USING MATLAB FUNCTIONS FOR DESIGNING CHEBYSEV FILTERS

P. Vojcinak, J. Koziorek, R. Hajovsky

VSB – Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Cybernetics and Biomedical Engineering
Ave. 17. listopadu 2172/15, 708 33 Ostrava – Poruba, the Czech Republic

Abstract

This paper deals with using MATLAB function and tools for designing first-order analogue Chebysev filters and IIR Chebysev filters. The first part of this paper is focused on a design of analogue filter via Chebyshev approximation approach, i. e. features and mathematical background of this iso-extremal approximation, approximation of normalized low-pass (NLP, also NDP in Czech) filter, and mathematical formulas for calculating fundamental parameters of NLP, such as constructing a tolerance scheme, Chebyshev approximation order, poles of transfer function, characteristic equation, and group delay. Due to frequency transformation formulas implemented in MATLAB, un-normalized forms of frequency functions for low-pass (LP, also DP in Czech), high-pass (HP, also the same in Czech), band-pass (BP, also PP in Czech), and band-stop (BS, also PZ in Czech) analogue filters are also available. The second part of this paper describes designing the first-order IIR Chebyshev filter via implemented MATLAB functions and Filter Visualization Tool (FVT), whereas conversion from an analogue form into a digital form is done and discussed for bilinear (Tustin) transformation only. Naturally, both design approaches are illustrated in form of some practical examples.

1 Introduction to Chebyshev polynomials

Chebyshev polynomials are some sequence of classic orthogonal polynomials, which are related to famous de Moivre’s formula and which can be defined recursively. We usually distinguish between two kinds of Chebyshev polynomials, thus [1]:

- the first kind $T_n(x)$,
- the second kind $U_n(x)$.

These Chebyshev polynomials are polynomials of nth degree and their sequence composes a polynomial sequence. In practical use, the Chebyshev polynomials are important in approximation theory, whereas the roots of the first kind of Chebyshev polynomials (so-called Chebyshev nodes) are used in polynomial interpolation [1]. This paper is focused on describing mathematical background of the first kind only.

The Chebyshev polynomials are also the solutions of these ordinary differential equations (ODE), thus [1]:

for the first kind

$$\left(1-x^2\right)\cdot\frac{d^2}{dx^2}\{f(x)\}-x\cdot\frac{d}{dx}\{f(x)\}+n^2\cdot f(x)=0 \qquad (1)$$

where $f(x)\equiv y=T_n(x)$ solution of this Chebyshev ODE.

for the second kind

$$\left(1-x^2\right)\cdot\frac{d^2}{dx^2}\{f(x)\}-3\cdot x\cdot\frac{d}{dx}\{f(x)\}+n\cdot(n+2)\cdot f(x)=0 \qquad (2)$$

where $f(x)\equiv y=U_n(x)$ solution of this Chebyshev ODE.

These second-order ODEs are special cases of Sturm-Liouville differential equation [2], because it is possible to transform Eq. (1) and Eq. (2) into these forms:

for the first kind

$$2\cdot\frac{d}{dx}\left\{\left(1-x^2\right)\cdot\frac{d}{dx}\,f(x)\right\}+2\cdot n^2\cdot f(x)=0 \tag{3}$$

where $f(x)\equiv y=T_n(x)$ solution of this Chebyshev ODE.

for the second kind

$$\frac{2}{3}\cdot\frac{d}{dx}\left\{\left(1-x^2\right)\cdot\frac{d}{dx}\,f(x)\right\}+\frac{2}{3}\cdot n\cdot(n+2)\cdot f(x)=0 \tag{4}$$

where $f(x)\equiv y=U_n(x)$ solution of this Chebyshev ODE.

2 Mathematical background

This part is centred on detail description of mathematical background, related to the Chebyshev polynomials – e g. definitions, orthogonality, representations, generating functions, recurrence relation, roots, and extrema.

2.1 Orthogonality

Generally, both kinds of Chebyshev polynomials are orthogonal; in case of the first kind, we assume:

- interval $x\in(-1,+1)$,
- weight function $w(x)=\left(1-x^2\right)^{-\frac{1}{2}}$.

Based on Sturm-Liouville theory and form of an orthonormal basis in the Hilbert space, we assume this formula [1] [4]:

$$\int_a^b f_n(x)\cdot f_m(x)\cdot w(x)\cdot dx \Rightarrow \int_{-1}^{+1} T_n(x)\cdot T_m(x)\cdot\frac{dx}{\sqrt{1-x^2}}=\begin{cases} =0 & m\neq n\\ =\pi/2=\left\|T_n(x)\right\|^2 & m=n, n\neq 0\\ =\pi=\left\|T_0(x)\right\|^2 & m=n=0\end{cases} \tag{5}$$

2.2 Representation

2.2.1 Polynomial representation

The first kind of Chebyshev polynomials can be defined and represented in these several forms. Because the Chebyshev polynomials are special cases of the ultraspherical (or Gegenbauer) polynomials, which themselves are special cases of Jacobi (or hypergeometric) polynomials [1] [4], because Gegenbauer differential equation has this form:

$$\left(1-x^2\right)\cdot\frac{d^2}{dx^2}\{f(x)\}-(2\cdot\alpha+1)\cdot x\cdot\frac{d}{dx}\{f(x)\}+n\cdot(n+2\cdot\alpha)\cdot f(x)=0 \tag{6}$$

Especially, if $\alpha=0$, then we get the form of Eq. (1); in case of $\alpha=1$, we get the form of Eq. (2). Following formulas are focused on the case of $\alpha=0$. The solution of Eq. (6) is given by Gegenbauer polynomials $C_n^{(\alpha)}(x)$[2], and Jacobi polynomials $P_n^{(a,b)}(x)$[3], when $a=b=-0{,}5$.

$$T_n(x)=\frac{1}{\begin{pmatrix} n-\frac{1}{2}\\ n\end{pmatrix}}\cdot P_n^{\left(-\frac{1}{2},-\frac{1}{2}\right)}(x)=\frac{(2\cdot n)!!}{(2\cdot n-1)!!}\cdot P_n^{\left(-\frac{1}{2},-\frac{1}{2}\right)}(x)=\frac{n}{2}\cdot C_n^0(x) \tag{7}$$

where

$$P_n^{\left(-\frac{1}{2},-\frac{1}{2}\right)}(x)=\frac{(-1)^n}{2^n\cdot n!}\cdot(1-x)^{\frac{1}{2}}\cdot(1+x)^{\frac{1}{2}}\cdot\frac{d^n}{dx^n}\left\{(1-x)^{n-\frac{1}{2}}\cdot(1+x)^{n-\frac{1}{2}}\right\} \tag{8a}$$

and [2]

$$C_n^0(x)=\sum_{m=0}^{[n/2]}\frac{(-1)^m\cdot(n-m-1)!}{(n-2\cdot m)!\cdot(m!)}\cdot(2\cdot x)^{n-2\cdot m}=\sum_{m=0}^{[n/2]}\frac{(-1)^m\cdot\Gamma(n-m)}{(n-2\cdot m)!\cdot(m!)}\cdot(2\cdot x)^{n-2\cdot m} \tag{8b}$$

Generalized form of Eq. (8b) is given by Eq. (9a), leading to Rodrigues formula, thus [2]:

$$C_n^{(\alpha)}(x)=\sum_{m=0}^{[n/2]}\frac{(-1)^m\cdot\Gamma(n-m+\alpha)}{\Gamma(\alpha)\cdot(n-2\cdot m)!\cdot(m!)}\cdot(2\cdot x)^{n-2\cdot m}=\frac{\Gamma\left(\alpha+\frac{1}{2}\right)}{\Gamma(2\cdot\alpha)}\cdot\frac{\Gamma(n+2\cdot\alpha)}{\Gamma\left(n+\alpha+\frac{1}{2}\right)}\cdot P_n^{\left(\alpha-\frac{1}{2},\alpha-\frac{1}{2}\right)}(x) \tag{9a}$$

and (Rodrigues formula) [2]

$$C_n^{(\alpha)}(x)=\frac{(-2)^n}{n!}\cdot\frac{\Gamma(n+\alpha)\cdot\Gamma(n+2\cdot\alpha)}{\Gamma(\alpha)\cdot\Gamma(2\cdot n+2\cdot\alpha)}\cdot\left(1-x^2\right)^{-\alpha+\frac{1}{2}}\cdot\frac{d}{dx^n}\left\{\left(1-x^2\right)^{n+\alpha-\frac{1}{2}}\right\} \tag{9b}$$

For the Chebyshev polynomials of the first kind, the Rodrigues formula has this form [4]:

$$T_n(x)=\frac{(-1)^n\cdot 2^n}{2\cdot(2\cdot n-1)!}\cdot\left(1-x^2\right)^{\frac{1}{2}}\cdot\frac{d^n}{dx^n}\left\{\left(1-x^2\right)^{n-\frac{1}{2}}\right\}=\frac{(-1)^n\cdot 2^n}{2\cdot(2\cdot n-1)!}\cdot G_n^{\left(-\frac{1}{2},-\frac{1}{2}\right)}(x) \tag{10a}$$

and

$$T_n(x)=\frac{(-2)^n}{2\cdot n!}\cdot\frac{\Gamma(n)\cdot\Gamma(n+1)}{\Gamma(2\cdot n)}\cdot G_n^{\left(-\frac{1}{2},-\frac{1}{2}\right)}(x)=\frac{n}{2}\cdot\frac{(-2)^n}{n!}\cdot\frac{\Gamma^2(n)}{\Gamma(2\cdot n)}\cdot G_n^{\left(-\frac{1}{2},-\frac{1}{2}\right)}(x)=\frac{n}{2}\cdot C_n^0(x) \tag{10b}$$

The result formulas in Eq. (7) and Eq. (10b) are the same.

For each non-negative integer n, both kinds of Chebyshev polynomials are polynomials of nth degree. They are even or odd function of x, whereas n is even or odd. In case of the first kind, we assume [1]:

$$T_n\left(1-2\cdot x^2\right)=(-1)^n\cdot T_{(2\cdot n)}(x) \tag{11}$$

Naturally, other polynomial sequences like Lucas polynomials, Dickson polynomials, or Fibonacci polynomials are also related to both kinds of Chebyshev polynomials. [1]

2.2.2 Integral representation

In case of an integral representation of the Chebyshev polynomials of the first kind, we assume this formula [1]:

$$T_n(x)=\frac{2^n\cdot(n!)^2\cdot\sqrt{x^2-1}}{2\cdot\pi\cdot i\cdot(2\cdot n)!}\cdot\int_C\frac{\left(z^2-1\right)^{n-\frac{1}{2}}}{(z-x)^{n+1}}\cdot dz \tag{12}$$

where i — imaginary unit,
C — arbitrary Jordan curve in the form of some integration area,
$z=x$ — inner point of the integration area,
$z=\pm 1$ — outer points of the integration area.

2.2.3 Trigonometric representation

In case of trigonometric representation of the Chebyshev polynomials of the first kind, let us consider these formulas, based on Euler's formula and de Moivre's formula:

for cosine function

$$x = \cos(\Phi) \Rightarrow T_n[x = \cos(\Phi)] = \cos(n \cdot \Phi) = \frac{1}{2} \cdot \{\exp(i \cdot n \cdot \Phi) + \exp(-i \cdot n \cdot \Phi)\} \qquad (13a)$$

then [1] [5]

$$T_n[\cos(\Phi)] = \cos(n \cdot \Phi) = \cos[n \cdot \arccos(x)] \quad |x| \le 1 \qquad (13b)$$

for hyperbolic cosine function

$$x = \cosh(\Phi) \Rightarrow T_n[x = \cosh(\Phi)] = \cosh(n \cdot \Phi) = \frac{1}{2} \cdot \{\exp(n \cdot \Phi) + \exp(-n \cdot \Phi)\} \qquad (14a)$$

then [1] [5]

$$T_n[\cosh(\Phi)] = \cosh(n \cdot \Phi) = \cosh[n \cdot \arg\cosh(x)] \quad |x| \ge 1 \qquad (14b)$$

Eq. (13b) and Eq. (14b) show, different approaches to defining the Chebyshev polynomials lead to different explicit formulas – it is very useful for designing the Chebyshev filter not only in pass band via Eq. (13b), but also in stop band via Eq. (14b).

The Chebyshev polynomials are also the solutions of Pell equation, which is any Diophantine equation having this form [6]:

$$x^2 - n \cdot y^2 = 1 \qquad (15a)$$

If we consider features of Euler's formula, de Moivre's formula, and Eq. (13a), then we can write Eq. (15a) in this form, thus:

$$T_n^2(x) - (x^2 - 1) \cdot U_{n-1}^2(x) = T_n^2(x) - (x^2 - 1) \cdot \frac{1}{n^2} \cdot \left[\frac{d}{dx} T_n(x)\right] = 1 \qquad (15b)$$

If we consider $T_n[x = \cos(\Phi)]$ and $U_{n-1}[x = \cos(\Phi)] = \sin(n \cdot \Phi)/\sin(\Phi)$, then we get Euler's formula:

$$\cos^2(n \cdot \Phi) - (-1) \cdot \sin^2(\Phi) \cdot \frac{\sin^2(n \cdot \Phi)}{\sin^2(\Phi)} = \cos^2(n \cdot \Phi) + \sin^2(n \cdot \Phi) = 1 \qquad (15c)$$

If we also consider Eq. (15c), $A^2(x) = T_n^2(x)$ and $B^2(x) = (x^2 - 1) \cdot U_{n-1}^2(x)$, then we get a fundamental solution of Eq. (15b), based on de Moivre's formula, because:

$$A^2(x) - B^2(x) = [A(x) + B(x)] \cdot [A(x) - B(x)] = 1 \qquad (15d)$$

and

$$\varphi_{1,2}[x = \cos(\Phi)] = A[\cos(\Phi)] \pm B[\cos(\Phi)] = \cos(n \cdot \Phi) \pm i \cdot \sin(n \cdot \Phi) = [\cos(\Phi) \pm i \cdot \sin(\Phi)]^n \qquad (15e)$$

and (fundamental solution)

$$\varphi_{1,2}[x = \cos(\Phi)] = \left[\cos(\Phi) \pm \sqrt{(-1) \cdot \sin^2(\Phi)}\right]^n \Rightarrow \varphi_{1,2}(x) = \left[x \pm \sqrt{x^2 - 1}\right]^n \qquad (15f)$$

then

$$T_n(x) = \frac{\varphi_1(x) + \varphi_2(x)}{2} = \frac{\left[x + \sqrt{x^2 - 1}\right]^n + \left[x - \sqrt{x^2 - 1}\right]^n}{2} \qquad (15g)$$

Eq. (15g) is directly related to Eq. (7) and Eq. (8b).

2.3 Generating functions

Based on the form of Eq. (15f), it is possible to define so-called exponential generating function for $T_n(x)$ in this form [1]:

$$EG[T_n(x), t] = \sum_{n=0}^{\infty} T_n(x) \cdot \frac{t^n}{n!} = \frac{e^{\sqrt[n]{\varphi_1(x)} \cdot t} + e^{\sqrt[n]{\varphi_2(x)} \cdot t}}{2} = \frac{\exp\left[\left(x + \sqrt{x^2 - 1}\right) \cdot t\right] + \exp\left[\left(x - \sqrt{x^2 - 1}\right) \cdot t\right]}{2} \qquad (16a)$$

Generating functions for $T_n(x)$ have these forms, thus [1] [4]:

$$G_1[T_n(x),t] = \sum_{n=0}^{\infty} T_n(x)\cdot t^n = \sum_{n=0}^{\infty} U_n(x)\cdot t^n - \frac{x\cdot t}{1-2\cdot x\cdot t+t^2} = \frac{1-x\cdot t}{1-2\cdot x\cdot t+t^2} \tag{17a}$$

and

$$G_2[T_n(x),t] = T_0(x) + 2\cdot\sum_{n=1}^{\infty} T_n(x)\cdot t^n = \sum_{n=0}^{\infty} U_n(x)\cdot t^n - \frac{t^2}{1-2\cdot x\cdot t+t^2} = \frac{1-t^2}{1-2\cdot x\cdot t+t^2} \tag{17b}$$

for $|x| \le 1$ and $|t| < 1$

2.4 Recurrence relation

Both kinds of Chebyshev polynomials are defined by the same recurrence relation [4]. Let us consider:

$$Q_n(x) \equiv \{T_n(x), U_n(x)\} \tag{18a}$$

then

$$Q_{n+1}(x) = 2\cdot x\cdot Q_n(x) - Q_{n-1}(x) \tag{18b}$$

where
$T_{n=0}(x) = \cos(0\cdot\Phi) = 1$ zero-order Ch. polynomial,
$T_{n=1}(x) \Rightarrow \cos(1\cdot\Phi) = \cos(\Phi) \Rightarrow x$ first-order Ch. polynomial,
$T_{n=2}(x) \Rightarrow \cos(2\cdot\Phi) \Rightarrow 2\cdot x^2 - 1$ second-order Ch. polynomial,
$T_{n=3}(x) \Rightarrow \cos(3\cdot\Phi) \Rightarrow 4\cdot x^3 - 3\cdot x$ third-order Ch. polynomial,
$T_{n=4}(x) \Rightarrow \cos(4\cdot\Phi) \Rightarrow 8\cdot x^4 - 8\cdot x^2 + 1$ fourth-order Ch. polynomial.

Eq. (18b) is also very useful, because it is closely related to characteristic equation and characteristic function of designed Chebyshev filter.

2.5 Roots and extrema

The Chebyshev polynomial of the nth order has n different roots (Chebyshev nodes) in the closed interval $\langle -1,+1\rangle$. These roots are used as polynomial interpolation nodes [1], whereas approach is based on solving this goniometrical equation:

$$\cos(x) = 0 \Rightarrow x = \frac{\pi}{2} + k\cdot\pi = \frac{\pi + 2\cdot k\cdot\pi}{2} = \frac{\pi}{2}\cdot(1+2\cdot k) \quad k \in Z \tag{19a}$$

Generally, zeros of $T_n(x)$ occur, when:

$$x_{0k} = \cos\left(\frac{\pi}{2}\cdot\frac{2\cdot k-1}{n}\right) \quad k = 1,2,\ldots,n \tag{19b}$$

where for even n $k = 1,2,\ldots,n/2$,
for odd n $k = 1,2,\ldots,(n-1)/2$.

Generally, extrema of $T_n(x)$ occur, when:

$$x_{Ek} = \cos\left(\frac{k\cdot\pi}{n}\right) \quad k = 0,1,\ldots,n \tag{19c}$$

Eq. (19b) is also useful for calculating the zeros of the characteristic equation of design Chebyshev filter.

3 Designing the first-order analogue Chebyshev filter

Designing the first-order analogue Chebyshev filter is based on the Chebyshev approximation, which uses so-called first Chebyshev approximation method. In this case, we find some polynomial solution in open interval $\Omega = (-1,+1)$. This solution has to approximate zero as best and with regular divergence [5]. We assume some approximate differential equation; in this case it is the Pell equation, mentioned in Eq. (15b) and converted into this form, thus:

$$\left(1-x^2\right)\cdot\left[\frac{d}{dx}T_n(x)\right]=n^2\cdot\left[1-T_n^2(x)\right] \tag{20}$$

where n an order of the first kind Chebyshev polynomials, or an approx. order.

3.1 Approximation of normalized low-pass filter

This approximation is based on the first-order Chebyshev polynomials and focused on the normalized low-pass (NLP) filter only. Due to frequency normalization, it is possible to convert ideal LP's requirements into a NLP prototype. Naturally, it is also possible to transform requirements for other types of analogue filters (LP, HP, BP, and BS) into NLP filter, namely via other frequency formulas and/or using implemented MATLAB functions. The standard results of this approximation approach are as follows, thus:

- transfer function,
- characteristic equation,
- group delay.

There are defined and shown some ideal requirements for frequency-magnitude characteristic of an ideal low-pass (LP) filter, see the right part of Fig. (1). Radial frequency ω_p is a cut-off radial frequency of filter pass-band, where magnitude values at positive radial frequencies are given by this formula, thus [5]:

$$\left|\hat{H}_{LP}(j\cdot\omega)\right|\equiv\left|\hat{H}_{DP}(j\cdot\omega)\right|=\begin{cases}=1 & \omega\in\left\langle 0,\omega_p\right\rangle\\ =0 & \omega\in\left(\omega_s,+\infty\right)\end{cases} \tag{21a}$$

Analogously to ideal low-pass filter, there are defined and shown some ideal requirements for normalized frequency-magnitude characteristic of an ideal normalized low-pass (NLP) filter, see the left part of Fig. (1). Normalized radial frequency Ω_p , naturally equalled to 1, is a normalized cut-off radial frequency of normalized filter pass-band, where magnitude values at positive normalized radial frequencies are given by this formula, thus [5]:

$$\left|\hat{H}_{NLP}(j\cdot\omega)\right|\equiv\left|\hat{H}_{NDP}(j\cdot\omega)\right|=\begin{cases}=1 & \Omega\in\left\langle 0,\Omega_p\right\rangle=\left\langle 0,1\right\rangle\\ =0 & \Omega\in\left(\Omega_s,+\infty\right)\end{cases} \tag{21b}$$

In case of NLP filter, we consider some scaling in the form of frequency normalization, when we create so-called normalized complex area:

$$p=\frac{s}{\omega_p}=\frac{\sigma+j\cdot\omega}{\omega_p}=\frac{\sigma}{\omega_p}+j\cdot\frac{\omega}{\omega_p}=\Sigma+j\cdot\Omega\Rightarrow\Omega_p=\frac{\omega}{\omega_p} \tag{21c}$$

where s un-normalized Laplace operator,
p normalized Laplace operator,
Σ real axis of normalized complex area,
Ω imaginary axis of normalized complex area.

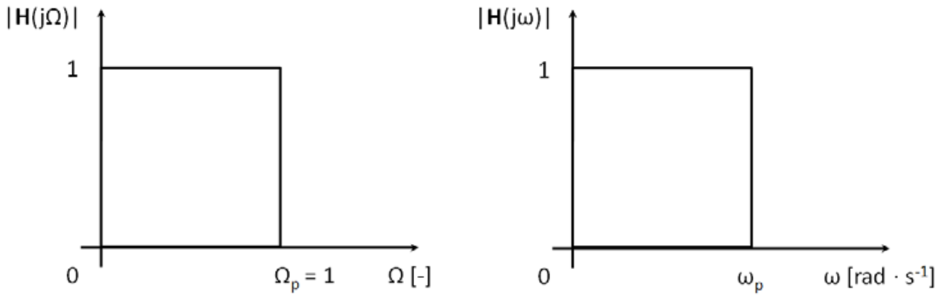

Figure 1: Ideal frequency-magnitude characteristics – normalized low-pass filter (NLP, left) and (un-normalized) low-pass filter (LP, right)

Because frequency-magnitude characteristics of an ideal NLP filter need not meet all the feasibility requirements [5], it is required to consider so-called tolerance scheme, or standard tolerance scheme (see Fig.2).

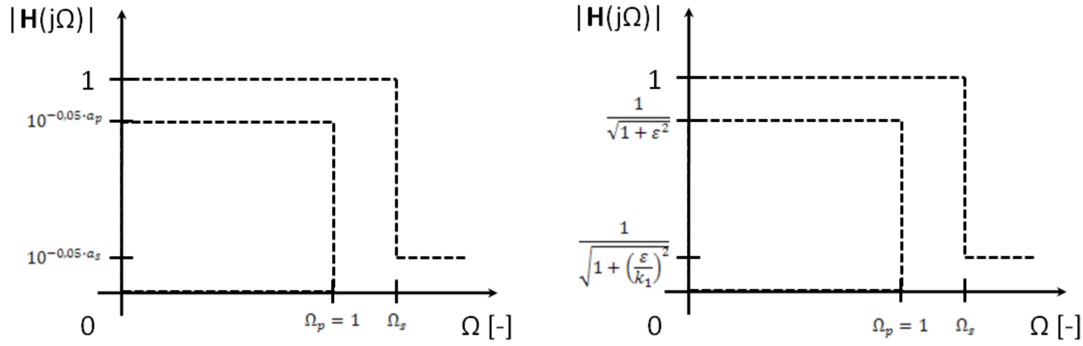

Figure 2: Normalized low-pass filter – tolerance scheme (left; including the primary parameters) and standard tolerance scheme (right; including the secondary parameters)

Some approximating rational function can approximate magnitude (equalled to 1) in band-pass, whereas certain dB-based error value (a_p) could be detected. If we approximate zero-magnitude in transitional band, corresponding characteristic must be continuous. Selectivity of designed filter is given by the transitional band, which is limited by dimensionless frequency of band-stop (Ω_s) and corresponding approximation error value (a_s) in this band. Parameters included in tolerance scheme represent so-called primary parameters related to the requirements for solving the problem of NLP's synthesis. Fortunately, the tolerance scheme's requirements could be given by so-called secondary parameters related to the standard tolerance scheme [5]. There are corresponding formulas between the tolerance scheme and the standard tolerance one, see corresponding magnitude levels shown in Fig. 2.

For band-pass, this identity is available, thus:

$$10^{-0,05\cdot a_p} = \frac{1}{\sqrt{1+\varepsilon^2}} \Rightarrow \begin{cases} a_p = 10\cdot\log\left(1+\varepsilon^2\right) \\ |\varepsilon| = \varepsilon = \sqrt{10^{0,1\cdot a_p}-1} \end{cases} \tag{22a}$$

For band-stop, this identity is available, thus:

$$10^{-0,05\cdot a_s} = \frac{1}{\sqrt{1+\left(\dfrac{\varepsilon}{k_1}\right)^2}} \Rightarrow \begin{cases} a_s = 10\cdot\log\left(1+\dfrac{\varepsilon^2}{k_1^2}\right) \\ |\varepsilon| = \varepsilon = k_1\cdot\sqrt{10^{0,1\cdot a_s}-1} \\ |k_1| = k_1 = \dfrac{\varepsilon}{\sqrt{10^{0,1\cdot a_s}-1}} \end{cases} \tag{22b}$$

For both bands, we consider this ratio formula:

$$\Omega_s = \frac{\Omega_p}{k} \Rightarrow k = \frac{\Omega_p}{\Omega_s} \tag{22c}$$

The Chebyshev approximation is the most important iso-extremal approximation, providing some oscillating characteristics either in band-pass (the first kind), in band-stop (the second kind), or in both bands. The iso-extermal approximations satisfy the requirements at the lowest order of approximating function. [5]

At the Chebyshev approximation of the NLP filter, the characteristic function, defining iso-extremal characteristic of approximation error in band-pass, is given by the Chebyshev polynomials.

3.2 Formulas for the NLP filter approximation

Following formulas will be used in practical example, so they are mentioned here.

3.2.1 Approximation order

The approximation order is one of the most important parameters, because other formulas are dependent upon its value, which is given by this formula [5]:

$$n \geq \frac{\arg\cosh\left(\frac{\Omega_p}{k_1}\right)}{\arg\cosh\left(\frac{\Omega_p}{k}\right)} = \frac{\ln\left[\frac{\Omega_p}{k_1} + \sqrt{\left(\frac{\Omega_p}{k_1}\right)^2 - 1}\right]}{\ln\left[\frac{\Omega_p}{k_1} + \sqrt{\left(\frac{\Omega_p}{k}\right)^2 - 1}\right]} \quad n \in Z^+ \setminus \{0\} \tag{23}$$

Eq. (23) does not guarantee, that n will be really an integer value (even or odd), so it is required to recalculate the values of a_s and k_1, see Eq. (22b), or the values of Ω_s and k, see Eq. (22c).

3.2.2 Poles of transfer function

There are two ways, how to calculate the poles of NLP's transfer function. The first way is based on this equation (generalized form) [5]:

$$\left|\hat{H}(j \cdot \Omega)\right|^2 = \frac{1}{1+\varepsilon^2 \cdot T_n^2(\Omega)} = \frac{1}{1+\left(\frac{X^n + X^{-n}}{2}\right)^2} \Rightarrow 1+\left(\frac{X^n + X^{-n}}{2}\right)^2 = 0 \tag{24a}$$

where

$$X_1 = X^n \Rightarrow X_v^{-1} = -e^{-j\cdot\frac{2\cdot v-1}{2\cdot n}\cdot\pi} \cdot B^{-1} = -\left(\sqrt{1+\frac{1}{\varepsilon^2}} - \frac{1}{\varepsilon}\right)^{\frac{1}{n}} \cdot e^{-j\cdot\frac{2\cdot v-1}{2\cdot n}\cdot\pi} \tag{24b}$$

and

$$X_2 = X^{-n} \Rightarrow X_v^1 = e^{j\cdot\frac{2\cdot v-1}{2\cdot n}\cdot\pi} \cdot B^1 = \left(\sqrt{1+\frac{1}{\varepsilon^2}} + \frac{1}{\varepsilon}\right)^{\frac{1}{n}} \cdot e^{j\cdot\frac{2\cdot v-1}{2\cdot n}\cdot\pi} \tag{24c}$$

Formulas for the transfer function poles are dependent upon the approximation order, thus [5]:

for even approximation order

$$p_{v(even)} = \frac{X_v + X_v^{-1}}{2} = \frac{B - B^{-1}}{2} \cdot \left\{\cos\left(\frac{2\cdot v - 1}{2 \cdot n}\cdot\pi\right) + j\cdot\sin\left(\frac{2\cdot v - 1}{2 \cdot n}\cdot\pi\right)\right\} \tag{24d}$$

for odd approximation order

$$p_{v(odd)} = \frac{X_v + X_v^{-1}}{2} = \frac{B - B^{-1}}{2} \cdot \left\{\cos\left(\frac{v}{n}\cdot\pi\right) + j\cdot\sin\left(\frac{v}{n}\cdot\pi\right)\right\} \tag{24e}$$

In case of the Chebyshev approximation, the poles of function $\left|\hat{H}(j \cdot \Omega)\right|^2$ are always uniformly located on an ellipse [5], having the primary half-axis and the secondary one:

for the primary half-axis

$$a = \frac{B - B^{-1}}{2} \tag{24f}$$

for the secondary half-axis

$$b = \frac{B + B^{-1}}{2} \tag{24g}$$

The second way is based on using the ellipse half-axes, see Eq. (24f) and Eq. (24g):

$$p_\mu = \alpha_\mu + j \cdot \beta_\mu = -a \cdot \sin\left(\frac{2\cdot\mu - 1}{2\cdot n}\cdot\pi\right) + j \cdot b \cdot \cos\left(\frac{2\cdot\mu - 1}{2\cdot n}\cdot\pi\right) \qquad (24h)$$

3.2.3 Transfer function

A form of formulas for calculating the poles of the NLP's transfer function is also dependent on the approximation order [5]:

for even approximation order

$$\hat{H}_{NLP(even)}(p) = \frac{1}{\varepsilon}\cdot\frac{1}{\prod_{\mu=1}^{m=n/2} 2^{2\cdot m-1}\cdot\left(p^2 - 2\cdot\alpha_\mu\cdot p + \alpha_\mu^2 + \beta_\mu^2\right)} \qquad (25a)$$

for odd approximation order

$$\hat{H}_{NLP(odd)}(p) = \frac{1}{\varepsilon\cdot(p+a)}\cdot\frac{1}{\prod_{\mu=1}^{m=(n-1)/2} 2^{2\cdot m}\cdot\left(p^2 - 2\cdot\alpha_\mu\cdot p + \alpha_\mu^2 + \beta_\mu^2\right)} \qquad (25b)$$

3.2.4 Characteristic equation

A form of formulas for calculating the NLP's characteristic equation is also dependent on the approximation order [5]:

for even approximation order

$$\varphi_{(even)}(p) = \prod_{\mu=1}^{m=n/2} 2^{2\cdot m-1}\cdot\left[p^2 + \cos^2\left(\frac{2\cdot\mu-1}{2\cdot n}\cdot\pi\right)\right] = \prod_{\mu=1}^{m=n/2} 2^{2\cdot m-1}\cdot\left[p^2 + \Omega_{0\mu}^2\right] \qquad (26a)$$

for odd approximation order

$$\varphi_{(odd)}(p) = \prod_{\mu=1}^{m=(n-1)/2} 2^{2\cdot m}\cdot p\cdot\left[p^2 + \cos^2\left(\frac{2\cdot\mu-1}{2\cdot n}\cdot\pi\right)\right] = p\cdot\prod_{\mu=1}^{m=(n-1)/2} 2^{2\cdot m}\cdot\left[p^2 + \Omega_{0\mu}^2\right] \qquad (26b)$$

3.2.5 Group delay

Because it is quite difficult to measure phase delay $\tau_\phi(\Omega)$, group delay $\tau_g(\Omega)$ is preferred in practical use. Group delay is defined as a negative first derivative of frequency-phase characteristic $\phi(\Omega)$, thus [5]:

$$\tau_g(\Omega) = -\frac{d}{d\Omega}\phi(\Omega) \qquad (27a)$$

Generally, if group delay is constant (it is desired), then frequency-phase characteristic of an ideal filter is linearly dependent upon frequency; only FIR filters are able to satisfy this idea.

In this case, we consider analogue filter and normalized frequency. Non-constant group delay in frequency domain influences step response in time domain. In case of the Chebyshev approximation, there are two ways, how to approximate this group delay. The first way is based on formula, which satisfies forms of the Chebyshev polynomials, and mentioned parameters, such as approximation order, ellipse-based placement of the transfer function's poles, parameters of the standard tolerance scheme, and form of the characteristic equation, thus [5]:

$$\tau_g(\Omega) = \frac{1}{1+\varepsilon^2\cdot T_n^2(\Omega)}\cdot\sum_{m=0}^{m=n-1}\frac{T_{2\cdot m}(\Omega)}{\varepsilon^2\cdot\sin\left(\frac{2\cdot m+1}{2\cdot n}\cdot\pi\right)}\cdot\sinh\left[(2\cdot n - 2\cdot m - 1)\cdot\operatorname{arccos}\left(\frac{B+B^{-1}}{2}\right)\right] \qquad (27b)$$

The second way is general (i. e. independent upon an approximate type for other analogue filters) and based on the feature of natural logarithm derivative [5]:

$$\tau_g(\Omega) = -\frac{d}{d\Omega}\phi(\Omega) = -\mathrm{Re}\left\{\frac{1}{\hat{H}(j\cdot\Omega)}\cdot\frac{d}{d\Omega}\hat{H}(j\cdot\Omega)\right\} \tag{27c}$$

4 Practical examples

4.1 Analogue normalized low-pass (NLP) filter

Let us consider these values of normalized radial frequencies and the primary parameters, thus:

for the primary parameters

$$a_p = 1[dB] \quad a_s = 20[dB] \tag{28a}$$

for normalized radial frequencies

$$\Omega_p = 1[-] \quad \Omega_s = 2{,}15[-] \tag{28b}$$

Values of other parameters are as follows:

$$\varepsilon = \sqrt{10^{0,1\cdot a_p} - 1} = \sqrt{10^{0,1\cdot 1} - 1} = 0{,}508847139 \cong 0{,}5088 \tag{28c}$$

and

$$k = \frac{\Omega_p}{\Omega_s} = \frac{1}{2{,}15} = 0{,}465116279 \cong 0{,}4651 \tag{28d}$$

and

$$k_1 = \frac{\sqrt{10^{0,1\cdot a_p} - 1}}{\sqrt{10^{0,1\cdot a_5} - 1}} = \frac{\sqrt{10^{0,1\cdot 1} - 1}}{\sqrt{10^{0,1\cdot 20} - 1}} = 0{,}103483852 \cong 0{,}1035 \tag{28e}$$

and (approximation order, we must round n to the nearest higher integer value)

$$n \geq \frac{\arg\cosh\left(\frac{\Omega_p}{k_1}\right)}{\arg\cosh\left(\frac{\Omega_p}{k}\right)} = \frac{\arg\cosh\left(\frac{1}{0{,}1035}\right)}{\arg\cosh\left(\frac{1}{0{,}4651}\right)} \cong 2{,}1141 \Rightarrow n = 3 \tag{28f}$$

Result of Eq. (28f) shows the approximation order is odd. Now, we must consider all the n-dependent formulas for odd approximation order only, and we must recalculate these parameters for new integer value of n, too:

$$k_1 = \frac{1}{\cosh[n\cdot\arg\cosh(\Omega_s)]} = \frac{1}{\cosh[3\cdot\arg\cosh(2{,}15)]} = 0{,}030026874 \cong 0{,}0300 \tag{28g}$$

and

$$a_s = 10\cdot\log\left(1 + \frac{\varepsilon^2}{k_1^2}\right) = 10\cdot\log\left(1 + \frac{0{,}508847139^2}{0{,}030026874^2}\right) = 24{,}59664076 \cong 24{,}5966 \tag{28h}$$

and (number of extrema, values of extrema in pass band)

$$\Omega_{0\mu} = \cos\left(\frac{2\cdot\mu - 1}{2\cdot n}\cdot\pi\right) \Rightarrow \left\| \overset{odd}{\mu = \frac{n-1}{2} = \frac{3-1}{2} = 1} \right\| \Rightarrow \Omega_{01} = \cos\left(\frac{2\cdot 1 - 1}{2\cdot 3}\cdot\pi\right) = \cos\left(\frac{\pi}{6}\right) = \frac{\sqrt{3}}{2} \tag{28ch}$$

Values of the ellipse parameters are as follows:

for the primary half-axis

$$a = \frac{1}{2}\left(\sqrt{1+\frac{1}{\varepsilon^2}}+\frac{1}{\varepsilon}\right)^{\frac{1}{n}} - \frac{1}{2}\left(\sqrt{1+\frac{1}{\varepsilon^2}}+\frac{1}{\varepsilon}\right)^{\frac{1}{n}} = 0{,}494170604 \cong 0{,}4942 \qquad (28i)$$

for the secondary half-axis

$$b = \frac{1}{2}\left(\sqrt{1+\frac{1}{\varepsilon^2}}+\frac{1}{\varepsilon}\right)^{\frac{1}{n}} + \frac{1}{2}\left(\sqrt{1+\frac{1}{\varepsilon^2}}+\frac{1}{\varepsilon}\right)^{\frac{1}{n}} = 1{,}115439189 \cong 1{,}1154 \qquad (28j)$$

Poles of transfer function are as follows:

for the first pole

$$p_{\mu=1} = -a\cdot\sin\left(\frac{2\cdot1-1}{2\cdot3}\cdot\pi\right) + j\cdot\cos\left(\frac{2\cdot1-1}{2\cdot3}\cdot\pi\right) \cong -0{,}2471 + j\cdot0{,}9660 \qquad (28k)$$

for the second pole

$$p_{\mu=2} = -a\cdot\sin\left(\frac{2\cdot2-1}{2\cdot3}\cdot\pi\right) + j\cdot\cos\left(\frac{2\cdot2-1}{2\cdot3}\cdot\pi\right) \cong -0{,}4942 + j\cdot0 = -0{,}4942 \qquad (28l)$$

for the third pole

$$p_{\mu=3} = -a\cdot\sin\left(\frac{2\cdot3-1}{2\cdot3}\cdot\pi\right) + j\cdot\cos\left(\frac{2\cdot3-1}{2\cdot3}\cdot\pi\right) \cong -0{,}2471 - j\cdot0{,}9660 \qquad (28m)$$

In MATLAB, we can create some algorithm, which calculates these poles, based on Eq. (28k), Eq. (28l), and Eq. (28m). The MATLAB code can be written like this:

```
a = 0.5*(Clen1 - Clen2)
b = 0.5*(Clen1 + Clen2)


for mikro = 1:n
PolyVLevePolorovine(mikro) = - a*sin((2*mikro - 1)*pi/(2*n)) +
i*b*cos((2*mikro - 1)*pi/(2*n));
PolyVPravePolorovine(mikro) = a*sin((2*mikro - 1)*pi/(2*n)) +
i*b*cos((2*mikro - 1)*pi/(2*n));
end


PolyVLevePolorovine(1:n)
PolyVPravePolorovine(1:n)
```

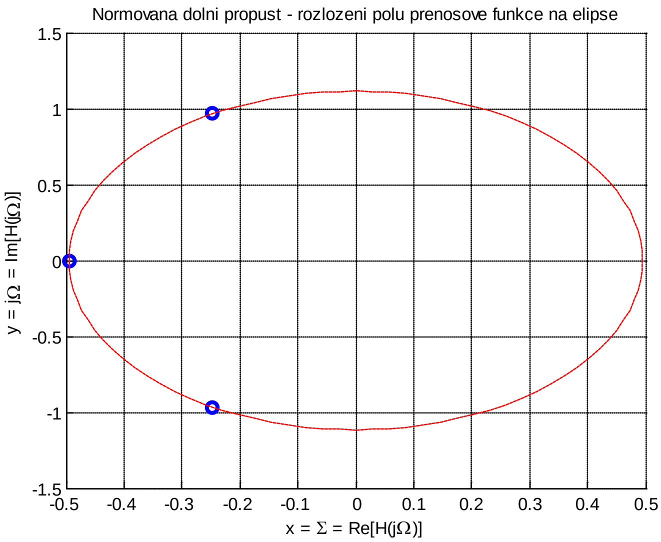

Figure 3: Normalized low-pass filter – elliptic placement of three poles of the transfer function $\hat{H}_{NLP}(p)$; left half-plane of the normalized complex area

Transfer function is as follows, see Eq. (25b):

$$\hat{H}_{NLP(n=3)}(p) = \frac{\hat{H}_0(p)}{\hat{H}_1(p)\cdot\hat{H}_2(p)} \cong \frac{0{,}4913}{p^3 + 0{,}9883\cdot p^2 + 1{,}2384\cdot p + 0{,}4913} \tag{28n}$$

and (normalized frequency-magnitude characteristic)

$$\left|\hat{H}_{NLP(n=3)}(p = j\cdot\Omega)\right| \cong \frac{0{,}4913}{\sqrt{\Omega^6 - 1{,}5001\cdot\Omega^4 + 0{,}5625\cdot\Omega^2 + 0{,}2414}} \tag{28o}$$

and (normalized frequency-phase characteristic)

$$\phi_{NLP(n=3)}(\Omega) \cong \arg\left[\frac{\Omega^3 - 1{,}2384\cdot\Omega}{0{,}4913 - 0{,}9883\cdot\Omega^2}\right] \Rightarrow \left\|\begin{matrix} first \\ quadrant \end{matrix}\right\| = arctg\left[\frac{\Omega^3 - 1{,}2384\cdot\Omega}{0{,}4913 - 0{,}9883\cdot\Omega^2}\right] \tag{28p}$$

In MATLAB code, Eq. (25b) can be written like this:

```
m = (n - 1)/2;
H0 = 1./(epsilonp*2^(2*m));
CitatelPrenosu = H0;


for i = 1:m
JmenovatelPrenosu = conv([1 a], prod([1, -2*real(PolyVLevePolorovine(i)),
(real(PolyVLevePolorovine(i))).^2+(imag(PolyVLevePolorovine(i))).^2], m));
Omega0 = cos((2*i - 1)*pi/(2*n));
Rovnice = conv([(1./(epsilonp*H0)) 0], prod([1 0 (Omega0(i)).^2], m));
end


Prenos = tf(CitatelPrenosu, JmenovatelPrenosu)
CharakteristickaRovnice = tf(Rovnice, [0 1])
```

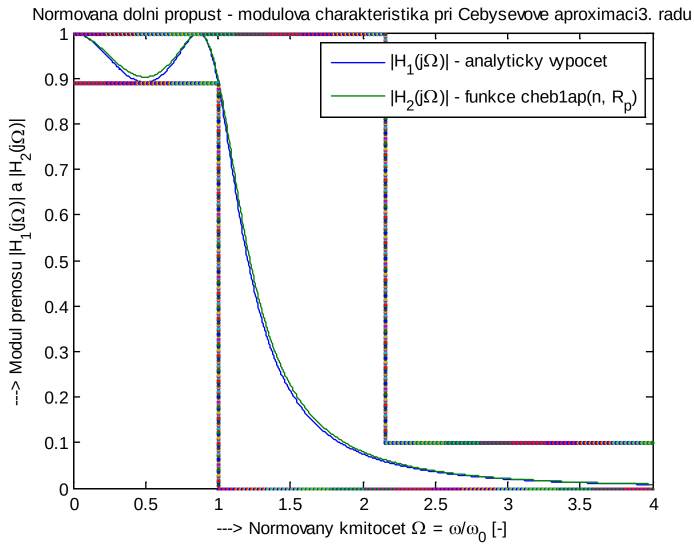

Figure 4a: Normalized low-pass filter – normalized frequency-magnitude characteristics for $\left|\hat{H}_1(j\cdot\Omega)\right|$ (obtained analytically) and $\left|\hat{H}_2(j\cdot\Omega)\right|$ (obtained by implemented MATLAB function of `cheb1ap(n, Rp)`); the Cartesian coordinates

Due to MATLAB possibilities and features, it is possible to plot not only normalized frequency-magnitude characteristic, but also other characteristics (e. g. real part of transfer function, imaginary part of transfer function, phase characteristic, group delay) in 3D graphs. For this transfer function form, mentioned in Eq. (28n), it is also possible to plot this function in 3D graph.

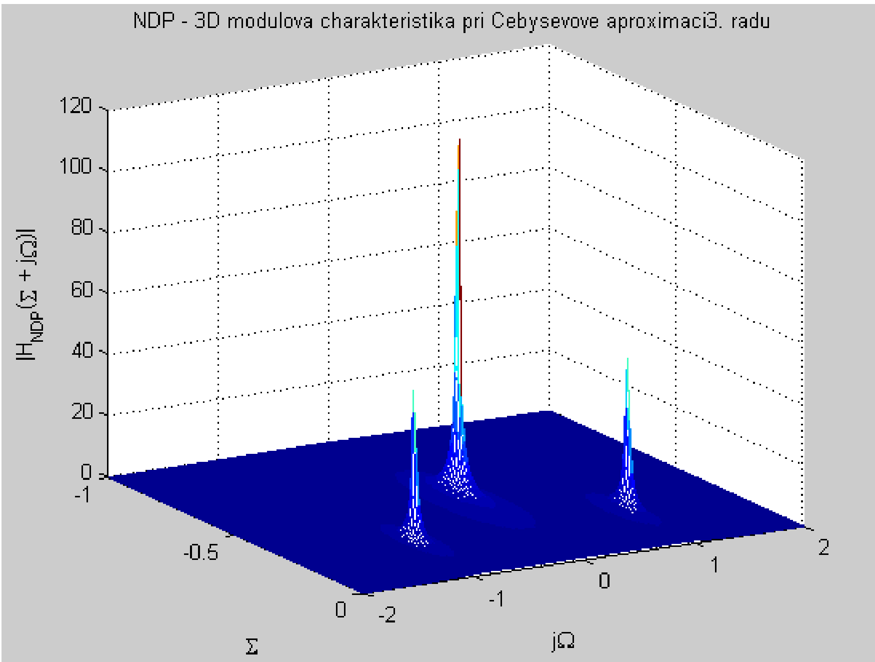

Figure 4b: Normalized low-pass filter – 3D graph of normalized frequency-magnitude characteristic (three poles of the transfer function are shown in the left part half-plane of the normalized complex area); the Cartesian coordinates

MATLAB code, where the normalized frequency-phase characteristic (unwrap mode) is plotted in 3D graph, is as follows:

```
n = 3;
Azimut = 60;
Elevace = 20;

B0 = 0.4913;
A0 = B0; A1 = 1.2384;
A2 = 0.9883; A3 = 1;

dSigma = 0.01; dOmega = dSigma;

[Delta, Sigma] = meshgrid(-1:dSigma:0, -2:dOmega:2);
s = Delta + j*Sigma;

FunkceZPrenos = B0./(A3*s.^3 + A2*s.^2 + A1*s + A0);
FunkceZSkupinove = - real((-3*s.^2 - 1.9766*s - 1.2384)./(s.^3 + 0.9883*s.^2 + 1.2384*s + 0.4913));
Imag = imag(FunkceZPrenos);
Real = real(FunkceZPrenos);

mesh(Delta, Sigma, unwrap(atan2(Imag, Real)));
```

Characteristic function (based on characteristic equation) is as follows, see Eq. (26b):

$$\varphi_{(n=3)}(p = j\cdot\Omega) = T_{(n=3)}(p = j\cdot\Omega) = j^3\cdot T_3(\Omega) = 4\cdot(j\cdot\Omega)^3 - 3\cdot\left(j^3\cdot\Omega\right) = 4\cdot p^3 + 3\cdot p \qquad (28r)$$

Group delay is as follows (in radians), see Eq. (27c):

$$\tau_g(\Omega) = -\operatorname{Re}\left\{\frac{1}{\hat{H}(j\cdot\Omega)}\cdot\frac{d}{d\Omega}\hat{H}(j\cdot\Omega)\right\} \cong \frac{0{,}9883\cdot\Omega^4 - 0{,}2500\cdot\Omega^2 + 0{,}6084}{\Omega^6 - 1{,}5001\cdot\Omega^4 + 0{,}5625\cdot\Omega^2 + 0{,}2414} \qquad (28s)$$

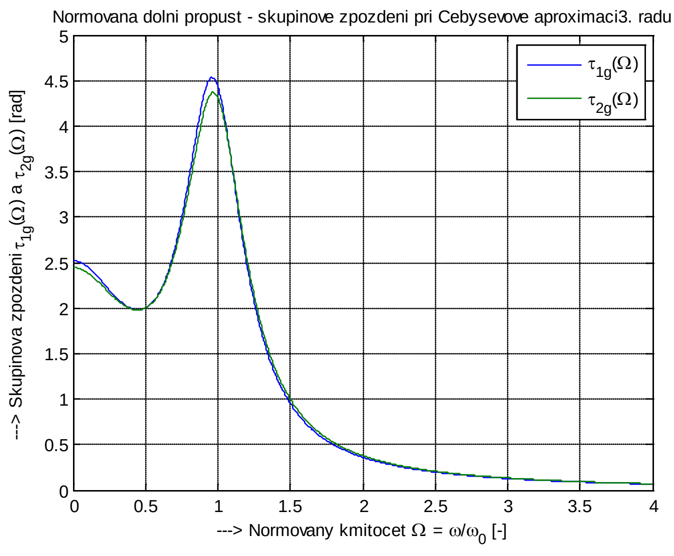

Figure 5: Normalized low-pass filter – normalized frequency-group delay characteristics for $\tau_{g1}(\Omega)$ (obtained by $\hat{H}_1(j\cdot\Omega)$) and $\tau_{g2}(\Omega)$ (obtained by $\hat{H}_2(j\cdot\Omega)$); the Cartesian coordinates

Using MATLAB, we created this own algorithm, based on self-developed formulas. Transfer functions $\hat{H}_1(j\cdot\Omega)$ (`Hs` in code) and $\hat{H}_2(j\cdot\Omega)$ (`H1s` in code) are the input parameters of this algorithm.

```
Hs = tf(NumHs, DenHs)
H1s = tf(NumH1s, DenH1s)

B01 = NumHs(length(NumHs)); A01 = DenHs(length(DenHs));
A11 = DenHs(length(DenHs) - 1); A21 = DenHs(length(DenHs) - 2);
A31 = DenHs(length(DenHs) - 3);

xOmega1 = A01 - A21*Omega.^2; dxOmega1 = - 2*A21*Omega;
yOmega1 = A11*Omega - A31*Omega.^3; dyOmega1 = A11 - 3*A31*Omega.^2;
NumTaug1 = dyOmega1.*xOmega1 - dxOmega1.*yOmega1;
DenTaug1 = xOmega1.^2 + yOmega1.^2;
Taug1 = NumTaug1./DenTaug1;

B02 = NumH1s(length(NumH1s)); A02 = DenH1s(length(DenH1s));
A12 = DenH1s(length(DenH1s) - 1); A22 = DenH1s(length(DenH1s) - 2);
A32 = DenH1s(length(DenH1s) - 3);

xOmega2 = A02 - A22*Omega.^2; dxOmega2 = - 2*A22*Omega;
yOmega2 = A12*Omega - A32*Omega.^3; dyOmega2 = A12 - 3*A32*Omega.^2;
NumTaug2 = dyOmega2.*xOmega2 - dxOmega2.*yOmega2;
DenTaug2 = xOmega2.^2 + yOmega2.^2;
Taug2 = NumTaug2./DenTaug2;
```

4.2 Analogue un-normalized filters

Due to MATLAB functions, implemented in Signal Processing Toolbox, it is possible to convert the normalized low-pass (NLP) filter into these un-normalized analogue filters, whereas [6] [7]:

- normalizedlow-pass (NLP) filter `cheb1ap(n, Rp)`,
- low-pass (LP) filter `lp2lp(b, a, Wo)`,
- high-pass (HP) filter `lp2hp(b, a, Wo)`,
- band-pass (BP) filter `lp2bp(b, a, Wo, Bw)`,
- band-stop (BS) filter `lp2bs(b, a, Wo, Bw)`.

Where

`n`	approximation order	n,
`Rp`	pass-band ripple	$R_p = 10^{-0,05\cdot a_p}$,
`b`	nominator of the NLP's transfer function	$\hat{B}_{NLP}(p)$,
`a`	denominator of the NLP's transfer function	$\hat{A}_{NLP}(p)$,
`Wo`	cut-off radial frequency	$\omega_0 = \omega_p = \dfrac{\omega_p}{\Omega_p}$,
`Bw`	radial frequency bandwidth	$B_\omega = 2\cdot\pi\cdot(f_H - f_L)$.

Naturally, these MATLAB functions are based on frequency transformation formulas:

for low-pass filter

$$\omega_p = \omega_0 \cdot \Omega_p = \omega_0 \quad \omega_s = \omega_0 \cdot \Omega_s \tag{29a}$$

for high-pass filter

$$\omega_p = \omega_0 \cdot \Omega_p^{-1} = \frac{\omega_0}{\Omega_p} = \omega_0 \quad \omega_s = \omega_0 \cdot \Omega_s^{-1} = \frac{\omega_0}{\Omega_s} \tag{29b}$$

for band-pass filter [9]

$$\Omega_p = \frac{\left|\omega_0^2 - \omega_{pd}^2\right|}{\omega_{pd} \cdot B} \quad B = \frac{\omega_{0H} - \omega_{0L}}{2 \cdot \pi} = f_{0H} - f_{0L} \tag{29c}$$

for band-stop filter [9]

$$\Omega_p = \frac{\omega_{pd} \cdot B}{\left|\omega_0^2 - \omega_{pd}^2\right|} \quad B = \frac{\omega_{0H} - \omega_{0L}}{2 \cdot \pi} = f_{0H} - f_{0L} \tag{29d}$$

For example, let us consider $\omega_0 = 2000 \cdot \pi \left[rad \cdot s^{-1}\right]$ (for LP and HP), $\omega_{pd} = 1500 \cdot \pi \left[rad \cdot s^{-1}\right]$, and $\omega_{pl} = 2500 \cdot \pi \left[rad \cdot s^{-1}\right]$ (for BP and BS). Radial frequency-magnitude characteristics of LP filter and BP filter are as follows:

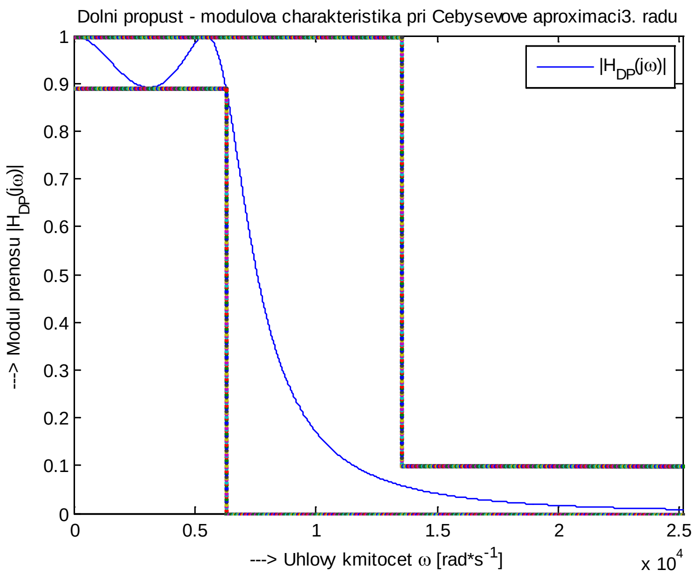

Figure 6a: Un-normalized low-pass filter – radial frequency-magnitude characteristic $\left|\hat{H}_{LP}(j \cdot \omega)\right|$; the Cartesian coordinates

MATLAB code is as follows:

```
[NumHDPs, DenHDPs] = lp2lp(NumHs, DenHs, omega0);
HDPs = tf(NumHDPs, DenHDPs)

[HDP, omegaDP] = freqs(NumHDPs, DenHDPs, omega);
plot(omegaDP, abs(HDP))
```

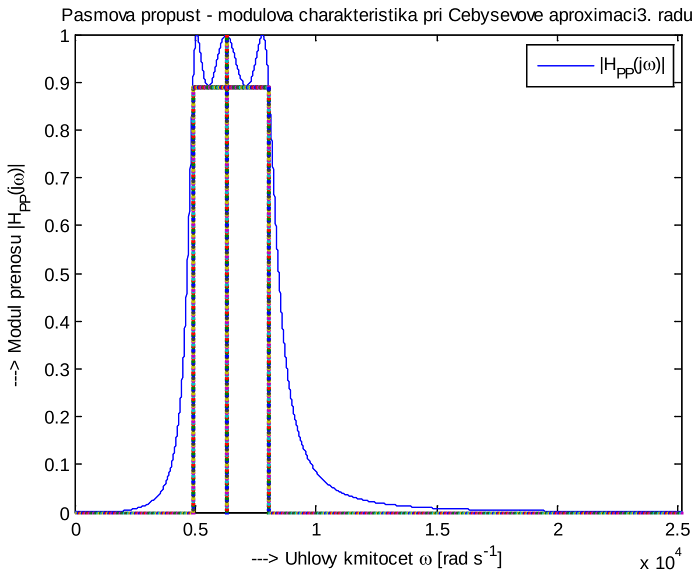

Figure 6b: Un-normalized band-pass filter – radial frequency-magnitude characteristic $\left|\hat{H}_{BP}\left(j\cdot\omega\right)\right|$; the Cartesian coordinates

MATLAB code is as follows:

```
[NumHPPs, DenHPPs] = lp2bp(NumHs, DenHs, omega0, Bomega);
HPPs = tf(NumHPPs, DenHPPs)

[HPP, omegaPP] = freqs(NumHPPs, DenHPPs, omega);
plot(omegaPP, abs(HPP))
```

4.3 Digital (IIR) un-normalized filters

Transfer function of IIR filter (IIR – Infinite Impulse Response) is given by this formula, thus [5]:

$$\hat{H}\left(z^{-1}\right)=\frac{\hat{B}\left(z^{-1}\right)}{\hat{A}\left(z^{-1}\right)}=\frac{\sum_{m=0}^{M} b_m\cdot z^{-m}}{1+\sum_{n=0}^{N} a_n\cdot z^{-n}}=\frac{b_0\cdot\prod_{\mu=1}^{M}\left(1-z_{0\mu}\cdot z^{-1}\right)}{\prod_{\nu=1}^{N}\left(1-z_{\infty\nu}\cdot z^{-1}\right)} \tag{30a}$$

Designing IIR filters is based on finding the coefficients of nominator $\hat{B}\left(z^{-1}\right)$ and denominator $\hat{A}\left(z^{-1}\right)$ of the transfer function, or finding zeros of $\hat{B}\left(z^{-1}\right)$ or poles of $\hat{A}\left(z^{-1}\right)$ to satisfy the requirements of tolerance scheme [5]. This design process is related to:

quadrature of magnitude characteristic

$$\hat{H}(z)\cdot\hat{H}\left(z^{-1}\right)=\left\|z=\exp\left(j\cdot\omega_d\right)\right\|\Rightarrow\left|\hat{H}\left(z=\exp\left(j\cdot\omega_d\right)\right)\right|^2 \tag{30b}$$

group delay

$$\tau_g\left(\omega_d\right)=\mathrm{Re}\left\{\frac{z}{\hat{A}(z)}\cdot\frac{d}{dz}\hat{A}(z)-\frac{z}{\hat{B}(z)}\cdot\frac{d}{dz}\hat{B}(z)\right\}_{z=\exp\left(j\cdot\omega_d\right)} \tag{30c}$$

Traditional approach is based on acceptance of analogue filters' approximations, because some methodology of NLP approximation has been managed yet. This paper demonstrates this idea.

During designing IIR filter using bilinear (Tustin) transformation, we consider the NLP approximation given by Eq. (28n) and use these approaches, thus [5]:

- **the first approach** – using frequency transformation, we obtain some analogue transfer functions; via AD transformation, we obtain some digital transfer function,
- **the second approach** – using AD transformation, we obtain some digital transfer functions; via frequency transformation, we obtain desired digital transfer function.

AD transformation must satisfy these conditions [5]:

- **the first condition** – stable analogue filter must be converted into stable digital filter,
- **the second condition** – fundamental frequency features of analogue filter must be kept in digital filter.

In MATLAB, there are two ways, how to approach the bilinear transformation, thus:

- **`bilinear(num, den, fs)`** – prewarped mode is not used to indicate "match" frequency, where `num` is a nominator of transfer function, `den` is a denominator of transfer function, and `fs` is sampling frequency [8]:

$$\hat{H}(z) = \hat{H}\left[s = 2 \cdot f_s \cdot \frac{z-1}{z+1} = \frac{2}{T_s} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \right] \tag{30d}$$

  and

$$\omega_d = 2 \cdot arctg\left(\frac{\omega_a}{2 \cdot f_s} \right) = 2 \cdot arctg\left(\frac{\omega_a \cdot T_s}{2} \right) \tag{30e}$$

- **`bilinear(num, den, fs, fp)`** – `fp` parameter specifies prewarping, which indicates "match" frequency, for which the frequency responses (before and after mapping) match exactly; this parameter value is identical with edge of pass-band of an analogue filter [8]:

$$\hat{H}(z) = \hat{H}\left[s = \frac{2 \cdot \pi}{tg\left(\frac{f_p}{f_s} \cdot \pi \right)} \cdot \frac{z-1}{z+1} = \frac{2 \cdot \pi}{tg\left(f_p \cdot T_s \cdot \pi \right)} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \right] \tag{30f}$$

  and

$$\omega_d = 2 \cdot arctg\left[\frac{\omega_a \cdot tg\left(\frac{f_p}{f_s} \cdot \pi \right)}{2 \cdot \pi \cdot f_p} \right] = 2 \cdot arctg\left[\frac{\omega_a \cdot tg\left(\frac{\omega_p}{2} \cdot T_s \right)}{\omega_p} \right] \tag{30g}$$

For example, let us consider $\omega_0 = \omega_p = 2000 \cdot \pi \left[rad \cdot s^{-1} \right]$ and $\omega_s = 8000 \cdot \pi \left[rad \cdot s^{-1} \right]$. Frequency ratio equals to:

$$p_\omega = \frac{\omega_0}{\omega_s} = \frac{2000 \cdot \pi}{8000 \cdot \pi} = \frac{1}{4} \tag{30h}$$

MATLAB codes, focused on this problem, are as follows:

for analogue NLP filter (s-plane)

```
[NumHNDPs, DenHNDPs] = cheby1(n, ap, Omegap, 'low', 's');
[HNDPs, omegaNDPs] = freqs(NumHNDPs, DenHNDPs);
```

for digital NLP filter (z-plane)

```
[NumHNDPz, DenHNDPz] = cheby1(n,ap,2*atan(Omegap/(2*fvz))/pi, 'low', 'z');
[HNDPz, omegaNDPz] = freqz(NumHNDPz, DenHNDPz);
```

for digital low-pass filter (prewarping mode is not considered)

```
[NumHDP1z, DenHDP1z] = cheby1(n,ap,2*atan(omega0*Omegap/(2*fvz))/pi, 'low',
'z');
[HDP1z, omegaDP1z] = freqz(NumHDP1z, DenHDP1z);
```

or (equivalent code)

```
[NumHDP2z, DenHDP2z] = bilinear(NumHDPs, DenHDPs, fvz);
[HDP2z, omegaDP2z] = freqz(NumHDP2z, DenHDP2z);
```

for digital low-pass filter (use of bilinear transformation, prewarping mode is considered)

```
[NumHDP3z, DenHDP3z] = cheby1(n, ap,
2*atan(omega0*Omegap*tan(pi*f0/fvz)/(2*pi*f0))/pi, 'low', 'z');
[HDP3z, omegaDP3z] = freqz(NumHDP3z, DenHDP3z);
```

or (equivalent code)

```
[NumHDP4z, DenHDP4z] = bilinear(NumHDPs, DenHDPs, fvz, f0);
[HDP4z, omegaDP4z] = freqz(NumHDP4z, DenHDP4z);
```

Fig. 7 shows normalized frequency-magnitude characteristics of designed digital low-pass filter. For detailed analysis of designed IIR filters, we can use a MATLAB tool called Filter Visualization Tool (`fvtool`).

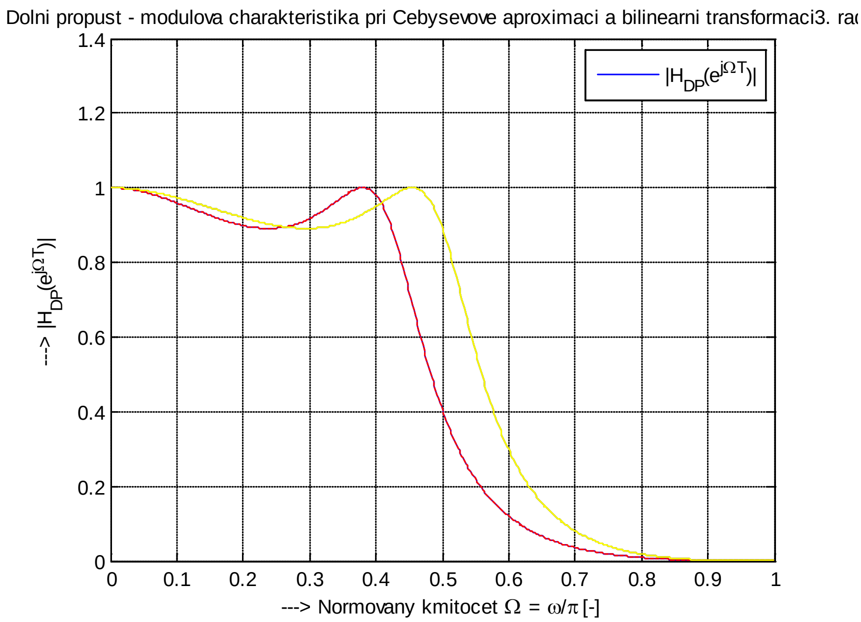

Figure 7: Un-normalized IIR low-pass filter – normalized frequency-magnitude characteristics, considering prewarped mode (yellow curve; we assume $\omega_s/\pi = 2 \Rightarrow \omega_0/\pi = p_\omega \cdot \omega_s/\pi = 0{,}5$) and non-prewarped mode (black curve); the Cartesian coordinates

5 Conclusions

This paper deals with using mathematical background, focused on the Chebyshev polynomials, selected MATLAB functions from Signal Processing Toolbox, and other tools (e. g. Filter Visulization Tool) for designing the first-order analogue Chebyshev filters and IIR Chebyshev filters. Because this paper is centred on mathematic viewpoint, there are a lot of fundamental (maybe essential) formulas in the first part of this paper. Well, we try to relate the Chebyshev polynomials to other polynomials (e. g. Jacobi, Gegenbauer) and formulas (Euler, de Moivre, Rodrigues), whereas most features of the Chebyshev polynomials are mentioned (definition, representations, recurrence relation, roots and extrema etc.).

The second part is focused on designing the first-order Chebyshev filters in *p*-plane (normalized analogue low-pass filter), *s*-plane (un-normalized analogue filters), and *z*-plane (IIR filters; normalized digital low-pass filter and un-normalized digital filters). Designing in p-plane is essential, it is this paper's kernel, for which we also consider some mathematical apparatus (fundamental approximation order, tolerance scheme, roots and extrema, normalized frequency characteristics, characteristic equation, group delay etc.). All these formulas are used to solve practical examples, based on approximation of the normalized low-pass filter. These examples include the formulas, 2D graphs, 3D graphs, and fragments of corresponding MATLAB code. Due to MATLAB&Simulink possibilities, it is possible to implement this approach, based on [5], not only in Simulink, but also in the field of FIR filters.

Acknowledgement

This work is supported by project SP2011/45, named "Data acquisition and processing from large distributed systems." of Student Grant System (VSB – Technical University of Ostrava) and by project TA01020932, named "Using Geothermal Energy for Renewable Energy Sources Systems Including Verification of Energy Accumulation." of Technology Agency of the Czech Republic.

References

[1] <http://en.wikipedia.org/wiki/Chebyshev_polynomials> [cit. 16th October 2011]
[2] <http://en.wikipedia.org/wiki/Gegenbauer_polynomials> [cit. 16th October 2011]
[3] <http://en.wikipedia.org/wiki/Jacobi_polynomials> [cit. 16th October 2011]
[4] J. Nečas et al.. *Aplikovaná matematika I (A až L)*. SNTL – Státní nakladatelství technické literatury, Praha, Česká republika, 1977
[5] V. Davídek, M. Laipert, M. Vlček. *Analogové a číslicové filtry*. Nakladatelství ČVUT, Praha, Česká republika, 345 s., 2006, ISBN 80-01-03026-1
[6] The Mathworks. *Product Help* – Analog Filters::Analog Lowpass Prototypes::cheb1ap. MATLAB 2010b Product Help [offline]
[7] The Mathworks. *Product Help* – Functions::Analog Filters::Analog Filter Transformation::lp2lp. MATLAB 2010b Product Help [offline]
[8] The Mathworks. *Product Help* – Functions::Analog Filters::Filter Discretization::bilinear. MATLAB 2010b Product Help [offline]
[9] J. Doleček. *Moderní učebnice elektroniky 6. díl – Kmitočtové filtry, generátory signálů a převodníky dat*. Nakladatelství BEN – technická literatura, Praha, Česká republika, 272 s., 2009, ISBN 978-80-7300-240-4

Petr Vojcinak
E-mail: petr.vojcinak@vsb.cz

Jiri Koziorek
E-mail: jiri.koziorek@vsb.cz

Radovan Hajovsky
E-mail: radovan.hajovsky@vsb.cz