

FAKULTET ELEKTROTEHNIKE
I RAČUNARSTVA, ZAGREB

Zavod za elektroničke sustave
i obradbu informacija

DIGITALNA OBRADBA SIGNALA
UPUTE ZA LABORATORIJSKE VJEŽBE

D. Petrinović, D. Petrinović, R. Bregović, H. Babić, B. Jeren

Zagreb, 2003.

DIGITALNA OBRADBA SIGNALA

UPUTE ZA LABORATORIJSKE VJEŽBE

Ime i prezime: _____

Mat. broj: _____

Smjer: _____

Vježba broj:	Datum održavanja vježbe
1	
2	
3	
4	
5	
6	

Zagreb, 2003.

Sadržaj

Predgovor	1
A. Opće napomene za laboratorijske vježbe	2
B. DSP maketa	3
Vježba 1. - Razmatranje efekata otipkavanja	6
Vježba 2. - Diskretna Fourier-ova transformacija (DFT)	14
Vježba 3. - Projektiranje FIR filtara metodom vremenskih otvora	29
Vježba 4. - Projektiranje optimalnih FIR filtara	43
Vježba 5. - Projektiranje IIR filtara bilinearnom transformacijom i njihova realizacija direktnom formom II	52
Vježba 6. - Kaskadna realizacija IIR filtara	69

Predgovor

Upute za laboratorijske vježbe iz predmeta Digitalna obradba signala, imaju za cilj lakše savladavanje koraka između teorije i prakse. Nastale su kao rezultat višegodišnjeg predavanja predmeta od strane prof. dr. H. Babića te potrebe za ilustriranjem svih efekata i problema koji se javljaju u području digitalne obradbe signala. Kroz vježbe polaznik stječe osnovna znanja za budući samostalni rad u ovom području.

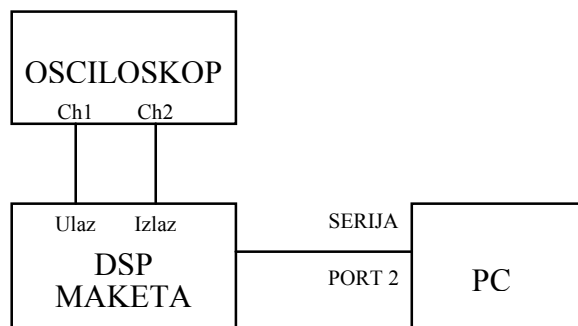
Sve vježbe se sastoje od četiri djela. Prvi dio predstavlja teoretske pripreme koje polaznik mora proučiti prije dolaska na laboratorijske vježbe. U njemu se polaznik podsjeća na osnove zakonitosti, izraze i definicije koje su potrebne za razumijevanje vježbe. Drugi dio se sastoji u rješavanju pismenog zadatka iz priprema. Rješenja se provjeravaju u toku održavanja laboratorijskih vježbi od strane asistenta. Treći dio se sastoji od programa pisanih u programskom paketu MATLAB. U njima se kroz niz simulacija uz numerički i grafički prikaz objašnjava materijal dan u teoretskom dijelu vježbi. Od polaznika se zahtijeva da analizira dobivene rezultate te po potrebi mijenja određen broj ulaznih parametara. Sva zapažanja kao i odgovore na konkretna pitanja postavljena u pripremama treba upisati na za to predviđeno mjesto. Zadnji dio vježba se sastoji od realizacija proučavanih parametara na konkretnom sklopovlju. U tu svrhu je na zavodu razvijena DSP maketa temeljena na DSP procesoru ADSP2181 firme Analog Devices. Opis makete dan je u jednom od poglavlja uvodnog dijela.

Vježbe su podijeljene u nekoliko tematskih cjelina. Na samom početku laboratorijskih vježbi proučavaju se efekti uslijed otipkavanja signala. To je osnova za daljnje razumijevanje svih ostalih aspekata digitalne obradbe signala. Posebnom vježbom su obuhvaćena svojstva diskretne Fourierove transformacije (DFT) te prikazan njezin značaj pri analizi realnih signala. Veći dio vježbi posvećen je nerekurzivnim (FIR) i rekurzivnim (IIR) digitalnim filtrima. Razmatrane su različite realizacije te problemi koji se javljaju prilikom izvedbe filtara na sklopovlju s ograničenom duljinom riječi. Kroz primjere je prikazano kako se pravilnom realizacijom može većina tih problema izbjeći, a posebno je skrenuta pažnja na one probleme koji nakon svih zahvata i dalje ostaju.

Autori se zahvaljuju svim ostalim suradnicima Zavoda za elektroničke sustave i obradbu informacija kao i studentima dodiplomske nastave koji su svojim radom pripomogli povećanju kvalitete laboratorijskih vježbi iz predmeta Digitalna obrada signala.

A. Opće napomene za laboratorijske vježbe

Na laboratorijskim vježbama koristit će se računalo, DSP maketa i osciloskop. Shema spajanja mjerne opreme prikazana je na Slici A.1. **Ne raspajati mjernu opremu !**




Slika A.1 Shema spajanja mjerne opreme

Nakon paljenja računala i podizanja operativnog sustava, treba upisati korisničko ime USER NAME: **STUDENT** i zaporku PASSWORD: **STUDENT**, domena **LSS**.

Po završetku procesa inicijalizacije operativnog sustava potrebno je u Exploreru otići na Z: disk (mrežni disk) u direktorij **Z:\DOS** i u njemu napraviti svoj radni pod-direktorij. Imenovati ga svojim prezimenom, ali u dužini od **8 znakova, bez specijalnih znakova i naših slova**. U svoj radni direktorij prekopirati kompletni sadržaj direktorija **Z:\DOS\VJDOS**.

U sklopu vježbi će se koristiti programski paket MATLAB. U pitanju je matematičko programsko okruženje s velikim matematičkim i grafičkim mogućnostima. Kratke upute za rad dane su u prilogu. Na računalima su instalirane dvije verzije ovog programa: 6.5 i 4.0 (starija). Laboratorijske vježbe uglavnom su koncipirane tako da u prvom dijelu vježbe studenti trebaju prema danim uputama napisati kratki Matlab program. Za to se može koristiti i Matlab 4 i Matlab 6.5 (koji ima prikladnije okruženje i veće mogućnosti). U drugom dijelu vježbe izvode se postojeće Matlab aplikacije kojima se ostvaruje i veza sa DSP maketom. **Za to se koristi isključivo Matlab 4.**

Matlab verzija 4 se pokreće dvostrukim klikom na ikonu na Desktopu  ili odabirom **Start-Programs-Matlab-Matlab4**, za Matlab 6.5, **Start-Programs-Matlab-Matlab65**. Po pokretanju programa otvara se komandni prozor (Command Window) sa znakom '»' što označava da program očekuje naredbu. U MATLAB programskom okruženju (komandnom prozoru) naredbom **cd z:\dos\user_dir** prijeći u **svoj** radni direktorij. Nakon toga prijeći u odgovarajući pod-direktorij **vjX** gdje X predstavlja redni broj vježbe koja se izvodi. Za prvu vježbu vj1, drugu vj2itd.

Za pisanje Matlab programa otvoriti Notepad (Start-Programs-Accessories-Notepad) ili Matlab-ov editor i u njemu upisivati linije koda. Datoteku spremiti pod nekim imenom, ali pri tome paziti da je ime različito od prethodno pripremljenih programa u direktoriju vježbe u kojem se radi. Imena programa moraju imati **ekstenziju .m** (da bi se izvodio u Matlabu). S obzirom da je Matlab interpreter, u toku programiranja se može (i vrlo je korisno!) isprobati kako pojedina napisana linija koda stvarno djeluje i to tako da se dotična linija (ili više njih) prekopiraju iz Notepad-a sa Edit-Copy i ubace u Matlab prozor s Edit-Paste (klasični način u Windows-ima), te se udari **Enter** da se linije izvedu.

Svi gotovi MATLAB programi (pa i novo-napisani) pokreću se upisivanjem imena programa (bez nastavka **‘.m’**) u komandni prozor MATLAB-a nakon znaka **‘>’**.

Na laboratorijskim vježbama nužno je napraviti zadane programe u Matlabu, odgovoriti na sva pitanja postavljena u pripremama, nacrtati tražene slike te izvesti zadane postupke posebno stoga što se stvari rađene na laboratorijskim vježbama mogu provjeravati i na ispitu. Skriptu će pregledati asistent.

Po završetku vježbe ugaziti DSP maketu, osciloskop i računalo (ne raspajati opremu!).

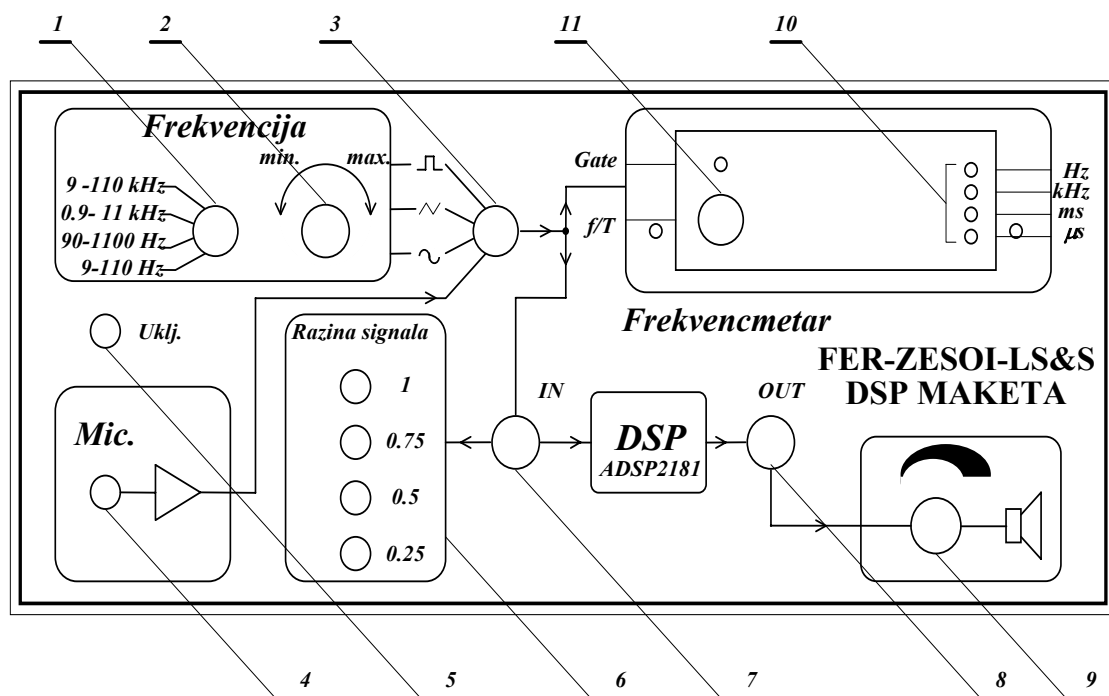
B. DSP maketa

Za praktičnu realizaciju programa za digitalnu obradu signala korištena je DSP maketa koja se sastoji od korisničkog sučelja, funkcijskog generatora s mjerilom frekvencije, A/D i D/A pretvornika te DSP procesora. Komunikacija DSP makete i računala ostvarena je preko serijskog porta računala pripadne programske podrške.

Sama DSP maketa sastoji se od slijedećih dijelova:

1. DSP procesor ADSP2181
2. AD/DA pretvornik AD1847
3. Funkcijski generator s frekvencijskim rasponom od 9Hz do 110kHz te mogućnošću generiranja pravokutnog, trokutastog ili sinusnog napona
4. Mjerilo frekvencije
5. Mikrofon
6. Zvučnik

Maketa je izrađena kao samostalan uređaj za digitalnu obradu ulaznih podataka koji mogu dolaziti s funkcijskog generatora ili nekog drugog vanjskog ulaza te njihovim prosljeđivanjem na izlaz. Na slici B.1 prikazana je prednja ploča makete.

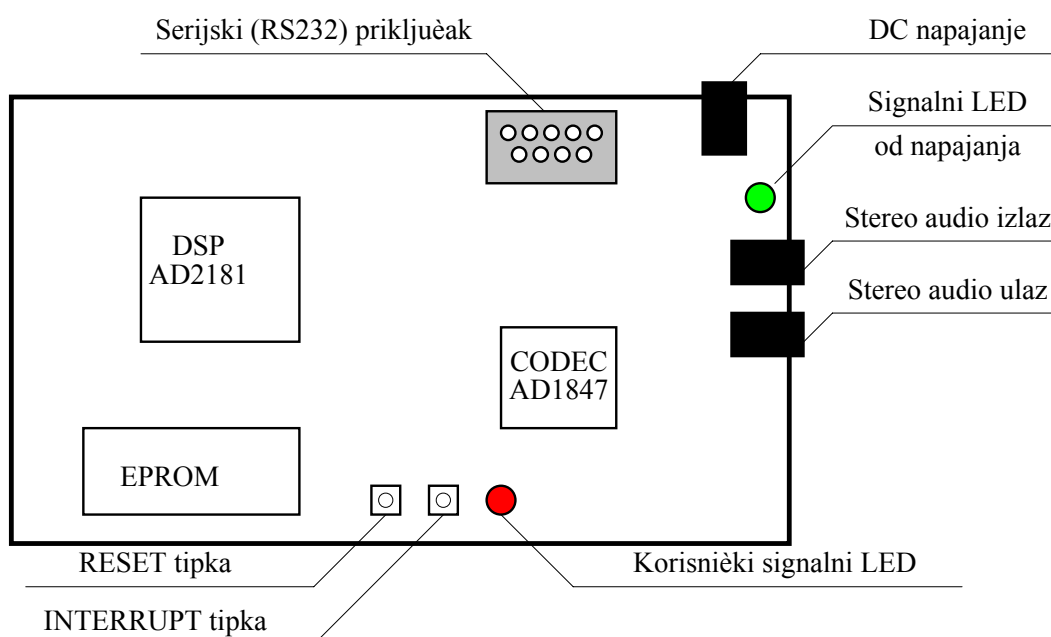


Slika B.1 Prednja ploča DSP makete

Funkcije preklopki, priključnica i LED dioda označenih brojevima 1 do 9 su:

1. Skokovita promjena frekvencije (po dekadama). Frekvencije pokrivaju područje signala (9Hz-110kHz)
2. Kontinuirana promjena frekvencije
3. Preklopka za odabir ulaznog signala u DSP
4. Mikrofonski ulaz. Ako je preklopka 3 u donjem položaju, mikrofonski signal ulazi u DSP
5. LED za prikaz priključenosti uređaja na mrežni napon
6. Pokazivač razine ulaznog signala u DSP
7. Ulazni BNC (može se priključiti osciloskop)
8. Izlazni BNC
9. Potenciometar za regulaciju jakosti signala na zvučniku
10. LED za prikaz trenutne mjerne jedinice
11. Tipkalo za odabir mjerenja frekvencije ili periode

DSP ploča smještena je na maketi ispod pleksiglasa. Pogled na maketu s gornje strane dan je na slici B.2.



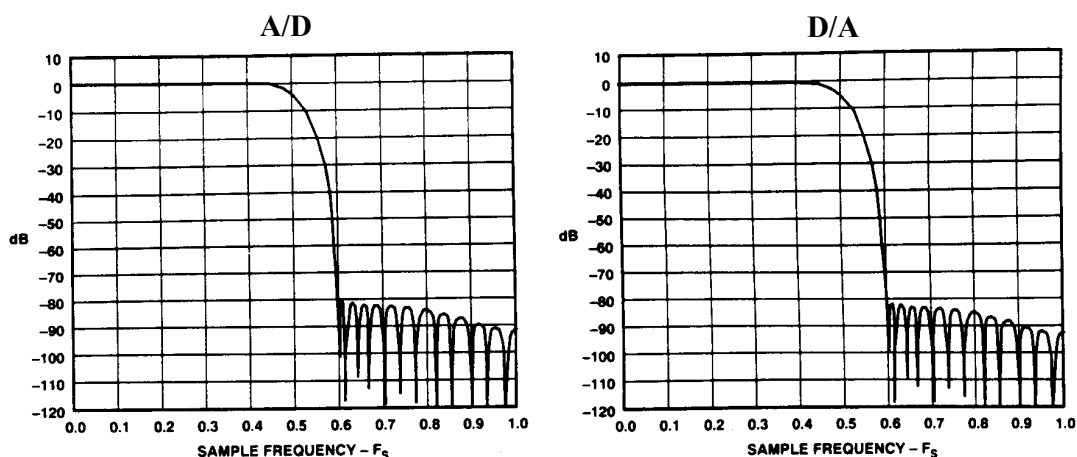
Slika B.2 Prednja ploča DSP makete

Osnovni dijelovi DSP ploče su:

1. DSP procesor ADSP 2181, frekvencija rada 33 MHz
2. A/D i D/A pretvornik AD1847, 16 bitni codec
3. RS232 korisničko sučelje
4. EPROM, sadrži program za komunikaciju s osobnim računalom preko RS232 te nekoliko primjera rada DSP ploče
5. stereo audio ulaz i izlaz

6. tipke za RESET i korisnički INTERRUPT
7. korisnički signalni LED
8. regulator napajanja 7805
9. konektor za proširenje
10. EZ-ICE konektor

U sastavu DSP makete nalazi se Codec AD1847 firme Analog Devices. Radi se o stereo A/D i D/A 16 bitnom sigma-delta pretvorniku. Prijenosne karakteristike anti-aliasing filtra A/D pretvornika i interpolacijskog filtra D/A pretvornika Codec-a dane su na Slici B.3.



Slika B.3 Prijenosne karakteristike A/D i D/A pretvornika

Treba uočiti da amplitudne karakteristike, prikazane na slici, ne ovise o iznosu frekvencije otipkavanja F_s (na osi apscisa brojka 1.0 označava F_s), pa su karakteristike pretvornika neovisne o promijeni frekvencije otipkavanja čime je jedan takav pretvornik primjenljiv za različite frekvencije otipkavanja bez ikakvih sklopovskih zahvata na sustav. Također treba uočiti da dobiveni anti-aliasing filter nije idealan što nikada niti ne može biti u stvarnoj izvedbi. Naime, filter počinje gušiti i malo prije $0.5 \cdot F_s$, te ima i prijelazno područje određene širine. To znači da će aliasinga u izvjesnoj mjeri ipak biti, o čemu treba voditi računa.

Codec je programabilan, a to znači da omogućuje promjenu pojačanja ulaznog i izlaznog signala, različita miješanja ulaznog (jednog ili više) i izlaznog signala u analognoj ili digitalnoj domeni kao i promjenu frekvencije otipkavanja. Zbog sklopovskih ograničenja codec-a dostupne su samo određene frekvencije otipkavanja dane tablicom B.1.

Tablica B.1 Dostupne frekvencije otipkavanja codec-a

Redni broj	1	2	3	4	5	6	7
F_o (kHz)	5,5125	6,615	8,0	9,6	11,025	16,0	18,9
Redni broj	8	9	10	11	12	13	14
F_o (kHz)	22,05	27,429	32,0	33,075	37,8	44,1	48,0

1. VJEŽBA

Razmatranje efekata otipkavanja

Da bi se analogni, vremenski kontinuirani signal mogao digitalno obrađivati potrebno ga je otipkati u vremenu. To znači da od vremenski kontinuiranog signala uzimamo samo određene uzorke. Ti se uzorci uzimaju najčešće s jednakim vremenskim razmakom T . Takav T se naziva period otipkavanja, pa vremenske trenutke u kojima se obavlja otipkavanje možemo opisati nizom:

$$t_n = T \cdot n, \quad -\infty < n < \infty. \quad (1.1)$$

Tim postupkom se od vremenski kontinuiranog signala $x_c(t)$ dobiva diskretni niz uzoraka $x[n]$, prema:

$$x[n] = x_c(nT), \quad -\infty < n < \infty. \quad (1.2)$$

Spektar vremenski diskretnog signala dobivenog otipkavanjem s periodom T je **kontinuirana funkcija periodična s periodom 2π** . Pri tome vremenski kontinuiranoj sinusoidi kružne frekvencije Ω odgovara diskretna sinusoida frekvencije ω i vrijedi:

$$\omega = \Omega T \quad [\text{rad}]. \quad (1.3)$$

Općenito, veza između spektra kontinuiranog signala $X_c(j\Omega)$ i spektra diskretnog signala $X(e^{j\omega})$ dana je izrazom:

$$X(e^{j\omega}) = \frac{1}{T} \sum_{n=-\infty}^{\infty} X_c(j\frac{\omega}{T} - j\frac{2\pi \cdot n}{T}). \quad (1.4)$$

Važan kriterij kojeg treba uzeti u obzir kod odabira frekvencije otipkavanja kontinuiranog signala dan je **Nyquist-ovim teoremom otipkavanja** koji slijedi. Neka je $x_c(t)$ frekvencijski ograničen signal za koji vrijedi:

$$X_c(j\Omega) = 0, \quad \text{za } |\Omega| > \Omega_N. \quad (1.5)$$

Tada je $x_c(t)$ jednoznačno određen svojim uzorcima $x[n] = x_c(nT)$, $-\infty < n < \infty$ ako vrijedi da je:

$$\Omega_s = \frac{2\pi}{T} > 2\Omega_N. \quad (1.6)$$

Drugim riječima, to znači da ako je najviša kružna frekvencija sadržana u signalu Ω_N , tada signal treba otipkavati s barem dvostruko većom frekvencijom Ω_s . U tom slučaju u diskretnom nizu $x[n]$ sačuvana je sva informacija sadržana u izvornom signalu $x_c(t)$. Ako frekvencija signala pređe pola frekvencije otipkavanja, dolazi do pojave koja se naziva **aliasing**, a posljedica je periodičnosti spektra diskretnog signala.

Pojavu možemo prikazati jednostavnim primjerom. Za sinusni (kosinusni) signal frekvencije 90 Hz dan izrazom:

$$x_c(t) = \cos(2\pi \cdot 90t) \quad (1.7)$$

i frekvencije otipkavanja $f_s = \frac{1}{T} = 100 \text{ Hz}$, dobivamo diskretni niz:

$$x[n] = \cos(2\pi \cdot \frac{90}{100} n) \quad (1.8)$$

koji je identičan nizu:

$$x_a[n] = \cos(2\pi \cdot \frac{10}{100} n). \quad (1.9)$$

što se može lako pokazati. To znači da kosinusoide frekvencija 10 Hz i 90 Hz imaju istog reprezentanta u vremenski diskretnoj domeni, pa je nakon otipkavanja nemoguće odrediti o kojoj se od njih dvije radilo. Isti niz dobili bi za frekvencije signala 110 Hz, 190 Hz, 210 Hz itd.

U praktičnoj primjeni, radi sprječavanja pojave aliasinga, se signal prije otipkavanja propušta kroz nisko propusni filter s graničnom frekvencijom $f_s / 2$ ili nižom, čime se ponište (priguše) sve više frekvencije koje bi nakon otipkavanja mogle prouzročiti aliasing. Takvi filteri su poznati pod nazivom **anti-alias filteri**.

Rekonstrukcija frekvencijski ograničenog otipkanog signala na osnovu njegovih vremenskih uzoraka

Ako je kontinuirani signal otipkan tako da nema aliasing-a, tada je moguće na osnovu dobivenih uzoraka signala $x[n]$ rekonstruirati izvorni vremenski kontinuirani signal. To se postiže primjenom interpolacijske funkcije oblika $\sin(x) / x$ prema izrazu:

$$x_r(t) = \sum_{l=-\infty}^{+\infty} x[l] \frac{\sin\left[\frac{\pi(t-lT)}{T}\right]}{\frac{\pi(t-lT)}{T}}, \quad (1.10)$$

gdje su $x[l]$ uzorci diskretnog signala, a $x_r(t)$ je rezultirajući rekonstruirani kontinuirani signal koji ima definiranu vrijednost za svaki trenutak t . Kao što se vidi iz (1.10), neki uzorak $x[l]$ treba pomnožiti s pripadnom interpolacijskom funkcijom kojoj je ishodište pomaknuto upravo na mjesto dotičnog uzorka. Za trenutak $t = lT$ koji odgovara tom uzorku, i nazivnik i brojnik interpolacijske funkcije su jednaki nuli pa se dobiva kvocijent 0/0 koji je u limesu jednak 1. Za dotični t , umnožak interpolacijske funkcije i vrijednosti samog uzorka upravo je jednak samom uzorku. Za sve ostale $t = jT$, $j \neq l$, koji korespondiraju mjestima ostalih uzoraka, umnožak je jednak nuli jer je argument sinusne funkcije višekratnik od π , pa je brojnik uvijek jednak 0, a nazivnik je različit od nule. Kada se množenje izvede za sve uzorke i doprinosi zbroje, u rekonstruiranom signalu na mjestima $t = nT$ ostaje doprinos samo n -tog uzorka i vrijedi:

$$x_r(nT) = x[n]. \quad (1.11)$$

To vrijedi za bilo koji n . Vrijednost interpolacijske funkcije između uzoraka dobiva se sumacijom doprinosa svih uzoraka prema danoj formuli.

Realni sinusni (kosinusni) niz

Realni sinusni niz definiran je izrazom $x[n] = A \cos(\omega_0 n + \phi)$, a A , ω_0 , ϕ su realni brojevi, gdje je A amplituda, ω_0 kružna frekvencija u radijanima/uzorku, te ϕ početna faza. Niz će biti periodičan s P (isti uzorci se ponavljaju nakon P uzoraka) ako vrijedi:

$$x[n] = A \cos(\omega_0 n + \phi) = A \cos(\omega_0 (n + P) + \phi), \quad (1.12)$$

a to će biti zadovoljeno ako je:

$$\omega_0 \cdot P = 2\pi \cdot k, \quad \text{odnosno} \quad \frac{2\pi}{\omega_0} = \frac{P}{k}, \quad (1.13)$$

Drugim riječima, ako je umnožak $\omega_0 P$ cjelobrojni višekratnik od 2π (P i k su cijeli brojevi), sinusni niz će biti periodičan. Najmanji P za koji vrijedi gornji uvjet, naziva se osnovni period niza. Ako u gornjem uvjetu P/k nije racionalni broj, niz neće biti periodičan iako će imati anvelopu oblika sinusoida.

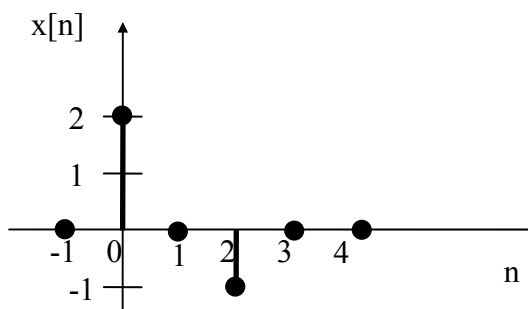
Lako se može pokazati da iz same definicije sinusnog niza proizlaze slijedeća interesantna svojstva. Niz uzoraka dobiven uz kutnu frekvenciju $\pi + \Delta$ jednak je nizu koji bi se dobio uz frekvenciju $\pi - \Delta$. Slično tome, nizovi dobiveni uz kružne frekvencije Δ i $2\pi - \Delta$ su jednaki. Iz svega ovoga slijedi da su nizovi dobiveni uz kružne frekvencije $\omega_0 + 2k\pi$, $-\pi < \omega_0 < \pi$, $k \in \mathbb{Z}$ **ne mogu razlikovati**, tj. identični su.

Zadatak za pripremu

Za signal prikazan slikom odrediti interpolacijsku funkciju oblika $\frac{\sin(x)}{x}$. Izračunati vrijednost funkcije za $t_1=0.5$ i $t_2=1.5$ ($T=1$), te skicirati doprinose interpolacijske funkcije za svaki od uzoraka.

$$x[n]=0 \text{ za } n<0$$

$$n>3$$



Rješenje zadatka:

PROGRAM ZA PRIKAZ SINUSNOG NIZA U MATLABU

PROGRAM ZA OTIPKAVANJE U MATLABU

b) Vremenski kontinuirani signal nije moguće egzaktno prikazati diskretnim sustavom kao što je računalo. Ono što se može napraviti jest prikaz pomoću velikog broja uzoraka (točaka) tako da dobivamo dojam 'kontinuiranog'. Treba definirati takvu 'kontinuiranu' vremensku os t koja će u rasponu pokrivenom sa L uzoraka (koliko ih promatramo) imati puno više točaka. To

se može napraviti pomoću Matlab izraza `t=[0:T/broj_toc:L*T]`; gdje srednji broj predstavlja korak, a između dva diskretna uzorka bit će *broj_toc*-1 točaka. Prije izraza za *t*, definirati neki proizvoljni broj točaka, npr. `broj_toc=20`.

c) Ostvariti unos frekvencije vremenski kontinuiranog signala f_0 funkcijom `input` te napisati naredbu za izračunavanje kružne frekvencije vremenski kontinuiranog signala Ω_0 . Na temelju poznatog odnosa, izraziti u Matlabu i kružnu frekvenciju **vremenski diskretnog** signala ω_0 pomoću Ω_0 .

d) Napisati Matlab izraz za 'kontinuirani' kosinusni signal x_c pomoću *t* i f_0 . Također izračunati i niz uzoraka x_s koji odgovaraju vrijednostima signala x_c u trenucima otipkavanja. Ovaj niz se dobiva pomoću *tn* i f_0 .

e) Izračunati uzorke vremenski diskretnog signala *x* dobivenog otipkavanjem, na temelju kružne frekvencije ω_0 i niza indeksa *n*.

f) Sada prikazati sve dobivene signale. Najprije pozvati funkciju `figure` za otvaranje novog prozora za sliku. U **prvom prozoru** (slici) pomoću funkcije `plot` prikazati signale x_c i x_s , svakog ovisno o pripadnoj vremenskoj osi (*t*, odnosno *tn*) pri čemu uzorke x_s treba prikazati kružićima. Označiti os apscisa oznakom za sekunde naredbom `xlabel('sec');`. Nadalje treba izgenerirati naslov za sliku. Naslov treba sadržavati trenutne vrijednosti varijabli f_0 i *T* (da kasnije znamo uz koje ulazne parametre je slika ostvarena), formatirane na određeni način te napisane u obliku stringa. Za generiranje stringa koristiti naredbu `sprintf` kako slijedi: `str=sprintf('f0=%0.2f, T=%0.2f',f0,T);`, a za postavljanje naslova na sliku: `title(str)`.

g) Ponovno pozvati funkciju `figure`, te u ovom **drugom prozoru** pomoću funkcije `stem` prikazati diskretne uzorke *x* ovisne o indeksima *n*. Os apscisa označiti pomoću `xlabel('uzorci');` te postaviti isti naslov kao i na prvu sliku.

5. Izvesti program uz slijedeće parametre: $L=10$, $f_0=1$, $T=0.1$ i promotriti slike. Nakon toga izvesti program uz isti *L* i *T*, ali za f_0 izabrati 4, pa onda 6, 9 i 11. Svaki put se generiraju po dvije nove slike. Usporediti ih i odgovoriti na pitanja:

Uz koje frekvencije kontinuiranog signala je nakon otipkavanja dobiven isti diskretni niz uzoraka? Provjerite računski da li rezultati programa odgovaraju teoriji.

Kako se naziva pojava do koje je došlo ? Obratiti pažnju na broj diskretnih uzoraka po jednoj periodi 'kontinuiranog' signala. Koliko ih najmanje mora biti da idealno rekonstruirani signal bude jednak polaznom ?

Zatvoriti sve prozore naredbom `close all`.

OTIPKAVANJE I INTERPOLACIJA SIGNALA (MATLAB APLIKACIJA)

6. Pokrenuti postojeći program **vj1** unosom naredbe **vj1** <ENTER> u komandni prozor MATLAB programskog okruženja. Ovim se otvara novi prozor s grafičkim sučeljem koje omogućava lako i pregledno razmatranje pojava vezanih uz otipkavanje i interpolaciju signala.

Sam izgled grafičkog sučelja dinamički se mijenja u ovisnosti o unesenim parametrima.

U vježbi će se proučavati efekti diskretizacije signala s posebnim naglaskom na interpolacija signala. Za te potrebe odabrati opciju **otipkavanje sinusnog signala** (padajući izbornik, desno gore). Frekvenciju otipkavanja podesiti na $f_0=50\text{ Hz}$, a frekvenciju signala na **5 Hz**. Za odabrani set parametara pritiskom na odgovarajući tab pogledati sam signal, $\sin(x)/x$ interpolaciju otipkanih uzoraka (mora se pričekati dok prođe sve uzorke !), rekonstruirani i izvorni signal te njihovu razliku.

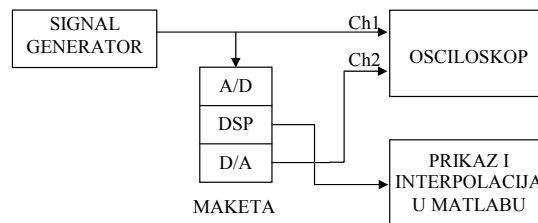
Vidljivo je da je sinusni signal otipkan sa 10 uzoraka po periodi čime je zadovoljen Nyquist-ov kriterij. Uočiti kako se primjenom interpolacijske funkcije na pojedine uzorke i sumacijom pojedinih doprinosa dobiva interpolirani signal. Obratiti pažnju na sljedeće: nakon interpolacije otipkanog signala beskonačnom funkcijom $\sin(x)/x$ bi, prema teoriji, dobiveni rekonstruirani signal morao biti identičan izvornom signalu. Ako se pogleda slika koja prikazuje razliku izvornog i rekonstruiranog signala, uočava se da su dva signala vrlo slična, ali da signal razlike ipak postoji. On je posljedica činjenice da se kod proračuna interpolacijske funkcije u ovom programu zanemaruju doprinosi uzoraka sinusa lijevo i desno od danog (na slici vidljivog) intervala.

Promijeniti frekvenciju signala na **45 Hz** i ponovo pogledati sve karakteristike. Signal koje frekvencije je dobiven interpolacijom ?

5 Hz; $T=0.2\text{ s}$

OTIPKAVANJE I INTERPOLACIJA NA DSP MAKETI

7. Pokrenuti program **alias**. Program služi za podešavanje frekvencije otipkavanja signala na ulazu u DSP maketu (a to je signal iz funkcijskog generatora). Prvo se u korisničkom sučelju odabere frekvencija otipkavanja, a zatim se potrebne instrukcije za podešavanje frekvencije otipkavanja “spuštaju” na DSP maketu. Dakle, u ovom se zadatku od cijele funkcionalnosti DSP makete koristi samo njen A/D i D/A pretvornik, tj. dio zadužen za otipkavanje i interpolaciju signala. DSP procesor će samo prosljeđivati uzorke sa A/D pretvornika na D/A pretvornik. Blok-shema sustava i tok signala za ovu vježbu dani su na slici.

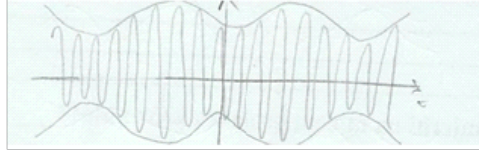


Odabrati frekvenciju otipkavanja iznosa **8 kHz**, resetirati DSP maketu. Nakon svakog resetiranja makete potrebno je pričekati minimalno 7 sekundi kako bi odradio inicijalizacijski program na DSP procesoru. Odabrati opciju **Prijenos programa na DSP maketu**. Odabrana akcija pokreće program na DSP procesoru koji otipkava ulazni signal (vidljiv na kanalu Ch 1 na osciloskopu) s odabranom frekvencijom otipkavanja i zatim takav signal prosljeđuje, bez obrade, na izlaz iz makete (kanal Ch 2 osciloskopa).

a) Dok program radi mijenjati frekvenciju ulaznog **sinusnog** signala u intervalu **od 1 kHz do 10 kHz** i promatrati signal dobiven na izlazu iz DSP sustava. Objasniti zašto i kako se mijenja frekvencija i amplituda izlaznog signala? Može li se to povezati sa amplitudno frekvencijskom karakteristikom A/D pretvornika? Pogledati uvodni dio skripte o DSP maketi, Slika 1.4.

-frekvencija se povećava, amplituda se mijenja. Nakon 3.8 kHz dolazi do preklapanja spektra i amplituda pada

b) Podesiti na ulazu sinus frekvencije **3.8 kHz** te pogledati izlaz iz makete na Ch2 osciloskopa. Kod toga vremensku bazu osciloskopa podesiti na iznos od **1 ms/ds** (milisekunda po dijelu skale), a osjetljivost Y-osi na **0,2 V/ds**. Odabrati mod rada osciloskopa '**NORM**' i **okidanje na Ch 2**. Kanal Ch 1 isključiti (preklopnik Ch1_Both_Ch2 u položaj Ch 2). Potencijometrom '**LEVEL**' podesiti mirnu sliku na osciloskopu. U grubo skicirati sliku u prostor ispod teksta. To što se vidi je rezultat nakon otipkavanja i interpolacije na maketi.



Pritisnuti tipku INTERRUPT na DSP maketi te u programu odabrati opciju **Obrada rezultata sa DSP makete**. Time se prekida izvršavanje programa na DSP maketi, prenose vrijednosti maketom otipkanih uzoraka (njih 4096) u računalu i pokreće MATLAB program za promatranje istih. Program prikazuje otipkane uzorke ulaznog signala, te kroz njih provodi interpolaciju. Promjenom broja promatranih uzoraka (npr. 40 za detaljniji prikaz i 100 za grublji) i rednog broja prvog početnog uzorka pogledati kako izgleda dio otipkanih uzoraka te interpoliranog signala. Skicirati originalni i otipkane uzorke u prostor ispod teksta. Usporediti signal interpoliran u Matlabu sa onim dobivenim na maketi koji je gore precrtan. Komentirati.



U ovom slučaju radi se o otipkavanju signala čija frekvencija je bliska polovici frekvencije otipkavanja (u ovom slučaju 4 kHz). Iako je Nyquist-ov teorem zadovoljen, signal biva uredno otipkan, ali na malo neobičan način. Uzorci su prvo malih vrijednosti, a onda u periodama signala koje slijede postaju sve veći pa se na kraju opet smanjuju. **Uočiti to na prikazu otipkanih uzoraka u Matlabu.** Kod interpolacije u Matlabu nema problema i signal je uredno rekonstruiran jer je korištena širina impulsnog odziva interpolatora proizvoljna i može biti dovoljno velika. S druge strane, stvarni interpolator na maketi ima ograničenja (rad u stvarnom vremenu, zahtjev za malim kašnjenjem sustava) zbog kojih se **ne može koristiti jako širok impulsni odziv interpolatora**. U slučaju interpolacije ovako neobičnih uzoraka kakvi su nastali kada su frekvencija signala i polovica frekvencije otipkavanja bliske, interpolator može još relativno dobro rekonstruirati signal u okolini uzoraka koji su veliki jer su susjedni korišteni uzorci također veliki. Međutim, interpolaciju malih uzoraka ne može ostvariti kako treba zato jer nedostaju doprinosi udaljenih velikih uzoraka (na udaljenostima većim od duljine odziva) koji bi se možda uzimali s malim težinama, ali bi ipak superpozicijom sudjelovali u rekonstrukciji. Zato nastaju propadi (dolovi) u signalu, tj. signal izgleda kao da je moduliran. Pojava je puno manje izražena što je frekvencija ulaznog signala dalje od polovice frekvencije otipkavanja. Zatvoriti prozor odabirom opcije POVRATAK.

Za one koji žele znati i više

c) Za potrebe proučavanja ponašanja D/A pretvornika generirat će se pobudni niz u diskretnoj domeni koji će se periodički slati na ulaz D/A pretvornika. Dakle, u ovom se zadatku ne koristi signal iz funkcijskog generatora već je digitalni signal “umjetno” generiran. Odabrati frekvenciju otipkavanja iznosa 8 kHz, impulsnu pobudu (jedinični impuls) te opciju **Generiranje pobudnog niza**. Na osciloskopu se promatra pravi rezultat, a na računalu rezultat simulacije.

DSP maketa: Na osciloskopu promatrati samo izlazni signal (kanal Ch 2). Namjestiti sliku jednog do dva impulsa odziva.

Proračun u MATLAB-u: Izvorni i u MATLAB-u numerički izračunat interpolirani signal vidljivi su u novo otvorenom MATLAB prozoru.

Zašto je odziv poprimio takav oblik?

Koliko iznosi širina glavne latice (mjeriti na mjestu prolaska kroz nulu)? _____[ms]

d) Odabrati opciju POVRATAK, resetirati maketu, odabrati novu frekvenciju otipkavanja iznosa **5,5125 kHz** te nanovo generirati pobudni niz. Koliko sada iznosi širina glavne latice? _____[ms].

Zašto se gornje dvije vrijednosti razlikuju? U kojoj su vezi te vrijednosti sa frekvencijom otipkavanja?

9. Ponoviti postupak za **sinusnu pobudu** frekvencije **0,48F**, te frekvencijom otipkavanja iznosa **5,5125 kHz**.

DSP maketa: Podesiti sliku na osciloskopu prema točki b). Promatrati samo signal na izlazu iz DSP sistema (Ch 2).

Proračun u MATLAB-u: U novo otvorenom MATLAB prozoru vidljiv je izvorni pobudni niz kao i rezultat interpolacije istoga (ponovo je riječ o simulaciji onoga što se događa na DSP maketi). Po želji podesiti poziciju početnog i posljednjeg promatranog uzorka da se dobro vide i detalji i gruba slika, te pogledati pobudni i interpolirani signal. Parametri koje je moguće mijenjati su **širina impulsnog odziva interpolatora** (broj uzoraka izvornog signala koji će biti obuhvaćeni interpolatorom) i **faktor interpolacije**. Širinom impulsnog odziva interpolatora određujemo kvalitetu interpolacije tj. što je taj broj veći to će za interpolaciju jedne točke biti korišten veći broj otipkanih uzoraka lijevo i desno od te točke čime se dobiva točnija vrijednost interpoliranog uzorka. Idealni interpolator morao bi imati beskonačnu širinu impulsnog odziva. Faktorom interpolacije određujemo broj interpoliranih uzoraka +1 između postojeća dva uzorka. Odabirom većeg broja uzoraka dobivamo kvalitetniji prikaz 'kontinuiranog' signala na ekranu, ali istovremeno produljujemo vrijeme računanja interpolacije. Odabrati širinu impulsnog odziva interpolatora **128** i faktor interpolacije **16**. Objasniti zašto postoji razlika između signala prikazanog na osciloskopu i signala dobivenog interpolacijom u programskom paketu MATLAB. (Frekv. sinusa blizu polovice frekv. otipkavanja!)

10. Mijenjati širinu impulsnog odziva interpolatora i faktor interpolacije. Posebno pogledati slučaj sa širinom impulsnog odziva interpolatora **10** i faktorom interpolacije **16**. Komentirati dobivene rezultate.

2. VJEŽBA

Diskretna Fourier-ova transformacija (DFT)

Bilo koji **aperiodski** vremenski diskretni niz $x[n]$ s uzorcima od $-\infty$ do ∞ može se prikazati u frekvencijskoj domeni pomoću para izraza:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}, \quad (2.1)$$

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) \cdot e^{j\omega n} d\omega. \quad (2.2)$$

U gornjim izrazima $X(e^{j\omega})$ predstavlja Fourier-ovu transformaciju diskretnih signala (*engl. Discrete Time Fourier Transform*, DTFT, a koristi se i kraći termin spektar). Fourier-ova transformacija postoji ako suma u izrazu (2.1) uniformno konvergira, a to uvijek vrijedi ako je suma apsolutnih vrijednosti niza $x[n]$ za sve indekse n konačna, tj.

$$\sum_{n=-\infty}^{\infty} |x[n]| < \infty. \quad (2.3)$$

Dobivena transformacija $X(e^{j\omega})$ je tada kontinuirana funkcija od ω i periodična s 2π .

Za bilo koji beskonačni **periodični** vremenski niz gornji uvjet (2.1) nije zadovoljen i Fourier-ova transformacija u striktnom smislu ne postoji kao kontinuirana funkcija već sadrži sumu delta funkcija na frekvencijama svih harmonika tog periodičnog signala (vidi kasnije). Stoga se takvi signali $\tilde{x}[n]$ (\sim naglašava periodičnost) perioda P često prikazuju pomoću sume P diskretnih kompleksnih eksponencijala frekvencija:

$$\omega_k = \frac{2\pi k}{P}, \text{ za } k=0, \dots, P-1 \quad (2.4)$$

što je dano izrazom:

$$\tilde{x}[n] = \frac{1}{P} \sum_{k=0}^{P-1} c_k e^{j\omega_k n} = \frac{1}{P} \sum_{k=0}^{P-1} c_k e^{j \frac{2\pi k n}{P}}, \quad n = 0, 1, \dots, P-1. \quad (2.5)$$

Koeficijente c_k razvoja u red, tj. kompleksne amplitude pripadnih eksponencijala moguće je naći prema izrazu:

$$c_k = \sum_{n=0}^{P-1} \tilde{x}[n] e^{-j \frac{2\pi k n}{P}}, \quad k = 0, 1, \dots, P-1. \quad (2.6)$$

Zbog analogije sa rastavom periodičnih vremenski kontinuiranih signala, često se izraz (2.5) naziva diskretni Fourier-ov red (*engl. Discrete Fourier Series*, DFS). Kod toga osim periodičnosti diskretnog niza u vremenskoj domeni $x[n] = x[n + rP]$ (r cijeli broj) vrijedi i periodičnost koeficijenata razvoja $c_k = c_{k+rP}$. Bilo kojih P susjednih uzoraka, bilo vremenskog niza, bilo njegovih koeficijenata razvoja jednoznačno opisuje taj periodični niz.

Pomoću ovako određenih koeficijenata diskretnog Fourier-ovog reda moguće je napisati čak i izraz za Fourier-ovu transformaciju $X(e^{j\omega})$ takvog periodičnog signala beskonačnog trajanja i beskonačne energije i to pomoću izraza:

$$X(e^{j\omega}) = \sum_{r=-\infty}^{\infty} \sum_{k=0}^{P-1} \frac{2\pi}{P} \cdot c_k \cdot \delta(\omega - \omega_k + 2\pi \cdot r) . \quad (2.7)$$

Ovaj izraz se ne može dobiti na osnovu (2.1) zbog problema konvergencije, ali se može pokazati da se njegovim uvrštavanjem u izraz za inverznu Fourier-ovu transformaciju (IDTFT) (2.2) dobiva polazni beskonačni periodični signal $\tilde{x}[n]$.

Fourier-ova transformacija vremenski diskretnih aperiodskih signala konačnog trajanja je kontinuirana funkcija od frekvencije ω , prema (2.1). Radi predstavljanja takve funkcije na računalima pogodno je koristiti njene uzorke, a ne funkciju. Matematički postupak kojim je moguće odrediti N **uzoraka Fourier-ove transformacije** $X(e^{j\omega})$ nekog **aperiodskog** signala konačnog trajanja naziva se diskretna Fourier-ova transformacija (*engl. Discrete Fourier Transform*, DFT) i dana je izrazima:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi n k}{N}} , \quad (2.8)$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j \frac{2\pi n k}{N}} . \quad (2.9)$$

Radi jednostavnosti izrazi se često zapisuju u obliku:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-nk} , \quad X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk} , \quad (2.10)$$

gdje je W_N dan izrazom:

$$W_N = e^{-j \frac{2\pi}{N}} . \quad (2.11)$$

Otipkavanje Fourier-ove transformacije izvedeno je u N ekvidistantnih točaka s frekvencijama ω_k , $0 \leq \omega_k < 2\pi$, tj. vrijedi:

$$X[k] = X(e^{j\omega}) \Big|_{\omega=\omega_k=2\pi k/N} , \quad k=0, \dots, N-1 . \quad (2.12)$$

Drugim riječima, koeficijenti DFT transformacije $X[k]$ predstavljaju uzorke od $X(e^{j\omega})$ na frekvencijama ω_k . Ulaz u DFT su uzorci diskretnog niza sa indeksima $n = 0, 1, \dots, N-1$.

Treba napomenuti da broj uzoraka DFT-a mora biti jednak ili veći od dužine aperiodskog niza R , dakle $N \geq R$. Ako je N strogo veći od R , niz tada treba nadopuniti nulama tako da je konačna dužina niza jednaka N . Za kauzalne je nizove postupak očit jer nule treba staviti iza zadnjeg uzorka originalnog niza, ali je situacija nešto složenija za nekauzalne nizove. Kod njih treba odgovarajući broj nula staviti također iza zadnjeg uzorka s pozitivnim indeksima, ali nakon nula treba staviti uzorke s negativnim indeksima i to tako da uzorak sa indeksa -1 dolazi na indeks $N-1$, sa indeksa -2 na $N-2$ i tako dalje. Na taj način je jedan **nekauzalni aperiodski niz pripremljen za analizu primjenom DFT-a** uz očuvanje njegovih faznih svojstava.

Uz uvjet $N > R$ se DFT-om dobije **više uzoraka otipkanog spektra** što daje vjerniju sliku o samom spektru niza $X(e^{j\omega})$. Da bi i uz veliki broj uzoraka DFT-a izračun bio brz, razvijen je niz algoritama koji su po krajnjem rezultatu potpuno ekvivalentni DFT-u, ali su učinkovitiji, a time i brži. Takvi algoritmi nazivaju se jednim imenom brza Fourier-ova transformacija (*engl. Fast Fourier Transform, FFT*).

S obzirom da izrazi za DFS (2.5) i (2.6) definirani za periodične diskretne signale izgledaju identično kao oni za DFT (2.8) i (2.9) kojim se predstavljaju aperiodski diskretni signali konačnog trajanja (osim imena c_k i $X[k]$, te periode P odnosno broja uzoraka N), postavlja se pitanje njihove veze. Može se pokazati da se svaki aperiodski diskretni signal trajanja N može pretvoriti u beskonačno periodičan tako da se ponavlja dotičnih N uzoraka niza, a sam aperiodski signal se uvijek može dobiti kao samo jedna perioda takvog periodiziranog signala. Takav postupak periodizacije u vremenskoj domeni rezultira diskretnim spektrom pa samim time i razvoj u DFS ima smisla. Možemo zaključiti da u definiciji DFT reprezentacije aperiodskog signala jednostavno prepoznamo da nas interesiraju samo vrijednosti $x[n]$ u intervalu $n = 0, 1, \dots, N-1$ jer znamo da su druge jednake nuli. Isto tako nas interesiraju samo vrijednosti $X[k]$ u intervalu $k = 0, 1, \dots, N-1$ jer su one jedine potrebne u izrazu (2.9). Međutim, ono što je važno i treba zapamtiti jest da iako je DFT de facto transformacija koja se koristi u praksi, rastav u DFS predstavlja teoretsku podlogu. **Inherentna periodičnost DFS transformacije ostaje i kod DFT-a, i u vremenskoj i u frekvencijskoj domeni.** To drugim riječima znači da iako znamo da imamo aperiodski signal, inverznim DFT-om on biva periodiziran.

Iako se po definiciji koristi za konačne aperiodske nizove, **pomoću DFT-a možemo odrediti i harmonijski sastav beskonačnog periodičnog signala** (tj. koeficijente njegovog DFS rastava) pod uvjetom da je **broj uzoraka DFT-a N jednak ili cjelobrojni višekratnik periode P** tog periodičnog niza. Ako to nije zadovoljeno, uzorci DFT-a $X[k]$ ne odgovaraju koeficijentima c_k i ne pokazuju očekivani harmonijski sastav. Jednostavno objašnjenje toga je činjenica da DFT pretpostavlja periodičko ponavljanje upravo tih N uzoraka, a pošto nije obuhvaćen cijeli broj perioda ulaznog niza, periodizirani niz ima neočekivane skokove na rubovima ponovljenih blokova. Činjenica da se radi o potpuno drugom periodičnom signalu od onog kojeg smo imali na početku uzrokuje i pojavu drugačijeg harmonijskog sastava.

Za uzorke DFT-a moguće je izračunati amplitudu $|X[k]|$ i fazu $\angle X[k]$ koji predstavljaju otipkanu amplitudnu i faznu karakteristiku (spektar) od $X(e^{j\omega})$ prema izrazima:

$$|X[k]| = \sqrt{\text{Re}\{X[k]\}^2 + \text{Im}\{X[k]\}^2}, \quad \angle X[k] = \text{atan2}(\text{Im}\{X[k]\}, \text{Re}\{X[k]\}) \quad (2.13)$$

gdje je sa atan2 označena 4-kvadrantna arctg funkcija od imaginarnog i realnog dijela kompleksnih uzoraka DFT-a $X[k]$.

U nastavku će biti opisana neka od svojstava DFT-a.

I. Realni vremenski niz

Koeficijenti DFT-a **realnog vremenskog niza** imaju slijedeće svojstvo

$$X[k] = X^*[-k], \quad (2.14)$$

tj. koeficijenti, simetrični oko 0, su konjugirano kompleksni, odnosno realni dio im je jednak, a imaginarni dio suprotnog predznaka

$$\text{Re}\{X[k]\} = \text{Re}\{X[-k]\}, \quad \text{Im}\{X[k]\} = -\text{Im}\{X[-k]\}. \quad (2.15)$$

Zapisano preko modula (magnituda) i faze

$$|X[k]| = |X[-k]|, \quad \angle X[k] = -\angle X[-k] \quad (2.16)$$

proizlazi da simetrični koeficijenti imaju jednake module, a pripadne faze su suprotnog predznaka. Zbog toga je za realne vremenske nizove dovoljno poznavati samo prvih $N/2 + 1$ koeficijenata DFT-a.

Napomena

Važno je ovdje prokomentirati što u stvari znači simetričnost ili antisimetričnost DFT-a. Već je rečeno da DFT za ulaz ima N uzoraka vremenskog niza između 0 i $N-1$. Rezultat DFT je niz uzoraka $X[k]$ koji su također između 0 i $N-1$. Prema izrazu (2.12) ti uzorci odgovaraju frekvencijama $0 \leq \omega_k < 2\pi$. Simetričnost/antisimetričnost je klasično definirana s obzirom na ishodište (u ovom slučaju, frekvenciju 0) pa bi promatrani raspon frekvencija trebao u stvari biti $-\pi \leq \omega_k < \pi$. Vidimo da uzorci DFT ne pokrivaju taj raspon. Ovdje stoga treba uzeti u obzir periodičnost DFT-a što znači da su uzorci DFT-a koji odgovaraju frekvencijama $\pi \leq \omega_k < 2\pi$ upravo jednaki onima u rasponu $-\pi \leq \omega_k < 0$. Dovoljno je, dakle, zamisliti da se **desna polovica DFT uzoraka na indeksima k od $N/2$ do $N-1$ (za N paran) nalazi i lijevo od ishodišta, na indeksima $-N/2$ do -1** . Uz ovo gornji uvjeti imaju smisla.

II. Simetrični/antisimetrični vremenski nizovi

Ako je realni vremenski niz dužine N **simetričan** (paran), tj. vrijedi:

$$x[n] = x[-n] \quad , \quad \text{za } n=1, \dots, \frac{N}{2} - 1 \quad (2.17)$$

tada su koeficijenti DFT-a **čisto realni** (imaginarni dio je jednak 0, $\text{Im}\{X[k]\} = 0$).

Analogno, ako je vremenski niz **antisimetričan** (neparan):

$$x[n] = -x[-n] \quad , \quad \text{za } n=1, \dots, \frac{N}{2} - 1 \quad \text{i} \quad x[0] = 0 \quad (2.18)$$

tada su koeficijenti DFT-a čisto imaginarni (realni dio je jednak 0, $\text{Re}\{X[k]\} = 0$).

Napomena

Ponovno treba dodati komentar da bi uvjet bio jasniji. Definicija simetričnosti općenito je vezana uz izgled signala s lijeve i desne strane ishodišta. Nasuprot tome, ulazni niz u DFT mora biti definiran za indekse od 0 do $N-1$. Ovdje također možemo primijeniti inherentnu periodičnost DFT transformacije i to onaj dio koji se odnosi na periodičnost u vremenskoj domeni. Naime, N uzoraka ulaznog niza predstavlja samo ono što gledamo, međutim DFT podrazumijeva da se tih N uzoraka periodički ponavlja u \pm beskonačnost. Ako se provede takva periodizacija, dobivamo signal koji s obzirom na ishodište može biti simetričan, asimetričan ili niti jedno od toga.

Iz ove diskusije jasniji je i način kako ostvariti **DFT nekauzalnog aperiodskog niza** koji ima uzorke i lijevo i desno od nultog indeksa. Kao što je rečeno, da bismo uopće mogli izračunati DFT tog niza, potrebno ga je prikazati između uzoraka 0 do $N-1$. Ako nekauzalni niz periodiziramo, uzorci sa negativne osi pojaviti će se oko indeksa $N-1$ i imat ćemo niz od kojeg možemo raditi DFT.

III. Pomak vremenskog niza

Najprije razmotrimo svojstvo pomaka vremenskog niza za Fourier-ovu transformaciju vremenski diskretnog signala (**DTFT**). Uzmimo diskretni niz $x[n]$, te ostvarimo drugi niz, pomaknut u odnosu na ovaj originalni za m koraka (m je cijeli broj), $x_m[n] = x[n - m]$. Ako je

DTFT prvog niza jednak $X(e^{j\omega})$, tada za DTFT niza pomaknutog za m koraka $X_m(e^{j\omega})$, vrijedi:

$$X_m(e^{j\omega}) = X(e^{j\omega})e^{-j\omega \cdot m}. \quad (2.19)$$

Pošto je modul od $e^{-j\omega \cdot m}$ jednak 1, moduli DTFT-a originalne i pomaknute sekvence su jednaki, a faze se razlikuju, jer član $e^{-j\omega \cdot m}$ unosi linearni fazni pomak.

Za **DFS transformaciju** koja podrazumijeva ulazni beskonačno **periodični** niz $\tilde{x}[n]$ perioda P vrijedi analogno svojstvo koje se može opisati izrazom:

$$c_{k,m} = c_k \cdot e^{-j\frac{2\pi k}{P} \cdot m} = c_k \cdot W_P^{k \cdot m} \quad (2.20)$$

gdje su c_k i $c_{k,m}$ DFS koeficijenti originalnog niza $\tilde{x}[n]$, te niza pomaknutog za m koraka $\tilde{x}[n-m]$. Sam pomak niza ne predstavlja nikakav problem s obzirom da je niz periodičan pa znamo da uzorci postoje i izvan raspona od 0 do $P-1$.

DFT transformacija je definirana za aperiodeske diskretne nizove $x[n]$ dužine N . Za nju također vrijedi gornje svojstvo koje se može zapisati kao:

$$X_m[k] = X[k]W_N^{km} \quad (2.21)$$

gdje su $X[k]$ i $X_m[k]$ DFT transformacije originalnog $x[n]$ niza s vrijednostima definiranim za indekse $n = 0, 1, \dots, N-1$, te niza pomaknutog za m koraka $x[n-m]$. U slučaju DFT-a, međutim, treba biti jako oprezan što znači pomak niza. Pošto se radi o aperiodeskom signalu, definiranom od 0 do $N-1$, a za sve druge vrijednosti jednak 0, to bi možda značilo da je pomakom u vremenu dio originalnog niza zauvijek izgubljen. To naravno nije tako posebice ako se podsjetimo na periodičnost u vremenu 'naslijeđenu' od DFS transformacije, a koja vrijedi i za DFT. Naime, ako originalni aperiodeski niz u glavi periodiziramo s periodom N , pomaknemo za m koraka i od takvog pomaknutog, periodiziranog niza uzmemo samo uzorke na indeksima od 0 do $N-1$, dobili smo pomaknuti aperiodeski niz. Ono što se de facto događa jest da se **aperiodeski niz pomiče cirkularno unutar N uzoraka** što odgovara operaciji modulo N . Dakle svojstvo (2.21) vrijedi za signal $x_m[n] = x[(n-m)_{\text{mod } N}]$.

IV. Signali u obliku periodičnih impulsa

Kod periodičnih signala koji imaju oblik niza impulsa, tj. kod kojih je mnogo uzoraka jednakih 0, spektar je širok. Gotovo svi uzorci DFS-a su različiti od 0. Vrijedi i dualno svojstvo, tj. da što je impuls u vremenu širi, to mu je spektar uži (sve više koeficijenata DFS-a je približno jednako 0). Diskretna periodična funkcija pogodna za promatranje ove pojave ima oblik Gauss-ove funkcije i definirana je na jednom periodu P izrazom:

$$x[n] = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{2(n-P/2)^2}{P^2\sigma^2}}, \quad n=0, \dots, P-1. \quad (2.22)$$

Promjenom parametra σ mijenja se širina Gauss-ovog zvona (što je σ manji, zvono je uže). Modul koeficijenata DFS-a ima ponovno oblik zvona, ali što je zvono u vremenu uže, zvono u Fourier-ovoj domeni je šire i obratno.

V. DFT diskretnog kosinusnog niza

Promatrajmo vremenski diskretni kosinusni niz amplitude A , početne faze φ_0 i frekvencije ω_0 , danog sa:

$$\tilde{x}[n] = A \cos(\omega_0 n + \varphi_0) = A \cos(2\pi / P \cdot m_0 \cdot n + \varphi_0), \quad \text{za } n = -\infty \text{ do } \infty \quad (2.23)$$

gdje su m_0 i P cijeli brojevi, $0 \leq m_0 \leq P/2$. Broj P predstavlja osnovni period takvog kosinusnog niza ili njegov višekratnik u slučaju da m_0 i P nisu koprim brojevi. Takav signal sadrži točno m_0 cijelih perioda kosinusnog signala unutar P susjednih uzoraka. Jasno je da se radi o beskonačnom periodičnom nizu. U skladu sa već rečenim (vidi izraz (2.7)), Fourier-ova transformacija (spektar, DTFT) $X(e^{j\omega})$ takvog niza ima oblik:

$$X(e^{j\omega}) = \sum_{r=-\infty}^{\infty} 2\pi \cdot \left(\frac{A}{2} e^{j\varphi_0} \cdot \delta(\omega - \omega_0 + 2\pi \cdot r) + \frac{A}{2} e^{-j\varphi_0} \cdot \delta(\omega + \omega_0 + 2\pi \cdot r) \right), \quad (2.24)$$

tj. ima delta funkcije na frekvencijama $\omega = \pm\omega_0 + 2r\pi$, $-\infty < r < \infty$, čije su kompleksne amplitude određene amplitudom A i početnom fazom φ_0 , dok je za sve ostale frekvencije $X(e^{j\omega})$ jednako 0.

Da bismo uopće mogli izračunati DFT, moramo ograničiti kosinusni niz i promatrati samo jedan njegov dio. To ograničavanje niza u vremenskoj domeni odgovara množenju kosinusnog niza sa uzorcima jednog interesantnog i korisnog tipa vremenski diskretnog niza koji se naziva **vremenski otvor** (*engl. window*). Postoji više vrsta vremenskih otvora koji se mogu koristiti za ovu namjenu, a najjednostavniji je pravokutni vremenski otvor (*engl. rectangular window*). Pravokutni otvor definiran je sa:

$$w[n] = \begin{cases} 1, & n=0, \dots, L-1 \\ 0, & \text{inace} \end{cases}, \quad (2.25)$$

tj. on ima uzorke jednake 1 samo za L uzoraka dok su svi ostali jednaki 0. Množenjem beskonačnog, kosinusnog niza sa ovim otvorom $w[n]$ dužine L uzoraka dobiva se konačan, aperiodski diskretni niz $x_L[n]$ koji ima vrijednost 0 za $n < 0$ i $n \geq L$, a za $n = 0, 1, \dots, L-1$ ima uzorke čije su vrijednosti jednake onima originalne kosinusoide $\tilde{x}[n]$. U praksi, ako se za ograničavanje signala koristi pravokutni vremenski otvor nije potrebno provoditi množenje već se jednostavno izdvoji L uzoraka te kosinusoide.

U spektralnoj domeni, operacija množenja uzoraka niza i vremenskog otvora odgovara tzv. periodičkoj konvoluciji danoj izrazom:

$$X_L(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\theta}) \cdot W(e^{j(\omega-\theta)}) d\theta, \quad (2.26)$$

gdje je $X_L(e^{j\omega})$ spektar ograničenog niza, $X(e^{j\omega})$ spektar periodičnog niza dan izrazom (2.24), a $W(e^{j\omega})$ spektar vremenskog otvora. Za pravokutni vremenski otvor vrijedi izraz:

$$W(e^{j\omega}) = \sum_{n=0}^{L-1} 1 \cdot e^{-j\omega \cdot n} = e^{-j\omega \cdot (L-1)/2} \cdot \frac{\sin(\omega L/2)}{\sin(\omega/2)} \quad (2.27)$$

$W(e^{j\omega})$ je krivulja s glavnom laticom i bočnim laticama (titrajima) koja prolazi kroz 0 u točkama s frekvencijom koje su višekratnici frekvencije $2\pi/L$, dok je vrijednost spektra otvora za $\omega = 0$ jednaka L . Gornji izraz opisuje spektar kauzalnog pravokutnog otvora

pozicioniranog na indeksima 0 do $L-1$. Eksponecijalni član ispred omjera sinusnih funkcija posljedica je pomaka otvora u vremenu za $(L-1)/2$ koraka u desno i predstavlja linearnu faznu karakteristiku s nagibom $-(L-1)/2$. Periodičkom konvolucijom se periodični **spektar otvora ponavlja na svim frekvencijama na kojima se nalaze delta funkcije** od $X(e^{j\omega})$ u rasponu frekvencija $\omega \in [-\pi, \pi)$ te tako nastaje spektar $X_L(e^{j\omega})$ dan izrazom:

$$X_L(e^{j\omega}) = \frac{A}{2} e^{j\varphi_0} \cdot W(e^{j(\omega-\omega_0)}) + \frac{A}{2} e^{-j\varphi_0} \cdot W(e^{j(\omega+\omega_0)}). \quad (2.28)$$

Ako od L uzoraka $x_L[n]$, $n = 0, \dots, L-1$ sada napravimo DFT u $N=L$ točaka, dobivamo N kompleksnih koeficijenata $X_L[k]$, $k = 0, \dots, N-1$ koji su identični vrijednostima funkcije $X_L(e^{j\omega})$ za $\omega = 2\pi k / N$. Varijabla k se u kontekstu DFT-a često u engl. literaturi naziva *bin*.

U slučaju kada vrijedi $L = P = N$, tj. kada smo vremenskim otvorom izdvojili točno jedan osnovni period signala koji sadržava m_0 perioda diskretne kosinusoide, te kada od tih L uzoraka izračunamo DFT u istom broju točaka, uzorci DFT-a su jednaki:

$$X[k] = \begin{cases} L \cdot A/2 \cdot e^{j\varphi_0} & k = m_0 \\ L \cdot A/2 \cdot e^{-j\varphi_0} & k = N - m_0 \\ 0 & \text{ostale } k \end{cases} \quad (2.29)$$

Samo uz ovaj odnos L , P i N , **uzorci DFT-a upravo odgovaraju koeficijentima DFS razvoja** pa izraz (2.29) predstavlja osnovu za korištenje DFT-a u svrhu analize spektralnog sastava periodičnih signala jer je vidljivo da je u koeficijentima DFT-a sadržana informacija i o amplitudi i o početnoj fazi svakog pojedinog harmonika ulaznog niza. U slučaju da se diskretni niz sastoji od više harmonijskih komponenata zajedničkog osnovnog perioda P , tada svakoj kosinusoidi na binu m_r odgovara jedan par DFT koeficijenata $X[m_r]$ i $X[N - m_r]$. Gornji izraz ne vrijedi u slučaju istosmjernog signala ($m_0 = 0$) ili signala frekvencije $\omega_0 = \pi$, tj. $m_0 = P/2$. Tada se dvije kompleksne eksponencijale preklapaju pa postoji samo jedan uzorak DFT-a čija je amplituda jednaka LA .

U slijedeća dva paragrafa bit će analiziran **utjecaj broja uzoraka DFT-a N** na interpretaciju samih DFT uzoraka. Ako sa L uzoraka obuhvatimo točno jedan osnovni period niza $\tilde{x}[n]$ (tj. vrijedi $P=L$), a izračunajmo **DFT u broju točaka N , koji je višekratnik od L** , tj. $N = r \cdot L$ (nadopunjavanjem niza s nulama), tada će svaki r -ti uzorak novog DFT-a biti jednak uzorcima originalnog DFT-a izračunatog u L točaka. Između tih r -tih višekratnika pojaviti će se $r-1$ novih uzoraka koji su različiti od 0. Ako je npr. $N=2L$, između parnih uzoraka, pojaviti će se još po jedan novi različit od 0. Ova situacija de facto odgovara **gušćem otipkavanju spektra** aperiodskog signala $x_L[n]$, čime se izgledom skupa uzoraka DFT-a približavamo stvarnom izgledu samog spektra $X_L(e^{j\omega})$.

Ako je uz isti uvjet $P=L$, **N proizvoljni broj veći od L** (dakle nije njegov višekratnik), i dalje se DFT-om dobivaju uzorci spektra $X_L(e^{j\omega})$ u ekvidistantnim točkama $\omega_k = 2\pi / N \cdot k$, $k = 0, \dots, N-1$, ali se ove frekvencije više ne poklapaju s frekvencijama harmonika ulaznog signala. Krivulja $X_L(e^{j\omega})$ je dakle otipkana na mjestima koja nam nisu pogodna za određivanje spektralnog sastava signala.

Na kraju, analizirajmo i utjecaj **veličine vremenskog otvora L** na značenje dobivenih DFT uzoraka. Ako je sa L uzoraka ograničenog kosinusnog niza $x_L[n]$ obuhvaćeno **više cijelih osnovnih perioda P** originalnog periodičnog niza od kojeg se krenulo, tj. vrijedi $L = i \cdot P$ gdje je i prirodni broj, kružna frekvencija ω_0 ulaznog diskretnog niza $\tilde{x}[n]$ se može izraziti kao:

$$\omega_0 = \frac{2\pi}{P} \cdot m_0 = \frac{2\pi}{L} \cdot i \cdot m_0 = \frac{2\pi}{L} \cdot l_0 \quad , \quad l_0 \text{ je cijeli broj,} \quad (2.30)$$

dakle jednaka je i višekratniku od $2\pi/L$. Taj višekratnik l_0 predstavlja upravo broj cijelih perioda diskretne kosinusoide na L uzoraka. U slučaju da se DFT izvodi u N točaka uz $N=L$, uzorci amplitude DFT-a dotičnog niza imati će sve uzorke jednake nuli osim onih na indeksu $k = l_0$ i $N - l_0$. Drugim riječima, za neki periodični niz sa proizvoljnim harmoničkim sastavom osnovnog perioda P , tek **svaki i -ti uzorak DFT-a odgovara DFS koeficijentima**.

Razmotrimo i slučaj kada uzorcima $x_L[n]$ od 0 do $L-1$ **nemamo obuhvaćen cijeli broj perioda** od $\tilde{x}[n]$, tj. odnos L i P je proizvoljan. Tada kružna frekvencija ω_0 diskretnog niza $\tilde{x}[n]$ više nije višekratnik od $2\pi/L$. Ako je ulazni niz bio samo jedna kompleksna eksponencijala zadane frekvencije ω_0 , to ne izaziva neke poteškoće. Problem je međutim ako imamo proizvoljni periodični niz osnovnog perioda P koji se sastoji od P takvih kompleksnih eksponencijala na frekvencijama $\omega_0 = 2\pi/P \cdot m_0$, $m_0 = 0, \dots, P-1$. Diskretna kosinusoida koja se analizira u ovom tekstu, također spada u ovu klasu jer se sastoji od dvije takve kompleksne eksponencijale (prema $\cos(\omega_0 n) = 1/2 \cdot (e^{j\omega_0 n} + e^{-j\omega_0 n})$) jedna eksponencijala je na pozitivnoj frekvenciji, a druga na negativnoj frekvenciji). Uslijed množenja sa vremenskim otvorom proizvoljne širine L u odnosu na P , spektar otvora ne prolazi kroz 0 na frekvencijama harmonika pa dolazi do međusobne interakcije harmonika u spektru $X_L(e^{j\omega})$. Ova pojava onemogućava analizu harmonijskog sastava takvog periodičnog signala primjenom DFT-a. Pojava se naziva '**razmazivanje**' spektra, a može se interpretirati i činjenicom da se periodizacijom aperiodskog niza $x_L[n]$ u stvari dobiva neki novi periodični signal, a ne polazni periodični niz $\tilde{x}[n]$.

Realizacija na DSP maketi

U dijelu vježbe s DSP maketom pokazat će se usporedba idealnih frekvencijski ograničenih signala i realno otipkanih. Time će se pokazati neki efekti otipkavanja na izgled DFT-a. U svim promatranim slučajevima, DFT se računa na vremenski ograničenim kosinusnim signalima pa vrijedi sve rečeno pod stavkom V. iz teoretskog dijela vježbe. Kako je u vježbi broj uzoraka otvora fiksiran, u općem slučaju nema cijeli broj perioda na promatranom odsječku signala pa dolazi do razmazivanja spektra (frekvencije komponenata od kojih je sastavljen signal ne podudaraju se s frekvencijama spektralnih komponenti koje postoje u DFT transformaciji).

Na vježbama će se promatrati dva signala istog oblika i frekvencije, ali dobivena na dva različita načina. **Prvi signal dobiven je otipkavanjem analognog signala** iz generatora funkcija ugrađenog u DSP maketu. Pošto je to realan signal on osim potrebnih harmonika osnovne frekvencije, koji su sastavni dio promatranog signala, sadrži i neke druge frekvencijske komponente koje su prigušene za neki iznos u odnosu na osnovne, ali doprinose iznosu ukupnog signala (u biti uzrokuju izobličenje signala). Pojava takvih komponenata posljedica je različitih nelinearnih efekata unutar generatora funkcija. Nadalje, kao posljedica filtracije određene ulaznom karakteristikom A/D pretvornika (antialiasing filter) dolazi do

odbacivanja harmonijskih komponenti koje se nalaze iznad polovice frekvencije otipkavanja. To odbacivanje nije idealno jer radi realnog filtra ipak dolazi do preslikavanja nekih komponenti u blizini polovine frekvencije otipkavanja. Te komponente su dosta prigušene što filtrom, što samim smanjenjem iznosa harmonika te ne utječu u nekoj većoj mjeri na signal. Proučavanjem spektra otipkanog signala možemo ustanoviti kvalitetu generatora funkcija te korištenog A/D pretvornika. Sama usporedba signala se na vježbama provodi na različitim frekvencijama za tri valna oblika (sinus, trokut i pravokutnik).

Kao **drugi (referentni) signal uzima se u Matlabu konstruirani signal** dobiven sintezom osnovnih harmonijskih komponenti, koje proizlaze iz idealnog (matematičkog) rastava signala po frekvencijskim komponentama tj. razvoja u Fourier-ov red. Matematički izrazi za promatrane valne oblike su slijedeći:

$$\text{Sinus:} \quad y(t) = \sin(2 \cdot \pi \cdot f \cdot t) \quad (2.31)$$

$$\text{Trokut:} \quad y(t) = \frac{8}{\pi^2} \cdot \sum_{n=1}^{\infty} (-1)^n \frac{\sin(2 \cdot \pi \cdot f \cdot (2 \cdot n - 1) \cdot t)}{(2 \cdot n - 1)^2} \quad (2.32)$$

$$\text{Pravokutnik:} \quad y(t) = \frac{4}{\pi} \cdot \sum_{n=1}^{\infty} \frac{\sin(2 \cdot \pi \cdot f \cdot (2 \cdot n - 1) \cdot t)}{2 \cdot n - 1} \quad (2.33)$$

Iz izraza je vidljivo da se trokutni i pravokutni signal sastoje od beskonačnog broja viših neparnih harmonika osnovne frekvencije f . U skladu sa realnim slučajem signala, pri konstrukciji referentnog signala uzimaju se samo one komponente koje su manje od polovice frekvencije otipkavanja. Ispuštanje viših harmonijskih komponenti uzrokuje izobličenje konstruiranog signala što je vidljivo na njegovom vremenskom prikazu i to naročito u slučaju kada je sintezom obuhvaćen mali broj frekvencijskih komponenti.

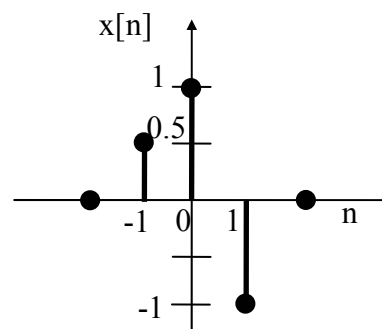
Zadatak za pripremu

1) Izvesti analitički izraz za Fourierovu transformaciju $X(e^{j\omega})$ aperiodskog, vremenski diskretnog niza $x[n]$ prikazanog slikom, koji je definiran za indekse $-1 \leq n \leq 1$, a za sve druge indekse je jednak 0. Odrediti kolika je vrijednost $X(e^{j\omega})$ za $\omega = \pi/4$.

2) Imajući u vidu da je niz **nekauzalni** odrediti diskretnu Fourierovu transformaciju $X[k]$ niza $x[n]$ u $N=4$ točke (pogledati DFT nekauzalnog aperiodskog niza iza definicije DFT-a u teoretskom dijelu vježbe). Odrediti i nacrtati amplitudu i fazu uzoraka DFT-a.

3) Odrediti DFT $X_c[k]$ kauzalnog niza $x_c[n]$ koji ima iste vrijednosti kao $x[n]$, ali je definiran za indekse $0 \leq n \leq 2$, a za ostale je 0. Ovaj niz de facto predstavlja pomaknuti niz za jedan korak u desno u odnosu na početni. Provjeriti da li za $X[k]$ i $X_c[k]$ vrijedi svojstvo pomaka vremenskog niza opisano u stavci III. teoretskog dijela vježbe.

Rješenje zadatka:



Praktična vježba

MATLAB PROGRAM ZA IZRAČUNAVANJE DFT-a APERIODSKOG NIZA

1. Za aperiodski, vremenski diskretan niz $x[n]$ zadan u zadatku za pripremu napisati Matlab program (skriptu) kojim se izračunavaju i prikazuju Fourierova transformacija vremenski diskretnog signala (DTFT) $X(e^{j\omega})$ i diskretna Fourierova transformacija (DFT) $X[k]$ tog diskretnog niza. Slijediti upute u nastavku.

a) DTFT je kontinuirana funkcija frekvencije ω pa se kao takva ne može direktno izračunati i prikazati korištenjem Matlaba. Ono što se međutim može napraviti jest izračunati i prikazati DTFT kao diskretni niz u vrlo velikom broju točaka. U skladu s tim treba najprije definirati broj frekvencija K u koliko će se DTFT izračunati. Neka je $K=1024$. Potrebne frekvencije ω ostvariti u rasponu $0 \leq \omega < 2\pi$ (jer 2π već predstavlja slijedeću periodu). Koristiti naredbu `om=[0:K-1]/K*2*pi` kojom se definira vektor `om` sa K elemenata čije vrijednosti su od 0 do malo manje od 2π .

b) Napisati u Matlabu izraz za DTFT prema analitičkom izrazu dobivenom u 1) dijelu rješenja zadatka iz pripreme. Označiti DTFT npr. imenom `xs`. Za imaginarnu jedinicu $\sqrt{-1}$ uzeti `j`. Koristiti bilo funkcije `sin()` i `cos()`, bilo `exp()`. Imati u vidu da pošto se u izrazu za `xs` kao varijabla pojavljuje vektor (`s` obzirom da se radi o DTFT-u to je vektor `s` frekvencijama `om`), sam `xs` je vektor iste veličine s vrijednostima DTFT-a za svaki element vektora `om`.

c) Izračunati amplitudni i fazni spektar **na temelju DFTF-a** `xs` korištenjem Matlab funkcija `abs()` (za amplitudni) i `angle()` (za fazni spektar). Osim toga izračunati realni i imaginarni dio spektra Matlab funkcijama `real()` i `imag()`.

d) Broj točaka N u koliko se izračunava DFT unijeti korištenjem naredbe `input`. Kod testiranja, može se odabrati $N=4$. Napraviti vektor `k` s indeksima DFT-a u rasponu $0 \leq k \leq N-1$.

e) Prema dijelu rješenja zadatka iz pripreme pod 2) realizirati u Matlabu diskretni niz $x_D[n]$ koji predstavlja ulaz u DFT, a dobiven je iz $x[n]$. Potrebni broj nula do ukupnih N točaka ostvariti pomoću naredbe `zeros(1,N-3)` (originalni niz ima 3 uzorka) te ih postaviti na pravo mjesto u vektoru.

f) Izračunati DFT korištenjem naredbe `fft(xD,N)` koja izračunava DFT u N točaka bilo pomoću učinkovitog FFT postupka ako je N potencija broja 2, bilo normalno na manje učinkovit način ako je N neki drugi broj.

g) Korištenjem istih naredbi kao u točki c) izračunati amplitudu i fazu te realni i imaginarni dio **DFT uzoraka**.

h) Slijede naredbe za prikaz. **Prije svake slike** pozvati funkciju `figure` za otvaranje novog prikaza. Na **prvoj slici** prikazati niz x_D pomoću naredbe `stem` uz odgovarajuće indekse `n=[0:N-1]`.

i) Na **drugoj slici** pomoću naredbe `subplot` (kojom se prikazuje više odvojenih grafova na jednoj slici) prikazati amplitudu i fazu DTFT-a i DFT uzoraka na način kako slijedi. Kod prikaza najprije sa `subplot(2,1,1)` pristupiti **prvom (gornjem) grafu**. Korištenjem naredbe oblika `plot(x1,y1,'-',x2,y2,'o')` na grafu prikazati **amplitudu DTFT-a punom linijom ('-')** te **amplitudu DFT-a simbolom ('o')**. Os apscisa za DTFT je vektor `om`, a za DFT bi trebao biti vektor `k`. Međutim, da bi obje amplitudne karakteristike

mogli prikazati na istom grafu što želimo učiniti, moramo vrijednosti indeksa DFT-a k , $0 \leq k \leq N-1$ prikazati kao funkciju frekvencije ω , $0 \leq \omega < 2\pi$, tj. kao $k \cdot 2\pi/N$. Broj točaka se naravno ne mijenja. Sa `ylabel('staviti_text')` označiti y-os ovog grafa da se zna što prikazuje. Pristupiti zatim sa `subplot(2,1,2)` **donjem grafu** i na analogan način prikazati **faze DTFT-a i DFT uzoraka**. Na kraju izgenerirati naslov za cijelu sliku koji ispiše vrijednosti korištenih K i N . Za to se koristi naredba `str=sprintf('K=%d, N=%d',K,N);`, koja će vrijednosti K i N ubaciti u string `str`, a zatim taj string treba postaviti na sliku naredbom `title(str)`.

j) Analogno prethodnom koraku, na **trećoj slici** na isti način kao pod stavkom i) prikazati **realne i imaginarne dijelove DTFT-a i DFT uzoraka**.

2. Izvesti program uz $N=4$ i promotriti dobivene prikaze. Zatim izvesti program uz $N=16$. Što se promijenilo sa prikazom DFT-a ?

DFT ima veću redukciju pri $N = 16$

Vizuelno provjeriti da li za DFT (kao i za DTFT) vrijede svojstva realnog niza opisana u stavci I. teoretskog dijela vježbe. Kako moraju izgledati amplituda, faza, te realni i imaginarni dio DFT-a s obzirom na indeks 0?

Izvesti program uz $N=1024$. Sjetiti se da je DTFT prikazan također u $N=1024$ točke. Kako sada izgledaju prikazi ? Čemu bi općenito težio DFT kada bi se broj uzoraka povećavao do beskonačnosti ?

MATLAB PROGRAM ZA IZRAČUNAVANJE DFT-A KOSINUSNOG NIZA

3. Napisati MATLAB program (skriptu) kojim se izračunava DFT beskonačno periodičnog, vremenski diskretnog kosinusnog niza $\tilde{x}[n] = \cos(\omega_0 n) = \cos(2\pi m/P \cdot n)$, u N točaka gdje varijabla m može poprimiti vrijednosti $0 \leq m \leq P/2-1$, a P je osnovni period niza. **Teoretska podloga dana je u stavci V. teoretskog dijela vježbe.** Raditi u skladu s uputama koje slijede.

a) Frekvencija kosinusnog niza ω_0 neka je definirana preko brojeva P i m , kako se vidi iz gornjeg izraza za $\tilde{x}[n]$. Ostvariti unos P i m korištenjem naredbe `input`.

b) Da bi se mogao raditi DFT, beskonačni niz treba ograničiti u vremenu na dužinu L . To se izvodi primjenom vremenskog otvora, a u ovom zadatku koristit će se najjednostavniji, pravokutni otvor. Broj L , te broj točaka N u koliko se izračunava DFT unijeti korištenjem naredbe `input`. Realizirati vektor n s indeksima vremenskog niza, $0 \leq n \leq L-1$ te vektor k s indeksima DFT-a, $0 \leq k \leq N-1$.

c) Izračunati skraćeni niz od $\tilde{x}[n]$ u L točaka, $x_L[n]$, uz indekse u vektoru n . S obzirom da se koristi pravokutni otvor dovoljno je samo izračunati niz za L indeksa. Da je korišten neki drugi tip otvora, trebalo bi pomnožiti uzorke signala i otvora.

d) Izračunati DFT niza $x_L[n]$ u N točaka uz indekse u vektoru k .

e) Izračunati amplitudu, fazu, te realni i imaginarni dio uzoraka DFT-a.

f) Na **prvoj slici** (kao i do sada prethodno pozvati naredbu `figure`) prikazati skraćeni kosinusni niz $x_L[n]$ pomoću naredbe `stem` uz odgovarajuće vremenske indekse.

g) Analogno prikazima u programu iz 1. zadatka, naredbom `subplot` prikazati na **drugoj slici amplitudu i fazu DFT-a**, a na **trećoj realni i imaginarni dio DFT**. Obzirom da

treba prikazati samo uzorke DFT-a, zamijeniti `plot` naredbu, naredbom `stem`. U naslovu slike neka su ispisane vrijednost korištenih N , L i m .

4. Izvesti program uz slijedeće parametre: $P=L=N=10$, $m=1$ i promotriti slike. Nakon toga izvesti program uz iste P , L i N , ali za m odabirati redom 2 pa 5. Svaki put se generiraju po tri nove slike.

Proučiti i usporediti slike koje prikazuju diskretni niz. Zatim proučiti slike koje prikazuju pripadne amplitude DFT uzoraka. Na koji način izgled amplitude DFT-a ovisi o broju m ? Koje je značenje broja m u smislu periodičnosti diskretnog niza na L uzoraka?

Zatvoriti sve postojeće slike naredbom `close all`. Izvesti program uz parametre: $P=L=N=10$, $m=1$. Pogledati diskretni niz. U glavi ga periodizirajte. Kakav je izgled tog periodiziranog niza s obzirom na simetričnost oko ishodišta? Pogledati amplitude i faze te realni i imaginarni dio DFT uzoraka s obzirom na simetričnost (misli se na simetričnost oko ishodišta – vidjeti svojstvo DFT-a pod II). Obratiti pažnju na veličine na y-osi. Jako male vrijednosti ($<10^{-13}$) ukazuju na to da se u stvari radi o vrijednostima koje bi trebale biti jednake nula, ali to nisu zbog numeričkih pogrešaka prilikom računanja. Ako su dakle bilo realni bilo imaginarni ili čak oba dijela DFT-a veoma mali brojevi, fazna vrijednost koja se izračunava na temelju njih nema smisla te se u prikazima ponaša nepredvidivo.

Izvesti sada program još uz $N=20$ i $N=40$, a ostale parametre zadržati istima i usporediti pripadne amplitude DFT uzoraka. Kako to da amplituda DFT-a uz veći N ima uzorke koje ne očekujemo s obzirom na postojanje samo jedne harmonijske komponente na frekvenciji ω_0 ? Što se događa povećavanjem broja uzoraka u kojima se radi DFT?

Zatvoriti sve postojeće slike. Izvesti program uz parametre: $P=L=N=10$, $m=1.4$, te promotriti dobivene slike. Kada m nije cijeli broj, a ostali parametri su jednaki, to znači da na tom broju uzoraka nema cijeli broj perioda kosinusa. Kako se to odražava na izgled amplitude i faze te realnog i imaginarnog dijela DFT uzoraka? Skicirati kako bi izgledao niz nastao periodiziranjem postojećih N uzoraka niza u vremenu. Prokomentirati.

FREKVENCIJSKA KARAKTERISTIKA REALNIH SIGNALA OTIPKANIH NA DSP MAKETI

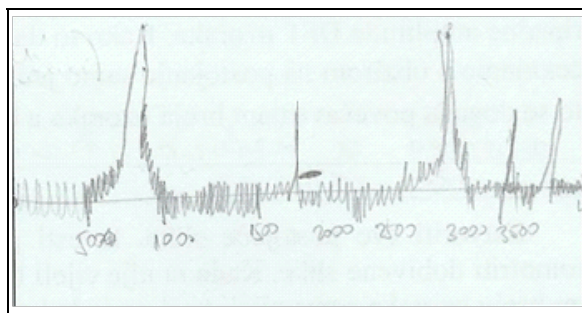
5. U komandnom prozoru MATLAB-a pokrenuti program **vj2** unosom naredbe **vj2 <ENTER>**. Prije toga je potrebno ući u direktorij **vj2**. Za usporedbu matematički konstruiranih signala sa signalima otipkanim s DSP makete sa padajućeg menija u gornjem desnom uglu odabrati opciju **Otipkavanje signala**. Odabrati frekvenciju signala **900 Hz** i **sinusni** valni oblik (ovo su parametri po kojima će MATLAB konstruirati željeni idealni signal). Korištenjem tipke **Prijenos programa na maketu** prenosi se program za otipkavanje signala na DSP maketu (prije prijena programa upaliti maketu i sačekati da prođe proces inicijalizacije). Na maketi podesiti **sinusni** valni oblik uz frekvenciju **900 Hz** (ovo je željeni realni signal koji dobivamo iz funkcijskog generatora). Na DSP maketi pritisnuti tipku 'interrupt' čime se zaustavlja izvršavanje programa uz spremanje otipkanih uzorka u memoriju DSP procesora. Tipkom **Prijenos podataka s makete** prenose se otipkani uzorci s

DSP makete u memoriju računala. Usporedba signala se provodi u vremenskoj i frekvencijskoj domeni.

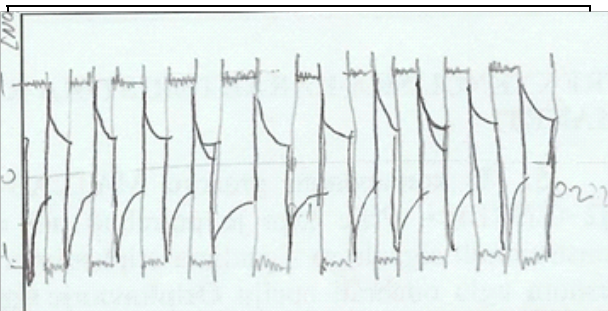
U vježbi se može odabrati da li se za ograničenje beskonačnog niza koristi pravokutni ili jedan nešto drugačiji otvor koji se naziva Blackmanov (vremenski otvori su detaljnije diskutirani u Vježbi 3.). Izgled ovog otvora u vremenskoj domeni je takav da u množenju sa signalom, sam signal nije skokovito prekinut već je postepeno prigušen prema rubovima promatranog intervala. Nadalje, djelovanje ovog otvora u frekvencijskoj domeni je takvo da su bočne latice otvora manjih amplituda pa se time može bolje istaknuti spektralni sastav signala, što se može i vidjeti. Prilikom izračunavanja spektra signala moguće je koristiti Blackman-ov vremenski otvor (aktiviranjem opcije **Korištenje vremenskog otvora**) radi poboljšanja karakteristike dobivenog spektra.

Za otipkani, odnosno konstruirani signal opisati razlike odnosno sličnosti u vremenskoj, odnosno frekvencijskoj domeni. Opisati utjecaj vremenskog otvora na spektar signala.

Ponoviti postupak za **pravokutni** signal frekvencije **900 Hz**. (odabrati željenu frekvenciju i valni oblik u aplikaciji, resetirati DSP maketu, prenijeti program na DSP maketu, odabrati željeni valni oblik i frekvenciju na DSP maketi, na maketi pritisnuti tipku 'interrupt', prenijeti podatke s DSP makete u računalo). Precrtati spektar otipkanog pravokutnog signala iz funkcijskog generatora i na sliku upisati porijeklo i teoretski očekivani iznos svake pojedine komponente koja se na slici ističe. Objasniti pojavu komponenti koje nisu bile teoretski očekivane.



Ponoviti postupak za **pravokutni** signal frekvencije **50 Hz**. Precrtati jednu do dvije periode otipkanog i konstruiranog signala u vremenu. Do promjene valnog oblika pravokutnog (a i trokutnog signala) dolazi zbog postojanja izmjenične, AC veze (serijski kondenzator) na ulazu u maketu. Frekvencijska karakteristika takvog ulaznog kruga je oblika visokog propusta, ali ono što posebno treba naglasiti jest da je **fazna karakteristika izrazito nelinearna** do cca. 500 Hz. Zbog nelinearnosti faze za prvih nekoliko harmonika u pravokutnom signalu dolazi do faznih izobličenja koja uzrokuju dobiveni valni oblik. Uz istu frekvenciju ponoviti postupak za **trokutni** i **sinusni** signal na ulazu. Zašto promjena valnog oblika nije vidljiva kod sinusnog signala?



Zato što sinusni signal sadrži samo jednu frekvenciju

Zašto je na višim frekvencijama ulaznog signala gore promatrani efekt manje izražen?
Zato što je fazna karakteristika ulaza na višim frekvencijama linearnija

Za one koje žele znati i više

MATLAB APLIKACIJA - DFS POMAKNUTOG NIZA

6. Ova vježba kao i sve koje slijede se također rade u programu **vj2**. Svojstvo DFS-a koje će se ovdje promatrati je utjecaj pomaka niza u vremenu na koeficijente DFS-a (stavka III. teoretskog dijela vježbe). Na padajućem izborniku u gornjem desnom kutu odabrati opciju **Pomak niza** čime će program generirati pravokutni periodični vremenski niz na osnovu zadanog broja uzoraka po periodi i širine bloka jedinica. Pored toga generira se isti takav signal, ali pomaknut u vremenu za zadani broj koraka. Parametri koji definiraju oba signala su broj uzoraka po periodi signala (N), širina bloka jedinica (M) te pomak u broju uzoraka između signala (m). Za početak odabrati kombinaciju $N=16$, $M=2$, $m=1$.

Proučiti pripadne DFS koeficijente (realni dio, imaginarni dio, modul, faza). Opisati uočena svojstva. Napisati izraz za fazni pomak te uvrstiti gore navedene koeficijente. Da li rezultati proračuna odgovaraju rezultatima dobivenim u MATLABU ?

Ponoviti postupak za različite vrijednosti parametara N , M i m te diskutirati rezultate.

MATLAB APLIKACIJA - DFT VREMENSKI SKALIRANOG NIZA

7. Za promatranje utjecaja vremenskog skaliranja niza na DFS koeficijente (stavka IV. teoretskog dijela vježbe) odabrati opciju **Skaliranje niza**. Ovom opcijom generiramo periodični niz čiji je jedan period duljine $N=64$ oblika Gauss-ovog zvona. Promjenom parametra σ (sigma) mijenja se širina zvona što odgovara skaliranju signala u vremenu. Parametar je moguće unijeti numerički (u intervalu $[.01;10]$ ili preko pomične trake (engl. slide bar). Za početak odabrati vrijednost $\sigma=1$. Program prikazuje, za odabrani parametar, signal u vremenskoj domeni te pripadne module DFS koeficijenata. Opisati prikazane slike.

Ponoviti postupak za $\sigma=.01$ i $\sigma=10$. Kakav je sada odnos između oblika vremenskog signala i pripadnih modula DFS koeficijenata? Eksperimentirati i s drugim vrijednostima parametra σ .

MATLAB APLIKACIJA - UTJECAJ SIMETRIČNOSTI VREMENSKOG NIZA NA KOEFICIJENTE DFS-A

8. Ovdje će se proučavati svojstva DFS-a vezana uz simetričnost odnosno antisimetričnost realnih nizova (stavka II. teoretskog dijela vježbe). Unutar grafičkog sustava odabrati opciju **Simetrični/antisim. niz**. Spomenuta svojstva će se proučavati na periodičnom nizu s dva bloka jedinica za simetrični slučaj, odnosno jedan blok jedinica i jedan blok minus jedinica za antisimetrični slučaj. Ostali uzorci vremenskog niza (signala) su nula. Ukoliko se odabere **16** uzoraka po periodi signala, **2** uzorka u bloku jedinica i **simetričan signal**, program će izgenerirati simetričan vremenski niz duljine 16, koji ima dva uzorka lijevo i dva uzorka desno od nule jednaka jedan, a sve ostale nula. Za tako definirani signal na donje dvije slike prikazan je iznos pripadnih koeficijenata DFS rastava. Opisati opažena svojstva DFS razvoja ovakvog signala (svojstva realnog dijela, imaginarnog dijela, modula i faze).

Ponoviti postupak za antisimetrični signal. Varirati broj uzoraka po periodi signala te širinu bloka jedinica.

3. VJEŽBA

Projektiranje FIR filtara metodom vremenskih otvora

FIR filtri su filtri s konačnim impulsnim odzivom (*engl. Finite Impulse Response*), a jedan od načina njihovog projektiranja je metoda s vremenskim otvorima. Kod projektiranja filtra ovom metodom, potrebno je najprije definirati oblik željene **idealne** frekvencijske karakteristike filtra $H_{id}(e^{j\omega})$. Inverznom Fourier-ovom transformacijom se zatim izračunavaju uzorci impulsnog odziva koji je za općenit slučaj beskonačnog trajanja, prema izrazu:

$$h_{id}[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_{id}(e^{j\omega}) e^{jn\omega} d\omega, \quad -\infty < n < \infty. \quad (3.1)$$

Radi jednostavnosti, često se odabire karakteristika $H_{id}(e^{j\omega})$ nulte faze, tj. kreće se od realne frekvencijske karakteristike koja je po odsječcima konstantna (iznosa 0 ili 1). Takva karakteristika se uobičajeno u engleskoj literaturi naziva *brick-wall* karakteristika, jer ima beskonačno uska prijelazna područja, što kao posljedicu ima beskonačni impulsni odziv. S obzirom da je frekvencijska karakteristika čisto realna funkcija, rezultirajući impulsni odziv je uvijek simetričan oko ishodišta $n=0$, tj. $h_{id}[n] = h_{id}[-n]$ (širi se jednako na obje strane).

Iako je izvedba takvih idealnih filtara nemoguća, metodom vremenskih otvora se projektiraju filtri koji aproksimiraju idealnu karakteristiku $H_{id}(e^{j\omega})$ i imaju konačan impulsni odziv. Postupak skraćivanja impulsnog odziva provodi se prema izrazu:

$$h[n] = \begin{cases} h_{id}[n] \cdot w[n], & \text{za } n \in S \\ 0, & \text{inace} \end{cases}, \quad (3.2)$$

odnosno množenjem idealnog impulsnog odziva $h_{id}[n]$ s uzorcima $w[n]$ koji predstavljaju neki od poznatih vremenskih otvora, također simetričnih oko ishodišta. U gornjem izrazu je sa S definiran skup indeksa za koje su $w[n]$, a time i $h[n]$ različiti od 0. Taj skup indeksa ovisi o tome da li je red filtra N paran ili neparan broj te je za oba slučaja definiran u drugom retku tablice 3.1. Red rezultirajućeg filtra bit će jednak N , ako vremenski otvor ima $N+1$ uzorak različit od nule.

Opisanim postupkom dobiva se impulsni odziv nekauzalnog sustava. Da bi se takav sustav mogao realizirati, potrebno ga je pretvoriti u kauzalni što se postiže jednostavnim vremenskim pomakom niza $h[n]$ za $N/2$ uzoraka u desno. Kao posljedica toga, fazna karakteristika više nije jednaka nuli, već postaje linearna s nagibom $N/2$. Ako je impulsni odziv kauzalnog FIR filtra dan s:

$$h_c[n] = \sum_{l=0}^N b_n \delta[n-l], \quad (3.3)$$

tada se koeficijenti b_n nalaze iz skraćenog impulsnog odziva $h[n]$ prema izrazima danim u trećem retku tablice 3.1, ovisno o parnosti reda filtra N .

Već je rečeno da idealni filtar s kojim se ulazi u postupak projektiranja ima pojačanje u području propuštanja jednako 1 te beskonačno gušenje (vrijednost 0) u području gušenja. Različitom rasporedom tih područja moguće je ostvariti filtre slijedećih tipova: niski propust

(LP), pojasni propust (BP), visoki propust (HP) i pojasna brana (BS). Za LP i HP filtre definira se **granična frekvencija** ω_c na rubu između područja propuštanja i područja gušenja $\omega_c \in [0, \pi]$, a za BP i BS definiraju se dvije frekvencije ω_l i ω_h , za koje vrijedi $\omega_l < \omega_h$. Te frekvencije određuju područje propuštanja (za BP), odnosno područje gušenja (za BS). Ako je red FIR filtra **N paran broj**, potrebno je odrediti $N + 1$ uzorak impulsnog odziva idealnog filtra simetrično oko nule za indekse $n \in \mathbf{S}$, gdje je skup \mathbf{S} definiran u drugom retku tablice 3.1, lijevi stupac. Izrazi za izračunavanje uzoraka idealnih impulsnih odziva za sva četiri tipa filtara su također dani u lijevom stupcu tablice.

Tablica 3.1 Tipovi idealnih filtara ovisni o parnosti reda filtra N

N paran	N neparan
$\mathbf{S} = \left\{ -\frac{N}{2}, \dots, \frac{N}{2} \right\}, n \in \mathbf{S}$	$\mathbf{S} = \left\{ -\frac{N+1}{2}, \dots, \frac{N-1}{2} \right\}, n \in \mathbf{S}$
$b_i = h \left[i - \frac{N}{2} \right], \text{ za } i = 0, \dots, N$	$b_i = h \left[i - \frac{N+1}{2} \right], \text{ za } i = 0, \dots, N$
$h_{lp}[n] = \begin{cases} \frac{\sin(n\omega_c)}{n\pi}, & n \neq 0 \\ \frac{\omega_c}{\pi}, & \text{ za } n=0 \end{cases}$	$h_{lp}[n] = \frac{\sin\left(\left(n + \frac{1}{2}\right) \cdot \omega_c\right)}{\left(n + \frac{1}{2}\right) \cdot \pi}$
$h_{bp}[n] = \begin{cases} \frac{2}{\pi n} \left(\cos \frac{n(\omega_h + \omega_l)}{2} \sin \frac{n(\omega_h - \omega_l)}{2} \right), & n \neq 0 \\ \frac{\omega_h - \omega_l}{\pi}, & \text{ za } n=0 \end{cases}$	$h_{bp}[n] = \frac{2}{\pi \left(n + \frac{1}{2}\right)} \cdot \left(\cos \frac{\left(n + \frac{1}{2}\right) \cdot (\omega_h + \omega_l)}{2} \sin \frac{\left(n + \frac{1}{2}\right) \cdot (\omega_h - \omega_l)}{2} \right)$
$h_{hp}[n] = \begin{cases} -\frac{\sin(n\omega_c)}{n\pi}, & n \neq 0 \\ 1 - \frac{\omega_c}{\pi}, & \text{ za } n=0 \end{cases}$	ne postoji
$h_{bs}[n] = \begin{cases} -\frac{2}{\pi n} \left(\cos \frac{n(\omega_h + \omega_l)}{2} \sin \frac{n(\omega_h - \omega_l)}{2} \right), & n \neq 0 \\ 1 - \frac{\omega_h - \omega_l}{\pi}, & \text{ za } n=0 \end{cases}$	ne postoji

Ako je N **neparan broj**, tada je potrebno odrediti $N + 1$ uzorak impulsnog odziva idealnog filtra simetrično oko nule, što se uz skup indeksa $n \in \mathbf{S}$ (pripadni \mathbf{S} definiran u drugom retku desnog stupca tablice 3.1) realizira pomakom za $1/2$ uzorka. Impulsni odzivi idealnih filtara računaju se za $n + 0.5 = -N/2, \dots, -0.5, 0.5, \dots, N/2$. Izrazi za LP i BP filtre dani su u desnom stupcu tablice. Međutim, BS i HP filtre nije moguće realizirati neparnim redom filtra. Treba se naime sjetiti da su impulsni odzivi FIR filtara dobiveni na ovaj način ili simetrični ili asimetrični, pa se $H(e^{j\omega})$ može prikazati kao suma kosinusa ili sinusa množenih koeficijentima koji ovise o $h[n]$. Za neparni N , tj. parni broj uzoraka impulsnog odziva (FIR TIP 2) treba sumirati članove oblika $b[m] \cdot \cos(\omega \cdot (m - 0.5))$. HP i BS filtri bi na π trebali imati

1, a kad se uvrsti $\omega = \pi$ ispada da je $b[m] \cdot \cos(\pm\pi/2) = 0$! Za TIP 3 to vrijedi za neparni broj uzoraka jer u pripadnom izrazu treba izračunati sinus od π što daje 0, a ne 1.

Ako se u postupku projektiranja FIR filtra koristi pravokutni vremenski otvor, čiji su svi koeficijenti $w[n]$ jednaki jedan, tada se to svodi na jednostavno odrezivanje beskonačnog idealnog impulsnog odziva na dužinu $N+1$. Zanimarivanjem preostalog dijela impulsnog odziva, unosi se greška koja mijenja željenu idealnu amplitudno-frekvencijsku karakteristiku. Greška može biti manja (pogotovo u području gušenja) ako se umjesto pravokutnog vremenskog otvora koriste otvori zvonolikog oblika koji prema rubu intervala postepeno guše uzorke impulsnog odziva. Neki od njih su: Bartlett-ov, Hamming-ov, Hann-ov i Blackman-ov. Broj uzoraka vremenskog otvora L mora biti za jedan veći od reda filtra N , $L = N+1$. Za parni, odnosno neparni L , vremenski otvori su mogu izračunati za indekse $n \in S$, prema izrazima danim u tablici 3.2.

Tablica 3.2 Tipovi vremenskih otvora ovisni o parnosti širine otvora L

L neparan	L paran
$S = \left\{ -\frac{L-1}{2}, \dots, \frac{L-1}{2} \right\}, n \in S$	$S = \left\{ -\frac{L}{2}, \dots, \frac{L}{2}-1 \right\}, n \in S$
$w_{bart}[n] = 1 - \frac{ n }{\frac{L-1}{2}}$	$w_{bart}[n] = 1 - \frac{\left n + \frac{1}{2} \right }{\frac{L}{2}}$
$w_{ha}[n] = \alpha + (1-\alpha) \cos\left(\frac{2\pi n}{L-1}\right),$ $\alpha = 0.54 \rightarrow \text{Hamming}$ $\alpha = 0.5 \rightarrow \text{Hann}$	$w_{ha}[n] = \alpha + (1-\alpha) \cos\left(\frac{2\pi \left(n + \frac{1}{2}\right)}{L-1}\right)$
$w_{black}[n] = 0.42 + 0.5 \cos\left(\frac{2\pi n}{L-1}\right) + 0.08 \cos\left(\frac{4\pi n}{L-1}\right)$	$w_{black}[n] = 0.42 + 0.5 \cos\left(\frac{2\pi \left(n + \frac{1}{2}\right)}{L-1}\right) + 0.08 \cos\left(\frac{4\pi \left(n + \frac{1}{2}\right)}{L-1}\right)$

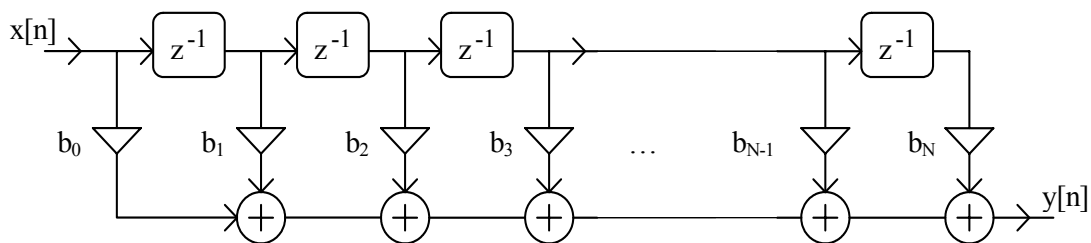
Neka karakteristična svojstva prethodno navedenih vremenskih otvora zbirno su prikazana u tablici 3.3 na slijedećoj strani. Vremenski otvori u frekvencijskoj domeni imaju valovit oblik s glavnom laticom na frekvenciji $\omega = 0$, **širine ovisne o tipu otvora i broju uzoraka**, kako je prikazano u trećem stupcu tablice 3.3. Vrhovi bočnih latica padaju s porastom frekvencije, a relativni odnosi amplituda glavne i najviše bočne laticе dani su za navedene otvore u drugom stupcu navedene tablice (**ne ovise o N !**).

Ako navedene vremenske otvore koristimo za projektiranje FIR filtara, dobivena amplitudno-frekvencijska karakteristika filtra ovisit će i o svojstvima primijenjenog otvora u frekvencijskoj domeni, jer je proizišla kao rezultat konvolucije spektra idealnog filtra i spektra otvora. **Tako će širina prijelaznog područja filtra zavisiti o širini glavne laticе korištenog otvora, a najveće pojačanje filtra u području gušenja** (zadnji stupac tablice 3.3), **zavisit će o odnosima amplituda glavne i najviše bočne laticе vremenskog otvora.**

Tablica 3.3 Karakteristike vremenskih otvora

Tip prozora	Valovitost bočne laticice (dB)	Širina glavne laticice	Max pojačanje filtra u području gušenja(dB)
Pravokutni	-13	$4\pi/(N+1)$	-21
Bartlett-ov	-25	$8\pi/N$	-25
Hann-ov	-31	$8\pi/N$	-44
Hamming-ov	-41	$8\pi/N$	-53
Blackman-ov	-57	$12\pi/N$	-74

FIR filtri s koeficijentima izračunatim na prethodno objašnjen način mogu se realizirati na više načina. Najjednostavniji način, koji direktno proizlazi iz jednadžbe diferencija kojom se može opisati ovaj sustav, prikazan je za red filtra N na slici 3.1. Realizacija je ostvarena elementima za kašnjenje, zbrajalima i množilima s konstantom, pri čemu konstante predstavljaju upravo izračunate koeficijente filtra.



Slika 3.1 Blok shema direktne realizacije FIR filtra

IMPLEMENTACIJA FIR FILTARA

Za implementaciju filtara općenito se najčešće koriste **procesori za digitalnu obradu signala** (engl. *Digital Signal Processor, DSP*). Kod implementacije je potrebno voditi računa o više aspekata koji će biti opisani i diskutirani u nastavku.

Kvantizacija

Rad u digitalnoj domeni podrazumijeva ne samo da su svi signali diskretizirani u vremenu već i da su diskretizirani po vrijednostima (amplitudi). To također znači da su i vrijednosti svih koeficijenata kod množenja kao i svi rezultati množenja i zbrajanja diskretne vrijednosti. Te vrijednosti nisu bilo koje već moraju biti jednake jednom od mogućih K diskretnih nivoa. Postupak kojim se to ostvaruje naziva se **kvantizacija**. Broj K je najčešće broj koji je potencija broja 2, pa vrijedi $K = 2^B$, pri čemu je B broj bita registara procesora ili drugog digitalnog sustava u kojima se pohranjuju uzorci signala i koeficijenti.

Pretvorba bilo koje realne vrijednosti u jedan od K diskretnih nivoa radi se tako da se dotični realni broj pomnoži s brojem K i zatim od rezultata uzme samo cijeli dio. Kod toga, međutim, decimalni dio rezultata utječe na to da li će cijeli broj biti prvi manji ili prvi veći kvantizacijski nivo u odnosu na rezultat, pa postoji nekoliko mogućnosti kako slijedi:

a) odsijecanje prema dolje

Za pozitivne vrijednosti, decimalni dio rezultata se ignorira i kao vrijednost uzima samo preostali cijeli dio. Negativne realne vrijednosti se zamjenjuju s prvim negativnijim cijelim brojem. Pošto je kvantizirana vrijednost uvijek ili jednaka ili manja od realne, greška uslijed kvantizacije je između -1 i 0.

b) odsijecanje prema nuli

Za pozitivne vrijednosti, ponašanje je isto kao pod a), ali za negativne vrijednosti se uzima prvi pozitivniji cijeli broj. Za pozitivne uzorke je kvantizirani broj uvijek manji od realnog, dok je za negativne uvijek veći (manje negativan). Zbog toga se greška kvantizacije nalazi u intervalu od -1 do 1.

c) zaokružnje na najbliži cijeli broj

Za pozitivne brojeve čiji je decimalni dio manji od 0.5, odabire se prvi manji cijeli broj, dok se za one čiji je decimalni dio veći ili jednak 0.5 odabire prvi veći cijeli broj. Za negativne brojeve je obrnuto. Ako je decimalni dio manji ili jednak 0.5, odabire se prvi pozitivniji cijeli broj, a za one druge odabiremo prvi negativniji. Ako se primjenjuje ova metoda, greška uslijed kvantizacije je uvijek između -0.5 i 0.5.

Tipični primjeri za sve tri metode dani su u tablici 3.4 koja slijedi.

Tablica 3.4 Primjeri kvantizacije signala

Realni broj	a)	b)	c)
1.9	1	1	2
1.5	1	1	2
1.1	1	1	1
0.9	0	0	1
0.5	0	0	1
0.1	0	0	0
-0.1	-1	0	0
-0.5	-1	0	0
-0.9	-1	0	-1
-1.1	-2	-1	-1
-1.5	-2	-1	-1
-1.9	-2	-1	-2

Prikaz brojeva s predznakom pomoću dvojnog komplementa

Iako na tržištu postoje DSP procesori koji mogu obavljati matematičke operacije u aritmetici pomičnoga zareza (*engl. floating point*) sa 7 decimalnih znamenki mantise i ogromnom dinamikom, još uvijek se najčešće koriste procesori s cjelobrojnom aritmetikom (*engl. fixed point*) zato što su jeftiniji, brži i jednostavniji. Kod ovakvih procesora, brojevi se najčešće prikazuju u obliku **dvojnog komplementa** (*engl. two's complement*) koji omogućava rad s pozitivnim i negativnim cijelim brojevima. Raspon brojeva koji se mogu prikazati s B bita (1 bit za predznak i $B-1$ bit za vrijednost) dan je u prva dva retka tablice 3.5. Radi usporedbe, u istom tom B -bitnom registru može se običnim **binarnim zapisom** prikazati cijeli broj bez predznaka između 0 i 2^{B-1} .

Tablica 3.5 Vezano uz prikaz brojeva u dvojnomo komplementu uz B bita

najveći pozitivni broj	$2^{B-1} - 1$
najveći negativni broj	-2^{B-1}
najveći pozitivni produkt	$(-2^{B-1}) \cdot (-2^{B-1}) = 2^{2B-2}$
najveći negativni produkt	$(2^{B-1} - 1) \cdot (-2^{B-1}) > -2^{2B-2}$

Na blok shemi prikazanoj na slici 3.1, vidljivo je da se kod realizacije filtra koriste zbrajala. Kod toga je prikaz cijelih brojeva u dvojnomo komplementu pogodan jer se za operacije zbrajanja (oduzimanja) može koristiti obično **binarno zbrajalo** (engl. *adder*). Promatrajmo to na 8-bitnom primjeru i zbrojimo dva broja prikazana u dvojnomo komplementu, 7 i -13. Pripadne vrijednosti u registrima su 7 (07h) i 243 (F3h). Sa 'h' su označeni heksadecimalni, a sa 'b' će biti označeni binarni brojevi dok su brojevi u decimalnom prikazu bez ikakve oznake. Binarnim zbrajanjem sadržaja registara $07h + F3h = FAh$ dobiva se broj koji predstavlja vrijednost -6 koja je upravo jednaka sumi ta dva broja. Ova činjenica vrijedi općenito. Podsjetnik: koji negativni broj je prikazan u registru jednostavno je odrediti ako se broju invertiraju bitovi i doda 1 (u primjeru $FAh = 1111\ 1010b$, invertiramo $\rightarrow 0000\ 0101b$ i dodamo 1 $\rightarrow 0000\ 0110b$, dakle to je -6).

Za realizaciju algoritama digitalne obrade signala, pored zbrajala potrebno je i množilo (engl. *multiplier*). Većina DSP procesora ima ugrađene, sklopovski podržane naredbe za realizaciju funkcije množenja. Postoje dvije vrste množila i to: **bez predznaka** (engl. *unsigned*) i **s predznakom** (engl. *signed*). Kod množila bez predznaka, produkt dva cijela broja bez predznaka prikazana s B bita je ponovno cijeli broj, ali dvostruke širine, tj. prikazan sa $2B$ bita. To se lako vidi ako pomnožimo dva najveća broja koji se mogu prikazati s B bita, $(2^{B-1}) \cdot (2^{B-1}) < (2^{2B-1})$, npr. $255 \cdot 255 = 65025 < 65535$. Množila s predznakom tretiraju ulazne faktore kao brojeve u dvojnomo komplementu, pa produkt izračunavaju kao produkt apsolutnih vrijednosti faktora, a predznak podešavaju prema predznacima faktora (prema XOR funkciji najviših bitova faktora). Raspon rezultata množila s predznakom dan je u trećem i četvrtom retku tablice 3.5. Vidljivo je dakle da je **produkt** sada moguće prikazati kao broj u dvojnomo komplementu s **registrom širine $2B-1$, a ne $2B$** .

Prikaz brojeva u obliku frakcija

Pošto se kao rezultati proračuna nekog filtra pojavljuju koeficijenti množenja koji nisu cijeli brojevi već decimalni, postavlja se pitanje kako realizirati potrebne matematičke operacije na cjelobrojnim procesorima. Ako se svi koeficijenti i signal u svim točkama sustava tako skaliraju da im se vrijednosti nalaze između -1 i 1, tada se za prikaz takvih brojeva može koristiti **frakcionalna aritmetika**. Veza između cjelobrojne i frakcionalne aritmetike je jednostavna: samo treba cjelobrojnu B -bitnu vrijednost u dvojnomo komplementu podijeliti brojem 2^{B-1} , što je dano primjerom za $B=3$ u tablici 3.6.

Tablica 3.6 Usporedba cjelobrojne i frakcionalne aritmetike

cjelobrojna	-4	-3	-2	-1	0	1	2	3
frakcionalna	-1	-0.75	-0.5	-0.25	0	0.25	0.5	0.75

Ovaj prikaz je samo jedna naša interpretacija stvarnog broja u registru, a sve operacije nad tim brojevima se ostvaraju kako je već opisano za cijele brojeve. Da to vrijedi i za množenje, promotrimo na jednostavnom primjeru. Za gornji primjer, $B=3$, pomnožimo

brojeve -0.75 i 0.5 . Ovi brojevi imaju pripadne cjelobrojne reprezentante u dvojnomo komplementu jednake -3 i 2 , koji se množe klasičnim množilom s predznakom. Kao produkt dobiva se vrijednost -6 u registru širine $2B-1$. Frakcionalni pandan ovog broja dobivamo tako da broj podijelimo s 2^{2B-2} što upravo daje vrijednost $-6/16 = -0.375 = -0.75 * 0.5$.

Ako se nakon operacije množenja produkt širine $2B-1$ želi skratiti na istu širinu koju imaju i faktori (tj. B bita), tada je potrebno od produkta uzeti najviših B bita. Tih B bita predstavljaju novu B -bitnu frakciju dobivenu kvantizacijom dugačke frakcije. Kvantizacija se može izvesti na tri poznata načina: običnim odsijecanjem, odsijecanjem simetrično oko nule i zaokruživanjem. Za gore navedeni primjer, promatrajmo obično odsijecanje na B bita. Vrijednost u registru širine $2B-1$ je -6 ili binarno $11010b$. Uzmimo najviša tri bita $110b$ što odgovara cijelomo broju -2 , odnosno frakciji -0.5 . Pošto broj -0.375 ne postoji u 3 -bitnim frakcijama uzimamo prvu manju vrijednost koja postoji.

Klasična realizacija FIR filtra u frakcionalnoj aritmetici

Postupkom projektiranja FIR filtra dobivamo koeficijente b_0 do b_N . Ti koeficijenti su realni brojevi beskonačne točnosti. Da bi se filter mogao realizirati B -bitnom frakcionalnom aritmetikom, potrebno je ove koeficijente kvantizirati na B -bitne frakcije (prije kvantizacije se koeficijenti prema potrebi mogu skalirati). Ova kvantizacija može se obaviti na tri već navedena načina. Signal na ulazu u filter koji je predstavljen realnim brojem, također je potrebno kvantizirati na B -bitnu frakciju, a tu operaciju obavlja A/D pretvornik

Kao što se vidi na slici 3.1, uzorke sa ulaza treba proslijediti na elemente za kašnjenje z^{-1} . Elementi za kašnjenje se realiziraju kao memorijske lokacije širine B bita. Sadržaji tih memorijskih lokacija nazivaju se **stanja filtra**, a kod FIR filtera odgovaraju zakašnjelim uzorcima ulaza. Stanja se množe s pripadnim koeficijentima, a dobiveni umnošci međusobno zbrajaju. Da bi greška uslijed operacija nad brojevima fiksne širine bila što manja, produkti širine $2B-1$ se obično ne skraćuju na širinu B , već se zbrajanje obavlja nad produktima pune širine $2B-1$. Tada je i izlaz filtra frakcionalna vrijednost širine $2B-1$, koju prije D/A pretvorbe (iste rezolucije od B bita kao i A/D), treba kvantizirati na širinu B . Za konkretnu implementaciju na jednom DSP procesoru pogledati slijedeće poglavlje.

ZA ONE KOJI ŽELE ZNATI VIŠE

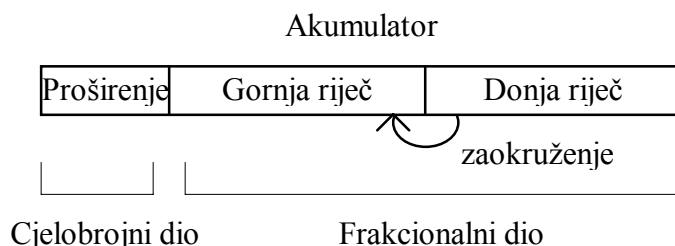
Realizacija FIR filtra na DSP procesoru

Implementacija FIR filtra provedena je na DSP procesoru AD2181 ugrađenom na DSP maketi. Ovaj DSP procesor je 16 -bitni procesor s cjelobrojnomo aritmetikom koji dakle radi sa 16 -bitnim internim registrima. Ulazni argumenti aritmetičko-logičkih operacija su 16 -bitni. Zbrajalo je 16 -bitno, a množilo 32 -bitno.

U sklopu množila postoji još jedan akumulator koji je 40 -bitni (32 bita + 8 bita proširenja). Akumulator se od običnog zbrajala razlikuje po tome što je jedan od ulaznih argumenata, kao i izlazni registar sa sumom jedan te isti fiksno određen registar. Struktura akumulatora prikazana je na slici 3.2. Pošto kod realizacije FIR filtra imamo više uzastopnih množenja uz akumulaciju rezultata (broj množenja odgovara redu filtra) postoji opasnost od izlaska iz dinamike (rezultat se više ne može prikazati u danom registru). Radi toga se akumulatori redovito izvode kao registri dvostruke širine uz nekoliko bita proširenja s lijeve strane, tj. u postupku računanja dozvoljava se privremena vrijednost u akumulatoru koja je veća od 1 .

Ukoliko se to proširenje sastoji od C bita, tada možemo provesti ukupno 2^C množenja uz akumuliraciju rezultata bez opasnosti od izlaska iz dinamike. Pošto je D/A pretvornik

također 16-bitni samo se 16-bitna gornje riječi uzima kao izlaz filtra. Da bi se smanjila pogreška uslijed ispuštanja donje riječi provodi se zaokružnje gornje riječi, tek nakon provedenih svih N množenja i pribrajanja. Pri tome treba provjeriti da li je rezultat unutar tražene dinamike, tj. da li je rezultat potpuno sadržan u gornjoj riječi. To znači da svi bitovi proširenja i najviši bit gornje riječi moraju biti jednaki (0 za pozitivni odnosno 1 za negativni broj). Ukoliko to nije zadovoljeno, potrebno je provesti zasićenje, tj. u akumulator upisati najveći pozitivni, odnosno najmanji negativni iznos. Time se sprečava da izlaz filtra poprimi pogrešan iznos (predznak) u slučaju prekoračenja dinamike. Jasno je da izlazni uzorak u tom slučaju nije dobro izračunat. Opisane metode osiguravaju malu pogrešku uslijed efekata konačne preciznosti.



Slika 3.2 Struktura akumulatora

Filtar je realiziran prema prikazanoj blok shemi na slici 3.1. Listing programa koja radi s 16 bitnim koeficijentima filtra i 16 bitnim ulaznim signalom dan je u nastavku.

```
{Dio 1}
{Definicija modula i konstanti}
MODULE/RAM      SFIR;           {Ime programa}
.CONST          N=45;           {Red filtra}

{Definicija ulaznih točaka u modul}
.ENTRY          ZAGLAVLJE;
.ENTRY          OBRADA;

{Definicija varijabli}
.VAR/PM/RAM/CIRC COEFS[N];      {Koeficijenti filtra}
.VAR/DM/RAM/CIRC DELAY[N];      {Stanja filtra}
.VAR/DM/RAM      sampl_rate;     {Frekvencija otipkavanja}
.GLOBAL          sampl_rate;

{Ucitavanje koeficijenata iz datoteke prilikom linkanja}
.INIT            COEFS : <SFIR.CFS>;

{Dio 2}
{Ovaj dio se izvodi samo jednom kod inicijalizacije filtra}
ZAGLAVLJE:
    I4=^COEFS;                  {Pokazivac na koeficijente}
    I2=^DELAY;                  {Pokazivac na stanja}
    L4=%COEFS;                  {Cirkularno adresiranje}
    L2=%DELAY;
    M2=1;                       {Offset registri}
    M4=1;
    CNTR=N;
    MX0=0;
    DO CLD UNTIL CE;
CLD:  DM(I2,M2)=MX0;             {Inicijalno brisanje stanja}
    RTS;

{Dio 3}
{Ovaj dio programa se izvodi za svaki otipkani uzorak}
OBRADA:                               {U MX1 nalazi se ulazni podatak}
    DM(I2,M2)=MX1;
```

```

CNTR=N-1; {Inicijalizacija akumulatora i prvih dvaju multiplikatora}
MR=0, MX0=DM(I2,M2), MY0=PM(I4,M4); {Konvolucija}
DO CONV UNTIL CE;
CONV: MR=MR+MX0*MY0(SS), MX0=DM(I2,M2), MY0=PM(I4,M4);
      MR=MR+MX0*MY0(RND); {Zaokruživanje rezultata}
      IF MV SAT MR; {Zasićenje u slučaju preljeva}
      MX1=MR1; {Izlazni podatak se vraća u MX1}
      RTS;
.ENDMOD;

```

Program se u grubo može podijeliti na tri dijela. U prvom dijelu se provodi definicija konstanti i rezervacija memorije za varijable. Ovaj dio se koristi od strane prevodioca prilikom prevođenja programa. U drugom dijelu, koji se poziva jednom na početku izvršavanja programa, provodi se inicijalizacija stanja filtra (brisanje sadržaja memorijskih lokacija u kojima će se čuvati stanja filtra) te inicijalizacija internih registara procesora (I, M i L) koji će se koristiti za indirektno adresiranje memorije. Treći dio programa se izvršava za svaki ulazni uzorak. Otipkani uzorak s A/D pretvornika ulazi u rutinu (na mjestu labela *obrada*) preko registra MX1, a na izlazu iz rutine registar MX1 sadržava izlazni uzorak iz filtra. Taj uzorak se pri idućem ulaznom uzorku šalje na D/A pretvornik.

U prikazanom programu treba uočiti slijedeću liniju:

```
MR=MR+MX0*MY0(SS), MX0=DM(I2,M2), MY0=PM(I4,M4);
```

Ova linija prikazuje pravu moć DSP procesora jer se istovremeno, u jednom ciklusu (u našem slučaju 30ns), provodi više nezavisnih operacija i to:

- Prva naredba: multiplikacija brojeva s predznacima (oznaka SS) jednog koeficijenta (MY0) i jednog stanja filtra (MX0) uz pribrajanje sumi prethodno dobivenih multiplikacija koja je već prisutna u registru MR
- Druga naredba: učitavanje u registar MX0 novog stanja sa memorijske lokacije u podatkovnoj memoriji (DM) koja je adresirana adresnim registrom I2
- Treća naredba: učitavanje u registar MY0 novog koeficijenta sa memorijske lokacije iz programske memorije (PM) koja je adresirana adresnim registrom I4
- Inkrementiranje (povećavanje za jedan) indeksnih registara (I2 i I4) koji služe za pristup koeficijentima i stanjima (u sklopu druge i treće naredbe)

Iterativnim ponavljanjem gornje linije (DO petlja) za sve koeficijente i stanja dobiva se iznos izlaznog uzorka. U gornjem programu nedostaje dio za komunikaciju sa A/D i D/A pretvornikom (preko jednog ulaznog i jednog izlaznog registra procesora) koji nije interesantan sa stanovišta funkcioniranja filtra.

Za slučaj s koeficijentima duljine 32 bita (stanja i ulazni podaci su i dalje 16-bitni) stvari su složenije jer je množilo na ovom DSP-u 16-bitno te se množenje mora provoditi u tri koraka na slijedeći način:

1. Pomnožiti stanje s donjom riječi koeficijenta (donjih 16 bita)
2. Pomaknuti dobiveni rezultat 16 bitova u desno (dijeljenje s 2^{16})
3. Pribrojiti dobivenom rezultatu umnožak stanja i gornje riječi koeficijenta (gornjih 16 bita)

Zadatak za pripremu

Za niskoopropusni FIR filter, granične frekvencije $\omega_c = 0.125 \cdot 2\pi$ i reda $N=5$ projektiran pomoću Hamming-ovog vremenskog otvora, izračunati $h_c[n]$ i $H_c(z)$ pripadnog kauzalnog filtra. Provesti kvantizaciju koeficijenata zaokruživanjem na najbliži cijeli broj sa 6 bita (5+1 predznak). Prikazati koeficijente u cjelobrojnom i frakcionalnom zapisu dvojnog komplementa. Nacrtati blok shemu filtra sa upisanim frakcijama. Izračunati frekvencijsku karakteristiku filtra $H_c(e^{j\omega})$ (ovo je FIR TIP2 – pogledati u materijalima sa predavanja), te skicirati amplitudno i fazno frekvencijsku karakteristiku filtra. Koristiti potrebne relacije iz teoretskog dijela vježbi.

Rješenje zadatka:

Praktična vježba

MATLAB PROGRAM ZA IZRAČUNAVANJE KOEFICIJENATA FIR FILTRA

1. Napisati Matlab program (skriptu) kojim se izračunavaju koeficijenti niskopropusnog FIR filtra granične frekvencije w_c korištenjem pravokutnog i Blackman-ovog vremenskog otvora te prikazuju amplitudno i fazno frekvencijske karakteristike otvora i dobivenog FIR filtra. Slijediti upute u nastavku.

a) Red filtra N unijeti korištenjem naredbe `input`. Napisati izraz za broj uzoraka vremenskog otvora L koji ujedno odgovara dužini konačnog impulsnog odziva FIR filtra.

b) Unos granične frekvencije w_c filtra unijeti korištenjem naredbe `input`.

c) Ispitati da li je N paran broj pomoću funkcije `rem` (ostatak kod dijeljenja), `rem(N,2)`, a zatim ovisno o tom uvjetu prema teoretskom dijelu pripreme definirati vektore indeksa n za nekauzalni impulsni odziv idealnog filtra (oko 0). Koristiti `if-then-else` strukturu.

d) U sklopu prethodno napisane `if-then-else` strukture ovisno o parnosti N napisati i izraze za uzorke impulsnog odziva idealnog niskopropusnog filtra hlp (izrazi u tablici 3.1). **ZA PARNI N PAZI !** Postoji uzorak na indeksu $n = 0$, tj. $hlp(N/2+1)$ za koji bi izraz za izračunavanje uzoraka impulsnog odziva rezultirao nulom u nazivniku. Da se pojednostavni izračunavanje uzoraka u ovom slučaju, izraz treba primijeniti za **sve indekse**, pa i za $n = 0$ usprkos tome što se Matlab žali na dijeljenje s nulom. U slijedećoj naredbi zatim treba vrijednost uzorka u $n = 0$ **posebno izračunati** prema danom izrazu.

e) Na **prvoj** slici (prije ove i svake nove slike reći prethodno `figure`) prikazati hlp sa `stem` uz odgovarajuće indekse vektora n .

f) Definirati vremenske otvore. Izraziti pravokutni otvor $w1$ dužine L korištenjem naredbe `ones(1,L)`. Uzorci Blackman-ovog otvora $w2$ ovise o parnosti N , a treba ih izračunati za prethodno definirani skup indeksa n . Stoga se u već postojećoj `if-then-else` strukturi može dodati i izračun uzoraka $w2$ prema izrazima iz tablice 3.2.

g) Prikazati vremenske otvore $w1$ i $w2$ na **drugoj** slici sa `stem` naredbom na malo drugačiji način i to: `stem(n,[w1' w2'],'filled');`. Objašnjenje: prethodno smo definirali retke $w1$ i $w2$. Korištenjem operatora `'` (transpozicija) ih pretvaramo u stupčane vektore. To je dakako samo privremeno (za prikaz) jer se kao transponirani vektori nigdje neće pohraniti. Ovo je potrebno jer se navedenim izrazom mogu prikazati uzorci 2 (i više) stupčana vektora na jednom grafu uz istu x-os. Isti oblik prikaza se može primijeniti i kod naredbe `plot`. Opcija `'filled'` označava da će kružići na vrhu uzoraka biti ispunjeni. (PAZI ! Izraz **ne radi u Matlabu 4**. Za taj Matlab primijeniti `stem` dva puta uz `hold on` naredbu između dva prikaza.)

h) Izračunati frekvencijske karakteristike vremenskih otvora $w1$ i $w2$. Za to se koristi funkcija `freqz` na način: `[w1,om]=freqz(w1,1,K);`. Funkcija traži slijedeće ulazne parametre. Prvi broj predstavlja koeficijente brojnika razlomljene racionalne funkcije. U našem slučaju su to koeficijenti impulsnog odziva vremenskog otvora. Drugi broj predstavlja koeficijente nazivnika kojih u slučaju FIR filtara (što i sami vremenski otvori na određeni način predstavljaju) nemamo pa se stavlja broj 1. Treći broj K definira broj frekvencijskih

točaka od 0 do π u koliko se evaluiira karakteristika koja je inače kontinuirana funkcija od frekvencije. Treba prije poziva funkcije definirati $K=1024$. Vrijednosti $W1$ predstavljaju kompleksne vrijednosti frekvencijske karakteristike za frekvencije sadržane u vektoru om .

i) Na **trećoj** slici prikazati amplitudne karakteristike vremenskih otvora. Koristiti `subplot` naredbu, pa na gornjem grafu prikazati amplitude dobivene funkcijom `abs()`, a u donjem grafu prikazati amplitude u dB pomoću `20*log10(abs())`. Kod toga u `plot` naredbi koristiti isti princip kao pod točkom g) s razlikom da su rezultati primjene funkcije `freqz` već stupčani vektori pa ih ne treba transponirati. Također je radi boljeg snalaženja zgodno frekvencijsku os prikazati normiranu s π (tada vrijednost frekvencije 1 odgovara π). Dakle, naredba za prikaz treba izgledati ovako: `plot(om/pi,[abs(W1) abs(W2)])`. Za prikaz amplitude u dB napraviti analogno.

j) Na temelju uzoraka vremenskih otvora $w1$ (pravokutnog) i $w2$ (Blackman-ovog) i uzoraka impulsnog odziva idealnog filtra *hlp* realizirati pripadne FIR filtre.

k) Na **četvrtoj** slici prikazati impulsne odzive dobivenih filtara $h1$ i $h2$ kako je to učinjeno pod točkom g).

l) Izračunati frekvencijske karakteristike filtara kako je to učinjeno pod točkom h)

m) Korištenjem funkcije `subplot` na **petoj** slici prikazati amplitudne karakteristike u dB (na gornjem grafu) i fazne karakteristike filtara (na donjem) kako je to učinjeno pod točkom i). Podsjetnik, fazna karakteristika izračunava se sa funkcijom `angle()`. U prikazu, vrijednost fazne karakteristike podijeliti s π radi boljeg očitavanja.

2. Izvesti program uz $N=30$ i graničnu frekvenciju $\omega_c=0.2\pi$ i promotriti prve tri slike. Na trećoj slici bi se trebali vidjeti amplitudni spektri vremenskih otvora radi usporedbe. Očito je međutim da spektre treba najprije nekako normirati da bi imali jednako pojačanje za DC komponentu prije nego se uspoređuju. To se može ostvariti tako da se sumiraju uzorci svakog pojedinog otvora i zatim sami uzorci otvora podijele sa dobivenim sumama. Dakle, treba nakon definiranja uzoraka otvora napisati slijedeće: `w11=w1/sum(w1);` i `w22=w2/sum(w2);`. Ako se sada u linijama koda u kojima se izračunavaju frekvencijske karakteristike otvora, zamijene $w1$ i $w2$ sa $w11$ i $w22$ i ponovno izvede program, moći će se lijepo vidjeti odnos amplitudnih spektara ova dva otvora. Ova akcija ima veze samo sa usporedbom spektara otvora. Filtre treba raditi sa uzorcima otvora kakvi jesu, dakle $w1$ i $w2$.

Zatvoriti dakle sve slike i ponovno izvesti program s istim parametrima, ali uz gornje izmjene. Proučiti sada sliku s prikazima spektara otvora. Povećati ju preko cijelog ekrana. Uočiti postojanje 'latica'. Uočiti da mjesta prolaska gornje krivulje (apsolutne vrijednosti frekv. karakt.) kroz nulu odgovaraju mjestima beskonačnog gušenja na donjoj krivulji izraženoj u dB. Kako je definirana glavna latica? Koji otvor rezultira širom glavnom laticom? Odrediti okvirno širine g. latica i usporediti ih sa onima u trećem stupcu tablice 3.3 uz $N=30$.

Na prikazu u dB ustanoviti koliko gušenje odgovara visini najviše bočne latica. Da bi se to lakše ustanovilo kliknuti na dotični graf na toj slici i u Matlabu otipkati `axis([0 1 -100 0])`. Tom naredbom ograničit će se raspon y-osi od 0 do -100 dB. Usporediti svoja zapažanja sa iznosima u drugom stupcu tablice 3.3. Koji otvor rezultira većim gušenjem bočnih latica ?

Izvesti sada program uz $N=55$ i $wc=0.2\pi$ i usporediti slike spektara otvora dobivene uz obje vrijednosti N . Kliknuti u graf i ponoviti `axis` naredbu. Provjeriti sada koliko je gušnje boč. latica, a kolike su širine gl. latica oba otvora. Što se od toga promijenilo s povećanjem N ?

povećanjem N latica se smanjuje, a gušenje ostaje isto

3. Sada promotriti slike impulsnih odziva konačnih FIR filtara dobivenih uz obje vrijednosti N , te njihove amplitudne i fazne spektre. Za grafove amplitudnih spektara primijeniti naredbu `axis([0 1 -100 5])`. Koji filter (dobiven od pravokutnog ili od Blackman-ovog otvora) rezultira većim gušenjem u području gušenja? Da li se gušenje i kako mijenja sa N ? Ustanoviti koliko je okvirno to gušenje i te brojke usporediti s brojkama u zadnjem stupcu tablice 3.3.

Promotriti prijelazna područja filtara. Da li ovise o tipu vrem. otvora? Da li ovise o N ?

MATLAB APLIKACIJA - UTJECAJ KVANTIZACIJE KOEFICIJENATA NA SVOJSTVA PROJEKTIRANIH FILTARA

4. U komandnom prozoru MATLAB-a pokrenuti program `vj3` unosom naredbe `vj3 <ENTER>`. Na najgornjem padajućem izborniku u gornjem desnom kutu grafičkog sučelja treba odabrati opciju **Projektiranje filtara**, a na izborniku **Odabir prikaza** odabrati **Kvantizacija koeficijenata** radi proučavanja efekata kvantizacije koeficijenata na svojstva filtra. Za odabrani broj bita na koji se kvantiziraju koeficijenti prikazuje se **amplitudno-frekvencijska** karakteristika filtra izračunata sa nekvantiziranim i kvantiziranim koeficijentima te pripadni odzivi na jediničnu stepenicu (uz odabranu opciju **Odziv na step**). Uz slijedeće parametre: **red filtra 20, Niski propust, graničnu frekvenciju 0.25 i pravokutni vremenski otvor**, odabrati **8 bita** za kvantizaciju i izredati sve tri metode kvantizacije. Ponoviti postupak za manji broj bita (npr. **4 bita**). Opisati kako broj bita za kvantizaciju koeficijenata utječe na izgled amplitudno-frekv. karakteristike? Koja metoda kvantizacije koeficijenata rezultira nešto lošijim karakteristikama filtra u odnosu na druge dvije? Za jedan odabir broja bita i metode kvantizacije pogledati kvantiziranu frekv. karakteristiku i odziv na jediničnu stepenicu. Zašto uz odsijecanje prema nuli odziv na step izgleda vrlo loše (pogledati izgled ampl.-frekv. karakteristike na niskim frekvencijama) ?

REALIZACIJA PROJEKTIRANIH FIR FILTARA NA DSP PROCESORU

5. U ovom zadatku bit će provedena realizacija projektiranih filtara i filtracija **na DSP procesoru**. Osim toga će se na temelju **istih kvantiziranih koeficijenata i kvantiziranih ulaznih uzoraka** prikupljenih s makete provesti simulacija filtracije **u Matlabu** pomoću funkcije **filter** te će biti napravljena usporedba rezultata dobivenih uz oba postupka.

U istoj aplikaciji `vj3` kao u prethodnom zadatku, s padajućeg menija **Odabir prikaza** odabrati opciju **DSP maketa**. Namjestiti željene parametre filtra na temelju kojih će Matlab program izračunati koeficijente: **Pojasni propust, red 40, granične frekvencije 0.2 i 0.3**

(normirano sa π), **Hamming-ov otvor**, te **frekvenciju otipkavanja 8 kHz**. Pritiskom na tipku **Start DSP (16-bit)** bit će generiran program za filtraciju za DSP te će dobiveni 16-bitni koeficijenti i sam program biti preneseni na maketu. Program na DSP maketi se automatski pokreće i provodi filtraciju koju možemo pratiti na osciloskopu. Najprije odrediti granične frekvencije pojasa propuštanja filtra izražene u Hz, pri čemu ovdje **normirana granična frekvencija 1 odgovara frekvenciji otipkavanja**. _____

Provjeriti rad filtra. Na kraju odabrati neku proizvoljnu ulaznu frekvenciju na maketi, pritisnuti tipku 'interrupt' te pritiskom na tipku **obrada (16 bita)** prenijeti snimljene podatke s makete na PC. Otvara se prozor u kojem su prikazana oba filtrirana signala (Matlab i DSP) te razlika među njima. Kolika je ta razlika?

Jedina razlika između dva postupka leži u činjenici da su **operacije** izvedene na DSP-u u **cjelobrojnoj aritmetici**, a u Matlabu u **aritmetici pomičnog zareza**. S obzirom da su i cjelobrojno množilo i akumulator na DSP-u u slučaju filtracije s 16-bitnim koeficijentima takve preciznosti da sadržavaju potpuni rezultat, nema apsolutno nikakve razlike u oba slučaja, tj. **greška je nula**. Svi međučvorovi su maksimalne točnosti kolika treba za prikaz rezultata što je u neku ruku i posljedica same FIR strukture jer nigdje osim eventualno na izlazu ne treba kvantizirati rezultate.

Za one koji žele i više

MATLAB APLIKACIJA - VREMENSKI OTVORI

6. U sklopu Matlab aplikacije odabrati opciju **Vremenski otvori** na padajućem izborniku u gornjem desnom kutu grafičkog sučelja. Za zadani broj točaka vremenskog otvora L , prikazuje se izgled otvora u vremenskoj domeni (gornja slika) i pripadni spektar (frekvencijska domena). Opisati dobivene karakteristike, te analizirati odnose širina glavnih latica i gušenje bočnih latica u ovisnosti o broju uzoraka vremenskog otvora, odnosno tipu otvora.

MATLAB APLIKACIJA - PROJEKTIRANJE FIR FILTARA

7. Vremenskim otvorima promatranim u prethodnoj točki će se u nastavku vježbe projektirati nerekurzivni (FIR) filtri. Za to odabrati opciju **Projektiranje filtara**. Za zadani red filtra N , korištenjem odabranog vremenskog otvora projektira se željeni filter (LP, HP, BP ili BS) zadanih graničnih frekvencija. Za početak odabrati red filtra **20**, **Niski propust**, graničnu frekvenciju **0.25** i **pravokutni** vremenski otvor. Postupak projektiranja je demonstriran kroz nekoliko slika. Najprije se prikazuje amplitudno-frekvencijska karakteristika idealnog filtra i pripadni (beskonačni) impulsni odziv (opcija **Idealan filter**). Množenjem impulsnog odziva s odabranim vremenskim otvorom (opcija **Impulsni odziv idealnog filtra**) provodi se skraćivanje impulsnog odziva na broj uzoraka koji odgovara redu filtra. Pripadna frekvencijska karakteristika (opcija **Frekvencijska karak. filtra**) uspoređena je s idealnom. Za različite tipove filtara, redova filtara i primijenjenih vremenskih otvora promotriti pojedine korake postupka projektiranja filtra metodom vremenskih otvora.

4. VJEŽBA

Projektiranje optimalnih FIR filtara

Kod projektiranja FIR filtara metodom fiksnih vremenskih otvora, valovitost u području gušenja zavisila je o tipu korištenog otvora. Primjenom adaptivnih vremenskih otvora (Kaiser, Dolph-Čebišev) moguće je utjecati na valovitost, ali je ona ista za cijelo frekvencijsko područje (za područje propuštanja i za područje gušenja). Metoda koja bi omogućila **nezavisnu** kontrolu nad valovitostima u pojedinim pojasevima, dala bi filtre nižeg reda za ostvarenje iste željene karakteristike.

Metode projektiranja kojima se to može ostvariti temelje se na **iterativnim optimizacijskim postupcima** koji se provode uz pomoć računala. Osnovna ideja kod takvih postupaka je određivanje amplitudno-frekvencijske karakteristike $A_e(e^{j\omega})$ koja aproksimira željenu frekvencijsku karakteristiku $D(e^{j\omega})$ prema nekom kriteriju uz primjenu težinske funkcije $W(\omega)$ koju definira korisnik i to na najbolji mogući način (dakle optimalno). Da bi se mogao provesti ovakav optimizacijski postupak, najprije treba definirati izraz za optimizacijsku pogrešku, kako slijedi:

$$E(\omega) = W(\omega)[D(e^{j\omega}) - A_e(e^{j\omega})] \quad (4.1)$$

Često korišteni optimizacijski kriterij je tzv. minimax ili Čebiševljev kriterij kod kojeg se unutar zadanih frekvencijskih intervala danih sa F traži takav odziv $A_e(e^{j\omega})$ (definiran preko uzoraka pripadnog impulsnog odziva $h_e[n]$) koji minimizira najveću (max) pogrešku aproksimacije, ili zapisano izrazom:

$$\min_{h_e[n]: 0 \leq n \leq M} \left(\max_{\omega \in F} |E(\omega)| \right). \quad (4.2)$$

F predstavlja uniju zatvorenih intervala frekvencija, a čine ga područja propuštanja i područja gušenja željenog $D(e^{j\omega})$. U optimizacijskom postupku se kreće od nekih početnih uzoraka $h_e[n]$, te se primjenom kriterija kroz više iteracija dolazi do optimalnog digitalnog filtra. Jednu takvu metodu projektiranja optimalnih FIR filtara predložili su 1972. Parks i McClellan i ona je našla najširu primjenu u praksi. Ova metoda opisana je u nastavku za primjer niskopropusnog filtra parnog reda, a moguće ju je primijeniti i za druge tipove filtara.

Promatrajmo niskopropusni FIR filter parnog reda N (neparni broj uzoraka impulsnog odziva), simetričnog impulsnog odziva s obzirom na nulu (nekauzalni filter). Za takav filter vrijedi: $h_e[n] = h_e[-n]$, te je njegova Fourier-ova transformacija realna i ima oblik:

$$A_e(e^{j\omega}) = h_e[0] + \sum_{n=1}^M 2h_e[n] \cos(\omega n), \quad (4.3)$$

gdje je $M = N/2$, cijeli broj. Kašnjenjem ovog nekauzalnog filtra za M uzoraka, dobiva se kauzalni filter (kojeg onda realiziramo) s impulsnim odzivom $h[n]$ i frekvencijskom karakteristikom $H(e^{j\omega})$ prema izrazima:

$$h[n] = h_e[n - M], n = 0, \dots, N \quad (4.4)$$

$$H(e^{j\omega}) = A_e(e^{j\omega}) \cdot e^{-j\omega M}. \quad (4.5)$$

Karakteristika željenog (idealnog) niskopropusnog filtra kojeg pokušavamo optimalno aproksimirati ovom metodom, definira se izrazom:

$$D(e^{j\omega}) = \begin{cases} 1, & 0 \leq \omega \leq \omega_p \\ 0, & \omega_s \leq \omega \leq \pi \end{cases}. \quad (4.6)$$

Karakteristika je definirana preko vrijednosti 1 i 0, te rubnih frekvencija područja propuštanja ω_p i područja gušenja ω_s . **Treba uočiti da prijelazno područje nije ni na koji način obuhvaćeno u ovoj definiciji.** Metoda optimizacije koju su opisali Parks i McClellan kreće od **zadanih frekvencija ω_p i ω_s , od reda filtra N , te omjera dozvoljenih odstupanja** aproksimacije od idealne karakteristike u području propuštanja $\pm \delta_1$ i području gušenja $\pm \delta_2$, tj. $K = \delta_1 / \delta_2$. Zadaje se samo omjer, dok iznosi odstupanja ostaju slobodne varijable optimizacijskog postupka. Pošto se odstupanja δ_1 i δ_2 mogu razlikovati, uvedena je težinska funkcija $W(\omega)$ koja omogućava različite težinske faktore pogreške aproksimacije za različita frekvencijska područja, a definirana je izrazom:

$$W(\omega) = \begin{cases} \frac{1}{K}, & 0 \leq \omega \leq \omega_p \\ 1, & \omega_s \leq \omega \leq \pi \end{cases}. \quad (4.7)$$

Promotrimo ponovno izraz (4.3) za realnu frekvencijsku karakteristiku FIR filtra tipa 1. Vidljivo je da se u izrazu javlja član oblika $\cos(\omega \cdot n)$ koji se može izraziti kao suma potencija od $\cos(\omega)$ korištenjem Čebiševljevih polinoma prema izrazu:

$$\cos(\omega n) = T_n(\cos \omega), \quad (4.8)$$

gdje je $T_n(\cos \omega)$ polinom n -tog stupnja u varijabli $\cos \omega$. Ovakvim supstitucijama se izraz (4.3) može napisati u obliku:

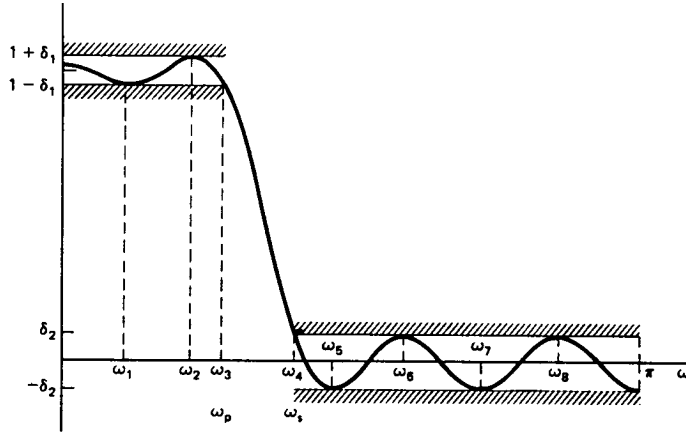
$$A_e(e^{j\omega}) = \sum_{n=0}^M a_n (\cos \omega)^n = P_M(\cos \omega), \quad (4.9)$$

gdje su koeficijenti a_n polinoma P_M u varijabli $\cos \omega$, mogu izračunati iz uzoraka impulsnog odziva $h_e[n]$. Dakle, na ovaj način je $A_e(e^{j\omega})$ izražen kao trigonometrijski polinom M -tog stupnja pa se problem aproksimacije niskopropusnog filtra prevodi na problem optimalne aproksimacije polinomom, na skupu disjunktних, zatvorenih intervala. Taj problem je u matematici već bio detaljno obrađen pa su se sve spoznaje o tome mogle direktno primijeniti kod proračuna FIR filtra.

Važnu ulogu u tom smislu ima jedan teorem koji se naziva **teorem alternacije** koji će biti sada objašnjen za primjeru primjene u projektiranju optimalnih FIR filtara. Dovoljan i nužan uvjet da karakteristika $A_e(e^{j\omega})$, odnosno njen pripadni polinom P_M , predstavljaju optimalnu aproksimaciju koja minimizira maksimalnu pogrešku $E(\omega)$ (koju ćemo označiti sa $|E|$), je taj da $E(\omega)$ napravi najmanje $(M+2)$ alternacije. Drugim riječima, tražimo da funkcija pogreške aproksimacije $E(\omega)$ naizmjenice poprima minimalnu, $-|E|$, pa zatim maksimalnu, $+|E|$, vrijednost u barem $(M+2)$ frekvencijske točke $\omega_i \in F$. Matematički izraženo, postoje barem $(M+2)$ vrijednosti $\omega_i \in F$, $\omega_1 < \omega_2 < \dots < \omega_{M+2}$, takvih da:

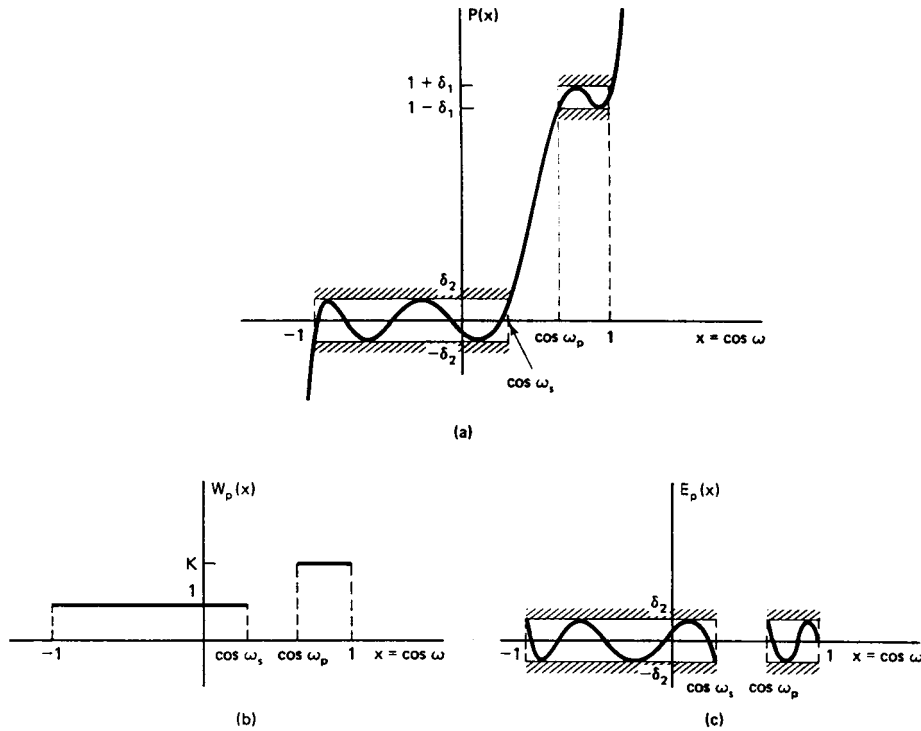
$$E(\omega_i) = -E(\omega_{i+1}) = \pm |E|, \quad i = 1, 2, \dots, (M+1). \quad (4.10)$$

Jedan primjer optimalne aproks. filtra reda $N = 14$ ($M = N/2 = 7$) dan je na slici (Slika 4.1.).



Slika 4.1. Optimalna aproksimacija filtra za red $N=14$.

Iz slike se vidi da $A_e(e^{j\omega})$ naizmjenično poprima gornje i donje granične vrijednosti devet puta na intervalu $[0, \pi]$ (točke na rubovima intervala se također računaju kao ekstremi) što je u skladu sa gore navedenim uvjetom optimalne aproksimacije. Pošto se čitav postupak provodi s transformiranom frekvencijskom osi, tj. umjesto ω uvodi se varijabla $x = \cos(\omega)$, funkcija aproksimacije $P(x)$, težinska funkcija $W_p(x)$ i funkcija pogreške $E_p(x)$ izgledaju kao na slikama (Slika 4.2. a), b), c)).



Slika 4.2. a), b), c)

Ako je zadovoljen uvjet alternacije funkcije $E(\omega)$, tada se za točke ω_i , u kojima $E(\omega)$ poprima vrijednost $\pm |E| = \pm \delta$, može napisati izraz dobiven iz (4.1):

$$W(\omega_i)[D(e^{j\omega_i}) - A_e(e^{j\omega_i})] = (-1)^{i+1} \cdot \delta, \quad i=1,2,\dots,(M+2). \quad (4.11)$$

Za definirani skup ω_i i pripadne $x_i = \cos(\omega_i)$, moguće je raspisivanjem $A_e(e^{j\omega_i})$ po potencijama od x_i prema izrazu (4.9), napisati skup linearnih jednadžbi s nepoznicama a_0, a_1, \dots, a_M i δ , što je dano s:

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^M & \frac{1}{W(\omega_1)} \\ 1 & x_2 & x_2^2 & \dots & x_2^M & \frac{-1}{W(\omega_2)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & x_{M+2} & x_{M+2}^2 & \dots & x_{M+2}^M & \frac{(-1)^{M+2}}{W(\omega_{M+2})} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ \delta \end{bmatrix} = \begin{bmatrix} D(e^{j\omega_1}) \\ D(e^{j\omega_2}) \\ \vdots \\ D(e^{j\omega_{M+2}}) \end{bmatrix}. \quad (4.12)$$

Rješavanjem ovog sustava linearnih jednadžbi moguće je za neki početni skup frekvencija ω_i odrediti nepoznane a_n i δ , te uvrštavanjem u izraz (4.9) odrediti aproksimativnu funkciju $A_e(e^{j\omega})$. Za tu funkciju potrebno je zatim odrediti novi skup točaka ω_i u kojima $E(\omega)$ postiže ekstremalne vrijednosti, te taj skup točaka proslijediti kao ulaz u novu iteraciju. Postupak konvergira i nakon određenog broja iteracija dobiva se optimalno rješenje. Međutim, za visoki red filtra, postupak temeljen na rješavanju velikog broja linearnih jednadžbi s puno nepoznanica je nepraktičan i spor, pa su Parks i McClellan primijenili metodu kod koje se koeficijenti a_n ne računaju u svakom koraku. Umjesto toga se na osnovu danog skupa ω_i računa samo novi δ prema slijedećem izrazu:

$$\delta = \frac{\sum_{n=1}^{M+2} b_n D(e^{j\omega_n})}{\sum_{n=1}^{M+2} b_n \frac{(-1)^{n+1}}{W(\omega_n)}}, \quad b_n = \prod_{\substack{i=1 \\ i \neq n}}^{M+2} \frac{1}{x_n - x_i}. \quad (4.13)$$

Ako je $A_e(e^{j\omega})$ definiran koeficijentima a_n koji zadovoljavaju (4.12) i ako je δ dan izrazom (4.13) tada funkcija greške $E(\omega)$ prolazi kroz $\pm \delta$ u $M+2$ frekvencije ω_i . Analogno tome, funkcija $A_e(e^{j\omega})$ prolazi kroz $1 \pm K \cdot \delta$ za $0 \leq \omega_i \leq \omega_p$, odnosno kroz $\pm \delta$ za $\omega_s \leq \omega_i \leq \pi$. Pošto znamo da je $A_e(e^{j\omega})$ trigonometrijski polinom M -tog reda, moguće je kroz $(M+1)$ -an od $(M+2)$ poznatih parova točaka provući Lagrange-ov interpolacijski polinom, te na osnovu njega odrediti novi skup točaka ω_i , u kojima $A_e(e^{j\omega})$ poprima ekstremalne vrijednosti. Ovako bez računanja koeficijenata a_n dolazimo do novog skupa točaka za slijedeću iteraciju. Optimizacijski postupak relativno brzo konvergira, jer se u svakom koraku mijenjaju sve vrijednosti ω_i . Na kraju se na osnovu konačnih ω_i i δ računaju koeficijenti a_n , odnosno koeficijenti impulsnog odziva $h_e[n]$.

FIR filtri dobiveni ovom metodom imaju najmanju maksimalnu pogrešku δ za definirano prijelazno područje $(\omega_s - \omega_p)$. Ako je težinska funkcija definirana kao u izrazu (4.7), tada je maksimalna pogreška aproksimacije u području gušenja jednaka $\delta_2 = \delta$, a u području propuštanja $\delta_1 = K\delta$. Ako za dani N , ω_p i ω_s nisu dobiveni zadovoljavajući δ_1 i

δ_2 , tada treba povećati ili N ili $(\omega_s - \omega_p)$ i ponoviti optimizacijski postupak. Kao što je već rečeno, postupak se može poopćiti i za neparni red filtra te za ostale tipove filtra (visoki propust, pojasni propust, pojasna brana, multiband).

Postoji aproksimativni izraz koji na osnovu zadanog prijelaznog područja i valovitosti u području propuštanja i području gušenja, procjenjuje potrebni red N optimalnog filtra za realizaciju željene karakteristike

$$N = \frac{-10 \log_{10}(\delta_1 \delta_2) - 13}{2.324 \cdot (w_s - w_p) \cdot 2 \cdot \pi}, \quad (4.14)$$

gdje su δ_1 i δ_2 valovitosti u području propuštanja i području gušenja, w_s i w_p granične frekvencije (iz intervala $[0;0.5]$), a N potrebni red filtra.

Zadatak za pripremu

Ukoliko se FIR filter projektira Parks-McClellanovom metodom. Odgovoriti na slijedeća pitanja: Koji se parametri filtra biraju (nezavisne varijable optimizacije), a što se optimizira. Kakva je uloga težinske funkcije?

Opisati te za red filtra $N=12$ skicirati amplitudno-frekvencijsku karakteristiku.

Rješenje zadatka:

Praktična vježba

MATLAB PROGRAM ZA IZRAČUNAVANJE KOEFICIJENATA OPTIMALNIH FIR FILTARA

1. Napisati Matlab program (skriptu) kojim se izračunavaju koeficijenti optimalnog FIR filtra metodom Parks i McClellana te prikazuju amplitudna i fazno-frekvencijska karakteristika dobivenog FIR filtra. Osnovna funkcija za projektiranje optimalnih FIR filtara spomenutom metodom je funkcija **remez**. Ovom skriptom ostvarit će se jedan konkretni optimalni niskopropusni filter. Slijediti upute u nastavku.

a) Unijeti red filtra $N = 12$;

b) Kao ulaz u funkciju, potrebno je definirati oblik željene, idealne karakteristike filtra. To se postiže definiranjem dva vektora, f i m . **Vektor f sadrži rubne frekvencije pojedinih pojasa, za svaki pojas po dvije točke.** Za ovaj filter, rubna kružna frekvencija pojasa propuštanja ω_p neka je jednaka $0.2 \cdot 2\pi$, a rubna kružna frekvencija pojasa gušenja ω_s neka je $0.3 \cdot 2\pi$. Definirati varijable $w_p = 0.2$ i $w_s = 0.3$, a pripadni vektor s frekvencijama f treba definirati ovako $f = [0 \ \omega_p \ \omega_s \ 0.5]$; . U tom izrazu prva dva broja definiraju područje propuštanja, a druga dva broja područja gušenja. Uočiti da su frekvencije zadane u rasponu $[0 \ 0.5]$, gdje frekvencija 0.5 odgovara kružnoj frekvenciji π , tj. polovici frekvencije otipkavanja.

c) **Vektor m** treba sadržavati **iznose modula** amplitudno-frekvencijske karakteristike unutar pojedinog pojasa filtra i to po dvije jednake vrijednosti za rubne točke pojasa. Za idealni filter bez pojačanja logično je u pojasu propuštanja uzeti vrijednost 1, a u pojasu gušenja 0. Dakle za vektor m treba uzeti $m = [1 \ 1 \ 0 \ 0]$

d) Ulazna vrijednost u funkciju **remez** je i **vektor s težinama w** . Ako želimo da oba pojasa imaju jednaku valovitost, tada težinsku funkciju treba definirati kao $w = [1 \ 1]$. Za odnos valovitosti $K = \delta_1 / \delta_2 = 10$, w se definira kao $w = [1/10 \ 1]$ i time dozvoljavamo 10 puta veću valovitost u području propuštanja nego u području gušenja, te obrnuto za $K = 0.1$ treba definirati $w = [1/0.1 \ 1]$. Dakle, što je veća težina, program se mora više truditi i konačni rezultat će biti manja valovitost u tom pojasu. U ovom programu odabrati $w = [4 \ 1]$.

e) Funkcija **remez** poziva se na slijedeći način: $b = \text{remez}(N, 2*f, m, w)$; . Kao rezultat vraća redak b , dužine $N+1$, s uzorcima impulsnog odziva optimalnog filtra ($b(1) = h_e[-M] = h[0]$, sve do $b(N+1) = h_e[M] = h[N]$). Uočiti da je funkcija pozvana s vrijednostima frekvencija pomnoženim s 2 ($2*f$) jer očekuje vrijednosti između 0 i 1.

f) Na **prvoj slici** (prije svake slike staviti **figure**) prikazati uzorke impulsnog odziva optimalnog filtra naredbom **stem** uz vremensku os definiranu vektorom $n = [0 : N]$.

g) Izračunati frekvencijsku karakteristiku filtra $H(e^{j\omega})$ pomoću **freqz** na način kako slijedi. Izračunati vektor frekvencija om u $brt = 1024$ točke, jednoliko raspoređenih u intervalu $0 \leq om < \pi$, u kojima će se računati frekvencijska karakteristika. Koristiti naredbe $om = [0 : (brt-1)] / brt * 0.5$; i $H = \text{freqz}(b, 1, 2*om*pi)$; . Uočiti da vektor om pokriva raspon frekvencija od 0 do 0.5, a u pozivu funkcije **freqz** množi se s 2π da bi se

dobila kružna frekvencija. U ovoj vježbi je to učinjeno na takav način da bude jednako sa gotovom Matlab aplikacijom vj4. Izračunati ampl.-frekv. karakteristiku, te istu tu karakteristiku, ali prikazanu u dB. Prikazati ove dvije karakteristike na **drugoj slici** pomoću subplot naredbe jednu ispod druge, uz frekvencije dane vektorom *om*.

h) Za detaljniju analizu karakteristike samo u **području propuštanja**, tj. od 0 do ω_p , treba malo drugačije definirati pripadnu frekvencijsku os. Dakle, raspon od 0 do 1 u brt točaka treba pomnožiti s ω_p , $om = [0 : (brt-1)] / brt * \omega_p$. Izračunati $H(e^{j\omega})$ za te frekvencije. Izračunati i pripadnu amplitudnu karakteristiku *AHp* te ju prikazati uz *om* na **trećoj slici** pomoću naredbe plot.

i) Izračunati minimalnu *mnAHp* i maksimalnu vrijednost *mxAHp* (vrhove i dna valova) amplitudne karakteristike u pojasu propuštanja. Koristiti naredbe min i max.

j) Za analizu karakteristike samo u **području gušenja**, pripadnu frekvencijsku os definirati sa $om = [0 : (brt-1)] / brt * (0.5 - ws) + ws$. Izračunati $H(e^{j\omega})$ za te frekvencije, te pripadnu amplitudnu karakteristiku *AHg* prikazati na **četvrtoj slici**.

k) Izračunati maksimalnu *mxAHg* vrijednost amplitudne karakt. u pojasu gušenja.

2. Izvesti program, pogledati dobivene slike i odgovoriti na slijedeća pitanja. Na temelju minimalne i maksimalne vrijednosti amplitudne karakt. u pojasu propuštanja odrediti valovitost u području propuštanja (željena vrijednost za to područje bila je 1):

$\delta_1 =$ _____ Koliko puta *AHp* poprima vrijednost $1 \pm \delta_1$: _____.

Na temelju *mxAHg* odrediti valovitost u području gušenja $\delta_2 =$ _____ Koliko puta *AHg* poprima vrijednost δ_2 : _____.

Kakav je odnos δ_1 i δ_2 ? _____
Da li ukupni broj ekstrema u području gušenja i području propuštanja odgovara uz red *N*?

3. U programu za granicu pojasa gušenja staviti $ws = 0.4$, čime je prošireno prijelazno područje filtra. Izvesti program i pogledati dobivene iznose valovitosti. Koja je promjena u odnosu na prethodni zadatak i zašto?

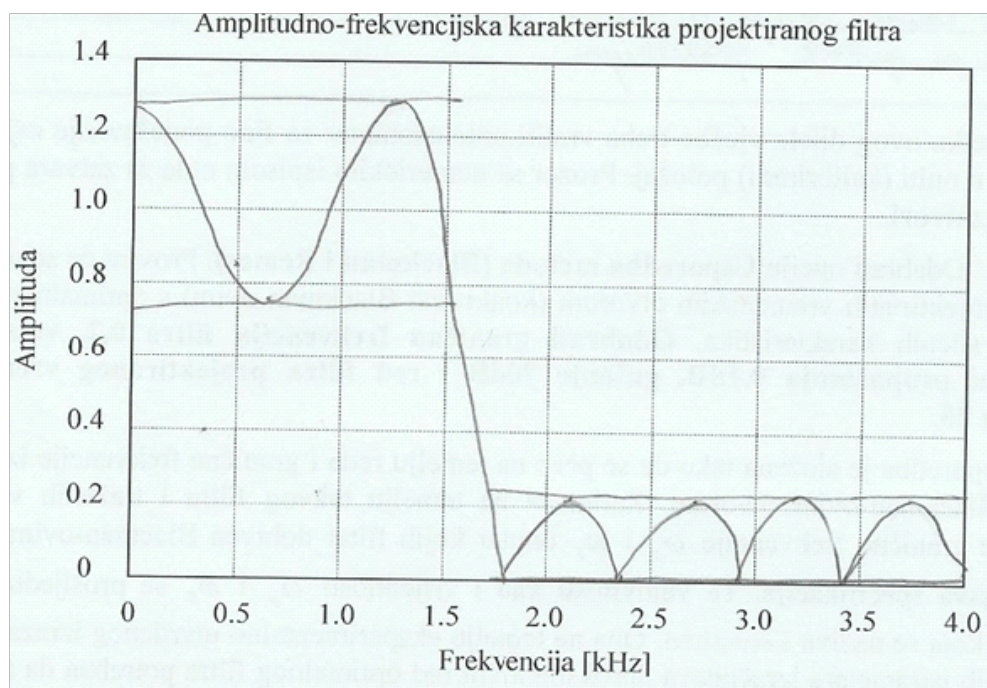
4. U programu vratiti $ws = 0.3$, a umjesto reda $N = 12$ staviti $N = 20$, izvesti programa i pogledati slike i izračunati valovitosti. Koja je promjena u odnosu na zadatak 2.?

IMPLEMENTACIJA OPTIMALNIH FILTARA NA MAKETI

5. U nastavku vježbe koristiti će se program vj4 koji automatski provodi postupak projektiranja optimalnih filtara i implementira ih na DSP maketi. Nakon pokretanja programa, u novootvorenom grafičkom sučelju mogu se odabirati parametri filtra te pogledati karakteristike odabirom **Amplitudno frekvencijska karakteristika**. Pojedina područja (propuštanje, gušenje) se mogu povećati ukoliko se mišem klikne negdje unutar tog područja. U naslovu povećane slike se ujedno ispisuju maksimalne i minimalne vrijednosti valovitosti za to područje. Klikom na sliku vraćamo se na prikaz cjelokupne karakteristike.

U modu definiranja parametara filtra redom filtra (opcija **Red optimalnog filtra**) zadati slijedeće karakteristike filtra: **niski propust, granične frekvencije područja propuštanja i područja gušenja 0.2 i 0.22, omjer valovitosti u području propuštanja i području**

gušenja 1:1 i red N=15. Frekvenciju otipkavanja odabrati 8 kHz i aktivirati prikaz x osi u kHz. Amplitudno-frekvencijsku karakteristiku filtra ucrtati na slijedeći graf.



U tablicu upisati frekvencije i amplitude ekstremalnih (maksimalnih i minimalnih) vrijednosti karakteristike te frekvencije prolazaka karakteristike kroz nulu. Koristiti povećavanje prikaza.

Tablica na osnovu podataka iz MATLAB-a:

Frekv. [kHz]		0.65	1.3	1.6	1.8	2.2	2.3	2.6	2.9	3.2	3.4	3.7	4
Amplituda		0.74	1.26	0.74	0	0.26	0	0.26	0	0.26	0	0.26	0

Na DSP maketi odabrati frekvenciju iznosa 320Hz kod koje je izlaz filtra približno jednak 1. Na osciloskopu uključiti samo drugi kanal. Potenciometrom za otklon po y-osi osciloskopa uz ovu frekvenciju ulaznog signala podesiti prikaz izlaznog signala kroz šest podioka i to od prvog do sedmog (koristiti potenciometar za fino podešavanje-kalibraciju). Pritisnuti tipku **Napravi i pokreni DSP program**, čime se prenosi i starta program za filtriranje na DSP maketi. Mijenjati frekvenciju ulaznog signala od minimalno moguće do polovine frekvencije otipkavanja i verificirati rad filtra. Mjerenje se provodi tako da se postepeno diže frekvencija signala i pri tome na osciloskopu traže ekstremi i nule prijenosne funkcije. Ustanoviti da li filter ima ekstreme i nule na mjestima koja se mogu vidjeti na gore nacrtanom grafu. **Nakon početnog podešavanja potenciometar za otklon po y-osi osciloskopa se on više ne smije dirati.**

Zašto zadnji vrh u području gušenja nije jednak iako se radi o filteru s jednolikom valovitošću u području gušenja?

Zbog aliasinga, tj. zadnji vrh je manji od ostalih zbog početka djelovanja AA filtra na maketi

6. Odabrati opciju **Polarni prikaz**. U novo otvorenom prozoru ispisane su numeričke vrijednosti nula prijenosne funkcije filtra. Pošto su nule općenito kompleksni brojevi za svaku nulu (redak) dana su tri podatka i to redom: frekvencija na kojoj se nula pojavljuje, udaljenost nule od ishodišta (radius) i kut nule u stupnjevima (proporcionalan frekvenciji). Na grafu su prikazani položaji nula u polarnom prikazu. Jedinična kružnica je posebno istaknuta.

Na osnovu preslikane amplitudno frekvencijske karakteristike projektiranog filtra i ispunjene tablice iz prethodne točke, polarnog prikaza rasporeda nula i danih pripadnih

numeričkih iznosa, objasniti međuzavisnost položaja (iznosa) nula i prijenosne karakteristike filtra. Ucertati na gornjem grafu koja nula utječe na koji dio frekvencijske karakteristike.

Po završetku ovog dijela vježbe treba vratiti potenciometar za fino podešavanje osjetljivosti po y-osi u nulti (kalibrirani) položaj. Prozor sa numeričkim ispisom nula se zatvara pritiskom na tipku **zatvori**.

7. Odabrati opciju **Usporedba metoda (Blackman i Remez)**. Provest će se usporedba filtara projektiranih vremenskim otvorom (konkretno Blackman-ovim) s optimalnim (remez) filtrima sličnih karakteristika. **Odabrati graničnu frekvenciju filtra 0.2, valovitost u području propuštanja 0.1dB, gušenje 70dB i red filtra projektiranog vremenskim otvorom 20.**

Usporedba je složena tako da se prvo na temelju reda i granične frekvencije izračunava FIR s Blackman-ovim otvorom. Zatim se na temelju takvog filtra i traženih valovitosti određuju granične frekvencije ω_p i ω_s unutar kojih filter dobiven Blackman-ovim otvorom zadovoljava specifikacije. Te valovitosti kao i vrijednosti ω_p i ω_s se prosljeđuju jednoj funkciji koja se naziva `remezord`. Ona na temelju eksperimentalno utvrđenog izraza (4.14) iz navedenih parametara izračunava aproksimativni red optimalnog filtra potreban da se ovakva karakteristika ostvari. Na temelju svih navedenih parametara remez funkcija zatim ostvaruje optimalni filter.

Nakon unošenja gornjih parametara, promjenom reda optimalnog filtra ostvariti da su karakteristike oba filtra podjednake. Koliki red je potreban za optimalni filter istih karakteristika?

Zatim smanjiti red filtra s vremenskim otvorom recimo na 15, a red optimalnog filtra ne mijenjati. Promotriti što se događa sa karakteristikama filtara, posebice optimalnog. Zatim povećati red filtra s vremenskim otvorom na 25 i sada ponovno pogledati što se dogodilo. Ukratko opisati.

Ono što se može uočiti jest da se smanjenjem reda filtra s vrem. otvorom širi prijelazno područje, ali je gušenje isto kao i prije (ono se ne mijenja s redom filtra). Optimalni filter koji se dobije ima veće gušenje jer je prijelazno područje šire (zahtjev na filter je slabiji). U slučaju povećanja reda filtra s vremenskim otvorom, prijelazno područje se sužava, a optimalni filter izračunat na temelju novih granica područja ima manje gušenje (filter s užim prijelaznim područjem je teže realizirati pa je valovitost veća).

Za one koji žele i više

8. Projektiranje pojasno-propusnog filtra vrši se tako da se umjesto dva definiraju tri područja (područje gušenja, područje propuštanja, područje gušenja). U tu svrhu definiramo vektor graničnih frekvencija $[0 \ 0.15 \ 0.2 \ 0.3 \ 0.35 \ 0.5]$, pripadni vektor amplituda $[0 \ 0 \ 1 \ 1 \ 0 \ 0]$ i vektor težinskih faktora $[1 \ 1 \ 1]$. Nadalje, projektiranje optimalnih filtara može se provesti i zadavanjem željenih valovitosti (opcija **Iznos dozvoljenih valovitosti**) na osnovu čega program izračunava potreban red filtra i stvarne granične frekvencije. Mijenjati parametre filtra i komentirati dobivene rezultate.

5. VJEŽBA

Projektiranje IIR filtera bilinearnom transformacijom i njihova realizacija direktnom formom II

U ovoj vježbi radi se projektiranje digitalnih filtera s beskonačnim impulsnim odzivom (*engl. Infinite Impulse Response - IIR*) ili tzv. rekurzivnih filtera metodom bilinearne transformacije. Za projektiranje samog filtera, međutim, kao i kod metode s jednakim impulsnim odzivom, koriste se postupci razvijeni za **projektiranje analognih filtera**. Razlozi za to leže u činjenici da je tokom vremena razvijen čitav niz metoda za dizajn analognih filtera (pa zašto onda to ne iskoristiti !), a osim toga ti se postupci uglavnom temelje na relativno jednostavnim (*tzv. closed-form*) izrazima. Ovi postupci međutim ne vode direktno do isto tako jednostavnih postupaka za digitalne filtre.

Bilinearna transformacija predstavlja jedan način kod kojeg se prijenosna funkcija analognog filtera $H_c(s)$ projektiranog u vremenski kontinuiranoj s -domeni preslikava u z -domenu da bi se tako ostvario digitalni filter. Ova transformacija dana je slijedećim izrazom:

$$s = \frac{2}{T} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right). \quad (5.1)$$

Danom transformacijom se cijela $j\Omega$ os iz s -ravnine preslikava u jedan obilazak po jediničnoj kružnici u z -ravnini, a lijeva s -poluravnina se preslikava unutar jedinične kružnice. Veza između kontinuirane frekvencije Ω i diskretne frekvencije ω dana je parom izraza:

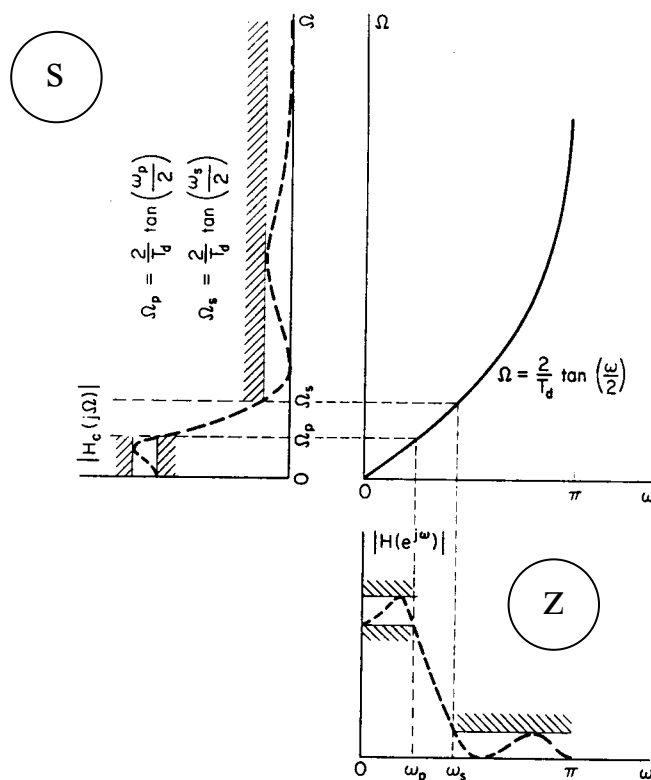
$$\Omega = \frac{2}{T} \operatorname{tg} \left(\frac{\omega}{2} \right), \quad (5.2)$$

$$\omega = 2 \operatorname{arctg} \left(\frac{\Omega T}{2} \right). \quad (5.3)$$

S obzirom da je krajnji cilj projektiranje digitalnog filtera, specifikacija filtera (rubne ili granične frekvencije i dopuštene valovitosti) **definira se u vremenski diskretnoj, z -domeni**. Na temelju specifikacije digitalnog filtera, najprije se moraju dobiti **specifikacije analognog filtera** što se izvodi primjenom izraza (5.2), tj. postupkom tzv. *frequency prewarping*. Ova transformacija vidi se i na Slici Slika 5.1. U cijelom ovom postupku se period otipkavanja T može zanemariti (tj. smatrati jednakim 1) jer se prvo primjenom izraza (5.2), a zatim primjenom (5.1) njegovo djelovanje dokine. Sa dobivenim graničnim frekvencijama u vremenski kontinuiranoj domeni ulazi se u postupak **projektiranja referentnog analognog filtera**.

Kao referentni filteri u s -domeni, najčešće se koriste filteri dobiveni slijedećim aproksimacijama: Butterworthova, Čebiševljeva tip I i tip II i eliptička (Cauerova). Svojstva ovih vremenski kontinuiranih filtera su već detaljno istražena kao i postupak njihovog projektiranja. Filteri realizirani ovim aproksimacijama se razlikuju po ponašanju u području propuštanja i području gušenja. Butterworthov filter je monoton u oba pojasa; Čebiševljev tipa I ima monotonu karakteristiku u području gušenja, a valovitu (*engl. equiripple*) u području propuštanja, dok je Čebiševljev tipa II upravo obrnute karakteristike; te na kraju, eliptički je valovit u oba pojasa, čime se za isti red postiže najuže prijelazno područje.

Kada je analogni filter izračunat, rezultirajuća prijenosna funkcija u s-domeni $H_c(s)$ se zatim bilinearnom transformacijom prema izrazu (5.1) **prevodi u vremenski diskretnu domenu** te se dobiva digitalni filter definiran prijenosnom funkcijom $H(z)$. Iz izraza (5.2) i (5.3) je vidljivo da je preslikavanje iz $-\infty \leq \Omega \leq \infty$ na $-\pi \leq \omega \leq \pi$ nelinearna operacija, o čemu treba voditi računa kod primjene ove metode. Transformacija je prikazana i grafički (Slika 5.1). Zbog nelinearnog preslikavanja, potrebno je, kao što je već rečeno, napraviti preslikavanje rubnih frekvencija područja filtra iz z-domene u s-domenu. Tako će konačna prijenosna funkcija diskretnog sustava dobivenog bilinearnom transformacijom imati željene rubne frekvencije područja. Treba napomenuti da ako je analogni filter bio stabilan, bit će stabilan i digitalni filter.



Slika 5.1. Grafički prikaz bilinearne transformacije

U programskom paketu MATLAB postoje četiri funkcije (aproksimacije) kojima se projektiraju IIR filteri metodom bilinearne transformacije. To su: 'butter', 'cheby1', 'cheby2', i 'ellip'. Na osnovu zadanih parametara (red filtra, tip filtra, granične frekvencije, valovitosti u području propuštanja i području gušenja), ove funkcije izvode prethodno opisan postupak projektiranja i kao rezultat daju koeficijente polinoma brojnika i nazivnika prijenosne funkcije $H(z)$. Rezultat se alternativno može dobiti u obliku polova, nula i konstantnog faktora ili u obliku jednadžbi stanja. Najčešći pristup kod projektiranja IIR filtera i filtera općenito je da se prvo definira željena karakteristika filtra. Sam red filtra kojim se željena karakteristika može ostvariti najčešće nije unaprijed poznat. Zbog toga i kod projektiranja filtera u Matlabu postoji drugi pristup. On podrazumijeva korištenje funkcija (za svaki tip aproksimacije) za određivanje minimalno potrebnog reda filtra za realizaciju željene karakteristike dane tipom filtra, valovitostima u području propuštanja odnosno gušenja, te rubnim točkama područja propuštanja, odnosno gušenja.

Još treba napomenuti da se bilinearnom transformacijom mogu projektirati svi tipovi filtera (NP, VP, PP, PB). To međutim znači da se za dotični tip mora isprojektirati odgovarajući analogni filter. Kod toga, sam postupak projektiranja analognog filtra

najjednostavniji je za niski propust, dok je za ostale tipove nešto složeniji. S obzirom da postoje transformacije u s-domeni kojima se niskopropusni filter može prevesti u bilo koji drugi tip, projektiranje filtera se zbog jednostavnosti često svodi na transformaciju specifikacije referentnog analognog filtera u domenu NP filtera, projektiranje odgovarajućeg NP filtera, te njegovu inverznu transformaciju u tip filtera od kojeg se krenulo. Na kraju se bilinearnom transformacijom iz analognog dobije digitalni filter. Ovaj postupak primijenjen je i kod projektiranja digitalnih IIR filtera u Matlabu. Treba još reći da i u diskretnoj z-domeni postoje izrazi za frekvencijske transformacije (tj. pretvorbu NP filtera u drugi tip).

Direktna realizacija IIR filtera

Diskretni IIR filter se na osnovu prijenosne funkcije $H(z)$ ili položaja polova i nula može realizirati na više načina, pri čemu se ističu tri osnovne realizacije: direktna, kaskadna i paralelna. U ovoj se vježbi radi sa direktnom realizacijom.

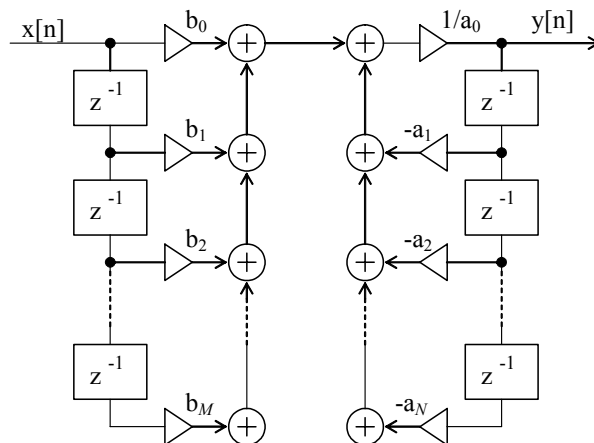
Ako je filter definiran prijenosnom funkcijom $H(z)$ oblika razlomljene racionalne funkcije:

$$H(z) = \frac{\sum_{m=0}^M b_m z^{-m}}{\sum_{m=0}^N a_m z^{-m}}, \quad (5.4)$$

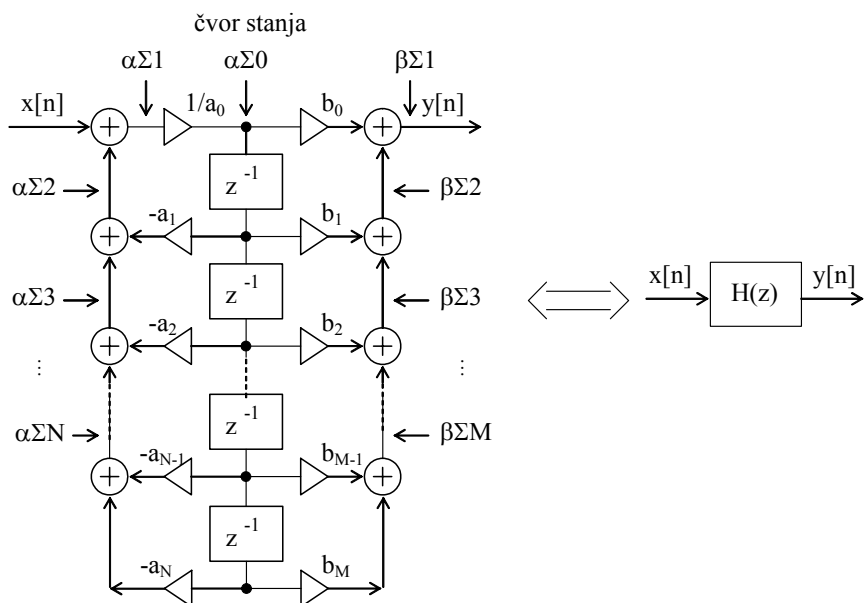
čiji su koeficijenti b_m i a_m dobiveni gore navedenom metodom, tada je pripadna jednačba diferencija dana izrazom:

$$a_0 y[n] = \sum_{m=0}^M b_m x[n-m] + \sum_{m=1}^N a_m y[n-m]. \quad (5.5)$$

Sistem opisan takvom jednačbom diferencija može se najjednostavnije realizirati tzv. direktnom formom I (Slika 5.2). Za praktičnu primjenu je nešto interesantnija (zbog manje elemenata) realizacija u obliku tzv. direktne forme II (Slika 5.3) koja proizlazi iz direktne forme I jednostavnim preslaganjem komponenti i objedinjavanjem elemenata za kašnjenje. Na slici su uneseni i nazivi pojedinih čvorova strukture.



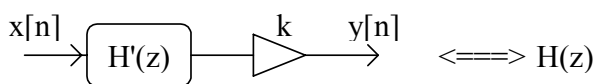
Slika 5.2. Realizacija filtera direktnom formom I



Slika 5.3. Realizacija filtra direktnom formom II

Kod primjene frakcionalne aritmetike radi implementacije ovakve realizacije filtra na DSP procesoru, treba voditi računa o iznosu koeficijenata b_m i a_m , te o dinamici signala u pojedinim čvorovima. U nastavku teksta bit će opisan jedan od načina realizacije filtera koji vodi računa o svim navedenim aspektima.

Kao rezultat projektiranja filtra u programu MATLAB, koeficijent a_0 je uvijek jednak 1, dok ostali koeficijenti mogu, ali i ne moraju biti manji od 1 po apsolutnoj vrijednosti. Najveća odnosno najmanja vrijednost koje se mogu prikazati u frakcionalnoj aritmetici jednake su 1 odnosno -1. Ako postoje koeficijenti filtra koji su veći od tih vrijednosti, **mora se provesti skaliranje**. Skaliranje se svodi na dijeljenje svih koeficijenata brojnika onim koeficijentom koji je po apsolutnoj vrijednosti najveći među njima. Analogno se radi sa koeficijentima nazivnika, s jedinom razlikom da se često umjesto s najvećim koeficijentom nazivnika dijeli s brojem koji je prva potencija broja 2 veća od tog najvećeg koeficijenta. Razlog tome leži u činjenici da se koeficijent a_0 u samoj strukturi koristi kao $1/a_0$, pa je nakon skaliranja veći od 1. S obzirom da se množenje s potencijama broja 2 de facto realizira kao posmak (*engl. shift*) sadržaja registra u lijevo za odgovarajući broj koraka, to je lakše realizirati nego množenje sa bilo kojim brojem većim od 1. Pošto se skaliranjem koeficijenata u stvari mijenja prijenosna funkcija filtra, da bi ukupna funkcija ostala ista, potrebno je signal na izlazu iz strukture pomnožiti s faktorom $k = b_{\max}/a_{\max}$. Blok shema rezultirajuće strukture dana je slikom (Slika 5.4).



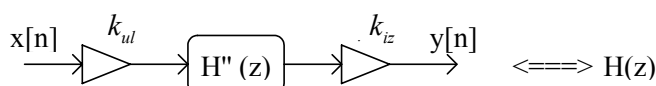
Slika 5.4. Nadomjesna shema nakon skaliranja koeficijenata nazivnika.

Kao što je već spomenuto, osim o veličini samih koeficijenata, treba voditi računa i o dinamici, tj. **veličini signala u pojedinim čvorovima strukture**. Ako je ta dinamika na nekom mjestu u strukturi prevelika, pripadni uzorci signala se neće moći prikazati uz korišteni broj bita što će uzrokovati velike pogreške u radu filtra. Jedan od načina kako se može isprojektirati filter da ne dolazi do prekoračenja dinamike, a da je **odnos signal-šum što bolji**, opisan je u ovom primjeru i korišten na laboratorijskim vježbama. pokušava se.

Dinamika se proračunava tako da se izračunava maksimalna vrijednost amplitudne karakteristike (tj. stacionarnog odziva sustava) u pojedinim čvorovima strukture. Drugim riječima, za i -ti čvor računa se $D(i) = \max_{\omega \in [0, \pi]} \left(|H_i(e^{j\omega})| \right)$. Ako je pojačanje filtra 1 (a sa takvima se ovdje radi) tada uz ulaz između ± 1 i izlaz mora biti u tim granicama. Ako je dinamika u čvoru $\alpha \Sigma 0$ unutar ± 1 tada su i sva stanja filtra (tj. vrijednosti iza elemenata za kašnjenje) unutar dinamike, a time i izlazi iza množila jer su koeficijenti već pravilno skalirani (a množenjem se ne može izaći iz dinamike). Do izlaska iz dinamike (*engl. overload*) može doći **isključivo na izlazima iz zbrajala** (čvorovi $\alpha \Sigma 1$ do $\alpha \Sigma N$, i $\beta \Sigma 1$ do $\beta \Sigma M$) i u čvoru stanja $\alpha \Sigma 0$.

Najprije se dinamike izračunavaju za nazivničku sekciju. Jedini način regulacije maksimalne dinamike u kritičnim čvorovima nazivničke sekcije postiže se **uvođenjem ulaznog pojačala** k_{ul} koje za potrebiti iznos priguši ulazni signal. Pojačanje tog pojačala inverzno je iznosu najvećeg $D(i)$ u nekom od navedenih čvorova.

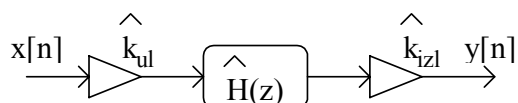
Nakon toga, dinamike se računaju za navedene čvorove brojničke sekcije. S obzirom da je nazivnički dio već podešen, jedini način da se ograniči dinamika u brojničkim čvorovima je **dodatno skaliranje koeficijenata brojnika** $b'' = scal \cdot b'$. Uvođenje dodatnog faktora skale kao i ulaznog pojačala uzrokuje promjenu prijenosne funkcije filtra pa u nadomjesnoj shemi filtra treba napraviti korekcije kako se vidi na slici (Slika 5.5). Izlazno pojačalo jednako je $k_{iz} = k / (k_{ul} \cdot scal)$.



Slika 5.5. Nadomjesna shema nakon ograničenja dinamike

Svi dosadašnji proračuni bili su napravljeni s realnim brojevima što dakako ne može biti pogodno za rad na DSP procesoru. Pošto su nakon ovih postupaka zadovoljeni svi uvjeti na iznose koeficijenata i dinamiku signala, dobiveni **realni koeficijenti se zaokružuju** na najbliže B -bitne frakcije.

Zaokruživanjem koeficijenata se dobiva nova prijenosna funkcija $\hat{H}(z)$ koja se razlikuje od $H''(z)$, pa se radi toga i dinamike proračunate uz realne koeficijente mijenjaju. **Postupak provjere dinamike se stoga mora ponoviti** sa zaokruženim B -bitnim koeficijentima. Ovaj postupak se osim toga razlikuje od prethodnog postupka provjere po tome što se sada najprije izračunavaju maksimalne dinamike **za sve navedene čvorove** i brojnika i nazivnika, a tek nakon toga se određuju pojačanja ulaznog i izlaznog pojačala u strukturi prema slici (Slika 5.5). Zbog činjenice da se jednom zaokruženi koeficijenti ne smiju više mijenjati, ulaznim faktorom pojačanja se regulira ukupna dinamika strukture, tako da ni u jednom čvoru brojnika ni nazivnika ne dolazi do prekoračenja dinamike. Izlaznim pojačalom kompenzira se djelovanje ulaznog pojačala da bi ukupni izlaz iz filtra bio u granicama ± 1 . Na kraju i ulazno i izlazno pojačalo također treba zaokružiti na B bita (dobije se \hat{k}_{ul} i \hat{k}_{izl}). Konačna nadomjesna shema dana je na slijedećoj slici, (Slika 5.6):



Slika 5.6. Konačna nadomjesna shema direktne realizacije

Za one koji žele znati više

Postupak projektiranja filtra ukratko opisan u prethodnom tekstu, bit će u nastavku dan po koracima sa navedenim svim potrebnim izrazima.

1) Nađi najveći koeficijent brojnika i nazivnika po apsolutnoj vrijednosti i neka su to pozitivni brojevi a_{\max} i b_{\max} .

2) Podijeli koeficijente brojnika s b_{\max} , čime se dobivaju koeficijenti b_0' do b_M' .

3a) Ako je $a_{\max} > 1$ (neki od a_1 do a_N je po modulu veći od 1), nađi prvu potenciju broja 2 veću od a_{\max} , neka je to $a_{\max 2}$, te podijeli koeficijente nazivnika s $a_{\max 2}$. Na taj način su svi skalirani koeficijenti od a_1' do a_N' manji od 1, a $1/a_0'$ je cijeli broj veći od 1, ali je potencija broja 2. Takvo pojačalo se realizira običnim shift-om (posmakom) u lijevo za odgovarajući broj koraka.

3b) Inače, ako je $a_{\max} = 1$, ostavi koeficijente nazivnika kakvi jesu, a $a_{\max 2}$ neka je jednak 1.

4) Pošto se pod 2) i 3) koeficijenti strukture mogu promijeniti, dobiva se nova prijenosna funkcija $H'(z)$, iste realizacije (Slika 5.3.), ali s novim koeficijentima. Da bi kompenzirali promjenu koeficijenata (kako se ukupni $H(z)$ ne bi promijenio), potrebno je uvesti jedno pojačalo na izlazu strukture koje ima pojačanje $k = b_{\max}/a_{\max 2}$. Ovim se završava postupak inicijalnog skaliranja koeficijenata.

Postupak proračuna dinamike kreće od proračuna za nazivnik i odvija se na slijedeći način:

5) proračun dinamike za čvor $\alpha\Sigma 0$ prema izrazu:

$$D(\alpha\Sigma 0) = \max_{\omega \in [0, \pi]} \left(\left| H_{\alpha\Sigma 0}(e^{j\omega}) \right| \right), \text{ uz } H_{\alpha\Sigma 0}(z) = \frac{1}{\sum_{n=0}^N a_n' z^{-n}} \quad (5.6)$$

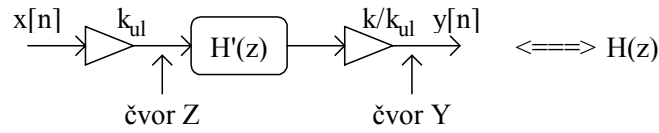
6) proračun dinamike za čvorove $\alpha\Sigma 2$ do $\alpha\Sigma N$ prema slijedećim izrazima, uz $i=2$ do N :

$$D(\alpha\Sigma i) = \max_{\omega \in [0, \pi]} \left(\left| H_{\alpha\Sigma i}(e^{j\omega}) \right| \right), \text{ uz } H_{\alpha\Sigma i}(z) = \frac{-\sum_{j=i-1}^N a_j' z^{-j}}{\sum_{n=0}^N a_n' z^{-n}} \quad (5.7)$$

Pojačanje ulaznog pojačala nalazi se prema izrazu:

$$k_{ul} = \frac{1}{\max(D(\alpha\Sigma 0), D(\alpha\Sigma 2), \dots, D(\alpha\Sigma N), 1)} \quad (5.8)$$

Kao što se vidi iz izraza, k_{ul} ovisi o iznosu najveće dinamike po svim čvorovima, ali i o jedinici. Ako su sve maksimalne dinamike po čvorovima manje od 1, to bi značilo da se ulazni signal može pojačati i za faktor veći od 1. Međutim, tada bi dinamika u čvoru Z bila prevelika i signal se ne bi mogao prikazati u frakcionalnoj aritmetici, iz čega slijedi ograničenje na $k_{ul} \leq 1$. Pošto smo uveli ulazno pojačalo, pojačanje izlaznog pojačala treba korigirati. Nova shema realizacije dana je na slijedećoj slici (Slika 5.7).



Slika 5.7. Nadomjesna shema nakon ograničenja dinamike u čvorovima nazivnika.

Postupak proračuna dinamike nastavlja se proračunom za brojnik, na što utječe i pojačalo k_{ul} i cijeli nazivnik, i odvija se na slijedeći način:

7) proračun dinamike za čvorove $\beta\Sigma 1$ do $\beta\Sigma M$ prema slijedećim izrazima, uz $i=1$ do M :

$$D(\beta\Sigma i) = \max_{\omega \in [0, \pi]} \left(\left| H_{\beta\Sigma i}(e^{j\omega}) \right| \right), \text{ uz } H_{\beta\Sigma i}(z) = k_{ul} \cdot \frac{\sum_{j=i-1}^M b_j' z^{-j}}{\sum_{n=0}^N a_n' z^{-n}} \quad (5.9)$$

Ako se nakon ovog koraka ustanovi da je dinamika pojedinih čvorova veća od dozvoljene, računa se faktor skale kojim trebamo pomnožiti koeficijente brojnika b_0' do b_M' , da bi se osigurali od prekoračenja dinamike. Osim u čvorovima $\beta\Sigma 1$ do $\beta\Sigma M$, i dinamika izlaza (čvor Y na Slici 5.4., $D(Y) = k/k_{ul} \cdot D(\beta\Sigma 1)$) mora biti ograničena na ± 1 , jer smo krenuli od pretpostavke da filter ima najviše pojačanje 1. Broj s kojim se množe koeficijenti brojnika se u skladu s tim dobiva prema izrazu:

$$scal = \frac{1}{\max(D(\beta\Sigma 1), \dots, D(\beta\Sigma M), D(Y))} \quad (5.10)$$

8) skaliranje koeficijenata brojnika i izlaznog pojačanja

$$\begin{aligned} b_i'' &= scal \cdot b_i', \quad i = 0 \text{ do } M, \\ \frac{k}{k_{ul}} &\Rightarrow k_{izl} = \frac{k}{k_{ul} \cdot scal} \\ H'(z) &\Rightarrow H''(z) \end{aligned} \quad (5.11)$$

Ako dinamika niti jednog čvora nije veća od ± 1 , tada je dozvoljeni faktor $scal=1$ i koeficijenti brojnika se postupkom pod 8) ne mijenjaju.

Pošto su nakon ovih postupaka zadovoljeni svi uvjeti na iznose koeficijenata i dinamiku signala, dobivene realne koeficijente treba zaokružiti na najbliže B -bitne frakcije.

9) zaokruživanje koeficijenata a_i' i b_i'' na B bita prema izrazima

$$\hat{a}_i = \frac{\text{round}(a_i' \cdot 2^{(B-1)})}{2^{(B-1)}}, \quad \hat{b}_i = \frac{\text{round}(b_i'' \cdot 2^{(B-1)})}{2^{(B-1)}} \quad (5.12)$$

Zaokruživanjem koeficijenata se dobiva nova prijenosna funkcija $\hat{H}(z)$ koja se razlikuje od $H''(z)$, pa se radi toga mora ponoviti i proračun dinamika.

10) proračun dinamike za čvor ($\alpha\Sigma 0$) prema izrazu:

$$\hat{D}(\alpha\Sigma 0) = \max_{\omega \in [0, \pi]} \left(\left| \hat{H}_{\alpha\Sigma 0}(e^{j\omega}) \right| \right), \text{ uz } \hat{H}_{\alpha\Sigma 0}(z) = \frac{1}{\sum_{n=0}^N \hat{a}_n z^{-n}} \quad (5.13)$$

11) proračun uz zaokružene koeficijente za $\alpha\Sigma 2$ do $\alpha\Sigma N$, te $\beta\Sigma 1$ do $\beta\Sigma M$

$$\hat{D}(\alpha\Sigma i) = \max_{\omega \in [0, \pi]} \left(\left| \hat{H}_{\alpha\Sigma i}(e^{j\omega}) \right| \right), \text{ uz } \hat{H}_{\alpha\Sigma i}(z) = \frac{\sum_{j=i-1}^N \hat{a}_j' z^{-j}}{\sum_{n=0}^N \hat{a}_n' z^{-n}} \quad (5.14)$$

$$\hat{D}(\beta\Sigma i) = \max_{\omega \in [0, \pi]} \left(\left| \hat{H}_{\beta\Sigma i}(e^{j\omega}) \right| \right), \text{ uz } \hat{H}_{\beta\Sigma i}(z) = \frac{\sum_{j=i-1}^M \hat{b}_j' z^{-j}}{\sum_{n=0}^N \hat{a}_n' z^{-n}} \quad (5.15)$$

12) traženje faktora pojačanja ulaznog i izlaznog pojačala prema izrazima

$$k_{naz} = \frac{1}{\max(\hat{D}(\alpha\Sigma 0), \hat{D}(\alpha\Sigma 2), \dots, \hat{D}(\alpha\Sigma N))} \quad (5.16)$$

$$k_{br} = \frac{1}{\max(\hat{D}(\beta\Sigma 1), \dots, \hat{D}(\beta\Sigma M))} \quad (5.17)$$

$$k_{ul} = \min(k_{br}, k_{naz}, 1) \quad \hat{k}_{ul} = \frac{\text{floor}(k_{ul} \cdot 2^{(B-1)})}{2^{(B-1)}} \quad (5.18)$$

$$k_{izl} = \frac{1}{\hat{D}(\beta\Sigma 1) \cdot \hat{k}_{ul}} \quad \hat{k}_{izl} = \frac{\text{floor}(k_{izl} \cdot 2^{(B-1)})}{2^{(B-1)}} \quad (5.19)$$

Na temelju izračunatih maksimalnih dinamika brojnika i nazivnika, pojačanje ulaznog pojačala k_{ul} se dobiva kao manja vrijednost između izračunatih k_{br} i k_{naz} , uz dodatno ograničenje da ulazno pojačanje smije biti maksimalno jednako 1 (objašnjenje za ovaj zadnji uvjet je već dano kod prvobitnog izračunavanja k_{ul}). Pošto je pojačanje ulaznog pojačala podešeno kako je navedeno u prethodnom paragrafu, pojačanje izlaznog pojačala se računa tako da dinamika na izlazu filtra bude 1, kao što je bilo pretpostavljeno na početku. Vrijednosti realnog (k_{izl}) i konačnog zaokruženog pojačanja izlaznog pojačala (\hat{k}_{izl}) izračunavaju se prema izrazima (5.19). Konačni parametri direktne realizacije IIR filtra dani su koeficijentima brojnika \hat{b}_i i koeficijentima nazivnika \hat{a}_i koji čine $\hat{H}(z)$, te atenuatorom \hat{k}_{ul} i pojačalom \hat{k}_{izl} , kako je dano slikom (Slika 5.6).

Direktna realizacija IIR filtra na DSP procesoru

Implementacija IIR filtra provedena je na DSP procesoru AD2181 ugrađenom na DSP maketi. Filtar je realiziran direktnom formom II. U procesu određivanja koeficijenata (programima u programskom paketu MATLAB) provedeno je skaliranje koeficijenata brojnika i nazivnika radi ograničavanja maksimalne dinamike u svim čvorovima filtra. Konačna realizacija izvedena na DSP procesoru odgovara slici, Slika 5.6, uz samu jezgru filtra ($H(z)$) prema slici, Slika 5.3. Verzija programa prikazana u nastavku radi s 16-bitnim koeficijentima. Ulazni i izlazni signal predstavljeni su 16-bitnim frakcijama. Nedostaje dio za komunikaciju sa AD/DA pretvornikom (*engl. codec*) koji nije interesantan sa stanovišta funkcioniranja filtra.

```

{Dio 1}
{ _Definicija modula i konstanti_ }
MODULE/RAM          SDIR;          { _Ime programa_ }
CONST              N=20;          { _Red filtra_ }

{ _Definicija ulazih točaka u modul_ }
ENTRY ZAGLAVLJE;
ENTRY OBRADA;

{ _Definicija varijabli_ }
VAR/PM/RAM/CIRC     COEFS[1+2*(N+1)+2]; { _Koeficijenti filtra_ }
VAR/DM/RAM/CIRC     DELAY[N];          { _Stanja filtra_ }
VAR/DM/RAM          sampl_rate;        { _Frekvencija otipkavanja_ }
GLOBAL              sampl_rate;

{ _Učitavanje koeficijenata iz datoteke prilikom linkanja_ }
INIT                COEFS : <SDIR.CFS>;
INIT                sampl_rate : FS_CFS;

{Dio 2}
{ _Ovaj dio se izvodi jedno prilikom inicijalizacije filtra_ }
ZAGLAVLJE:
    I2=^DELAY;          { _Pokazivač na stanja_ }
    I4=^COEFS;          { _Pokazivač na koeficijente_ }
    L2=%DELAY;          { _Cirkularno adresiranje_ }
    L4=%COEFS;
    M2=1;               { _Offset registri_ }
    M3=0;
    M4=1;

{ _Inicijalno brisanje stanja filtra_ }
MX0=0;
CNTR=%DELAY;
DO CLRBUF UNTIL CE;
CLRBUF: DM(I2,M2)=MX0; { _nuliraj stanja filtra_ }
RTS;

{Dio 3}
{ _Ovaj dio se izvodi za svaki otipkani uzorak_ }
OBRADA: { U MX1 nalazi se ulazni podatak }
        MX0 = MX1; { _Stavi podatak u MX0_ }
        MY0=PM(I4,M4);
        MR=MX0*MY0(SS),MX0=DM(I2,M2),MY0=PM(I4,M4);
        CNTR=N-1; { postavljanje brojača }
        DO NA_LOOP UNTIL CE; { petlja za računanje nazivnika }
NA_LOOP: MR=MR-MX0*MY0(SS),MX0=DM(I2,M2),MY0=PM(I4,M4);
        MR=MR-MX0*MY0(SS),AX0=PM(I4,M4); { učitaj 1/a0 u AX0...pot. broja 2 }

{ _Provjera da li se može množiti s 1/a0_ }
IF MV JUMP NA_SAT;
SE=EXP MR1(HI),MX0=DM(I2,M3); { eksponent trenutnog rezultata }
AY0=SE;
AR=AX0+AY0,MY0=PM(I4,M4); { suma eksponenata }
IF LE JUMP NA_NORM; { može se množiti, odi gdje treba }
AR=PASS MR1; { postavi flagove }
IF GE JUMP NA_POZ; { umnozak veći od 1, skaci }
MR1=0X8000; { umnozak 'veći' od -1, saturate }
JUMP BROJNIK
NA_POZ: MR1=0X7FFF; { umnozak 'veći' od 1, saturate }
JUMP BROJNIK;
NA_SAT: MX0=DM(I2,M3),MY0=PM(I4,M4); { pripremi za brojnik }
        IF MV SAT MR; { ako je overflow, saturate }
        JUMP BROJNIK;

{ _množenje realizirano kao shift gornjeg i donjeg dijela_ }
NA_NORM: SE=AX0;
        SR=ASHIFT MR1(HI);
        SR=SR OR LSHIFT MR0(LO);

```

```

MR0=SR0;
MR1=SR1;
MR=MR(RND);

{ _racunanje_brojnickog_FIR_dijela_ }
BROJNIK: DM(I2,M2)=MR1,MR=MR1*MY0(SS);
MY0=PM(I4,M4);
CNTR=N;
DO BR_LOOP UNTIL CE;
BR_LOOP: MR=MR+MX0*MY0(SS),MX0=DM(I2,M2),MY0=PM(I4,M4);
IF MV JUMP BR_SAT;
MX0=MR0;
MX1=MR1,MR=MX0*MY0(US);
MR0=MR1;
MR1=MR2;
MR=MR+MX1*MY0(SS),AX0=PM(I4,M4);
SE=EXP MR1(HI);
AY0=SE;
AR=AX0+AY0;
IF LE JUMP BR_NORM;
AR=PASS MR1;
IF GE JUMP BR_POZ;
MR1=0X8000;
JUMP IZLAZ;
BR_POZ: MR1=0X7FFF;
JUMP IZLAZ;
BR_SAT: AX0=PM(I4,M4);
IF MV SAT MR;
JUMP IZLAZ;
{ _množenje_s_izlaznim_pojačalom_ }
BR_NORM: SE=AX0;
SR=ASHIFT MR1(HI);
SR=SR OR LSHIFT MR0(LO);
MR0=SR0;
MR1=SR1;
MR=MR(RND);
{ _izlazni_filtirani_podatak_ }
IZLAZ: MX1=MR1;
RTS;
.ENDMOD;

```

Analizom programa može se zaključiti da procesor mora po jednom uzorku ulaznog signala izvršiti P operacija prema slijedećem izrazu

$$P = 60 + 2 \cdot N \quad (5.20)$$

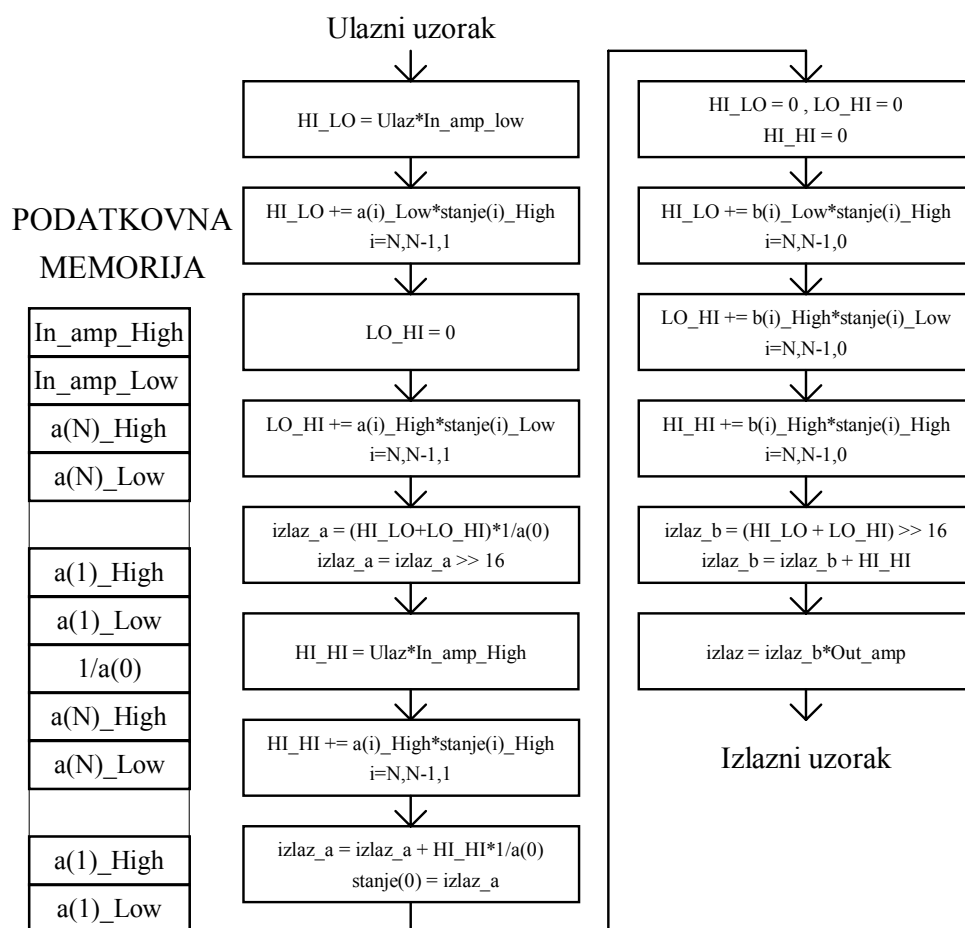
gdje je N red filtra. Da bi program ispravno radio procesor mora imati dovoljno vremena da obradi jedan uzorak ulaznog signala (jedan odziv na prekid (*engl. interrupt*)) prije nego što naiđe slijedeći. Taj zahtjev je prikazan nejednakošću

$$f_{proc} > P \cdot f_o \quad (5.21)$$

gdje je f_o frekvencija otipkavanja, a f_{proc} frekvencija izvršavanja instrukcija. Kombiniranjem izraza (5.20) i (5.21) dobiva se izraz za maksimalni red filtra, koji se može realizirati ovim programom

$$N < \frac{1}{2} \cdot \left(\frac{f_{proc}}{f_o} \right) - 30 \quad (5.22)$$

što uz frekvenciju izvršavanja instrukcija (rada procesora) $f_{proc} = 33.33 MHz$ i maksimalne frekvencije otipkavanja $f_o = 48 kHz$ iznosi $N=316$.



Slika 5.8. (lijevo) Raspored koeficijenata u memoriji za 32 bitni slučaj.

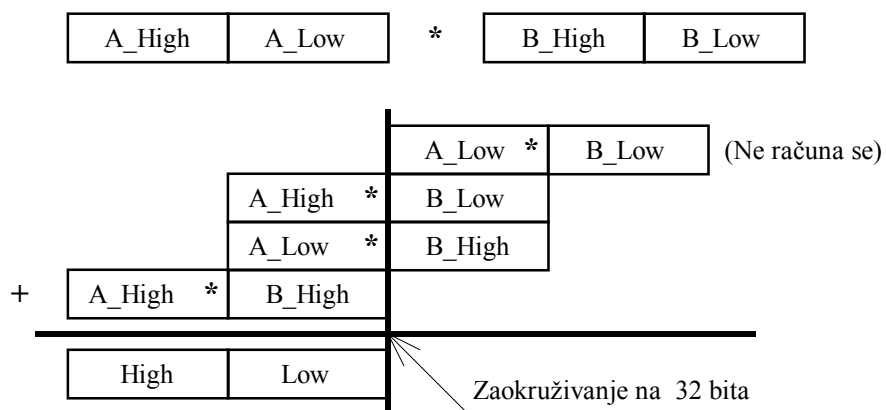
Slika 5.9. (desno) Dijagram toka za direktnu realizaciju II u dvostrukoj preciznosti.

U slučaju **realizacije sa 32 bitnim koeficijentima** (koriste se po 2 16-bitna registra, odnosno memorijske lokacije), potrebno je za razliku od FIR slučaja imati i 32 bitna stanja. Sama realizacija je bolje od prethodnog slučaja jer se zahvaljujući duljim koeficijentima (veći broj bita) **moгу realizirati filtri za koje je 16 bitna realizacija nemoguća** kao što će biti pokazano na vježbama. Na slici, Slika 5.8, prikazan je raspored koeficijenata u podatkovnoj memoriji dok su stanja filtra smještena u programsku memoriju i to po dvije riječi za svako stanje (viša 16-bitna riječ, niža 16-bitna riječ).

Zbog što efikasnijeg izvršavanja programa korištena je realizacija prema dijagramu toka prikazanim slikom (Slika 5.9.). Pojašnjenje množenja korištenog u realizaciji dano je na slici, Slika 5.10. Svaki redak na slici se provodi za sve koeficijente (brojnika ili nazivnika) i tek onda se provodi zaokruživanje na gornjih 16 bita. Na taj način se stalno učitavaju samo koeficijenti i stanja i to odvojeno viša i niža riječ, dok se 32-bitni međurezultat sprema u registre, a ne u memoriju. Time je broj pristupa memoriji (transfer podataka) smanjen.

Ukupno produljenje trajanja prolaska jednog uzorka kroz filter je na taj način povećano za oko tri puta, ali je time dobivena mogućnost realizacije filtera s većim zahtjevima na prijenosnu karakteristiku. Pogreška jedne 32-bitne multiplikacije unesena zbog ovakvog načina računanja iznosi (-1;0] bita radi izostavljanja umnoška nižih riječi ($A_{Low} * B_{Low}$) i [-0.5;0.5] bita zbog zaokruživanja na 16 bita umnoška miješanih kombinacija nižih i viših riječi ($A_{High} * B_{Low}$ i $A_{Low} * B_{High}$) te njihovog naknadnog pomaka 16 bita u desno što daje ukupnu grešku iznosa (-1.5;0.5) bita najmanje težine (LSB) 32-bitne riječi.

Ulazni i izlazni uzorci su i dalje 16 bitni što je ograničeno od strane ostatka sklopovlja implementiranog na DSP maketi.



Slika 5.10. Prikaz 32-bitnog množenja upotrebom više 16-bitnih množenja (svaki gornji blok je 16-bitni)

Zadatak za pripremu

Treba projektirati digitalni NP filter reda $N=2$, granične frekvencije $\omega_g = \pi/3$ pomoću bilinearne transformacije uz period otipkavanja $T=2$. Koristiti Butterworthovu aproksimaciju. Nacrtati direktnu realizaciju II filtra s upisanim kvantiziranim koeficijentima u obliku frakcija dobivenim zaokruživanjem. Koeficijente po potrebi skalirati. *Na raspolaganju imate 16-bitni*

Rješenje zadatka: *DSP procesor (1 bit je pretnat).*

Praktična vježba

MATLAB PROGRAM ZA REALIZACIJU IIR FILTARA BILINEARNOM TRANSF.

1. Napisati Matlab program (skriptu) kojim se izračunavaju koeficijenti niskopropusnog eliptičkog IIR filtra te prikazuju njegova amplitudna i fazno-frekvencijska karakteristika te karakteristika grupnog kašnjenja. Osim dobivanja koeficijenata filtra, treba realizirati filter kvantiziran na B bita, te napraviti odziv na step funkciju kvantiziranog i nekvantiziranog filtra. Slijediti upute u nastavku.

a) Najprije treba definirati parametre za ostvarenje željene karakteristike. Neka su rubne frekvencije propusnog područja i područja gušenja filtra: $\omega_p = 0.2$; i $\omega_s = 0.3$; (frekvencija 1 odgovara kružnoj frekvenciji π). Željena valovitost u području propuštanja neka je $R_p = 0.1 \text{ dB}$, a u području gušenja $R_s = 60 \text{ dB}$. Navedene parametre treba iskoristiti za određivanje **reda** N eliptičkog filtra koji će zadovoljiti navedene zahtjeve. To se izvodi korištenjem Matlab funkcije **ellipord** uz poziv: `[N,wc]=ellipord(wp, ws, Rp, Rs)`. Frekvencija w_c je rezultirajuća granična frekvencija filtra.

b) Osnovna funkcija za projektiranje eliptičkog filtera je funkcija **ellip**. Za ostvarenje niskopropusnog filtra treba ju pozvati na slijedeći način: `[b,a]=ellip(N,Rp,Rs,wc);`. Dobivaju se vektori s koeficijentima brojnika b (FIR sekcija) te nazivnika a (IIR sekcija) razlomljene racionalne funkcije kojom je opisan filter.

c) Na **prvoj slici** (prije svake slike staviti `figure`) prikazati 150 uzoraka impulsnog odziva dobivenog filtra naredbom `impz(b,a,150)`.

d) Izračunati frekvencijsku karakteristiku filtra $H(e^{j\omega})$ pomoću `freqz` u $brt = 1024$ točke, `om = [0 : (brt-1)] / brt*pi`; `H = freqz(b,a,om);`. Izračunati amplitudnu (u dB) i fazno-frekvencijsku karakteristiku. Na **drugoju slici** pomoću `subplot(3,1,x)` naredbe prikazati ove dvije karakteristike sa `plot` funkcijom jednu ispod druge. Ispod njih prikazati i karakteristiku grupnog vremena kašnjenja. Ovu zadnju prikazati korištenjem funkcije `grpdelay(b,a);`.

e) Na **trećoj slici** prikazati polove i nule prijenosne funkcije funkcijom `zplane(b,a)`.

2. Izvesti program, pogledati dobivene slike i odgovoriti na slijedeća pitanja. S kojim redom je realiziran filter? Da li je filter stabilan? _____ Zašto fazna karakteristika izgleda tako 'ružno'?

Ponoviti još jednom samo naredbe za izračunavanje i prikaz fazne karakteristike, ali tako da se primijeni funkcija **unwrap**, `unwrap(angle(H))` koja umjesto izračunavanja faze sa diskontinuitetima tj. između $-\pi$ i π , na mjestima diskontinuiteta dodaje potrebni višekratnik od 2π da bi karakteristika na tim mjestima bila kontinuirana. Treba imati u vidu da su skokovi od π na mjestima nula karakteristike regularni i imaju veze isključivo sa samim izgledom karakteristike. Sada se može bolje vidjeti da je **fazna karakteristika IIR filtra nelinearna**. Gdje je ta nelinearnost, a time i grupno vrijeme kašnjenja najveće? _____

3. U nastavku treba pripremiti koeficijente filtra za realizaciju. To podrazumijeva njihovo skaliranje da bi se mogli prikazati između ± 1 , te kvantizaciju na željeni broj bita.

Podešavanje dinamike ovdje ne treba raditi. Kod kojim se to ostvaruje treba u skripti staviti na mjesto **prije prikaza** frekvencijskih karakteristika. Slijediti upute.

a) Provesti skaliranje koeficijenata brojnika i nazivnika (koristiti funkciju `max`). Izračunati faktor k kojim treba pomnožiti karakteristiku ostvarenu sa skaliranim koeficijentima da bi ukupna karakteristika bila jednaka početnoj (pogledati teoretski dio vježbe, Slika 5.4).

b) Unijeti broj bita za kvantizaciju B pomoću funkcije `input`.

c) Kvantizirati skalirane koeficijente brojnika b_s zaokruživanjem na najbliži cijeli broj korištenjem ovakve naredbe: `bk=round(bs*2^(B-1))/2^(B-1);`. Ponoviti analogno za koef. nazivnika a_s i faktor k .

d) Izračunati frekvencijsku karakteristiku s kvantiziranim koeficijentima h_k . PAZI ! U brojniku više nije samo vektor kvant. koeficijenata b_k već umnožak b_k i kvant. faktora k_k .

e) U plot naredbama za prikaz amplitudne i fazno-frekvencijske karakteristike dodati i prikaz odgovarajućih kvantiziranih karakteristika **na istim grafovima**. Za podsjetnik, prikaz više krivulja na istom grafu se radio tako da se u plot naredbi za y os definira potrebn broj stupčastih vektora u ovom obliku $[x1 \ x2]$. Ako veličine koje prikazujemo nisu stupčani vektori već reci, treba ih prikazati transponirane, na ovaj način: $[x1' \ x2']$. Prije prikaza h i h_k treba provjeriti kojeg tipa su ti vektori.

f) Iza prikaza polova i nula nekvantizirane karakteristike, na **četvrtoj slici** prikazati polove i nule kvantizirane prijenosne funkcije.

g) Izračunati i prikazati odziv na step za nekvantizirani i kvantizirani filter na slijedeći način. Najprije definirati jediničnu stepenicu u `brt` uzoraka naredbom `s=ones(1,brt);`. Primjenom naredbe `filter`, npr. `y=filter(b,a,s);` izračunati svaki od odziva. Naredba **filter** služi za realizaciju IIR filtera pomoću tzv. transponirane direktne forme II (slična DIR II) te filtraciju signala danog varijablom s . PAZI ! Za skalirani i kvantizirani filter ponovno vrijedi da u brojniku nije b_k već $b_k \cdot k_k$. Prikazati odzive od 1. do 80-tog uzorka na **petoj slici** pomoću `subplot` naredbe jedan ispod drugog.

4. Zatvoriti sve slike i ponovno izvesti program. Za broj bita za kvant. uzeti $B=13$. Promotriti karakteristike, položaj polova i nula te odzive na step. Zatim izvesti program uz $B=11$. Iako se karakteristika nije drastično promijenila, što se dogodilo s polovima kvantizirane karakteristike ? Kako se to odražava na izgled odziva na step?

step - nestabilan

Izvesti program uz $B=10$. Uočiti da nije prikazana amplitudna karakteristika kvantiziranog filtra. Provjeriti koliko iznose faktor k i njegova kvantizirana vrijednost k_k . Zašto je k_k jednak 0? Kolika je vrijednost najmanje frakcije koja se može prikazati s 10 bita?

5. Zatvoriti sve slike osim prvih pet (uz $B=13$). U skripti promijeniti parametar na novi iznos $\omega_s = 0.25$; i ponovno izvesti program. Za broj bita za kvant. sada uzeti $B=20$. To je potrebn broj bita koji treba uzeti da ne postoji značajna razlika između originalne i kvantizirane karakteristike. Zašto je taj broj viši u odnosu na prethodni filter? Da li je i red filtra viši i zašto?

$N=8$; uže prijelazno područje

REALIZACIJA IIR FILTRA NA DSP MAKETI

Pokrenuti program **vj5** u MATLAB okruženju. U aplikaciji se projektira IIR filter odabrane realizacije i parametara. U ovoj vježbi radi se **direktna realizacija**. Za projektirani filter se odabirom prikaza sa padajućeg menija gore desno može promatrati prikaz polova i nula, amplitudno i fazno frekvencijska karakteristika (za cijelo ili samo pojedina frekvencijsko područje – klik mišem), te impulsni odziv i odziv na step.

Osim toga se na osnovu dobivenih koeficijenata filtera izračunavaju koeficijenti koji su prilagođeni za rad s frakcionalnom aritmetikom i koji će dalje biti korišteni za filtraciju na maketi. Parametar koji se može mijenjati je broj bita za zaokruživanje koeficijenata. Odabirom prikaza **Proračun dinamike** prolazi se cijeli postupak skaliranja i kvantizacije koeficijenata ukratko opisan u teoretskom dijelu vježbe. Grafički su prikazane usporedne karakteristike dobivene iz nekvantiziranih i kvantiziranih koeficijenata.

6. Odabrati sljedeći skup parametara za **direktnu realizaciju** filtera: **niski propust, Čebiševljev tip I, 4. red, granična frekvencija 0.25, valovitost u području propuštanja 1dB**. Odabrati zaokruživanje koeficijenata na **8 bita**. Odabrati za prikaz **Proračun dinamike** i pratiti korake određivanja koeficijenata.

Nakon određivanja koeficijenata za rad na maketi, u aplikaciji odabrati frekvenciju otipkavanja **8kHz**. Pritiskom na tipku **Start DSP (16 bita)** prenosi se 16-bitna verzija projektiranog filtera na DSP maketu. Po završetku prijenosa, program se sam starta. Promjenom ulazne frekvencije provjeriti ispravnost rada filtera.

Odabrati neku frekvenciju manju od polovine frekvencije otipkavanja i pritiskom na tipku 'interrupt' na DSP maketi prekinuti izvršavanje programa na DSP procesoru uz spremanje pobudne i odzivne sekvence duljine 4096 uzoraka u memoriju. Pritiskom na tipku **obrada 16** vrši se prijenos podataka u računalo te **usporedba filtriranja sa i bez kvantizacije stanja filtera** (DSP maketa odnosno MATLAB simulacija), pri čemu se u oba slučaja koriste isti kvantizirani koeficijenti na 16 bita i iste pobudne sekvence. Razlika odziva skalirana je na broj bita, tj. razlika od 1 LSB znači vrijednost $1 \cdot 1/2^{15} = 0.000305$. Opisati dobivene rezultate.

3 LSB

7. Projektirati filter sa sljedećim karakteristikama: **Pojasni propust, Eliptički, 5. red, granične frekvencije 0.3 i 0.4, valovitost u području propuštanja 2dB, valovitost u području gušenja 50dB**. Odabrati za prikaz **Proračun dinamike**. Obratiti pažnju na dinamike čvorova nazivničke sekcije. Da li je 16 bita za zaokruživanje koeficijenata dovoljno za realizaciju ovoga filtera? Da li je dovoljno 32 bita? Objasniti prednost 32-bitne realizacije u odnosu na 16-bitnu za slučaj 16-bitnog ulaznog signala? (Pogledati dio teksta ispod Slika 5.9 i Slika 5.10 na kraju teoretskog dijela vježbe.)

Pritiskom na tipku **Start DSP (32 bita)** prenosi se 32-bitna verzija projektiranog filtera na DSP maketu. Mijenjanjem frekvencije provjeriti rad filtera. Prekinuti program na DSP procesoru pritiskom na tipku 'interrupt' za proizvoljnu frekvenciju ulaznog signala unutar prijelaznog područja filtera. Pritiskom na tipku **obrada 32** vrši se usporedba filtriranja sa i bez kvantizacije stanja filtera. Koliko sada iznosi greška zbog kvantizacije stanja (PAZI, sada 1 LSB odgovara drugom iznosu) ? Koliko je to izraženo u 16 bita ($1+15$)?

$1.53 \cdot 10^{-15}$

8. Smanjiti red filtera na **N=3** i ponoviti 32-bitnu realizaciju. Opisati promjenu dobivene amplitudne karakteristike filtera (za **N=3** prema **N=5**), te iznosa ulaznog i izlaznog pojačala. Što se smanjenjem reda filtera poboljšalo, a što pokvarilo?

minimalno gušenje se ne mijenja

9. Odabrati način zadavanja filtra preko valovitosti (**Definicija filtra preko valovitosti**) što se postiže pritiskom na tipku **Definicija filtra preko reda**, čime ta tipka mijenja natpis. U ovom modu odabrati slijedeće parametre filtra: **Pojasni propust, Čebiševljev tip II, valovitost u području propuštanja 1dB, valovitost u području gušenja 45dB, granične frekvencije pojedinih područja 0.17, 0.2, 0.25 i 0.28**. Koji red filtra je dobiven? _____. Pogledati prikaz polova i nula. Da li broj polova odgovara redu filtra ? _____ (Za pojasni propust i pojasnu branu broj polova je dvostruki u odnosu na red filtra.)

Odabrati opciju **Proračun dinamike**. Koliko iznose ulazna i izlazna pojačala? Da li je projektirani filter praktički upotrebljiv? Obrazložiti odgovor.

imamo veliko prigušenje ulaznog signala, pa ga na ulazu uopće ne vidimo

6. VJEŽBA

Kaskadna realizacija IIR filtara

Kaskadna realizacija je najčešće korištena realizacija vremenski diskretnih IIR filtara jer posjeduje više dobrih svojstava. Za razliku od direktne realizacije kojom se prijenosna funkcija $H(z)$ neposredno realizirala, oblik prijenosne funkcije se kod kaskadne realizacije znatno mijenja. Umjesto razlomljene racionalne funkcije s polinomom u brojniku i nazivniku, ovdje se kreće od prijenosne funkcije izražene **umnoškom korijenih faktora**. Najprije treba naći **korijene brojnika (nule) i korijene nazivnika (polove)**, te zatim izraziti $H(z)$ preko korijenih faktora i faktora pojačanja k kako je dano izrazom (6.1):

$$H(z) = k \cdot \frac{\prod_{i=1}^{M_1} (1 - nr_i z^{-1}) \cdot \prod_{i=1}^{M_2} (1 - nc_i z^{-1}) (1 - nc_i^* z^{-1})}{\prod_{i=1}^{N_1} (1 - pr_i z^{-1}) \cdot \prod_{i=1}^{N_2} (1 - pc_i z^{-1}) (1 - pc_i^* z^{-1})} \quad (6.1)$$

Sustav može imati realne i kompleksne nule (nr_i i nc_i) i realne i kompleksne polove (pr_i i pc_i). Kompleksni polovi i nule se uvijek javljaju u konjugirano kompleksnim parovima (nc_i i nc_i^* , pc_i i pc_i^*). Red brojnika je $M = M_1 + 2M_2$, a red nazivnika je $N = N_1 + 2N_2$. Pod redom cijelog IIR filtra smatra se **red nazivnika**.

Kaskadna struktura se dobiva **kombiniranjem** parova realnih faktora i konjugirano kompleksnih parova **u faktore drugog reda** (tzv. bikvadratne sekcije). Na taj način dobiva se prijenosna funkcija u obliku **produkta polinoma drugog reda** od z^{-1} u brojniku i nazivniku

$$H(z) = k \cdot \frac{P_1(z^{-1}) \cdot P_2(z^{-1}) \cdot \dots \cdot P_{M_s}(z^{-1})}{Q_1(z^{-1}) \cdot Q_2(z^{-1}) \cdot \dots \cdot Q_{N_s}(z^{-1})} \quad (6.2)$$

gdje je $M_s = M/2$ za parni M , odnosno $M_s = (M+1)/2$, za neparni M . Analogno vrijedi za N_s . Ako je M_s manji od N_s , tada se u brojnik dodaje još $(N_s - M_s)$ trivijalnih polinoma $P_{M_s+1}(z^{-1}) = \dots = P_{N_s}(z^{-1}) = 1$. Na taj način dobivamo jednak broj polinoma u brojniku i nazivniku pa se $H(z)$ može prikazati kao produkt kvocijenata po jednog polinoma brojnika i jednog polinoma nazivnika

$$H(z) = k \cdot \prod_{n=1}^{N_s} R_n(z^{-1}) \quad , \quad \text{gdje je} \quad R_n(z^{-1}) = \frac{P_i(z^{-1})}{Q_j(z^{-1})} \quad (6.3)$$

Opći oblik jednog člana $R_n(z^{-1})$ dan je izrazom:

$$R_n(z^{-1}) = \frac{1 + b_{1n} z^{-1} + b_{2n} z^{-2}}{1 + a_{1n} z^{-1} + a_{2n} z^{-2}} \quad (6.4)$$

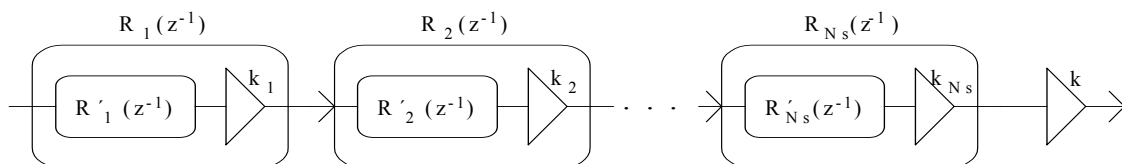
Kao što je rečeno, ovakav oblik dobiva se uparivanjem dva realna pola (ili nule) kao i uparivanjem konjugirano kompleksnih parova polova (nula). Lako se može pokazati da za oba slučaja mora vrijediti: $a_{1n} = 2r \cos \varphi < 2$ i $a_{2n} = r^2 < 1$ što proizlazi iz uvjeta stabilnosti sustava (svi polovi unutar jedinične kružnice). Koeficijenti brojnika nisu ograničeni.

Odabir polinoma $P_i(z^{-1})$ i $Q_j(z^{-1})$ u izrazu (6.3) ovisi o više faktora. Broj mogućih kombinacija uparivanja polinoma u brojniku i nazivniku je jako velik i iznosi $(N_s!)^2$. Svi takvi sustavi imaju jednak $H(z)$, ali se razlikuju po dinamikama signala u pojedinim čvorovima strukture što kod realizacije s konačnim brojem bita direktno utječe na odnos signal/šum. Egzaktna metoda kojom bi se postiglo optimalno uparivanje obzirom na odnos signal/šum je vrlo složena i u praksi se ne koristi. Umjesto toga, koriste se jednostavnije metode (od palca) koje ne daju optimalno rješenje, ali dobiveni rezultati zadovoljavaju. U ovoj vježbi koristi se jedna takva metoda koju ćemo opisati u nastavku.

Kod sekcija $R_n(z^{-1})$ kod kojih je nazivnik dobiven uparivanjem para polova bliskih jediničnoj kružnici (veliki Q faktor), u amplitudno-frekvencijskoj karakteristici postoje značajna izdignuća. To uzrokuje pojavu velikog prekoračenja dinamike u čvorovima nazivnika pa se signal na ulazu u tu sekciju mora jako atenuirati. Zbog cjelobrojne realizacije, ova atenuacija kviri odnos signal/šum. Ako se takva sekcija nalazi na početku kaskade, tada se na samom ulazu u sistem ubacuje veliki šum kojeg zatim pojačavaju sekcije s manjim Q faktorima. S druge strane, ako se takva sekcija nalazi na kraju kaskade, tada ta zadnja sekcija značajno pojačava i šum i signal koji dolazi iz prijašnjih sekcija oko frekvencije svog pola, što rezultira nejednolikom spektralnom razdiobom šuma. Iz ovog su uočljiva **два kriterija odabira redoslijeda nazivničkih polinoma** $Q_j(z^{-1})$: jedan po padajućim modulima polova (prvo najveći Q), a drugi po rastućim modulima polova (prvo najmanji Q). Program u sklopu vježbe koji realizira IIR filter kaskadnom realizacijom omogućava korištenje oba navedena kriterija.

Nakon fiksiranja redoslijeda nazivničkih polinoma, **odabir brojnika** za svaku pojedinačnu kaskadu radi se tako da dinamika na izlazu iz te kaskade bude **što bliža jedinici**, tj. da signal nije niti jako pojačan niti jako atenuiran. Tako se postiže bolji odnos signal/šum u čvorovima strukture. Postupak će biti detaljnije opisan kod analize dinamike po čvorovima strukture.

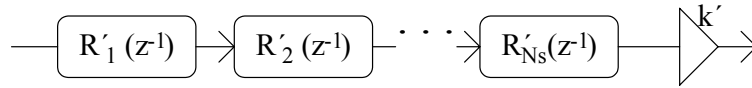
Pošto se svaki od članova $R_n(z^{-1})$ realizira direktnom formom II drugog reda (*engl. biquad*) s frakcionalnom aritmetikom, potrebno je **skalirati koeficijente brojnika i nazivnika** da poprime vrijednosti unutar ± 1 , slično postupku opisanom u prošloj vježbi. Koeficijenti nazivnika se prema potrebi dijele s 2 (ako je $a_{1n} > 1$), a koeficijenti brojnika se dijele s po modulu većim između b_{1n} i b_{2n} u slučaju da je barem jedan od ta dva po modulu veći od 1. Na taj način su svi skalirani koeficijenti a_{1n}' , a_{2n}' , b_{0n}' , b_{1n}' i b_{2n}' , člana $R_n'(z^{-1})$ unutar ± 1 , a faktor $1/a_{0n}'$ iznosi ili 1 ili 2. Da bi prijenosna funkcija ostala ista kao i prije skaliranja, potrebno je uz svaki član $R_n'(z^{-1})$ uvesti jedno pojačalo k_n (Slika 6.1).



Slika 6.1. Nadomjesna shema kaskadne realizacije.

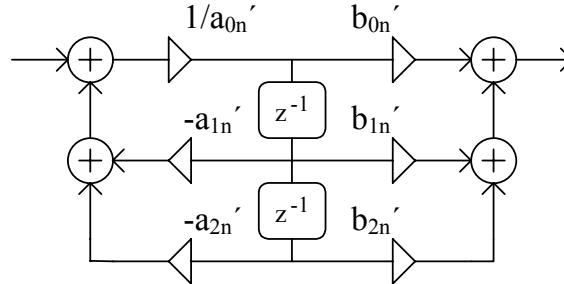
Sva pojačala k_1 , k_2 , ..., k_{N_s} i k , se mogu objediniti u jedno pojačalo k' na izlazu iz strukture (Slika 6.2) čiji je iznos

$$k' = k_1 \cdot k_2 \cdot \dots \cdot k_{N_s} \cdot k \quad (6.5)$$



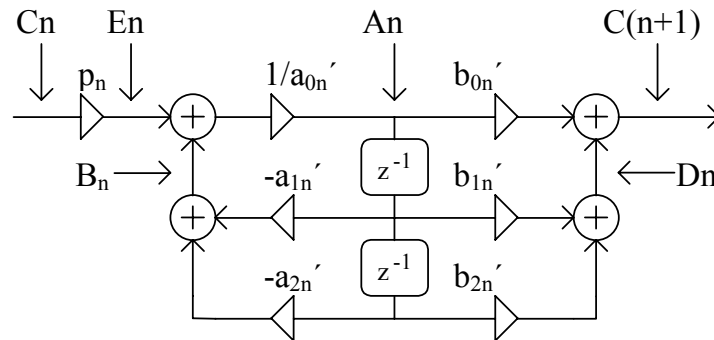
Slika 6.2. Nadomjesna shema kaskadne realizacije s objedinjenim pojačalima

Svaki član $R'_n(z^{-1})$ realizira se kao bikvadratna sekcija (Slika 6.3).



Slika 6.3. Bikvadratna sekcija

Ovim postupkom su svi koeficijenti filtra početno skalirani, no za realizaciju s konačnim brojem bita, potrebno je provesti i **provjeru dinamike u svim čvorovima** strukture. Kao što je već opisano kod realizacije direktnom metodom, na dinamiku u nazivničkoj sekciji (čvorovi A_n , B_n i E_n) utječe se uvođenjem ulaznog množila p_n (pojačalo ili atenuator) koji osigurava da dinamika u navedenim čvorovima bude što veća, ali unutar ± 1 (Slika 6.4).



Slika 6.4. Bikvadratna sekcija s označenim čvorovima u kojima se provjerava prekoračenje dinamike.

Kod proračuna dinamike za n -tu kaskadu, **potrebno je uzeti u obzir ukupnu prijenosnu funkciju prethodnih kaskada** koja se dobiva kao produkt prijenosnih funkcija tih kaskada. Kao posljedica toga, dinamika u čvoru C_n može biti i manja od 1, te p_n može biti i pojačalo (ako su A_n i B_n manji od 1), no maksimalni iznos tog pojačanja je ograničen dinamikom čvora E_n .

Kao što je već rečeno, kod odabira brojnika tražimo takav brojnik koji uz odabrano ulazno pojačalo p_n ima dinamiku čvora C_{n+1} što bližu jedinici, bilo manju, bilo veću. Za tako odabrani brojnik, radi se provjera dinamike čvorova C_{n+1} i D_n . Ako su obje dinamike manje od 1, tada se brojnik ne dira, jer je sve u redu, no ako je u bilo kojem od ta dva čvora dinamika veća od 1, tada se brojnik skalira za potrebni faktor. Kao i kod proračuna dinamike za nazivnik i ovdje je potrebno uzeti u obzir prijenosnu funkciju prethodnih kaskada. Nakon proračuna dinamike za svaku kaskadu, izlazno pojačalo k' se ažurira za faktor ulaznog pojačala p_n kao i za faktor kojim je skaliran brojnik.

Ponavljanjem gornjih postupaka za sve kaskade, dobivaju se koeficijenti sekcija i ulazna pojačala p_n te jedno izlazno pojačalo k' , tako skalirani da niti u jednom čvoru strukture ne dođe do prekoračenja dinamike. Opaska: p_n mogu biti i veći od 1, jer nam je u ovom slučaju važnije održati što veću dinamiku i bolji odnos signal/šum, nego voditi računa o složenosti realizacije pojačala p_n . Dobiveni skalirani koeficijenti se sada mogu zaokružiti na najbliže B -bitne frakcije.

Nakon zaokruženja koeficijenata potrebno je **ponoviti postupak provjere dinamike**, no kao i kod direktne realizacije se na dinamiku pojedine kaskade utječe isključivo promjenom ulaznog pojačala jer se koeficijenti brojnika više ne mogu mijenjati. Naravno, i to pojačalo mora biti B -bitna frakcija. Izlazno pojačalo k' se određuje tako da dinamika izlaza bude što bliže ± 1 (koliko to omogućava točnost B -bitne frakcije).

Kaskadna realizacija je pogodna za implementaciju vremenski diskretnih sustava iz više razloga. Prijenosna funkcija se dobiva kao kaskada sekcija istog oblika kod kojih se razlikuju samo vrijednosti koeficijenata, pa se cijeli filter realizira ponavljanjem jednog te istog programskog odsječka N_s puta, sa stanjima i koeficijentima sekcije filtra kao ulaznim argumentima. Pošto je jedna sekcija dobivena uparivanjem jednog para nula i jednog para polova (najčešće konjugirano kompleksnih), **promjena koeficijenata sekcije** uslijed zaokruženja **utječe isključivo na taj par polova i nula**, dok na sve ostale nema utjecaja. Kod direktne realizacije to nije bio slučaj jer je promjena jednog koeficijenta utjecala na položaje svih polova (nula). Ovo svojstvo kaskadne realizacije čini ju **najmanje osjetljivom** na promjene koeficijenata pa se i zbog te činjenice najčešće koristi. Još jedno dobro svojstvo kaskadne realizacije je to da se dinamike pojedinih čvorova strukture **mogu bolje regulirati** nego kod direktne realizacije. Na primjer, kod oštih filtera direktna realizacija traži značajnu atenuaciju signala na ulazu, dok se kod kaskadne realizacije prijenosna funkcija razbija u sekcije s umjerenim Q faktorima pa se signal više puta atenuira za manji iznos, što je povoljnije sa stanovišta odnosa signal/šum.

Za one koji žele znati više

Kaskadna realizacija IIR filtra na DSP procesoru

Realizacija filtra provedena je na DSP procesoru AD 2181 firme Analog Devices. U MATLAB programskom paketu razvijeni su programi za projektiranje filtera bilinearnom transformacijom, skaliranje koeficijenata radi ograničavanja dinamike svih čvorova unutar granica pogodnih za realizaciju u frakcionalnom obliku na cjelobrojnom procesoru i konačnu realizaciju u obliku asemblerskog programa za DSP procesor.

Provedene su dvije realizacije i to jedna sa 16-bitnim koeficijentima i 16-bitnim stanjima (jednostruka preciznost) i druga s 32-bitnim koeficijentima i 32-bitnim stanjima (dvostruka preciznost). U nastavku je dana jezgra programa za 16-bitnu verziju. Kao i prije nedostaje dio za komunikaciju s codec-om.

```
{Dio 1}
MODULE/RAM      SIIR;
CONST           N=5;           {BROJ KASKADA}
VAR/PM/RAM/CIRC COEFS[7*N+2];
                {KOEFIJIENTI:   N*(PN A2 A1 A0 B0 B1 B2) Aizl}
VAR/DM/RAM/CIRC DELAY[2*N]; {MEMORIJA ZA STANJA}
VAR/DM/RAM      sampl_rate; {POMOĆNA VARIJABLA}
GLOBAL          sampl_rate;
ENTRY ZAGLAVLJE; {DEFINICIJA LABELA}
ENTRY OBRADA;
INIT            COEFS : <SIIR.CFS>;
                {INICIJALIZACIJA KOEFIC. IZ DATOTEKE SIIR.CFS}
```

{Dio 2}

ZAGLAVLJE:

```

I4=^COEFS;           {POKAZIVAČ NA KOEFICIJENTE}
I3=^DELAY;           {POKAZIVAČ NA STANJA.}
L4=%COEFS;           {DEFINICIJA CIRKULARNIH BUFFERA}
L3=%DELAY;
M0=0;                 {DEFINICIJA OFSETA ZA KRETANJE}
M1=1;                 {KROZ KOEFICIJENTE I STANJA}
M2=-1;
M3=2;
M4=1;
CNTR=2*N;             {POČETNO BRISANJE STANJA}
DO CLRBUF UNTIL CE;
CLRBUF: DM(I3,M1)=0;
RTS;

```

{Dio 3}

```

OBRADA: MR=0;         {ULAZ JE U REGISTRU MX1}
MR1=MX1;
MY0=PM(I4,M4);
CNTR=N;               {PETLJA ZA SVE KASKADE}
DO BIQUAD UNTIL CE;
MR=MR-MR1*MY0(SS),MY1=PM(I4,M4),MX1=DM(I3,M1);
                                     {KOEf. A2, STANJE Z2}
MR=MR-MX1*MY1(SS),MY0=PM(I4,M4),MX0=DM(I3,M0);
                                     {KOEf. A1, STANJE Z1}
MR=MR-MX0*MY0(SS),MY1=PM(I4,M4); {KOEf. A0}
MR=MR-MR1*MY1(SS),MY0=PM(I4,M4); {KOEf. B0}
MR=MR(RND),MY1=PM(I4,M4); {KOEf. B1}
IF MV SAT MR;          {ZASIĆENJE?}
DM(I3,M2)=MR1,MR=MR1*MY0(SS); {STANJE MR->Z1}
MR=MR+MX0*MY1(SS),MY1=PM(I4,M4); {KOEf. B2}
MR=MR+MX1*MY1(RND),DM(I3,M3)=MX0; {Z1->Z2}
MY0=PM(I4,M4);          {PN+1}
BIQUAD: IF MV SAT MR;   {ZASIĆENJE?}
MR=MR1*MY0(RND),AY0=PM(I4,M4); {IZLAZNO POJAČALO}
SE=EXP MR1(HI);
AX0=SE;
AR=AX0+AY0;
IF GT JUMP OVERFLOW;
SE=AY0;
SR=ASHIFT MR1 (HI);
MR1=SR1;
MR=MR(RND);
JUMP OUT_PORT;
OVERFLOW: AR=PASS MR1;
IF LT JUMP NEGATIV;
MR1=0X7FFF;
JUMP OUT_PORT;
NEGATIV: MR1=0X8000;
OUT_PORT: RTS;          {IZLAZ JE U REGISTRU MR1}
.ENDMOD;

```

Za svaki uzorak ulaznog signala DSP procesor mora izvršiti P operacija danih slijedećim izrazom $P = 42 + 11 \cdot N$ gdje je N_s broj kaskada. Prije je objašnjeno da za regularan rad moramo zadovoljiti uvjet: $f_{proc} > P \cdot f_0$. Kombiniranjem gornjih dviju relacija dobiva se izraz za maksimalni broj kaskada filtera koji se može realizirati ovim programom:

$$N < \frac{1}{11} \cdot \left(\frac{f_{proc}}{f_0} - 42 \right), \quad (6.6)$$

što uz frekvenciju rada procesora $f_{proc} = 33.33 MHz$ i frekvencije otipkavanja $f_0 = 48 kHz$ iznosi $N = 59$. To odgovara filtru 118 reda.

Zadatak za pripremu

Za vremenski diskretni sustav dan s prijenosnom funkcijom $H(z)$ nacrtati direktnu II i kaskadnu realizaciju te upisati iznose koeficijenata zapisane u 4 bita (3 bita + predznak). Koristiti metodu zaokruživanja na najbliži cijeli broj. Po potrebi skalirati koeficijente radi postizanja sto bolje dinamike. Da li je filter stabilan ?

$$H(z) = \frac{(0,53 + 1,54 \cdot z^{-1}) \cdot (1,89 + 2z^{-1} + z^{-2})}{(0,35 + 0,045 \cdot z^{-1} - 0,7 \cdot z^{-2}) \cdot (1 - 0,23 \cdot z^{-1})}$$

Rješenje zadatka:

Praktična vježba

KASKADNA REALIZACIJA IIR FILTRA MATLAB APLIKACIJOM

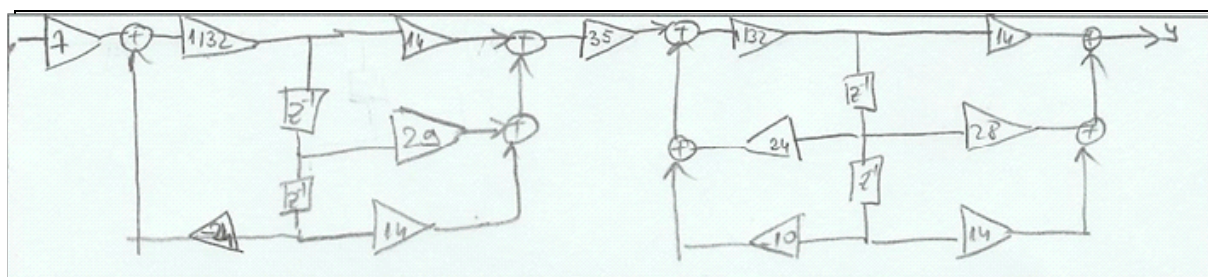
Samo projektiranje IIR filtera metodom bilinearne transformacije provoditi će se istim postupcima kao što se to radilo i kod direktne realizacije filtera. Nakon toga se, na temelju dobivenih koeficijenata, provodi rastavljanje prijenosne funkcije na bikvadratne sekcije, te određivanje redoslijeda kaskada i kvantizacija koeficijenata kako je objašnjeno u teoretskom dijelu vježbe.

Kod većine projektiranih filtera provodit će se usporedba dobivenih (kaskadnih) sa filterima ostvarenim direktnom realizacijom. U vezi s tim projektirane filtre je moguće usporediti s rezultatima iz prethodne vježbe jer se projektiraju isti tipovi filtera, ili ponavljanjem postupaka odabirom opcije Direktna realizacija (s obzirom da se korištenom Matlab aplikacijom mogu ostvariti i jedni i drugi filteri).

Nakon zadavanja parametara filtera uvijek **vizualno provjeriti** da li dobivena frekvencijska karakteristika odgovara željenoj. Ako nakon unosa svih parametara filter ne izgleda kako bi trebao (npr. gušenje je ostalo isto kao u prethodnom zadatku), treba ponovno unijeti vrijednost barem jedne granične (rubne) frekvencije.

Kod prikaza kvantizirane karakteristike provjeriti **da li je iscrtana kvantizirana amplitudna karakteristika** (ljubičastom bojom) jer se program može ponašati kao da je sve u redu, a u stvari se karakteristika ne može realizirati sa zadanim brojem bita pa nije nacrtana.

1. Unutar MATLAB okruženja pokrenuti program vj6. Odabrati slijedeći skup parametara filtera za kaskadnu realizaciju, **padajući Q faktori: niski propust, Čebiševljev tip I, 4. red, valovitost u području propuštanja 1dB, granična frekvencija 0.25**. Odabrati zaokružnje koeficijenata na **6 bita**. Odabrati za prikaz Proračun dinamike. Proučiti postupak (korak po korak) kvantizacije koeficijenata filtera opisan u teoretskom dijelu vježbe.



U gornji prostor nacrtati realizaciju promatranog filtera s upisanim cjelobrojnim vrijednostima koeficijenata i dinamikama čvorova A1 do E1 i A2 do E2 (promotriti sliku, Slika 6.4). Promotriti dobivenu kvantiziranu karakteristiku. -faza se ne mijenja, a amplituda je povećana na višim frekv. nego prije kvantizacije

Odabrati **8 bita** i ponovo odabrati prikaz **Proračun dinamike**. Usporedite dobivene karakteristike i dinamike sa prethodnim slučajem (sa 6 bita). U kojem slučaju su dinamike u čvorovima bliže ± 1 (a to je u stvari cilj ovakvog postupka proračuna) ?

Pogledati izgled kvantizirane karakteristike uz poredak nazivnika po **rastućim Q faktorima** i zaokružnje na **8 bita**. Zatim ponoviti postupak proračuna dinamike uz ovaj

poredak nazivnika i zaokruženje na **6 bita** te usporediti dinamike u svim čvorovima strukture u odnosu na slučaj s padajućim Q faktorima. Prokomentirati.

Pogledati izgled kvantiziranih karakteristika uz direktnu realizaciju i zaokruženje na **8 i 6 bita** i usporediti ih sa onima dobivenim kaskadnom realizacijom. Koja realizacija je bolja ?

2. Vratiti **definiciju preko reda filtra** klikom na odgovarajuću tipku u aplikaciji. U nastavku **odabrati** slijedeće parametre filtra: **Niski propust, Čebiševljev tip II, gušenje u stop bandu=80dB, 6. red, granična frekvencija 0.1, rastući Q faktori**. Koliki je minimalni broj bita s kojim se može realizirati ovaj filter? ____ Koliki je ako se odabere realizacija uz padajuće Q faktore? ____ Koliki je broj bita potreban za direktnu realizaciju?

3. Projektirati filter sa slijedećim karakteristikama: **Pojasni propust, Eliptički, 5. red, valovitost u području propuštanja 2dB, valovitost u području gušenja 50dB, granične frekvencije 0.3 i 0.4.** Odabrati raspored po **rastućim Q faktorima**. Da li je filter izvediv u 16 bita? Usporedi slučaj sa direktnom realizacijom. Kolike su maksimalne dinamike za slučaj kaskadne i direktne realizacije?

KASKADNA REALIZACIJA IIR FILTRA NA DSP PROCESORU

5. Uz specifikaciju filtra iz prethodne točke odabrati frekvenciju otipkavanja **8kHz**. Pritiskom na tipku **Start DSP** (16 bita) prenosi se 16-bitna verzija projektiranog filtra na DSP maketu. Mijenjanjem frekvencije provjeriti rad filtra. Prekinuti program na DSP procesoru pritiskom na tipku 'interrupt' za proizvoljnu frekvenciju ulaznog signala unutar radnog područja filtra. Pritiskom na tipku **obrada 16** vrši se usporedba filtriranja sa i bez kvantizacije stanja filtra (DSP maketa odnosno MATLAB simulacija). Koliko iznosi greška zbog kvantizacije stanja (izražena je u LSB za 16 bita) ?
 $LSB=1/2^{15}$; na $f=9.43\text{ kHz}$ je od -150 do 200

Ponoviti postupak za 32 bitnu realizaciju. Koliko sada iznosi greška zbog kvantizacije? Za usporedbu sa direktnom realizacijom pogledati 7. zadatak iz prošle vježbe ili ponoviti postupak uz direktnu realizaciju.

6. Projektirati filter sa slijedećim parametrima: **pojasna brana, Čebiševljev tip II, 8. red, valovitost u području gušenja 120dB, granične frekvencije 0.2 i 0.23**. Ovo je primjer jedne pojasne brane s vrlo ostrim zahtjevima. Uz odabir frekvencije otipkavanja **fs=8KHz** i raspored sekcija po **rastućim Q faktorima** provesti 32-bitnu realizaciju filtra za DSP procesor (**Start DSP (32 bita)**). Nakon pokretanja programa na DSP procesoru, na **osciloskopu očitati** slijedeće vrijednosti (ili vršnu ili peak to peak, ali za sve jednako):

Ulazni napon (ne ovisi o frekvenciji) = _____

Izlazni napon filtra pri frekvenciji od 1KHz (područje propuštanja) = _____

Izlazni napon (samo signal, bez šuma) filtra pri frekvenciji 1.7KHz (područje gušenja) = _____. Obratiti pažnju na frekvenciju mjerenog signala i upisati ju _____.

Na osnovu dobivenih rezultata prema relaciji (6.7) odrediti slijedeće vrijednosti:

$$20 \cdot \log_{10} \left(\frac{U_{izl}}{U_{ul}} \right) [\text{dB}] \quad (6.7)$$

Iznos u području propuštanja = $\frac{9.79}{}$ [dB]

Iznos u području gušenja = $\frac{56.9}{}$ [dB]

Gušenje signala u **području propuštanja** ne ovisi o filteru već o ostalim dijelovima sustava za obradu signala (codec, analogni filteri, DSP maketa). Ono je gotovo neovisno o frekvenciji u nama interesantnom frekvencijskom području, a iznos se može odrediti mjerenjem u području propuštanja.

Razlika gušenja u području gušenja i području propuštanja određuje stvarni iznos gušenja samog filtera koje iznosi = $\frac{47 \text{ dB}}{}$

Prilikom projektiranja filtera traženo je od filtera da u području gušenja guši 120 dB. Mjerenjem je ustanovljeno da je **dobiveni rezultat puno manji**. Zašto? Odgovor na ovo pitanje potražiti u slijedećoj točki.

7. Pri ulaznoj frekvenciji od 1.7KHz (područje gušenja) pritiskom na tipku 'interrupt' prekinuti izvršavanje programa. Pritiskom na tipku **Obrada 32** učitavaju se snimljeni podaci na DSP procesoru radi usporedbe sa MATLAB simulacijom. Razlika tih dvaju odziva je posljedica kvantizacije stanja i ona iznosi : _____ bita 32-bitne riječi.

U glavnom prozoru programa vj6, odabrati tipku **KRAJ** (zatvaranje prozora). U nastavku ćemo pogledati spektar ulaznog i izlaznog signala unosom slijedećeg niza naredbi iz komandne linije MATLABA:

```
clear          % Brisanje varijabli iz memorije

load ul          % Učitavanje ulaznih podataka i spremanje u varijablu ulaz
ulaz=dat';

load iz          % Učitavanje izlaznih podataka i spremanje u varijablu izlaz
izlaz=dat';

w=blackman(1024); % Kreiranje 1024 uzorka Blackman-ovog vrem. otvora

% množenje dijela ulaznog i izlaznog signala s otvorom radi ograničavanja signala, ali
% i radi smanjenja efekata koji se javljaju uslijed korištenja kratkotrajne FFT
su=w.*ulaz(1025:2:3072);
si=w.*izlaz(1025:2048);

% određivanje FFT-a
fsu=fft(su);
fsi=fft(si);

% Određivanje amplitudnog spektra i njegovo normiranje dijeljenjem s N/2=512
% (da signal vršne amplitude 1odgovara 0 dB u prikazu u dB) i izračun u dB
afsu=20*log10(abs(fsu)/512);
afsi=20*log10(abs(fsi)/512);
```


% Normiranje signala. Time se izbacuje utjecaj ostatka digitalnog sistema.

```
max_ul=max(afsu);  
sp_ulaz=afsu-max_ul;  
sp_izlaz=afsi-max_ul;
```

% Prikaz ulaznog i izlaznog spektra jedan preko drugog (samo prva polovica spektra).
% Os x prikaza, om, je izražena u Hz, od 0 do 4 kHz (pola frekv. otipkavanja).

```
figure  
om=[0:511]/512*4000;  
plot(om,[sp_ulaz(1:512) sp_izlaz(1:512)]); grid
```

% Prikaz spektara jednog ispod drugog radi lakšeg promatranja.

```
figure  
subplot(2,1,1); plot(om,sp_ulaz(1:512)); grid;  
subplot(2,1,2); plot(om,sp_izlaz(1:512)); grid;
```

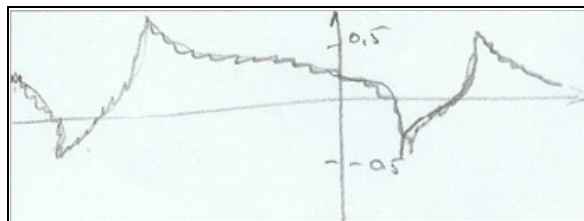
Opisati dobivene slike. Koliko je gušenje osnovne komponente ulaznog signala? Zašto postoji drugi pik u ulaznoj i izlaznoj prijenosnoj karakteristici? Koja je to komponenta i zašto postoji? Koliko se on nalazi ispod osnovne harmonijske komponente ulaznog signala?

djelomično propuštanje na -50dB

8. Na DSP maketi odabrati kao pobudu **pravokutni napon frekvencije 100 Hz**- Na osciloskopu prikazati jednu do dvije periode ulaznog signala. Nakon resetiranja DSP procesora (prije pokretanja korisničkog programa) signal na izlazu trebao bi biti jednak ulaznom (Ch1=1V/ds, Ch2=0,5V/ds). Razlika signala je posljedica ostatka sklopova koji se uz DSP procesor nalaze u lancu za digitalnu obradu signala. Tako je npr. eksponencijalni pad izlaznog napona posljedica **AC-veze** koja postoji na ulazu i izlazu sustava za digitalnu obradu signala, a kašnjenje signala nastaje radi konačnog trajanja obrade signala.

Projektirati filtar sa slijedećim karakteristikama: **Niski propust, Eliptički tip aproksimacije, 12. red, valovitost u području propuštanja 3dB, valovitost u području gušenja 80dB, granična frekvencija 0.05**. Uz frekvenciju otipkavanja iznosa **48kHz** i **rastući izbor poretka polova sekcija drugog reda**, provesti **32 bitnu** realizaciju filtra i istu pokrenuti na DSP-u. U prazni uokvireni prostor ucrtati sliku s osciloskopa (izlazni signal).

Usporediti rezultate s MATLAB simulacijom.



9. Isprobati primjer iz prethodne točke u **16-bitnoj realizaciji** i to uz odabir polova bikvadratnih sekcija najprije po rastućem, a zatim po padajućem poretku.

Dobije se da je izlazni signal dobiven na maketi **konstantan i jednak 1** (odnosno 0 kada se gleda na osciloskopu, zbog AC-veze). Naime, u ovom slučaju se uz kvantizaciju koeficijenata od 16 bita i kvantizaciju stanja dogodi jedna pojava koja se naziva **limit cycle**. Posljedica je **nelinearnosti kvantizacije u povratnoj vezi** nazivničkih sekcija ili **prekoračenja dinamike** u aritmetici dvostrukog komplementa u nekom od čvorova. Do ovog prekoračenja bi u ovom slučaju usprkos dobrom proračunu moglo doći zato što je ulazni napon pravokutan, a ne sinus jedne komponente (za što je proračun napravljen).

7. VJEŽBA

Efekti kvantizacije kod IIR filtara

Sagledavanje svih efekata kod projektiranja i realizacije IIR filtara nije potpuno ako se detaljno ne prouče i efekti koji nastaju uslijed ograničene točnosti procesora ili sklopovlja kojima se takav filter realizira. U prošlim vježbama je pored različitih realizacija analiziran i efekt kvantizacije koeficijenata filtra. Međutim, treba voditi računa o tome da i ulazni i izlazni signal kao i stanja filtra također moraju biti kvantiziranih iznosa. U općenitom slučaju, broj bita za prikaz koeficijenata i broj bita za prikaz signala i stanja ne mora biti isti.

Efekti kvantizacije stanja došli su do izražaja u prethodnim vježbama pri realizaciji filtara na DSP procesoru. Ovisno o realizaciji, kvantizacija se provodila na 16 (jednostruka preciznost) ili 32 bita (dvostruka preciznost), a samim tim bila je definirana i točnost filtriranja signala. Pogreška je promatrana u usporedbi s rezultatima filtriranja u MATLAB-u koji radi s puno većom točnošću bez kvantizacije stanja, ali koristi iste kvantizirane koeficijente.

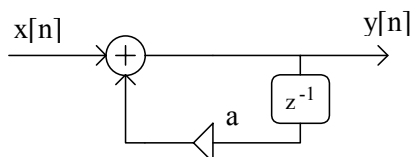
Greška nastala uslijed kvantizacije ulaznog signala se relativno lako može analizirati, jer grešku kvantizacije možemo promatrati kao signal pogreške koji se pribraja ulazu i koji se zatim filtrira jednako kao i sam signal. Nešto je složenija analiza grešaka nastalih kvantizacijom stanja unutar filtra, a pojave vezane uz to biti će opisane u nastavku.

Zero input limit cycle

Ako na stabilne vremenski diskretne IIR sisteme realizirane aritmetikom beskonačne preciznosti djelujemo pobudom koja u određenom trenutku padne u nulu i ostane u nuli, tada se odziv takvih sistema asimptotski približava nuli. Za istu takvu pobudu i isti sistem, ali realiziran aritmetikom konačne preciznosti, odziv može nastaviti beskonačno oscilirati iako je pobuda ostala u nuli. Takva pojava se naziva ‘**zero input limit cycle**’, a posljedica je ili nelinearnosti kvantizacije u povratnoj vezi ili prekoračenja dinamike kod zbrajanja u two's complement aritmetici.

- **Limit cycle uslijed zaokruživanja ili odsijecanja pri kvantizaciji**

Ova pojava je vrlo složena i teško se analizira općenito, pa je možemo promotriti na jednostavnom primjeru sistema prvog reda, opisanog jednačom diferencijala (7.1) i blok shemom (Slika 7.1).



Slika 7.1. Blok shema IIR strukture 1. Reda.

$$y[n] = a \cdot y[n-1] + x[n], \quad |a| < 1 \quad (7.1)$$

Pretpostavimo da dužina registra za pohranu koeficijenta a , ulaza $x[n]$ i varijable stanja $y[n-1]$, iznosi 4 bita (3 bita + 1 bit predznak). Zbog registara konačne dužine, 7-bitni produkt $a \cdot y[n-1]$ se mora ili zaokružiti ili odsjeći na 4 bita prije sumacije s $x[n]$ i pohrane u

registar stanja. Na primjer, pretpostavimo sistem s $a = 0.5 = 0 \wedge 100$, početnim stanjem jednakim nuli i pobudom $x[n] = 7/8 \cdot \delta[n] = (0 \wedge 111) \cdot \delta[n]$. Da bismo pronašli $y[1]$, množimo $y[0]$ s a i dobije se 7-bitni broj $0 \wedge 111 \mid 100$ koji treba zaokružiti na 4-bitnu vrijednost. Ovaj broj, $7/16$, je točno na pola između dva 4 bitna kvantizacijska nivoa, $3/8$ i $4/8$. Ako u takvim slučajevima zaokružujemo na prvi veći broj, tada se zaokruženjem dobiva $0 \wedge 100 = 0.5$. Pošto je pobuda od $n=1$ na dalje jednaka nuli, vrijednost kvantiziranog produkta je identična izlaznoj vrijednosti $y[n]$. Ponavljanjem izvođenja jednadžbe diferencija, dobiva se $y[2] = 0 \wedge 010 = 0.25$ i $y[3] = 0 \wedge 001 = 0.125$. U oba ova slučaja donji dio produkta je jednak nuli pa zaokruženje nije potrebno. No, kod traženja $y[4]$ zaokruženjem $a \cdot y[3] = 0 \wedge 000 \mid 100$ dobiva se $0 \wedge 001 = 0.125$, što se dobiva i za sve daljnje uzorke izlaza za $n > 3$. Vidljivo je da iako je ulaz pao u nulu, izlaz ostaje zauvijek na vrijednosti 0.125. Kada bi koeficijent a imao vrijednost $a = -0.5$, ponavljanjem gornjeg postupka može se opaziti da se na izlazu javljaju oscilacije, tj. uz nulu na ulazu, izlaz naizmjenice poprima vrijednost 0.125 i -0.125. Oba ova slučaja se tretiraju kao pojava zero input limit cycle.

U prethodnom primjeru je razmatran slučaj kvantizacije s funkcijom zaokruženja na najbliži cijeli broj (rounding). Slična ponašanja javljaju se i kod kvantizacije odsijecanjem, odnosno odsijecanjem simetričnim prema nuli.

- **Limit cycle uslijed prekoračenja dinamike**

Promatrajmo sistem drugog reda opisan jednadžbom diferencija

$$y[n] = a_1 \cdot y[n-1] + a_2 \cdot y[n-2] + x[n] \quad (7.2)$$

Kvantizacija je posljedica zaokruženja na 4 bita. Do prekoračenja dinamike može doći kod zbrajanja two's complement vrijednosti zaokruženih produkata. Npr. neka je $a_1 = 0.75 = 0 \wedge 110$ i $a_2 = -0.75 = 1 \wedge 010$, ulazni signal $x[n]$ neka je nula za $n \geq 0$ i početna stanja $y[-1] = 0.75 = 0 \wedge 110$ i $y[-2] = -0.75 = 1 \wedge 010$. Izlaz za $n = 0$ dobiva se kao

$$y[0] = 0 \wedge 110 \times 0 \wedge 110 + 1 \wedge 010 \times 1 \wedge 010$$

tj. ako izračunamo produkte:

$$y[0] = 0 \wedge 100 \mid 100 + 0 \wedge 100 \mid 100$$

što nakon zaokruženja daje

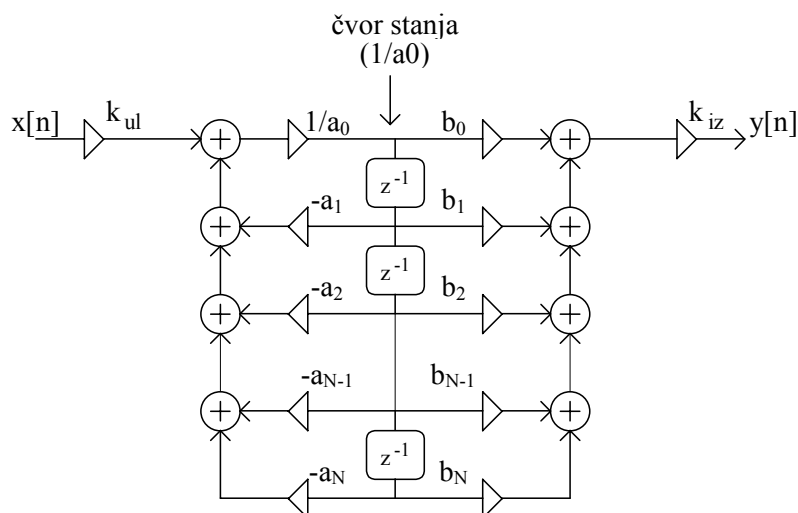
$$y[0] = 0 \wedge 101 + 0 \wedge 101 = 1 \wedge 010 = -0.75$$

Vidljivo je da two's complement sumacijom dva velika pozitivna broja dobivamo negativni broj, što predstavlja značajnu pogrešku. Ponavljanjem jednadžbe diferencije dobivamo $y[1] = 0.75$ koji je dobiven sumacijom dva velika negativna broja. Očito je da će izlaz nastaviti oscilirati između 0.75 i -0.75 tako dugo dok god je na ulazu nula.

Ovakva pojava je vrlo neugodna jer uzrokuje oscilacije gotovo maksimalnom amplitudom iako je na ulazu nula. Ona se može spriječiti tako da se umjesto klasične two's complement aritmetike koristi aritmetika sa zasićenjem (saturation). Njenom primjenom, suma dva velika pozitivna broja koja bi u normalnom slučaju rezultirala negativnim brojem, zamjenjuje se s maksimalnim pozitivnim brojem. Analogno, suma dva velika negativna broja, zamjenjuje se s maksimalnim negativnim brojem.

Direktna realizacija

Za ilustraciju programske realizacije direktne filtarske strukture dan je odsječak funkcije napisane za programski paket MATLAB, koja u potpunosti simulira rad jednog cjelobrojnog procesora. U odsječku se pozivaju funkcije za zbrajanje i množenje 'intadd.m' i 'intmult.m' koje simuliraju rad s brojevima fiksne preciznosti, a blok shema samog filtra prikazana je slikom (Slika 7.2).



Slika 7.2. Blok shema IIR filtra za direktnu realizaciju.

Funkcija 'INTADD.M'

```
function sum=intadd(s1,s2) % funkcija za zbrajanje sa zasićenjem
s=s1+s2; % zbroji s1 i s2
if (s<mns), % ako je rezultat manji od najmanjeg
    s=mns; % negativnog broja
    l=1; % zamijeni ga s max negativnim brojem
end; % označi da se desilo zasićenje
if (s>mxs), % ako je rezultat veći od najvećeg
    s=mxs; % pozitivnog broja
    l=1; % zamijeni ga s max pozitivnim brojem
end; % označi da se desilo zasićenje
sum=s; % rezultat je jednak s
```

Funkcija 'INTMULT.M'

```
function prd=intmult(s,c,round_type) % funkcija za množenje i kvantizaciju
% produkta
if (c==mxc), % ako je koeficijent jednak 1,
    p=s; % produkt je jednak signalu s
else, if (c==mnc), % inače, ako je koeficijent jednak -1,
    p=-s; % produkt iznosi -s
else, % za sve ostale slučajeve,
    p=s*c; % nađi produkt signala i koeficijenta
    if (round_type==0), % ako je round_type = 0
        p=floor(p/mxc); % odsijeci produkt na broj bita signala
    else, if (round_type==1), % inače, ako je round_type = 1
        p=fix(p/mxc); % simetrično odsijeci oko nule
    else, % inače,
        p=round(p/mxc); % zaokruži na najbližu frakciju
    end;
end;
end;
if (p==mxs+1), % ako je kao produkt dobivena vrijednost 1
    p=mxs; % zamijeni je s max pozitivnom vrijednošću
    l=1; % označi da se dogodilo zasićenje
end;
prd=p; % rezultat množenja je u prd
```

Programski odsječak za direktnu realizaciju

```

ce=intmult(cc,scal(1),rt); % pomnoži signal ulaz. pojačalom
sum=0; % obriši sumu za nazivnik
for cv=N:-1:1, % od N-tog do prvog stanja
    pr=intmult(st(cv),-naz(cv+1),rt); % pomnoži stanje i koeficijent ai
    sum=intadd(sum,pr); % akumuliraj u sumu
end;
sum=intadd(sum,ce); % dodaj skalirani signal
mn=mx/naz(1); % vrijednost 1/a0
sh=round(log10(mn)/log10(2)); % nađi broj shift-anja
ca=intshift(sum,sh,rt); % prema potrebi shiftaj --> time
% se dobije signal u čvoru 1/a0
sum=0; % obriši sumu za brojnik
for cv=N:-1:1, % od N-tog do prvog stanja
    pr=intmult(st(cv),br(cv+1),rt); % pomnoži stanje i koeficijent bi
    sum=intadd(sum,pr); % akumuliraj u sumu
end;
pr=intmult(ca,br(1),rt); % čvor (1/a0) * b(0)
sum=intadd(sum,pr); % dodaj i njega u sumu
for cv=N:-1:2, % napravi pomak stanja
    st(cv)=st(cv-1); % u i-to ubaci (i-1)-vo stanje
end;
st(1)=ca; % prvo stanje poprima vrijednost
% čvora 1/a0
% množenje signala s izlaznim pojačalom kiz
iz=0;
um=scal(2); % vrijednost izlaznog pojačala
bs=floor(um/mx); % cijeli dio pojačala
um=rem(um,mx); % decimalni dio pojačala
if (bs~=0), % ako postoji cijeli dio
    for bss=1:bs, % napravi uzastopnu sumaciju
        iz=intadd(iz,sum);
    end;
end;
izf=intmult(sum,um,rt); % produkt s decimalnim dijelom
iz=intadd(izf,iz); % zbroji cijeli i decimalni dio

```

Ulazne varijable u gore prikazani programski odsječak su:

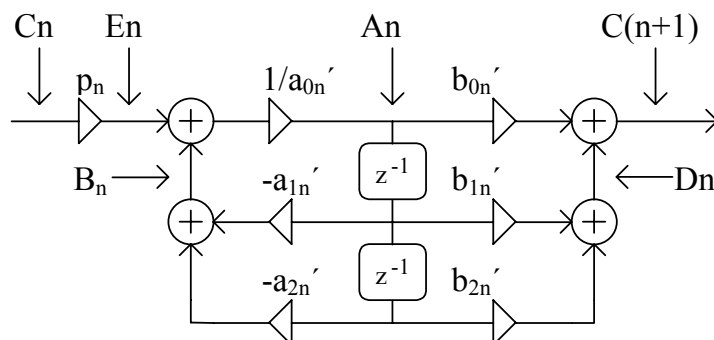
- **N** je red IIR filtra;
- **cc** je vrijednost uzorka ulaznog signala;
- **br** je redak s **(N+1)**-im elementom koji predstavljaju cjelobrojne koeficijente brojnika prijenosne funkcije **H(z)**;
- **naz** je također redak iste dimenzije, a sadrži koeficijente nazivnika;
- **scal** je stupac ili redak s dva elementa koji predstavljaju vrijednosti ulaznog i izlaznog pojačala;
- **mx** je $2^{\text{brbit}-1}$ i predstavlja frakcionalnu vrijednost 1, gdje je **brbit** broj bita za frakcionalne koeficijente;
- **st** je redak s **N** elemenata i sadrži stanja filtra;
- **rt** je varijabla koja sadrži način kvantizacije produkata i prosljeđuje se funkciji **intmult**;

Na kraju izvođenja ovog odsječka varijabla **iz** predstavlja izlaz izlaznog pojačala filtra, a to je ujedno i jedan izlazni uzorak cijelog filtra. Uzastopnim ponavljanjem ovog odsječka za sve uzorke ulaznog signala dobiva se ukupni odziv filtra.

Kaskadna realizacija

Za ilustraciju programske realizacije kaskadne filterske strukture dan je odsječak funkcije također napisane za MATLAB, koja u potpunosti simulira rad jednog cjelobrojnog

procesora. U odsječku se pozivaju iste funkcije za zbrajanje i množenje kao kod primjera direktne realizacije. Blok shema jedne kaskade filtra s ucrtanim čvorovima prikazana je na slikom (Slika 7.3)



Slika 7.3. Blok shema jedne kaskade IIR filtra.

Programski odsječak za kaskadnu realizaciju

```
for cas=1:brcas+1, % za sve kaskade i zadnje pojačalo
    cec=0;
    um=scal(cas); % vrijednost ulaznog pojačala
    bs=floor(um/mxc); % cijeli dio multiply-era
    um=rem(um,mxc); % decimalni dio
    if (bs~=0), % ako postoji cijeli dio
        for bss=1:bs, % napravi uzastopnu sumaciju
            cec=intadd(cec,cc);
        end;
    end;
    cef=intmult(cc,um,rt); % nađi produkt s decim. dijelom
    ce=intadd(cef,cec); % zbroji cijeli i decimalni dio
    if (cas<(brcas+1)), % ako to nije pojačalo na izlazu
        % iz filtra, tada radi i biquad
        cpb2=intmult(st(cas,2),-naz(cas,3),rt); % x(k-2) * -a(2)
        cpb1=intmult(st(cas,1),-naz(cas,2),rt); % x(k-1) * -a(1)
        cb=intadd(cpb1,cpb2); % čvor b
        ca=intadd(cb,ce); % čvor a
        if (naz(cas,1)==mxc/2), % ako je a(0)=0.5
            ca=intadd(ca,ca); % pomnoži čvor a s dva
        end;
        cpd2=intmult(st(cas,2),br(cas,3),rt); % x(k-2) * b(2)
        cpd1=intmult(st(cas,1),br(cas,2),rt); % x(k-1) * b(1)
        cd=intadd(cpd1,cpd2); % čvor d
        cpc=intmult(ca,br(cas,1),rt); % čvor a * b(0)
        cc=intadd(cpc,cd); % čvor c
        st(cas,2)=st(cas,1); % x(k-2) = x(k-1)
        st(cas,1)=ca; % x(k-1) = čvor a
    end;
end;
```

Ulazne varijable u ovaj programski odsječak su:

- **brcas** je broj kaskada IIR filtra;
- **cc** je inicijalno vrijednost uzorka ulaznog signala;
- **br** je matrica s **brcas** redaka i 3 stupca, a predstavlja cjelobrojne koeficijente brojnika pojedinih kaskada;
- **naz** je također matrica **brcas** × 3, a sadrži koeficijente nazivnika;
- **scal** je stupac s (**brcas**+1) redaka s vrijednostima ulaznih pojačala u pojedinu kaskadu, a zadnji element stupca sadrži vrijednost izlaznog pojačala;
- **mxc** je $2^{\text{brbit}-1}$ i predstavlja frakcionalnu vrijednost 1, gdje je **brbit** broj bita za frakcionalne koeficijente;
- **st** je matrica s **brcas** redaka i dva stupca i sadrži stanja po pojedinim kaskadama;

- **rt** je varijabla koja sadrži način kvantizacije produkta i proslijeđuje se funkciji **intmult**.

Praktična vježba

UTJECAJ KVANTIZACIJE: SIMULACIJA U MATLABU

1. Unutar MATLAB okruženja pokrenuti program **vj7**. Odabrati slijedeći skup parametara filtra: **niski propust, Chebyshev tip II, 4. red, granična frekvencija 0.2, valovitost u području gušenja 45dB**. Odabrati **8 bita** za kvantizaciju koeficijenata.

Odabrati **12 bita** za kvantizaciju signala, i **impulsnu pobudu**. Za sve tipove kvantizacije ponoviti postupak uz isti broj bita i istu pobudu. Pogledati **Amplitudno fazno frekventnu karakteristiku, Simulaciju u MATLABU i Grešku kvantizacije stanja**. Komentirati rezultate, uz koji tip kvantizacije je greška najmanja i koliko iznosi izražena u broju bita ?

A.-F. karakteristika ne ovisi o vrsti kvantizacije; najmanja greška kvantizacije je pri zaokruživanju na najmanji cijeli broj za sve vrste realizacija

Ponoviti za sva tri tipa kvantizacije, ali uz kvantizaciju signala na **8 bita**. Koje sve pojave možete uočiti ?

A-F karakteristike su iste; greška kvantizacije je veća za direktnu realizaciju za sve vrste kvantizacije

Odabrati kao pobudu **step** i sva tri načina kvantizacije s **12 bita**. Kolika je maksimalna vrijednost odziva (nadvišenja) uz realna stanja, a kolika je uz cjelobrojna stanja ? Zbog čega se javlja prekoračenje dinamike ?

max za direktnu - 1.1(realna) ; 1(cjelobrojna)
max za kaskadnu - 1.15(realna) ; 1 (cjelobrojna)

Odabrati kao pobudu **sinus** najprije frekvencije koja **ulazi u pass band**, a zatim i frekvencije **u stop bandu** te uz sva tri načina kvantizacije s **8 i 12 bita** analizirati dobivene rezultate. Kakav je odnos greške uz frekvenciju u pass band-u i uz frekvenciju u stop band-u ? za 8 bita- premali broj bita za direktnu realizaciju

na 12 bita - nema velike razlike između cjelobrojne i realne

2. Ponoviti **kompletni postupak iz točke 1. za kaskadnu realizaciju** filtra s **rastućim Q** faktorom. Analizirati razlike između jedne i druge realizacije, te odnose maksimalnih grešaka.

- najlošije je odsjecanje prema dolje, tj. tada je najveća greška kvantizacije

- najmanja greška kvantizacije je za zaokruživanje na najbliži cijeli broj

-povećanjem broja bita za sve kvantizacije greške se smanjuju

3. Ponoviti gornje postupke za **Butterworth-ov filter 6 reda i tip filtra po izboru**. Komentirati razlike u odnosu na prethodni filter.

- Chebyshev tip II za niski propust ima veće greške kvantizacije za sve vrste kvantiziranja

- za Butterworth-ov filter se primjećuju male razlike A-F karakteristike

UTJECAJ KVANTIZACIJE: REALIZACIJA NA DSP PROCESORU

4. U svim primjerima ove i prethodnih vježbi provedena je konstrukcija filtra koja bi trebala neovisno o tipu realizacije, kvantizaciji koeficijenata, odnosno kvantizacije stanja, osigurati ispravan rad filtra u cijelom frekvencijskom području. Pri tome su korištene metode koje pretpostavljaju sinusni signal, maksimalne amplitude, na ulazu u sistem i promjenom njegove frekvencije kroz cijelo područje traže se maksimalne vrijednosti pojedinih interesantnih točaka digitalnog sistema, i na osnovu njih vrše odabir iznosa ukupnog ulaznog i izlaznog pojačanja. Treba uočiti da je naglasak na ulaznom signalu koji se sastoji od jednog harmonika. U praksi su mogući signali koji se sastoje od mnoštva harmonika te kao takvi nisu obuhvaćeni gornjom analizom. Zbog toga je moguće da uz takav signal filter izađe iz linearnog područja (dinamike ± 1). To predstavlja naročiti problem kod realizacije filtra u dvostrukoj preciznosti gdje u programu nije provedena aritmetika s zasićenjem te u slučaju izlaska iz dinamike dolazi do WRAP-AROUND efekta (brojevi veći od 1 postaju negativni, a manji od -1 pozitivni). Posljedica toga je javljanje oscilacija visokih frekvencija.

U ovoj točki promatra se unaprijed izračunati niskopropusni eliptički filter 12 reda, granične frekvencije $0.05F_0$, te valovitošću u propusnom području iznosa 0.01dB i gušenjem u području gušenja od 120dB . Provedena je realizacija na ADSP2181 procesoru kao kaskadna izvedba s koeficijentima i stanjima duljine 32 bita te frekvencijom otipkavanja od 48kHz . Izvršna verzija programa nalazi se u datoteci 'sat.exe'. Pritiskom na tipku **Pokreni sat.exe** pokreće se dotičan program na DSP procesoru.

Na ulaz dovesti sinusni signal i mijenjati frekvenciju od minimalno moguće do polovine frekvencije otipkavanja. Promjena ne smije biti previše brza kako bi se filter stigao utitravati. Opisati promjene izlaznog signala u ovisnosti o ulaznoj frekvenciji.

dogđa se WRAP_AROUND efekt - brojevi veći od 1 postaju negativni, a manji od 1 pozitivni

VRIJEDI ZA SVE VALNE OBLIKE

Ponoviti postupak uz trokutni signal na ulazu.

Ponoviti postupak uz pravokutni signal na ulazu.

Opisati dobivene razlike u ovisnosti o odabranom valnom obliku ulaznog signala.
