Εργαστήριο Ψηφιακής Επεξεργασίας Σήματος
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Η/Υ

Ψηφιακή Επεξεργασία Σήματος

Εργαστήριο 2

Συστήματα Διακριτού Χρόνου

Ηράκλειο 2025
Δρ. Κωνσταντίνος Καραμπίδης

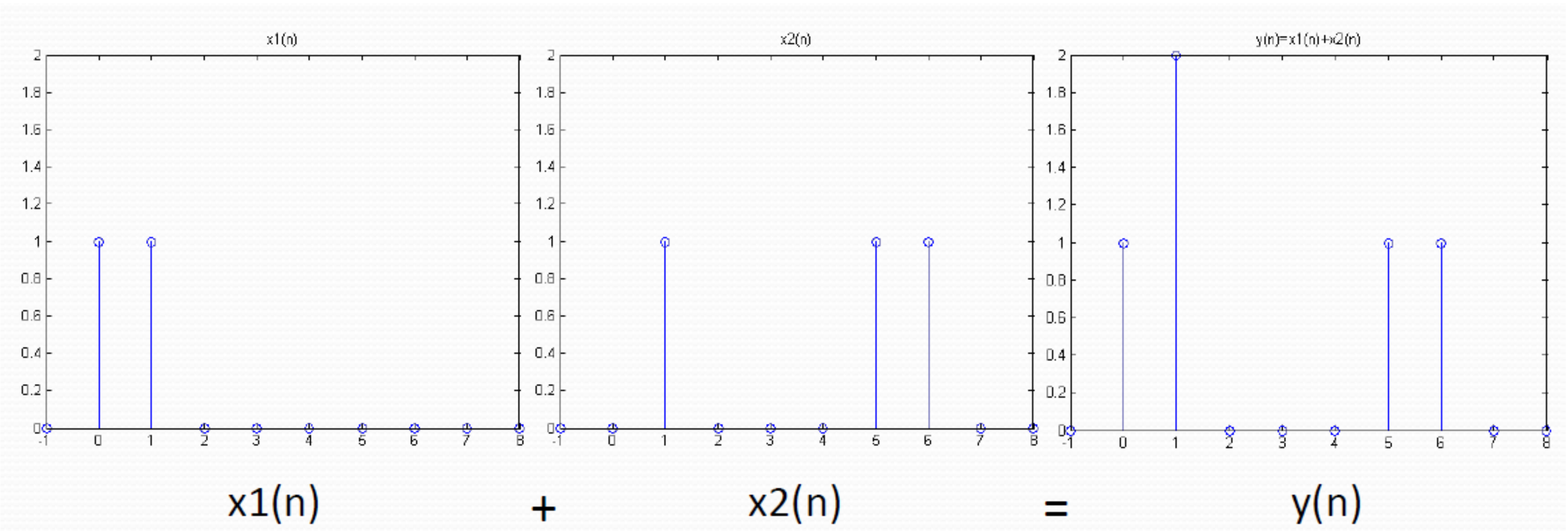
Πράξεις διακριτών σημάτων

Πρόσθεση	$x(n) + y(n)$
Αφαίρεση	$x(n) - y(n)$
Πολλαπλασιασμός	$x(n) * y(n)$
Διαίρεση	$x(n) / y(n)$ με $y(n) \neq 0$

Οι πράξεις εκτελούνται **ανά στοιχείο** και για την ίδια τιμή της ανεξάρτητης μεταβλητής

Παράδειγμα

Έστω ότι θέλουμε να προσθέσουμε τα δύο παρακάτω σήματα x1(n) και x2(n):



Ανάλυση Σημάτων

Χρησιμοποιώντας της διακριτή ακολουθία δέλτα **δ**, μπορούμε να αναλύσουμε ένα τυχαίο σήμα.
Αυτό πραγματοποιείται με το άθροισμα των κατάλληλα μετατοπισμένων δ(n) τα οποία έχουν πολλαπλασιαστεί με έναν συντελεστή βάρους.
Ο συντελεστής βάρους αντιστοιχεί στην εκάστοτε τιμή του σήματος x(n), όπως φαίνεται στην παρακάτω παράδειγμα:

$$x(n) = \cdots + x(-1)\delta(n+1) + x(0)\delta(n) + x(1)\delta(n-1) + x(2)\delta(n-2) + \cdots$$

ΑνάλυσηΣημάτων

$$x(n) = \cdots + x(-1)\delta(n+1) + x(0)\delta(n) + x(1)\delta(n-1) + x(2)\delta(n-2) + \cdots$$

Αυτό το άθροισμα μπορεί να γραφτεί περιληπτικά:

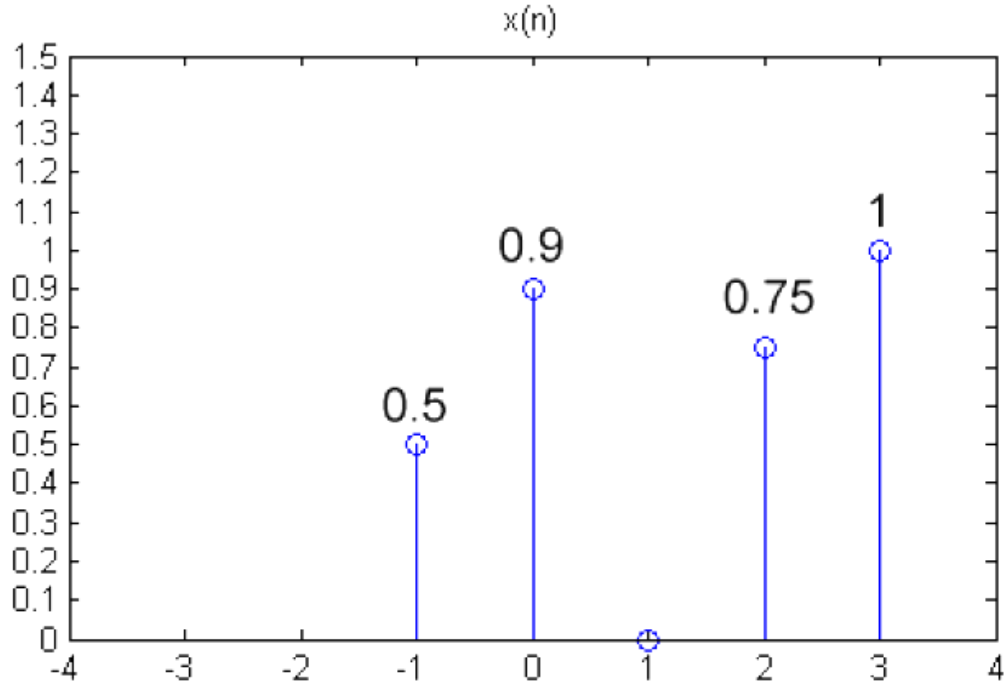
$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k)$$

όπου κάθε όρος x(k) δ(n-k), είναι ένα σήμα με πλάτος x(k) τη χρονική στιγμή n=k, ενώ μηδενίζεται για οποιαδήποτε άλλη τιμή του.

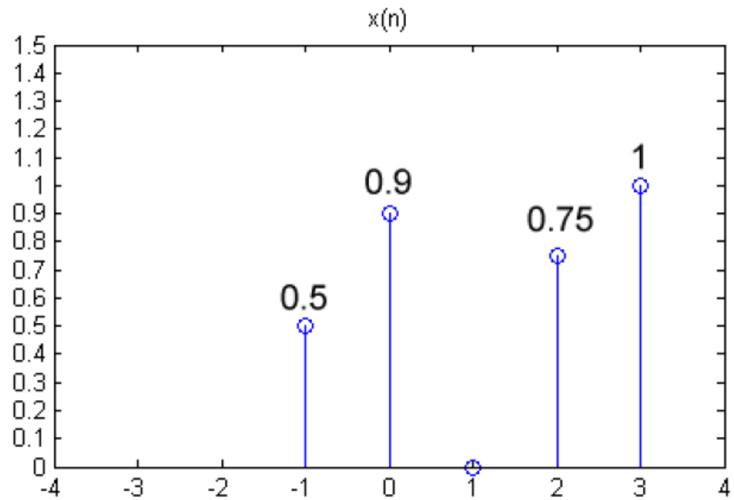
Παράδειγμα 1 - Ανάλυση Σημάτων

Χρησιμοποιώντας την έκφραση ανάλυσης σημάτων, γράψτε σε μορφή αθροισμάτων διακριτών συναρτήσεων δέλτα το παρακάτω σήμα:

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k)$$



Παράδειγμα 1 - Ανάλυση Σημάτων

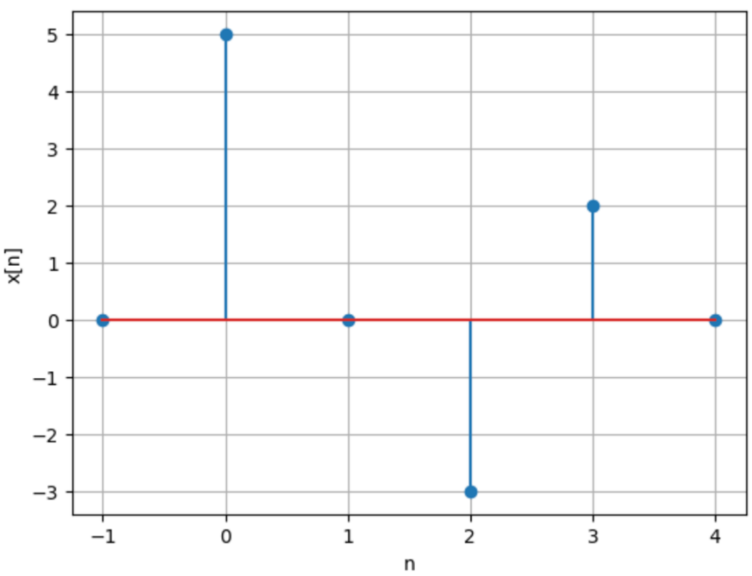


Το παραπάνω σήμα μπορεί να γραφεί σε μορφή αθροισμάτων διακριτών συναρτήσεων δέλτα ως:

$$x(n) = 0.5*\delta(n+1) + 0.9*\delta(n) + 0.75*\delta(n-2) + \delta(n-3)$$

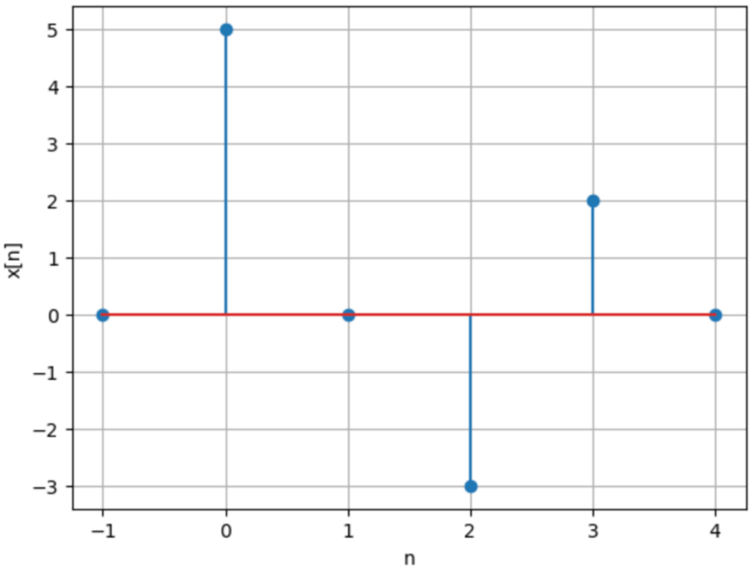
Παράδειγμα 2 - Ανάλυση Σημάτων

Χρησιμοποιώντας την έκφραση ανάλυσης σημάτων, γράψτε σε μορφή αθροισμάτων διακριτών συναρτήσεων δέλτα το παρακάτω σήμα:



Παράδειγμα 2 - Ανάλυση Σημάτων

Το παραπάνω σήμα μπορεί να γραφεί σε μορφή αθροισμάτων διακριτών συναρτήσεων δέλτα ως:

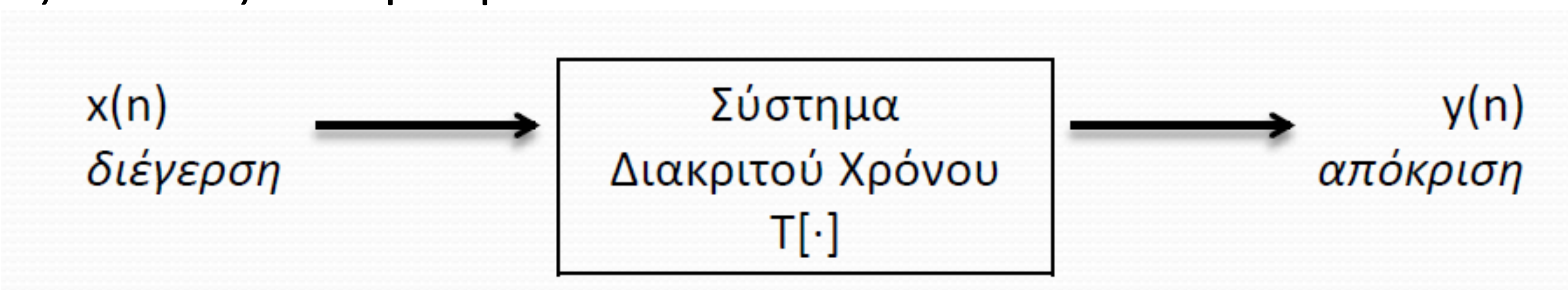


$$x(n) = 5\delta(n) - 3\delta(n-2) + 2\delta(n-3)$$

Συστήματα Διακριτού Χρόνου

Ένα σύστημα διακριτού χρόνου δέχεται μια είσοδο διακριτού χρόνου x(n) και παράγει μια έξοδο διακριτού χρόνου y(n), μετασχηματίζοντας το.

Το σήμα εισόδου μπορεί να ονομαστεί ως διέγερση ενώ το σήμα εξόδου ως απόκριση.



Αυτή η διαδικασία συμβολίζεται ως εξής:

$$y(n) = T[x(n)]$$

Ιδιότητες Διακριτών Σημάτων

Αιτιατά Συστήματα

Αιτιατό είναι το σύστημα στο οποίο η έξοδος για κάθε χρονική στιγμή n_0 εξαρτάται από την είσοδο τη χρονική στιγμή n_0 και παρελθοντικές χρονικές στιγμές (όχι μελλοντικές).
Για παράδειγμα το παρακάτω σύστημα είναι αιτιατό:

$$y(n) = \alpha x(n) - \beta x(n-1)$$

Μη αιτιατό είναι το σύστημα στο οποίο η έξοδος τη χρονική στιγμή n εξαρτάται από μελλοντικές χρονικές στιγμές στην είσοδο.

Ιδιότητες Διακριτών Σημάτων

Αιτιατά Συστήματα – Παράδειγμα

Έστω το σήμα $y[n]=0.5y[n-1]+x[n]$, όπου:

- $y[n]$ είναι η έξοδος του συστήματος τη χρονική στιγμή n,
- $y[n-1]$ είναι η έξοδος του συστήματος στο προηγούμενο χρονικό βήμα [n-1] και
- $x[n]$ είναι η είσοδος στο σύστημα τη χρονική στιγμή n.

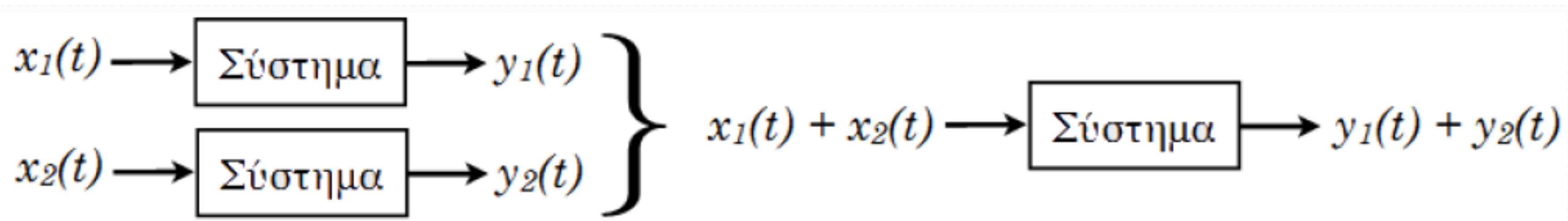
Το σύστημα είναι αιτιατό διότι:

- Ανά πάσα στιγμή n, η έξοδος $y[n]$ εξαρτάται από την τρέχουσα είσοδο $x[n]$ και την προηγούμενη έξοδο $y[n-1]$
- Δεν εξαρτάται από μελλοντικές τιμές της εισόδου.

Ιδιότητες Διακριτών Σημάτων

Αρχή της Υπέρθεσης ή Επαλληλίας

Η αρχή της Υπέρθεσης ή Επαλληλίας: ορίζεται εάν το άθροισμα των εξόδων πολλαπλών συστημάτων είναι ίσο με την έξοδο του συστήματος όπου ως είσοδο έχει το άθροισμα των εισόδων αυτών των συστημάτων.



Η αρχή της υπέρθεσης ορίζεται όταν σε ένα σύστημα ισχύει:

$$T[x_1(n) + x_2(n)] = T[x_1(n)] + T[x_2(n)]$$

Ιδιότητες Διακριτών Σημάτων

Ομογένεια

Ένα σύστημα ονομάζεται **Ομογενές** εάν ο πολλαπλασιασμός της εισόδου με μία σταθερά οδηγεί τον πολλαπλασιασμό της εξόδου με την ίδια ακριβώς σταθερά.

$$ax(t) \longrightarrow \boxed{\text{Σύστημα}} \longrightarrow ay(t)$$

Ένα σύστημα καλείται Ομογενές όταν:

$$T[\alpha x(n)] = \alpha T[x(n)]$$

όπου α μια σταθερά.

Ιδιότητες Διακριτών Σημάτων

Γραμμικά Συστήματα

Ένα σύστημα είναι γραμμικό εάν είναι **ομογενές** και για το οποίο ισχύει η **αρχή της υπέρθεσης**

Επομένως για να είναι γραμμικό ένα σύστημα θα πρέπει να ισχύει η σχέση:

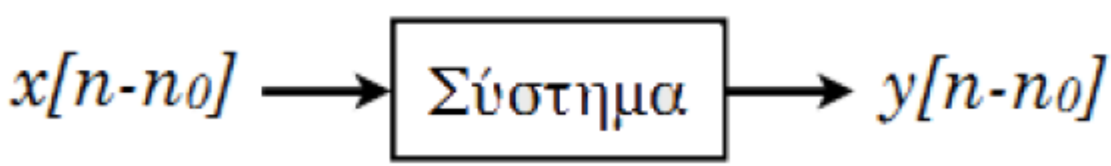
$$T[\alpha_1 x_1(n) + \alpha_2 x_2(n)] = \alpha_1 T[x_1(n)] + \alpha_2 T[x_2(n)]$$

Στις σημειώσεις τις θεωρίας μπορείτε να βρείτε διάφορα παραδείγματα με γραμμικά και μη γραμμικά συστήματα.

Ιδιότητες Διακριτών Σημάτων

Χρονικά Αμετάβλητα Συστήματα

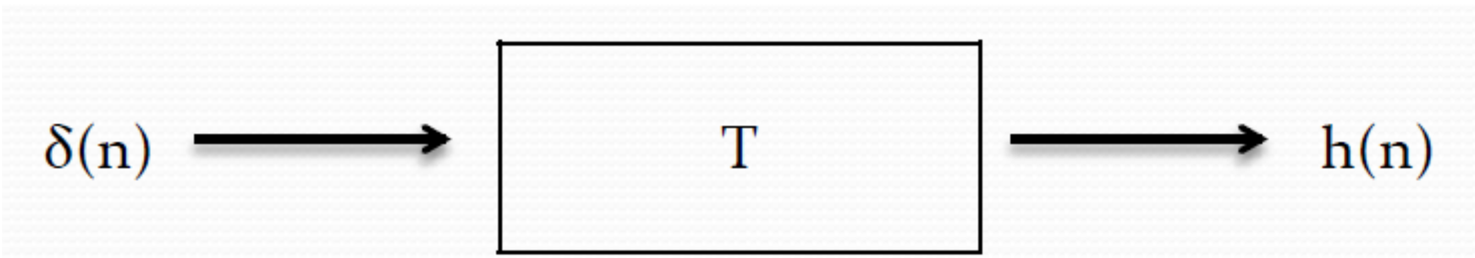
Αν σε ένα σύστημα υπάρχει μια μετατόπιση (καθυστέρηση) στην είσοδο κατά n_0 και παράγει μια έξοδο με την ίδια μετατόπιση n_0, τότε το σύστημα ονομάζεται **Χρονικά Αμετάβλητο ή Αμετάβλητο στην μετατόπιση.**



Εάν ένα σύστημα είναι και Γραμμικό και Χρονικά αμετάβλητο θα το γράφουμε για συντομία ως: **ΓΧΑ** ή **LSI** (Linear Shift Invariant).

Κρουστική απόκριση Impulse Response

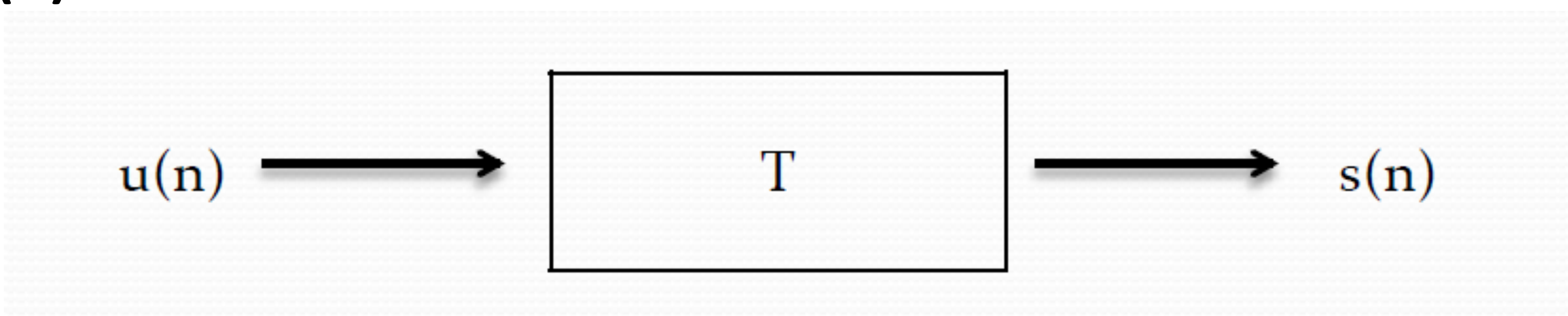
Εάν σε ένα γραμμικό και χρονικά αμετάβλητο (ΓΧΑ) σύστημα η είσοδος είναι η μοναδιαία κρουστική ακολουθία δ(n), τότε το σήμα εξόδου (απόκριση) ονομάζεται **Κρουστική απόκριση h(n).**



Η κρουστική απόκριση h(n) θα είναι αιτιατή **αν και μόνο αν** είναι ίση με το μηδέν για κάθε n<0.

Βηματική απόκριση - Step Response

Εάν σε ένα γραμμικό και χρονικά αμετάβλητο (ΓΧΑ) σύστημα η είσοδος είναι η μοναδιαία βηματική ακολουθία u(n), τότε το σήμα εξόδου(απόκριση) ονομάζεται **Βηματική απόκριση s(n).**



Εξισώσεις Διαφορών

Η γενική μορφή μιας γραμμικής εξίσωσης διαφορών με σταθερούς συντελεστές είναι:

$$y(n) = \sum_{k=0}^{q} b(k)x(n-k) - \sum_{k=1}^{p} a(k)y(n-k)$$

όπου a(k) και b(k) είναι σταθερές οι οποίες καθορίζουν το σύστημα. Οι εξισώσεις διαφορών παρέχουν μια μέθοδο υπολογισμού της απόκρισης ενός συστήματος για μια τυχαία είσοδο. Για την λύση τέτοιων εξισώσεων είναι συχνά απαραίτητο να υπολογιστεί ένα σύνολο Αρχικών Συνθηκών.

Για ένα ΓΧΑ σύστημα το οποίο περιγράφεται από μια εξίσωση διαφορών, η κρουστική του απόκριση h(n), υπολογίζεται λύνοντας την αντίστοιχη εξίσωση διαφορών για x(n)=δ(n) και y(n)=h(n).

Εξισώσεις Διαφορών

Η συνάρτηση lfilter στο scipy μας βοηθά να επιλύσουμε εξισώσεις διαφορών με στόχο να βρούμε την απόκριση του συστήματος:

Σύνταξη

scipy.signal.lfilter(b, a, x)

όπου: b=[, b1,…, bm] και α=[α0, α1,…, αm] είναι οι συντελεστές της εξίσωσης διαφορών, ενώ το διάνυσμα x είναι ο πίνακας με τις τιμές του σήματος εισόδου του συστήματος.

Η Εντολή “lambda”

Με την εντολή “lambda” μπορούμε να ορίσουμε μια συνάρτηση δυναμικά, με την εξής σύνταξη: όνομα_συνάρτησης = lambda(έκφραση)

Για παράδειγμα, για να ορίσουμε μια συνάρτηση υπολογισμού τετραγώνου: my_square = lambda x: x*x

Για να την χρησιμοποιήσουμε: my_square(5) το αποτέλεσμα θα είναι 25.

Η συνάρτηση αυτή μπορεί να χρησιμοποιηθεί προσωρινά.

Πρέπει να αναφέρουμε ότι αντί του x θα μπορούσαμε να είχαμε γράψει οποιοδήποτε έγκυρο όνομα μεταβλητής.

Άσκηση

Να γράψετε στο χαρτί σε μορφή αθροισμάτων διακριτών συναρτήσεων δέλτα δ, και στην συνέχεια να παρασταθεί γραφικά το σήμα:

$$x(n) = \left\{ -2, 1, -\frac{1}{3}, \underset{\uparrow}{\frac{1}{2}}, 2, 1, 0, 1, 0, 0, 3 \right\}$$

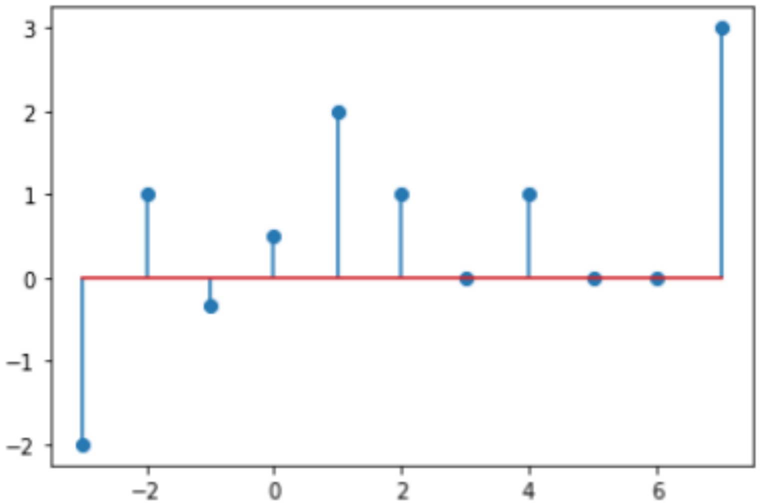
Το βέλος δείχνει την τιμή για την χρονική στιγμή n=0.

Λύση

$$x(n) = \left\{-2, 1, -\frac{1}{3}, \underset{\uparrow}{\frac{1}{2}}, 2, 1, 0, 1, 0, 0, 3\right\}$$

x(n) = -2δ(n+3) + δ(n+2) – 1/3δ(n+1) + 1/2δ(n) +2δ(n-1) +δ(n-2) + δ(n-4) + 3δ(n-7)

Γραφική αναπαράσταση




```
import numpy as np
import matplotlib.pyplot as plt
delta = lambda n: n==0
n=range(-3,8,1)
x=np.zeros(len(n));
h=np.zeros(len(n));
h[1]=1;
out=[]
for i in range(0,len(n),1):
   x[i]=-2*delta(n[i]+3) + delta(n[i]+2) -
1/3*delta(n[i]+1)+ 1/2*delta(n[i]) + 2*delta(n[i]-1) + delta(n[i]-2)+ delta(n[i]-4) + 3*delta(n[i]-7);
   out.append(x[i])
plt.stem(n,out)
```

Υπολογισμός Κρουστικής απόκρισης

Να υπολογίσετε και να παραστήσετε γραφικά την κρουστική απόκριση του παρακάτω αιτιατού σήματος:

$$y(n) = -0.9y(n-1) + x(n)$$

1ος Τρόπος Λύσης:

Λύνουμε την εξίσωση διαφορών για $x(n)=\delta(n)$ και $y(n)=h(n)$

Άρα: $h(n) = -0.9h(n-1) + \delta(n)$

Υπολογισμός αρχικών συνθηκών:

$h(0) = -0.9 * 0 + 1 = 1$

$h(1) = -0.9 * h(0) + \delta(1)= -0.9 * 1 + 0 = -0.9$

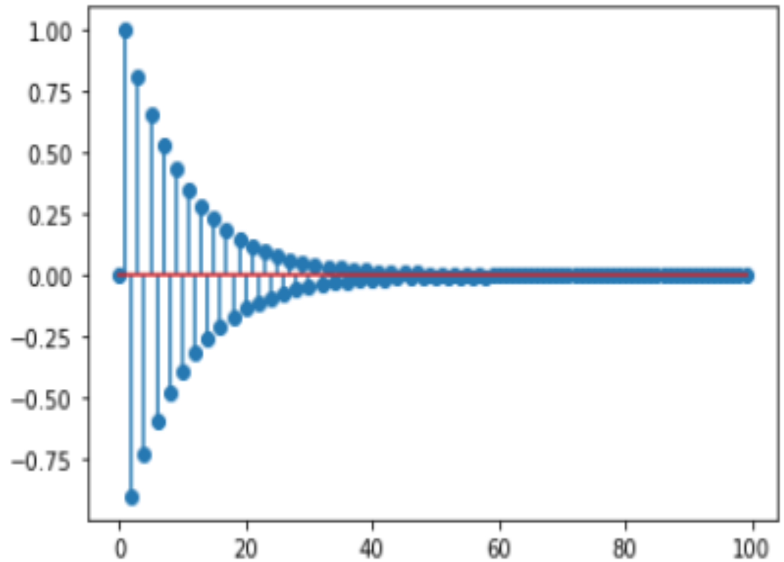
$h(2) = -0.9 * h(1) + \delta(2) = -0,9 * -0,9 + 0 = 0,81$

$h(3) = -0.9 * h(2) + \delta(3) = -0,9 * 0,81 + 0 = -0,729$

Υπολογισμός Κρουστικής απόκρισης

1ος Τρόπος Λύσης

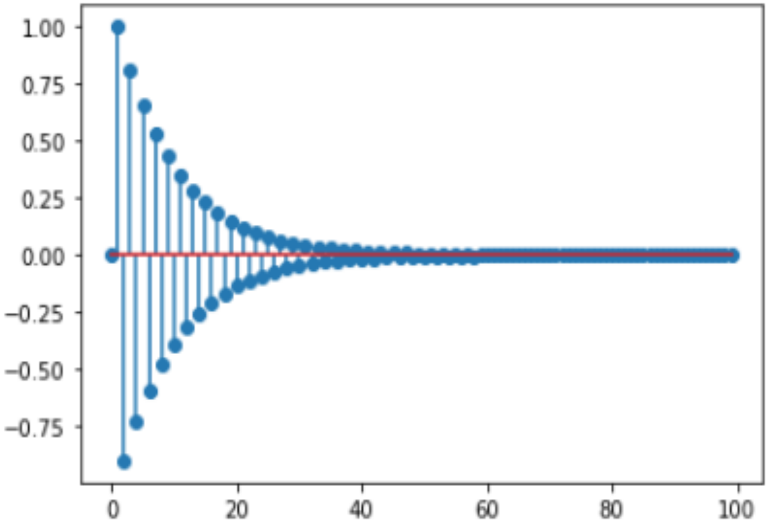
```
import numpy as np
import matplotlib.pyplot as plt
n=range(0,100,1)
h=np.zeros(len(n));
delta = lambda n: n==0
h[1]=1
for i in range(2,len(n),1):
   h[i]=-0.9*h[i-1]+int(delta(n[i]))
plt.stem(n,h)
```



Υπολογισμός Κρουστικής απόκρισης

1ος Τρόπος Λύσης – Άλλη προσέγγιση

```
import numpy as np
import matplotlib.pyplot as plt
h[1]=1
for n in range(2,100):
    h[n]=-0.9*h[n]+int(delta(n))
plt.stem(range(0,100,1),h)
```



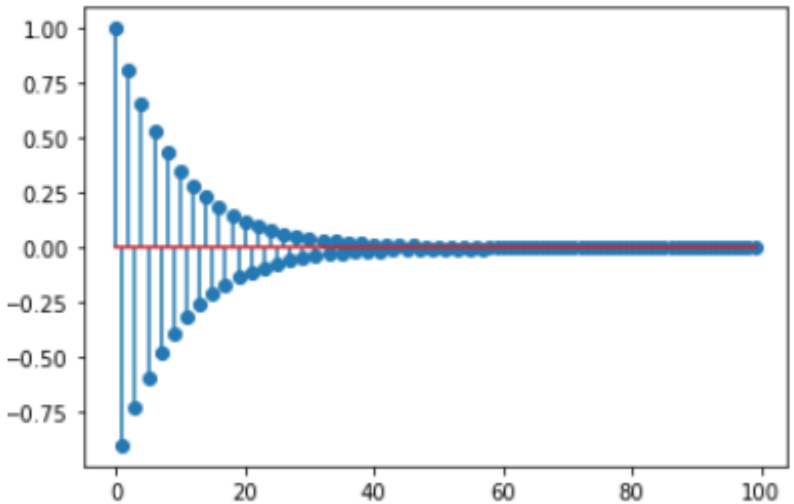
Υπολογισμός Κρουστικής απόκρισης

Ο 2ος Τρόπος Λύσης είναι να χρησιμοποιήσουμε την συνάρτηση lfilter του scipy.

- Έχοντας υπόψη την γενική μορφή μιας γραμμικής εξίσωσης διαφορών με σταθερούς συντελεστές, χωρίζουμε τα y από τα x.
- Στην συνέχεια βρίσκουμε τους συντελεστές a και b (συντελεστές της εξίσωσης διαφορών).
- Το a αντιστοιχεί στους συντελεστές βάρους των y, ενώ το b αντιστοιχεί στους συντελεστές βάρους των x.

Υπολογισμός Κρουστικής απόκρισης

```
## y(n) = -0.9y(n-1) + x(n) => y(n) + 0.9y(n-1) = x(n)
import scipy
from scipy import signal
a=[1 , 0.9]
b=[1, 0]
x=np.zeros(100)
for n in range(0,100,1):
    temp=int(delta(n))
    x[n]=temp
n=range(0,100,1)
y= scipy.signal.lfilter(b,a,x)
plt.stem(n,y)
```



Υπολογισμός Βηματικής απόκρισης

Να υπολογίσετε και να παραστήσετε γραφικά την βηματική απόκριση του παρακάτω αιτιατού σήματος:
y(n)= y(n-1)+ x(n)-x(n-8)

<u>1ος Τρόπος Λύσης</u>

Λύνουμε την εξίσωση διαφορών για x(n)=u(n) και y(n)=s(n)

s(n) = s(n-1) + u(n)-u(n-8)

Υπολογισμός αρχικών συνθηκών:

$$u(n) = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$$

s(0) = s(-1) + u(0) - u(-8) = 0 + 1 - 0 = 1

s(1) = s(0) + u(1) - u(-7) = 1 + 1 - 0 = 2

s(2) = s(1) + u(2) - u(-6) = 2 + 1 - 0 = 3

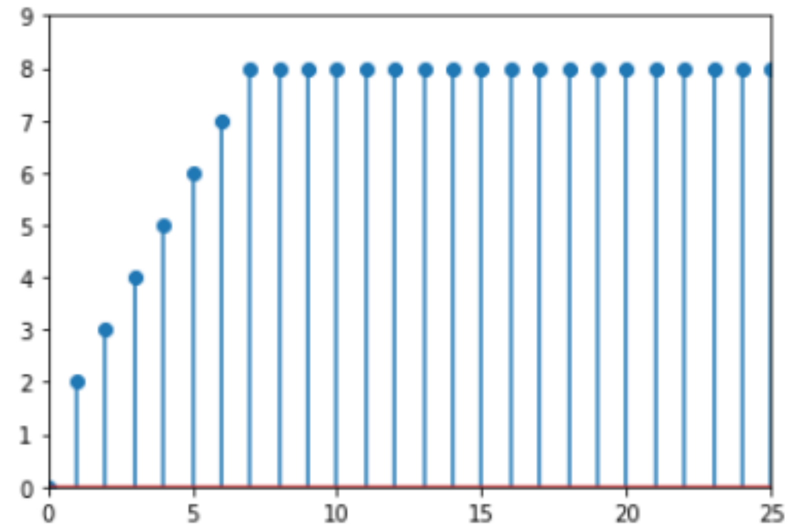
....

s(7) = s(6) + u(7) -u(-1) = 7 + 1 -0 = 8

s(8) = s(7) + u(8) -u(0) = 8 + 1 -1 = 8

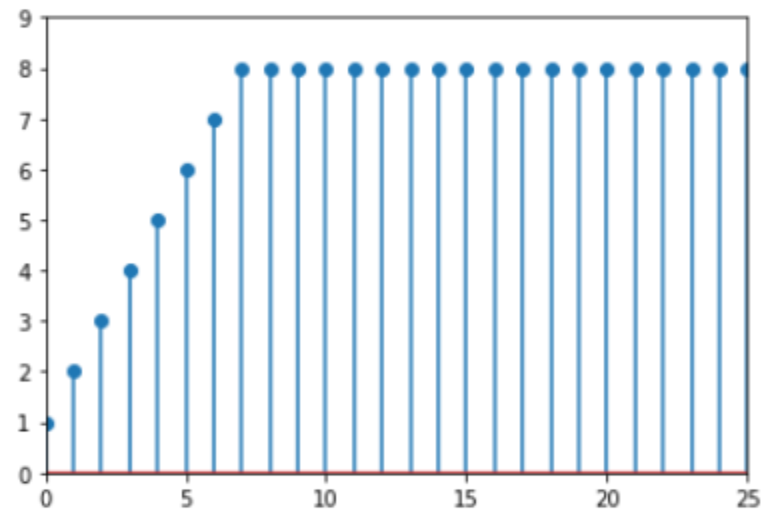
s(9) = s(8) + u(9) -u(1) = 8 + 1 -1 = 8

```
n=range(0,100,1)
s=np.zeros(len(n));
u = lambda n: n>=0
s[1]=2
for i in range(2,len(n),1):
    s[i]=s[i-1]+int(u(n[i]))-int(u(n[i]-8))
plt.stem(s)
plt.xlim([0, 25])
plt.ylim([0, 9])
```



2ος Τρόπος Λύσης χρησιμοποιώντας την συνάρτηση lfilter
y(n) = y(n-1) + x(n) – x(n-8) => y(n) – y(n-1) = x(n) – x(n-8)

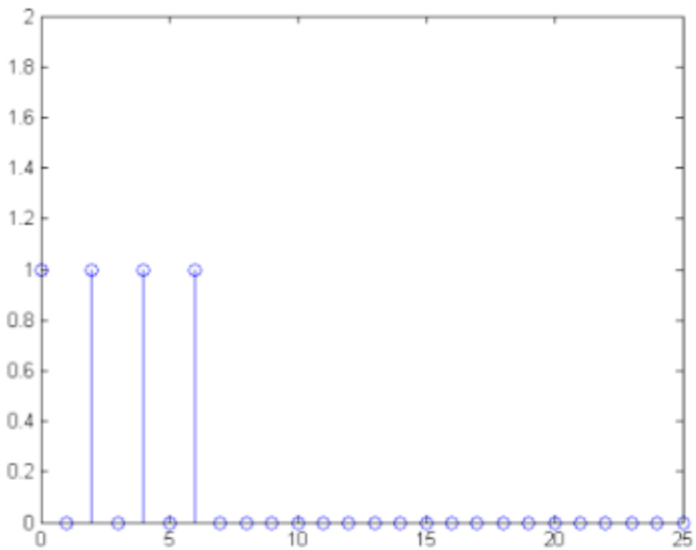
```
n=range(0,100,1)
unit = lambda n: n>=0
a=[1,-1,0,0,0,0,0,0,0]
b=[1,0,0,0,0,0,0,0,-1]
u=np.zeros(100)
for n in range(0,100,1):
   temp=int(unit(n))
   u[n]=temp
y=scipy.signal.lfilter(b,a,u)
plt.stem(y)
plt.xlim([0, 25])
plt.ylim([0, 9])
```



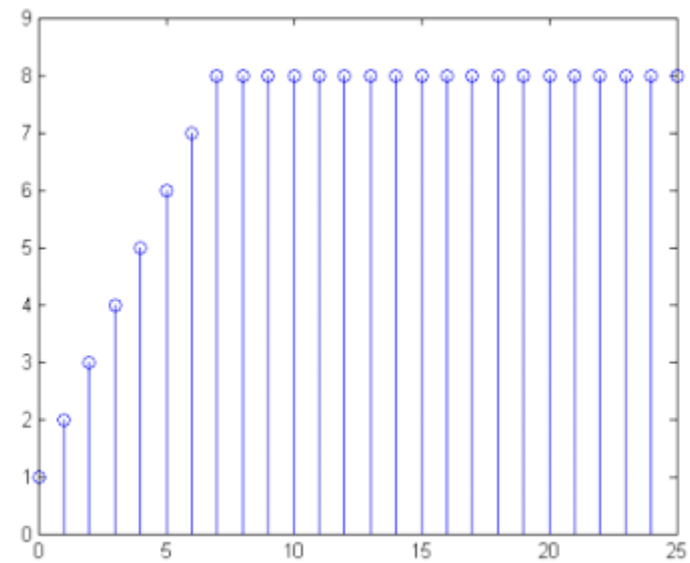
Άσκηση

Σχεδιάστε την απόκριση του προηγούμενου συστήματος με είσοδο $\mathbf{x(n) = a^n u(n)}$, για a=1 και a=-1.

```
##y(n)= y(n-1) + x(n) –x(n-8) => y(n) = y(n-1) + a^n u(n) –a^(n-8) u(n-8)
a=1
s[1]=0
n=range(0,100,1)
unit = lambda n: n>=0
for i in range(2,100,1):
   s[i]=s[i-1]+a**n[i]*unit((n[i]))-a**(n[i]-8)*unit((n[i]-8))
plt.stem(s)
plt.xlim([0, 25])
plt.ylim([0, 9])
```



a=-1



a=1