

Laboratorijska vježba 5

Z-transformacija

Zadatak 5.1

Korištenjem osobina Z-transformacije kao i tabele Z-transformacija, odrediti Z-transformaciju signala:

$$x(n) = (n-2)(0.5)^{(n-2)} \cos\left[\frac{\pi}{3}(n-2)\right] u(n-2)$$

Rješenje

Analitičkim proračunom dobijamo:

$$X(z) = \frac{0.25z^{-3} - 0.5z^{-4} + 0.0625z^{-5}}{1 - z^{-1} + 0.75z^{-2} - 0.25z^{-3} + 0.0625z^{-4}}$$

```
1 # -*- coding: utf-8 -*-
2 """
3 Korištenjem osobina Z-transformacije kao i tabele Z-transformacija,
4 odrediti Z-transformaciju signala
5 x(n) = (n-2)*(0.5)^(n-2)*cos[pi/3(n-2)]*u(n-2)
6 """
7
8 import numpy as np
9 import sympy
10 from plot_zplane import zplane
11 import scipy.signal
12
13 n=np.arange(0,7,1)
14
15 b = np.array([0,0,0,0.25,-0.5,0.0625]) #nule
16 a = np.array([1,-1,0.75,-0.25,0.0625]) #polovi
17
18 delta=np.heaviside(n,1)-np.heaviside(n-0.01,1)
19
20 #provjeriti sekvencu x1
21 x1=scipy.signal.lfilter(b,a,delta)#provjera na osnovu rješenja
22 x2=((n-2)*((1/2)**(n-2))*np.cos(np.pi*(n-2)/3))*np.heaviside(n-2,1) #na a sekvencu
23
24 #x1 i x2 trebaju biti identični ako je proračun tačan
25
26 zplane(a,b) #crtanje polova i nula u z ravni
```

Zadatak 5.2

Odrediti inverznu Z-transformaciju signala:

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

Rješenje

Rastavljanjem na parcijalne razlomke dobije se
 Odrediti inverznu Z-transformaciju signala:

$$X(z) = \frac{1}{2} \left(\frac{1}{1-z^{-1}} \right) - \frac{1}{2} \left(\frac{1}{1-\frac{1}{3}z^{-1}} \right)$$

```

1 # -*- coding: utf-8 -*-
2 #####
3 #Odrediti inverznu Z transformaciju signala X(z) = z/3z^2 - 4z + 1.
4 #####
5 import numpy as np
6 import sympy
7 from plot_zplane import zplane
8 import scipy.signal
9
10 b = np.array([0,1])
11 a = np.array([3,-4,1])
12
13 print(scipy.signal.residuez(b,a)) # za odre ivanje polova pri rastavljanju datog
    izraza
14
15 zplane(a,b)
16
17 #####
18 # Za diferencijalnu jedna inu:
19 # y[n] - 3.25y[n - 1] = x[n - 1]      0.75y[n - 2] + 3x[n - 2]
20 #a) Odrediti prijenosnu funkciju sistema kori tenjem Z-transformacije.
21 #b) Napisati Matlab kod za crtanje nula i polova prijenosne funkcije.
22 #c) Da li je sistem stabilan?
23 #####
24
25
26 b = np.array([0,1,3])
27 a = np.array([1,-3.25,0.75])
28
29 print(scipy.signal.residuez(b,a)) # za odre ivanje polova pri rastavljanju datog
    izraza
30
31 zplane(a,b)

```

Zadatak 5.3

Neka su date dvije sekvence $x_1(n)$ i $x_2(n)$ čije su Z transformacije $X_1(z) = 2 + 3z^{-1} + 4z^{-2}$ i $X_2(z) = 3 + 4z^{-1} + 5z^{-2} + 6z^{-3}$. Odrediti $X_3(z) = X_1(z)X_2(z)$ korištenjem naredbe *convolve* u programskom jeziku Python. Nakon toga potrebno je odrediti $x_2(n)$ korištenjem naredbe *deconvolve*.

Rješenje

Prema osobina Z transformacije, množenje u z-domenu odgovara konvoluciji u vremenskom domenu. Prema definiciji Z-transformacije, sekvence $x_1(n)$ i $x_2(n)$ se mogu napisati kao: $x_1(n) = \underline{2}, 3, 4$, $x_2(n) = \underline{3}, 4, 5, 6$

Ukoliko želimo da podijelimo jedan polinom sa drugim, onda nam je potrebna operacija dekonvolucije, gdje se računa podjele polinoma b sa polinomom a, te pohranjuje u novi polinom p sa ostatkom r.

```

1 # -*- coding: utf-8 -*-
2 #####
3 #Neka su date dvije sekvence x1(n) i x2(n) ije su Z transformacije
4 #X1(z) = 2 + 3z^(-1) + 4z^(-2) i
5 #X2(z) = 3 + 4z^(-1) + 5z^(-2) + 6z^(-3). Odrediti X3(z) = X1(z)X2(z)
6 #####
7 import numpy as np
8 import scipy.signal as sc

```

```

9
10 x1=np.array([2, 3, 4])
11 x2=np.array([3, 4, 5, 6])
12 x3 = sc.convolve(x1,x2)
13 print(x3)
14
15
16
17 #####
18 #Za prethodni zadatak odrediti x2(n)
19 #####
20 x3 = np.array([6,17,34,43,38,24])
21 x1=np.array([2, 3, 4])
22 x2 = sc.deconvolve(x3,x1)
23 print(x2)

```

Zadatak 5.4

Odrediti Z-transformaciju za signal $y(n)$, ako vrijedi da je $Y(z) = X(z)H(z)$, a $X(z) = z + 2 + 3z^{-1}$ i $H(z) = 2z^2 + 4z + 3 + 5z^{-1}$.

Zadatak 5.5

Data je diferencna jednačina diskretnog sistema:

$$y[n] - 3.25y[n-1] = x[n-1] - 0.75y[n-2] + 3x[n-2]$$

- Odrediti prijenosnu funkciju sistema korištenjem Z-transformacije
- Napisati Python kod za crtanje nula i polova prijenosne funkcije
- Da li je sistem stabilan?

Rješenje

```

1 # -*- coding: utf-8 -*-
2 #####
3 #Odrediti inverznu Z transformaciju signala  $X(z) = z/3z^2 - 4z + 1$ .
4 #####
5 import numpy as np
6 import sympy
7 from plot_zplane import zplane
8 import scipy.signal
9
10 b = np.array([0,1])
11 a = np.array([3,-4,1])
12
13 print(scipy.signal.residuez(b,a)) # za odredjivanje polova pri rastavljanju datog
    izraza
14
15 zplane(a,b)
16
17 #####
18 # Za diferencijalnu jednačinu:
19 #  $y[n] - 3.25y[n-1] = x[n-1] - 0.75y[n-2] + 3x[n-2]$ 
20 #a) Odrediti prijenosnu funkciju sistema korištenjem Z-transformacije.
21 #b) Napisati Matlab kod za crtanje nula i polova prijenosne funkcije.
22 #c) Da li je sistem stabilan?
23 #####
24
25
26 b = np.array([0,1,3])
27 a = np.array([1,-3.25,0.75])

```

```

28
29 print(scipy.signal.residuez(b,a)) # za odre ivanje polova pri rastavljanju datog
    izraza
30
31 zplane(a,b)

```

Python kod za crtanje zplane

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from matplotlib import patches
4 from matplotlib.figure import Figure
5 from matplotlib import rcParams
6
7 def zplane(b,a,filename=None):
8     """Plot the complex z-plane given a transfer function.
9     """
10
11     # get a figure/plot
12     ax = plt.subplot(111)
13
14     # create the unit circle
15     uc = patches.Circle((0,0), radius=1, fill=False,
16                         color='black', ls='dashed')
17     ax.add_patch(uc)
18
19     # The coefficients are less than 1, normalize the coefficients
20     if np.max(b) > 1:
21         kn = np.max(b)
22         b = b/float(kn)
23     else:
24         kn = 1
25
26     if np.max(a) > 1:
27         kd = np.max(a)
28         a = a/float(kd)
29     else:
30         kd = 1
31
32     # Get the poles and zeros
33     p = np.roots(a)
34     z = np.roots(b)
35     k = kn/float(kd)
36
37     # Plot the zeros and set marker properties
38     t1 = plt.plot(z.real, z.imag, 'go', ms=10)
39     plt.setp( t1, markersize=10.0, markeredgewidth=1.0,
40             markeredgecolor='k', markerfacecolor='g')
41
42     # Plot the poles and set marker properties
43     t2 = plt.plot(p.real, p.imag, 'rx', ms=10)
44     plt.setp( t2, markersize=12.0, markeredgewidth=3.0,
45             markeredgecolor='r', markerfacecolor='r')
46
47     ax.spines['left'].set_position('center')
48     ax.spines['bottom'].set_position('center')
49     ax.spines['right'].set_visible(False)
50     ax.spines['top'].set_visible(False)
51
52     # set the ticks
53     r = 1.5; plt.axis('scaled'); plt.axis([-r, r, -r, r])
54     ticks = [-1, -.5, .5, 1]; plt.xticks(ticks); plt.yticks(ticks)
55
56     if filename is None:
57         plt.show()
58     else:
59         plt.savefig(filename)

```

```
60  
61  
62     return z, p, k
```