

Feladat leírása: Egy olyan frekvenciaosztásos rendszer szűrőinek elkészítése, melyben 10kHz sávszélességű analóg csatornán két hangjelet egymástól függetlenül át lehet vinni.

Megvalósítás menete:

- Terveztem egy aluláteresztő és egy sáváteresztő szűrőt. Az aluláteresztő 4500-5000 Hz frekvenciáig átengedő, a sáváteresztő szűrő 5000-5500 és 9500-10000 Hz frekvenciák közt átengedő. Mindkét szűrő FIR szűrő.
- Mindkét bemeneti hang esetében eltávolítottam az 5 kHz-nél magasabb frekvenciákat az előbb tervezett aluláteresztő szűrő segítségével.
- Az egyik hangot frekvenciáját eltoltam 5 kHz-cel.
- Az így kapott két transzformált hangot összeadtam.
- Az „átvitel” után kettébontottam a jelet: az összeg-jelből aluláteresztő szűrővel visszanyertem az alsó sávban átvitt hangot.
- A másik hangot pedig először a sáváteresztő szűrővel „kivágtam” az összeg-jelből, majd visszatoltam az eredeti frekvenciatartományába.

Tapasztalatok, eredmények:

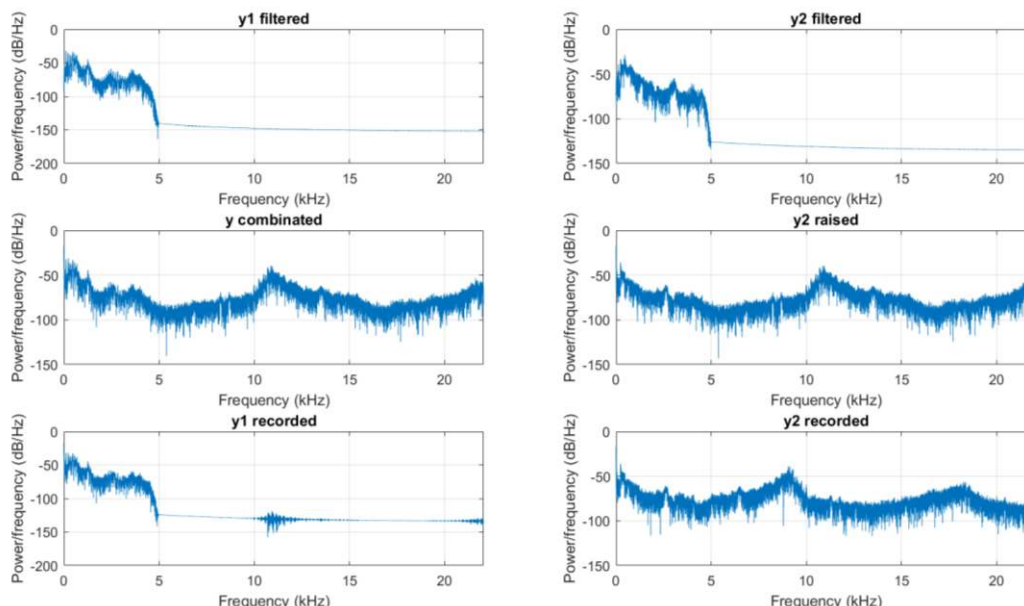
Sajnos a jel visszaállítása nem volt teljesen sikeres: Az első hangot vissza tudtam nyerni, bár abba is belehallatszott a másik hang (a Gibbs-oszcilláció miatt nem volt tökéletes a szűrő). A szűrő tervezésénél az áthallatszódást tudtam csökkenteni az szűrési tartomány amplitúdójának állításával. A nagyobbik probléma viszont a másik hang visszanyerésénél volt: szinte semmi sem maradt meg az eredeti hangból. Az okát keresve végül arra jutottam, hogy a raiseBy függvényre gyanakodtam, vagy közelítőleg sem jó eltolási frekvenciára.

A következő programrészlettel például azt vártam, hogy a kiindulási hangot visszakapom, de ez egyáltalán nem volt elmondható:

```
offsetFrequency=5000; % Todo: Mennyire kell eltolni? Nincs rossz válasz, csak rossz kombinációk.  
y2_raised = raiseBy(y2_filtered, Fs, offsetFrequency);  
y2_deraised = raiseBy(y2_raised, Fs, -offsetFrequency);  
disp('Na, most!');  
sound(y2_deraised, Fs);  
pause;
```

Az eltolási frekvenciához széles tartományból (20-10000) választott értékekkel próbálgattam, de nem jutottam eredményre.

A másik meghökkentő dolog pedig a spektrumoknál jelentkezett. Azt váltam, hogy a raiseBy függvény eltolja jobbra, de a kiindulásihoz inkább hasonló spektrumot kaptam.



Forráskód:

```
[y1,Fs1] = audioread('ferfi2.wav');
[y2,Fs2] = audioread('noi2.wav');

% Először is ha a két Fs nem egyezik, gond van
if Fs1 ~= Fs2
    disp('A két Fs nem egyezik. ');
    return;
end

Fs = Fs1;

% Mivel nem biztos, hogy a két hang hossza megegyezik, a rövidebbet
% megnyújtjuk
if length(y1)<length(y2)
    y1=[y1; zeros(length(y2)-length(y1), 1)];
else
    y2=[y2; zeros(length(y1)-length(y2), 1)];
end

% Először az y1-et és az y2-t is szűrni kell, hogy ne legyen egyikben sem
% olyan frekvenciájú hang, amit nem tudunk átvinni a számukra kijelölt
% csatornán. (Ami maximum 5kHz, hiszen két hangjelet kell átvinni egy
% 10kHz-es csatornán)

Hd=FIR_lowpass; % Todo: Ide kell annak a függvénynek a meghívása, amit az
FDATool kiad magából
y1_filtered = Hd.filter(y1);
Hd=FIR_lowpass;
y2_filtered = Hd.filter(y2);

disp('Nyomj ENTER-t a második lejátszásához!');
sound(y1_filtered, Fs);
pause;
disp('Nyomj ENTER-t a folytatáshoz!');
sound(y2_filtered, Fs);
pause;

% Ezután az egyik hangot feltoljuk a második csatornába
offsetFrequency=5000; % Todo: Mennyire kell eltolni? Nincs rossz válasz, csak
rossz kombinációk.
y2_raised = raiseBy(y2_filtered, Fs, offsetFrequency);

% Végül egyesítjük a két hangot
y_comb = y1_filtered + y2_raised;
disp('Nyomj enter a folytatáshoz!');
sound(y_comb, Fs);
pause;

% Most már csak szét kell őket választani
% Ehhez egy aluláteresztő és egy sáváteresztő szűrő kell
lowpassFilter = FIR_lowpass;
bandpassFilter = FIR_bandpass;
y1_rec = lowpassFilter.filter(y_comb);
y2_rec_raised = bandpassFilter.filter(y_comb);
y2_rec = raiseBy(y2_rec_raised, Fs, -offsetFrequency);
```

```
disp('Nyomj enter a folytatáshoz!');
sound(y1_rec, Fs);
pause;
disp('Nyomj enter a folytatáshoz!');
sound(y2_rec, Fs);
pause;

%kétféle módon is próbáltam a spektrumot vizsgálni, egyik sem adott kielégítő
válaszokat...
subplot(3,2,1);
% Y1_filtered = abs(fft(y1_filtered));
% plot(10:10000,Y1_filtered(10:10000));
plot(psd(spectrum.periodogram,y1_filtered,'Fs',Fs,'NFFT',length(y1_filtered)))
;
title('y1 filtered');
subplot(3,2,2);
% Y2_filtered = abs(fft(y2_filtered));
% plot(10:10000,Y2_filtered(10:10000));
plot(psd(spectrum.periodogram,y2_filtered,'Fs',Fs,'NFFT',length(y2_filtered)))
;
title('y2 filtered');
subplot(3,2,3);
% Y_comb = abs(fft(y_comb));
% plot(10:10000,Y_comb(10:10000));
plot(psd(spectrum.periodogram,y_comb,'Fs',Fs,'NFFT',length(y_comb)));
title('y combined');
subplot(3,2,4);
% Y2_raised = abs(fft(y2_raised));
% plot(10:10000,Y2_raised(10:10000));
plot(psd(spectrum.periodogram,y2_raised,'Fs',Fs,'NFFT',length(y2_raised)));
title('y2 raised');
subplot(3,2,5);
% Y1_rec = abs(fft(y1_rec));
% plot(10:10000,Y1_rec(10:10000));
plot(psd(spectrum.periodogram,y1_rec,'Fs',Fs,'NFFT',length(y1_rec)));
title('y1 recorded');
subplot(3,2,6);
% Y2_rec = abs(fft(y2_rec));
% plot(10:10000,Y2_rec(10:10000));
plot(psd(spectrum.periodogram,y2_rec,'Fs',Fs,'NFFT',length(y2_rec)));
title('y2 recorded');
```