Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva
Zavod za elektroničke sustave i obradbu informacija

Filtarski slogovi

Treća laboratorijska vježba iz Obradbe informacija (FER-2)

Tomislav Petković

1. Uvod

Laboratorijske vježbe iz Obradbe informacija se izvode na računalu. Za vježbe se koristi programski sustav MATLAB. Osim osnovnih mogućnosti programskog sustava MATLAB koristimo i modul Simulink koji je vizualni alat za simuliranje sustava. S MATLAB-om i Simulink-om ste se upoznali na LiV-u MATLAB.

2. Priprema

Prije svake vježbe potrebno se pripremiti za vježbu tako da ponovite teoriju s predavanja vezanu uz gradivo vježbe te tako da riješite sve pripremne zadatke.

2.1. Pripremni zadaci

Prije dolaska na laboratorijsku vježbu pročitajte sve zadatke koje ćemo raditi. Primijetite da uz neke od zadatke piše (PRIPREMA). To su zadaci koje morate riješiti prije dolaska na vježbu. Rješenja uredno napišite **rukom** na papiru. U zaglavlje svakog papira s pripremom napišete vaše ime i prezime te JMBAG.

Pripremni zadaci su: 3.1-1a), 3.2-1a), 3.2-1c), 3.2-2a) i 3.2-2c).

Pripreme zadatke 3.2-1a), 3.2-1c), 3.2-2a) i 3.2-2c) u kojima se traži kod napišite na računalu. Napisani kod NIJE POTREBNO prepisivati na papir.

2.2. Eliminacijska pitanja

Prilikom pripreme za vježbu morate usvojiti osnovne pojmove vezane uz temu vježbe. Tijekom laboratorijske vježbe nastavnik vam može postaviti neko od dolje navedenih eliminacijskih pitanja. Ukoliko ne znate odgovoriti na eliminacijsko pitanje vježba se boduje s nula bodova bez mogućnosti nadoknade.

Eliminacijska pitanja vezana uz treću laboratorijsku vježbu su:

1) Navedite koju funkciju pogreške minimiziramo kada FIR filtar projektiramo projekcijskom metodom.

2) Opišite kako iz matrice transformacije određujemo impulsne odzive filtara u analizirajućem dijelu filtarskog sloga.

3) Opišite kako iz inverzne matrice transformacije određujemo impulsne odzive filtara u rekonstrukcijskom dijelu sloga.

4) Koje su dimenzije (broj stupaca i redaka) matrice transformacije i koje su dimenzije matrice inverzne transformacije za DCT u N točaka?

Upute su dostupne putem internetski stranica http://www.fer.unizg.hr/predmet/obrinf/laboratorij.
Molimo da eventualne primjedbe i uočene pogreške šaljete e-poštom na adresu tomislav.petkovic.jr@fer.hr.

5) Koje su dimenzije matrice transformacije i koje su dimenzije matrice inverzne transformacije za MDCT u $2N$ točaka?

3. Rad u laboratoriju

Prije početka rada uključite dnevnik (naredba `diary`). Bez obzira na dnevnik također vam preporučamo da rješenje svakog zadatka spremite kao MATLAB m-skriptu ili Simulink model čije ime sadrži redni broj zadatka.

Bodove iz laboratorija stječete tijekom vježbi. Treća laboratorijska vježba nosi četiri boda. Priprema donosi jedan bod za ispravno rješenje zadatka 3.2-1a). Rad na vježbi donosi vam tri boda na način da se jedan bod daje za rezultate svakog do tri zadatka.

Svaki student može samo jednom demonstrirati rješenje zadatka. Dakle ako je demonstrirano rješenje zadatka krivo ne dobivate bodove. NEMA popravnih zadataka!

Dio zadataka označen je kao (PRIPREMA), dok je dio označen kao (ZA ONE KOJI ŽELE ZNATI VIŠE). Zadatke za pripremu ste riješili prije vježbi, dok zadatke za one koji žele znati više ne trebate rješavati. Predlažemo da ih preskočite te se vratite na njih ako obavezne zadatke završite prije predviđenog vremena.

3.1. Projektiranje FIR filtara i filtarski slog

Propuštamo li isti signal paralelno kroz različite filtre govorimo o filtarskom slogu. Kada konstruiramo filtarski slog obično zahtjeve postavljamo na svaki filtar u slogu te na slog u cjelini. Zatim se iz zahtjeva definira specifikacija svakog filtra u slogu te se isti projektira nekom od metoda za projektiranje filtara.

Razmotrimo najprije kako bi projektirali pojedini filtar iz sloga iz njegove specifikacije. Označimo s $A_d(\omega)$ željenu amplitudnu karakteristiku tog filtra. Želimo li naći FIR filtar tipa I (paran red filtra N, neparan broj uzoraka impulsnog odziva, simetričan odziv) koji najbolje aproksimira željenu karakteristiku u smislu srednje kvadratne pogreške možemo koristi teorem o projekciji. Amplitudna karakteristika FIR filtra tipa I je dana izrazom

$$A(\omega) = \sum_{m=0}^{N/2} a[m]\cos(\omega m),$$

pri čemu je veza koeficijenata $a[m]$ i uzoraka impulsnog odziva $h[n]$ dana izrazom

$$h[n] = \begin{cases} \frac{1}{2} a\left[\frac{N}{2} - n\right], & 0 \le n < \frac{N}{2} \\ a[0], & n = \frac{N}{2} \\ \frac{1}{2} a\left[n - \frac{N}{2}\right], & \frac{N}{2} < n \le N \end{cases}.$$

Koeficijenti $a[m]$ su prema teoremu o projekciji određeni izrazima

$$a[0] = \frac{1}{2\pi}\int_{-\pi}^{\pi} A_d(\omega)d\omega$$

i

$$a[m] = \frac{1}{\pi}\int_{-\pi}^{\pi} A_d(\omega)\cos(\omega m)d\omega.$$

Korištenjem navedenih izraza možemo projektirati FIR filtar tipa I koji je optimalan u smislu srednjeg kvadratnog odstupanja dobivene od željene karakteristike $A_d(\omega)$. Opisani postupak u MATLAB-u provodi naredba `firls`. Naredbi `firls` se uz red filtra mora zadati specifikacija amplitudno-frekvencijske karakteristike $A_d(\omega)$. Pri tome se specifikacija zadaje samo za pozitivne vrijednosti frekvencija i ograničena je na aproksimaciju pravcima. Specifikacija se sastoji od dva vektora s parnim brojem elemenata. U prvom vektoru se navode normirane frekvencije, a u drugom željena amplituda na frekvencijama zadanim prvim vektorom. Na primjer, filtar petog reda s karakteristikom prikazanom na slici 3. bi projektirali na slijedeći način:

```
» N = 5;                          % red filtra
» A = 1;                          % amplituda
» Fa = 0.25;                      % normirana donja granična frekvencija
» Fb = 0.75;                      % normirana gornja granična frekvencija
» firls(N, [0 Fa Fa Fb Fb 1], [0 0 A A 0 0])

ans =

   -0.3183    0.0000    0.5000    0.0000   -0.3183

» freqz(ans)                      % crtamo amplitudnu i faznu karakteristiku
```

Primijetite da je karakteristika zadana u tri segmenta, prvom na intervalu $\langle 0, f_a \rangle$ gdje je amplituda 0, drugom na intervalu $\langle f_a, f_b \rangle$ gdje je amplituda 1 i trećem na intervalu $\langle f_b, f_s/2 \rangle$ gdje je amplituda 0. Frekvencije su uvijek normirane tako da je $f_s = 2$. Karakteristiku za negativne frekvencije nije potrebno zadavati jer pretpostavljamo FIR filtar s realnim impulsnim odzivom, odnosno amplitudna karakteristika mora biti simetrična funkcija.

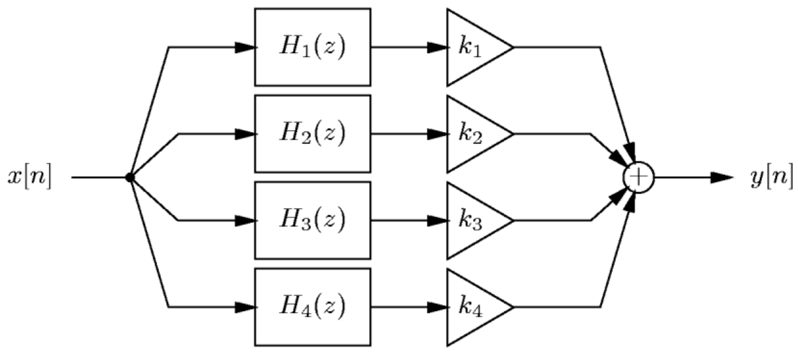
35 minuta

Zadatak 3.1-1 Filtarski slog za razdvajanje frekvencijskih pojaseva

Primjer filtarskog sloga prikazan je na slici 1. i predstavlja jednu moguću izvedbu audio ekvalizatora s četiri pojasa. Odaberemo li prijenosne funkcije filtara u slogu tako da vrijedi

$$\left|H_1(e^{j\omega}) + H_2(e^{j\omega}) + H_3(e^{j\omega}) + H_4(e^{j\omega})\right| = 1.$$

uz odabir jediničnih pojačanja ($k_1 = 1$, $k_2 = 1$, $k_3 = 1$ i $k_4 = 1$) filtarski slog ne mijenja signal. Mijenjanjem pojačanja k_i možemo utjecati samo na dio signala odabran filtrom $H_i(z)$. U ovom zadatku ćemo korištenjem projekcijske metode projektirati četiri FIR filtra koji nam mogu poslužiti za analizu snimljenog audio signala ili poboljšanje istog.

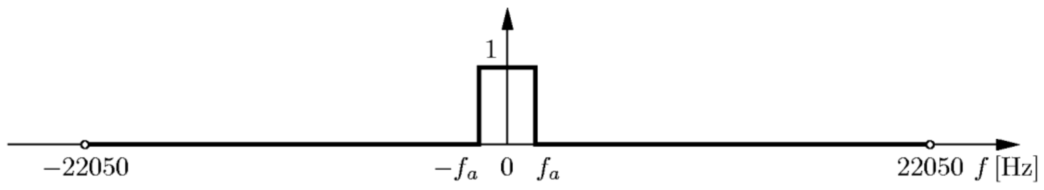


Slika 1. Filtarski slog koji se sastoji od četiri filtra

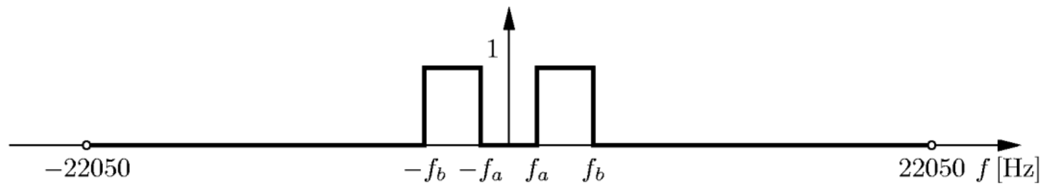
a) (PRIPREMA) Razmotrite četiri željene prijenosne karakteristike filtra prikazane na slikama 2., 3., 4. i 5. Pokažite da za FIR filtre tipa I projektirane projekcijskom metodom iz zadanih željenih karakteristika bez obzira na odabrani red filtra N uvijek vrijedi $\left|H_1(e^{j\omega}) + H_2(e^{j\omega}) + H_3(e^{j\omega}) + H_4(e^{j\omega})\right| = 1$ (točnije $H_1(e^{j\omega}) + H_2(e^{j\omega}) + H_3(e^{j\omega}) + H_4(e^{j\omega}) = e^{-j\omega N/2}$).

b) Korištenjem naredbe `firls` odredite impulsne odzive četiri FIR filtra tipa I koji aproksimiraju željene prijenosne karakteristike prikazane na slikama 2., 3., 4. i 5. Neka je red svih filtara $N = 32$ te neka je $f_a = 1470\text{Hz}$, $f_b = 4410\text{Hz}$, $f_c = 10290\text{Hz}$ i $f_d = 22050\text{Hz}$. Frekvencija očitavanja vremenski diskretnog signala je $f_s = 44100\text{Hz}$. Prilikom projektiranja pripazite na pretvorbu zadanih graničnih frekvencija u potrebne normirane frekvencije za naredbu `firls`.

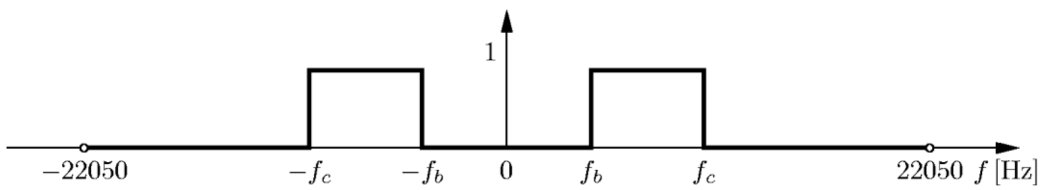
c) Naredba `freqz` za zadani impulsni odziv crta amplitudnu i faznu frekvencijsku karakteristiku pri čemu je amplitudna karakteristika prikazana u decibelima. Korištenjem naredbe `freqz` nacrtajte amplitudne i fazne frekvencijske karakteristike dobivenih filtara te amplitudnu i faznu frekvencijsku karakteristiku njihovog paralelnog spoja. Poklapaju li se karakteristike s zadanim karakteristikama?



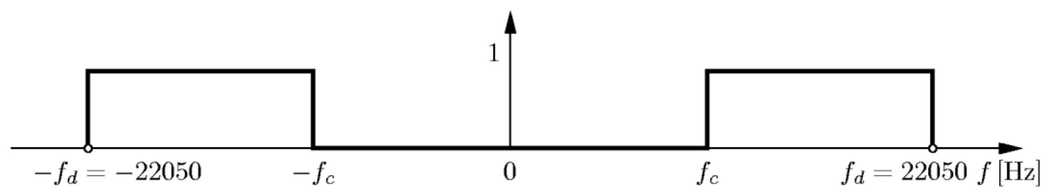
Slika 2. Željena karakteristika prvog filtra u slogu



Slika 3. Željena karakteristika drugog filtra u slogu



Slika 4. Željena karakteristika trećeg filtra u slogu



Slika 5. Željena karakteristika četvrtog filtra u slogu

d) (ZA ONE KOJI ŽELE ZNATI VIŠE) Model sa slike 1. izvedite u Simulinku ili u MATLAB-u.

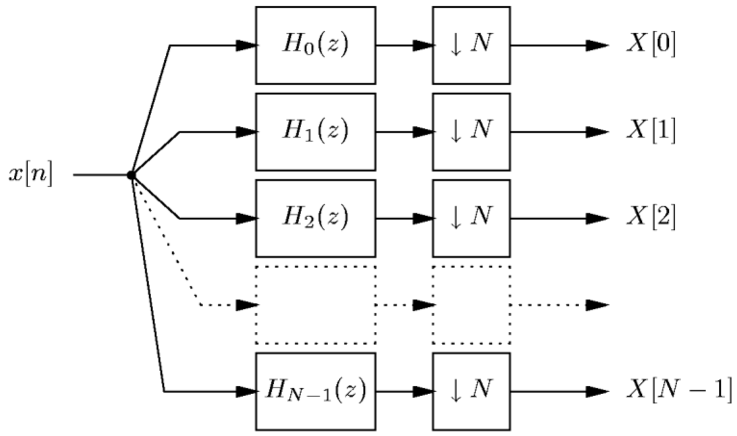
Ako koristite Simulink kao blok za FIR filtar koristite `Overlap-Add FFT Filtar` ili `Overalp-Save FFT Filtar` s time da kao broj točaka FFT-a odaberete 128. Potrebni koeficijenti FIR filtra su izračunate vrijednosti impulsnog odziva iz podzadatka c). Ulaz u filtarski slog neka bude izlaz iz `From Audio Device` bloka. Izlaz spojite na `To Audio Device` blok za reprodukciju signala. Svi navedeni blokovi se nalaze u skupini `Signal Processing Blockset`. U `Simulation` izborniku odaberite

`Accelerator` način simulacije[1]. Isprobajte razne vrijednosti pojačanja umjesto zadanih $k_1 = 1$, $k_2 = 1$, $k_3 = 1$ i $k_4 = 1$. Koje vrijednosti pojačanja bi odabrali za isticanje niskih frekvencija, a koje za isticanje visokih frekvencija?

Ako vam je simulacija prespora uz `Accelerator` način simulacije ili ako nemate na raspolaganju ubrzanje simulacije blok za snimanje zamijenite s blokom za učitavanje zvučnog signala i blok za reprodukciju zvučnog signala zamijenite s blokom za snimanje signala u datoteku. Nakon završetka obrade snimljeni signal možete poslušati korištenjem bilo kojeg programa za reprodukciju zvuka, a možete koristiti i naredbe `wavread` i `soundsc` unutar MATLAB-a.

3.2. DCT i MDCT filtarski slogovi

Drugi dio treće laboratorijske vježbe bavi se DCT i MDCT filtarskim slogovima. Kako ćemo razmatrati filtarske slogove za veće vrijednosti duljine bloka N Simulink nije pogodan alat jer je crtanje i povezivanje svih blokova vremenski zahtjevno. Razmotrimo stoga najprije kako bi u MATLAB-u realizirali slog prikazan slikom 6.



Slika 6. Analizirajući filtarski slog s podočitavanjem

Neka je ulazni signal $x[n]$ spremljen u varijabli `x` i neka su koeficijenti brojnika i nazivnika prijenosne funkcije $H_0(z)$ spremljeni u vektorima `A0` i `B0`. Naredbe koje su nam potrebne su `filter` i `downsample`. Naredba `filter` računa odziv LTI sustava na pobudu, dok naredba `downsample` vrši podočitavanje (ili poduzorkovanje) signala.

```
» x = [1:8] .^ 2;                    % definiramo ulazni signal
» A0 = [1];                          % nazivnik prijenosne funkcije
» B0 = [1 0 -1];                     % brojnik prijenosne funkcije
» filter(B0, A0, x)                  % računamo izlazni signal

ans =

     1     4     8    12    16    20    24    28

» downsample(ans, 4)                 % ostavljamo svaki 4 uzorak izlaza

ans =
```

[1] `Accelerator` način simulacije temeljem nacrtanog modela generira C kod koji se zatim prevodi. To dovodi do značajnog ubrzanja simulacije, no prilikom svake izmjene modela potrebno je pričekati na prevođenje.

```
    1    16
```

Za rekonstrukcijski slog umjesto naredbe `downsample` koristiti ćemo naredbu `upsample` koja vrši nadočitavanje (ili naduzorkovanje) signala.

```
» upsample([1 16], 4)                  % iza svakog uzorka dodajemo 3 nule

ans =

    1     0     0     0    16     0     0     0
```

Pri tome je važno razlikovati naredbe `upsample` i `downsample` od naredbi `interp` i `decimate`. Naime, nadočitavanje i podočitavanje s faktorom L su jednostavne operacije koje ostavljaju svaki L-ti uzorak (podočitavanje), što možemo opisati izrazom

$$y[n] = x[nL],$$

odnosno dodaju iza svakog uzorka još $L-1$ nula (nadočitavanje), što možemo opisati izrazom

$$y[n] = \begin{cases} x\left[\frac{n}{L}\right], & n/L \in \mathbb{Z} \\ 0 & n/L \notin \mathbb{Z} \end{cases}.$$

Interpolacija i decimacija osim tih osnovnih operacija uvijek pretpostavljaju postojanje filtra koji za je slučaj decimacije konstruiran tako da Nyquistov uvjet bude zadovoljen, dok je za slučaj interpolacije konstruiran tako da računa vrijednosti signala u novo-dodanim uzorcima korištenjem neke interpolacijske funkcije[2].

50 minuta

Zadatak 3.2-1 DCT filtarski slog

Prvi filtarski slog s podočitavanjem kojeg ćemo realizirati će biti DCT filtarski slog prikazan na slici 7. koji se temelji na DCT-II transformaciji. DCT-II transformacija je opisana izrazom

$$X[k] = \sqrt{\frac{2-\delta[k]}{N}} \sum_{n=0}^{N-1} x[n] \cos\frac{(2n+1)k\pi}{2N},$$

dok je inverzna transformacija (koja odgovara DCT-III transformaciji) opisana izrazom

$$x[n] = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} X[k] \frac{1}{\sqrt{1+\delta[k]}} \cos\frac{(2n+1)k\pi}{2N},$$

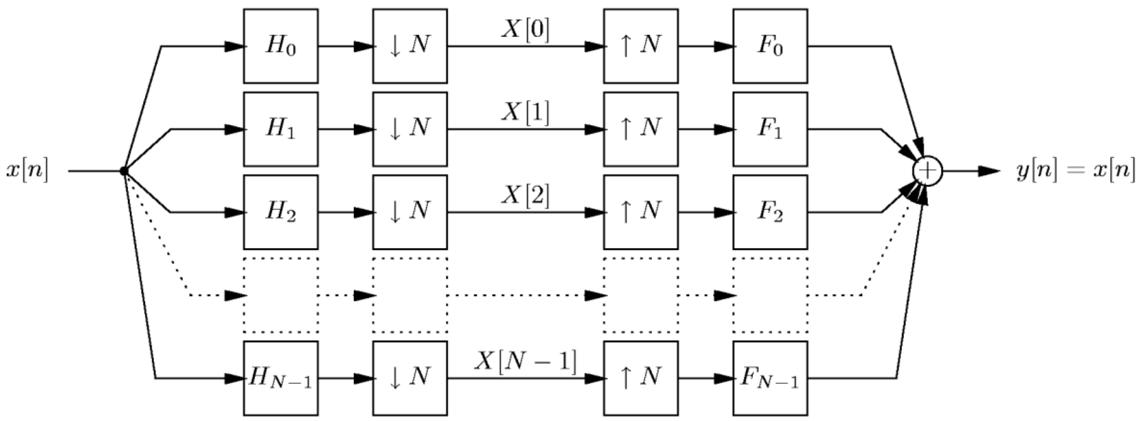
U MATLAB-u naredbe `dct` i `idct` realiziraju navedene transformacije.

Za realizaciju sloga koristiti ćemo naredbe `filter`, `downsample` i `upsample`. No prvo je potrebno izračunati impulsne odzive svakog filtra u slogu. Prisjetimo se da koeficijenti FIR filtara u analizirajućem dijelu sloga odgovaraju zrcaljenim redcima matrice diskretne kosinusne transformacije, dok koeficijenti rekonstrukcijskih FIR filtara odgovaraju stupcima matrice inverzne diskretne kosinusne transformacije. Pokažimo kako bi u MATLAB-u odredili te koeficijente za $N = 4$.

```
» N = 4;                               % transformacija je u 4 točke
» C = dct(eye(N))                      % matricu transformacije računamo tako da
                                       % kao argumet dct funkcije postavimo
C =                                    % jediničnu matricu

    0.5000    0.5000    0.5000    0.5000
    0.6533    0.2706   -0.2706   -0.6533
    0.5000   -0.5000   -0.5000    0.5000
    0.2706   -0.6533    0.6533   -0.2706
```

[2] U literaturi se ponekad izrazi miješaju tako da čitatelj uvijek mora biti oprezan te iz konteksta zaključiti radi li se o ubacivanju ili izbacivanju uzoraka ili se pak vrši i dodatna filtracija.

```
» H1 = C(2, 4:-1:1)                    % impulsni odziv drugog analizirajućeg filtra
                                       % odgovara ZRCALJENOM drugom retku
H1 =

   -0.6533   -0.2706    0.2706    0.6533

» CT = idct(eye(4))                    % matricu inverzne transformacije možemo
                                       % dobiti na jednak način
CT =                                   % CT mora biti transponirana matrica C

    0.5000    0.6533    0.5000    0.2706
    0.5000    0.2706   -0.5000   -0.6533
    0.5000   -0.2706   -0.5000    0.6533
    0.5000   -0.6533    0.5000   -0.2706

» F1 = CT(:, 2)                        % drugi rekonstrukcijski filtar odgovara
                                       % drugom stupcu (nema zrcaljenja)
F1 =

    0.6533
    0.2706
   -0.2706
   -0.6533
```



Slika 7. DCT filtarski slog s podočitavanjem

Pogledajmo sada kako bi izgledao programski odsječak koji računa koeficijente svih analizirajućih i rekonstrukcijskih filtara za proizvoljni N:

```
N = 32;

% Impulsni odzivi analizirajućih filtara su spremljeni u H{i}
C = dct(eye(N));
for i = 1 : N
    H{i} = C(i, N:-1:1);
end

% Impulsni odzivi rekonstrukcijskih filtara su spremljeni u F{i}
CT = idct(eye(N));
for i = 1 : N
    F{i} = CT(:, i);
end
```

Ako se ulazni signal nalazi u varijabli `x` i ako izlazni signal želimo spremiti u varijablu `y` mogući programski odsječak za realizaciju DCT sloga sa savršenom rekonstrukcijom za proizvoljni N bi bio:

```
1.    % Analizirajući dio DCT sloga.
2.    for i = 1 : N
3.        X{i} = downsample(filter(H{i}, 1, x), N, N-1);
4.    end
5.
6.    % Rekonstrukcijski dio DCT sloga.
7.    y = zeros(numel(X{1})*N, 1);
8.    for i = 1 : N
9.        y = y + filter(F{i}, 1, upsample(X{i}, N));
10.   end
```

Ukoliko želimo izvršiti neku manipulaciju s dobivenim koeficijentima rastava `X{i}` odgovarajući kod bi stavili između linija 4. i 6. Pri razmatranju primjene na kompresiju signala tipično želimo ukloniti one koeficijentima rastava `X{i}` koji ne nose informaciju.

a) (PRIPREMA) Prema navedenim odsječcima napišite na računalu m-skriptu ili funkciju koja za zadani `N` i signal `x` računa koeficijente rastava `X{i}` i rekonstrukciju `y`. Funkciju napišite tako da na jednostavan način iz samo nekih komponenti `X{i}` rastava bude moguće izračunati rekonstrukciju `y`.

Za ovaj zadak možete koristiti vlastiti snimljeni govorni signal ili neki od audio signala iz arhive dostupne na stranicama predmeta koja sadrži signale preuzete sa stranice http://www.xiph.org/. Za zadatak je potreban jednokanalni zvučni zapis pa ako koristite prethodno snimljeni višekanalni zvuk prije korištenja zbrojite sve kanale te normalizirajte amplitudu zbroja tako da ista bude po apsolutnoj vrijednosti manja ili jednaka jedinici.

b) Uz odabir $N = 32$ prvo pokažite da računanjem izlaza `y` temeljem svih elemenata rastava `X{i}` dobivamo savršenu rekonstrukciju signala (do na numeričku pogrešku). Zatim odredite rekonstrukciju `y` samo iz prvih 4, pa 8 i na kraju prvih 16 uzoraka spektra. Korištenjem naredbe `soundsc` preslušajte sve tri dobivene rekonstrukcije. Kako kvaliteta zvuka ovisi o odabiru elemenata rastava `X{i}` temeljem kojih računamo rekonstrukciju?

c) (PRIPREMA) Odnos signal-šum definiramo kao omjer snage signala i snage šuma,

$$SNR = \frac{P_{\text{signal}}}{P_{\text{šum}}} = \frac{\sum x^2[n]}{\sum (y[n]-x[n])^2}.$$

Korištenjem programa iz zadatka a) napišite program koji računa SNR za signala rekonstruiranog temeljem prvih $k < N$ komponenti rastava `X{i}`. Pri tome je $x[n]$ polazni audio signal i $y[n]$ izlazni signal iz sloga rekonstruiran na temelju prvih k komponenti.

d) Ako pri računanju rekonstrukcije `y` uzimamo samo prvih $k < N$ komponenti rastava `X{i}` nacrtajte kako SNR ovisi o k za $k \in \{1, \dots, N-1\}$. Možete li povezati dobivenu krivulju s kvalitetom zvuka?

50 minuta Zadatak 3.2-2 MDCT filtarski slog

Nešto kompliciraniji od DCT filtarskog sloga jest MDCT filtarski slog koji je prikazan na slici 8. MDCT filtarski slog temelji se na MDCT transformaciji koja je dana izrazom

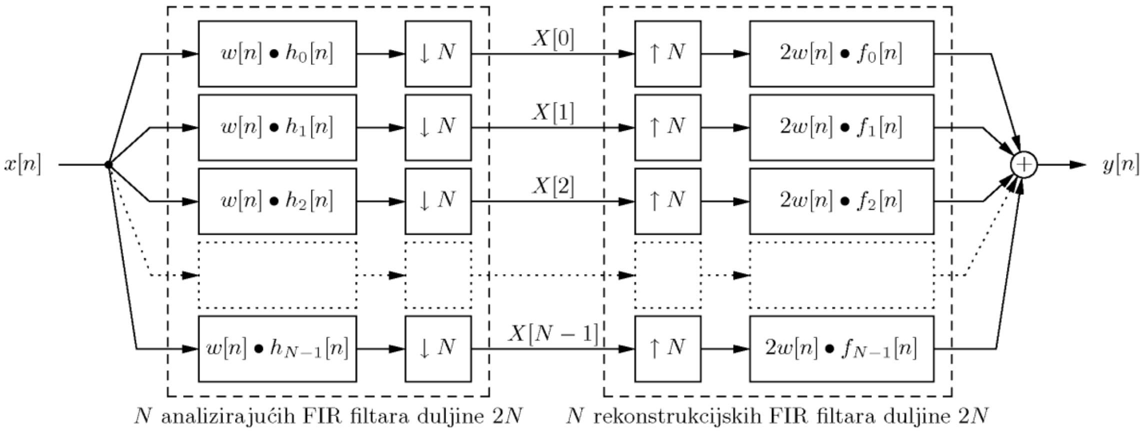
$$X[k] = \sum_{n=0}^{2N-1} x[n] \cos \frac{(2n+1+N)(2k+1)\pi}{4N},$$

dok je inverzna transformacija dana izrazom

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \cos \frac{(2n+1+N)(2k+1)\pi}{4N}$$

U MATLAB-u nažalost ne postoje funkcije za implementaciju MDCT i IMDCT transformacije, no zato su za potrebe vježbi napisane funkcije `mdct` i `imdct` koje možete preuzeti sa stranica predmeta i koje se ponašaju isto kao i `dct` i `idct` funkcije. Pogledajmo kako te funkcije koristimo za određivanje impulsnih odziva analizirajućih i rekonstrukcijskih FIR filtara:

```
» N = 2;                                 % transformacija je u 2 točke
» C = mdct(eye(2*N))                     % duljina bloka je 2N, a matrica
                                         % transformacije je N sa 2N
C =

    0.3827   -0.3827   -0.9239   -0.9239
   -0.9239    0.9239   -0.3827   -0.3827

» F0 = C(1, end:-1:1)                    % impulsni odzivi analizirajućih filtara su
                                         % jednaki ZRCALJENIM retcima matrice
F0 =                                     % transformacije C

   -0.9239   -0.9239   -0.3827    0.3827

» IC = imdct(eye(N))                     % matrica inverzne transformacije je
                                         % dimenzija 2N sa N
IC =                                     % znate li koliko je C*IC, a koliko je IC*C?

    0.1913   -0.4619
   -0.1913    0.4619
   -0.4619   -0.1913
   -0.4619   -0.1913

» C0 = IC(:,1)                           % impulsni odzivi rekonstrukcijskih filtara
                                         % su jednaki stupcima matrice IC
C0 =

    0.1913
   -0.1913
   -0.4619
   -0.4619
```



Slika 8. MDCT filtarski slog s podočitavanjem

Još jedna značajna razlika u odnosu na DCT filtarski slog s podočitavanjem iz zadatka 3.2-1 jest u primjeni vremenskog otvora $w[n]$. Za savršenu rekonstrukciju vremenski otvor $w[n]$ mora zadovoljiti uvjet

$$w^2[n] + w^2[N+n] = 1.$$

Kako se otvor primjenjuje? Primijetite da ga prema slici 8. primjenjujemo i na filtre analizirajućeg i na filtre rekonstrukcijskog dijela sloga. Pri tome je za analizirajuće filtre potrebno računati $w[n] \circ h_i[n]$, a za rekonstrukcijske filtre $2w[n] \circ f_i[n]$. Oznaka $\circ$ označava Schurov (ili Hadamardov) umnožak matrica - radi se o množenju elemenata matrice po članovima (operacija `.*` u MATLAB-u). Sada možemo pogledati i kako bi izgledao programski odsječak koji računa koeficijente svih analizirajućih i rekonstrukcijskih filtara za proizvoljni N:

```
N = 32;

% Prvo računamo vremenski otvor w koji se koristi za Vorbis.
n = 0 : 2*N - 1;
w = sin(pi/2*sin(pi/2/N*(n + 1/2)).^2);

% Impulsni odzivi analizirajućih filtara su spremljeni u H{i}
C = mdct(eye(2*N));
for i = 1 : N
    H{i} = w .* C(i,2*N:-1:1);
end

% Impulsni odzivi rekonstrukcijskih filtara su spremljeni u F{i}
IC = imdct(eye(N));
for i = 1 : N
    F{i} = 2 * w.' .* IC(:,i);
end
```

a) (PRIPREMA) Prema navedenim programskim odsječcima napišite na računalu m-skriptu (ili funkciju) koja za zadani `N` i signal `x` računa koeficijente rastava `X{i}` i rekonstrukciju `y`. za MDCT transformaciju uz korištenje nekog od vremenski otvora iz podzadatka e). Funkciju napišite tako da na jednostavan način iz samo nekih komponenti `X{i}` rastava bude moguće izračunati rekonstrukciju `y`.

Kao i za prethodni i za ovaj zadak možete koristiti vlastiti snimljeni govorni signal ili neki od audio signala iz arhive dostupne na stranicama predmeta koja sadrži signale preuzete sa stranice http://www.xiph.org/. Za zadatak je potreban jednokanalni zvučni zapis pa ako koristite prethodno snimljeni višekanalni zvuk prije korištenja zbrojite sve kanale te normalizirajte amplitudu zbroja tako da ista bude po apsolutnoj vrijednosti manja ili jednaka jedinici.

b) Korištenjem koda iz podzadatka a) uz odabir $N = 32$ prvo pokažite da računanjem izlaza `y` temeljem svih elemenata rastava `X{i}` dobivamo savršenu rekonstrukciju signala uz kašnjenje od $N = 32$ uzorka (i do na numeričku pogrešku). Zatim odredite rekonstrukciju `y` samo iz prvih 4, 8 i 16 uzoraka spektra. Korištenjem naredbe `soundsc` preslušajte sve dobivene rekonstrukcije. Kako kvaliteta zvuka ovisi o odabiru elemenata rastava `X{i}` temeljem kojih računamo rekonstrukciju? Je li rekonstrukcija kvalitetnija od rekonstrukcije u zadatku 3.2-1b).

c) (PRIPREMA) Prilagodite rješenje pripremnog zadatka 3.2-1c) za MDCT transformaciju.

d) Ako pri računanju rekonstrukcije `y` uzimamo samo prvih $k < N$ komponenti rastava `X{i}` nacrtajte kako SNR ovisi o k za $k \in \{1, \ldots, N-1\}$. Možete li povezati dobivenu krivulju s kvalitetom zvuka? Usporedite dobivenu krivulju s krivuljom iz zadatka 3.2-1d).

e) (ZA ONE KOJI ŽELE ZNATI VIŠE) U MDCT filtarskom slogu moguće je koristiti razne vremenske otvore. Ponovite prethodni zadatak za slijedeće vremenske otvore:

1. $w_1[n] = \frac{1}{\sqrt{2}}$

2. $w_2[n] = \sin\frac{(2n+1)\pi}{4N}$
3. $w_3[n] = \sin\left(\frac{\pi}{2}\sin^2\frac{(2n+1)\pi}{4N}\right)$

4. Literatura

1. Sanjit K. Mitra, *Digital Signal Processing – A Computer Based Approach*, McGraw-Hill, 1998., http://www.mhhe.com/engcs/electrical/mitra/
2. Stephen A. Martuci, *Symmetric Convolution and the Discrete Sine and Cosine Transforms*, IEEE Transactions on Signal Processing, sv. 42., br. 5., str. 1038-1051, svibanj 1994. http://dx.doi.org/10.1109/78.295213
3. Princen, J. i Bradley, A, *Analysis/Synthesis filter bank design based on time domain aliasing cancellation*, IEEE Transactions on Acoustics, Speech and Signal Processing, sv. 34., br. 5., str. 1153-1161, listopad 1986.
4. H. Babić, *Signali i sustavi (zavodska skripta)*, FER, Zagreb 1996., http://sis.zesoi.fer.hr/predavanja/pdf/sis_2001_skripta.pdf
5. T. Petković, *Kratke upute za korištenje MATLAB-a*, FER, Zagreb, travanj 2005., http://www.fer.unizg.hr/_download/repository/matlab_upute.pdf
6. MATLAB Technical Documentation, The MathWorks, http://www.mathworks.com/help/techdoc/
7. Simulink Technical Documentation, The MathWorks, http://www.mathworks.com/help/toolbox/simulink/
8. Signal Processing Toolbox Technical Documentation, The MathWorks, http://www.mathworks.com/help/toolbox/signal/