

## VEŽBA 3 – Obrada signala u frekvencijskom domenu metodom overlap-add

### Potrebno predznanje

- Poznavanje programskog jezika C
- Diskretna Furijeova transformacija

### Šta će biti naučeno tokom izrade vežbe

Tokom izrade ove vežbe biće naučena primena brze Furijeove transformacije u cilju efikasne implementacije FIR filtra korišćenjem osobina Furijeove transformacije i konvolucije. Za rekonstrukciju signala biće korišćena metoda preklapanja i dodavanja (eng. overlap-add). Takođe će biti dat primer obrade koja zahteva manipulaciju spektrom signala koju nije moguće direktno implementirati u vremenskom domenu.

### Motivacija

Obrada signala u frekvencijskom domenu predstavlja veoma važan aspekt obrade digitalnih signala. Pre svega, zbog osobina Furijeove transformacije često korišćene operacije kao što su konvolucija i korelacija u frekvencijskom domenu imaju jednostavniji oblik nego u vremenskom domenu, što omogućava računski efikasniju implementaciju FIR filtara, autokorelacije i sličnih operacija. Takođe, obrada u frekvencijskom domenu omogućava implementaciju algoritama koji zahtevaju direktnu manipulaciju nad amplitudskim ili faznim spektrom signala, što nije uvek moguće postići u vremenskom domenu. Treća važna primena je kompresija signala jer se obradom u transformisanom domenu može bolje kontrolisati raspodela greške kvantizacije i time postići perceptualno bolji kvalitet kompresovanog signala. Pošto praktično svi algoritmi za kompresiju audio signala koriste neku varijantu overlap-add metode, razumevanje ove metode je ključno za razumevanje funkcionisanja audio koda.

## 1 Teorijske osnove

Kao što je rečeno u ranijim vežbama, digitalni filtri prema formi funkcije prenosa mogu da se podele u dve kategorije – filtre sa konačnim impulsnim odzivom (eng. Finite Impulse Response, FIR) i filtre sa beskonačnim impulsnim odzivom (eng. Infinite Impulse Response, IIR). Prednost IIR filtara je to što omogućavaju postizanje mnogo bolje karakteristike sa manjim redom filtra, dok je prednost FIR filtara to što su uvek stabilni i što omogućavaju postizanje linearne fazne karakteristike. S druge strane, osnovna mana FIR filtara je njihova računaska kompleksnost jer je za postizanje karakteristike uporedive sa IIR filtrima osmog ili desetog reda potreban FIR filter sa nekoliko stotina koeficijenata.

Izlazni signal FIR filtra u vremenskom domenu  $y(n)$  računa se kao konvolucija ulaznog signala  $x(n)$  i impulsnog odziva filtra  $h(n)$ :

$$y(n) = x(n) * h(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$$

Kao što se vidi iz ove formule, za implementaciju FIR filtra N-tog reda potrebno je N množenja i N-1 sabiranje po odbirku. Međutim, prelaskom u frekvencijski domen konvolucija prelazi u obično množenje:

$$X(k) = DFT\{x(n)\}, \quad H(k) = DFT\{h(n)\}$$

$$Y(k) = X(k)H(k)$$

$$y(n) = DFT^{-1}\{Y(k)\}$$

S obzirom da algoritam za brzu Furijeovu transformaciju ima složenost  $O(N\log N)$ , za filtriranje u frekvencijskom domenu potrebno je približno  $2\log_2 N + 1$  kompleksnih, odnosno  $8\log_2 N + 4$  realnih množenja po odbirku. U tabeli 1 dato je poređenje broja množenja po odbirku potrebnih za filtriranje u vremenskom i frekvencijskom domenu u zavisnosti od reda FIR filtra.

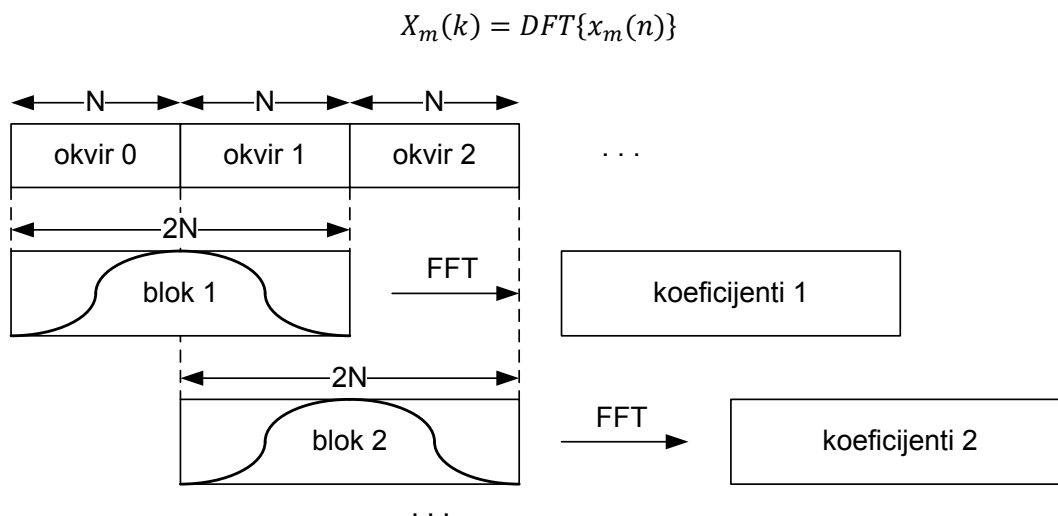
Red filtra	Broj množenja (vremenski domen)	Broj množenja (frekvencijski domen)
<b>32</b>	32	44
<b>64</b>	64	52
<b>128</b>	128	60
<b>256</b>	256	68
<b>512</b>	512	76

Tabela 1 Poređenje kompleksnosti filtriranja u vremenskom i frekvencijskom domenu

Da bi se eliminisala izobličenja u izlaznom signalu koja mogu nastati kao posledica diskontinuiteta na ivicama blokova obrade, filtriranje u frekvencijskom domenu vrši se tako da postoji izvesno preklapanje među susednim blokovima (obično 50%), a blokovi odbiraka se množe s prozorskom funkcijom pre i posle obrade u frekvencijskom domenu. Proces podele signala na blokove i transformacije u frekvencijski domen prikazan je na slici 1. Signal se najpre deli na blokove dužine N koji se obično nazivaju okviri ili frejmovi (eng. frame). Zatim se tekući i prethodni okvir spajaju u jedan blok obrade  $x_m(n)$  dužine  $L=2N$  i množe prozorskom funkcijom  $w(n)$ :

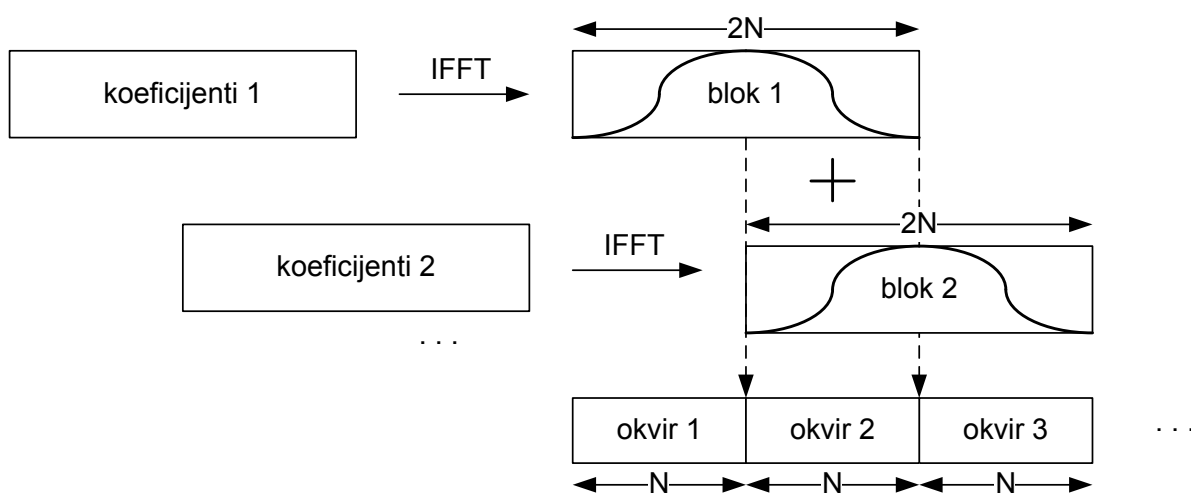
$$x_m(n) = x((m-1)N + n)w(n), \quad n = 0, 1, \dots, 2N-1$$

Posle prozoriranja, blok obrade se transformiše u frekvencijski domen brzom Furijeovom transformacijom dužine  $2N$ , čime se dobija  $2N$  kompleksnih koeficijenata:



Slika 1 Podela ulaznog signala na blokove obrade

Posle obrade u frekvencijskom domenu obrađeni blok odbiraka  $Y_m(k)$  rekonstruiše se Overlap-add metodom kao što je prikazano na slici 2.



Slika 2 Rekonstrukcija izlaznog signala Overlap-add metodom

Obrađeni blok koeficijenata se prvo vrati u vremenski domen primenom inverzne brze Furijeove transformacije kako bi se dobio izlazni blok  $y_m(n)$  dužine  $L=2N$  odbiraka:

$$y_m(n) = DFT^{-1}\{Y_m(k)\}$$

Zatim se prvih  $N$  odbiraka tekućeg bloka i drugih  $N$  odbiraka prethodnog bloka ponovo množe prozorskom funkcijom  $w(n)$  i sabiraju kako bi se rekonstruisao izlazni okvir od  $N$  odbiraka:

$$y(mN + n) = y_m(n)w(n) + y_{m-1}(n + N)w(n + N), \quad n = 0, 1, \dots, N - 1$$

Da bi se obezbedila rekonstrukcija bez izobličenja, prozorska funkcija mora da zadovoljava uslov

$$w(n)^2 + w(N - 1 - n)^2 = 1$$

Jedna od najčešće korišćenih prozorskih funkcija koja zadovoljava ovaj uslov je sinusna prozorska funkcija definisana izrazom

$$w(n) = \sin\left(\frac{\pi}{2}\left(n + \frac{1}{2}\right)\right), \quad n = 0..2N - 1$$

## Zadaci

### Zadatak 1

U datom projektu implementirati funkciju `void obrada(double *in, double *out, int N)` koja realizuje transformaciju ulaznog okvira u frekvencijski domen i rekonstrukciju izlaznog okvira metodom Overlap-add. Funkcija ima sledeće argumente:

- `in` – ulazni okvir
- `out` – izlazni okvir
- `N` – dužina okvira

U implementaciji koristiti sledeće pomoćne bafere koji su definisani u izvornoj datoteci:

- `time_buffer` – blok odbiraka u vremenskom domenu dužine  $2N$
- `fft_buffer` – blok koeficijenata u frekvencijskom domenu dužine  $2N$
- `in_delay` – prethodni ulazni okvir dužine  $N$
- `out_delay` – drugih  $N$  odbiraka prethodnog izlaznog bloka
- `window` – prozorska funkcija dužine  $2N$

Direktna i inverzna diskretna Furijeova transformacija realizovane su u okviru funkcija `void fft(double *in, double *out, int order)` i `void ifft(double *in, double *out, int order)`. Parametar `order` treba da ima vrednost 10 (veličina DFT-a 1024). Format bafera u frekvencijskom domenu je isti kao kod funkcije `rfft` koja je korišćena na DSP platformi:

indeks	0	1	2	3	4	5	...
vrednost	$\text{Re}\{X(0)\}$	$\text{Re}\{X(L/2)\}$	$\text{Re}\{X(1)\}$	$\text{Im}\{X(1)\}$	$\text{Re}\{X(2)\}$	$\text{Im}\{X(2)\}$	

Rešenje testirati datim ulaznim datotekama. U izlaznoj datoteci ne sme biti приметnih izobličenja.

### Zadatak 2

Proširiti funkciju `obrada` implementiranu u prethodnom zadatku tako da vrši filtriranje ulaznog signala pojasnim filtrom sa opsegom frekvencija 300 – 4000 Hz. Pretpostaviti da je frekvencija odabiranja  $f_s = 44100$  Hz.

### Zadatak 3

Modifikovati funkciju `obrada` implementiranu u zadatku 1 tako da se spektar ulaznog signala modifikuje tako da amplitudski spektar ostane nepromenjen a da faza koeficijenta  $X(k)$  bude  $0.25\pi k^2$ .