

Laboratorijska vježba 1

Osobine diskretnih signala

Diskretni signal, za razliku od kontinualnih signala, da bi mogao da se "obrađuje" u računarskim analizama, jednoznačno je definisan u diskretnim vremenskim trenucima. Ovi trenuci, nastali su odabiranjem analognih signala u jednakim vremenskim intervalima. U slučaju da je amplituda diskretnog signala kvantovana, možemo reći da se radi o digitalnim signalima. Iste osobine koje važe za analogne, važe i za digitalne, sa razlikom što digitalne signale označavamo sa $x[n]$, gdje $n \in \mathbb{Z}$ i zapisujemo kao sekvencu brojeva, odnosno funkcije diskretne promjenljive.

Elementarni diskretni signali

Odskočna ili Hevisajdova funkcija

Odskočna ili hevisajdova funkcija uobičajno se obilježava i definiše na sljedeći način:

$$x[n] = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

Zadatak 1.1

Koristeći programski jezik Python, nacrtati diskretnu Hevisajdovu funkciju $u[n]$ na vremenskoj osi na intervalu $(-10,10)$, zatim nacrtati signale:

a) $x[n] = 2u[n-4]$

b) $x[n] = -0.5u[n+3]$

```
1 # -*- coding: utf-8 -*-
2 """
3 Koriste i programski jezik Python, nacrtati diskretnu
4 Hevisajdovu funkciju na vremenskoj osi na intervalu (-10,10)
5 """
6
7 import numpy as np
8 import matplotlib.pyplot as plt
9 """
10 x[n]=u[n]
11 """
12
13 n=np.arange(-7,7,1)
14 x=1.*(n>=0)
15
16 plt.stem(n,x)
17
18
19 """
```

```

20 x[n]=2u[n-4]
21 """
22 plt.figure(2)
23 x=2.*(n>=4)
24
25 plt.stem(n,x)
26
27 """
28 x[n]=-0.5u[n+3]
29 """
30 plt.figure(3)
31 x=-0.5*(n>=-3)
32
33 plt.stem(n,x)
34
35 #-----
36 # II na in
37 #-----
38 import numpy as np
39 import matplotlib.pyplot as plt
40
41 def stepfun(t,n): #t=n, a n=0 za u[n]
42     x=np.zeros(len(t)) #sve ce biti nule za duzinu
43     l=0;
44     while l<len(t):
45         if t[l]<n:
46             x[l]=0
47         else:
48             x[l]=1
49         l=l+1
50     return x
51
52 n = np.arange(-7,7,1)
53
54 """
55 x[n]=u[n]
56 """
57 plt.figure(4)
58 x1=stepfun(n, 0) #t je ustvari n
59 plt.stem(n,x1) #stem koristimo najcesce za diskretne signale
60 plt.xlabel("n")
61 plt.ylabel("x[n]")
62 plt.title("x[n]=u[n]")
63 plt.grid()
64
65
66 """
67 x[n]=2u[n-4]
68 """
69 plt.figure(5)
70 x1=2*stepfun(n, 4) #t je ustvari n
71 plt.stem(n,x1) #stem koristimo najcesce za diskretne signale
72 plt.xlabel("n")
73 plt.ylabel("x[n]")
74 plt.title("x[n]=u[n]")
75 plt.grid()
76
77
78 """
79 x[n]=-0.5u[n+3]
80 """
81 plt.figure(6)
82 x1=-0.5*stepfun(n, -3) #t je ustvari n
83 plt.stem(n,x1) #stem koristimo najcesce za diskretne signale
84 plt.xlabel("n")
85 plt.ylabel("x[n]")
86 plt.title("x[n]=u[n]")
87 plt.grid()

```

Jedinična impulsna ili Dirakova funkcija

Jedinična ili Dirakova funkcija definiše se na sljedeći način:

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$

Zadatak 1.2

Koristeći programski jezik Python, kreirati i nacrtati jedinični Dirakov impuls $\delta[n-1]$ na vremenskoj osi u rasponu (0,10), zatim nacrtati signal $x(t)=\delta[t-2]$

```

1 # -*- coding: utf-8 -*-
2 """
3 Koriste i programski jezik Python, kreirati i nacrtati jedini ni Dirakov impuls
4 delta[n-1] na vremenskoj osi u rasponu (0,10).
5 """
6
7 import numpy as np
8 import matplotlib.pyplot as plt
9 # =====
10 # x(t)=delta(t)
11 # =====
12 x=np.zeros(10) #ne prikazuje negativne vrijednosti
13 x[0]=1
14
15 plt.stem(x)
16 # =====
17 # x(t)=delta(t-2)
18 # =====
19 x=np.zeros(10) #ne prikazuje negativne vrijednosti
20 x[2]=1
21
22 plt.figure(2)
23 plt.stem(x)
24
25 #-----
26 # II na in
27 #-----
28 import numpy as np
29 import matplotlib.pyplot as plt
30
31 def stepfun(t,n):
32     x=np.zeros(len(t))
33     l=0;
34     while l<len(t):
35         if t[l]<n:
36             x[l]=0
37         else:
38             x[l]=1
39         l=l+1
40     return x
41
42 t=np.linspace(-5,5,1000)
43
44 # =====
45 # x(t)=delta(t)
46 # =====
47 #step = 1 #ovo nije moglo jer bi to znacilo da se radi o step funkciji
48 step=t[1]-t[0] #koristimo ovako jer se radi o nizovima i treba nam samo jedan
49 #impuls
50 x1=stepfun(t,0) #u(t)

```

```

51 x2=stepfun(t,step) #u(t-1)
52
53 # x1=stepfun(t,0)
54 # x2=stepfun(t,1)
55 plt.figure(3)
56
57 plt.plot(t,x1-x2) #u(t)-u(t-1)
58 plt.xlabel("t")
59 plt.ylabel("x(t)")
60 plt.title("x(t) = delta(t)")
61 plt.grid()
62
63 # =====
64 # x(t)=delta(t-2)
65 # =====
66
67 plt.figure(4)
68 x1=stepfun(t,2)
69 x2=stepfun(t,2.01)
70
71 plt.plot(t,x1-x2) #u(t-2)-u(t-2.01) povorka impulsa od 2 - povorka impulsa od
72 # 2.01
73 plt.xlabel("t")
74 plt.ylabel("x(t)")
75 plt.title("x(t) = delta(t-2)")
76 plt.grid()

```

Zadatak 1.3

Koristeći programski jezik Python, nacrtati funkciju $x[n] = \delta[n] + 3\delta[n-2]$ na vremenskoj osi u rasponu $(-10,10)$.

```

1 # -*- coding: utf-8 -*-
2 """
3 Koriste i programski jezik Python, nacrtati funkciju
4 x[n]=delta[n]+3delta[n-2] na vremenskoj osi u rasponu (-10,10).
5 """
6
7 import numpy as np
8 import matplotlib.pyplot as plt
9
10 #n=np.arange(-10,10,1)
11 x=np.zeros(10)
12
13 x[0]=1
14 x[2]=3
15
16 plt.stem(x)
17 plt.show()
18
19 # =====
20 # Napisati kod za drugi na in
21 # =====

```

Zadatak 1.4

Koristeći programski jezik Python nacrtati diskretni signal AWGN srednje vrijednosti $\mu = 0$ i standardne devijacije $\sigma^2 = 1$

```

1 # -*- coding: utf-8 -*-
2 """
3 Koriste i programski jezik Python nacrtati
4 diskretni signal AWGN srednje vrijednosti 0 i standardne
5 devijacije 1
6 """
7
8 import numpy as np
9 import matplotlib.pyplot as plt
10
11 noise=np.random.normal(0,1,100) #modul random implementira generatore
12                                #pseudo-slu ajnih brojeva za razli ite
13                                #raspodjele
14
15 plt.stem(noise)
16 plt.show()

```

U nastavku se nalaze neki dodatni primjeri za vježbu:

Zadatak 1.5

```

1 # -*- coding: utf-8 -*-
2 """
3 Koriste i programski jezik Python, nacrtati eksponencijalnu funkciju.
4 """
5
6 import numpy as np
7 import matplotlib.pyplot as plt
8
9 n=np.arange(-5,5,1) #proizvoljan opseg
10 plt.stem(n,np.exp(n)) #crtamo funkciju x[n]=np.exp(n)

```

Zadatak 1.6

```

1 # -*- coding: utf-8 -*-
2 """
3 Koriste i programski jezik Python, formirati slu ajan signal N=10. tj. signal
4 iji uzorci predstavljaju slu ajno odabrane cijele brojeve iz
5 intervala (-5,10). Koristiti naredbu, random.random_integers(a,b,c)
6 """
7
8 import numpy as np
9 import matplotlib.pyplot as plt
10
11 x=np.random.random_integers(-5,10,20)
12 plt.stem(x)
13 plt.show()

```

Zadatak 1.7

```

1 # -*- coding: utf-8 -*-
2 """
3 Koriste i programski jezik Python, nacrtati diskretni sinusoidalni
4 signal sa periodom uzorkovanja 0.001s ija je frekvencija 5Hz.
5 """
6
7 import numpy as np

```

```
8 import matplotlib.pyplot as plt
9
10 t=np.arange(-50,50,1)
11 x=np.cos(2*np.pi*t*0.05) #primjer periodi nog signala
12
13 plt.stem(t,x)
14 plt.show()
```