H(e^{j\omega}) - H^{\text{aim}}(\omega)|^2 \leqslant t, \omega \in \Omega^{\cdot} \end{aligned}$$

Let us name $H^{\text{aim}}(\omega)$ — desired frequency response.

$$\text{minimize} \quad \max_{\omega \in \Omega} |H(e^{j\omega}) - H^{\text{aim}}(\omega)|,$$

where Ω — frequency interval,
and $(h_0, h_1, \ldots, h_{n-1})^T$ — filter coefficients (decision variable).

This problem can be easily reformulated as

$$\begin{array}{rl} \text{minimize} & t \\ \text{st} & |H(e^{j\omega}) - H^{\text{aim}}(\omega)|^2 \leqslant t, \omega \in \Omega \end{array}.$$

Both above problems have infinite number of constraints.

We can also add weight $w : [0, 2\pi] \to [0, \infty)$ (since different importance for different ω).

- It is usually piecewise constant.

We can also add weight $w : [0, 2\pi] \to [0, \infty)$ (since different importance for different ω).

- It is usually piecewise constant.

How to convert an infinite-constraint problem into a finite one?

We can also add weight $w : [0, 2\pi] \to [0, \infty)$ (since different importance for different ω).

- It is usually piecewise constant.

How to convert an infinite-constraint problem into a finite one?

Discretization of ω!

- $H_i^{\text{aim}} = H^{\text{aim}}(\omega_i)$ discretized desired response for ω_i,
- $N_1 - 1$ frequencies ω_i in passband,
- $N - N_1 + 1$ frequencies in stopband.

After discretization the optimization problem becomes

$$\begin{aligned} \text{minimize} \quad & t \\ \text{st} \quad & w^2(\omega_i)|H(e^{j\omega_i}) - H_i^{\text{aim}}|^2 \leqslant t, i = 1, 2, \dots, N. \end{aligned}$$

After discretization the optimization problem becomes

$$\begin{array}{ll} \text{minimize} & t \\ \text{st} & w^2(\omega_i)|H(e^{j\omega_i}) - H_i^{\text{aim}}|^2 \leqslant t, i = 1, 2, \ldots, N. \end{array}$$

And denote as H_i^{real} — real part of H_i^{aim} and H_i^{im} — imaginary part.

Let us recall that

$$H(e^{j\omega}) = \sum_{k=0}^{n-1} h_k(c_k(\omega) - js_k(\omega)),$$

so

$$w^2(\omega_i)|H(e^{j\omega_i}) - H_i^{\text{aim}}|^2 =$$

$$= w^2(\omega_i)\left|\sum_{k=0}^{n-1} h_k(c_k(\omega_i) - js_k(\omega_i)) - H_i^{\text{real}} - jH_i^{\text{im}}\right|^2 =$$

$$= w^2(\omega_i)\left(\left[\sum_{k=0}^{n-1} h_k c_k(\omega_i) - H_i^{\text{real}}\right]^2 + \left[\sum_{k=0}^{n-1} h_k s_k(\omega_i) + H_i^{\text{im}}\right]^2\right).$$

And let us denote:

$$\alpha_1(\omega_i) := w(\omega_i)\Big[\sum_{k=0}^{n-1} h_k c_k(\omega_i) - H_i^{\text{real}}\Big],$$

$$\alpha_2(\omega_i) := w(\omega_i)\Big[\sum_{k=0}^{n-1} h_k s_k(\omega_i) + H_i^{\text{im}}\Big].$$

And let us denote:

$$\alpha_1(\omega_i) := w(\omega_i)\Big[\sum_{k=0}^{n-1} h_k c_k(\omega_i) - H_i^{\text{real}}\Big],$$

$$\alpha_2(\omega_i) := w(\omega_i)\Big[\sum_{k=0}^{n-1} h_k s_k(\omega_i) + H_i^{\text{im}}\Big].$$

We can now reformulate the constraints as $\alpha_1^2(\omega_i) + \alpha_2^2(\omega_i) \leqslant t$. Observe that α_1 and α_2 depend on the vector of decision variables $(h_0, h_1, \dots, h_{n-1})^T$.

And let us denote:

$$\alpha_1(\omega_i) := w(\omega_i)\Big[\sum_{k=0}^{n-1} h_k c_k(\omega_i) - H_i^{\mathrm{real}}\Big],$$

$$\alpha_2(\omega_i) := w(\omega_i)\Big[\sum_{k=0}^{n-1} h_k s_k(\omega_i) + H_i^{\mathrm{im}}\Big].$$

We can now reformulate the constraints as $\alpha_1^2(\omega_i) + \alpha_2^2(\omega_i) \leqslant t$.
Observe that α_1 and α_2 depend on the vector of decision variables $(h_0, h_1, \ldots, h_{n-1})^T$.
Let us proceed to the semidefinite formulation.

It has been shown in [3] that the constraints $\alpha_1^2(\omega_i) + \alpha_2^2(\omega_i) \leqslant t$ can be reformulated in terms of SDP.
Let us define (for each ω_i) a matrix

$$D_i := \begin{bmatrix} t & \alpha_1(\omega_i) & \alpha_2(\omega_i) \\ \alpha_1(\omega_i) & 1 & 0 \\ \alpha_2(\omega_i) & 0 & 1 \end{bmatrix}.$$

It has been shown in [3] that the constraints $\alpha_1^2(\omega_i) + \alpha_2^2(\omega_i) \leqslant t$ can be reformulated in terms of SDP.
Let us define (for each ω_i) a matrix

$$D_i := \begin{bmatrix} t & \alpha_1(\omega_i) & \alpha_2(\omega_i) \\ \alpha_1(\omega_i) & 1 & 0 \\ \alpha_2(\omega_i) & 0 & 1 \end{bmatrix}.$$

It is positive semidefinite if and only if

$$\begin{aligned} t - \alpha_1^2(\omega_i) - \alpha_2^2(\omega_i) &\geqslant 0, \\ t - \alpha_1^2(\omega_i) &\geqslant 0, \\ t - \alpha_2^2(\omega_i) &\geqslant 0, \\ t &\geqslant 0. \end{aligned}$$

It has been shown in [3] that the constraints $\alpha_1^2(\omega_i) + \alpha_2^2(\omega_i) \leqslant t$ can be reformulated in terms of SDP.
Let us define (for each ω_i) a matrix

$$D_i := \begin{bmatrix} t & \alpha_1(\omega_i) & \alpha_2(\omega_i) \\ \alpha_1(\omega_i) & 1 & 0 \\ \alpha_2(\omega_i) & 0 & 1 \end{bmatrix}.$$

It is positive semidefinite if and only if

$$\begin{aligned} t - \alpha_1^2(\omega_i) - \alpha_2^2(\omega_i) &\geqslant 0, \\ t - \alpha_1^2(\omega_i) &\geqslant 0, \\ t - \alpha_2^2(\omega_i) &\geqslant 0, \\ t &\geqslant 0. \end{aligned}$$

The first inequality implies the other three.

D_i is positive semidefinite if and only if the first inequality holds.
It means that the each constraint $\alpha_1^2(\omega_i) + \alpha_2^2(\omega_i) \leqslant t$ is equivalent to $D_i \succcurlyeq 0$.

D_i is positive semidefinite if and only if the first inequality holds.
It means that the each constraint $\alpha_1^2(\omega_i) + \alpha_2^2(\omega_i) \leqslant t$ is equivalent to $D_i \succcurlyeq 0$.
The decision variable is a vector $x = \big(t, h_0, h_1, \ldots, h_{n-1}\big)^T \in \mathbb{R}^{n+1}$.
Remembering that all D_i depend on x, we can finally write that

$$F(x) := \begin{bmatrix} D_1 & & & 0 \\ & D_2 & & \\ & & \ddots & \\ 0 & & & D_N \end{bmatrix}.$$

It is obvious that $F(x)$ is positive semidefinite if and only if each block D_i is positive semidefinite. It is also easy to see that $F(x)$ is an affine function of the decision variable x.

It is obvious that $F(x)$ is positive semidefinite if and only if each block D_i is positive semidefinite. It is also easy to see that $F(x)$ is an affine function of the decision variable x.

Final SDP problem

Now we can reformulate the optimization problem as

$$\begin{aligned} \text{minimize} \quad & t \\ \text{st} \quad & F(x) \succcurlyeq 0. \end{aligned}$$

Where $x = \left(t, h_0, h_1, \ldots, h_{n-1}\right)^T$ is a vector of decision variables and the parameters (frequencies ω_i and desired frequency responces H_i^{aim}) are *hidden* in D_i.

It is obvious that $F(x)$ is positive semidefinite if and only if each block D_i is positive semidefinite. It is also easy to see that $F(x)$ is an affine function of the decision variable x.

Final SDP problem

Now we can reformulate the optimization problem as

$$\begin{aligned} \text{minimize} \quad & t \\ \text{st} \quad & F(x) \succcurlyeq 0. \end{aligned}$$

Where $x = \left(t, h_0, h_1, \ldots, h_{n-1}\right)^T$ is a vector of decision variables and the parameters (frequencies ω_i and desired frequency responces H_i^{aim}) are *hidden* in D_i.

It is a semidefinite programming problem.

Numerical experiments

Consider a filter.

Filter's frequency response is given by

$$H^{\text{aim}}(\omega) = \begin{cases} e^{-j42\omega} & \text{for } 0 \leqslant \omega \leqslant 0.45\pi, \\ 0 & \text{for } 0.5\pi \leqslant \omega \leqslant \pi. \end{cases}$$

With an arbitrary filter length of $n = 85$.

Numerical experiments

Consider a filter.

Filter's frequency response is given by

$$H^{\text{aim}}(\omega) = \begin{cases} e^{-j42\omega} & \text{for } 0 \leqslant \omega \leqslant 0.45\pi, \\ 0 & \text{for } 0.5\pi \leqslant \omega \leqslant \pi. \end{cases}$$

With an arbitrary filter length of $n = 85$.

It is a low-pass filter — it passes the signal for frequencies $\omega \leqslant 0.45\pi$ and blocks it for $\omega \geqslant 0.5\pi$, (since $|e^{-j42\omega}| = 1$).

Numerical experiments

Consider a filter.

Filter's frequency response is given by

$$H^{\text{aim}}(\omega) = \begin{cases} e^{-j42\omega} & \text{for } 0 \leqslant \omega \leqslant 0.45\pi, \\ 0 & \text{for } 0.5\pi \leqslant \omega \leqslant \pi. \end{cases}$$

With an arbitrary filter length of $n = 85$.

It is a low-pass filter — it passes the signal for frequencies $\omega \leqslant 0.45\pi$ and blocks it for $\omega \geqslant 0.5\pi$, (since $|e^{-j42\omega}| = 1$).

We will create 300 sample frequencies uniformly spread on the passband and stopband ($\omega_1, \ldots, \omega_{142}$ and $\omega_{143}, \ldots, \omega_{300}$).

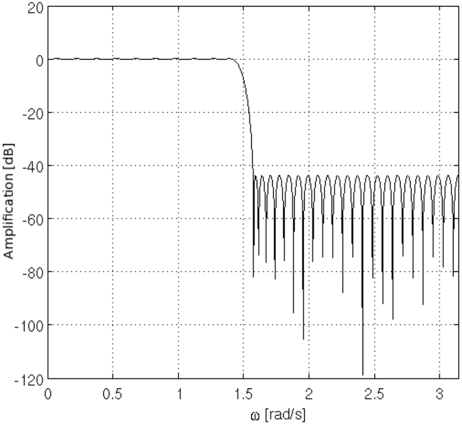


Figure: Absolute value of the transfer function $H(e^{j\omega})$ of the designed low-pass filter

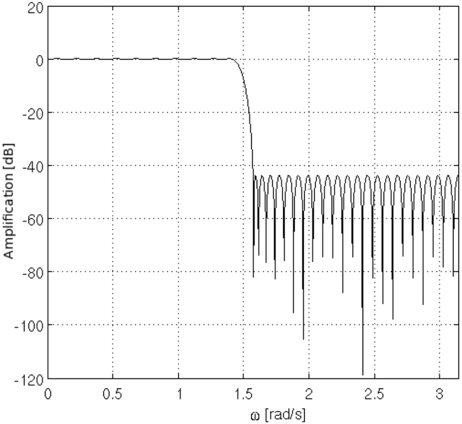


Figure: Absolute value of the transfer function $H(e^{j\omega})$ of the designed low-pass filter

It is easy to see that it is an equiripple low-pass filter.

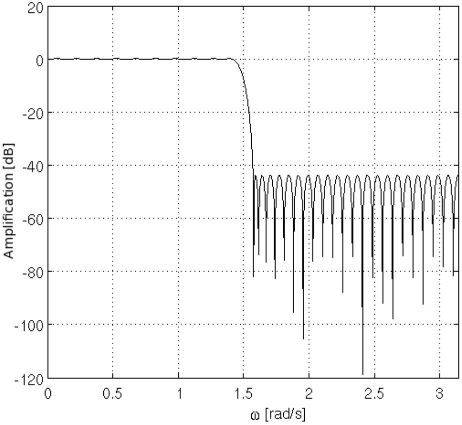


Figure: Absolute value of the transfer function $H(e^{j\omega})$ of the designed low-pass filter

It is easy to see that it is an equiripple low-pass filter. Semidefinite (block-diagonal) matrix was of a size 900×900.

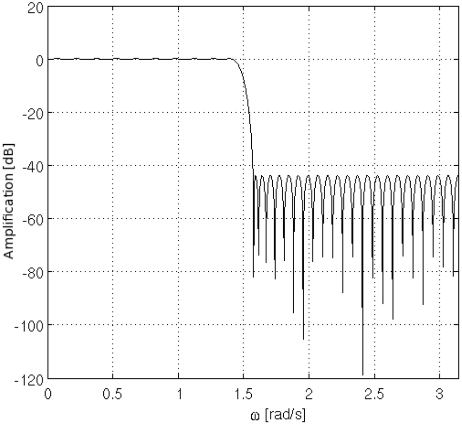


Figure: Absolute value of the transfer function $H(e^{j\omega})$ of the designed low-pass filter

It is easy to see that it is an equiripple low-pass filter.
Semidefinite (block-diagonal) matrix was of a size 900×900.
Those results have been obtained in MATLAB with SDPT3 solver.

Frequency response given by:

$$H^{\text{aim}}(\omega) = \begin{cases} (\frac{\omega}{0.9\pi} + \frac{1}{2})e^{-j60\omega} & \text{for } 0 \leqslant \omega \leqslant 0.45\pi, \\ 0 & \text{for } 0.5\pi \leqslant \omega \leqslant \pi. \end{cases}$$

Frequency response given by:

$$H^{\text{aim}}(\omega) = \begin{cases} (\frac{\omega}{0.9\pi} + \frac{1}{2})e^{-j60\omega} & \text{for } 0 \leqslant \omega \leqslant 0.45\pi, \\ 0 & \text{for } 0.5\pi \leqslant \omega \leqslant \pi. \end{cases}$$

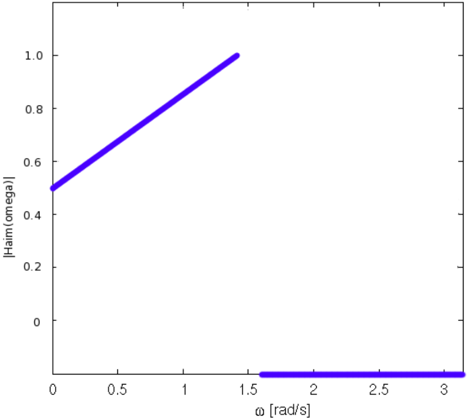


Figure: Absolute value of the desired transfer function

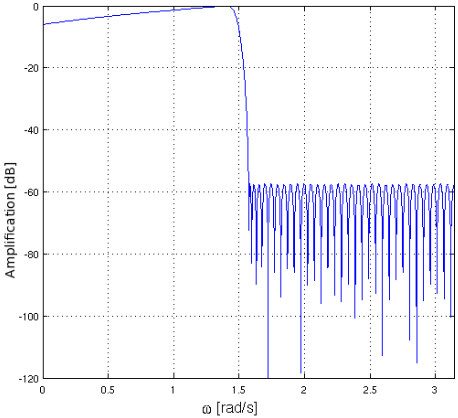


Figure: Absolute value of the transfer function $H(e^{j\omega})$ of the designed low-pass filter

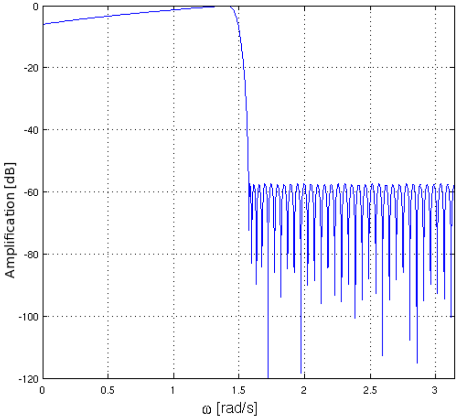


Figure: Absolute value of the transfer function $H(e^{j\omega})$ of the designed low-pass filter

It was caluclated with 300 sample frequencies (as earlier) and a filter length of $n = 120$.

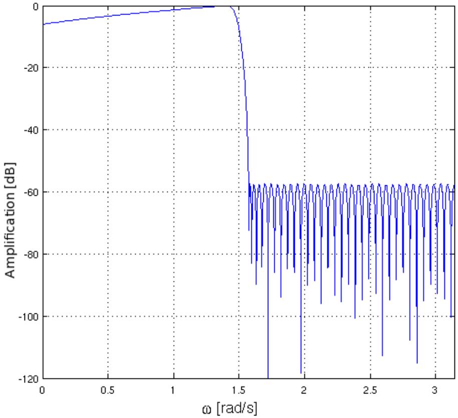


Figure: Absolute value of the transfer function $H(e^{j\omega})$ of the designed low-pass filter

It was caluclated with 300 sample frequencies (as earlier) and a filter length of $n = 120$.
It is actually a **6dB loss**, which is exactly what it should be for an absolute value of transfer function $\frac{1}{2}$ at $\omega = 0$.

Conclusions

- Digital filters are important in everyday engineering.

Conclusions

- Digital filters are important in everyday engineering.
- Regardless of over 50 years of digital filter design there is still work to be done.

Conclusions

- Digital filters are important in everyday engineering.
- Regardless of over 50 years of digital filter design there is still work to be done.
- Novel methods based on SDP allow to efficiently design filters with **desired characteristcs**,

Conclusions

- Digital filters are important in everyday engineering.
- Regardless of over 50 years of digital filter design there is still work to be done.
- Novel methods based on SDP allow to efficiently design filters with **desired characteristcs**,
- (and SOCP as well).

Conclusions

- Digital filters are important in everyday engineering.
- Regardless of over 50 years of digital filter design there is still work to be done.
- Novel methods based on SDP allow to efficiently design filters with **desired characteristcs**,
- (and SOCP as well).

Future work

Conclusions

- Digital filters are important in everyday engineering.
- Regardless of over 50 years of digital filter design there is still work to be done.
- Novel methods based on SDP allow to efficiently design filters with **desired characteristcs**,
- (and SOCP as well).

Future work

- IIR filter desgin with SOCP programming.

Conclusions

- Digital filters are important in everyday engineering.
- Regardless of over 50 years of digital filter design there is still work to be done.
- Novel methods based on SDP allow to efficiently design filters with **desired characteristcs**,
- (and SOCP as well).

Future work

- IIR filter desgin with SOCP programming.
- Multidimensional FIR filters with SDP methods.

References

- W.-S. Lu i in.: *A Weighted Least-Squares Method for the Design of Stable 1-D and 2-D IIR Digital Filters*, IEEE Trans. Signal Proc., 46(1):1–10, January 1998.
- W.-S. Lu, T. Hinamoto: *Optimal Design of IIR Digital Filters with Robust Stability Using Conic Quadratic Programming*, IEEE Trans. Signal Proc., 51(6):1581–1592, June 2003.
- F. Wysocka-Schillak: *Projektowanie filtrów cyfrowych jedno- i dwuwymiarowych z zastosowaniem aproksymacji równomiernie falistej*, D.Sc. thesis at AT-R Bydgoszcz, Poland, 2005.

References

- S.-P. Wu, S. Boyd, L. Vandenberghe: *FIR Filter Design via Spectral Factorization and Convex Optimization* w: B. Datta, *Applied Computational Control, Signal and Communications*, 1997.
- D. Gorinevsky, S. Boyd: *Optimization-Based Design and Implementation of Multidimensional Zero-Phase IIR Filters*, IEEE Trans. Circuts and Systems, 53(2):372–383, February 2006.
- A. Antoniou, W.-S. Lu: *Practical Optimization: Algorithms and Engineering Applications*, Springer, New York 2007

Thank you for your attention!