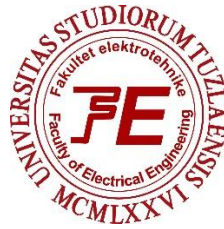


UNIVERZITET U TUZLI
FAKULTET ELEKTROTEHNIKE
ODSJEK: TELEKOMUNIKACIJE



PROJEKAT

Obrada digitalnih signala

Ime i prezime: Ernad Kovačević

Tuzla, datum 12.1.2021.

IZVJEŠTAJ:

Prilikom pokretanja programa dobijamo meni koji nam pruža odgovarajuće mogućnosti

- DFT
- IDFT
- Inverzna Z transformacija
- Pomijeranje sekvence

DFT:

DFT predstavlja diskretnu furijerovu transformaciju koja se u pythonu ostvaruje sa funkcijom `fft()` iz `scipy` modula. Radi samo sa pozitivnim dijelom X ose. Kada izaberemo ovu opciju prvo dobijamo zahtjev za unos sekvence na koju želimo primijeniti `dft` a funkcija za unos sekvence je definisana u `DFT.py` fajlu.

Funckija `unos()` prvo zahtijeva da se unese veličina sekvence `N`. Zatim se kreira prazan niz `x=np.array([])` u kojeg ćemo spremati amplitude naše sekvence. Unos amplitude se vrši sve dok je `i`, koje je inicijalizirano na vrijednost 0, manje od `N`. Nakon toga tu sekvenciju proslijeđujemo u funkciju `dft()` koja je implementirana u `DFT.py` fajlu.

Potom se prvo iscertava naša ulazna sekvenca pomoću funkcije `stem()` iz `matplotlib.pyplot` modula a onda se izvršava `fft()` po sljedećoj formuli.

$$y[k] = \sum_{n=0}^{N-1} e^{-2\pi j \frac{kn}{N}} x[n],$$

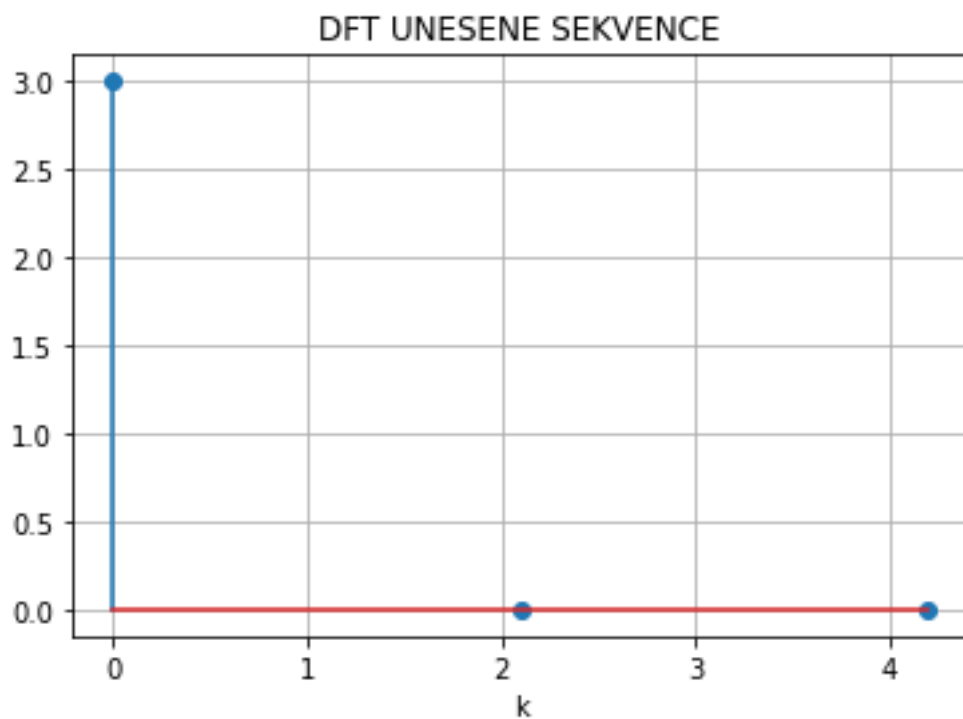
Isertavanje `fft`-a vršimo u granicama od 0 do 2π pa se upravo radi toga koristi funkcija `linspace(0,pi*2,N+1)`, `N=len(„sekevenca“)` i predstavlja broj uzoraka. Ako nam osa bude veća nego što smo je zadali na početku mozemo koristiti `w=w[:-1]` sa kojom oduzimamo uzorak ili više njih ako je potrebno tj služi nam da se ose poklope.

Nakon toga se ponovo pomoću funkcije `stem()` crta spektar dobijen sa `fft` i ispisuju se vrijednosti amplitude `x(k)`.

Primjer za sekvencu `x[n]={1,1,1}` `n` počinje od 0 .



Slika 1.1. Ulazna sekvenca $x[n]$



Slika 1.2. Spektar $x(k)$

IDFT:

Inverzna DFT se određuje preko formule:

Tuzla, datum 12.1.2021.

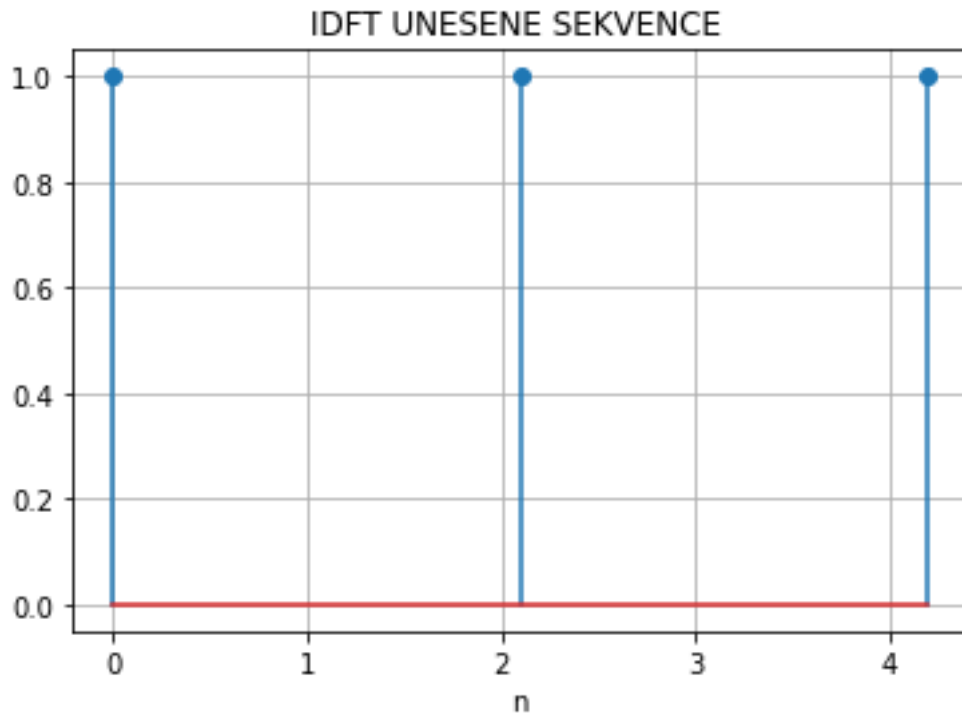
$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} e^{2\pi j \frac{kn}{N}} y[k]$$

Funkcija u pythonu koju koristimo je **ifft()**. Kada izaberemo opciju „2“ prvo se traži unos sekvence što je ista funkcija kao i ona koja je korištena za DFT. Nakon što unesemo sekvencu poziva se funkcija **idft()** koja je implementirana u DFT.py. Ostalo je sve isto.

Primjer: $x(k) = \{3, 0, 0\}$



Slika 1.3. Sekvenca $x(k)$, koja je ista kao i u predhodnom zadatku što smo dobili dft-om



Slika 1.4. Sekvenca x(n) i vidimo da odgovara sekvenci iz primjera DFT

INVERZNA Z TRANSFORMACIJA:

Inverzna Z transformacija nam zahtijeva prvo unos nula i polova sa funkcijama **unos.polova()** i **unos.nula()**. Funkcije za unos nula i polova su iste jedino se razlikuju u tekstu koji se ispisuje prilikom unosa .

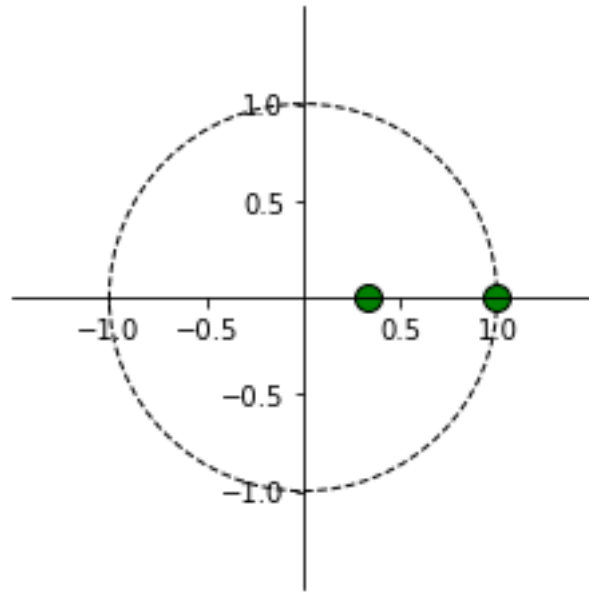
Funckije unosa rade tako da se napravi jedan prazan niz sa `np.array([])` i definisemo neki brojač `i=0`, nakon toga unosimo koliko polova ili nula imamo . Npr: ako imamo oblik $1+z^{-3}$ unosimo `N=4` (1,0,0,1), nakon čega ulazimo u while petlju koja će se izvršavati sve dok je `i<N`. Unutar petlje pomoću komande **insert()** ubacujemo elemente na kraj niza.

Kada unesemo sekvence koje predstavljaju polove i nule vrijednosti koje nam vrate funkcije `unos.polova()` i `unos.nula()` prosljeđujemo u funkciju `i_z_tr(pol,nule)` koja je implementirana u `Z.py`. Unutar te funkcije imamo funkciju **zplane()**, koja nam crta jediničnu kružnicu i na njoj crta nule i polove, gdje su nule označene kružićima a polovi sa X.

Funckija **residues()** iz modula `scipy` nam daje zapis RPK. Gdje R predstavlja koeficijente odnosno amplitude, P su polovi a K je ostatak.

Primjer:

$$X(z) = \frac{z}{3z^2 - 4z + 1}$$



Slika 1.5. Jedinična kružnica dobijena sa funkcijom `zplane()`

```
(array([-0.5,  0.5]), array([0.33333333, 1.        ]), array([], dtype=float64))
```

Slika 1.6. Ispis koji se dobija korištenjem funkcije `residues()`

POMIJERANJE SEKVENCE ($n \pm N$):

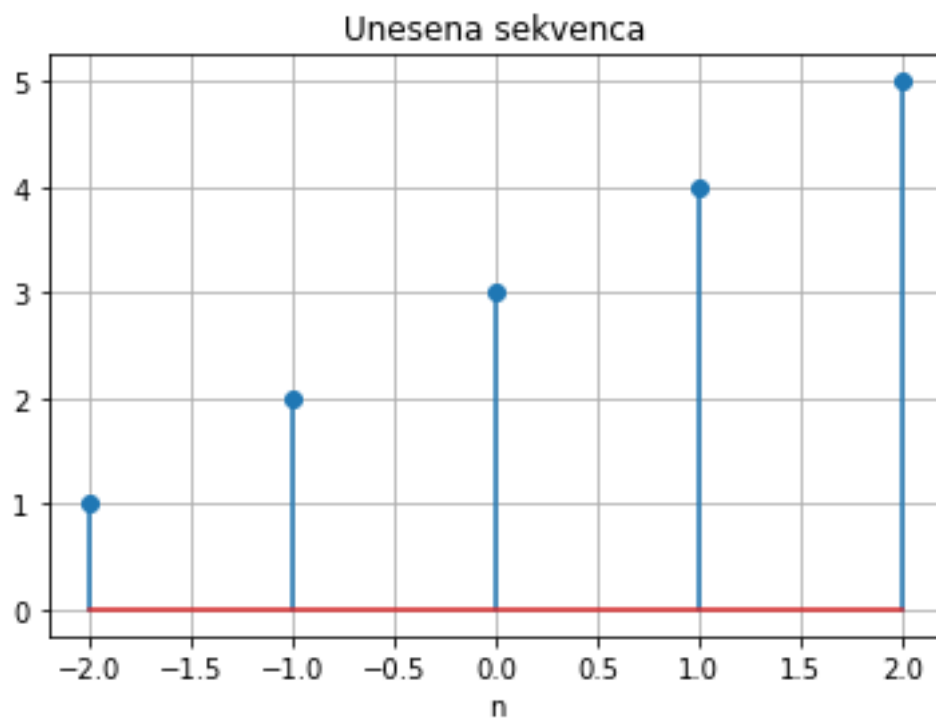
Ukoliko izaberemo opciju 4 koja je ujedno i zadnja opcija koju možemo koristiti u ovom projektu, dobijamo mogućnost za unos sekvence koju možemo pomijerati lijevo ili desno. Izvršava se funkcija **unos()** koja je implementirana u `pomijeranje.py` fajlu. Prvo zahtijeva unos donje i gornje granice x ose, koje koristi za `np.arange()` za crtanje signala.

Prvi dio funkcije je sam unos naše sekvence postupak je isti kao i u predhodnim slučajevima (`dft`, `idft`). Nakon toga se od korisnika traži da unese broj za koji želi pomjeriti sekvencu. Ako se izabere negativan broj znači da će se i sekvenca zakasnit za tu vrijednost (pomijeranje u desno) , a pozitivan broj pomijera u lijevo, nakon čega se crta naša unešena sekvenca

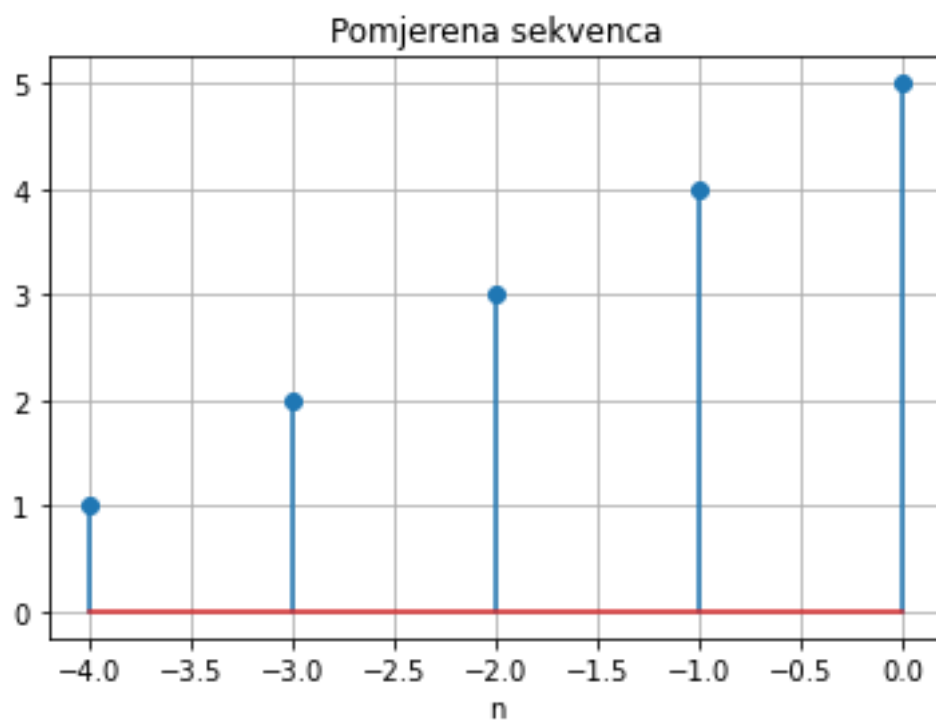
U zavisnosti od broja kojeg smo unijeli za pomijeranje imamo if naredbu koja crta signale za odabrane slučajeve.

Primjer:

$x[n] = \{1, 2, 3, 4, 5\}$ $x[n+2] = ?$



Slika 1.7. Sekvenca $x[n]$



Slika 1.8. $X[n+2]$

Tuzla, datum 12.1.2021.