

Notch Filtering Suitable for Real Time Removal of Power Line Interference

Pavel ZAHRADNIK¹, Miroslav VLČEK²

¹Dept. of Telecommunication Engg., Czech Technical University in Prague, Technická 2, 16627 Praha, Czech Republic

²Dept. of Applied Mathematics, Czech Technical University in Prague, Na Florenci 25, 11000 Praha, Czech Republic

zahradni@fel.cvut.cz, vlcek@fd.cvut.cz

Abstract. *This paper presents a high performance notch filtering for real time suppression of power line interference in a general signal. The disturbing signal is suppressed using an optimal notch FIR filter with tunable notch frequency. The tuning of the filter preserves its selectivity, most importantly the specified attenuation at the notch frequency. One example and two Matlab functions demonstrate the performance, robustness and usefulness of the proposed procedure for the design and tuning of optimal notch FIR filters suitable in the real time notch filtering.*

Keywords

Power line, interference suppression, FIR filter, notch filter.

1. Introduction

Weak signals, e.g. biosignals and signals originating from various sensors are almost always contaminated by power line interference. In the preprocessing of such signals, attenuation of the induced power line interference is desired. Power line interference is represented by a narrow band harmonic signal with varying amplitude and slowly varying frequency in a narrow frequency range. An efficient processing approach is to filter the corrupted signal using a notch filter. In order to keep the distortion of the desired signal as low as possible, the suppression band of the notch filter should be as narrow as possible. In order to preserve the form of the possible pulse like components in the desired signal, the notch filter should have a linear phase frequency response. These requirements naturally lead to notch FIR filters with a very narrow notch band. Because of the slowly varying frequency of the disturbing harmonic signal, adaptive filters are usually used. Numerous references in adaptive notch filtering are available, see e.g. [1]-[6]. The drawback of these approaches, which are usually based on LMS minimization or neural networks, consists in their suboptimality in terms of the filter length related to its selectivity. Another disadvantage is that the attenuation of these adaptive filters at the notch frequency varies in the adaptation, and conse-

quently a strong attenuation of the disturbing signal at the notch frequency is not guaranteed. Moreover, the attenuation at the notch frequency is usually not specified at all and its actual value results from the adaptation process, see e.g. [6]. In our approach presented here, we focus on high performance real time capable attenuation of the disturbing narrow band signal. This approach is based on an optimal equiripple notch FIR filter with a specified width of its notch band and with strong attenuation at its notch frequency. The equiripple notch FIR filter is optimal in terms of the shortest possible filter length related to its frequency specification. It is assumed in our approach that the instantaneous frequency of the disturbing harmonic signal is an input information obtained from the power line, e.g. by evaluating the zero crossings. The value of the instantaneous frequency is forwarded to the filtering system, and the notch frequency of the filter is adjusted accordingly in real time. Note that the proposed filter design and the adjustment of its notch frequency are based on formulas. No numerical procedures are involved. This makes the presented approach an attractive option for the real time signal processing using digital signal processors. Basic terms are introduced in Section 2. Sections 3 and 4 summarize the design of the initial optimal notch FIR filter. Section 5 presents the tuning of the initial notch FIR filter. Finally, Section 6 presents an example of the filter design and of its tuning. For reference and for checking its performance, both procedures are summarized in the form of a Matlab code.

2. Basic Terms

We assume a general FIR filter of type I represented by its impulse response $h(k)$ with odd length of $N = 2n + 1$ coefficients and with even symmetry (1). In our further considerations we use the a -vector $a(k)$, which is related to the impulse response $h(k)$

$$a(0) = h(n) , a(k) = 2h(n+k) = 2h(n-k) , k = 1 \dots n . \quad (1)$$

Further, we introduce a real variable w

$$w = \cos(\omega T) = \cos\left(2\pi \frac{f}{f_s}\right) \quad (2)$$

and we represent the FIR filter by the real valued zero phase transfer function $Q(w)$, which we express in form of the expansion

$$Q(w) = \sum_{k=0}^n a(k) T_k(w) \tag{3}$$

where $T_k(w) = \cos(k \arccos w)$ is the Chebyshev polynomial of the first kind. The zero phase transfer function of an optimal notch FIR filter is

$$Q(w) = \frac{y_m - Z_{p,q}(w, \kappa)}{y_m + 1} \tag{4}$$

where $Z_{p,q}(w, \kappa)$ is the Zolotarev polynomial [9]. For illustration, the form of a Zolotarev polynomial is shown in Fig. 1.

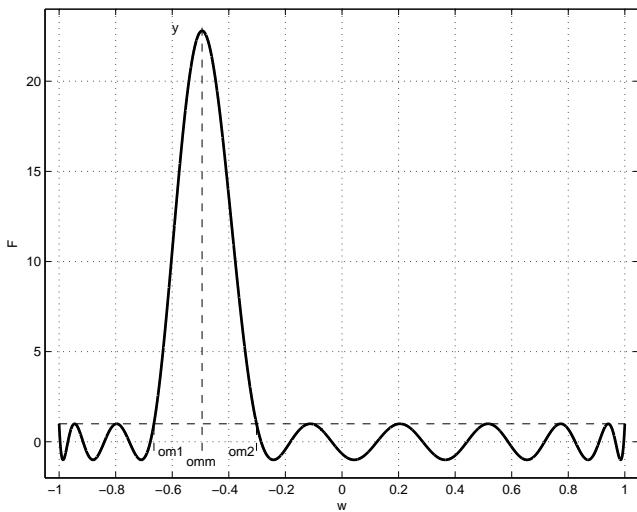


Fig. 1. Zolotarev polynomial $Z_{12,6}(w, 0.79023439)$.

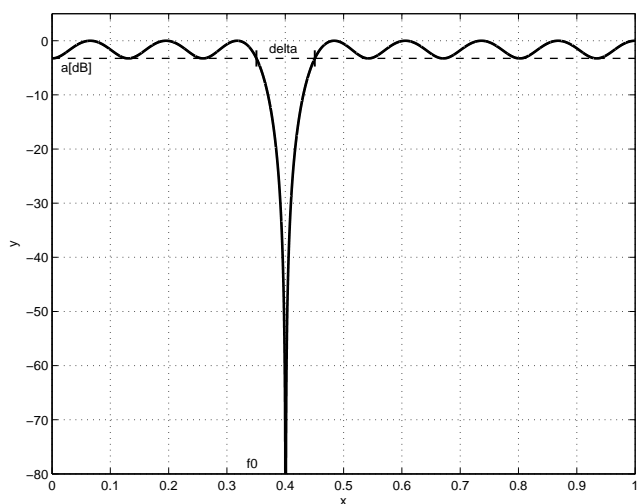


Fig. 2. Amplitude frequency response $20 \log |H(e^{j2\pi f/f_s})|$ [dB].

3. Optimal Notch FIR Filter

An optimal notch FIR filter (Fig. 2) is specified by the notch frequency f_0 , the width of the notchband Δf , the sampling frequency f_s , and the attenuation in the passbands a [dB]. There are two goals in FIR filter design. The first is to obtain the minimal filter length of N coefficients satisfying the filter specification. The second goal of the design is to evaluate the impulse response $h(k)$. In the standard design of an optimal notch FIR filter, which is represented by the numerical Parks-McClellan procedure (e.g. function *firpm* in Matlab), the exact filter length is not available because there is no approximating function. Consequently, the filter length is not the result of the design, but it is an input argument. The filter length is either estimated or successively adjusted in order to meet the filter specification, or it is obtained from empirical approximating formulas. A prohibitive property of the Parks-McClellan procedure in adaptive filtering lies in the fact that it is based on numerical optimization, and it consequently lacks determinism. Thus the number of arithmetic operations in the filter design cannot be predicted, as it depends on its convergence properties in minimizing the error function. Moreover, a successful design is not guaranteed in the Parks-McClellan procedure. The alternative design approach that we use here is based on the analytical design introduced in [7]. This design is based on a closed form solution. Formulas are available for exact determining the filter length and also for evaluating the coefficients of the impulse response. The proposed notch filtering with variable notch frequency is based on the design of an initial notch FIR filter and continuous adjusting of the notch frequency according to the instantaneous power line frequency. The determinism of this design makes it attractive in adaptive filtering. The initial filter is an optimal notch FIR filter with equiripple behavior of its frequency response in both passbands. The available notch frequencies are quantized because there is an integer number of ripples (Fig. 2) in both passbands. From the initial filter, related notch FIR filters are derived by real time fine tuning of its notch frequency to the desired value. This tuning consists in multiplying the a -vector of the initial filter by a transformation matrix, resulting in the a -vector of the tuned filter (Sec. 5). Note that there is a true zero value at the notch frequency. The achievable attenuation a_0 [dB] at the notch frequency is limited only by the arithmetics used in the filter design. In Matlab, the attenuation a_0 [dB] is usually better than -300 dB.

4. Design of the Initial Notch FIR Filter

Let us specify the initial optimal equiripple notch FIR filter (Fig. 2) by the notch frequency f_0 [Hz], width of the notch band Δf [Hz], sampling frequency f_s [Hz] and by the attenuation in both passbands a [dB]. The design procedure consists of a small number of steps as follows: Calculate the Jacobi elliptic modulus κ

$$\kappa = \sqrt{1 - \frac{1}{\tan^2(\varphi_1)\tan^2(\varphi_2)}} \quad (5)$$

for the auxiliary values φ_1 and φ_2

$$\varphi_1 = \frac{\pi}{f_s} \left(f_0 + \frac{\Delta f}{2} \right), \quad \varphi_2 = \frac{\pi}{f_s} \left(\frac{f_s}{2} - f_0 + \frac{\Delta f}{2} \right). \quad (6)$$

Calculate the rational values p/n and q/n

$$\frac{p}{n} = \frac{F(\varphi_1, \kappa)}{\mathbf{K}(\kappa)}, \quad \frac{q}{n} = \frac{F(\varphi_2, \kappa)}{\mathbf{K}(\kappa)}, \quad (7)$$

where $\mathbf{K}(\kappa)$ is a complete elliptic integral of the first kind, which here represents the elliptic quarter-period. Determine the maximizer y_m

$$y_m = \frac{2}{1 - 10^{0.05a[\text{dB}]}} - 1. \quad (8)$$

Calculate the auxiliary value σ_m

$$\sigma_m = F\left(\arcsin \frac{\sqrt{\cos\left(2\pi\frac{f_0}{f_s}\right) - \cos\left[\frac{2\pi}{f_s}\left(f_0 + \frac{\Delta f}{2}\right)\right]}}{\kappa \operatorname{sn}\left(\frac{p}{n}\mathbf{K}(\kappa), \kappa\right) \sqrt{1 + \cos\left(2\pi\frac{f_0}{f_s}\right)}}, \kappa\right). \quad (9)$$

Calculate and round up the value n

$$n \geq \frac{\ln(y_m + \sqrt{y_m^2 - 1})}{2\sigma_m Z\left(\frac{p}{n}\mathbf{K}(\kappa), \kappa\right) - 2\Pi\left(\sigma_m, \frac{p}{n}\mathbf{K}(\kappa), \kappa\right)}, \quad (10)$$

which represents the degree $n = p + q$ of the Zolotarev polynomial $Z_{p,q}(w, \kappa)$. Calculate the integer indices p and q of the Zolotarev polynomial $Z_{p,q}(w, \kappa)$

$$p = \left\lceil n \frac{F(\varphi_1, \kappa)}{\mathbf{K}(\kappa)} \right\rceil, \quad q = \left\lceil n \frac{F(\varphi_2, \kappa)}{\mathbf{K}(\kappa)} \right\rceil. \quad (11)$$

The square brackets in (11) stand for rounding. For values p, q (11), κ (5) and y_m (8) evaluate the impulse response $h(k)$ and the a -vector $a(k)$ using the algebraical procedure summarized in Tab. 1. Check the actual notch frequency f_m of the initial notch FIR filter

$$f_m = \frac{f_s}{2\pi} \arccos \left[1 - 2 \operatorname{sn}^2 \left(\frac{p}{n}\mathbf{K}(\kappa), \kappa \right) + 2 \frac{\operatorname{sn} \left(\frac{p}{n}\mathbf{K}(\kappa), \kappa \right) \operatorname{cn} \left(\frac{p}{n}\mathbf{K}(\kappa), \kappa \right)}{\operatorname{dn} \left(\frac{p}{n}\mathbf{K}(\kappa), \kappa \right)} Z \left(\frac{p}{n}\mathbf{K}(\kappa), \kappa \right) \right]. \quad (12)$$

Because of its inherent quantization mentioned above, the actual notch frequency f_m usually differs slightly from the specified notch frequency f_0 . In the next section, the notch frequency f_m is tuned to the specified value f_0 . In the calculations, the Jacobi elliptic Zeta function $Z(x, \kappa)$ in (10), (12)

and the incomplete elliptic integral of the first kind $F(x, \kappa)$ in (7), (9) are evaluated by the arithmetic-geometric mean [10]. The Jacobi elliptic integral of the third kind $\Pi(x, y, \kappa)$ in (10) is evaluated by a fast procedure proposed in [7]. As no tool is available for the algebraic design of optimal notch FIR filters, we present the design procedure in the form of a Matlab function *firnotch* in Tab. 3. The function *firnotch* accepts the specified notch frequency f_0 [Hz], the width of the stopband Δf [Hz], the sampling frequency f_s [Hz], and the attenuation in the passbands a [dB]. It outputs $N = 2n + 1$ coefficients of the impulse $h(k)$, $n + 1$ coefficients of the vector $a(k)$, actual notch frequency f_m [Hz], attenuation at the notch frequency a_m [dB] and actual attenuation in the passbands a_{act} [dB]. The presented code proves the speed and robustness of the design of the initial notch filter.

5. Tuning of the Initial Notch FIR Filter

The purpose of the tuning presented here is to adjust the actual notch frequency f_m of the initial filter to the specified value f_0 . For the tuning, we present an efficient algebraic procedure which is a simplified version of the procedure introduced in [8]. The tuning procedure results from the zero phase transfer function of the tuned filter, which is

$$Q_t(w) = \sum_{k=0}^n a(k) T_k(\lambda w \pm \lambda') = \sum_{k=0}^n a(k) \sum_{m=0}^k \alpha_k(m) T_m(w). \quad (13)$$

Based on (13), the a -vector $a_t(k)$ of the tuned filter and the a -vector $a(k)$ of the initial filter are related by a triangular transformation matrix A

$$a_t(k) = [a_t(0) \ a_t(1) \ \dots \ a_t(n)] = [a(0) \ a(1) \ \dots \ a(n)] \times \begin{bmatrix} \alpha_0(0) & 0 & 0 & 0 & \dots & 0 \\ \alpha_1(0) & \alpha_1(1) & 0 & 0 & \dots & 0 \\ \alpha_2(0) & \alpha_2(1) & \alpha_2(2) & 0 & \dots & 0 \\ \alpha_3(0) & \alpha_3(1) & \alpha_3(2) & \alpha_3(3) & \dots & 0 \\ \vdots & & & & \ddots & \\ \alpha_n(0) & \alpha_n(1) & \alpha_n(2) & \alpha_n(3) & \dots & \alpha_n(n) \end{bmatrix} = a(k)A. \quad (14)$$

Rapid evaluation of the coefficients $\alpha_k(m)$ of the transformation matrix A is essential in adaptive filtering. A fast algorithm for the coefficients $\alpha_k(m)$ is summarized in Tab. 2. It is worth noting that the presented tuning of the notch frequency preserves both the attenuation a_0 [dB] at the notch frequency and the attenuation a [dB] in the passbands. The Matlab function *tune* of the tuning procedure is presented in Tab. 4. The function *tune* accepts the impulse response $h(k)$ and the notch frequency f_m [Hz] of the initial filter, the desired notch frequency f_0 [Hz], the sampling frequency f_s [Hz] and returns the impulse response $h_t(k)$ of the tuned filter.

<i>given</i>	p, q, κ, y_m
<i>initialisation</i>	$n = p + q, w_1 = 1 - 2 \operatorname{sn}^2 \left(\frac{p}{n} \mathbf{K}(\kappa), \kappa \right), w_2 = 2 \operatorname{sn}^2 \left(\frac{q}{n} \mathbf{K}(\kappa), \kappa \right) - 1, w_a = \frac{w_1 + w_2}{2}$ $w_m = w_1 + 2 \frac{\operatorname{sn} \left(\frac{p}{n} \mathbf{K}(\kappa), \kappa \right) \operatorname{cn} \left(\frac{p}{n} \mathbf{K}(\kappa), \kappa \right)}{\operatorname{dn} \left(\frac{p}{n} \mathbf{K}(\kappa), \kappa \right)} Z \left(\frac{p}{n} \mathbf{K}(\kappa), \kappa \right)$
	$\alpha(n) = 1, \alpha(n+1) = \alpha(n+2) = \alpha(n+3) = \alpha(n+4) = \alpha(n+5) = 0$
<i>body</i>	
<i>(for</i>	$m = n+2 \text{ to } 3)$
	$8c(1) = n^2 - (m+3)^2, 4c(2) = (2m+5)(m+2)(w_m - w_a) + 3w_m[n^2 - (m+2)^2]$
	$2c(3) = \frac{3}{4}[n^2 - (m+1)^2] + 3w_m[n^2w_m - (m+1)^2w_a] - (m+1)(m+2)(w_1w_2 - w_mw_a)$
	$c(4) = \frac{3}{2}(n^2 - m^2) + m^2(w_m - w_a) + w_m(n^2w_m^2 - m^2w_1w_2)$
	$2c(5) = \frac{3}{4}[n^2 - (m-1)^2] + 3w_m[n^2w_m - (m-1)^2w_a] - (m-1)(m-2)(w_1w_2 - w_mw_a)$
	$4c(6) = (2m-5)(m-2)(w_m - w_a) + 3w_m[n^2 - (m-2)^2], 8c(7) = n^2 - (m-3)^2$
	$\alpha(m-3) = \frac{1}{c(7)} \sum_{\mu=1}^6 c(\mu)\alpha(m+4-\mu)$
<i>(end loop on m)</i>	
<i>normalisation</i>	$s(n) = \frac{\alpha(0)}{2} + \sum_{m=1}^n \alpha(m)$
<i>a-vector</i>	$a(0) = (-1)^p \frac{\alpha(0)}{2s(n)}, \text{ (for } m = 1 \text{ to } n), a(m) = (-1)^p \frac{\alpha(m)}{s(n)}, \text{ (end loop on } m)$
<i>impulse response</i>	$h(n) = \frac{y_m - a(0)}{y_m + 1}, \text{ (for } m = 1 \text{ to } n), h(n \pm m) = -\frac{a(m)}{2(y_m + 1)}, \text{ (end loop on } m)$

Tab. 1. Fast algorithm for evaluating the a -vector $a(k)$ and the impulse response $h(k)$.

<i>given</i>	f_m, f_0, f_s, k
<i>initialization</i>	if $f_m < f_0 : \lambda = \frac{\cos \left(2\pi \frac{f_m}{f_s} \right) - 1}{\cos \left(2\pi \frac{f_0}{f_s} \right) - 1}, s = 1, \text{ else : } \lambda = \frac{\cos \left(2\pi \frac{f_m}{f_s} \right) + 1}{\cos \left(2\pi \frac{f_0}{f_s} \right) + 1}, s = -1$
	$\lambda' = 1 - \lambda, \alpha_k(k+1) = \alpha_k(k+2) = \alpha_k(k+3) = 0, \alpha_k(k) = \lambda^k$
<i>body</i>	
<i>(for</i>	$\mu = -3 \dots k-4)$
	$\alpha_k(k-\mu-4) =$ $\{$ $-2s \left[(\mu+3)(2k-\mu-3) - \frac{\lambda'}{\lambda} (k-\mu-3)(2k-2\mu-7) \right] \alpha_k(k-\mu-3)$ $\quad + 2 \frac{\lambda'}{\lambda} (k-\mu-2) \alpha_k(k-\mu-2)$ $+ 2s \left[(\mu+1)(2k-\mu-1) - \frac{\lambda'}{\lambda} (k-\mu-1)(2k-2\mu-1) \right] \alpha_k(k-\mu-1)$ $\quad + \mu(2k-\mu) \alpha_k(k-\mu)$ $\} / (\mu+4)(2k-\mu-4)$
<i>(end loop on μ)</i>	

Tab. 2. Fast algorithm for evaluating the coefficients $\alpha_k(m)$ of transformation matrix A .

```

function [n, h, f_m, adB_pass, adB_m]=firnotch(f_0, delta_f, fs, adB) % Design
om_0_pi=2*pi*f_0/fs; delom=2*pi*delta_f/fs;
om_0=om_0_pi/pi; ym_s=-1+2/(1-10^(0.05*adB)); omp=om_0_pi-delom/2; oms=om_0_pi+delom/2; ws=cos(oms);
wp=cos(omp); w_0=cos(om_0_pi); fip=(pi-omp)/2; fis=oms/2; const=1/(tan(fis)*tan(fip));
k=sqrt(1-const*const); quar=ellipke(k*k); Ffis=ielli_fu(fis,k); Ffip=ielli_fu(fip,k); pn=Ffis/quar;
qn=Ffip/quar; if fip==fis pn=0.5; qn=0.5; end; n=degrez_fu(ym_s,pn,k,w_0,ws); n=ceil(n); p=round(pn*n);
q=round(qn*n); [alph,wm]=zol_fu(p,q,k); tm=alph; tm=tm/2; tm(1)=2*tm(1); tp=fliplr(tm);
hn=[tp(1,1:max(size(tp))-1) tm]; om_m_pi=acos(wm); om_m=om_m_pi/pi; mazer=mag_fu(hn,[om_m_pi]);
if mazer<ym_s n=n+1; p=round(pn*n); q=round(qn*n); [alph,wm]=zol_fu(p,q,k); tm=alph; tm=tm/2; tm(1)=2*tm(1);
tp=fliplr(tm); hn=[tp(1,1:max(size(tp))-1) tm]; om_m_pi=acos(wm); om_m=om_m_pi/pi;
mazer=mag_fu(hn,[om_m_pi]); end; adB_pass=20*log10(1-2/(mazer+1)); tm=alph; tm=tm/2; tm(1)=2*tm(1);
tp=fliplr(tm); hn=[tp(1,1:max(size(tp))-1) tm]; hn=-hn;
hn(1,1+floor(max(size(hn))/2))=mazer+hn(1,1+floor(max(size(hn))/2)); hn=hn/(1+mazer);
adB_m=20*log10(abs(mag_fu(hn,[om_m_pi]))); f_m=om_m*fs/2; h=hn;
function y=Jacobi_Zeta_AGM_fu(u,k) % Jacobi Zeta function using AGM
sma=1e-12; kc=sqrt(1-k*k); a(1)=1; b(1)=kc; c(1)=k; n=1; while(c(n)>sma) n=n+1; a(n)=0.5*(a(n-1)+b(n-1));
b(n)=sqrt(a(n-1)*b(n-1)); c(n)=0.5*(a(n-1)-b(n-1)); end; fi(n)=(2^(n-1))*a(n)*u; y=0;
for m = n:-1:2 cor=c(m)/a(m)*sin(fi(m)); fi(m-1)=0.5*(fi(m)+asin(cor)); y=y+c(m)*sin(fi(m)); end
function y=ielli_fu(fi,k) % Incomplete Elliptic Integral of the First Kind using AGM
delta=1e-16; kc=sqrt(1-k*k); a0=1; b0=kc; c0=k; fi0=fi; akml=a0; bkml=b0; ckml=c0; kml=1; fikml=fi0; n=0;
while(ckml>delta) n=n+1; ak=(akml+bkml)/2; bk=sqrt(akml*bkml); ck=(akml-bkml)/2; el=floor((1+2*fikml/pi)/2);
fik=fikml+atan2(bkml*tan(fikml),akml)+el*pi; fikml=fik; akml=ak; bkml=bk; ckml=ck; end; y=fik/(ak^2*n);
function y=ellipj_fu(u,n,k) % Jacobi elliptic integral of third kind PI
quarter=ellipke(k*k); si=ellipj(u,k*k); s=ellipj((1:n)*quarter/n,k*k); su=ellipj((1:n)*quarter/n+u,k*k);
sup=ellipj((1:n)*quarter/n-u,k*k); v=u*k*k*s(1)*s(n-(1:n-1)).*s(n+1-(1:n-1));
num=1-k*k*s(1)*s(n-(1:n-1)).*sup(n+1-(1:n-1)); den=1+k*k*s(1)*s(n-(1:n-1)).*su(n+1-(1:n-1));
r=log(num./den)/2 + v; a=diag(n-1:-1:1)*ones(n-1); b=ones(n-1)-tril(ones(n-1)); y=flipud(1/n*(a-n)*r');
function y=degrez_fu(ym,pn,k,wm,ws) % Degree of Zolotarev polynomial
cit=sqrt((wm-ws)/(wm+1)); quar=ellipke(k*k); [sn,cn,dn]=ellipj(pn*quar,k*k); jme=k*sn; argu=asin(cit/jme);
sigmam=ielli_fu(argu,k); pe=round(pn*1000); eli3=ellipj_fu(sigmam,1000,k)'; Pii=eli3(pe);
cit=log(ym+sqrt(ym*ym-1)); jme=2*sigmam*Jacobi_Zeta_AGM_fu(pn*quar,k)-2*Pii; y=cit/jme;
function [a,wm]=zol_fu(p,q,km) % Zolotarev polynomial
n=p+q; quarter=ellipke(km*km); up=p*quarter/n; [sn,cn,dn]=ellipj(up,km*km);
wp=2*(cn./dn)^2-1; ws=2*cn^2-1; wm=ws+2*sn*cn/dn*Jacobi_Zeta_AGM_fu(up,km);
wq=(wp+ws)/2; alfa=zeros(1,n+6); alfa(n+1)=1;
for m=n+2:-1:3 c7=(n*(n-(m-3))*(m-3))/8; c6=((2*m-5)*(m-2)*(wm-wq)+3*wm*(n*(n-(m-2))*(m-2)))/4;
c5=3/8*(n*(n-(m-1))*(m-1))+3/2*wm*(n*(n*(m-1)*(m-1)*wq)-1/2*(m-1)*(m-2)*(wp*ws-wm*wq);
c4=3/2*(n*(n-m)*wm*wm*(m-wq)+wm*(n*(n*(m-1)*(m-1)*wq)-1/2*(m-1)*(m-2)*(wp*ws-wm*wq);
c3=3/8*(n*(n-(m+1))*(m+1))+3/2*wm*(n*(n*(m-1)*(m-1)*wq)-1/2*(m+1)*(m+2)*(wp*ws-wm*wq);
c2=((2*m+5)*(m+2)*(wm-wq)+3*wm*(n*(n*(m+2)*(m+2)))/4; c1=(n*(n-(m+3))*(m+3))/8;
alfa(m-2)=(c6*alfa(m-1)-c5*alfa(m)+c4*alfa(m+1)-c3*alfa(m+2)+c2*alfa(m+3)-c1*alfa(m+4))/c7; end
alfa=[alfa(1)/2, alfa(2:n+1)]; suma=alfa(1); for m=1:n suma=suma+alfa(m); end; a=(-1)^p*alfa/suma;
function y=mag_fu(hn,om) % Magnitude of odd length FIR filter
nn=(max(size(hn))-1)/2; y = hn(1+nn)*ones(max(size(om)),1); for m=1:nn y = y + 2*hn(nn+m+1)*cos(m*om); end
    
```

Tab. 3. Matlab code - design of the initial optimal notch FIR filter.

```

function h_t=tune(h, f_m, f_0, fs) % tuning f_m --> f_0
om_m_pi=2*pi*f_m/fs; om_0_pi=2*pi*f_0/fs; n=(max(size(h))-1)/2;
a=h(1,(max(size(h))+1)/2:max(size(h))); a=2*a; a(1)=a(1)/2; at=0*a;
for row=1:n+1,
if cos(om_m_pi)>cos(om_0_pi)
lambda=(-1+cos(om_m_pi))/(-1+cos(om_0_pi)); VecA=VectorA_fu(max(size(a)-1),lambda,0,row); end;
if cos(om_m_pi)<cos(om_0_pi)
lambda=(1+cos(om_m_pi))/(1+cos(om_0_pi)); VecA=VectorA_fu(max(size(a)-1),lambda,1,row); end;
at=at+VecA*a(row); end
a=at; tm=a; tm=tm/2; tm(1)=2*tm(1); tp=fliplr(tm); h_t=[tp(1,1:max(size(tp))-1) tm];
function A_row=VectorA_fu(n,lambda,dir,row)
A_row=zeros(1,n+1); if dir == 0 A_row(1:row)=tposun_fu(row-1,lambda, 1-lambda,-1);
elseif dir == 1 A_row(1:row)=tposun_fu(row-1,lambda, 1-lambda,1);
else A_row(1:row)=tposun_fu(row-1, 0, 1); size(A_row); end
function A=MatrixA_fu(n,lambda,dir)
for i=1:n+1, row=zeros(1,n+1); A(i,:)=row; end
if dir == 0 for i=1:n+1, aplus=tposun_fu(i-1,lambda, 1-lambda,-1); A(i,1:i)=aplus; end
elseif dir == 1 for i=1:n+1, aminus=tposun_fu(i-1,lambda,1-lambda,1); A(i,1:i)=aminus; end
else for i=1:n+1, aminus=tposun_fu(i-1,1,0,1); A(i,1:i)=aminus; end; end
function alfa=tposun_fu(n,lambda,lambdac,s)
if n==0, alfa(1)=1; else alfa=zeros(1,n+4); alfa(n+1)=lambda^n; for k=1:1:n d1=k*(2*n-k);
d2=2*s*(k-1)*(2*n+1-k)-lambdac/lambdac*(n+1-k)*(2*n+1-2*k); d3=4*lambdac/lambdac*(n+2-k);
d4=-2*s*(k-3)*(2*n+3-k)-lambdac/lambdac*(n+3-k)*(2*n+7-2*k); d5=(k-4)*(2*n+4-k);
alfa(n+1-k)=(alfa(n+2-k)*d2+alfa(n+3-k)*d3+alfa(n+4-k)*d4+alfa(n+5-k)*d5)/d1; end
alfa=[alfa(1)/2 alfa(2:n+1)]; end
    
```

Tab. 4. Matlab code - tuning of the initial optimal notch FIR filter.

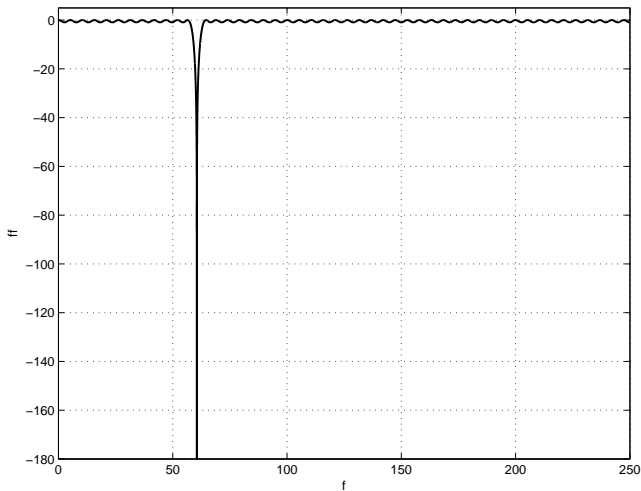


Fig. 3. Amplitude frequency response $20 \log |H(e^{j2\pi f/f_s})|$ [dB].

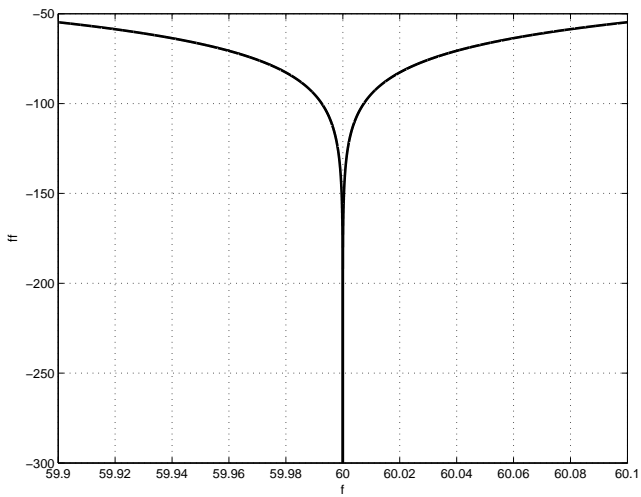


Fig. 4. Amplitude frequency response $20 \log |H(e^{j2\pi f/f_s})|$ [dB].

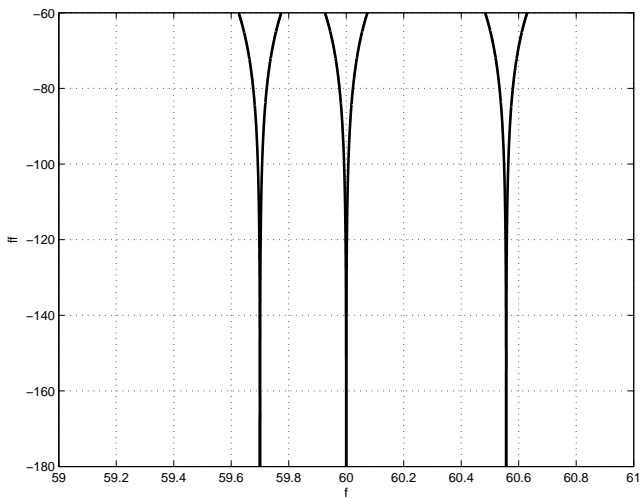


Fig. 5. Amplitude frequency responses $20 \log |H(e^{j2\pi f/f_s})|$ [dB].

k	60.5565 Hz	60 Hz	59.7 Hz	
0	.190	0.02747557	0.02109292	0.01831150
1	.189	0.00272714	-0.00906831	-0.01306689
2	.188	0.00031912	-0.00911154	-0.01018648
3	.187	-0.00243456	-0.00870548	-0.00730212
4	.186	-0.00398011	-0.00674753	-0.00351480
5	.185	-0.00335408	-0.00317951	0.00091920
6	.184	-0.00077524	0.00094633	0.00481146
7	.183	0.00240572	0.00402062	0.00677250
8	.182	0.00441003	0.00477672	0.00607874
9	.181	0.00402370	0.00304997	0.00320152
10	.180	0.00132618	-0.00004129	-0.00031606
11	.179	-0.00227932	-0.00263027	-0.00260301
12	.178	-0.00479183	-0.00309215	-0.00244342
13	.177	-0.00472168	-0.00095813	0.00003875
14	.176	-0.00196874	0.00276685	0.00339517
15	.175	0.00204552	0.00609537	0.00560129
16	.174	0.00510759	0.00712599	0.00516208
17	.173	0.00543137	0.00511951	0.00197960
18	.172	0.00269636	0.00097888	-0.00251275
19	.171	-0.00169712	-0.00316742	-0.00606178
20	.170	-0.00533988	-0.00509010	-0.00679926
21	.169	-0.00613414	-0.00369643	-0.00429481
22	.168	-0.00349907	0.00030751	0.00013551
23	.167	0.00122997	0.00472808	0.00407247
24	.166	0.00547238	0.00704043	0.00527753
25	.165	0.00680992	0.00578198	0.00293570
26	.164	0.00436360	0.00141527	-0.00183255
27	.163	-0.00064340	-0.00384514	-0.00650624
28	.162	-0.00549058	-0.00718988	-0.00849556
29	.161	-0.00743783	-0.00675204	-0.00656825
30	.160	-0.00527368	-0.00268328	-0.00158641
31	.159	-0.00005960	0.00284967	0.00388133
32	.158	0.00538239	0.00678922	0.00690387
33	.157	0.00799684	0.00684975	0.00578412
34	.156	0.00621031	0.00281462	0.00103775
35	.155	0.00087221	-0.00326766	-0.00480876
36	.154	-0.00513869	-0.00813737	-0.00853323
37	.153	-0.00846647	-0.00908715	-0.00796260
38	.152	-0.00715235	-0.00548064	-0.00320287
39	.151	-0.00178377	0.00082480	0.00334037
40	.150	0.00475380	0.00642307	0.00820320
41	.149	0.00882753	0.00819275	0.00872653
42	.148	0.00807706	0.00503167	0.00453827
43	.147	0.00278001	-0.00147166	-0.00217147
44	.146	-0.00422590	-0.00783438	-0.00777064
45	.145	-0.00906280	-0.01054676	-0.00912941
46	.144	-0.00896075	-0.00800904	-0.00534663
47	.143	-0.00384322	-0.00146844	0.00169373
48	.142	0.00355726	0.00559001	0.00827437
49	.141	0.00915776	0.00930460	0.01082544
50	.140	0.00977959	0.00755538	0.00787800
51	.139	0.00495278	0.00117679	0.00092058
52	.138	-0.00275433	-0.00642823	-0.00633811
53	.137	-0.00910110	-0.01109930	-0.00993729
54	.136	-0.01051032	-0.01019079	-0.00781773
55	.135	-0.00608563	-0.00406262	-0.00098870
56	.134	0.00182774	0.00404976	0.00695218
57	.133	0.00888529	0.00975126	0.01172143
58	.132	0.01113103	0.00987817	0.01066762
59	.131	0.00721696	0.00426206	0.00424921
60	.130	-0.00079214	-0.00411744	-0.00415148
61	.129	-0.00850696	-0.01070732	-0.01001026
62	.128	-0.01162197	-0.01185090	-0.01009953
63	.127	-0.00832094	-0.00681114	-0.00426700
64	.126	-0.00033418	0.00177297	0.00442620
65	.125	0.00796711	0.00927543	0.01131279
66	.124	0.01196617	0.01158647	0.01262745
67	.123	0.00937146	0.00737345	0.00756615
68	.122	0.00152971	-0.00114568	-0.00123066
69	.121	-0.00727126	-0.00936419	-0.00905861
70	.120	-0.01215013	-0.01277482	-0.01167130
71	.119	-0.01034301	-0.00943877	-0.00758965
72	.118	-0.00277035	-0.00108116	0.00105890
73	.117	0.00642937	0.00780603	0.00964686
74	.116	0.01216432	0.01238706	0.01352330
75	.115	0.01121147	0.01012282	0.01053303
76	.114	0.00403003	0.00218458	0.00221203
77	.113	-0.00545565	-0.00714667	-0.00700434
78	.112	-0.01200350	-0.01278784	-0.01215228
79	.111	-0.01195487	-0.01162254	-0.01042032
80	.110	-0.00528157	-0.00422070	-0.00268974
81	.109	0.00436821	0.00544118	0.00691014
82	.108	0.01166711	0.01212306	0.01320832
83	.107	0.01255415	0.01217109	0.01277911
84	.106	0.00649742	0.00552171	0.00580550
85	.105	-0.00318853	-0.00424063	-0.00399621
86	.104	-0.01115922	-0.01181607	-0.01136230
87	.103	-0.01299377	-0.01306882	-0.01231513
88	.102	-0.00765056	-0.00728638	-0.00631644
89	.101	0.00194089	0.00241738	0.00342707
90	.100	0.01048855	0.01078341	0.01167914
91	.99	0.01326225	0.01326375	0.01399124
92	.98	0.00871533	0.00849912	0.00910344
93	.97	-0.00065164	-0.00093400	-0.00036901
94	.96	-0.00966821	-0.00991814	-0.00933805
95		0.93533674	0.93511433	0.93570836

Tab. 5. Coefficients of the Impulse Responses $h(k)$.

6. Example of the Design and Tuning

Let us design a notch FIR filter specified by the notch frequency $f_0 = 60$ Hz, the width of the notch band $\Delta f = 6$ Hz, the sampling frequency $f_s = 500$ Hz and the attenuation in the passbands $a = -1$ dB. For a demonstration of the tuning, let us tune its notch frequency to 59.7 Hz.

We start with the design of the initial notch FIR filter. The function `[n,h,f_m,adB_pass,adB_m]=firmotch(60,6,500,-1)` in Tab. 3 returns the impulse response $h(k)$ of the initial notch FIR filter 191 coefficients in length, actual notch frequency $f_m = 60.5565$ Hz, attenuation in the passbands -0.94 dB and attenuation at the notch frequency -301.62 dB. We continue with the tuning of its notch frequency f_m to the specified value $f_0 = 60$ Hz. The function `h_t=tune(h, 60.5565, 60, 500)` in Tab. 4 returns the impulse response $h_t(k)$ with the attenuation of -302.92 dB at notch frequency 60 Hz. The corresponding amplitude frequency response is shown in Fig. 3 and a detailed view of its notch band is shown in Fig. 4. Finally, we tune the notch frequency f_m of the initial filter to the specified value 59.7 Hz using `h_t=tune(h, 60.5565, 59.7, 500)`. Its attenuation amounts to -301.85 dB at notch frequency 59.7 Hz. The notch bands of the particular filters are shown in Fig. 5. For checking, the impulse responses of the notch FIR filters are summarized in Tab. 5.

7. Conclusions

We have presented a fast and robust procedure for the design and tuning of narrow band notch FIR filters. The proposed procedure is suitable for real time notch filtering of unwanted power line interference in various signals. We have included two Matlab functions which prove the speed and robustness of the presented procedure. One example demonstrates the design and tuning of the notch FIR filter.

Acknowledgements

The authors wish to thank the Grant Agency of the Czech Republic for supporting this research through project P-102/11/ 1795: Novel selective transforms for non-stationary signal processing.

References

- [1] MARTENS, S. M. M., MISCHI, M., OEI, S. G., BERGMANS, J. W. M. An improved adaptive power line interference canceller for electrocardiography. *IEEE Transactions on Biomedical Engineering*, 2006, vol. 53, no. 11, p. 2220 - 2231.
- [2] CHAVAN, M. S., AGARWALA, R., UPLANE, M. D. Design and implementation of digital FIR equiripple

notch filter on ECG signal for removal of power line interference. *WSEAS Transactions on Signal Processing*, 2008, vol. 4, no. 4, p. 221 - 230.

- [3] OLGUIN, D. O., LARA, F. B., CHAPA, S. O. M. Adaptive notch filter for EEG signals based on the LMS algorithm with variable step-size parameter. In *Proceedings of the 39th International Conference on Information Sciences and Systems*. Baltimore (USA), 2005.
- [4] LEE, J. W., LEE, G. K. Design of an adaptive filter with a dynamic structure for ECG signal processing. *International Journal of Control, Automation and Systems*, 2005, vol. 3, no. 1, p. 137 - 142.
- [5] KUMAR, Y., MALIK, G. K. Performance analysis of different filters for power line interference reduction in ECG signal. *International Journal of Computer Applications*, 2010, vol. 3, no. 7, p. 1 - 6.
- [6] FERDJALLAH, M., BARR, R. E. Adaptive digital notch filter design on the unit circle for the removal of powerline noise from biomedical signals. *IEEE Transactions on Biomedical Engineering*, 1994, vol. 41, no. 6, p. 529 - 536.
- [7] ZAHRADNIK, P., VLČEK, M. Fast analytical design algorithms for FIR notch filters. *IEEE Transactions Circuits and Systems I: Regular Papers*, 2004, vol. 51, no. 3, p. 608 - 623.
- [8] ZAHRADNIK, P., VLČEK, M. An analytical procedure for critical frequency tuning of FIR filters. *IEEE Transactions Circuits and Systems II-Express Briefs*, 2006, vol. 53, no. 1, p. 72 - 76.
- [9] VLČEK, M., UNBEHAUEN, R. Zolotarev polynomials and optimal FIR filters. *IEEE Transactions on Signal Processing*, 1999, vol. 47, no. 3, p. 717 - 730.
- [10] ABRAMOWITZ, M., STEGUN, I. *Handbook of Mathematical Functions*. New York (USA): Dover Publications, 1972.

About Authors ...

Pavel ZAHRADNIK was born in Melnik. He received his M.Sc. in Microelectronics from the Czech Technical University in Prague in 1986 and PhD degree in Communication engineering from the Czech Technical University in Prague in 1991. He is a full professor at the Czech Technical University in Prague. As an Alexander von Humboldt Fellow, he spent two years at the University Erlangen-Nürnberg, Germany. He is author or co-author of more than 100 papers. His research interests include approximation of digital filters, digital signal, image and video processing.

Miroslav VLČEK was born in Prague. He studied physics at Charles University in Prague. He has DSc degree in Telecommunications from the Czech Technical University in

Prague where he is a full professor and head of the department of Applied mathematics at the Faculty of Transportation Sciences. In 1978 he spent a year at Queen Mary College, London University. Then as an Alexander von Humboldt Fellow, he spent a total of almost two years at the University Erlangen-Nürnberg Germany. In 1998, he spent two

months in the USA on an Eisenhower Fellowship observing university management. Between 2000-2010 he has been vice-rector for international relations. His research interest is in digital signal processing, higher transcendental functions, and digital filter approximations. He is author/co-author of more than 120 papers and 2 monographs.