Ψηφιακή Επεξεργασία Σημάτων

Συστήματα Διακριτού Χρόνου
(Discrete-Time Systems)

Κυριακίδης Ιωάννης

2011

Τελευταία ενημέρωση: 11/11/2011

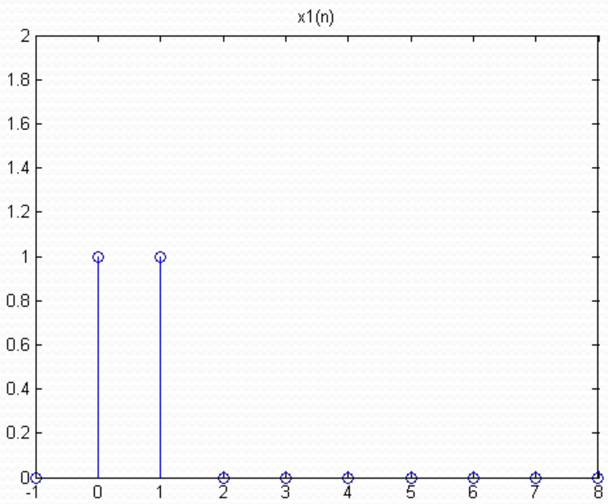
Πράξεις διακριτών σημάτων (υπενθύμιση)

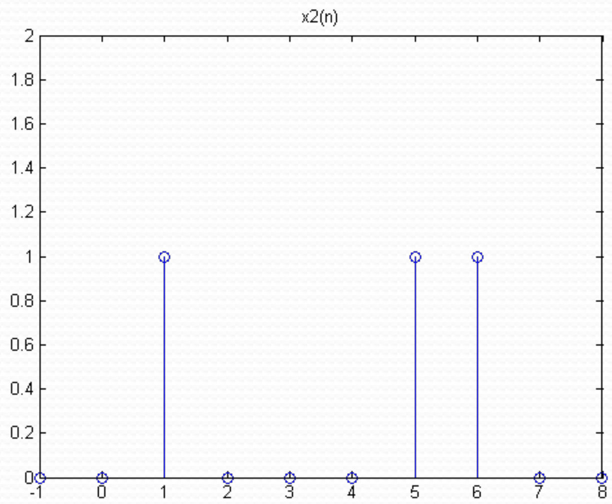
Πρόσθεση	x(n) + y(n)
Αφαίρεση	x(n) – y(n)
Πολλαπλασιασμός	x(n) * y(n)
Διαίρεση	x(n) / y(n) με y(n)≠0

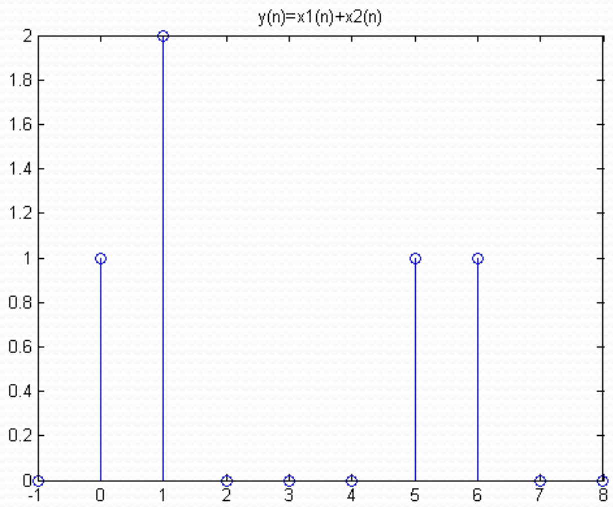
- Οι πράξεις εκτελούνται **ανά στοιχείο** και για την ίδια τιμή της ανεξάρτητης μεταβλητής

Παράδειγμα πράξης

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







x1(n) + x2(n) = y(n)

Ανάλυση Σημάτων

- Χρησιμοποιώντας της διακριτή ακολουθία δέλτα **δ(n)**, μπορούμε να αναλύσουμε ένα τυχαίο σήμα x(n).
- 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:

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

Ανάλυση Σημάτων

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

- Αυτό το άθροισμα μπορεί να γραφτεί περιληπτικά:

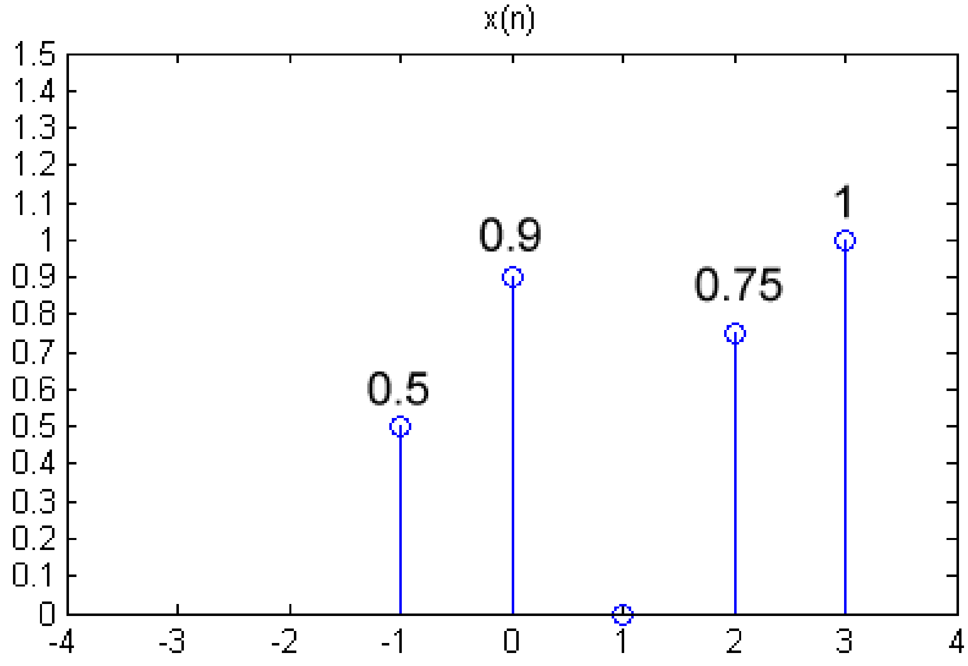
$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k)$$

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 n.

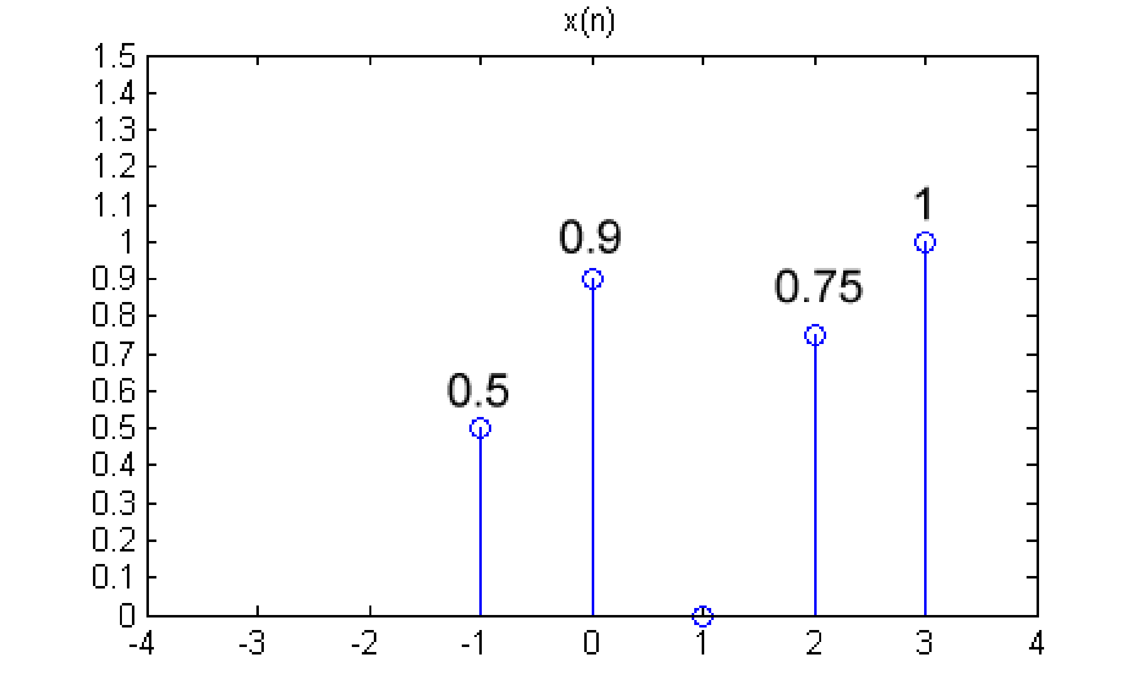
Ανάλυση Σημάτων

- Χρησιμοποιώντας την έκφραση ανάλυσης σημάτων που είδαμε, γράψτε στο χαρτί σε μορφή αθροισμάτων διακριτών συναρτήσεων δέλτα το παρακάτω σήμα:

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



Ανάλυση Σημάτων



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

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

Συστήματα Διακριτού Χρόνου

- Ένα σύστημα διακριτού χρόνου δέχεται μια είσοδο διακριτού χρόνου x(n) και παράγει μια έξοδο διακριτού χρόνου y(n), μετασχηματίζοντας το x(n).
- Το σήμα εισόδου μπορεί να ονομαστεί ως διέγερση ενώ το σήμα εξόδου ως απόκριση.

x(n) *διέγερση* → Σύστημα Διακριτού Χρόνου T[·] → y(n) *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 <u>είναι</u> αιτιατό:

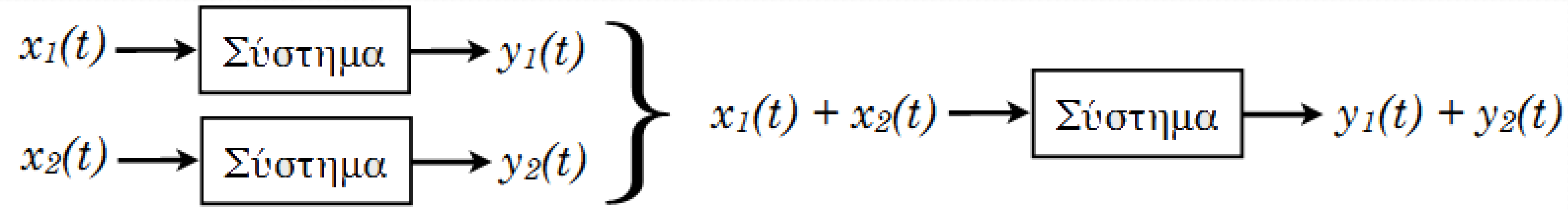
$$y(n) = \alpha x(n) - \beta x(n-1)$$

- Μη αιτιατό είναι το σύστημα στο οποίο η έξοδος τη χρονική στιγμή n_0 εξαρτάται από μελλοντικές χρονικές στιγμές στην είσοδο.

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.



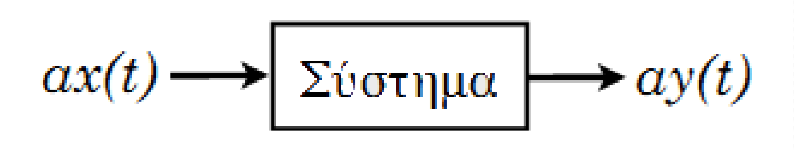
- Η αρχή της υπέρθεσης ορίζεται όταν σε ένα σύστημα ισχύει:

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

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.



- 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**.

- Άρα 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 n0 και παράγει μια έξοδο με την ίδια μετατόπιση n0, τότε το σύστημα ονομάζεται **Χρονικά Αμετάβλητο ή Αμετάβλητο στην μετατόπιση**.

$$x[n-n_0] \longrightarrow \boxed{\text{Σύστημα}} \longrightarrow y[n-n_0]$$

- Εάν ένα σύστημα είναι και Γραμμικό και Χρονικά αμετάβλητο θα το γράφουμε για συντομία ως: **ΓΧΑ** ή **LSI**, από τις λέξεις Linear Shift Invariant.

Κρουστική απόκριση (Impulse Response)

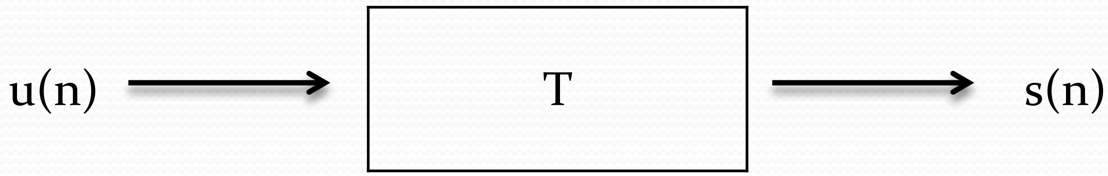
- Εάν σε ένα γραμμικό και χρονικά αμετάβλητο (ΓΧΑ) σύστημα η είσοδος είναι η μοναδιαία κρουστική ακολουθία δ(n), τότε το σήμα εξόδου(απόκριση) ονομάζεται **Κρουστική απόκριση h(n)**.

$$\delta(n) \longrightarrow \boxed{\;T\;} \longrightarrow h(n)$$

- 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 <u>βηματική</u> 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 x(n).
- 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 filter στο Matlab μας βοηθά να επιλύσουμε εξισώσεις διαφορών με στόχο να βρούμε την απόκριση του συστήματος:

  y=filter(b, α, x);

  όπου:

  b=[b0, 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.

- Το διάνυσμα y έχει ίδιο μήκος με το x.
- Επίσης πρέπει οπωσδήποτε $\alpha_0 \neq 0$.

Η Εντολή “inline” του Matlab

- Με την εντολή “inline” μπορούμε να ορίσουμε μια συνάρτηση δυναμικά, με την εξής σύνταξη:

  όνομα_συνάρτησης = inline('εκφραση');

- Για παράδειγμα, για να ορίσουμε μια συνάρτηση υπολογισμού τετραγώνου:

  ```
  >> mySqrt = inline('x^2')
  ```

- Για να την χρησιμοποιήσουμε:

  ```
  >> mySqrt(2)
  ```

  Το αποτέλεσμα θα είναι ans = 4.

- Η συνάρτηση αυτή μπορεί να χρησιμοποιηθεί προσωρινά. Αυτό σημαίνει πως όταν κλείσουμε το Matlab η συνάρτηση δεν θα υπάρχει.
- Πρέπει να αναφέρουμε ότι αντί του x θα μπορούσαμε να είχαμε γράψει οποιοδήποτε έγκυρο όνομα μεταβλητής.

Εξάσκηση

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

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

Το βελάκι δείχνει την τιμή για την χρονική στιγμή n=0

- Λύση:

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)

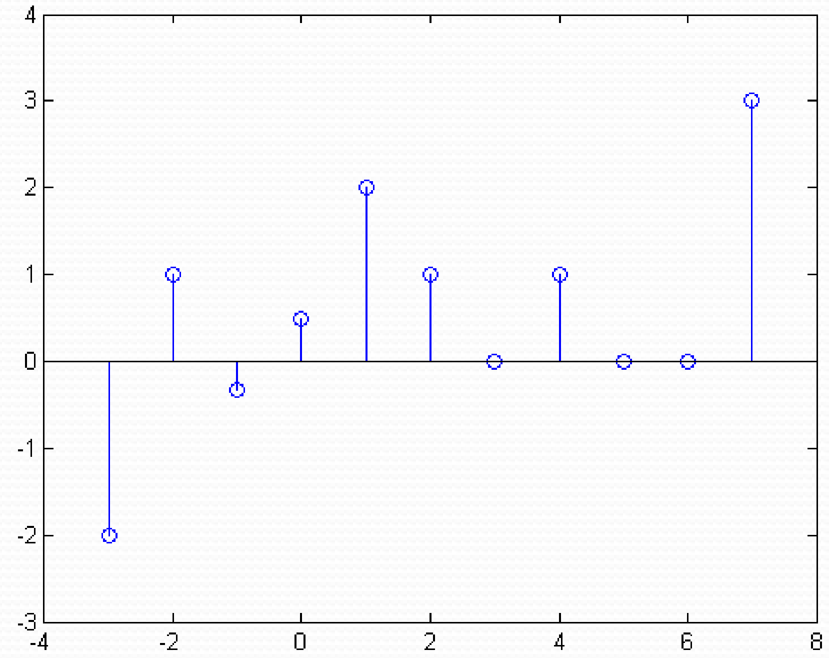
Εξάσκηση

- Για να υπολογιστεί και παρασταθεί γραφικά στο Matlab:

```
n=-3:7;
x=zeros(1,length(n));
d=inline('n==0'); %Synartisi delta

for i=1:length(n)
   x(i) = -2*d(n(i)+3) + d(n(i)+2) - 1/3*d(n(i)+1)
          + 1/2*d(n(i)) + 2*d(n(i)-1) + d(n(i)-2)
          + d(n(i)-4) + 3*d(n(i)-7);
end

stem(n,x);
```



Υπολογισμός Κρουστικής απόκρισης

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

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

- 1^{ος} 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:

  $h(0) = -0.9 * 0 + 1 = 1$
  $h(1) = -0.9 * h(0) + \delta(1) = -0.9 * 1 + 0 = -0.9$
  $h(2) = -0.9 * h(1) + \delta(2) = -0{,}9 * -0{,}9 + 0 = 0{,}81$
  $h(3) = -0.9 * h(2) + \delta(3) = -0{,}9 * 0{,}81 + 0 = -0{,}729$

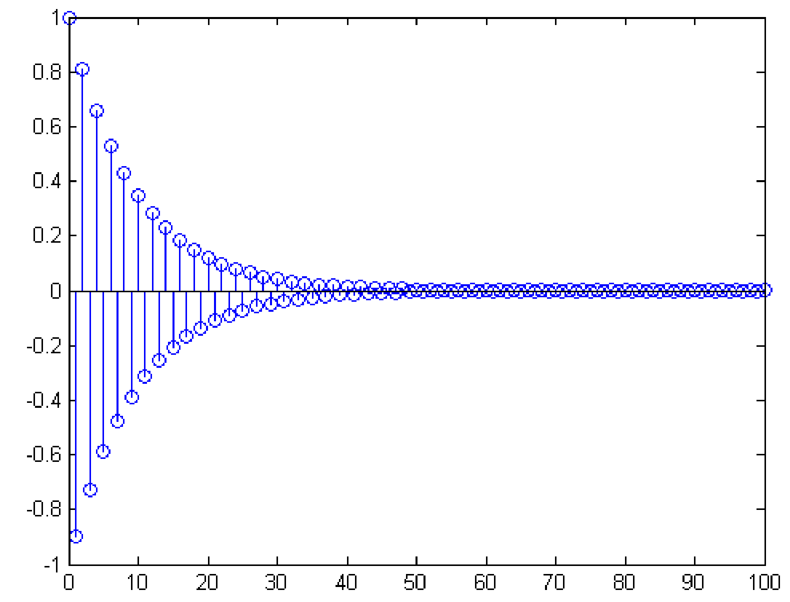
Υπολογισμός Κρουστικής απόκρισης

- $1^{ος}$ Τρόπος Λύσης (κώδικας Matlab), μια πιθανή λύση:

```
n=0:100;
h=zeros(1,length(n));
d=inline('n==0');

h(1)=1;
for i=2:length(n)
    h(i) = -0.9 * h(i-1) + d(n(i));
 end

stem(n,h);
```



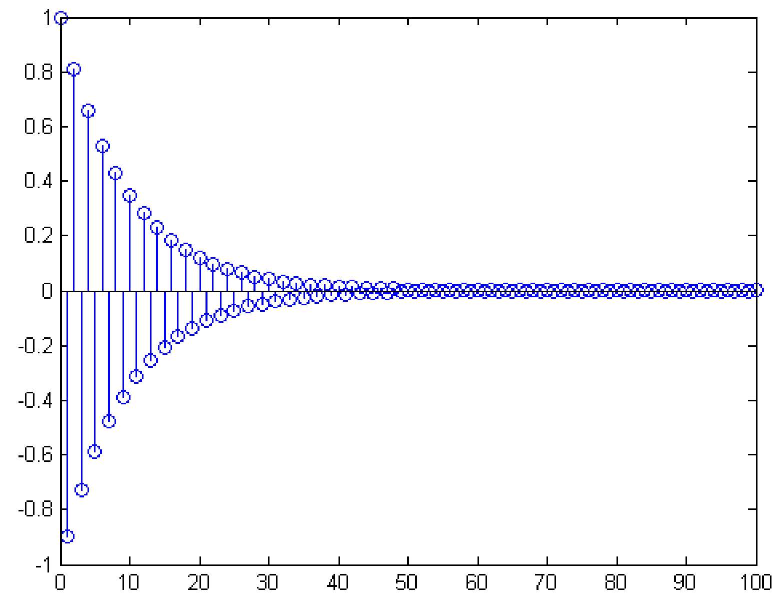
Υπολογισμός Κρουστικής απόκρισης

- 1ος Τρόπος Λύσης (κώδικας Matlab), μια άλλη πιθανή λύση:

```
d=inline('n==0');

h(1)=1;
for n=2:101
   h(n)= -0.9 * h(n-1) + d(n);
end

stem(0:100,h);
```



Υπολογισμός Κρουστικής απόκρισης

- Ο 2ος Τρόπος Λύσης είναι να χρησιμοποιήσουμε την συνάρτηση filter του Matlab.
- Αρχικά θα πρέπει να “βρούμε” τα a και b (συντελεστές της εξίσωσης διαφορών).
- Έχοντας υπόψη την γενική μορφή μιας γραμμικής εξίσωσης διαφορών με σταθερούς συντελεστές, χωρίζουμε τα y από τα x.
- Έτσι μπορούμε να βρούμε τους συντελεστές a και b.
- Όπου οι συντελεστές a αντιστοιχούν στους συντελεστές βάρους των y, ενώ οι συντελεστές b αντιστοιχούν στους συντελεστές βάρους των x.

Υπολογισμός Κρουστικής απόκρισης

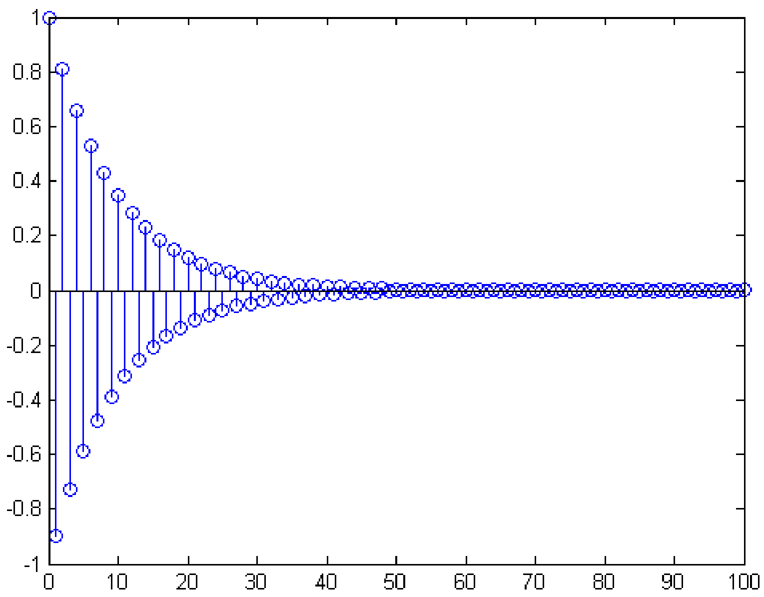
- y(n) = -0.9y(n-1) + x(n) =>

y(n) + 0.9y(n-1) = x(n)   χωρίζουμε τα y από τα x

- Άρα:

  a=[1, 0.9] και b=[1, 0]   βρίσκουμε τα a και b

```
n=0:100;
d=inline('n==0');
a=[1 0.9];
b=[1 0];
y=filter(b, a, d(n));
stem(n, y);
```



Υπολογισμός Βηματικής απόκρισης

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

  $y(n) = y(n-1) + x(n) – x(n-8)$

- 1ος Τρόπος Λύσης:

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

  Άρα: $s(n) = s(n-1) + u(n) – u(n-8)$

  Υπολογισμός αρχικών συνθηκών:

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

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

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

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

….

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

$s(8) = s(7) + u(8) – u(0) = 8 + 1 – 1 = 8$

$s(9) = s(8) + u(9) – u(1) = 8 + 1 – 1 = 8$

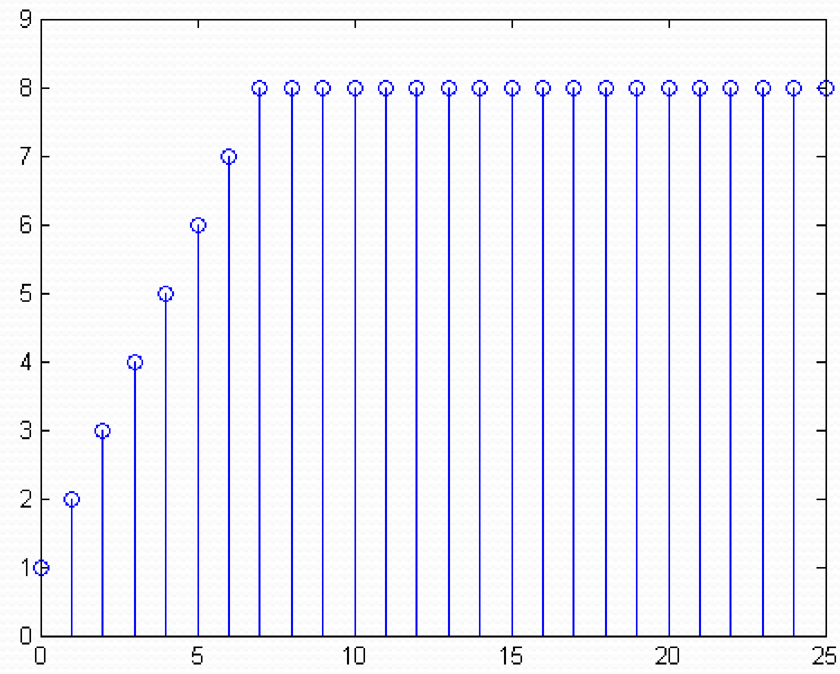
Υπολογισμός Βηματικής απόκρισης

- 1ος Τρόπος Λύσης (κώδικας Matlab), μια πιθανή λύση:

```
n=0:100;
s=zeros(1,length(n));
u=inline('n>=0');

s(1)=1;
for i=2:length(n)
    s(i)=s(i-1) + u(n(i)) - u(n(i)-8);
end

stem(n,s);
axis([0 25 0 9]);
```



Υπολογισμός Βηματικής απόκρισης

- Ο 2ος Τρόπος Λύσης χρησιμοποιώντας την συνάρτηση filter του Matlab.
- y(n) = y(n-1) + x(n) – x(n-8) =>

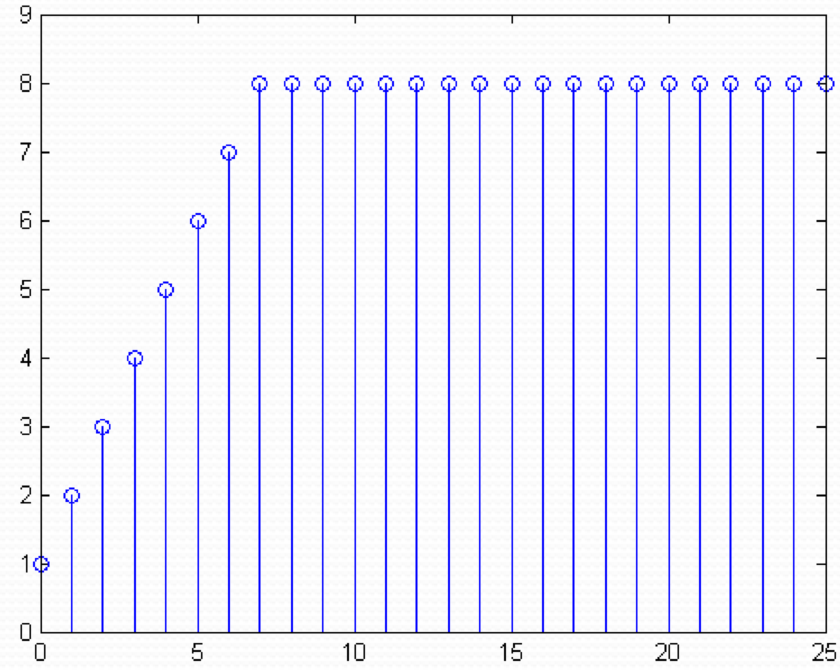
y(n) – y(n-1) = x(n) – x(n-8) χωρίζουμε τα y από τα x

```
n=0:100;
u=inline('n>=0');

a=[1 -1 0 0 0 0 0 0 0]; %apo to y
b=[1 0 0 0 0 0 0  0 -1]; %apo to x

y=filter(b, a, u(n));

stem(n, y);
axis([0 25 0 9]);
```

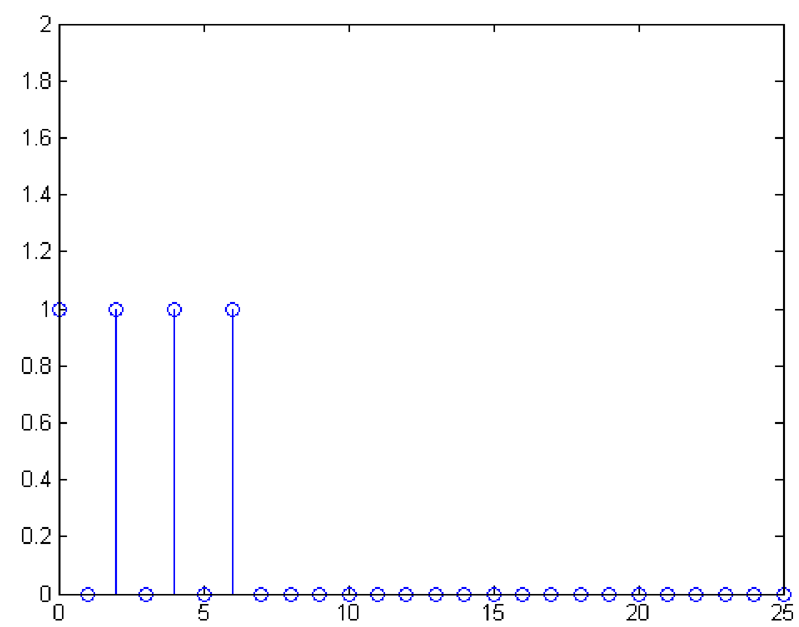


Εξάσκηση

- Σχεδιάστε την απόκριση του προηγούμενου συστήματος για είσοδο **$x(n) = a^n u(n)$**, για a=-1 και a=1.
- $y(n) = y(n-1) + x(n) – x(n-8)$
- Αντικαθιστώντας έχουμε: $y(n) = y(n-1) + a^n u(n) – a^{n-8} u(n-8)$

```
n=0:100;
s=zeros(1,length(n));
u=inline('n>=0');

a=-1;
s(1)=1;
for i=2:length(n)
   s(i) = s(i-1) + a^n(i)*u(n(i))
          - (a^(n(i)-8)*u(n(i)-8));
end
stem(n, s);
```

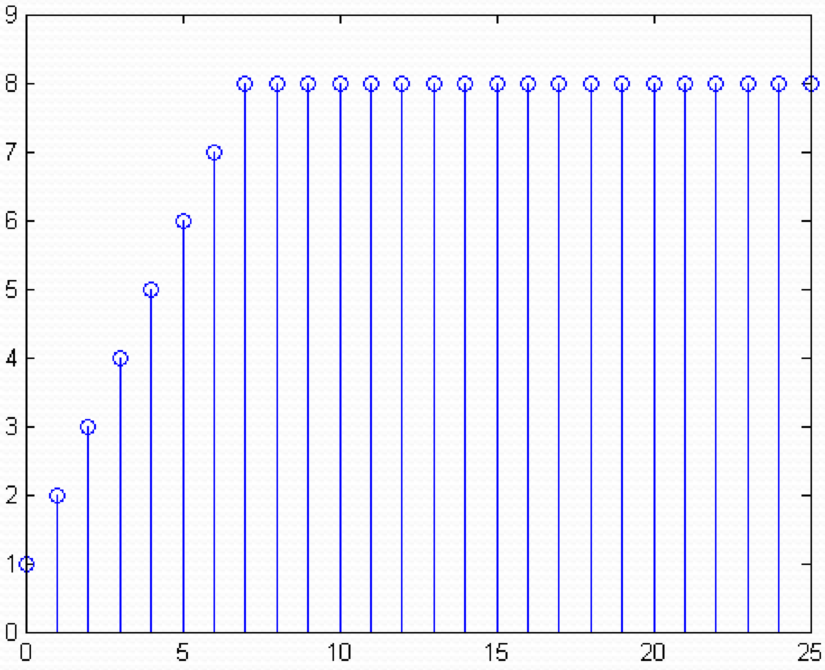


Εξάσκηση

- Για a=1, η υλοποίηση μας, μας επιτρέπει να αλλάξουμε απλά την τιμή της μεταβλητής.

```
n=0:100;
s=zeros(1,length(n));
u=inline('n>=0');

a= 1;
s(1)=1;
for i=2:length(n)
    s(i) = s(i-1) + a^n(i)*u(n(i))
            - (a^(n(i)-8)*u(n(i)-8));
end
stem(n, s);
```



Εξάσκηση (με συνάρτηση filter)

- Για να λύσουμε την άσκηση με την εντολή filter:
- $y(n) = y(n-1) + a^{n}\, u(n) - a^{n-8}\, u(n-8) \Rightarrow$

$y(n) - y(n-1) = a^{n}\, u(n) - a^{n-8}\, u(n-8)$ χωρίζουμε τα y από τα u

```
n=0:100;
u=inline('n>=0');
a=-1;

ac=[1 -1 0 0 0 0 0 0 0];
bc=[1 0 0 0 0 0 0 0 -1];
x=u(n).*(a.^n);

y=filter(bc, ac, x);
stem(n, y);
```

- Για a=1 απλά αλλάζουμε την τιμή της μεταβλητής

Απορίες - Ερωτήσεις ;