SKAITMENINIS SIGNALŲ APDOROJIMAS IR MAŠININIS MOKYMASIS 2022
Laboratorinis darbas nr. 2

SKAITMENINIŲ FILTRŲ TYRIMAS

Ž. Marma, E MEI-2 gr. Dėstytojas D. Sokas
KTU, Elektros ir elektronikos fakultetas

Įvadas

Laboratorinio darbo tikslas – išmokti įgyvendinti ir tirti skaitmeninių filtrų sistemą sprendžiant elektrokardiografinių signalų apdorojimo problemą.

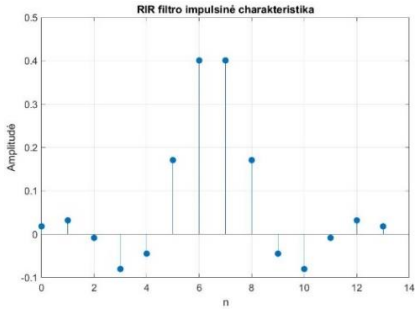
Laboratorinio darbo užduotis – suprojektuoti skaitmeninius filtrus ir jais apdoroti elektrokardiogramos signalus. Laboratoriniam darbui realizuoti buvo naudojamas 13 EKG signalas.

Skaitmeninio RIR filtro įgyvendinimas ir tyrimas

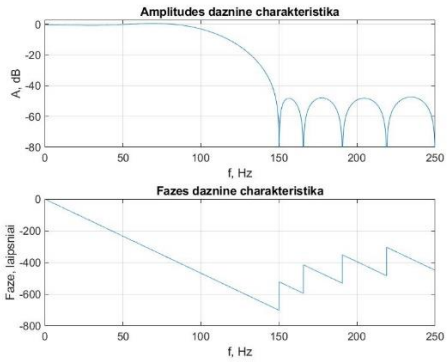
Žemųjų dažnių RIR filtro skirtuminė lygtis (1) gauta iš duotos struktūrinės schemos

$$y[n] = 0{,}0181x[n] + 0{,}0319x[n-1] - 0{,}0084x[n-2] - 0{,}0803x[n-3] - 0{,}0449x[n-4] + 0{,}1709x[n-5] + 0{,}408x[n-6] + 0{,}408x[n-7] + 0{,}1709x[n-8] - 0{,}0499x[n-9] - 0{,}0803x[n-10] - 0{,}0084x[n-11] + 0{,}0319x[n-12] + 0{,}0181x[n-13]. \quad (1)$$

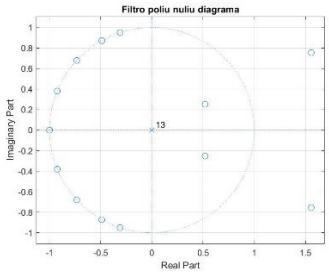
Toliau buvo rastos RIR filtro charakteristikos: impulsinė charakteristika (1 pav.), amplitudės ir fazės dažninės charakteristikos (2 pav.) ir nulių-polių diagrama (3 pav.).



1 pav. RIR filtro impulsinė charakteristika.



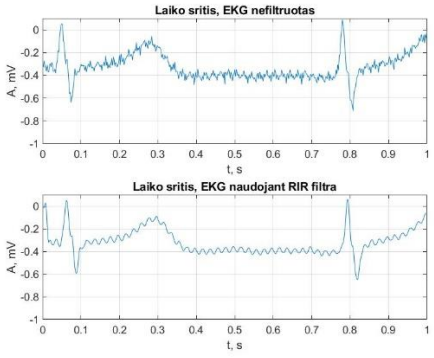
2 pav. RIR filtro amplitudės ir fazės dažninės charakteristikos.



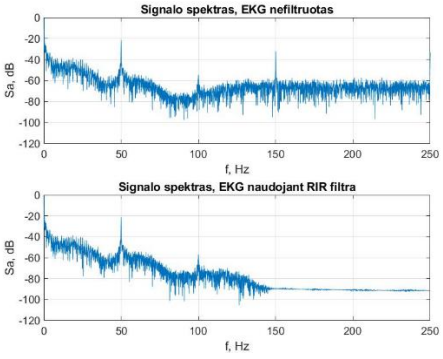
3 pav. RIR filtro nulių-polių diagrama.

Išanalizavus filtro impulsinę charakteristiką (1 pav.) matyti, jog sumodeliuotas filtras yra stabilus, nes į sistemą įvedus vienetinį šuolį filtro atsakas nusistovi. Remiantis amplitudės dažnio charakteristika (2 pav.) filtro pralaidumo juosta yra šiek mažesnė nei 100 Hz. Nulių-polių diagramoje (3 pav.) pastebima, kad nuliai, esantys ant vienetinio apskritimo rodo dažnius ties kuriais signalas yra labiausiai silpninamas. Pats filtras yra stabilus, nes poliai yra vienetinio apskritimo viduje.

Suprojektuoto RIR filtro parametrai: filtro pjūvio dažnis yra 100 Hz, pereinamoji juosta ties 100 Hz – 150 Hz, maksimalus filtro pralaidumo juostos bangavimas lygus 1,27 dB (0,61dB - (-0,668dB)), minimalus slopinimas filtro slopinimo dažnių juostoje lygus 47,34dB. Filtre esantis filtro pralaidumo juostos bangavimas lems amplitudės iškraipymus pralaidumo zonoje – vienus dažnius neženkliai stiprins, o kitus neženkliai silpnins.



4 pav. EKG signalo dalis laiko srityje naudojant RIR filtrą.



5 pav. EKG signalas dažnių srityje naudojant RIR filtrą.

Įgyvendinus RIR filtrą laiko srityje (4 pav.) galima matyti švaresnį EKG signalą (aukšto dažnio dedamųjų panaikinimas). Analizuojant signalą dažnių srityje (5 pav.) akivaizdžiai galima matyti stiprus slopinimas nuo 150 Hz. Tačiau signale vis dar lieka elektros tinklo dažnio (50 Hz) dedamoji ir jos harmonikos.

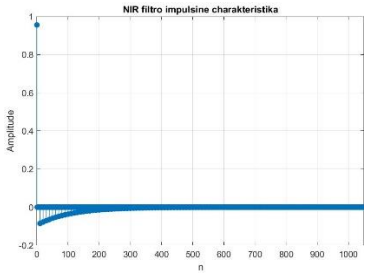
Skaitmeninio NIR filtro įgyvendinimas ir tyrimas

Naudojantis laboratorinio darbo apraše esančiomis formulėmis buvo rasti filtrui projektuoti reikalingi koeficientai ir pateikiamos skirtuminė (2) ir perdavimo lygtys (3):

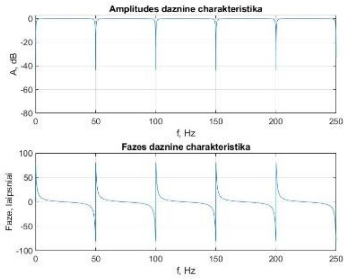
$$y[n] = k_1 x[n] - k_2 x[n-10] + k_3 x[n-10] = 0{,}9554x[n] - 0{,}9548x[n-10] + 0{,}9631x[n-10], \quad (2)$$

$$H(z) = \frac{k_1 - k_2 z^{-N}}{1 - k_3 z^{-N}} = \frac{0{,}9554 - 0{,}9548 \cdot z^{-10}}{1 - 0{,}9631 \cdot z^{-10}}, \quad (3)$$

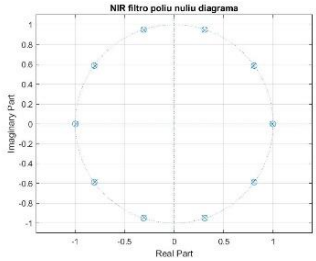
Toliau buvo rastos NIR filtro charakteristikos: impulsinė charakteristika (6 pav.), amplitudės ir fazės dažninės charakteristikos (7 pav.) ir nulių-polių diagrama (8 pav.).



6 pav. NIR filtro impulsinė charakteristika.

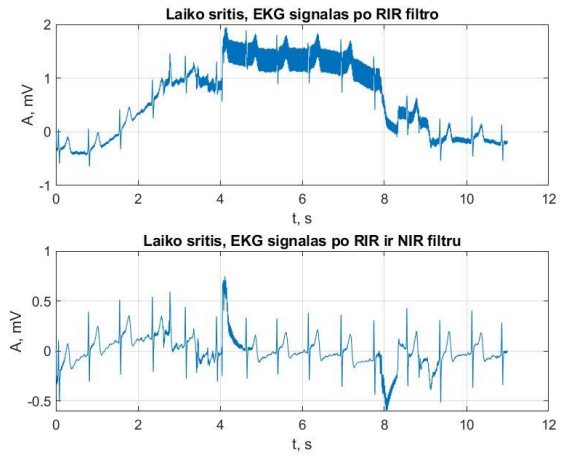


7 pav. NIR filtro amplitudės ir fazės dažninės charakteristikos.



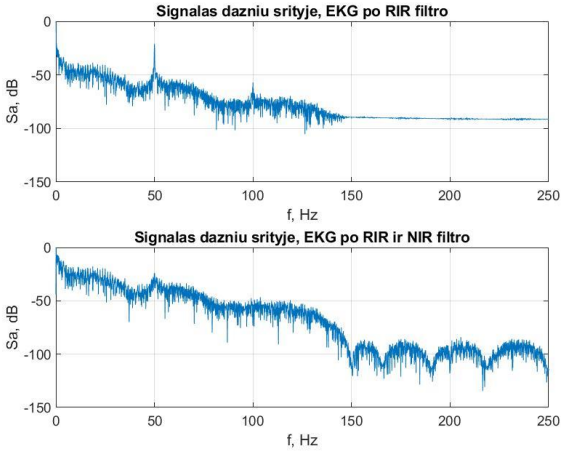
8 pav. NIR filtro nulių polių diagrama.

Projektuojant NIR tipo filtrą slopinamų dažnių juostos plotis Δf buvo pasirinktas 1,5 Hz, filtro slopinimo reikšmė S išpjovos centre buvo pasirinkta 150 kartų t.y. 43,52 dB. Analizuojant NIR filtro impulsinę charakteristiką (6 pav.) akivaizdu, kad sukurtas filtras yra stabilus, nes filtro atsakai į vienetinį šuolį artėja link nulio. Amplitudės ir fazės charakteristikoje (7 pav.) galima matyti, kad yra slopinami 50 Hz ir jo harmonikų (100 Hz, 150 Hz ... n*50 Hz) signalai. Būtent taip yra pašalinamas elektros tinklo dažnio sukuriamas triukšmas iš signalo. Polių nulių diagramoje (8 pav.) galima matyti, kad poliai yra vienetinio apskritimo viduje (labai neženkliai), todėl filtras stabilus.



9 pav. EKG signalas laiko srityje po RIR apdorojimo ir EKG signalas po RIR ir NIR filtrų apdorojimo.

Iš EKG signalo laiko srityje grafiko (9 pav.) galima matyti, jog iš EKG signalo yra pašalinamas dreifas ir dar stipriau slopinamas aukšto dažnio triukšmas, likęs po RIR filtro. Būtent tokios formos signalas yra tinkamesnis medicinos darbuotojams, analizuojantiems sveikatos parametrus iš EKG signalų.



10 pav. EKG signalas dažnių srityje po RIR apdorojimo ir EKG signalas po RIR ir NIR filtrų apdorojimo.

Analizavus EKG signalą dažnių srityje po NIR filtro (10 pav.) galima pastebėti nuslopinimą ties 50 Hz ir 100 Hz dažniu. Nors nuo 150 Hz dažnių signalo dedamosios yra slopinamos RIR filtro galima pastebėti dar padidėjusius 50 Hz signalo harmonikų slopinimus.

Daugiasparčio RIR filtro įgyvendinimas

Toliau darbe buvo analizuojamas daugiaspartis filtras kurį galima būtų naudoti vietoj šukų tipo NIR filtro. Šis filtras susideda iš dviejų nuosekliai sujungtų decimatorių, pagrindinio žemųjų dažnių filtro ir dviejų interpoliatorių. Kuriant šį filtrą projektuojamo žemųjų dažnių filtrui buvo pasirinktas 1 Hz slopinimo juostos ribinis dažnis ($f_{sl} = 1\ Hz$), o pralaidumo juostos ribinis dažnis parinktas 0,4 Hz

($f_{pr} = 0{,}4\ Hz$). Toliau buvo surastas M koeficientas pagal lygtį (4)

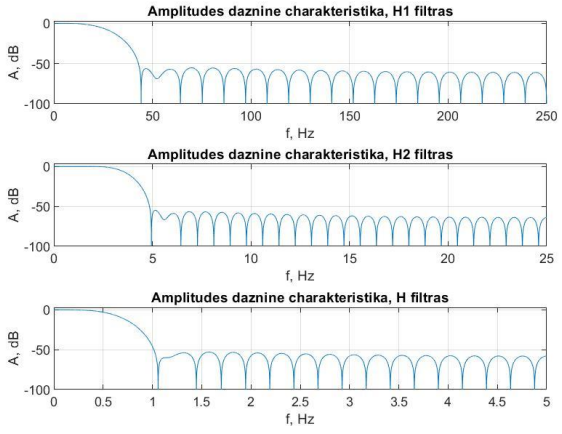
$$\left({f_{sl}}^2 - {f_{pr}}^2\right)M^3 - \left(f_{sl} - f_{pr}\right)^2 M^2 + 2f_d\left(f_{sl} + f_{pr}\right)M - {f_d}^2 = 0\,. \quad (4)$$

Išsprendus šią lygtį gauti trys sprendiniai, iš kurių tik vienas realusis rodantis, kad M yra lygus 55,1. Tai įvertinus M buvo pasirinktas 50. Toliau buvo apskaičiuoti optimalūs decimacijos koeficientai D_{1opt} ir D_{2opt} pagal pateiktas formules (5, 6)

$$D_{1,opt} \approx \frac{2M\left(1-\sqrt{\frac{MF}{2-F}}\right.}{2-F(M+1)}, \quad (5)$$

$$D_{2,opt} = \frac{M}{D_{1,opt}}. \quad (6)$$

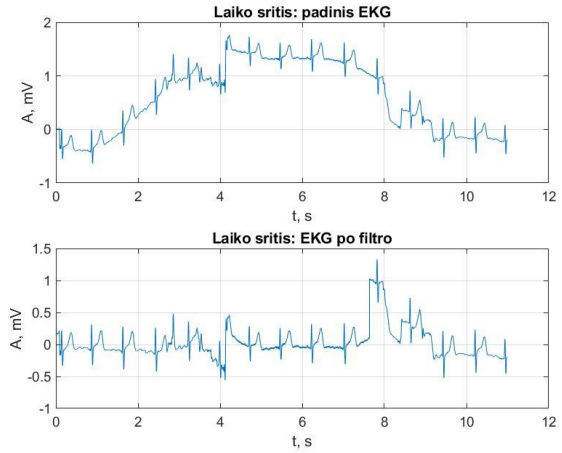
Gautos reikšmės D_{1op} = 12,9 ir D_{2op} =3,9. Kadangi šių koeficientų sandauga turi būti lygi M, buvo parinktos atitinkamai 10 ir 5 koeficientų D_1 ir D_2 reikšmės. Daugiaspartis filtras buvo įgyvendintas atliekant dažnio mažinimo operacijas su MATLAB funkcija `downsample` ir papildomai sukurtu apsauginiu filtru. Tuo tarpu dažnio didinimo operacijos buvo atliktos su MATLAB funkcija `upsample` ir tais pačiais apsauginiais filtrais. Apsauginiai filtrai buvo suprojektuoti siekiant atkartoti `decimate` funkcijoje esantį apsaugos efektą. Todėl pirmajame filtre buvo pasirinkta 45 filtro eilė, antrajame 60, o ŽDF 40. Visi šie RIR filtrai buvo sukurti naudojantis `fir1` funkciją. Kadangi visų (H(z), H_1(z) ir H_2(z)) suprojektuotų filtrų slopinimas slopinimo juostoje turi būti bent -50 dB yra pateikiama šių filtrų amplitudės dažninės charakteristikos (11 pav.).



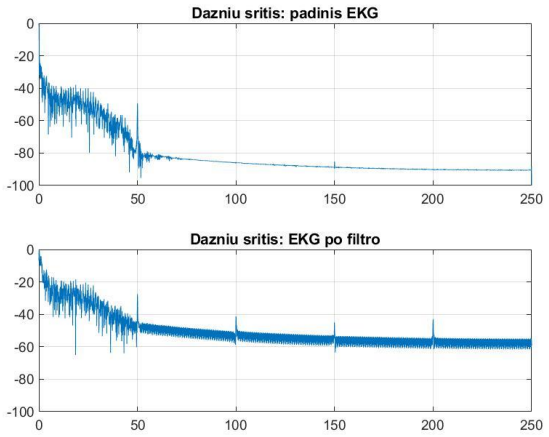
11 pav. RIR filtrų esančių daugiasparčiame filtre amplitudžių dažninės charakteristikos

Kaip galima matyti grafike (11 pav.) visi trys filtrai slopinimo juostoje signalą slopina bent 50 dB, todėl galima teigti, kad yra suprojektuoti tinkamai.

Toliau atliekant suprojektuoto filtro analizę prieš apdorojant EKG signalą su daugiasparčiu filtru, signalas buvo apdorotas su žemų dažniu filtru. Šio pradinio filtro analizę nebus atliekama, tačiau jis buvo sukurtas su MATLAB „Filter Designer“ programa ir skirtas pašalinti signale esančias didesnes nei 50 Hz dedamąsias (filtro pjūvio dažnis yra 47,5 Hz). Taip pat buvo įvertintas vėlinimas kurį sukuria daugiaspartis filtras, jis buvo lygus 1681 laiko atsakų (šiek tiek virš 3 sekundžių). Šis vėlinimas buvo kompensuotas į signalą įvedus nulių. Gautas nufiltruotas EKG signalas su daugiasaičiu filtru pateiktas laiko srityje (12 pav.) ir dažnių srityje (13 pav.).



12 pav. EKG signalas laiko srityje apdorojus daugiasparčiu filtru.

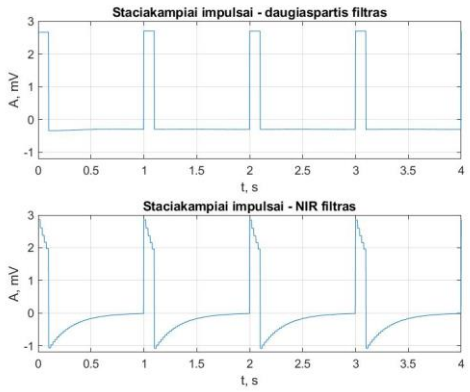


13 pav. EKG signalas dažnių srityje apdorojus daugiasparčiu filtru.

Iš EKG signalo laiko srities grafiko (12 pav.) galima matyti, jog iš signalo buvo pašalintas dreifas. Tik vienoje vietoje (ties 8 sekundėmis) filtras veikė ne visai teisingai. Analizuojant filtro spektrą (13 pav.), kad aukštų dažnių dedamosios yra mažiau slopinamos ir galima pastebėti nežymias 50 Hz signalo dedamąsias.

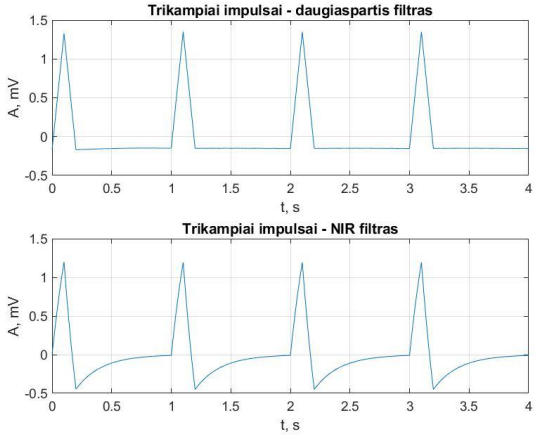
Elektrokardiogramos apdorojimo filtrų testavimas

Toliau sukurtas daugiaspartis filtras buvo testuojama registruojamus reaguojant į stačiakampius (14 pav.) ir trikampius impulsus (15 pav.). Filtro rezultatai yra palyginami su NIR šukų tipo filtru.



14 pav. Daugiasparčio ir NIR filtrų testavimas su stačiakampiais impulsais.

Testuojant filtrus su stačiakampiu signalu galima (14 pav.) matyti, jog suprojektuoto daugiasparčio filtro maksimalus nuokrypis (0,35 mV) yra mažesnis, nei NIR filtro (1,05 mV). Tačiau pagal EN 60601-2-51 standarte pateikiamus nurodymus atlikus šį testą nuokrypis neturėtų viršyti 0,1 mV todėl suprojektuotas filtras neatitiktų reikalavimų.



15 pav. Daugiasparčio ir NIR filtrų testavimas su trikampiais impulsais.

Testuojant filtrus su trikampiais impulsais yra įvertinamas amplitudės sumažėjimas, kuris neturėtų būti didesnis nei 12 %. Šio testo metu vėl geresnis atrodė daugiaspartis filtras (15 pav.). NIR tipo šukų filtro amplitudė sumažėjo 20,1 % (nuo 1,5 mV iki 1,19 mV), o daugiasparčio filtro amplitudė sumažėjo 10,1 % (nuo 1,5 mV iki 1,34 mV).

Rezultatai

Laboratoriniame darbe buvo įgyvendinti ir ištirti skaitmeniniai filtrai sprendžiant elektrokardiografinių signalų apdorojimo problemą. Buvo suprojektuoti filtrai skirti apdoroti elektrokardiogramos signalus – pašalinti aukštų dažnių triukšmą, elektros tinklo sukuriamas pašalines 50 Hz harmonikų dedamąsias ir pašalinti signale esantį dreifą. MATLAB programa buvo realizuotas RIR filtras, pašalintais aukštų dažnių triukšmą. Suprojektuotas NIR šukų tipo filtras skirtas pašalinti bazinės linijos dreifą bei pramoninio įtampos tinklo dedamąją, įskaitant ir jos kartotines harmonikas. Taip pat darbe sukurtas daugiaspartis filtras kuris savo veikimu yra panašus į NIR filtrą tačiau naudoja mažiau skaičiavimo resursų.

Diskusija

Sukūrus daugiaspartį filtrą iš EKG signalo dreifas yra sėkmingai pašalinamas (12 pav.), tačiau iš signalo spektro (13 pav.) galima matyti, kad nėra pilnai pašalinama elektros tinklo 50 Hz dedamoji ir jos harmonikos. Tai rodo, kad NIR šukų tipo filtras yra paprastesnis ir efektyvesnis. Šie daugiasparčiai filtrai turėtų būti kuriami tik jeigu filtravimo reikia energiją taupančioje įterptinėje sistemoje. Tuo tarpu sistemoje turinčioje skaičiavimo resursus prioritetas turėtų būti teikiamas NIR šukų tipo filtrams.

Testuojant daugiaspartį filtrą su stačiakampiais ir trikampiais impulsais, abejais atvejais signalas žymiai ilgesnį laiko tarpą išbuvo neigiamos amplitudės zonoje. Tai galimai signalizuoja neteisingą veikimą ir norint šį filtrą naudoti medicinos srityje reikėtų detalesnės analizės.

Išvados

Laboratoriniame darbe buvo atliktas EKG signalo filtravimas pašalinant nenaudingas dedamąsias. Darbe sukurti ir išanalizuoti skaitmeniniai filtrai reikalingi filtravimui atlikti.

Priedai

Pagrindinės užduoties MATLAB programos kodas:

```
clc, clear, close all

set(0,'defaultAxesFontName','TimesNewRoman')
set(0,'defaultAxesFontSize',10)

load("signalai/EKG_13")
fd = 500; %add _Hz            % Diskretizavimo
freq
ts = 11;
t =(0:(ts*fd)-1)*1/fd;


%% 3.3.1 RIR filtras

% Filtro koeficientai
b = [0.0181, 0.0319, -0.0084, -0.0803, -0.0449,
0.1709, 0.4008, 0.4008, ...
    0.1709, -0.0449, -0.0803, -0.0084, 0.0319,
0.0181];
a = zeros(1,length(b));
a(1) = 1;

ekg_filtered = filter(b, a, ekg);
figure();
plot(t, ekg_filtered)

%% 3.3.2
figure();
impz(b, a);
ylabel("Amplitude"); xlabel('n'); title('RIR
filtro impulsine charakteristika'); xlim([0 14]);
grid on;
f_saveFig("3.3.2-RirImpls")

%% 3.3.3
f_plotFreqz(b, a, fd, "3.3.3-amplitFazesChar")


%% 3.3.4
```

```
figure()
zplane(b,a)
grid on; title('Filtro poliu nuliu diagrama');
f_saveFig("3.3.4-RirZeroPole")

%% 3.3.5 - plot for time
figure()
subplot(211);
plot(t, ekg);
xlabel('t, s'); ylabel('A, mV');
grid on; title('Laiko sritis, EKG nefiltruotas');

subplot(212);
plot(t, ekg_filtered);
xlabel('t, s'); ylabel('A, mV');
grid on; title('Laiko sritis, EKG naudojant RIR
filtra');
f_saveFig("3.3.5-EkgCompare", true);

%% plot freq: plot signal in freqency domain
figure()

nfft = length(ekg);
ekg_freq = abs(fft(ekg))/nfft;
ekg_freq = 20*log10(ekg_freq/max(ekg_freq));
k = 0:1:nfft-1;
f = k*fd/nfft;

subplot(211)
plot(f, ekg_freq);
xlabel('f, Hz'); ylabel('Sa, dB'); xlim([0
fd/2]); ylim([-120 0]);
grid on; title('Signalo spektras, EKG
nefiltruotas');

ekgFilter_freq = abs(fft(ekg_filtered))/nfft;
ekgFilter_freq =
20*log10(ekgFilter_freq/max(ekgFilter_freq));
subplot(212)
plot(f, ekgFilter_freq);
xlabel('f, Hz'); ylabel('Sa, dB'); xlim([0
fd/2]); ylim([-120 0]);
grid on; title('Signalo spektras, EKG naudojant
RIR filtra');
f_saveFig("3.3.5-EkgCompareFreq", true);

%% 3.4 NIR Filtras

fDelta_Hz = 1.5;

% constans:
f0_Hz = 50;
K0 = 1;
L = 3;
S = mag2db(150);

%3.4.2
K = K0 * 10^(-S/20);
Kr = K0 * 10^(-L/20);

%3.4.3
N = fd/f0_Hz;

%3.4.4
beta = sqrt((Kr^2 - K0^2)/(K^2 - Kr^2)) *
tan((N*pi*fDelta_Hz) / (2*fd));

k1 = (K0 + (K*beta)) / (1 + beta);
k2 = (K0 - (K*beta)) / (1 + beta);
k3 = (1 - beta) / (1 + beta);


bNIR = [k1, zeros(1,N-1), -k2];
aNIR = [1, zeros(1,N-1), -k3];

%3.4.6
ekg_afterNIR = filter(bNIR, aNIR, ekg_filtered);

% analyzsis
%3.4.7
figure();
impz(bNIR, aNIR);
ylabel("Amplitude"); xlabel('n'); title('NIR
filtro impulsine charakteristika'); grid on;
f_saveFig("3.4.7-NirImpls")

%3.4.8
f_plotFreqz(bNIR, aNIR, fd, "3.4.8-
amplitFazesCharNIR")

% 3.4.9
figure();
zplane(bNIR, aNIR)
grid on; title('NIR filtro poliu nuliu
diagrama');
f_saveFig("3.4.9-NirZeroPole")

% 3.4.10 - plot for time
figure();
subplot(211);
plot(t, ekg_filtered);
xlabel('t, s'); ylabel('A, mV');
grid on; title('Laiko sritis, EKG signalas po RIR
filtro');

subplot(212);
plot(t, ekg_afterNIR);
xlabel('t, s'); ylabel('A, mV');
grid on; title('Laiko sritis, EKG signalas po RIR
ir NIR filtru');
f_saveFig("3.4.10-EkgCompare", true)

% plot freq: plot signal in freqency domain
% (Same as in 1 Lab) - use function
figure();

nfft = length(ekg_filtered);
ekg_freq = abs(fft(ekg_filtered))/nfft;
ekg_freq = 20*log10(ekg_freq/max(ekg_freq));
k = 0:1:nfft-1;
f = k*fd/nfft;

subplot(211)
plot(f, ekg_freq);
xlabel('f, Hz'); ylabel('Sa, dB'); xlim([0
fd/2]); ylim([-150 0]);
grid on; title('Signalas dazniu srityje, EKG po
RIR filtro')

ekg_afterNIR_fq = abs(fft(ekg_afterNIR))/nfft;
ekg_afterNIR_fq =
20*log10(ekg_afterNIR_fq/max(ekg_afterNIR_fq));
subplot(212)
plot(f, ekg_afterNIR_fq);
xlabel('f, Hz'); ylabel('Sa, dB'); xlim([0
fd/2]); ylim([-150 0]);
grid on; title('Signalas dazniu srityje, EKG po
RIR ir NIR filtro')
f_saveFig("3.4.10-EkgCompareFreq", true)

figure();
f_plotAllSignalsTime(t, ekg, ekg_filtered,
ekg_afterNIR)


figure();
f_plotAllSignalsFreq(ekg, ekg_filtered,
ekg_afterNIR, fd)

function f_saveFig(figName, usingSubplots)
    if ~exist('usingSubplots','var')
      usingSubplots = false; % third parameter
does not exist, so default it to something
    end
```

```
    set(gca, 'units', 'normalized'); %Just making
sure it's normalized
    Tight = get(gca, 'TightInset');  %Gives you
the bording spacing between plot box and any axis
labels
                                     %[Left
Bottom Right Top] spacing
    NewPos = [Tight(1) Tight(2) 1-Tight(1)-
Tight(3) 1-Tight(2)-Tight(4)]; %New plot position
[X Y W H]
    if usingSubplots == false % matlab fucks up
subplots
        set(gca, 'Position', NewPos);
    end
    saveas(gca,"outFigs/"+figName+".jpg");
end

function f_plotAllSignalsTime(t, initial, RIR,
NIR)
    subplot(311)
    plot(t, initial);
    title('Laiko sritis neapdoroto ekg');
    subplot(312);
    plot(t, RIR);
    title('Laiko sritis po RIR');

    subplot(313);
    plot(t, NIR);
    title('Laiko sritis po RIR ir NIR');
end

function f_plotAllSignalsFreq(initial, RIR, NIR,
fd)
    subplot(311)
    [x1, y1] = getFreqOfSignal(initial, fd);
    plot(x1, y1);
    title('Dazniu sritis neapdoroto ekg');
    xlim([0 fd/2])

    subplot(312);
    [x2, y2] = getFreqOfSignal(RIR, fd);
    plot(x2, y2);
    title('Dazniu sritis po RIR');
    xlim([0 fd/2])

    subplot(313);
    [x3, y3] = getFreqOfSignal(NIR, fd);
    plot(x3, y3);
    title('Dazniu sritis po RIR ir NIR');
    xlim([0 fd/2])
end

function [x_fq, y_fq] = getFreqOfSignal(sig, fd)
    nfft = length(sig);
    sig_fq = abs(fft(sig))/nfft;
    sig_fq = 20*log10(sig_fq/max(sig_fq));
    k = 0:1:nfft-1;

    x_fq = k*fd/nfft;
    y_fq = sig_fq;
end

function f_plotFreqz(b, a, fd, figName)
    n = 15000;
    figure();
    [h_fq,w_fq] = freqz(b, a, n, fd);

    subplot(211)
    plot(w_fq, 20*log10(abs(h_fq)))
    xlabel('f, Hz'); ylabel('A, dB'); ylim([-80
3]);
    title('Amplitudes daznine charakteristika');
grid on

    subplot(212)
    plot(w_fq, 360/(2*pi)*unwrap(angle(h_fq)))
    xlabel('f, Hz'); ylabel('Faze, laipsniai')
    title('Fazes daznine charakteristika'); grid
on
    f_saveFig(figName, true);
end
```

Papildomos užduoties daugiasparčio filtro kūrimo MATLAB programos kodas:

```
%%
clc, clear, close all;
set(0,'defaultAxesFontName','TimesNewRoman')
set(0,'defaultAxesFontSize',10)

load("signalai/EKG_13")

f_SL_Hz = 1; % projektuojamo zemu dazniu filtro
slopinimo juostos ribinis daznis
f_pr_Hz = 0.4; % pralaidumo juostos ribinis
daznis
f_d_Hz = 500;

time_s = 11;
time_n =(0:(time_s*f_d_Hz)-1)*1/f_d_Hz;

% Solve equation to find M
syms x
eqn = (f_SL_Hz^2)*x^3 - (f_pr_Hz^2 -
((f_SL_Hz+f_pr_Hz)^2))*x^2 +
(2*f_d_Hz*(f_SL_Hz+f_pr_Hz))*x -f_d_Hz^2 == 0;
M_opt = vpasolve(eqn, x);

M = 50; % Decimacijos koeficientas parenkamas
toks, kad diskretizavimo daznio ir koeficiento M
dalybos
        % rezultatas buti sveikas skaicius

F = (f_SL_Hz - f_pr_Hz)/f_SL_Hz;

D1_s = (2*M*(1 - sqrt( ((M*F) / (2-F)))));
D1_d = 2 - F*(M*1);

D1_opt = D1_s / D1_d;

D2_opt = M/D1_opt;

D1 = 10;
D2 = 5;
if D1*D2 ~= M
    disp("error");
end

%% pries paduodant naudoiti ta pati filtra kai p1
uzduotyje
Fs_low = 500;  % Sampling Frequency

N_low = 90;  % Order
Fpass = 45;  % Passband Frequency
Fstop = 47;  % Stopband Frequency
Wpass = 1;   % Passband Weight
Wstop = 10;   % Stopband Weight

% Calculate the coefficients using the FIRLS
function.
[b_low, a_low] = firls(N_low, [0 Fpass Fstop
Fs_low/2]/(Fs_low/2), [1 1 0 0], [Wpass Wstop]);
ekg = filter(b_low, a_low, ekg);
freqz(b_low, 1, 15000, f_d_Hz);


% Decimation
if 0 % using matlab API
    ekg_1 = decimate(ekg, D1, 'fir');
    ekg_2 = decimate(ekg_1, D2, 'fir');
    current_FD = f_d_Hz / (D1 * D2)
else
```

```
    b_safety_H1 = fir1(45,(24.9/(f_d_Hz/2))); %
Naudojamas panasus filtras i decimate (filtro
eile = 30; fp() < (500 / 2*10)
    [b_H1_fqH, b_H1_fqW] =  freqz(b_safety_H1, 1,
15000, f_d_Hz);
    ekg_1 = filter(b_safety_H1, 1, ekg);
    ekg_1 = downsample(ekg_1, D1);
    current_FD = f_d_Hz / D1;

    b_safety_H2 = fir1(60,(3.5/(current_FD/2)));
% Naudojamas panasus filtras i decimate (filtro
eile = 30; fp() < (50 / 2*5)
    [b_H2_fqH, b_H2_fqW] =  freqz(b_safety_H2, 1,
15000, current_FD);
    ekg_2 = filter(b_safety_H2, 1, ekg_1);
    ekg_2 = downsample(ekg_2, D2);
    current_FD = current_FD / D2;
end

% Low-pass filtras
b_lowPass = fir1(40, (0.6/(current_FD/2)));
[B_H_fqH, B_H_fzW] =  freqz(b_lowPass, 1, 15000,
current_FD);
ekg_3 = filter(b_lowPass, 1, ekg_2);


if 0 % usisng MATLAB API
    ekg_4 = interp(ekg_3, D2);
    ekg_5 = interp(ekg_4, D1); % dreifas
else % make filter ourself
    ekg_4 = upsample(ekg_3, D2);
    ekg_4 = filter(b_safety_H2, 1, ekg_4);
    ekg_4 = ekg_4 * D2;

    ekg_5 = upsample(ekg_4, D1);
    ekg_5 = filter(b_safety_H1, 1, ekg_5);
    ekg_5 = ekg_5 * D1;

end
ekg_6 = ekg - ekg_5;

% check this API:
% grpdelay()


figure() % Plot filters magnitute responces
subplot(311);
plot(b_H1_fqW, 20*log10(abs(b_H1_fqH)))
xlabel('f, Hz'); ylabel('A, dB'); ylim([-100 3]);
grid on; title('Amplitudes daznine
charakteristika, H1 filtras');
subplot(312);
plot(b_H2_fqW, 20*log10(abs(b_H2_fqH)))
xlabel('f, Hz'); ylabel('A, dB'); ylim([-100 3]);
grid on; title('Amplitudes daznine
charakteristika, H2 filtras');
subplot(313);
plot(B_H_fzW, 20*log10(abs(B_H_fqH)))
xlabel('f, Hz'); ylabel('A, dB'); ylim([-100 3]);
grid on; title('Amplitudes daznine
charakteristika, H filtras');
saveas(gca,"outFigs/extra-H1-3-mag.jpg");


figure() % Plot EKG signals
subplot(311);
plot(time_n, ekg);
title('Laiko sritis: pradinis EKG');
xlabel('t, s'); ylabel('A, mV'); grid on;


subplot(312);
plot(time_n, ekg_5);
title('Laiko sritis: Dreifas');
xlabel('t, s'); ylabel('A, mV'); grid on;


subplot(313);
plot(time_n, ekg_6);
title('Laiko sritis: po multirat- RIR');
xlabel('t, s'); ylabel('A, mV'); grid on;


% delay of filter
velinimas = length(b_safety_H1) +
length(b_safety_H2)*D1 +
length(b_lowPass)*D1*D2/2


ekg_withZeros = [zeros(1, velinimas), ekg];
dreif_withZeros = [ekg_5, zeros(1, velinimas)];


ekg_noDreif = ekg_withZeros - dreif_withZeros;
figure()
subplot(211);
plot(time_n, ekg_withZeros(velinimas+1:end));
xlabel('t, s'); ylabel('A, mV');
grid on; title('Laiko sritis: padinis EKG');

subplot(212);
plot(time_n, ekg_noDreif(velinimas+1:end));
xlabel('t, s'); ylabel('A, mV');
grid on; title('Laiko sritis: EKG po filtro');
saveas(gca,"outFigs/extra-EkgAmpl.jpg");

% figure()
% plot(time_n, dreif_withZeros(velinimas+1:end));
% title('Laiko sritis: dreifas');
% xlabel('t, s'); ylabel('A, mV'); grid on;



figure()
subplot(211)
[x1, y1] =
f_getFreqOfSignal(ekg_withZeros(velinimas+1:end),
f_d_Hz);
plot(x1, y1); ylim([-100 0]);
title('Dazniu sritis: padinis EKG');  grid on;
xlim([0 f_d_Hz/2])

subplot(212);
[x2, y2] =
f_getFreqOfSignal(ekg_noDreif(velinimas+1:end),
f_d_Hz);
plot(x2, y2);
title('Dazniu sritis: EKG po filtro');  grid on;
xlim([0 f_d_Hz/2]); ylim([-100 0]);
saveas(gca,"outFigs/extra-Spekkt.jpg");



function [x_fq, y_fq] = f_getFreqOfSignal(sig,
fd)
    nfft = length(sig);
    sig_fq = abs(fft(sig))/nfft;
    sig_fq = 20*log10(sig_fq/max(sig_fq));
    k = 0:1:nfft-1;

    x_fq = k*fd/nfft;
    y_fq = sig_fq;
end
```

Papildomos užduoties daugiasparčio filtro testavimo MATLAB programos kodas:

```
clc, clear, close all;
set(0,'defaultAxesFontName','TimesNewRoman')
set(0,'defaultAxesFontSize',10)

fd = 500;

t_sec = 15;
```

```
t = 0: 1/fd: t_sec-1/fd;

% Test with square
t_sq = length(t)/t_sec/10; % 100ms
y_sq = zeros(1, fd);
for i = 1:t_sq
    y_sq(i) = 3;
end
y_sq = repmat(y_sq, 1, t_sec);
plot(t, y_sq)

[y_multi, vel] = f_getMultirate(y_sq, fd);

figure()
subplot(211)
plot(t, y_multi(vel+1:end))
xlabel('t, s'); ylabel('A, mV'); xlim([0 4]);
ylim([-1.2 3]);
grid on; title('Staciakampiai impulsai -
daugiaspartis filtras');

yNir = f_getNirFilter(y_sq, fd);
subplot(212)
plot(t, yNir)
xlabel('t, s'); ylabel('A, mV'); xlim([0 4]);
ylim([-1.2 3]);
grid on; title('Staciakampiai impulsai - NIR
filtras');
saveas(gca,"outFigs/5.test-Square.jpg");

%% Test wih triangle
fd = 500;
t_sec = 15;
t = 0: 1/fd: t_sec-1/fd;

t_triangle_sec = 0.202;
t_temp = 0: 1/fd: t_triangle_sec-1/fd;

for ii = 1:length(t_temp)
    if t_temp(ii) <= 0.1
        y_triangle(ii) = 15*t_temp(ii);
    else
        y_triangle(ii) = -15*t_temp(ii) + 3;
    end
end

y_triangle = [y_triangle, zeros(1, fd -
length(t_temp))];
y_triangle = repmat(y_triangle, 1, t_sec);
figure()
plot(t, y_triangle)

% get filters
[y_multi, vel] = f_getMultirate(y_triangle, fd);

figure()
subplot(211)
plot(t, y_multi(vel+1:end))
xlabel('t, s'); ylabel('A, mV'); xlim([0 4]);
ylim([-0.5 1.5]);
grid on; title('Trikampiai impulsai -
daugiaspartis filtras');

yNir = f_getNirFilter(y_triangle, fd);
subplot(212)
plot(t, yNir)
xlabel('t, s'); ylabel('A, mV'); xlim([0 4]);
ylim([-0.5 1.5]);
grid on; title('Trikampiai impulsai - NIR
filtras');
saveas(gca,"outFigs/5.test-Triangle.jpg");



function [out_y, velinimas] =
f_getMultirate(sig_y, f_d_Hz)
    D1 = 10
    D2 = 5

    % Decimation
    b_safety_H1 = fir1(45,(24.9/(f_d_Hz/2))); %
Naudojamas panasus filtras i decimate (filtro
eile = 30; fp() < (500 / 2*10)
    ekg_1 = filter(b_safety_H1, 1, sig_y);
    ekg_1 = downsample(ekg_1, D1);
    current_FD = f_d_Hz / D1;

    b_safety_H2 = fir1(60,(3.5/(current_FD/2)));
% Naudojamas panasus filtras i decimate (filtro
eile = 30; fp() < (50 / 2*5)
    ekg_2 = filter(b_safety_H2, 1, ekg_1);
    ekg_2 = downsample(ekg_2, D2);
    current_FD = current_FD / D2;

    % Low-pass filtras
    b_lowPass = fir1(40, (0.6/(current_FD/2)));
    ekg_3 = filter(b_lowPass, 1, ekg_2);


    % iterpolate
    ekg_4 = upsample(ekg_3, D2);
    ekg_4 = filter(b_safety_H2, 1, ekg_4);
    ekg_4 = ekg_4 * D2;

    ekg_5 = upsample(ekg_4, D1);
    ekg_5 = filter(b_safety_H1, 1, ekg_5);
    ekg_5 = ekg_5 * D1; % dreifas

    ekg_6 = sig_y - ekg_5;

    velinimas = length(b_safety_H1) +
length(b_safety_H2)*D1 +
length(b_lowPass)*D1*D2/2

    ekg_withZeros = [zeros(1, velinimas), sig_y];
    dreif_withZeros = [ekg_5, zeros(1,
velinimas)];

    out_y = ekg_withZeros - dreif_withZeros;
end

function out_y = f_getNirFilter(sig_y, fd)
    fDelta_Hz = 1.5;
    f0_Hz = 50;
    K0 = 1;
    L = 3;
    S = mag2db(150);

    %3.4.2
    K = K0 * 10^(-S/20);
    Kr = K0 * 10^(-L/20);

    %3.4.3
    N = fd/f0_Hz;

    %3.4.4
    beta = sqrt((Kr^2 - K0^2)/(K^2 - Kr^2)) *
tan((N*pi*fDelta_Hz) / (2*fd));

    k1 = (K0 + (K*beta)) / (1 + beta);
    k2 = (K0 - (K*beta)) / (1 + beta);
    k3 = (1 - beta) / (1 + beta);

    bNIR = [k1, zeros(1,N-1), -k2];
    aNIR = [1, zeros(1,N-1), -k3];

    %3.4.6
    out_y = filter(bNIR, aNIR, sig_y);
end
```