

Laboratorijska vježba 2

Transformacija diskretnih signala u vremenu

Pomjeranje signala u vremenu

Kao i kod kontinualnih signala, mnoge operacije nad signalima se mogu predstaviti jednostavnom modifikacijom vremenske promjenljive u analitičkim izrazima za signale. Signal $x[n - n_0]$ nije ništa drugo nego pomjeranje signala $x[n]$ za n_0 uzoraka. Ukoliko je $n_0 > 0$, to je zapravo kašnjenje signala (pomjeranje u desno), a ukoliko je $n_0 < 0$, to je takozvano prednjačenje signala (pomjeranje u lijevo).

Inverzija vremena

Inverzija signala $x[n]$ u vremenu, analitički predstavljeno kao $x[-n]$, predstavlja "sliku u ogledalu" signala $x[n]$.

Skaliranje amplitude

Skaliranje signala po amplitudi se postiže tako što se amplituda signala pomnoži konstantom A , tako da se za diskretni signal dobija $x[n]=Ax[n]$.

Skaliranje vremena

Skaliranje signala $x[n]$ u vremenu, analitički predstavljeno kao $x[an]$ predstavlja skaliranje signala $x[n]$ vremenskom konstantom a .

- za $a > 1$ funkcija se sužava duž vremenske ose (ubrzavanje)
- za $0 < a < 1$ funkcija se širi duž vremenske ose (usporavanje)
- za $a < 0$ funkcija se rotira oko y-ose (vrijeme "teče unazad")

Zadatak 2.1

Nacrtati signal $x[n]$, a zatim nacrtati signal $x[3n]$.

```
1 # -*- coding: utf-8 -*-
2 """
3 Nacrtati signal x[n], a zatim nacrtati signal x[3n].
4 """
5
6 import numpy as np
7 import matplotlib.pyplot as plt
8
9 n=np.arange(-10,10,1)
10 n1=3*n
11
12 x=n
13 y=n
```

```

14 plt.xlabel('n')
15 plt.ylabel('x[n]')
16
17 plt.stem(n,x)
18 plt.show()
19
20 plt.xlabel('n')
21 plt.ylabel('x[3n]')
22
23 plt.stem(n1,y)#ne elimo pove ati amplitudu nego interval pa slijedi (-30,30,1)
24 plt.show()
25
26
27 #za a>1 funkcija se suava du vremenske ose (ubrzavanje)

```

Zadatak 2.2

Koristeći programski jezik Python:

- nacrtati signal $x[n] = e^n$
- invertovati signal pod a)
- zakasniti originalni signal za 2 uzorka
- uprednjačiti originalni signal za 2 uzorka

```

1 # -*- coding: utf-8 -*-
2 """
3 Koriste i programski jezik Python:
4     a)nacrtati signal x[n] = e^n
5     b)invertovati signal pod a)
6     c)zakasniti originalni signal za 2 uzorka
7     d)uprednja iti originalni signal za 2 uzorka
8 """
9 import numpy as np
10 import matplotlib.pyplot as plt
11
12 n=np.arange(-15,5,1) #interval ide od [-5,5) -> ne uklju uju i posljednju
13                        #vrijednost
14
15 #posmatra se da je u x[-5]=e^-5*1,...
16 #a)
17 plt.grid()
18 plt.xlabel('n')
19 plt.ylabel('x[n]=e^n')
20 plt.title('a) signal x[n]=e^n ')
21 plt.stem(n,np.exp(n))
22 plt.show()
23
24 #b)
25 n_invert=n[::-1] #invertovanje niza n, posmatra se da je u x[-5]=e^-5*-1,...
26
27 plt.grid()
28 plt.xlabel('n')
29 plt.ylabel('x[-n]')
30 plt.title('b) invertovani signal')
31 plt.stem(n,np.exp(n_invert))
32 plt.show()
33
34 #c)
35 pomak=2 #pomicanje niza, pomak n0>0 -> ka njenje
36 x=np.exp(n)
37 x_pomak=np.roll(x,pomak) #roll(x,shift,axis=None)
38 x_pomak[0:pomak]=0 #ako bismo izostavili ovu komandu,
39 #sekvencabi se pomicala udesno, odnosno po to sekvencaside do 5

```

```

40 #(ne uklju uju i 5), nastavila bi pomak sa po etka ose, tj. od -5
41
42 plt.grid()
43 plt.xlabel('n')
44 plt.ylabel('x[n-2]')
45 plt.title('c) zaka njen originalni signal za 2 uzorka ')
46 plt.stem(n,x_pomak)
47 plt.show()
48
49 #d)
50 pomak=-2 #pomjeranje niza, pomjeraj n0<0 -> prednja enje
51 x=np.exp(n)
52 x_pomak=np.roll(x,pomak)
53
54
55 plt.grid()
56 plt.xlabel('n')
57 plt.ylabel('x[n+2]')
58 plt.title('d) prednja enje originalni signal za 2 uzorka ')
59 plt.stem(n,x_pomak)
60 plt.show()

```

Zadatak 2.3

Koristeći programski jezik Python nacrtati signal $x[n]=n*(x[n]-x[n-4])$, zatim je potrebno predstaviti:

- a) negativnu vrijednost signala
- b) invertovati signal

```

1 # -*- coding: utf-8 -*-
2 """
3 Koriste i Python programski jezik nacrtati signal x[n]=n*(x[n]-x[n-4])
4 """
5
6
7 import numpy as np
8 import matplotlib.pyplot as plt
9
10 n=np.arange(-5,5,1)
11 x=np.heaviside(n,1)
12
13 z=n*x
14
15 plt.stem(n,z)
16 plt.show()
17
18 """
19 Koriste i Python programski jezik nacrtati signal x[n]=x[n]-x[n-4]
20 """
21
22 import numpy as np
23 import matplotlib.pyplot as plt
24
25 n=np.arange(-5,5,1)
26 x=np.heaviside(n,1)
27
28 plt.stem(n,x)
29 plt.show()
30
31 # negativna vrijednost signala x[n]
32 x1=-x
33 plt.stem(n,x1)
34 plt.show()

```

```

35 """
36 Koriste i Python programski jezik nacrtati invertovat signal
37 x[n]=x[n]-x[n-4]
38 """
39
40 n=np.arange(-5,5,1)
41 x=np.heaviside(-n,1)
42
43 plt.stem(n,x)
44 plt.show()

```

Zadatak 2.4

Koristeći programski jezik Python nacrtati signal $x[n] = e^n * (x[n - 4] - x[n - 6])$, zatim je potrebno predstaviti:

- invertovati signal
- zakašnjen signal za 2 pomaka

```

1 # -*- coding: utf-8 -*-
2 """
3 Koriste i Python programski jezik nacrtati signal x[n]=e^n*(x[n-4]-x[n-6])
4 """
5
6
7 import numpy as np
8 import matplotlib.pyplot as plt
9
10 n = np.arange(-2,7,1)
11 x = np.exp(n)
12 y = np.heaviside(n-4, 1)
13
14 z=x*y
15
16 plt.stem(n,z)
17 plt.show()
18
19 """
20 Koriste i Python programski jezik nacrtati invertovan signal x[n] ->
21 zatim ga zakasniti za 2.
22 """
23
24 import numpy as np
25 import matplotlib.pyplot as plt
26
27 n = np.arange(-2,7,1)
28 x = np.exp(n)
29 y = np.heaviside(n-4, 1)
30
31 z=x*y
32
33 #invertovan
34 plt.stem(-n,z)
35 plt.show()
36
37 #invertovan pa pomjeren za 2 koraka desno
38 plt.stem(-n+2,z)
39 plt.show()

```