

Laboratorijska vježba 07: Uzorkovanje i kvantizacija

Za izradu laboratorijske vježbe treba koristiti odgovarajuću Jupyter Notebook datoteku. Urađenu vježbu je potrebno konvertirati u PDF format, a zatim je PDF datoteku potrebno predati do postavljenog roka koristeći platformu Zamger.

Ime i prezime studenta, broj indeksa:

Amar Hasečić, 2116/18673

Datum izrade izvještaja:

07.05.2024

Zadatak 1.

Potrebno je implementirati funkciju *sinusoida(pocetak, kraj, frekv)* koja izračunava funkciju $y = \sin(x)$ u intervalu $[pocetak, kraj]$ sa korakom $1/frekv$. Ova funkcija kao rezultat treba vratiti niz vrijednosti x i y , respektivno.

Nakon toga potrebno je implementirati funkciju *uzorkovanje(x, y, fuzor, frekv)* koja uzorkuje funkciju y u specifičnim tačkama x . Funkcija vraća nove uzorkovane nizove x_{new} i y_{new} . Parametar *fuzor* definira frekvenciju uzorkovanja na način da se novi uzorkovani nizovi x_{new} i y_{new} formiraju tako da se iz postojećih nizova x i y uzima svaki $frekv/fuzor$ uzorak.

Naprimjer, ukoliko su definirane sljedeće vrijednosti:

$x = [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0]$, $y = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0]$, $frekv = 10$, $fuzor = 5$,

potrebno je da rezultat izvođenja funkcije budu sljedeće vrijednosti:

$x_{new} = [0, 0.2, 0.4, 0.6, 0.8, 0]$ $y_{new} = [0, 2, 4, 6, 8, 0]$

Kako je frekvencija uzorkovanja *fuzor* dva puta manja od frekvencije *frekv* kojom je prikazan originalni signal, to se uzima svaki drugi uzorak iz nizova x i y . Detaljne informacije o uzorkovanju se mogu naći U PDF materijalima na platformi C2 (poglavlje 4, str 106).

Rješenje:

```
import numpy as np
def sinusoida(pocetak, kraj, frekv):
```

```

x = np.arange(pocetak, kraj+ 1/frekv, 1/frekv)
y = np.sin(x)
return x, y

def uzorkovanje(x,y,fuzor,frekv):
    xnew = []
    ynew = []

    for i in range(0, len(x), int(frekv/fuzor)):
        xnew.append(x[i])

    for i in range(0, len(y), int(frekv/fuzor)):
        ynew.append(y[i])

    return xnew, ynew

```

Nakon implementacije funkcije, potrebno je izvesti programski kod ispod kako bi se dobio prikazani ispis. Osim toga, potrebno je dodati još dva primjera uzorkovanja (jedan primjer bez pojave alijasinga i jedan primjer sa pojavom alijasinga) sa proizvoljno od strane studenta odabranim frekvencijama uzorkovanja *fnovo* i proizvoljno odabranim parametrima *kraj*.

```

from scipy.interpolate import interp1d
import matplotlib.pyplot as plt
from math import pi

[x, y] = sinusoida(0, 20, 1000)
[xnew, ynew] = uzorkovanje(x, y, 1, 1000)

print("Uzorkovanje-bez pojave alijasinga:")

plt.figure(1)
plt.subplot(211)
plt.title('Originalni signal i uzorci')
plt.plot(x, y)
plt.plot(xnew, ynew, 'bo')
plt.figure(2)
plt.title('Rekonstruirani signal jednostavnom linearnom interpolacijom')
plt.plot(xnew, ynew)
plt.plot(xnew, ynew, 'bo')
plt.figure(3)
plt.title('Rekonstruirani signal kubnom interpolacijom')
iy = interp1d(xnew, ynew, kind='cubic')
plt.plot(x, iy(x))
plt.plot(xnew, ynew, 'bo')
plt.tight_layout()
plt.show()

```

```

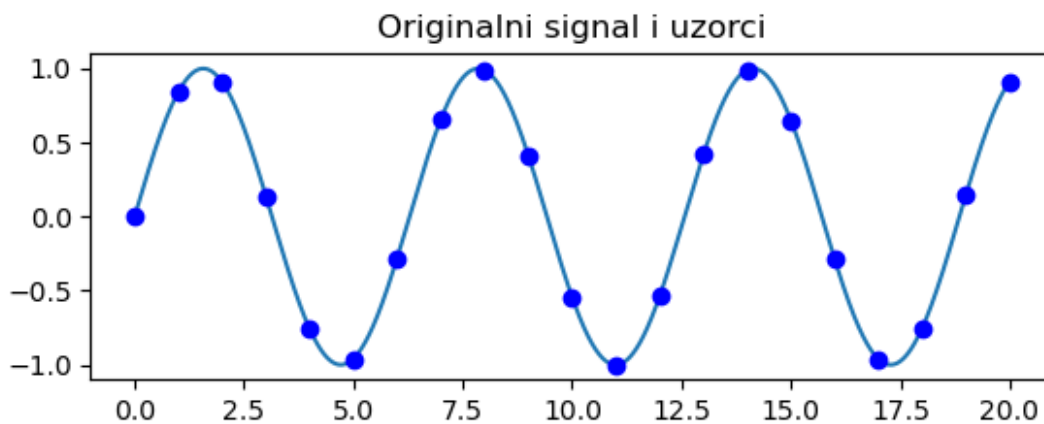
[x, y] = sinusoida(0, 50, 1000)
[xnew, ynew] = uzorkovanje(x, y, 0.1, 1000)

print("Uzorkovanje-sa pojavom alijasinga:")

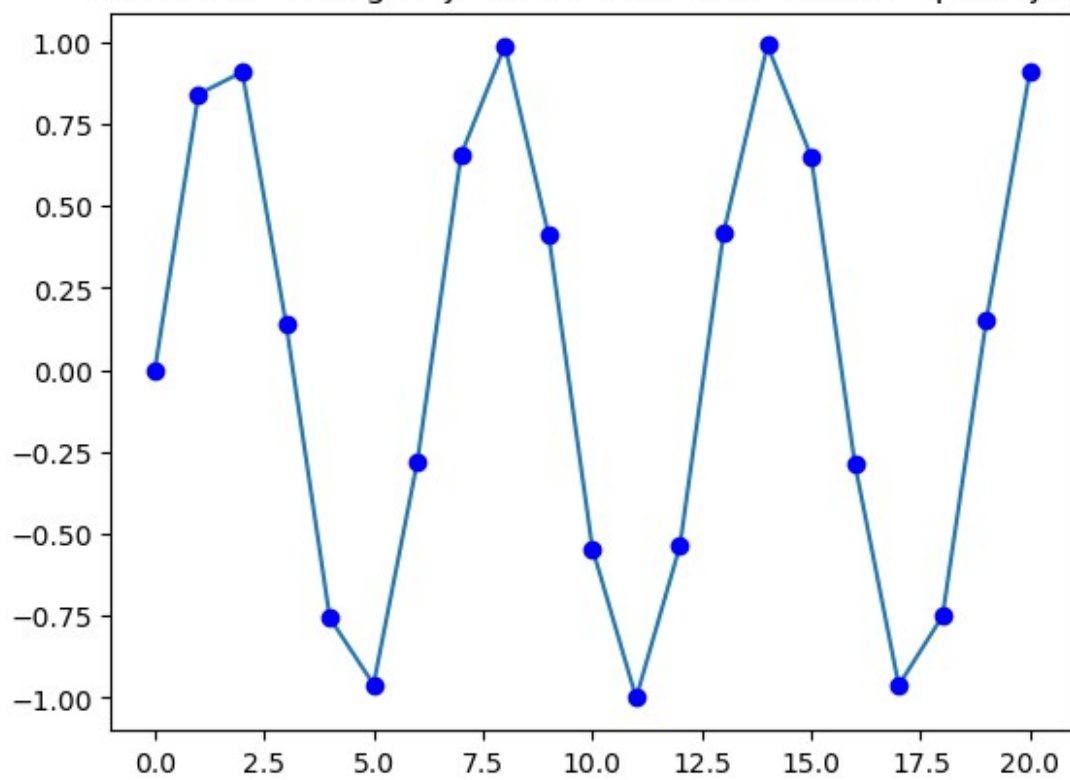
plt.figure(1)
plt.title('Originalni signal i uzorci')
plt.plot(x, y)
plt.plot(xnew, ynew, 'bo')
plt.figure(2)
plt.title('Rekonstruirani signal jednostavnom linearnom
interpolacijom')
plt.plot(xnew, ynew)
plt.plot(xnew, ynew, 'bo')
plt.figure(3)
plt.title('Rekonstruirani signal kubnom interpolacijom')
iy = interp1d(xnew, ynew, kind='cubic')
plt.plot(x, iy(x))
plt.plot(xnew, ynew, 'bo')
plt.tight_layout()
plt.show()

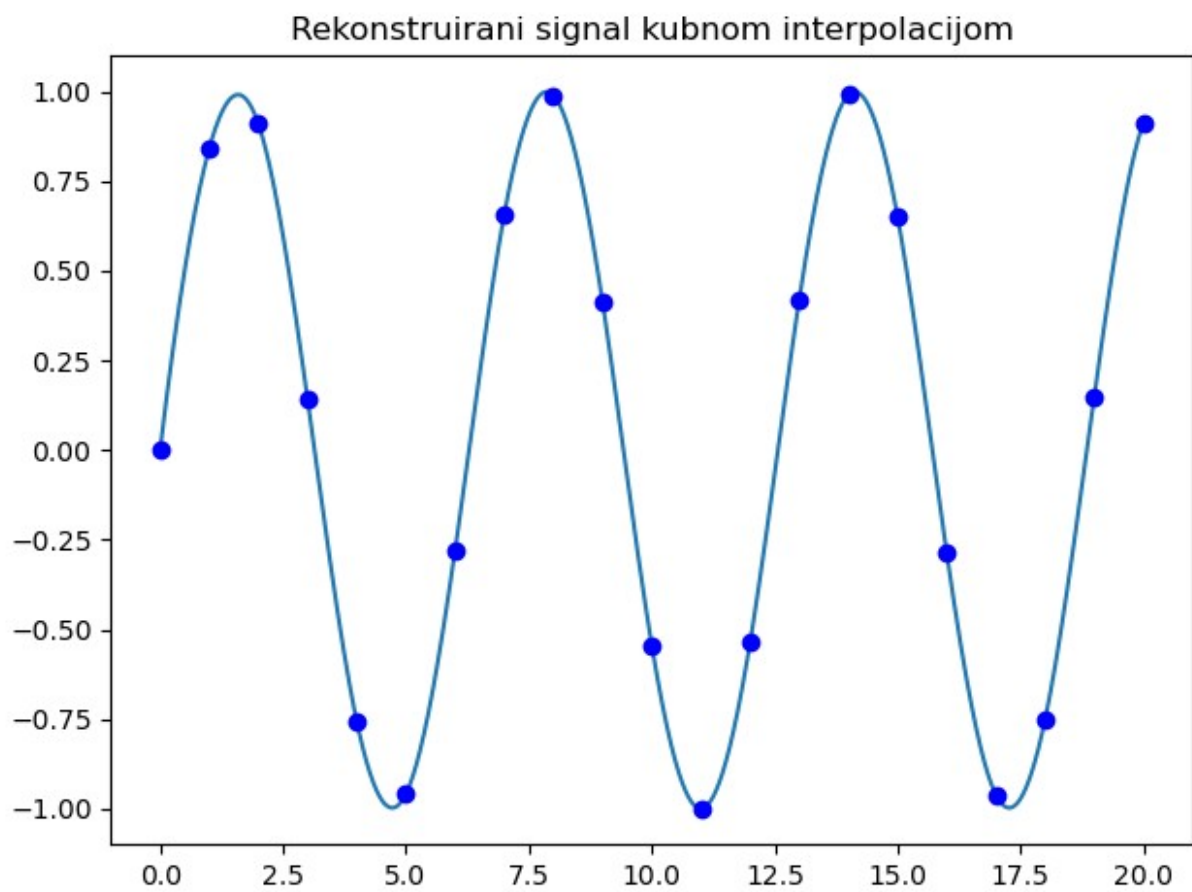
```

Uzorkovanje-bez pojave alijasinga:



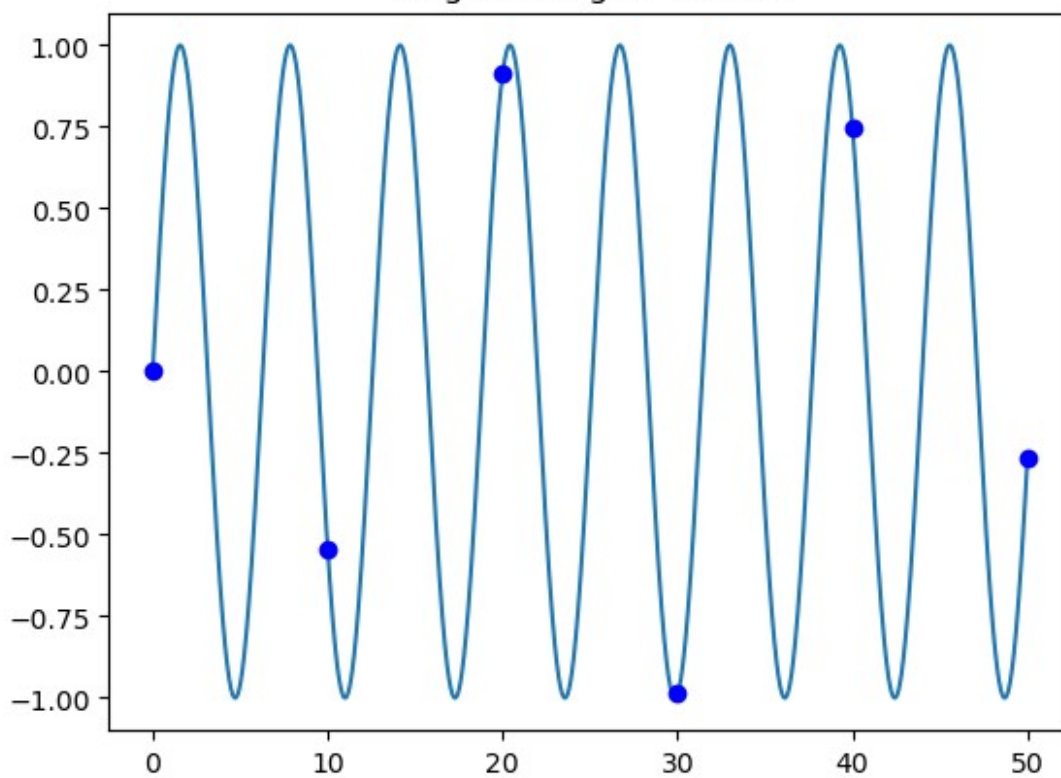
Rekonstruirani signal jednostavnom linearnom interpolacijom



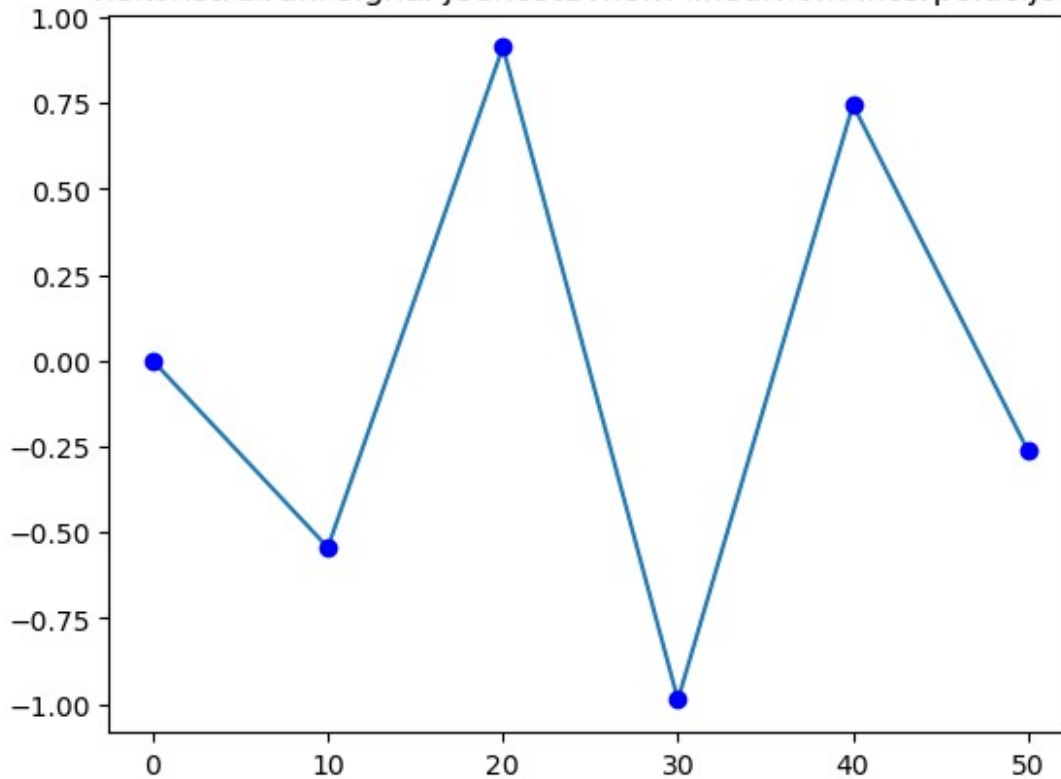


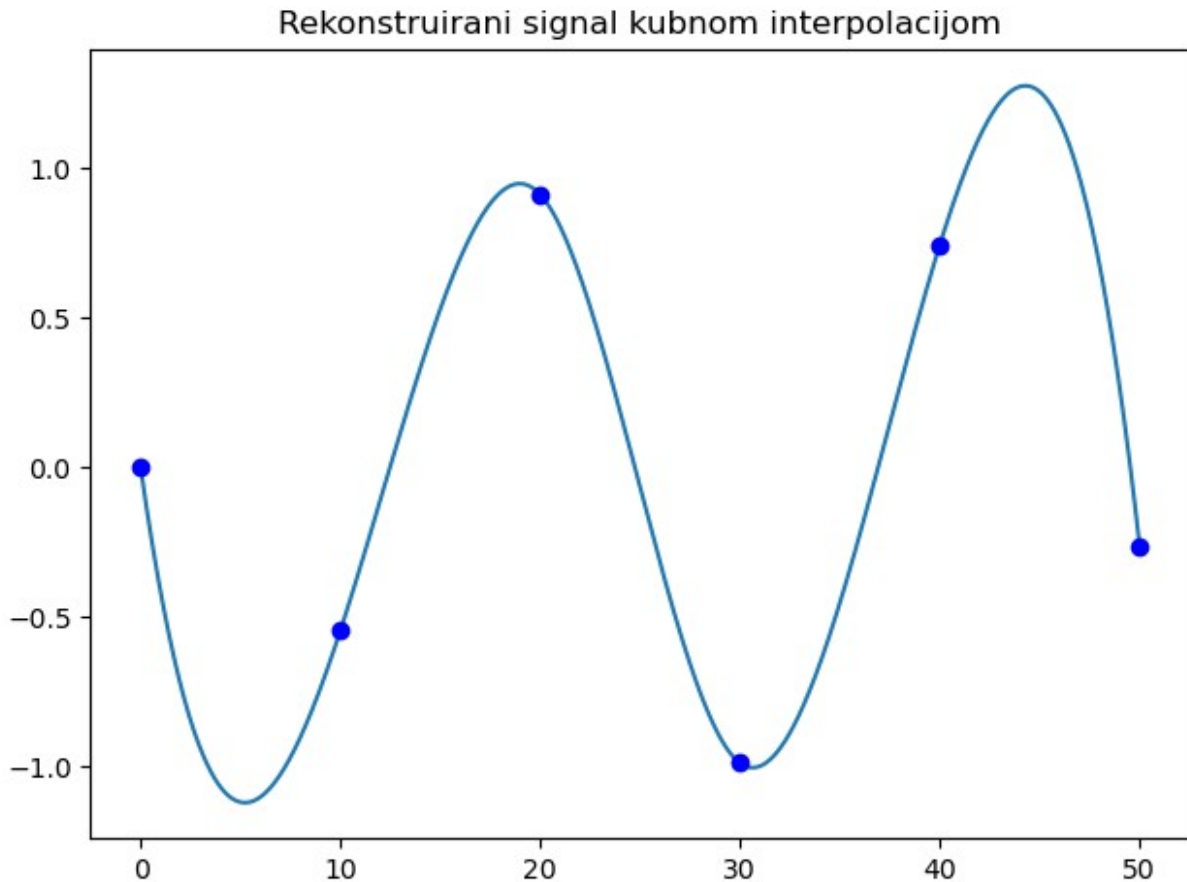
Uzorkovanje-sa pojavom alijasinga:

Originalni signal i uzorci



Rekonstruirani signal jednostavnom linearnom interpolacijom





```
#Testovi
```

```
from scipy.interpolate import interp1d
import matplotlib.pyplot as plt
from math import pi
```

```
[x, y] = sinusoida(0, 2*pi, 2000)
[xnew, ynew] = uzorkovanje(x, y, 1, 2000)
```

```
print("Uzorkovanje-bez pojave alijasinga:")
```

```
plt.figure(1)
plt.subplot(211)
plt.title('Originalni signal i uzorci')
plt.plot(x, y)
plt.plot(xnew, ynew, 'bo')
plt.figure(2)
plt.title('Rekonstruirani signal jednostavnom linearnom
interpolacijom')
plt.plot(xnew, ynew)
plt.plot(xnew, ynew, 'bo')
plt.figure(3)
```

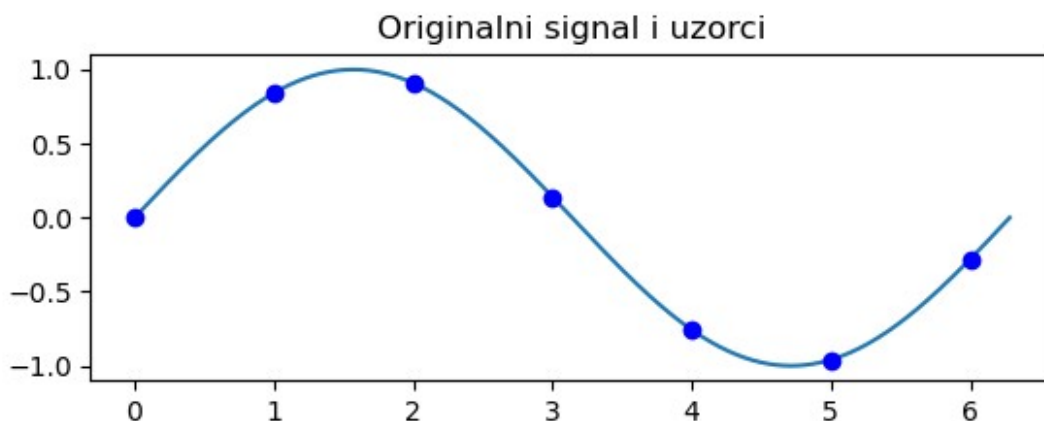
```
plt.title('Rekonstruirani signal kubnom interpolacijom')
iy = interp1d(xnew, ynew, kind='cubic', bounds_error=False,
fill_value="extrapolate")
plt.plot(x, iy(x))
plt.plot(xnew, ynew, 'bo')
plt.tight_layout()
plt.show()
```

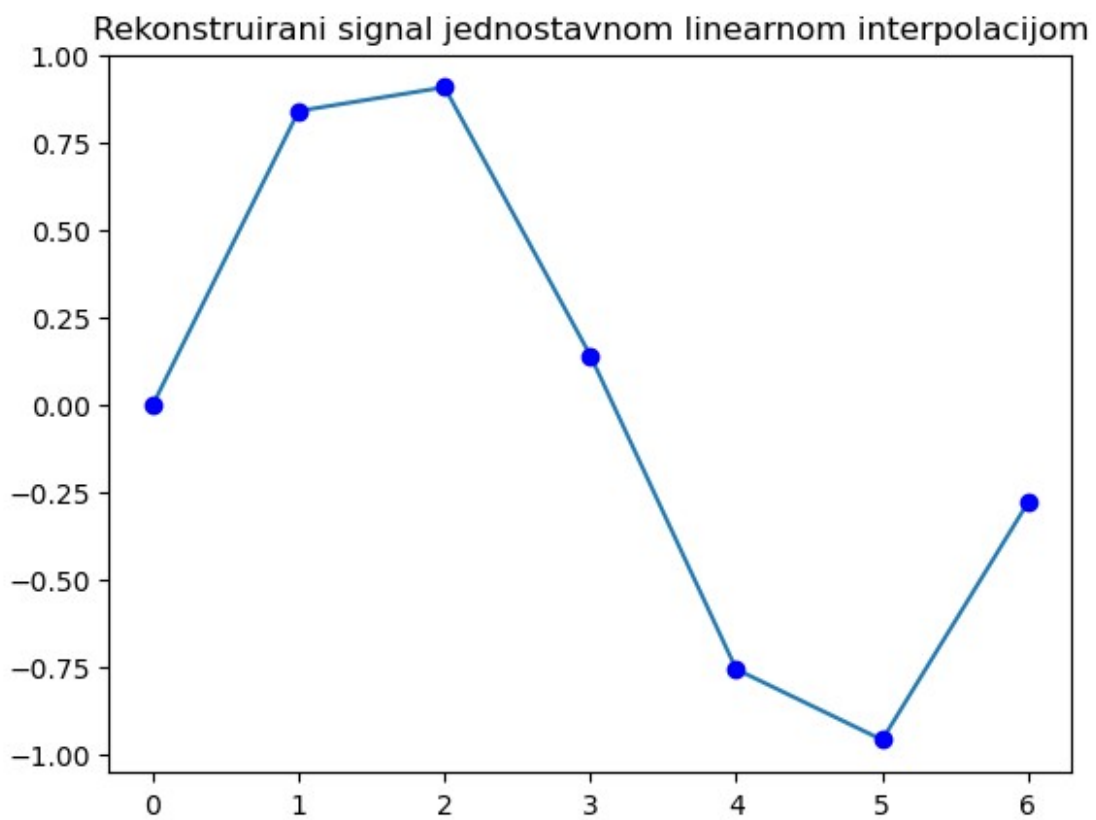
```
[x, y] = sinusoida(0, 100, 100)
[xnew, ynew] = uzorkovanje(x, y, 1, 1000)
```

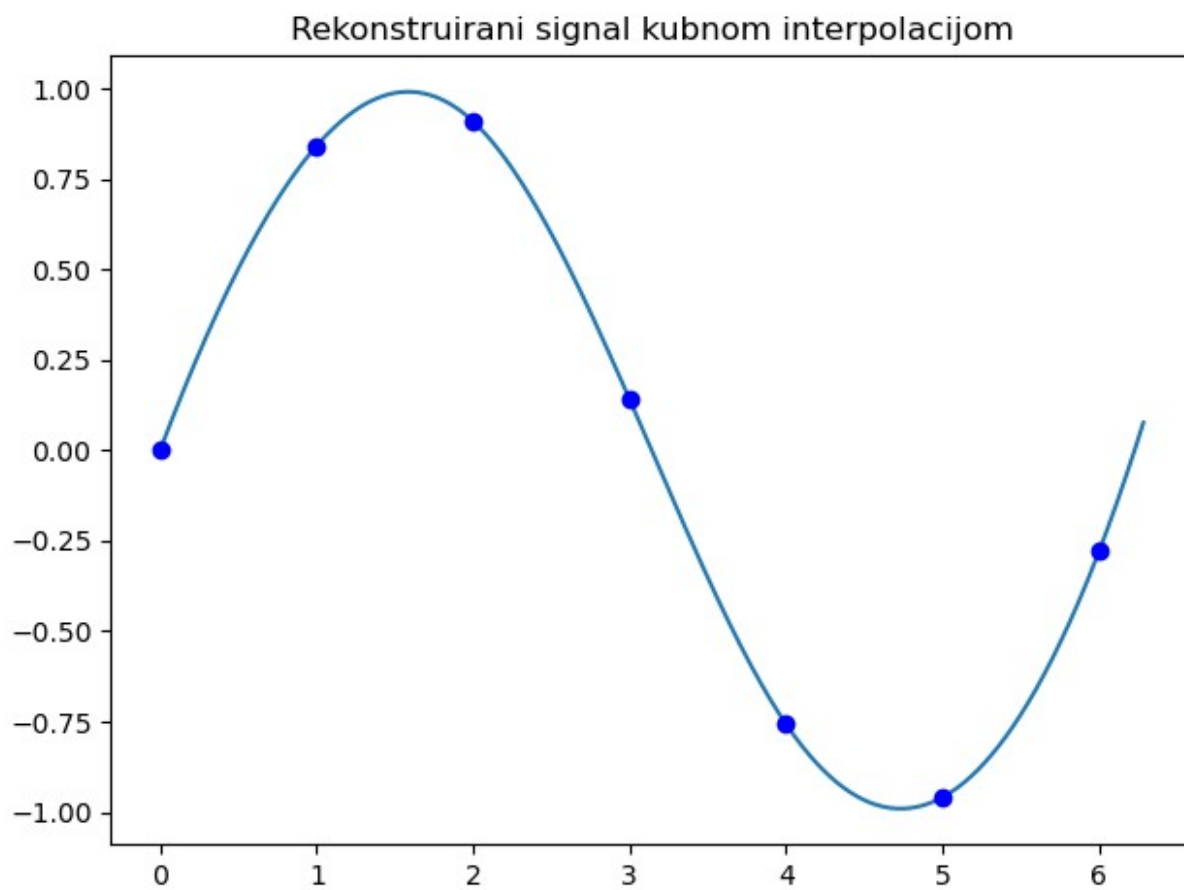
```
print("Uzorkovanje-sa pojavom alijasinga:")
```

```
plt.figure(1)
plt.title('Originalni signal i uzorci')
plt.plot(x, y)
plt.plot(xnew, ynew, 'bo')
plt.figure(2)
plt.title('Rekonstruirani signal jednostavnom linearnom
interpolacijom')
plt.plot(xnew, ynew)
plt.plot(xnew, ynew, 'bo')
plt.figure(3)
plt.title('Rekonstruirani signal kubnom interpolacijom')
iy = interp1d(xnew, ynew, kind='cubic')
plt.plot(x, iy(x))
plt.plot(xnew, ynew, 'bo')
plt.tight_layout()
plt.show()
```

Uzorkovanje-bez pojave alijasinga:

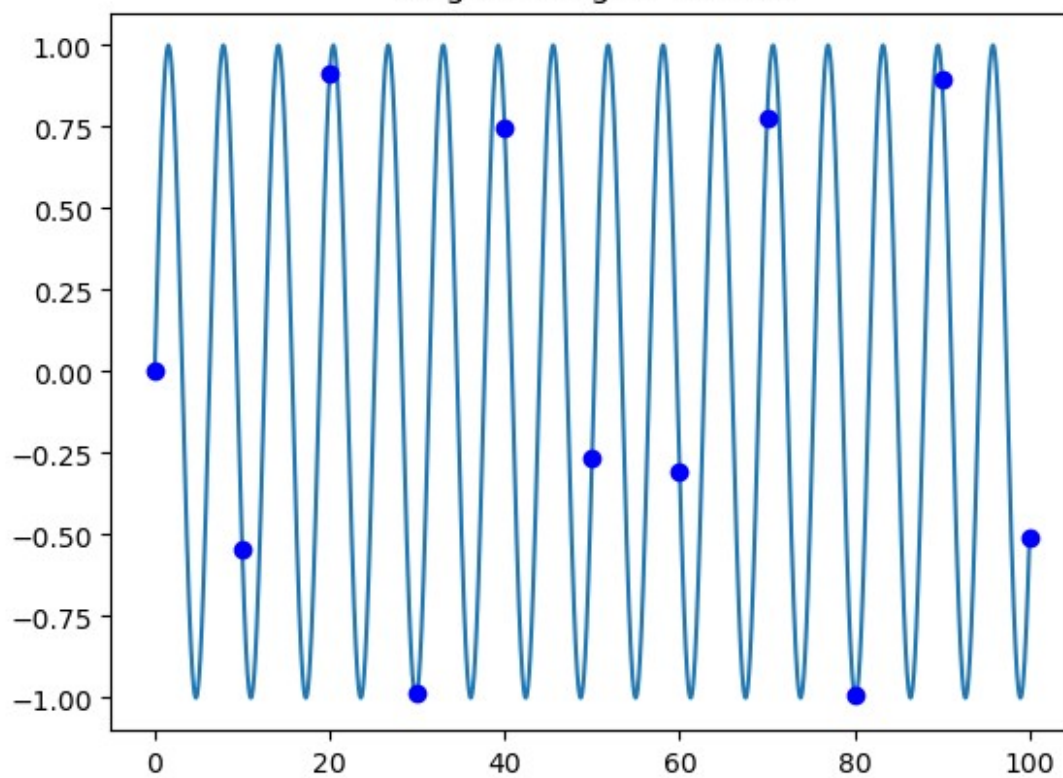




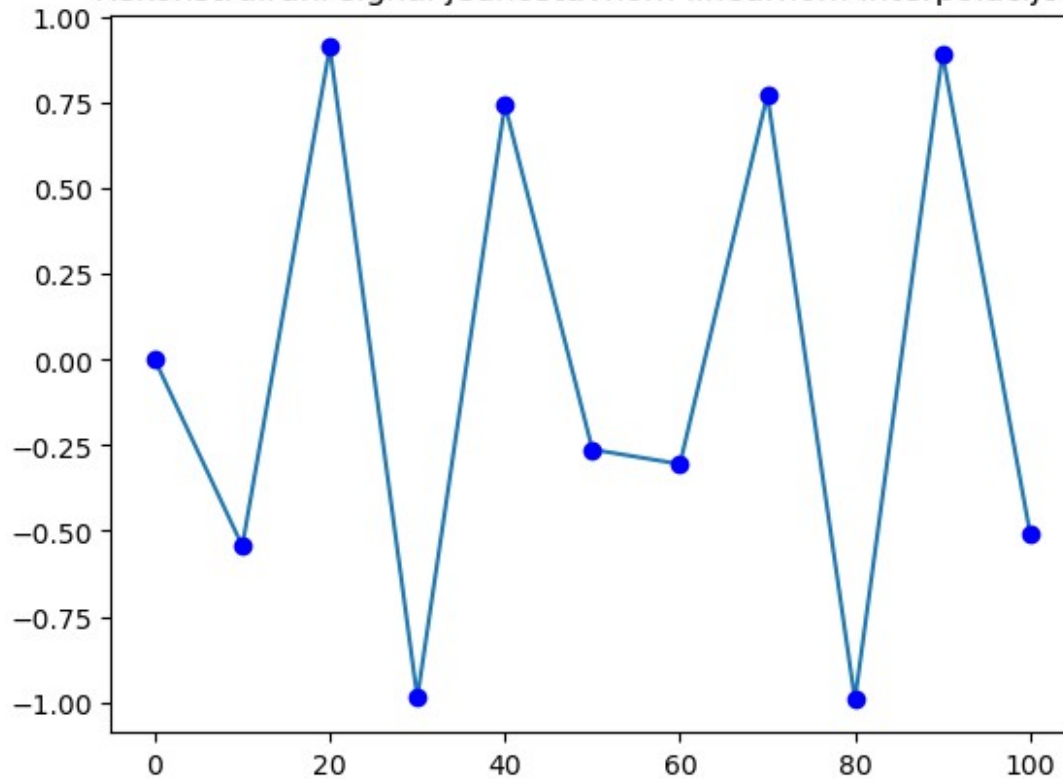


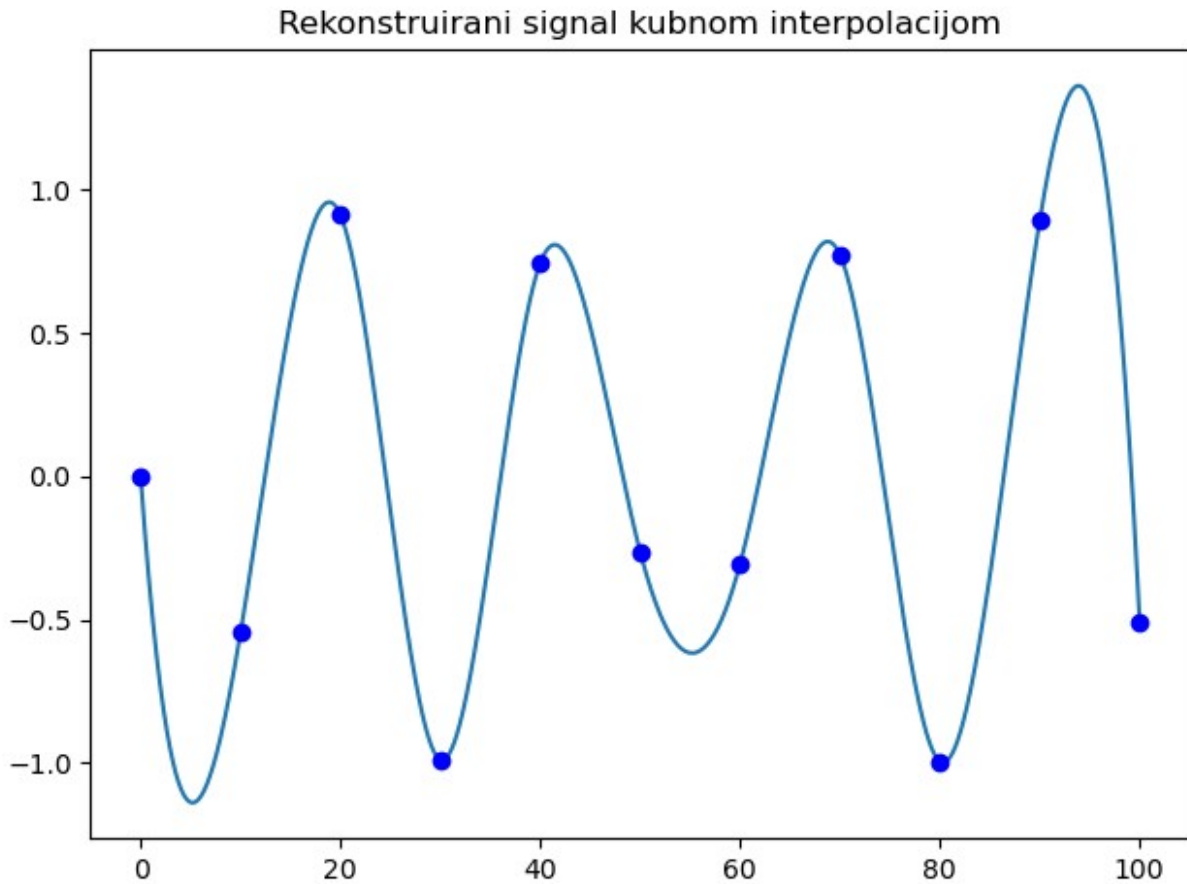
Uzorkovanje-sa pojavom alijasinga:

Originalni signal i uzorci



Rekonstruirani signal jednostavnom linearnom interpolacijom





Zadatak 2.

Potrebno je implementirati funkciju $kvantizacija(y, broj_nivoaa)$ koja izvodi kvantizaciju niza vrijednosti y na broj kvantizacijskih nivoa $broj_nivoaa$. Ova funkcija kao rezultat treba vratiti novi niz kvantiziranih vrijednosti.

Izračunavanje vrijednosti svakog nivoa se izvodi prema sljedećoj formuli:

$$r = \frac{\max(y) - \min(y)}{n - 1}$$

$$q_i = q_{i-1} + r, q_0 = \min(y)$$

pri čemu r predstavlja rezoluciju tj. korak kvantizacije, a q_i predstavlja vrijednost kvantizacijskog nivoa na poziciji i , pri čemu je n broj kvantizacijskih nivoa.

Nakon određivanja svih kvantizacijskih nivoa, svakoj vrijednosti y potrebno je dodijeliti vrijednost onog kvantizacijskog nivoa koji je najbliži stvarnoj vrijednosti y .

Naprimjer, ukoliko su definisane sljedeće vrijednosti:

$y = [0, 0.2, 0.4, 0.6, 0.8, 1]$, $n = 3$,

potrebno je da rezultat funkcije budu sljedeće vrijednosti:

$y = [0, 0, 0.5, 0.5, 1, 1]$,

jer je rezolucija jednaka 0.5 za 3 kvantizacijska nivoa između 0 i 1, a kvantizacijski nivoi su 0, 0.5 i 1.

Detaljne informacije o kvantizaciji se mogu naći U PDF materijalima na platformi C2 (poglavlje 4, str 106).

Rješenje:

```
def najdi_najblizi(niz, broj):
    najblizi = niz[0]
    for i in range(1, len(niz)):
        if abs(niz[i] - broj) < abs(najblizi - broj):
            najblizi = niz[i]
    return najblizi

def kvantizacija(y, broj_nivoa):
    r = (max(y) - min(y)) / (broj_nivoa - 1)

    q = []
    q.append(min(y))

    for i in range(1, broj_nivoa):
        q.append(q[i-1] + r)

    ynew = []
    for i in range(0, len(y)):
        najblizi_nivo = najdi_najblizi(q, y[i])
        ynew.append(najblizi_nivo)

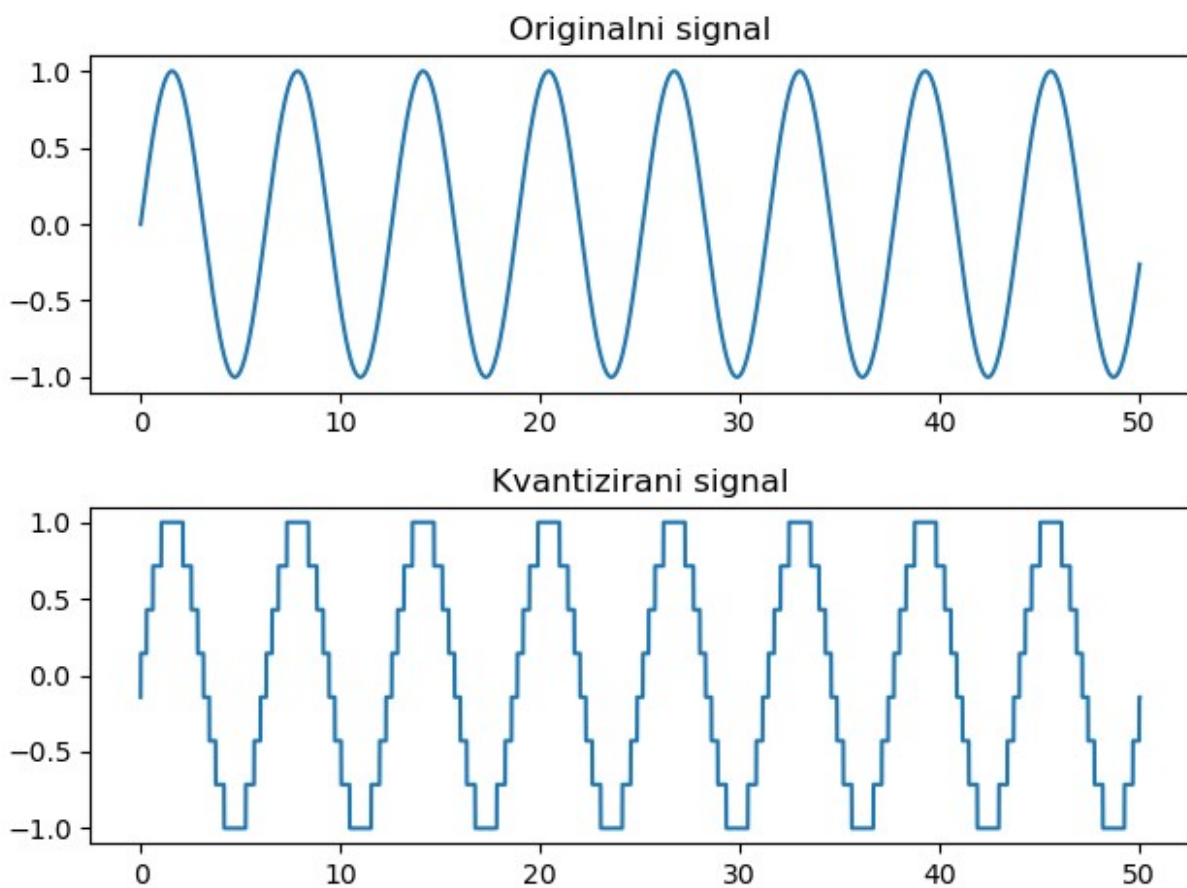
    return ynew
```

Nakon implementacije funkcije, potrebno je izvesti programski kod ispod kako bi se dobio prikazani ispis. Osim toga, potrebno je dodati još dva primjera kvantizacije sa proizvoljnim brojem kvantizacijskih nivoa (jedan primjer gdje je *broj_nivoa* > 8 i jedan primjer gdje je *broj_nivoa* < 8).

```
import matplotlib.pyplot as plt
```

```
[x, y] = sinusoida(0, 50, 1000)
ynew = kvantizacija(y, 8)

plt.figure(1)
plt.subplot(211)
plt.title('Originalni signal')
plt.plot(x, y)
plt.subplot(212)
plt.title('Kvantizirani signal')
plt.plot(x, ynew)
plt.tight_layout()
plt.show()
```



#Testovi

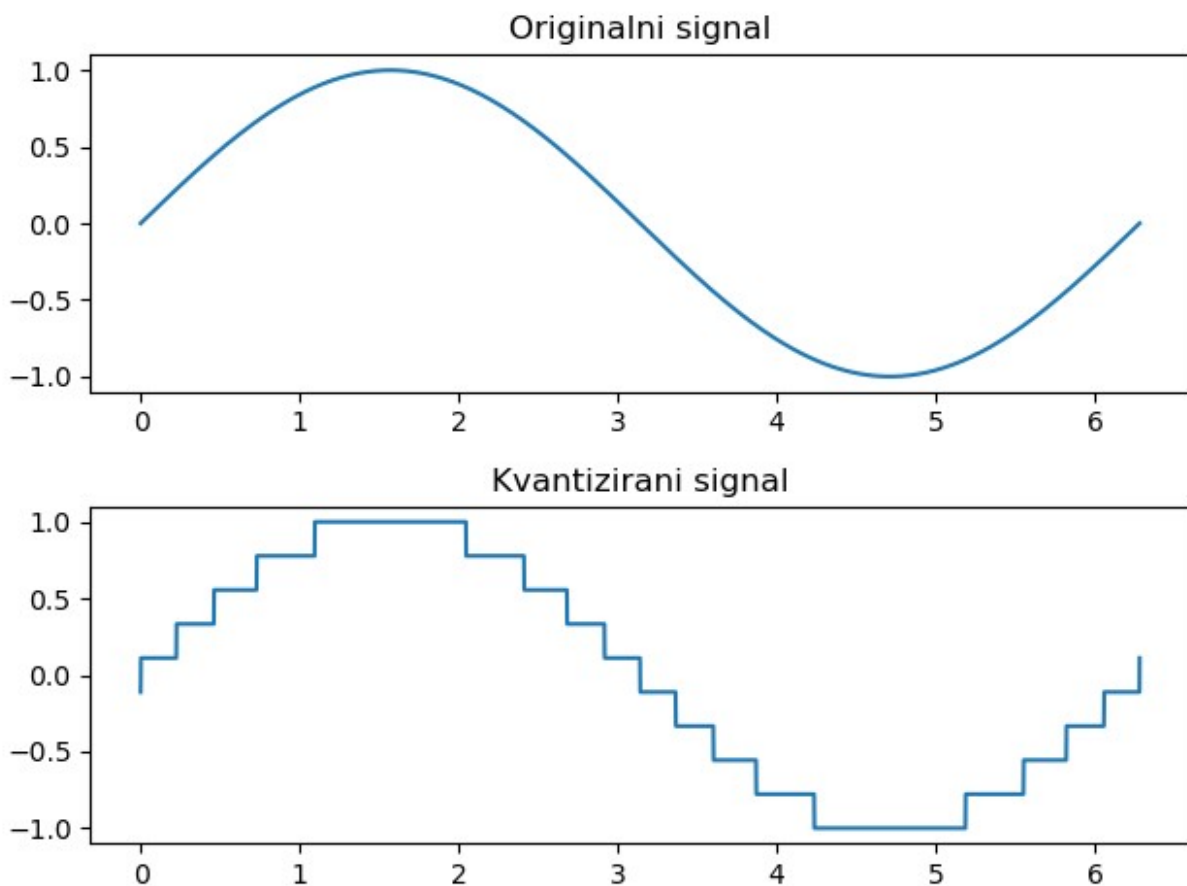
```
[x, y] = sinusoida(0, 2*pi, 1000)
ynew = kvantizacija(y, 10)

plt.figure(1)
plt.subplot(211)
plt.title('Originalni signal')
plt.plot(x, y)
```

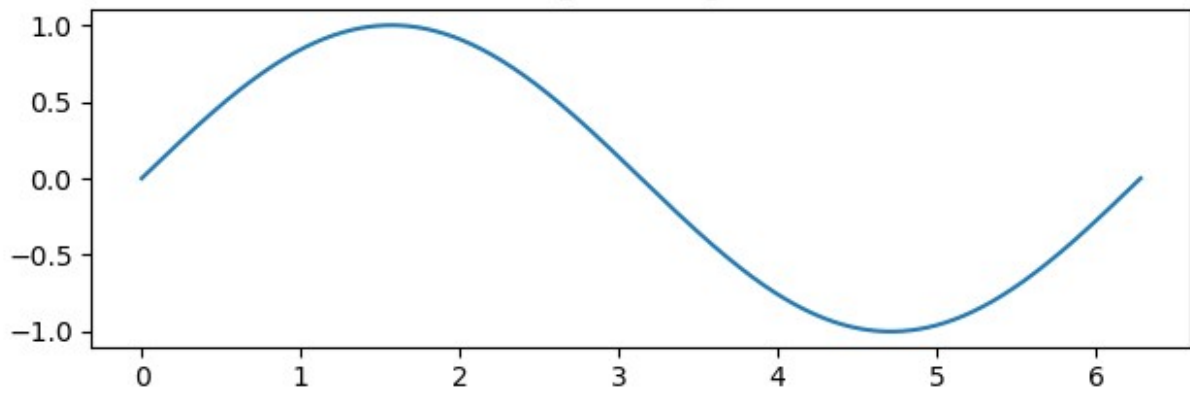
```
plt.subplot(212)
plt.title('Kvantizirani signal')
plt.plot(x, ynew)
plt.tight_layout()
plt.show()
```

```
[x, y] = sinusoida(0, 2*pi, 1000)
ynew = kvantizacija(y, 5)
```

```
plt.figure(1)
plt.subplot(211)
plt.title('Originalni signal')
plt.plot(x, y)
plt.subplot(212)
plt.title('Kvantizirani signal')
plt.plot(x, ynew)
plt.tight_layout()
plt.show()
```



Originalni signal



Kvantizirani signal

