

# Laboratorijska vježba 3

---

## Teorijske osnove

---

### DEFINICIJA:

Sistem je linearan ako i samo ako sistem zadovoljava osobine aditivnosti i homogenosti. Drugačije rečeno, sistem je linearan ako je odziv sistema na linearnu kombinaciju dva proizvoljna signala jednak linearnoj kombinaciji odziva ta dva signala.

### DEFINICIJA:

Sistem je vremenski invarijantan ako zakašnjeli ulaz  $x(t - t_0)$  uzrokuje izlaz koji zakašnjen za istu vrijednost kao i ulaz tj  $y(t - t_0)$ .

### DEFINICIJA:

Sistem je kauzalan ako je trenutna vrijednost izlaza sistema funkcija trenutne i/ili prošlih vrijednosti ulaza.

### DEFINICIJA:

Sistem je bez memorije ako je trenutna vrijednost izlaza sistema funkcija samo trenutne vrijednosti ulaza. Svi sistemi bez memorije su kauzalni dok obrnuto ne važi. Svi akauzalni sistemi su sistemi sa memorijom.

### DEFINICIJA:

Sistem je inverzan ako mjerenjem izlaza sistema možemo odrediti ulazne vrijednosti sistema.

### DEFINICIJA:

Sistem je BIBO (*bounded-input bounded-output*) stabilan ako ograničen ulaz  $0 < |x(t)| < \alpha < \infty$  uzrokuje ograničen izlaz tj  $0 < |y(t)| < \beta < \infty$ .

### Zadatak 3.1

Za sisteme opisane jednačinama:

a)  $y(t) = 5x(t)$

b)  $y(t) = 4tx^2(t)$

c)  $y(t) = x^2(t) + 4x(t-1)$

d)  $y(t) = \frac{4}{x^2(t)}$

e)  $y(t) = 10x(t)\sin(2\pi t)$

f)  $y = \frac{dx(t)}{dt}$

realizovati testove za provjeru linearnosti, vremenske invarijantnosti i stabilnosti koristeći Python.

### Rješenje

a)  $y(t) = 5x(t)$

```

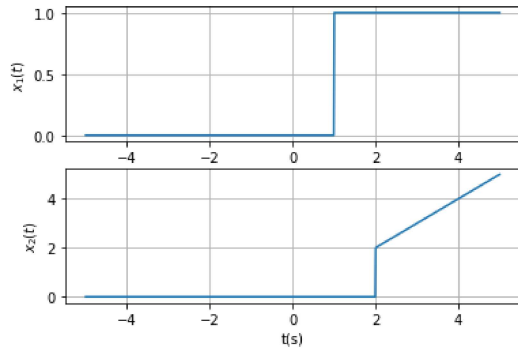
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # y(t)= 5 x(t)
5
6 #TEST LINEARNOSTI
7 t=np.linspace(0,10,1000)
8
9 a=2
10 b=3
11 x1 = np.heaviside(t-1,1);
12 x2 = t*np.heaviside(t-2,1);
13 y1 = 5*x1;
14 y2 = 5*x2;
15 x3 =5*(a*x1 + b*x2);
16 x4 =a*y1 + b*y2;
17 plt.subplot(2,1,1);
18 plt.plot(t,x1);
19 plt.ylabel("$x_1(t)$")
20 plt.xlabel("t(s)");
21 plt.grid()
22 plt.subplot(2,1,2);
23 plt.plot(t,x2);
24 plt.ylabel("$x_2(t)$")
25 plt.xlabel("t(s)");
26 plt.grid()
27
28 plt.figure(2)
29 plt.subplot(2,1,1);
30 plt.plot(t,x3,'r');
31 plt.ylabel("$x_3(t)$")
32 plt.xlabel("t(s)");
33 plt.grid()
34 plt.subplot(2,1,2);
35 plt.plot(t,x4,'r');
36 plt.ylabel("$x_4(t)$")
37 plt.xlabel("t(s)");
38 plt.grid()
39
40
41 #Test vremenske invarijantnosti
42 # y(t)= 5 x(t)

```

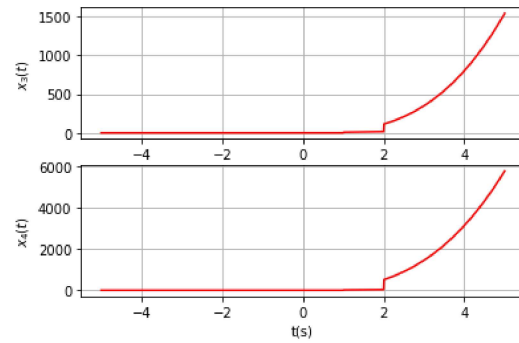
```

43
44 t0 = 2;
45 x = t*np.heaviside(t-2,1);
46 x_t0 = (t-t0)*np.heaviside(t-2-t0,1); #zakasnimu ulazni signal x(t)
47 y1 = 5*x_t0; #sistem
48
49 y=5*x # sistem
50 y2=5*(t-t0)*np.heaviside(t-2-t0,1) # zakasnimu izlazni signal
51
52 plt.figure(3)
53
54 plt.subplot(2,1,1);
55 plt.plot(t,y1,'g');
56 plt.ylabel('y1(t)');
57 plt.xlabel('t(s)');
58 plt.grid()
59
60 plt.subplot(2,1,2);
61 plt.plot(t,y2,'g');
62 plt.ylabel('y2(t)');
63 plt.xlabel('t(s)');
64 plt.grid()
65
66
67 # Test stabilnosti
68
69 t=np.linspace(0,10,1000)
70
71 #0granicen signal
72 x1 = np.heaviside(t-1,1);
73 #Neogranicen signal
74 x2 = t*np.heaviside(t-2,1);
75
76 y1 = 5*x1;
77 y2 = 5*x2;
78
79 plt.figure(5)
80 plt.subplot(4,1,1);
81 plt.plot(t,x1);
82 plt.ylabel('x_1(t)');
83 plt.xlabel('t(s)');
84 plt.subplot(4,1,2);
85 plt.plot(t,y1);
86 plt.ylabel('y_1(t)');
87 plt.xlabel('t(s)');
88 plt.subplot(4,1,3);
89 plt.plot(t,x2);
90 plt.ylabel('x_2(t)');
91 plt.xlabel('t(s)');
92 plt.subplot(4,1,4);
93 plt.plot(t,y2);
94 plt.ylabel('y_2(t)');
95 plt.xlabel('t(s)');

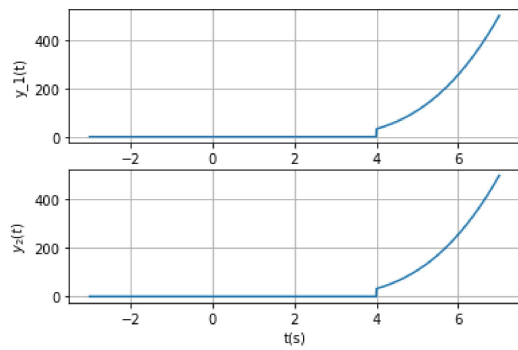
```



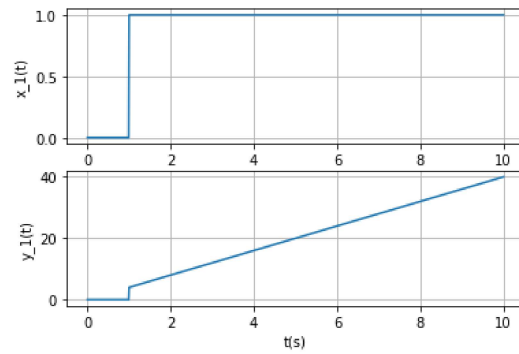
Slika 3.1: Test linearnosti



Slika 3.2: Test linearnosti



Slika 3.3: Test vrem invarijantnosti



Slika 3.4: Test stabilnosti

b)  $y(t) = 4tx^2(t)$

```

1
2 #####
3 # mojModul.py
4 #####
5 import numpy as np
6
7 def rect(t,tau):
8     x = np.zeros(len(t))
9     for k,tk in enumerate(t):
10         if np.abs(tk) > tau/2.:
11             x[k] = 0
12         else:
13             x[k] = 1
14     return x;
15
16 def tri(t,tau):
17     x = np.zeros(len(t))
18     for k,tk in enumerate(t):
19         if np.abs(tk) > tau/1.:
20             x[k] = 0
21         else:
22             x[k] = 1 - np.abs(tk)/tau
23     return x;
24
25 def sigShift(x,t,t0):
26     y=x;
27     t=t+t0;
28     return y, t;
29

```

```

30 def sigFold(t,x):
31     y=np.flipud(x);
32     t=-np.flipud(t);
33     return y, t;
34
35 def sigScale(t,x,k):
36     y=x;
37     t=t/k;
38     return y, t;

1
2 #####
3 # LV2_p2.py
4 #####
5
6
7
8
9 import numpy as np
10 import matplotlib.pyplot as plt
11 import mojModul as moj
12
13 #####
14 # y(t)= 4*t* x^2(t)
15 #TEST LINEARNOSTI
16 #####
17 t=np.linspace(-5,5,1000)
18
19 a=2
20 b=3
21 x1 = np.heaviside(t-1,1);
22 x2 = t*np.heaviside(t-2,1);
23 y1 = 4*t*x1*x1;
24 y2 = 4*t*x2*x2;
25 x3 =a*y1 + b*y2;
26 x4 =4*t*(a*x1 +b*x2)*(a*x1 +b*x2)
27
28 plt.subplot(2,1,1);
29 plt.plot(t,x1);
30 plt.ylabel("$x_1(t)$")
31 plt.xlabel("t(s)");
32 plt.grid()
33 plt.subplot(2,1,2);
34 plt.plot(t,x2);
35 plt.ylabel("$x_2(t)$")
36 plt.xlabel("t(s)");
37 plt.grid()
38
39 plt.figure(2)
40 plt.subplot(2,1,1);
41 plt.plot(t,x3,'r');
42 plt.ylabel("$x_3(t)$")
43 plt.xlabel("t(s)");
44 plt.grid()
45 plt.subplot(2,1,2);
46 plt.plot(t,x4,'r');
47 plt.ylabel("$x_4(t)$")
48 plt.xlabel("t(s)");
49 plt.grid()
50
51
52 #####
53 #Test vremenske invarijantnosti
54 #y(t)= 4*t* x^2(t)
55 #####
56 t0 = 2;
57 x = t*np.heaviside(t-2,1);
58 x_t0,t1 = moj.sigShift(x,t,t0); #zakasnimo ulazni signal x(t)

```

```

59 y1 = 4*t*x_t0*x_t0;#sistem
60
61 y = 4*t*x*x; # sistem
62 y2,t2= moj.sigShift(y,t,t0); # zakasnimo izlazni signal
63
64 #####
65 if np.array_equal(y1,y2):
66     print("Sistem je vremenski invarijantan.")
67 else:
68     print("Sistem nije vremenski invarijantan.")
69 #####
70 plt.figure(3)
71 plt.subplot(2,1,1);
72 plt.plot(t1,y1);
73 plt.ylabel('y_1(t)');
74 plt.xlabel('t(s)');
75 plt.grid()
76
77 plt.subplot(2,1,2);
78 plt.plot(t2,y2);
79 plt.ylabel('$y_2(t)$');
80 plt.xlabel('t(s)');
81 plt.grid()
82
83 #####
84 # Test stabilnosti
85 # y(t)= 4*t* x^2(t)
86 #####
87 t=np.linspace(0,10,1000)
88
89 #Ogranicen signal
90 x1 = np.heaviside(t-1,1);
91
92 y1 = 4*t*x1*x1;
93
94
95 plt.figure(5)
96 plt.subplot(2,1,1);
97 plt.plot(t,x1);
98 plt.ylabel('x_1(t)');
99 plt.xlabel('t(s)');
100 plt.grid()
101 plt.subplot(2,1,2);
102 plt.plot(t,y1);
103 plt.ylabel('y_1(t)');
104 plt.xlabel('t(s)');
105 plt.grid()

```

c)  $y(t) = x^2(t) + 4x(t-1)$

```

1 #####
2 # LV2_p3.py
3 #####
4
5 import numpy as np
6 import matplotlib.pyplot as plt
7 import mojModul as moj
8
9 #####
10 # y(t)= x^2(t) + 4x( t 1 )
11 #TEST LINEARNOSTI
12 #####
13 t=np.linspace(-5,5,1000)
14
15
16 a=2
17 b=3
18 x1 = np.heaviside(t,1);

```

```

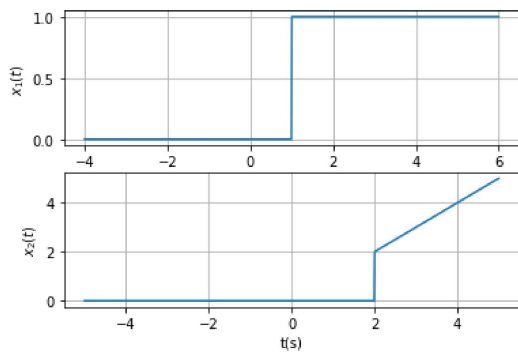
19 x11,t1=moj.sigShift(x1,t,1);
20 x2 = t*np.heaviside(t-2,1);
21 x21,t2=moj.sigShift(x2,t,1);
22
23 y1 = x1*x1 + 4*x11
24 y2 = x2*x2 +4*x21
25
26 x3 =a*y1 + b*y2;
27 x4 =(a*x1 +b*x2)*(a*x1 +b*x2) + 4*(a*x11 +b*x21)
28
29 #####
30 if np.array_equal(x3,x4):
31     print("Sistem je linearan.")
32 else:
33     print("Sistem je nelinearan.")
34 #####
35
36
37
38 plt.subplot(2,1,1);
39 plt.plot(t1,x1);
40 plt.ylabel("$x_1(t)$")
41 plt.xlabel("t(s)");
42 plt.grid()
43 plt.subplot(2,1,2);
44 plt.plot(t,x2);
45 plt.ylabel("$x_2(t)$")
46 plt.xlabel("t(s)");
47 plt.grid()
48
49 plt.figure(2)
50 plt.subplot(2,1,1);
51 plt.plot(t,x3,'r');
52 plt.ylabel("$x_3(t)$")
53 plt.xlabel("t(s)");
54 plt.grid()
55 plt.subplot(2,1,2);
56 plt.plot(t,x4,'r');
57 plt.ylabel("$x_4(t)$")
58 plt.xlabel("t(s)");
59 plt.grid()
60
61
62 #####
63 #Test vremenske invarijantnosti
64 # y(t)= x^2(t) + 4x( t 1 )
65 #####
66 t0 = 2;
67 x = np.heaviside(t,1);
68 x1,t1=moj.sigShift(x1,t,1);
69
70 x_t0,t1 = moj.sigShift(x,t,t0); #zakasnimo ulazni signal x(t)
71 x_t10,t3 = moj.sigShift(x1,t1,t0); #zakasnimo ulazni signal x(t)
72 y1 = x_t0*x_t0 + 4* x_t10      #sistem
73
74 y = x*x + 4*x1; # sistem
75 y2,t2= moj.sigShift(y,t,t0); # zakasnimo izlazni signal
76
77 #####
78 if np.array_equal(y1,y2):
79     print("Sistem je vremenski invarijantan.")
80 else:
81     print("Sistem nije vremenski invarijantan.")
82 #####
83 plt.figure(3)
84 plt.subplot(2,1,1);
85 plt.plot(t1,y1);
86 plt.ylabel('y_1(t)')

```

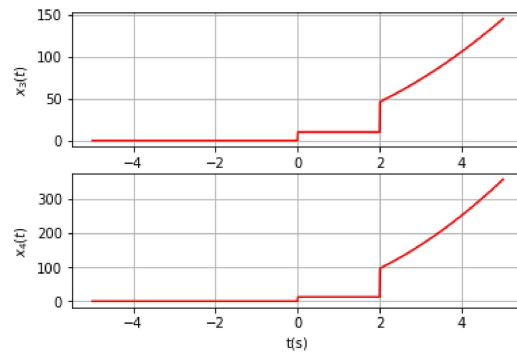
```

87 plt.xlabel('t(s)');
88 plt.grid()
89
90 plt.subplot(2,1,2);
91 plt.plot(t2,y2);
92 plt.ylabel('$y_2(t)$');
93 plt.xlabel('t(s)');
94 plt.grid()
95
96 #####
97 # Test stabilnosti
98 #  $y(t) = x^2(t) + 4x(t-1)$ 
99 #####
100 t=np.linspace(0,10,1000)
101
102 #Ogranicen signal
103 x1 = np.heaviside(t-1,1);
104 x2,t1=moj.sigShift(x1,t,1);
105
106 y1 = x1*x1 + 4*x2;
107
108
109 plt.figure(5)
110 plt.subplot(2,1,1);
111 plt.plot(t,x1);
112 plt.ylabel('x_1(t)');
113 plt.xlabel('t(s)');
114 plt.grid()
115 plt.subplot(2,1,2);
116 plt.plot(t1,y1);
117 plt.ylabel('y_1(t)');
118 plt.xlabel('t(s)');
119 plt.grid()

```

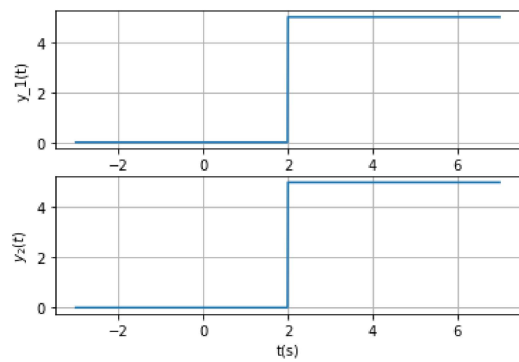


Slika 3.5: Test linearnosti

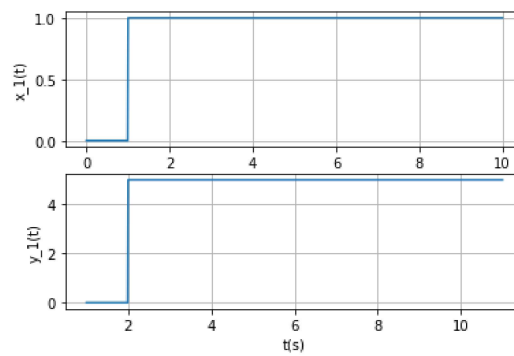


Slika 3.6: Test linearnosti





Slika 3.7: Test vrem invarijantnosti



Slika 3.8: Test stabilnosti

d)  $y(t) = \frac{4}{x^2(t)}$

e)  $y(t) = 10x(t)\sin(2\pi t)$

f)  $y = \frac{dx(t)}{dt}$