

Osnove obradbe signala: Brza Fourierova transformacija

Tomislav Petković

Sveučilište u Zagrebu

siječanj 2022.



Uvod

U obradbi signala računalom ograničeni smo na signale konačnog trajanja.

Diskretna Fourierova transformacija u N točaka ili DFT_N je stoga prirodni izbor transformacije za rastav signala konačnog trajanja.

U ovoj cjelini ćemo razmotriti efikasne postupke računanja DFT_N koje zajedno nazivamo **brzom Fourierovom transformacijom** ili skraćeno **FFT**-om, od engl. *fast Fourier transform*.

Osim toga uspostaviti ćemo vezu između DFT_N i DTFT transformacije te pokazati kako se FFT koristi za efikasno računanje konvolucija i korelacija signala konačnog trajanja.

Diskretna kompleksna eksponencijala

Bazne funkcije DFT_N su $\phi_k[n] = \exp(2\pi j \frac{nk}{N})$, $k = 0, \dots, N - 1$.

Radi preglednosti uvodimo oznaku

$$W = \exp(-2\pi j) = e^{-2\pi j}. \quad (1)$$

Desni gornji i donji indeks za W interpretiramo kao elemente brojnika i nazivnika koji množe eksponent $-2\pi j$, odnosno

$$W^k = e^{-2\pi jk}, \quad W_N = e^{-2\pi j/N} \quad \text{i} \quad W_N^{nk} = e^{-2\pi j \frac{nk}{N}}. \quad (2)$$

Sada izraz za računanje spektra možemo napisati kao

$$X[k] = \langle x[n], \phi_k[n] \rangle = \sum_{n=0}^{N-1} x[n] W_N^{nk} \quad (3)$$

Svojstva diskretne eksponencijale

Eksponencijala W_N^n je periodična s periodom N ,

$$W_N^{n+N} = W_N^n \cdot W_N^N = W_N^n \cdot 1 = W_N^n. \quad (4)$$

Pomak eksponenta od $N/2$ u W_N^n samo mijenja predznak,

$$W_N^{n+N/2} = W_N^n \cdot W_N^{N/2} = W_N^n \cdot (-1) = -W_N^n. \quad (5)$$

Navedena svojstva periodičnosti i promjene predznaka ćemo koristiti za povećanje efikanosti računanja DFT_N .

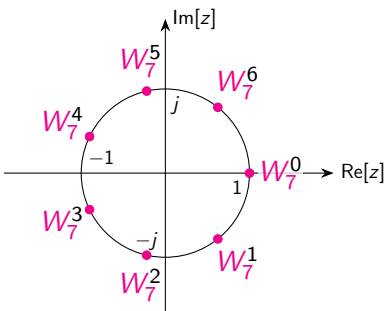
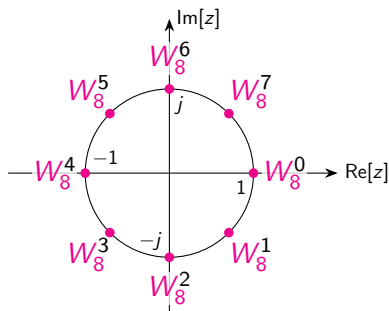
Brojevi W_N leže na kružnici

Eksponencijala W_N^n čini grupu N -tih korijena iz jedinice.

To znači da su kompleksni brojevi

$$z_n = W_N^n = \exp\left(-\frac{2\pi j}{N}n\right) \quad (6)$$

jednoliko raspoređeni po jediničnoj kružnici u kompleksnoj ravlini z .



Broj operacija potreban za računanje DFT_N

U općem slučaju su i signal i spektar kompleksni.

Neka je $x[n] = x_{\text{Re}}[n] + jx_{\text{Im}}[n]$ i neka je $X[k] = X_{\text{Re}}[k] + jX_{\text{Im}}[k]$.

Onda spektar iz signala računamo kao:

$$X_{\text{Re}}[k] = \sum_{n=0}^{N-1} x_{\text{Re}}[n] \cos\left(2\pi \frac{nk}{N}\right) + \sum_{n=0}^{N-1} x_{\text{Im}}[n] \sin\left(2\pi \frac{nk}{N}\right) \quad (7)$$

$$X_{\text{Im}}[k] = - \sum_{n=0}^{N-1} x_{\text{Re}}[n] \sin\left(2\pi \frac{nk}{N}\right) + \sum_{n=0}^{N-1} x_{\text{Im}}[n] \cos\left(2\pi \frac{nk}{N}\right) \quad (8)$$

Broj operacija potreban za izravno računanje DFT_N jest:

- ▶ $4N^2$ množenja realnih brojeva, i
- ▶ $4N^2 - 2N$ zbrajanja realnih brojeva.

Složenost izravnog računanja DFT_N

Obzirom na potreban broj operacija možemo reći da je složenost izravnog računanja DFT_N kvadratna, odnosno $O(N^2)$.

U provedenoj analizi složenosti smo zanemarili:

- ▶ Vrijeme potrebno za prijenos podataka (operacije indeksiranja, adresiranja, dohvata i pohrane).
- ▶ Računanje vrijednosti trigonometrijskih funkcija $\cos(\cdot)$ i $\sin(\cdot)$, no njih je dovoljno jednom izračunati za neki N i pohraniti.

Za moderna računala **ne možemo zanemariti** vremena dohvata i pohrane podataka jer procesor radi značajno brže od memorije i ostalih medija na kojima su pohranjeni podaci.

Bez obzira na to konstrukciju FFT-a započinjemo kroz redukciju složenosti.

Efikasno računanje DFT-a

Pristupe konstrukciji algoritama za efikasno računanje DFT-a u N točaka možemo podijeliti u dvije skupine:

1. faktorizacijske metode koje se koriste principom podijeli-pa-vladaj i koje su primjenjive samo ako je broj točaka N složen broj koji se može rastaviti na svoje faktore, i
2. ostale metode koje su primjenjive i za prosti N .

U prvu skupinu između ostalog spadaju Cooley-Tukeyev algoritam (1965.), Bruunov algoritam (1978.) i algoritam relativno-prostih faktora.

U drugu skupinu između ostalog spadaju Raderov algoritam (1968.), Bluesteinov algoritam (1970.) i Winogradov algoritam (1978.).

Princip podijeli-pa-vladaj i DFT

Sve podijeli-pa-vladaj metode pretpostavljaju da se broj točaka N može rastaviti na produkt svojih faktora,

$$N = r_1 r_2. \tag{9}$$

Pri tome faktori r_1 i r_2 općenito ne moraju biti (relativno) prosti.

Tada je moguće pokazati da se DFT_N može razložiti u kombinaciju DFT_{r_1} i DFT_{r_2} transformacija.

Kako su složenosti DFT_{r_1} i DFT_{r_2} smanjenje s $O(N^2)$ na $O(r_1^2)$ i $O(r_2^2)$ ako iz njih možemo efikasno izračunati traženu DFT_N onda ostvarujemo uštedu u broju operacija.

Cooley-Tukeyev algoritam (1965.) je upravo takav algoritam.

Cooley-Tukeyev algoritam

U Cooley-Tukeyevom algoritmu korak n i indeks k se raspisuju kao

$$n = n_1 r_1 + n_0 \quad \text{i} \quad k = k_1 r_2 + k_0, \quad (10)$$

gdje su n_0 , n_1 , k_0 i k_1 novi cjelobrojni podindeksi.

U izvodu moramo pratiti kako četiri nova podindeksa omogućavaju faktorizaciju eksponencijale

$$W_N^{(n_1 r_1 + n_0)(k_1 r_2 + k_0)}, \quad (11)$$

što izraze čini nepreglednima pa ćemo zato razmotriti jednostavnije slučajeve kada rastavljammo samo n ili samo k .

Istodobnom upotrebom tih jednostavnijih rastava dobivamo Cooley-Tukeyev algoritam (samostalan neobavezan rad).

Korijen-2 decimacijske metode

Neka je N složen broj oblika r^m , gdje su $r, m \in \mathbb{N}$.

Tada metodu podijeli-pa-vladaj zovemo **korijen- r** metoda.

Za binarna računala je jako dobar izbor $r = 2$ koji daje $N = 2^m$ jer omogućava pravilno efikasno razlaganje. Tada govorimo o korijen-2 metodi (engl. *radix-2*).

Ovisno o tome rastavljamo li n ili k govorimo o:

1. **decimaciji u vremenu** (engl. *decimation in time*)
ako n predstavljamo u obliku $n = 2n_1 + n_0$, i
2. **decimaciji u frekvenciji** (engl. *decimation in frequency*)
ako k predstavljamo u obliku $k = 2k_1 + k_0$.

Decimacija signala u vremenu

Za korijen-2 decimaciju u vremenu signal $x[n]$ razlažemo na njegove **parno** i **neparno** indeksirane uzorke.

Neka signal $x[n]$ ima N uzoraka,

$$x[n] = \{x[0], x[1], x[2], x[3], \dots, x[N-2], x[N-1]\}. \quad (12)$$

Parno indeksiranih uzoraka ima $N/2$ i oni su

$$x[2n] = \{x[0], x[2], x[4], \dots, x[N-4], x[N-2]\}. \quad (13)$$

Neparno indeksiranih uzoraka ima $N/2$ i oni su

$$x[2n+1] = \{x[1], x[3], x[5], \dots, x[N-3], x[N-1]\}. \quad (14)$$

Spektar nakon decimacije u vremenu

Primjenimo sada takav rastav na DFT_N :

$$\begin{aligned}
 X[k] &= \text{DFT}_N[x[n]] = \sum_{n=0}^{N-1} x[n] W_N^{nk} \\
 &= \underbrace{\sum_{n=0}^{N/2-1} x[2n] W_N^{2nk}}_{\text{parno indeksirani}} + \underbrace{\sum_{n=0}^{N/2-1} x[2n+1] W_N^{(2n+1)k}}_{\text{neparno indeksirani}} \\
 &= \sum_{n=0}^{N/2-1} x[2n] W_{N/2}^{nk} + W_N^k \sum_{n=0}^{N/2-1} x[2n+1] W_{N/2}^{nk} \\
 &= \text{DFT}_{N/2}[x[2n]] + W_N^k \text{DFT}_{N/2}[x[2n+1]] \quad (15)
 \end{aligned}$$

Dobiveni izraz pokazuje kako pomoću dvije $\text{DFT}_{N/2}$ možemo izračunati DFT_N .

Signali $x[2n]$ i $x[2n+1]$ su duplo kraći od $x[n]$, odnosno $\text{DFT}_{N/2}$ je duljine $N/2$ uzoraka, a DFT_N je duljine N . Kako to rješavamo?

Periodičnost spektra

U rastavu (15) nismo komentirali što se događa s indeksom k .

Za DFT_N indeks k ide od nule do $N - 1$.

Za $\text{DFT}_{N/2}$ indeks k ide od nule do $N/2 - 1$.

No možemo pokazati da je spektar periodičan pa ako nam treba

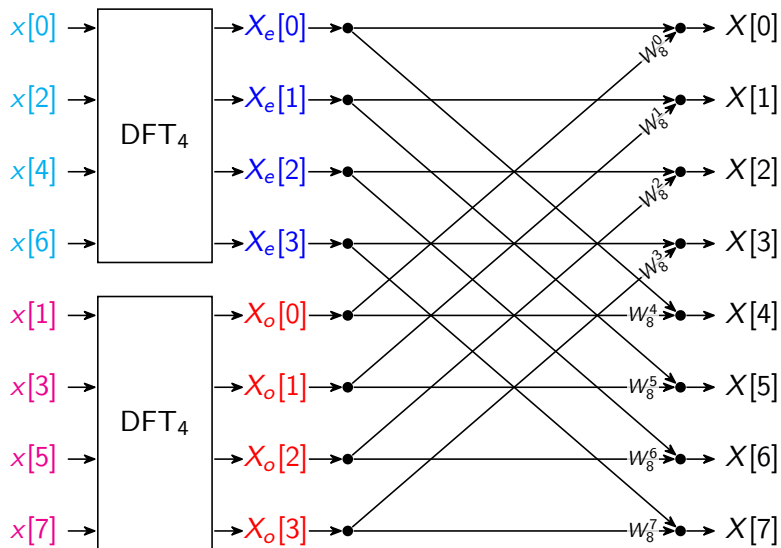
$$X_e[k] = \text{DFT}_{N/2}[x[2n]] \quad \text{i} \quad X_o[k] = \text{DFT}_{N/2}[x[2n+1]] \quad (16)$$

za $k > N/2$ onda radimo periodično proširenje.

U praksi jednostavno koristimo modulo indeksiranje,

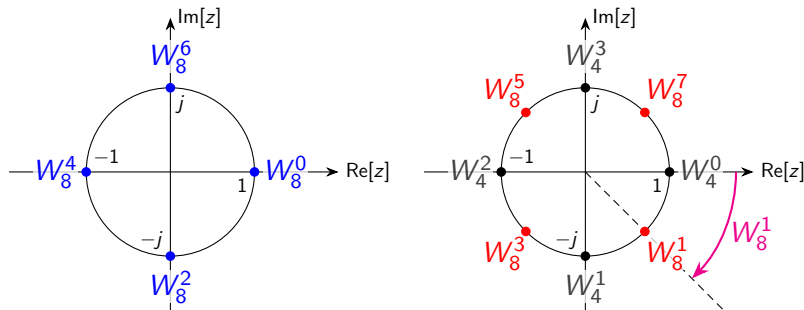
$$X_e[\langle k \rangle_{N/2}] \quad \text{i} \quad X_o[\langle k \rangle_{N/2}], \quad (17)$$

gdje je $\langle k \rangle_N$ oznaka za $k \bmod N$.

Računanje DFT_8 iz dvije DFT_4 

Rotacije u kompleksnoj ravnini

Podjela koju smo napravili za $N = 8$ grupira N -te korijene iz jedinice u dvije podgrupe, W_8^{2n} i W_8^{2n+1} .



Množenjima s W_N^k koja odgovaraju rotacijama (engl. *twiddle*) podgrupu W_4^{nk} koja se preklapa s W_8^{2nk} (parno indeksirani) preklapamo s $W_8^{(2n+1)k}$ (neparno indeksirani).

Rekurzivni rastav

Pokazani rastav možemo ponavljati sve dok ne dođemo do pojedinačnih uzoraka ulaznog signala.

Složenost svakog spajanja dvije $\text{DFT}_{N/2}$ u jednu DFT_N jest:

- ▶ N množenja kompleksnih brojeva, i
- ▶ N zbrajanja kompleksnih brojeva.

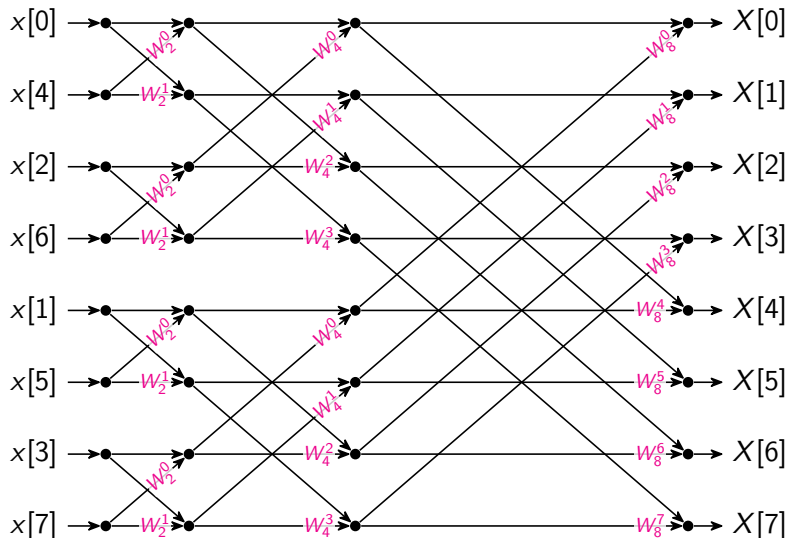
Na svakoj razini rastava složenost svih spajanja jest $O(N)$.

Kako razina rastava ima m ukupna složenost jest $O(mN)$.

Vrijedi $N = 2^m$ pa imamo ukupno $m = \log_2 N$ razina rastava.

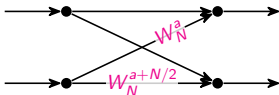
Prema tome konačna složenost potpunog rekurzivnog rastava za korijen-2 decimaciju u vremenu jest $O(N \log_2 N)$.

Potpuni rekurzivni rastav za DFT_8

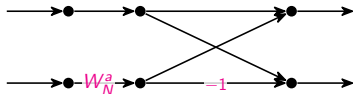


Leptir

Uvidom u dobiveni graf toka signala uočavamo pravilnu ponavljajuću strukturu koju zovemo leptir (engl. *butterfly*).

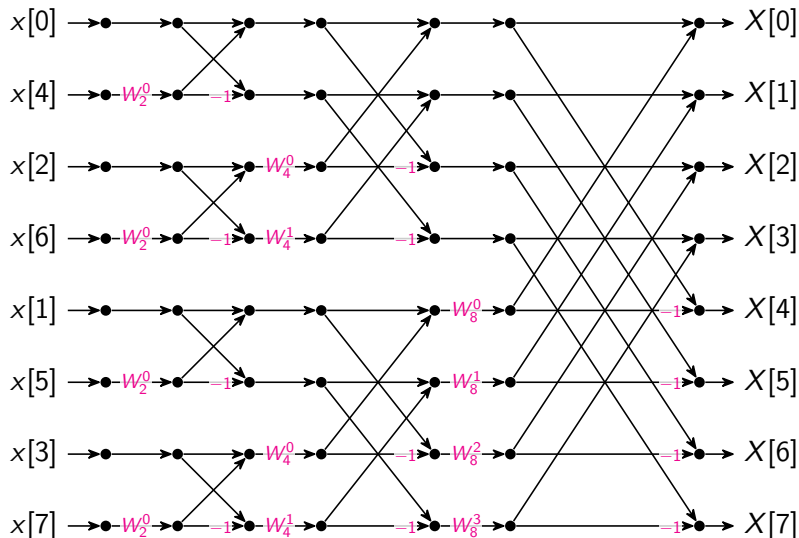


Svaki leptir je moguće dodatno pojednostavniti korištenjem svojstva promjene predznaka izlučivanjem zajedničkog dijela W_N^a , $a \in \mathbb{Z}$.



Time dodatno smanjujemo potreban broj množenja.

Efikasniji potpuni rastav za DFT_8



Poredak uzoraka signala i spektra

U prikazanom grafu toka signala uočavamo da su uzorci spektra $X[k]$ normalno poredani po k dok su uzorci signala $x[n]$ nekako izmiješani po n .

Kažemo da je signal indeksiran zrcalno u bitovima (engl. *bit reversed*), odnosno bitove čitamo zrcalno s desna na lijevo umjesto normalnim redoslijedom.

$(k)_{10}$	0	1	2	3	4	5	6	7
$(k)_2$	000	001	010	011	100	101	110	111
$(n)_2$	000	100	010	110	001	101	011	111
$(n)_{10}$	0	2	4	6	1	3	5	7

Raspolavljanje signala u vremenu

Za korijen-2 decimaciju u frekvenciji signal $x[n]$ razlažemo na njegovu lijevu i desnu polovicu.

Neka signal $x[n]$ ima N uzoraka,

$$x[n] = \{x[0], x[1], \dots, x[\frac{N}{2} - 1], x[\frac{N}{2}], \dots, x[N - 1]\}. \quad (18)$$

Lijeva polovica signala ima $N/2$ uzoraka i oni su

$$x_L[n] = \{x[0], x[1], \dots, x[\frac{N}{2} - 2], x[\frac{N}{2} - 1]\}. \quad (19)$$

Desna polovica signala ima $N/2$ uzoraka i oni su

$$x_R[n] = x[n + \frac{N}{2}] = \{x[\frac{N}{2}], x[\frac{N}{2} + 1], \dots, x[N - 1]\}. \quad (20)$$

Spektar nakon raspolavljanja u vremenu

Primjenimo sada takav rastav na DFT_N :

$$\begin{aligned}
 X[k] &= \text{DFT}_N[x[n]] = \sum_{n=0}^{N-1} x[n] W_N^{nk} \\
 &= \underbrace{\sum_{n=0}^{N/2-1} x[n] W_N^{nk}}_{\text{lijeva polovica}} + \underbrace{\sum_{n=N/2}^{N-1} x[n] W_N^{nk}}_{\text{desna polovica}} \\
 &= \sum_{n=0}^{N/2-1} x[n] W_N^{nk} + \sum_{n=0}^{N/2-1} x[n + \frac{N}{2}] W_N^{(n+N/2)k} \\
 &= \sum_{n=0}^{N/2-1} (x[n] + x[n + \frac{N}{2}] W_2^k) W_N^{nk} \tag{21}
 \end{aligned}$$

Da bi dobili konačno razlaganje moramo provesti decimaciju u frekvenciji.

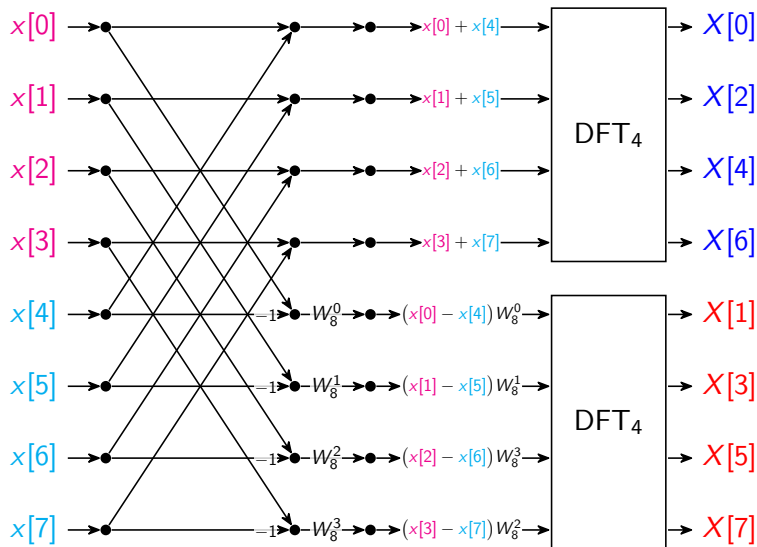
Decimacija spektra u frekvenciji

Decimacijom u frekvenciji za **parno indeksirane** uzorke spektra dobivamo

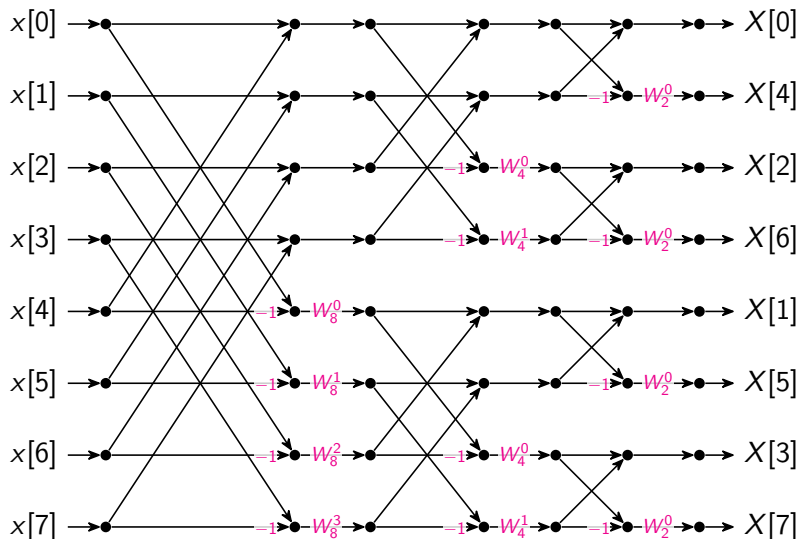
$$\begin{aligned}
 X[2k] &= \sum_{n=0}^{N/2-1} \left(x[n] + x\left[n + \frac{N}{2}\right] W_2^{2k} \right) W_N^{n2k} \\
 &= \sum_{n=0}^{N/2-1} \left(x[n] + x\left[n + \frac{N}{2}\right] \right) W_{N/2}^{nk} \\
 &= \text{DFT}_{N/2} \left[x[n] + x\left[n + \frac{N}{2}\right] \right], \tag{22}
 \end{aligned}$$

a za **neparno indeksirane** uzorke

$$\begin{aligned}
 X[2k+1] &= \sum_{n=0}^{N/2-1} \left(x[n] + x\left[n + \frac{N}{2}\right] W_2^{2k+1} \right) W_N^{n(2k+1)} \\
 &= \sum_{n=0}^{N/2-1} \left(x[n] - x\left[n + \frac{N}{2}\right] \right) W_N^n W_{N/2}^{nk} \\
 &= \text{DFT}_{N/2} \left[\left(x[n] - x\left[n + \frac{N}{2}\right] \right) W_N^n \right]. \tag{23}
 \end{aligned}$$

Računanje DFT_8 pomoću dvije DFT_4 

Potpuni rastav za DFT_8



Prijenos podataka (dohvat i pohrana)

Pokazali kako smanjiti broj operacija, no nismo razmatrali vrijeme potrebno za dohvat podataka.

Današnji procesori su višestruko brži u usporedbi s brzinom unutrašnje (npr. RAM) i vanjske (npr. SSD, HDD) memorije.

Ta razlika se uobičajeno proteže kroz više redova veličine i mora se uzeti u obzir kod svake obradbe signala (ili podataka).

Smanjenje broja operacija ne može ubrzati obradbu signala ako signale (ili bilo koje druge podatke) ne možemo pravodobno dostaviti procesoru.

Koliko traje dohvat podataka?

Razmotrimo okvirno vrijeme dohвата podataka (latencija) te ga normalizirajmo tako da najbrže mjesto pohrane podataka vraća rezultat unutar jedne sekunde.

Gdje su podaci?	Latencija	
	Stvarna	Normalizirana
L1	1 ns	1 s
L2	3 ns	3 s
L3	10 ns	10 s
RAM	< 100 ns	< 2 minute
SSD	50-150 μ s	15 sati-2 dana
HDD	1-10 ms	12 dana-4 mjeseca
Internet	~ 100 ms	~ 3 godine

Samoprilagođujući program

Obzirom na izobilje raznovrsnih računalnih arhitektura te značajno različita vremena latencija nije praktično redizajnirati algoritam za svaku arhitekturu iznova.

Umjesto toga razumniji pristup jest definirati kako se neki program (ili algoritam) može samostalno prilagoditi nekoj konkretnoj računalnoj arhitekturi.

Tada govorimo o **samoprilagođujućim programima**.

Dva temeljna primjera su:

- ▶ ATLAS (od engl. *Automatically Tuned Linear Algebra Software*), vidi tehnički izvještaj R. C. Whaleya i J. J. Dongarra, i
- ▶ FFTW (od engl. *Fastest Fourier Transform in the West*), vidi rad M. Friga i S. G. Johnsona “The Design and Implementation of FFTW3”.

Biblioteka FFTW

U praksi nećete samostalno implementirati FFT nego ćete koristiti neku od gotovih implementacija.

Jako dobra implementacija jest FFTW autora Matteo Friga i Stevena G. Johnsona s MIT-a koja je slobodno dostupna na <https://www.fftw.org/>.

- ▶ Otvorenog je koda (GNU GPL).
- ▶ Napisana u programskom jeziku C.
- ▶ Podržava SIMD instrukcije na većini modernih procesora (Intel, ARM).
- ▶ Samoprilagođuje se hardveru kroz konstrukciju računalnih planova.

Veza DFT-a i DTFT-a

DFT_N transformaciju smo uveli kao transformaciju konačnih signala duljine N uzoraka koja računa spektar duljine N uzoraka.

No DFT_N možemo interpretirati kao očitavanje spektra DTFT-a.

Takva interpretacija nam omogućava da primjenom FFT-a efikasno numerički izračunamo očitani DTFT.

Razmotritmo sada vezu između DFT-a i DTFT-a.

Za to nam treba svojstvo **dualnosti**.

Svojstvo **dualnosti** se uobičajeno izriče za signale, no primijetite da ono za sobom osim dualnosti signala povlači i dualnost svojstava Fourierove transformacije.

Dualnost periodičnosti i diskretnosti

Za Fourierove transformacije **periodičnost** i **diskretnost** su **dualne**, odnosno **periodičnost** u jednoj domeni povlači **diskretnost** u drugoj domeni.

Prema tome vrijedi:

- ▶ **periodičan** signal ima **diskretan** spektar,
- ▶ **diskretan** signal ima **periodičan** spektar, i
- ▶ **periodičan** i **diskretan** signal ima **diskretan** i **periodičan** spektar.

Zadnji slučaj odgovara upravo DFT_N transformaciji jer su za nju i signal i spektar **diskretni** što znači da su ujedno i **periodični**.

Dualnost i preklapanje u vremenu

Uočimo da **očitanje spektra** može uzrokovati **preklapanje signala**.

Prilikom izvoda teorema o očitavanju smo pokazali da **očitanje signala** uzrokuje **periodizaciju spektra**, odnosno ako je $\text{CTFT}[x_c(t)] = X_c(\Omega)$ i ako je $x_d[n] = x_c(nT_s)$ onda je

$$\text{DTFT}[x_d[n]] = X_d(\omega) = \frac{1}{T_s} \sum_{m \in \mathbb{Z}} X_c(\omega/T_s - 2\pi m). \quad (24)$$

Dualno, ako **očitamo spektar** $X_d(\omega)$ s frekvencijskim korakom očitavanja $\omega_s = 2\pi/N$ onda u vremenskoj domeni dolazi do **periodizacije signala**, odnosno

$$\text{IDTFT}[X_d(\omega) \text{comb}_{\omega_s}(\omega)] = \frac{N}{2\pi} \sum_{m \in \mathbb{Z}} x_d[n - Nm]. \quad (25)$$

Dualnost i preklapanje u vremenu

U jednadžbi (25) nema **preklapanja signala** u vremenu ako je signal konačnog trajanja L manjeg ili jednakog N .

Kako u praksi uvijek radimo s konačnim brojem uzoraka možemo jednostavno spriječiti **preklapanje signala**.

Ograničenje na konačan broj od L uzoraka modeliramo množenjem signala s pravokutnim vremenskim otvorom

$$w_L[n] = \begin{cases} 1, & 0 \leq n < L \\ 0, & \text{inače} \end{cases}. \quad (26)$$

Za razliku od očitavanja signala gdje idealni filter protiv preklapanja spektra uopće nije realizabilan, od **preklapanja signala** nas idealno štiti upravo vremenski otvor.

Dualnost i preklapanje u vremenu

Slično kao kod Nyquistovog uvjeta sada je jasno da nema **preklapanja signala** ako je zadovoljen uvjet:

$$\omega_s = \frac{2\pi}{N} \leq \frac{2\pi}{L}. \quad (27)$$

Dakle za svaki $N \geq L$ nema **preklapanja signala** pa vrijedi

$$\text{DFT}_N[x[n]w_L[n]] = X[k] = X(\omega_s k) \quad (28)$$

gdje je

$$X(\omega) = \text{DTFT}[x[n]w_L[n]]. \quad (29)$$

Napomena: Konstanta $\frac{N}{2\pi}$ iz (25) nestaje kada umjesto korištenja funkcije comb uzmemo uzorke spektra u $\omega_k = 2\pi \frac{k}{N}$.

Primjene FFT-a

FFT i sve njegove izvedenice su često korištene u praksi jer ih kako smo do sada pokazali možemo koristiti za efikasno računanje spektra DFT_N i DTFT transformacija.

Osim računanja spektra FFT se još između ostalog koristi za:

1. efikasno računanje **konvolucija** konačnih signala,
2. efikasno računanje **korelacija** konačnih signala,
3. efikasnu blokovsku obradbu signala neograničenog trajanja,
4. efikasno množenje velikih cijelih brojeva, itd.

Na ovom predmetu ćemo samo razmotriti prve dvije primjene, dakle za efikasno računanje **konvolucije** i **korelacije**.

Linearna konvolucija

Linearna konvolucija dva signala $x[n]$ i $y[n]$ je

$$x[n] * y[n] = \sum_{m \in \mathbb{Z}} x[m] y[n - m]. \quad (30)$$

Kod računanja računalom od posebnog interesa je konvolucija signala konačnog trajanja.

Neka $x[n]$ ima N_x uzoraka različitih od nule za n od 0 do $N_x - 1$.

Neka $y[n]$ ima N_y uzoraka različitih od nule za n od 0 do $N_y - 1$.

Onda $x[n] * y[n]$ ima najviše $N_x + N_y - 1$ uzoraka različitih od nule za n od 0 do $N_x + N_y - 1$.

Izravna složenost računanja (30) jest $O(N_x N_y)$, a za slučaj $N_x = N_y = N$ složenost postaje kvadratna $O(N^2)$.

Računanje linearne konvolucije

Neka je $x[n] = \{2, 1, 0, -2\}$ i $y[n] = \{4, 0, -2, 3\}$.

Izračunajte $z[n] = x[n] * y[n]$.

Moramo izračunati

$$z[n] = x[n] * y[n] = \sum_{m \in \mathbb{Z}} x[m]y[n - m], \quad (31)$$

što raspisano postaje:

$$\begin{aligned} z[0] &= x[0]y[0] \\ z[1] &= x[0]y[1] + x[1]y[0] \\ z[2] &= x[0]y[2] + x[1]y[1] + x[2]y[0] \\ z[3] &= x[0]y[3] + x[1]y[2] + x[2]y[1] + x[3]y[0] \\ z[4] &= x[1]y[3] + x[2]y[2] + x[3]y[1] \\ z[5] &= x[2]y[3] + x[3]y[2] \\ z[6] &= x[3]y[3] \end{aligned} \quad (32)$$

Računanje linearne konvolucije

Da bi izračunali konvoluciju prema tome moramo izračunati umnoške svih parova uzoraka što je najlakše napraviti tablično:

		$z[0]$	$z[1]$	$z[2]$	$z[3]$	$z[4]$	$z[5]$	$z[6]$
		<u>2</u>	1	0	-2			
<u>4</u>	8	4	0	-8				
0	0	0	0	0				
-2	-4	-2	0	4				
3	6	3	0	-6				

Zbrajanje elemenata po obrnutim dijagonalama daje rješenje

$$z[n] = x[n] * y[n] = \{\underline{8}, 4, -4, -4, 3, 4, -6\}. \quad (33)$$

Cirkularna konvolucija

Cirkularnu, kružnu ili cikličku konvoluciju definiramo za signale konačnog trajanja.

Cirkularna konvolucija dva signala $x[n]$ i $y[n]$ jednake duljine od N uzoraka jest novi signal od N uzoraka definiran kao

$$x[n] \circledcirc y[n] = \sum_{m=0}^{N-1} x[m] y[\langle n - m \rangle_N] \quad (34)$$

gdje je $\langle n - m \rangle_N$ oznaka za $n - m \bmod N$.

Kada znamo N umjesto opće oznake \circledcirc možemo pisati npr. $x \circledcirc y$ za $N = 4$ ili $x \circledcirc_{14} y$ za $N = 14$, a ako je N jasan iz konteksta onda koristimo i $x \circledast y$.

Izravna složenost računanja izraza (34) jest $O(N^2)$.

Računanje cirkularne konvolucije

Neka je $x[n] = \{2, 1, 0, -2\}$ i $y[n] = \{4, 0, -2, 3\}$.
Izračunajte $x[n] \circledast y[n]$.

Moramo izračunati

$$\tilde{z}[n] = x[n] * y[n] = \sum_{m \in \mathbb{Z}} x[m] y[\langle n - m \rangle_N], \quad (35)$$

što raspisano postaje:

$$\begin{aligned} \tilde{z}[0] &= x[0]y[0] + x[1]y[3] + x[2]y[2] + x[3]y[1] \\ \tilde{z}[1] &= x[0]y[1] + x[1]y[0] + x[2]y[3] + x[3]y[2] \\ \tilde{z}[2] &= x[0]y[2] + x[1]y[1] + x[2]y[0] + x[3]y[3] \\ \tilde{z}[3] &= x[0]y[3] + x[1]y[2] + x[2]y[1] + x[3]y[0] \end{aligned} \quad (36)$$

Računanje cirkularne konvolucije

Kod računanja (36) primijetite **preklapanje elemenata** u gornjem desnom kutu koje je značajna razlika u odnosu na linearnu konvoluciju (32) i koje je posljedica preklapanja signala (25).

No zbog toga cirkularnu konvoluciju možemo izračunati izravno iz linearne konvolucije periodizacijom s periodom 4:

$$\begin{array}{rcccc}
 & \underline{8} & 4 & -4 & -4 \\
 + & \textcolor{teal}{3} & \textcolor{teal}{4} & \textcolor{teal}{-6} & \\
 \hline
 \underline{11} & 8 & -10 & -4 &
 \end{array}$$

Tražena cirkularna konvolucija jest

$$\tilde{z}[n] = \textcolor{violet}{x}[n] \textcircled{4} \textcolor{teal}{y}[n] = \{\underline{11}, 8, -10, -4\}. \quad (37)$$

Preslikavanje konvolucije za DTFT i za DFT_N

Teorem o konvoluciji kaže da se konvolucija u vremenskoj domeni preslikava u običan umnožak u frekvencijskoj domeni.

Sukladno teoremu o konvoluciji za DTFT vrijedi

$$\text{DTFT}[x[n] * y[n]] = \text{DTFT}[x[n]] \cdot \text{DTFT}[y[n]]. \quad (38)$$

Slično, za DFT_N također vrijedi

$$DFT_N[x[n] \circledast y[n]] = DFT_N[x[n]] \cdot DFT_N[y[n]]. \quad (39)$$

Primijetite da su konvolucije **različite**, za DTFT koristimo linearnu, a za DFT_N cirkularnu konvoluciju.

Konvolucija i FFT

Ako su oba signala duljine N uzoraka onda je složenost računanja linearne i cirkularne konvolucije $O(N^2)$.

Korištenjem FFT-a možemo za velike N efikasnije izračunati **cirkularnu** konvoluciju kao

$$x[n] \circledast y[n] = \text{IDFT}_N [\text{DFT}_N [x[n]] \cdot \text{DFT}_N [y[n]]] \quad (40)$$

Složenost običnog množenja je $O(N)$. Ako koristimo FFT onda je složenost tri transformacije $O(N \log_2 N)$. Prema tome ukupna složenost računanja cirkularne konvolucije pomoću izraza (40) je $O(N \log_2 N)$, no s dosta većom skrivenom konstantom.

No za filtraciju signala nam treba **linearna** konvolucija!

Linearna pomoću cirkularne konvolucije

Ako želimo da cirkularna konvolucija postane jednaka linearnoj onda:

- ▶ moramo odabrati N takav da $x[n]$, $y[n]$ i $z[n] = x[n] * y[n]$ budu kraćeg trajanja ili jednakog N kako bi izbjegli **preklapanje signala**,
- ▶ moramo osigurati da **preklapanje** u (36) bude nula, i
- ▶ moramo osigurati da (36) postane jednak (32).

Sve navedeno se postiže **dometkom nula** na desno do duljine N koja je veća ili jednaka duljini svih signala $x[n]$, $y[n]$ i $z[n]$ (provjerite samostalno za vježbu).

Veza linearne i cirkularne konvolucije

Neka su $x[n]$ i $y[n]$ signali konačnog trajanja duljina N_x i N_y .

Ako želimo da za $n = 0, 1, \dots, N - 1$ vrijedi

$$z[n] = x[n] * y[n] = x[n] \circledast y[n] = \tilde{z}[n] \quad (41)$$

onda

- ▶ moramo odabrati $N \geq N_x + N_y - 1$ i
- ▶ moramo signale $x[n]$ i $y[n]$ proširiti nulama nadesno do duljine N .

Napomena: Radi efikasnosti često biramo najmanji N koji je potencija broja 2 veća ili jednaka $N_x + N_y - 1$.

Deterministička linearna korelacija

Deterministička linearna korelacija dva signala $x[n]$ i $y[n]$ jest:

$$r_{xy}[n] = x[n] \star y[n] = \sum_{m \in \mathbb{Z}} x[n+m] y^*[m] \quad (42)$$

$$r_{yx}[n] = y[n] \star x[n] = \sum_{m \in \mathbb{Z}} y[n+m] x^*[m] \quad (43)$$

Korelacija nije komutativna! No za relane signale vrijedi

$$r_{xy}[n] = r_{yx}[-n]. \quad (44)$$

Ako su $x[n]$ i $y[n]$ konačnog trajanja od N_x i N_y uzoraka onda $r_{xy}[n]$ i $r_{yx}[n]$ imaju najviše $N_x + N_y - 1$ uzoraka različitih od nule:

- ▶ $r_{xy}[n]$ na intervalu od $-N_y + 1$ do $N_x - 1$, i
- ▶ $r_{yx}[n]$ na intervalu od $-N_x + 1$ do $N_y - 1$.

Računanje linearne korelacije

Neka je $x[n] = \{2, 1, 0, -2\}$ i $y[n] = \{4, 0, -2, 3\}$.

Izračunajte $r_{xy}[n] = x[n] \star y[n]$.

Moramo izračunati

$$r_{xy}[n] = x[n] \star y[n] = \sum_{m \in \mathbb{Z}} x[n+m] y^*[m], \quad (45)$$

što raspisano postaje:

$$\begin{aligned} r_{xy}[-3] &= x[0]y[3] \\ r_{xy}[-2] &= x[0]y[2] + x[1]y[3] \\ r_{xy}[-1] &= x[0]y[1] + x[1]y[2] + x[2]y[3] \\ r_{xy}[0] &= x[0]y[0] + x[1]y[1] + x[2]y[2] + x[3]y[3] \\ r_{xy}[1] &= x[1]y[0] + x[2]y[1] + x[3]y[2] \\ r_{xy}[2] &= x[2]y[0] + x[3]y[1] \\ r_{xy}[3] &= x[3]y[0] \end{aligned} \quad (46)$$

Računanje linearne korelacije

Da bi izračunali korelaciju opet moramo izračunati umnoške svih parova uzoraka pa to radimo tablično:

	<u>2</u>	1	0	-2
<u>4</u>	<u>8</u>	4	0	-8
0	0	0	0	0
-2	-4	-2	0	4
3	6	3	0	-6

$r_{xy}[-3]$ $r_{xy}[-2]$ $r_{xy}[-1]$ $r_{xy}[0]$ $r_{xy}[1]$ $r_{xy}[2]$ $r_{xy}[3]$

Zbrajanje elemenata po dijagonalama daje rješenje

$$r_{xy}[n] = x[n] \star y[n] = \{6, -1, -2, \underline{2}, 8, 0, -8\}. \quad (47)$$

Deterministička cirkularna korelacija

Deterministička cirkularna korelacija dva signala $x[n]$ i $y[n]$ duljine N je novi signal iste duljine definiran kao:

$$\tilde{r}_{xy}[n] = x[n] \star y[n] = \sum_{m=0}^{N-1} x[\langle n+m \rangle_N] y^*[m] \quad (48)$$

$$\tilde{r}_{yx}[n] = y[n] \star x[n] = \sum_{m=0}^{N-1} y[\langle n+m \rangle_N] x^*[m] \quad (49)$$

Slično kao i za linearnu korelaciju za realne signale vrijedi

$$\tilde{r}_{xy}[n] = \tilde{r}_{yx}[\langle -n \rangle_N]. \quad (50)$$

Izravna složenost računanja izraza (48) i (49) jest $O(N^2)$.

Računanje cirkularne korelacije

Neka je $x[n] = \{2, 1, 0, -2\}$ i $y[n] = \{4, 0, -2, 3\}$.

Izračunajte $\tilde{r}_{xy}[n] = x[n] \star y[n]$.

Moramo izračunati

$$\tilde{r}_{xy}[n] = x[n] \star y[n] = \sum_{m \in \mathbb{Z}} x[\langle n + m \rangle_N] y^*[m], \quad (51)$$

što raspisano postaje:

$$\begin{aligned} \tilde{r}_{xy}[0] &= x[0]y[0] + x[1]y[1] + x[2]y[2] + x[3]y[3] \\ \tilde{r}_{xy}[1] &= x[0]y[3] + x[1]y[0] + x[2]y[1] + x[3]y[2] \\ \tilde{r}_{xy}[2] &= x[0]y[2] + x[1]y[3] + x[2]y[0] + x[3]y[1] \\ \tilde{r}_{xy}[3] &= x[0]y[1] + x[1]y[2] + x[2]y[3] + x[3]y[0] \end{aligned} \quad (52)$$

Računanje cirkularne konvolucije

Kod računanja (52) primijetite **preklapanje elemenata** u donjem lijevom kutu koje je značajna razlika u odnosu na linearnu korelaciju (46) i koje je posljedica preklapanja signala (25).

No zbog toga cirkularnu korelaciju možemo izračunati izravno iz linearne korelacije periodizacijom s periodom 4:

$$\begin{array}{rcccc}
 & \underline{2} & 8 & 0 & -8 \\
 + & & \textcolor{teal}{6} & \textcolor{teal}{-1} & \textcolor{teal}{-2} \\
 \hline
 & \underline{2} & 14 & -1 & -10
 \end{array}$$

Tražena cirkularna korelacija jest

$$\tilde{r}_{xy}[n] = \textcolor{violet}{x}[\textcolor{violet}{n}] \star \textcolor{teal}{y}[\textcolor{teal}{n}] = \{\underline{2}, 14, -1, -10\}. \quad (53)$$

Preslikavanje korelacije za DTFT i za DFT_N

Teorem o korelaciji kaže da se korelacija u vremenskoj domeni preslikava u umnožak dva spektra od kojih je drugi konjugiran u frekvencijskoj domeni.

Sukladno teoremu o korelaciji za DTFT vrijedi

$$\text{DTFT}[\textcolor{red}{x}[n] \star \textcolor{blue}{y}[n]] = \text{DTFT}[\textcolor{red}{x}[n]] \cdot \text{DTFT}[\textcolor{blue}{y}[n]]^* \quad (54)$$

$$\text{DTFT}[\textcolor{blue}{y}[n] \star \textcolor{red}{x}[n]] = \text{DTFT}[\textcolor{blue}{y}[n]] \cdot \text{DTFT}[\textcolor{red}{x}[n]]^* \quad (55)$$

Slično, za DFT_N također vrijedi

$$DFT_N[\textcolor{red}{x}[n] \star \textcolor{blue}{y}[n]] = DFT_N[\textcolor{red}{x}[n]] \cdot DFT_N[\textcolor{blue}{y}[n]]^* \quad (56)$$

$$DFT_N[\textcolor{blue}{y}[n] \star \textcolor{red}{x}[n]] = DFT_N[\textcolor{blue}{y}[n]] \cdot DFT_N[\textcolor{red}{x}[n]]^* \quad (57)$$

Linearna deterministička korelacija i FFT

Slično kao i kod konvolucije za efikasno računanje determinističke korelacije možemo koristiti FFT uz složenost $O(N \log_2 N)$ uz dosta veću skrivenu konstantu.

Glavna razlika u odnosu na računanje konvolucije jest da osim dometka nula rezultat također pomičemo u vremenu.

Naime, signal korelacije postoji i lijevo i desno od točke $n = 0$ dok standardne implementacije FFT-a očekuju desne signale koji počinu od $n = 0$ pa zato:

- ▶ računamo li $r_{xy}[n]$ koraku n dodajemo $-N_y + 1$, i
- ▶ računamo li $r_{yx}[n]$ koraku n dodajemo $-N_x + 1$.

Linearna pomoću cirkularne korelacije

Ako želimo da cirkularna korelacija postane jednaka linearnoj onda:

- ▶ moramo odabrati N takav da $x[n]$, $y[n]$ i $r_{xy}[n] = x[n] \star y[n]$ budu kraćeg trajanja od N kako bi izbjegli preklapanje signala,
- ▶ moramo osigurati da **preklapanje** u (52) bude nula, i
- ▶ moramo osigurati da (52) postane jednak (46) do na pomak.

Sve navedeno se postiže **dometkom nula** do duljine N koja je veća ili jednaka duljini svih signala $x[n]$, $y[n]$ i $r_{xy}[n]$ pri čemu:

- ▶ prvom signalu $x[n]$ dodajemo nule nalijevo i
- ▶ drugom signalu $y[n]$ dodajemo nule nadesno.

Veza linearne i cirkularne korelacije

Neka su $x[n]$ i $y[n]$ signali konačnog trajanja duljina N_x i N_y .

Ako želimo da za $n = 0, 1, \dots, N - 1$ vrijedi

$$\begin{aligned} r_{xy}[n - N_y + 1] &= x[n - N_y + 1] \star y[n] \\ &= x[n - N_y + 1] \oplus y[n] = \tilde{r}_{xy}[n] \end{aligned} \quad (58)$$

onda

- ▶ moramo odabrati $N \geq N_x + N_y - 1$,
- ▶ moramo signal $x[n]$ proširiti s $N_y - 1$ nula nalijevo i pomaknuti za $N_y - 1$ nadesno, i
- ▶ moramo signal $y[n]$ proširiti s $N_x - 1$ nula nadesno.

Preporučeno čitanje

- ▶ P. Prandoni, M. Vetterli, “Signal Processing for Communications” (<https://sp4comm.org/>), dio 4.7.2.
- ▶ Ruye Wang, “Introduction to Orthogonal Transforms: With Applications in Data Processing and Analysis”, Cambridge University Press, dio 4.2.5.
- ▶ H. Babić, “Signali i sustavi” (http://sis.zesoi.fer.hr/predavanja/pdf/sis_2001_skripta.pdf), dio 14.8.
- ▶ M. Vetterli, J. Kovačević, V. K. Goyal, “Foundations of Signal Processing” (<https://fourierandwavelets.org/>), poglavlje 3. dijelovi 3.9.1. i 3.9.2.