Σήματα και Συστήματα

Σήματα _Lab 2 & 3

Κώστας Μαριάς
Ρούλα Χατζάκη

2021

Σήματα Συνεχούς και Διακριτού Χρόνου

Τα σήματα διακρίνονται σε:

- Σήματα **συνεχούς χρόνου**, όπου ο χρόνος (t) είναι συνεχής μεταβλητή (**ανεξάρτητη**) με τιμές στο σύνολο των **πραγματικών αριθμών**. Το πλάτος του σήματος x(t) (εξαρτημένη μεταβλητή) μπορεί να πάρει τιμές σε ένα συνεχές πεδίο τιμών, ή να πάρει συγκεκριμένες τιμές από ένα διακριτό πεδίο τιμών.
  - t=t1:step:t2,

    όπου step το χρονικό διάστημα μεταξύ των δειγμάτων του χρόνου t. Αν το step είναι πολύ μικρό (πχ step=0.01) τότε το σήμα είναι συνεχούς χρόνου.

- Σήματα **διακριτού χρόνου**, όπου ο χρόνος (n) (**ανεξάρτητη**) δέχεται συγκεκριμένες τιμές στο σύνολο των **ακεραίων αριθμών**. Το πλάτος του σήματος x(n) (εξαρτημένη μεταβλητή) μπορεί επίσης να πάρει τιμές σε ένα συνεχές πεδίο τιμών, ή να πάρει συγκεκριμένες τιμές από ένα διακριτό πεδίο τιμών.
  - n=n1:step:n2,

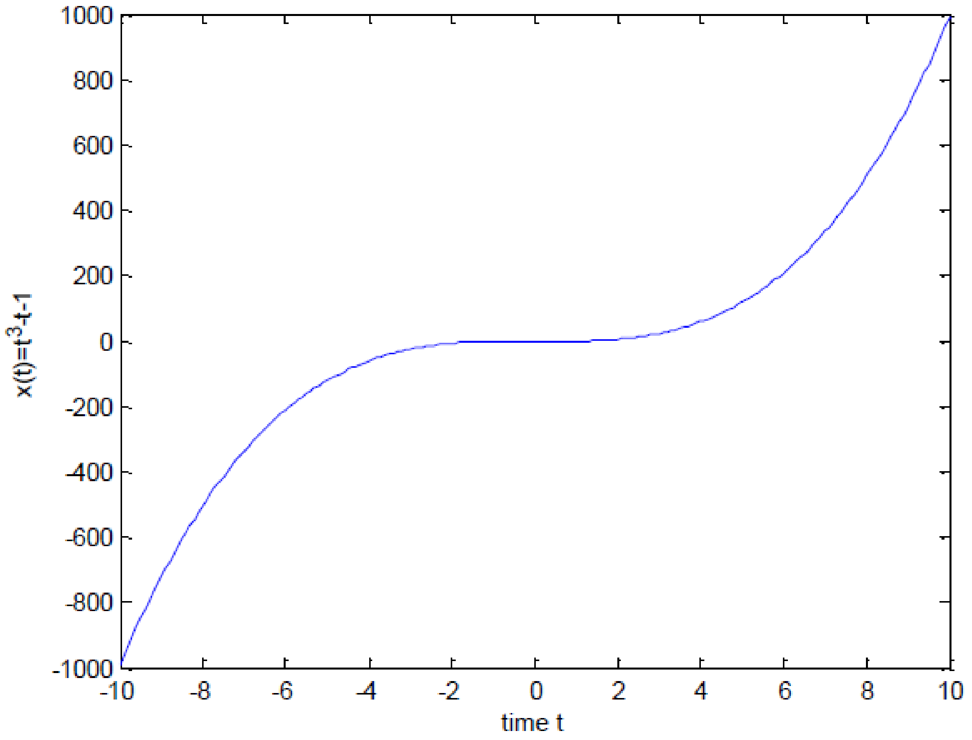
    όπου step το χρονικό διάστημα μεταξύ των δειγμάτων του χρόνου n. Αν το step είναι αρκετά μεγάλο(πχ step=1) τότε το σήμα είναι διακριτού χρόνου.

- Έτσι, με κριτήριο τον τύπο της ανεξάρτητης μεταβλητής (χρόνος) και της εξαρτημένης μεταβλητής (πλάτος), προκύπτουν οι τέσσερις κατηγορίες σημάτων που φαίνονται στον πίνακα:

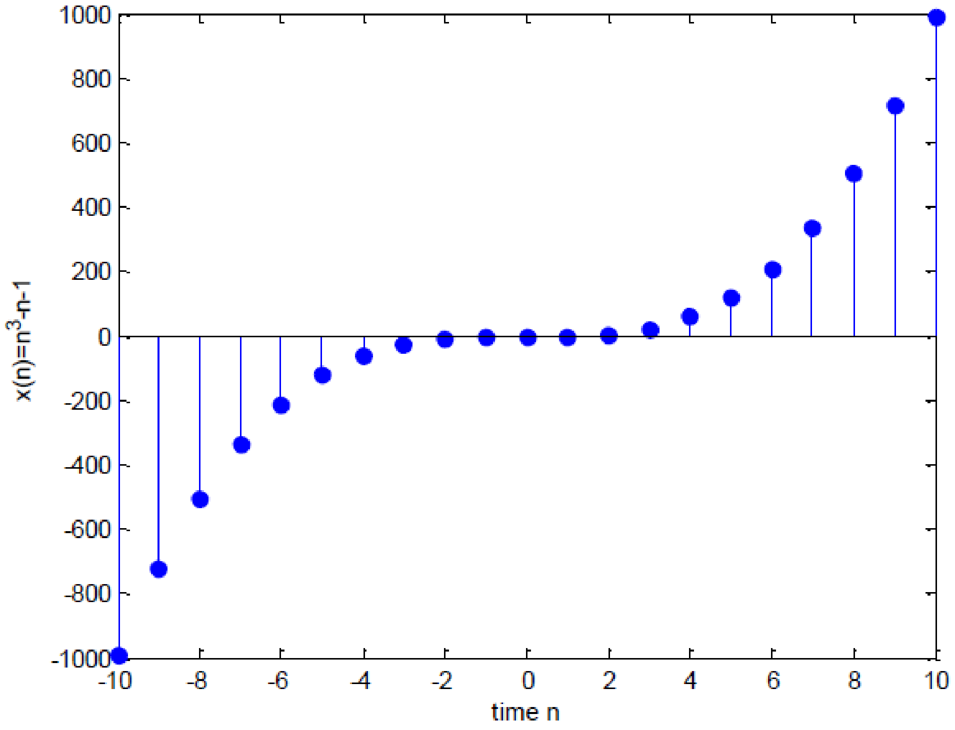
Σήμα	Χρόνος	Πλάτος
Διακριτού χρόνου	Διακριτός	Συνεχές
	Διακριτός	Διακριτό
Συνεχούς χρόνου	Συνεχής	Συνεχές
	Συνεχής	Διακριτό

Σήματα Συνεχούς και Διακριτού Χρόνου

Σήμα συνεχούς χρόνου



Σήμα διακριτού χρόνου



Αναπαράσταση σημάτων & Δειγματοληψία

Για να μπορέσουμε να αναπαραστήσουμε ένα αναλογικό σήμα στο Matlab/Octave θα πρέπει να εφαρμόσουμε κατάλληλη δειγματοληψία. Η συχνότητα ή ρυθμός δειγματοληψίας μετριέται σε Hertz και μας δείχνει πόσα δείγματα έχουν ληφθεί από τον δειγματολήπτη σε διάρκεια ενός δευτερολέπτου.

Εάν η συχνότητα δειγματοληψίας $\mathbf{f_s}$ είναι 1000Hz σημαίνει ότι ο δειγματολήπτης δημιουργεί 1000 δείγματα σε κάθε δευτερόλεπτο σήματος. Η χρονική απόσταση των δειγμάτων υπολογίζεται από τον τύπο $\mathbf{f_s} = \frac{\mathbf{1}}{\boldsymbol{Ts}}$, όπου το $\mathbf{T_s}$ είναι η περίοδος δειγματοληψίας.

Σύμφωνα με το θεώρημα δειγματοληψίας Nyquist-Shannon τα σήματα μπορούν να ανακατασκευαστούν πλήρως από την δειγματοληπτημένη μορφή τους εάν η συχνότητα δειγματοληψίας ($\boldsymbol{f_s}$) είναι μεγαλύτερη ή ίση από το διπλάσιο της μέγιστης συχνότητάς τους (**2*f*₀ *συχνότητα Nyquist***) , δηλαδή, $\boldsymbol{f_s \geq 2f_0}$.

Αναπαράσταση σημάτων & Δειγματοληψία

Για την αναπαράσταση σημάτων σε Matlab/Octave έχετε υπόψιν τα παρακάτω:

- Ένα συνεχές σήμα ***x*(*t*)** απεικονίζεται με το διάνυσμα **[*x*(0), *x*(1), … , *x*(*L*-1)]** μήκους ***L***, που προκύπτει από περιορισμό της διάρκειάς του στο διάστημα [0, *T*] και δειγματοληψία στα σημεία $\mathbf{t_n} = \boldsymbol{nTs}$, όπου ***Ts*** η περίοδος δειγματοληψίας.
- Η συχνότητα δειγματοληψίας ***fs*=1/*Ts*** επιλέγεται πολύ μεγαλύτερη από τη συχνότητα Nyquist, ώστε το δειγματοληπτημένο σήμα να «μοιάζει» με αναλογικό. Ένας πρακτικός κανόνας είναι 10 φορές η μέγιστη συχνότητα του σήματος για μια καλή αναπαράσταση του αρχικού και 100 φορές για μία «όμοια».[Το **βήμα** που χρησιμοποιούμε σε έναν αλγόριθμο στο Matlab/Octave για την αναπαραγωγή του σήματος είναι η περίοδος δειγματοληψίας ***Ts***]
- Η διάρκεια ***T* = *LTs*** του σήματος επιλέγεται αρκετά μεγάλη ώστε να έχουμε ικανοποιητική πληροφορία.
- Στο πεδίο συχνότητας το σήμα απεικονίζεται με διάνυσμα [***X*(0), *X*(1), … , *X*(*N*-1)]** μήκους ***N*** που αντιστοιχεί σε *N*-ισαπέχουσες διακριτές συχνότητες στο διάστημα **[0, *fs*]** με ανάλυση ***fo*=*fs*/*N*=1/(*NTs*).**

Αναπαράσταση σημάτων & Δειγματοληψία

Παράδειγμα 1:

Να βρεθεί η ελάχιστη δειγματοληπτική συχνότητα Nyquist fs του αναλογικού σήματος:

- $S(t) = 10 + 5\sin(1000\pi t) + 15\sin(2000\pi t) + 5\sin(3000\pi t)$

Λύση:

Από την θεωρία είδαμε ότι:

$$s(t) = \underbrace{A}_{\text{Πλάτος}} \sin(\underbrace{\omega_0}_{\text{Κυκλική συχνότητα}} t + \underbrace{\varphi_0}_{\text{Αρχική φάση}})$$

όπου,

$$\omega_0 = 2\pi \underbrace{f_0}_{\substack{\text{Συχνότητα}\\(\text{Hz})}} = \frac{2\pi}{\underbrace{T_0}_{\substack{\text{Περίοδος}\\(\text{sec})}}}$$

$S(t) = 10 + 5\sin(1000\pi t) + 15\sin(2000\pi t) + 5\sin(3000\pi t)$

Λύνοντας ως προς f,την σχέση ω=2πf

$f_1 = 0$ $f_2 = 500$ $f_3 = 1000$ $f_4 = 1500$ Μέγιστη συχνότητα σήματος fmax

Άρα, για να έχω fs>2fmax => **fs>3000Hz**

Βασικές εντολές γραφικών παραστάσεων σε Matlab/Octave:

» figure(1) %δημιουργία εικόνας 1

» clf % καθαρίζει την εικόνα από προηγούμενη χρήση. Αν θέλουμε να διαγράψουμε όλες τις εικόνες χρησιμοποιούμε την close all

» plot(x,t) %δισδιάστατη απεικόνιση της γραφικής παράστασης της μεταβλητής x σε συνάρτηση με τη μεταβλητή t

» subplot(x,t,z) %περισσότερες από μία γραφικές παραστάσεις στην οθόνη

» title(΄…. ΄) %τοποθετούμε τίτλο στη γραφική παράσταση

» xlabel(΄…΄) %τοποθετούμε την αντίστοιχη ονομασία της μεταβλητής στον άξονα x

» ylabel(΄….΄) %τοποθετούμε την αντίστοιχη ονομασία της μεταβλητής στον άξονα y

» xlim([xmin xmax]) % θέτουμε κατώτατα και ανώτατα όρια τιμών για τον άξονα x

» ylim([ymin ymax]) % θέτουμε κατώτατα και ανώτατα όρια τιμών για τον άξονα y

» axis([xmin xmax ymin ymax]) % (εναλλακτικά του xlim,ylim) καθορίζεται η βαθμολογία των αξόνων

Σήματα συνεχούς χρόνου

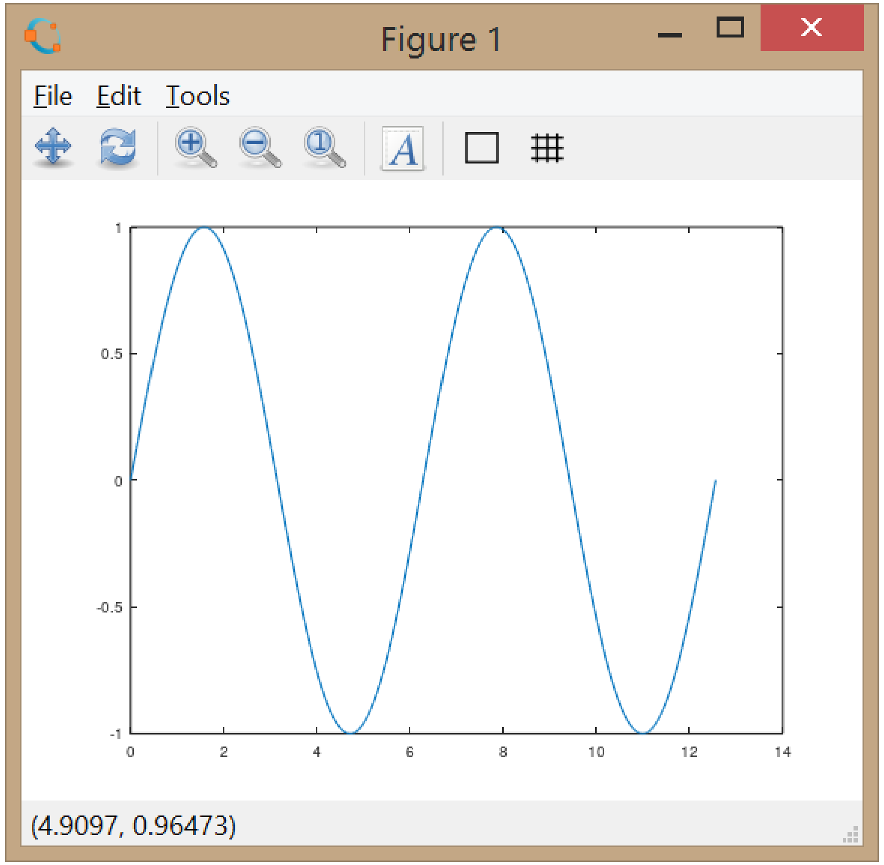
<u>Παράδειγμα 2:</u>

Δημιουργία της γραφικής παράστασης του ημιτονοειδούς σήματος **sin(x)** στο διάνυσμα χρόνου **0 ≤x≤ 4π** και 100 δείγματα.

- x=0:pi/100:4*pi; % χρονικές στιγμές δειγματοληψίας απο 0 εως 4π με βήμα = π/100

%Το παραπάνω μπορεί να γραφτεί και με την χρήση της εντολής linspace ως x=linspace(0,4*pi,100);

- y=sin(x);
- plot(x,y)



Σήματα συνεχούς χρόνου

Παράδειγμα 2 συνέχεια:

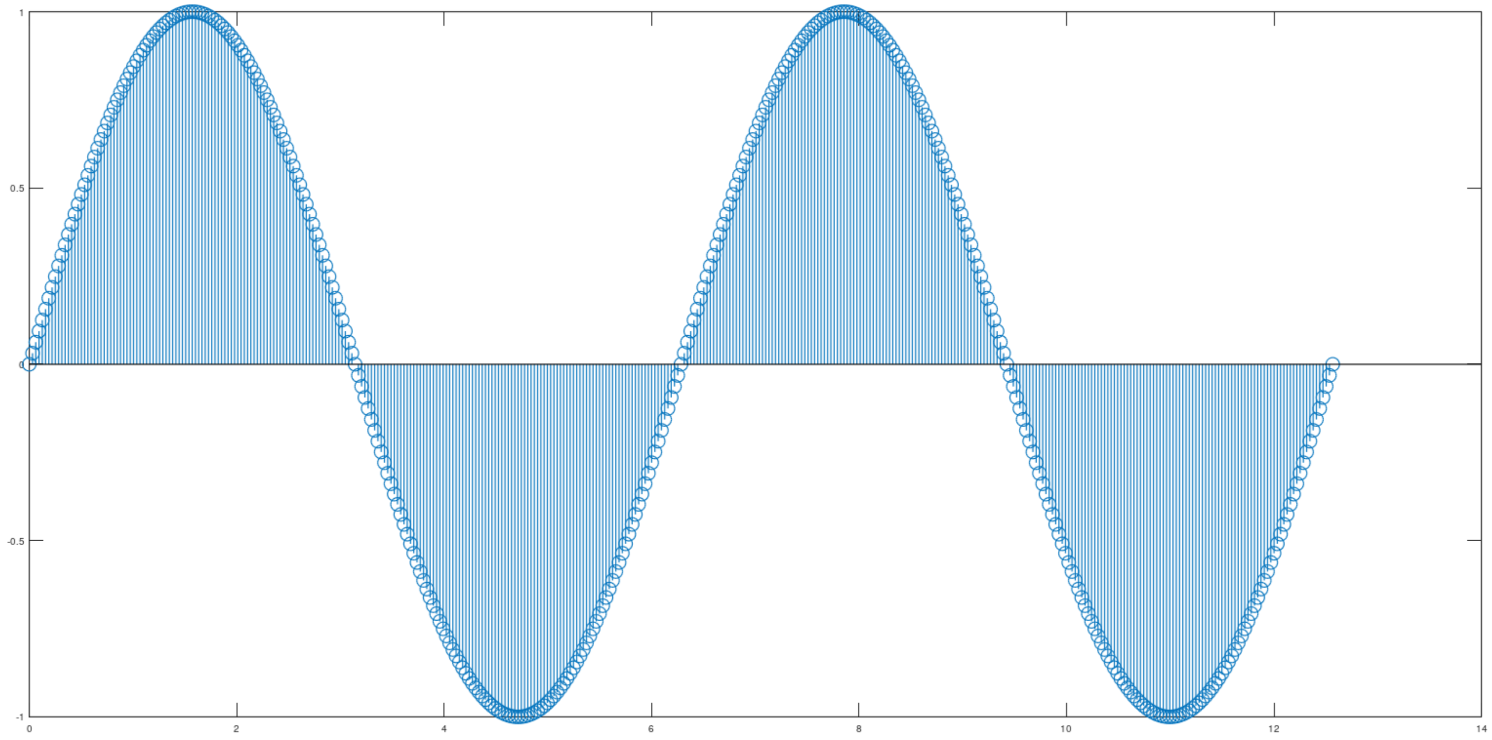
Αν στο παραπάνω παράδειγμα αντικαταστήσουμε την εντολή

» plot(x,y)

με την εντολή

» stem(x,y)

Θα έχουμε αυτήν την μορφή του γραφήματος.



Σήματα συνεχούς χρόνου

Παράδειγμα 3 :

Δημιουργήστε στο Matlab/Octacve την γραφική παράσταση της συνάρτησης **x = sin(2 π f t)** για την οποία ισχύει **0 ≤ t ≤ 0,05** και **f = 100 Hz**.

Λύση:

Με την χρήση της συνάρτησης **plot(x,t)** μπορούμε να κάνουμε τη γραφική παράσταση της μεταβλητής **x** σε συνάρτηση με τη μεταβλητή **t**.

Για να προχωρήσουμε στον αλγόριθμο θα πρέπει να βρούμε πρώτα την περίοδο δειγματοληψίας του σήματος. Ακολουθώντας την οδηγία για $f_s = 10 * f_{max}$, έχουμε $f_s = 10 * 100 = 1000Hz$. Οπότε, $T_s = 1/ f_s = 1/ 1000 = 0,001$.

Οπότε το βήμα για τον αλγόριθμο θα είναι 0,001.

Σήματα συνεχούς χρόνου

Συνέχεια λύσης παραδείγματος 3 :

Στον editor πληκτρολογήστε:

» clear all % διαγραφή του χώρου εργασίας

» close all % κλείσιμο όλων των γραφικών παραστάσεων

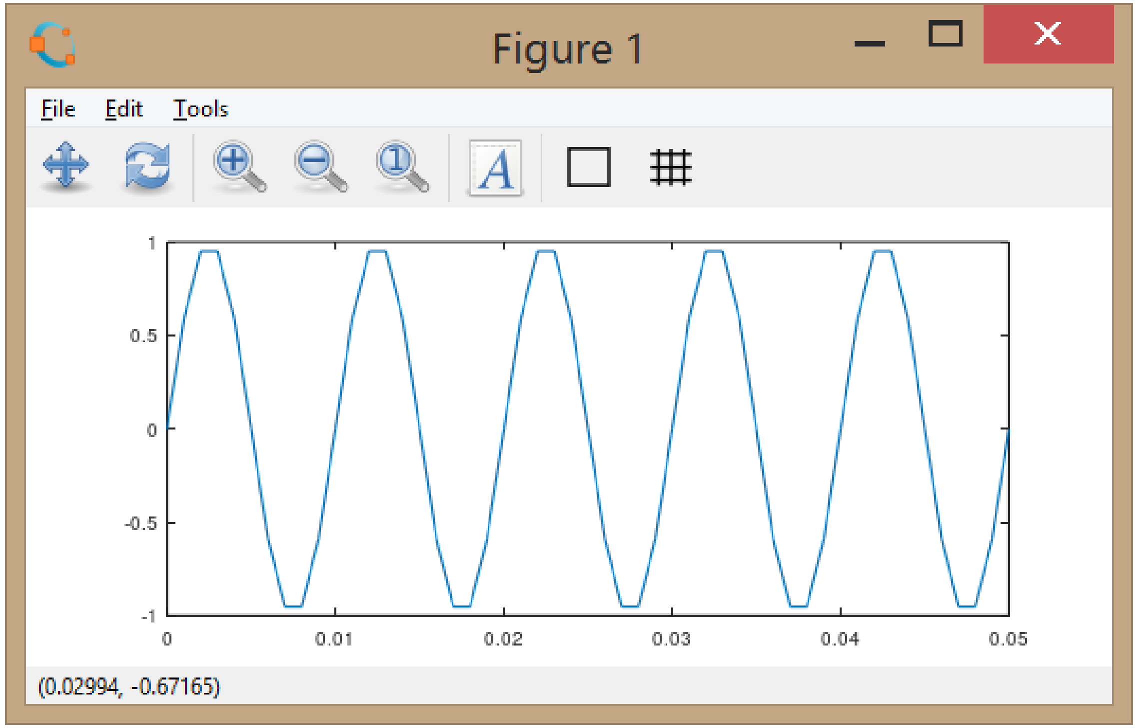
» clc % εκκαθάριση του παραθύρου εντολών

» f=100 % συχνότητα δειγματοληψίας 100 Hz

» t=0:0.001:0.05; % χρονικές στιγμές δειγματοληψίας απο 0 εως 0.5 με βήμα 0.001

» x=sin(2*pi*f*t); % διάνυσμα σήματος

» plot(t,x) % γραφική παράσταση συναρτήσει του χρόνου

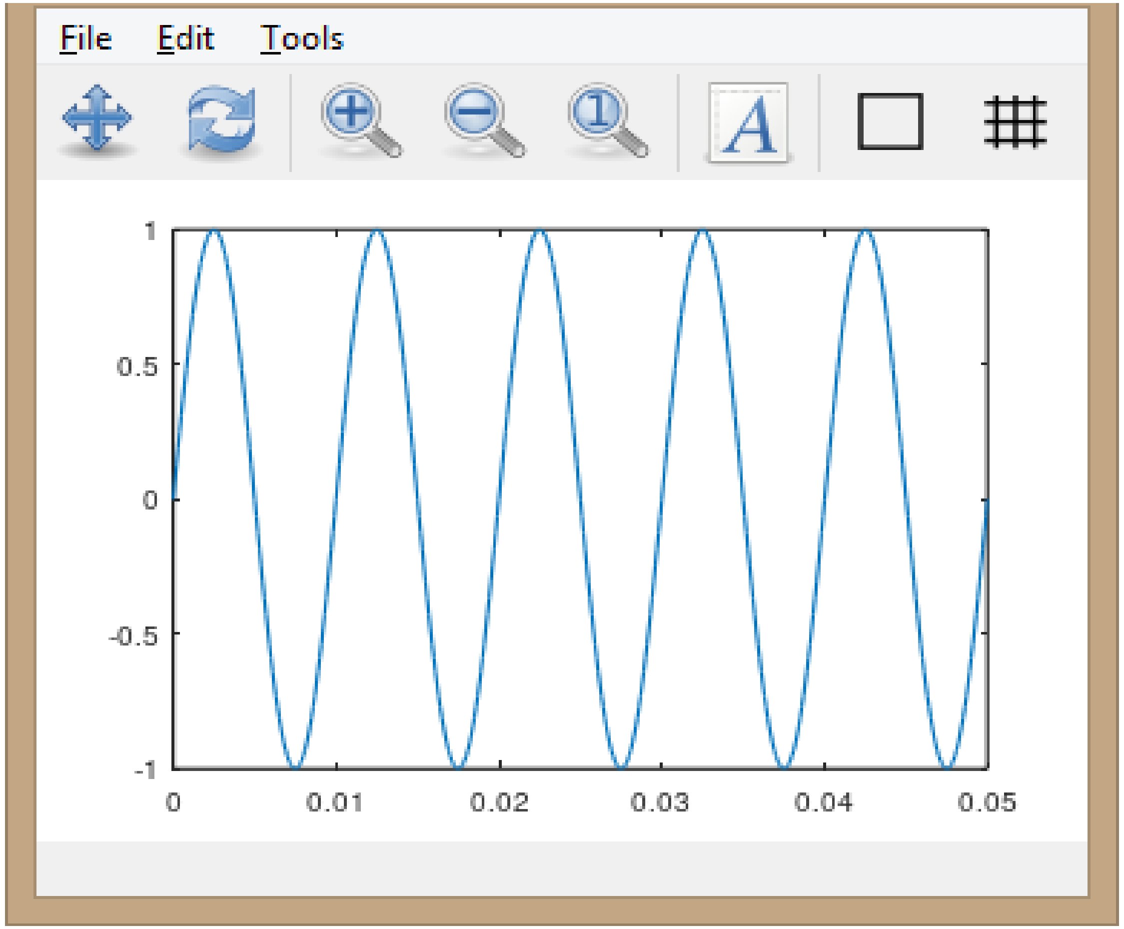


Σήματα συνεχούς χρόνου

Συνέχεια λύσης παραδείγματος 3 :

Από την γραφική αναπαράσταση του σήματος φαίνεται ότι η περίοδος δειγματοληψίας είναι μεγάλη.

Δείτε πως βελτιώνεται η μορφή του σήματος αν χρησιμοποιήσουμε βήμα 0.0001 αντί 0.001.



Σήματα συνεχούς χρόνου

Παράδειγμα 4

Δημιουργήστε τις γραφικές παραστάσεις των x=sin(ωt) με f=100Hz, και y=0.5cos(ωt) με f=300Hz στο διάνυσμα χρόνου 0 ≤t≤ 0.05 και βήμα 0.0001.

» figure(1); clf

» t=0:0.0001:0.05; % % χρονικές στιγμές δειγματοληψίας από 0 εως 0.05 με βήμα 0.0001

» x=sin(2*pi*100*t); διάνυσμα σήματος x

» y=0.5*cos(2*pi*300*t); διάνυσμα σήματος y

» subplot(2,1,1), plot(t,x); πολλαπλές γραφικές παραστάσεις σε μια οθόνη (2 γραμμές, 1στήλη, 1ο γράφημα)

» title('plot sin') % τίτλος πρώτου γραφήματος

» axis([0,0.05,-1,1]); % κατώτατα και ανώτατα όρια αξόνων x,y

» xlabel('t'); % ονομασία αξόνα x 2ου γραφήματος

» ylabel('sin(2pift)'); % ονομασία αξόνα y 2ου γραφήματος

» subplot(2,1,2), plot(t,y); πολλαπλές γραφικές παραστάσεις σε μια οθόνη (2 γραμμές, 1στήλη, 2ο γράφημα)

» title('plot cos') % τίτλος δεύτερου γραφήματος

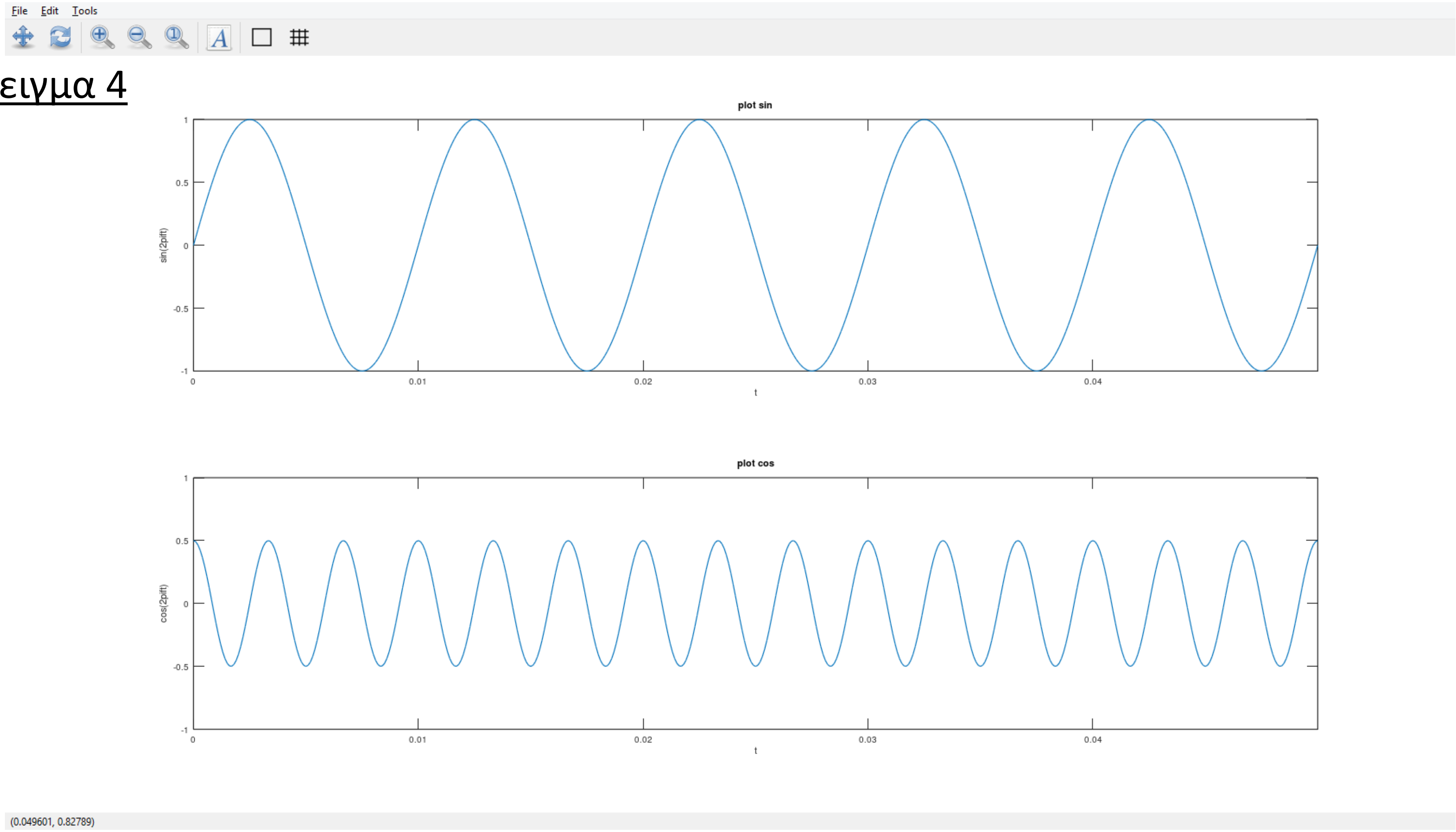
» axis([0,0.05,-1,1]); % κατώτατα και ανώτατα όρια αξόνων x,y 2ου γραφήματος

» xlabel('t'); % ονομασία αξόνα x 2ου γραφήματος

» ylabel('cos(2pift)'); % ονομασία αξόνα y 2ου γραφήματος

Σήματα συνεχούς χρόνου

Παράδειγμα 4



Σήματα συνεχούς χρόνου

» **Παράδειγμα 5**

Έχουμε τα αναλογικά σήματα x1=cos(6πt), x2=cos(14πt), x3=cos(26πt), δημιουργήστε τις γραφικές παραστάσεις των σημάτων στο διάστημα χρόνου [0,1] καθώς και το άθροισμα αυτών x=x1+x2+x3, με fs=1000Hz.

Λύση

» clear; % καθαρισμός μνήμης

» clf; % κλείσιμο παράθυρων εικόνων

» f=1000; % συχνότητα δειγματοληψίας 1000 Hz

» Ts=1/f; % περίοδος δειγματοληψίας

» t=0:Ts:1; % χρονικές στιγμές δειγματοληψίας απο 0 εως 1 με βήμα 0.001

» x1=cos(6*pi*t); % διάνυσμα σήματος x1

» x2=cos(14*pi*t); % διάνυσμα σήματος x2

» x3=cos(26*pi*t); % διάνυσμα σήματος x3

» x=x1+x2+x3; % διάνυσμα σήματος

» figure(1); % παράθυρο εικόνων 1

» subplot(4,1,1); % πολλαπλά γραφήματα, 4γραμμές, 1 στήλη, 1ο γράφημα το επόμενο Plot

» plot(t,x1); % γραφική παράσταση x1

» title('x1=cos(6*pi*t)'); % τίτλος πρώτου γραφήματος

» subplot(4,1,2), plot (t,x2);

» title('x2=cos(14*pi*t)');

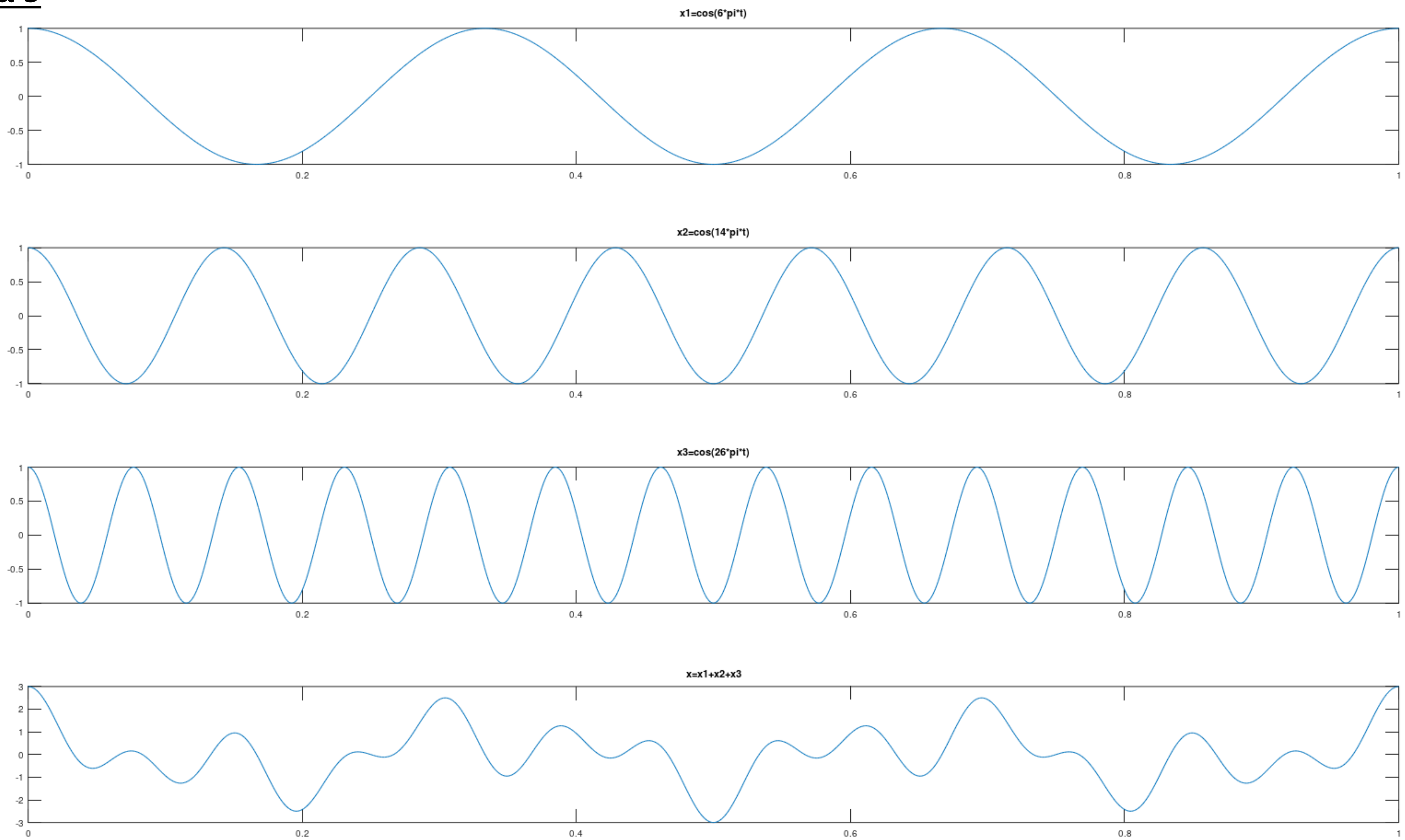
» subplot(4,1,3), plot (t,x3);

» title('x3=cos(26*pi*t)');

» subplot(4,1,4), plot (t,x);

» title(' x=x1+x2+x3 ');

Παράδειγμα 5



Σήματα συνεχούς χρόνου

» **Παράδειγμα 5 συνέχεια**

Μπορούμε επίσης να ανακατασκευάσουμε το σήμα ως ένα άθροισμα από μετατοπισμένα *sincs (πληκτρολογήστε help sinc στο command window για λεπτομέρειες)* πολλαπλασιασμένα το κάθε ένα με την τιμή του δείγματος του σήματος.

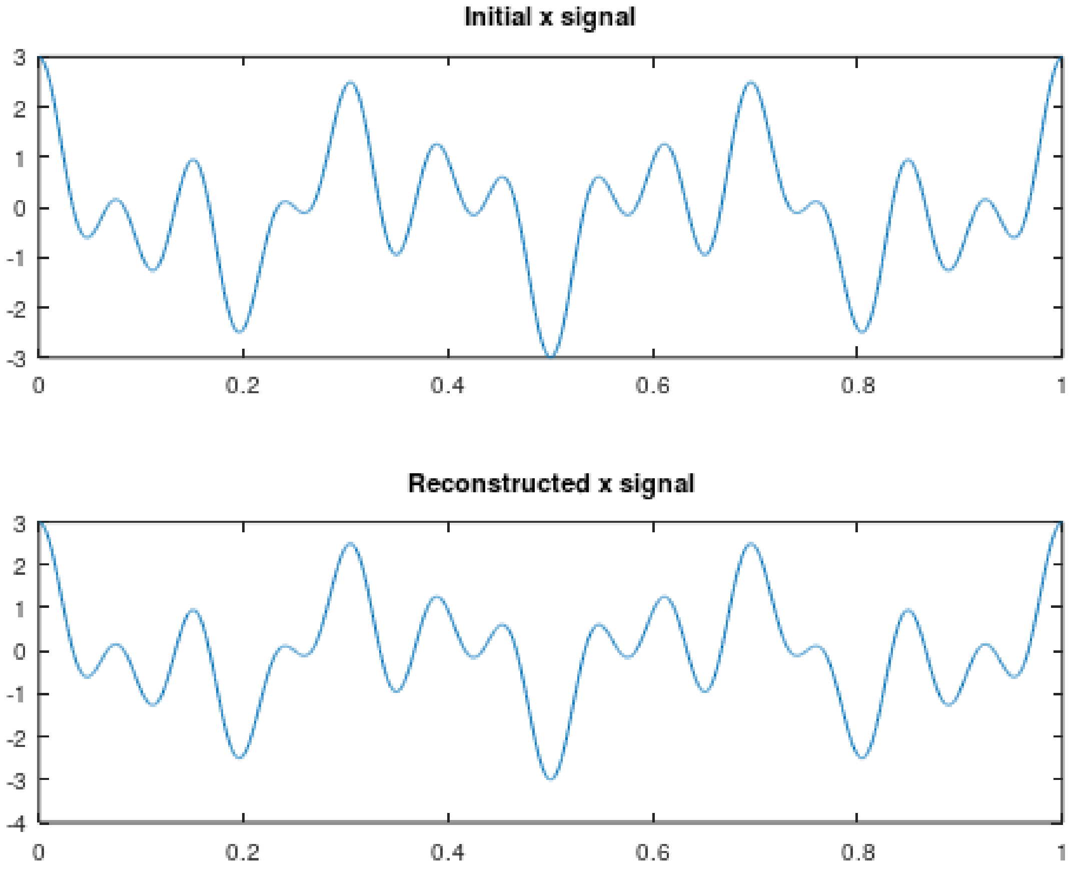
$$\operatorname{sinc} x = \frac{\sin(\pi x)}{\pi x}.$$

Συνεχίστε το πρόγραμμα σας πληκτρολογώντας:

```
» for i=1:length(x)
»   reconstructedx(i)=sum(x.*sinc(f*(t(i)- t)));
» end
» figure(2);
» subplot(2,1,1), plot (t,x);
» title('Initial x signal');
» subplot(2,1,2), plot (t,reconstructedx);
» title(' Reconstructed x signal ');
```

Σήματα συνεχούς χρόνου

Όπως φαίνεται στην εικόνα το αρχικό και το ανακατασκευασμένο σήμα είναι ίδια. Οπότε δεν έχει χαθεί πληροφορία. Επαναλάβετε το παράδειγμα δειγματοληπτώντας με συχνότητα μικρότερη από την Nyquist π.χ 10Hz και δείτε πώς το σήμα είναι τελείως διαφορετικό.



Σήματα διακριτού χρόνου

- Ένα διακριτό σήμα δηλώνεται με x(n), όπου η μεταβλητή n λαμβάνει ακέραιες τιμές και παριστάνει διακριτές χρονικές στιγμές. $x(n) = \{x(n)\} = \{ K, x(-1), x(0), x(1), K\}$

Στο Matlab/Otave μπορούμε να παραστήσουμε μια πεπερασμένης διάρκειας ακολουθία με ένα διάνυσμα γραμμή.

» n =[−3,− 2,−1,0,1,2,3];

- Κάποια από τα βασικά σήματα διακριτού χρόνου είναι:

  - Κρουστικό σήμα $$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & \alpha\lambda\lambda o\acute{u} \end{cases}$$
  - Μοναδιαίας κλίμακας $$r[n] = \begin{cases} n, & n \geq 0 \\ 0, & \alpha\lambda\lambda o\acute{u} \end{cases}$$
  - Μοναδιαίου βήματος $$u[n] = \begin{cases} 1, & n \geq 0 \\ 0, & \alpha\lambda\lambda o\acute{u} \end{cases}$$
  - Εκθετικό σήμα $$e[n] = a^{|n|}$$

Σήματα διακριτού χρόνου

Παράδειγμα 6

❖Κρουστικό σήμα δ[n]

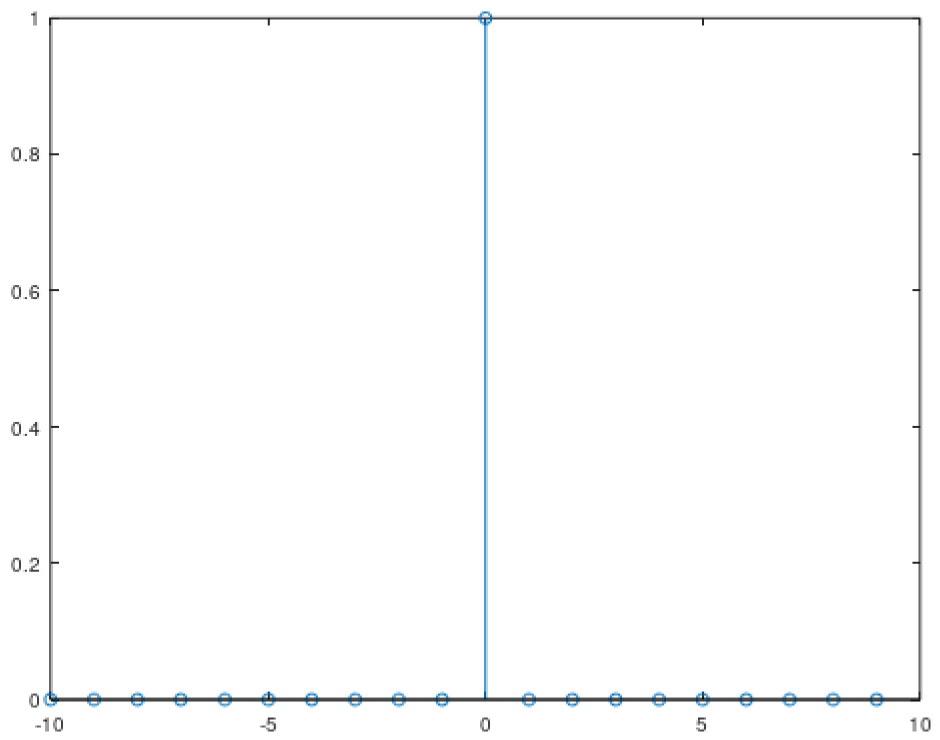
» n =[−10:9]; %Διάνυσμα χρόνου

» d =[(n ==0)]; %όταν το n==0 τότε παίρνω 1

» axis ([-10 9 -0,5 1,5]); % όρια αξόνων

» stem (n,d); % αναπαράσταση σήματος

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & \alpha\lambda\lambda o\acute{\upsilon} \end{cases}$$



Σήματα διακριτού χρόνου

<u>Παράδειγμα 7</u>

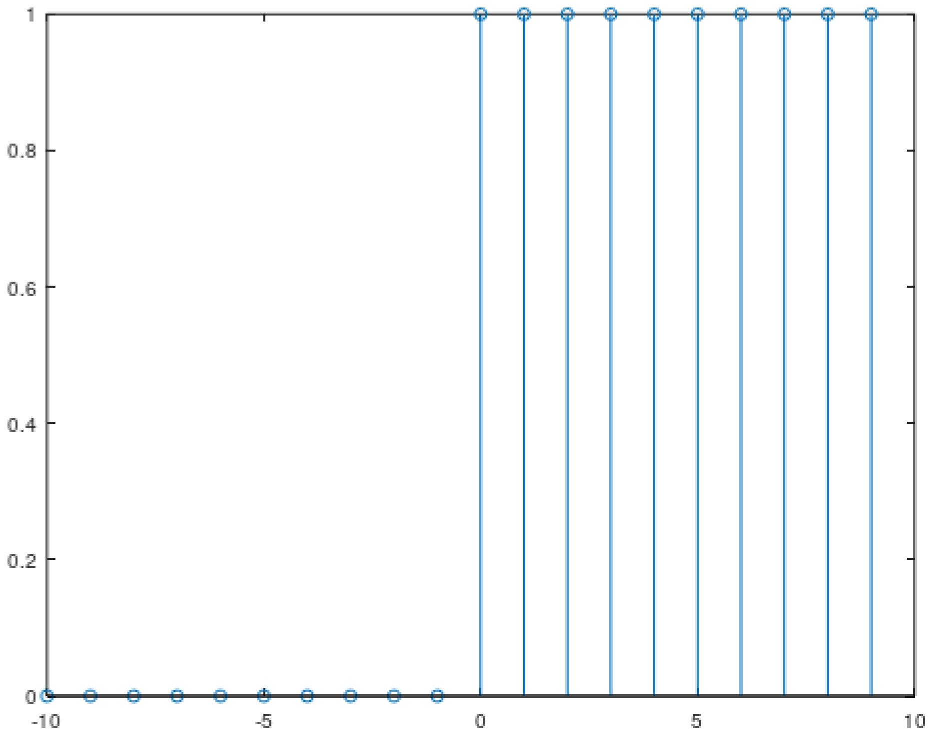
❖Μοναδιαίο βήμα u[n]

```
» n =[-10:9];
» u1 =[(n>=0)];
» axis ([-10 9 -0,5 1,5]);
» stem (n,u1);
```

Στον παραπάνω κώδικα η γραμμή *u=[(n>=0)];*
 μπορεί να αντικατασταθεί από:

```
» for i= 1:length(n)
»   if n(i)<0
»     u1(i)=0;
»   else
»     u1(i)=1;
»   end
» end
```

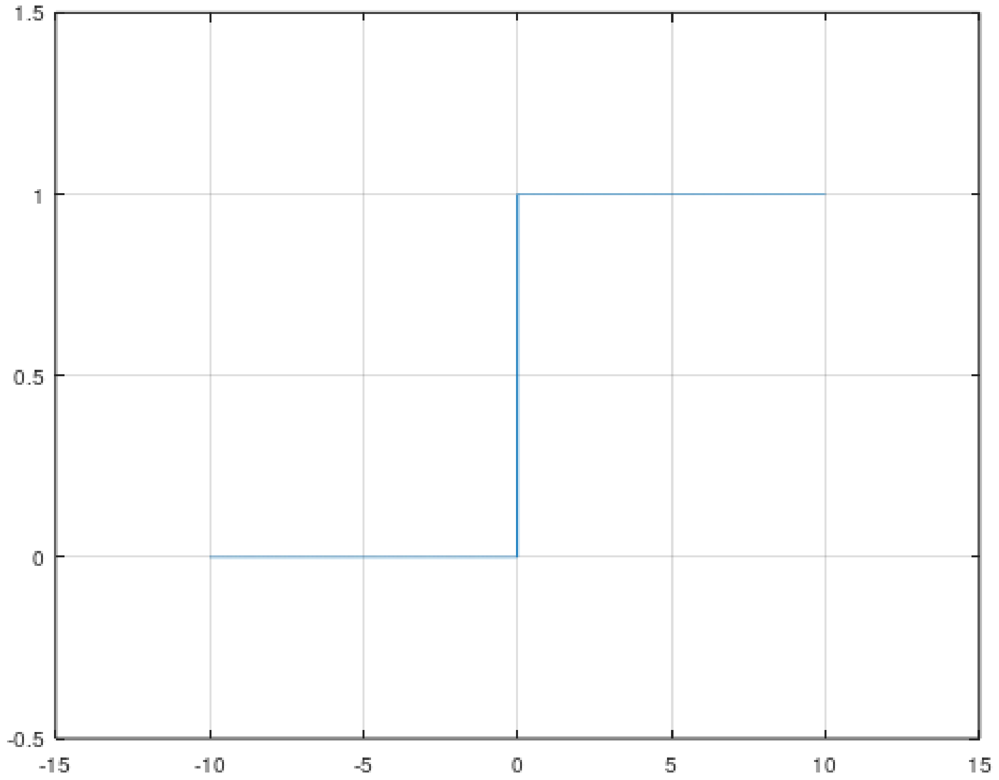
$$u[n] = \begin{cases} 1, & n \geq 0 \\ 0, & \alpha\lambda\lambda\text{ού} \end{cases}$$



Μοναδιαίο βήμα συνεχούς χρόνου

Παράδειγμα 8

- Μοναδιαίο βήμα συνεχούς χρόνου y[t]
  - » clear all;
  - » clf;
  - » t =[-10:0.01:10];
  - » y=[(t>=0)];
  - » figure(1);
  - » plot(t,y)
  - » grid
  - » xlim([-15 15]);
  - » ylim([-0.5 1.5]);



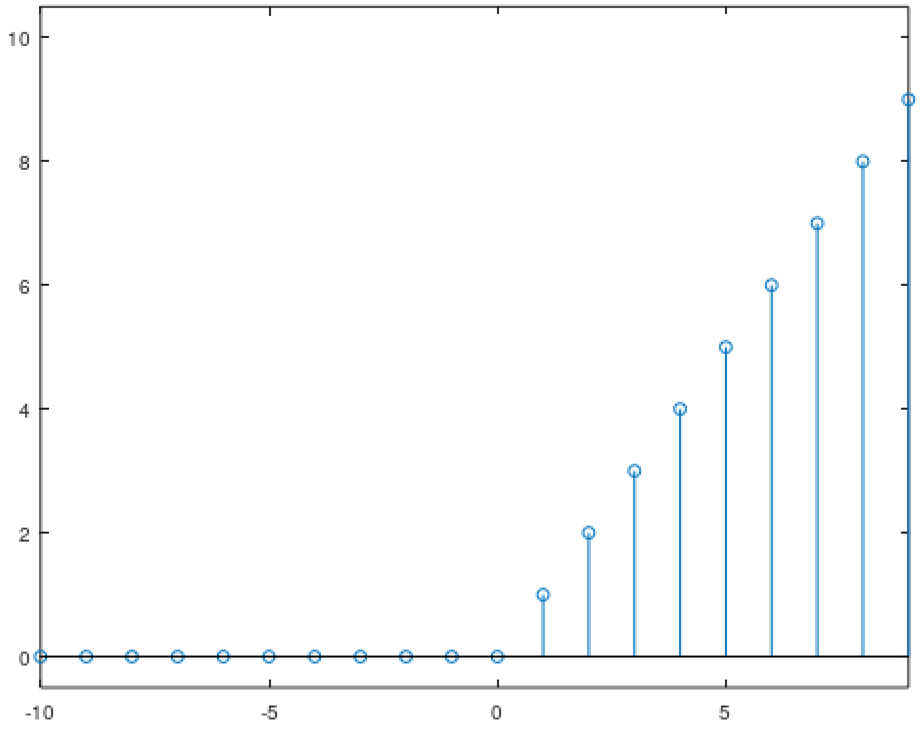
Σήματα διακριτού χρόνου

Παράδειγμα 9

❖Μοναδιαία κλίμακα r[n]

```
» n = [-10:9];
» r = n.*(n>=0);
» figure(1)
» stem(n,r);
» axis([-10 9 -0.5 10.5])
```

$$r[n] = \begin{cases} n, & n \geq 0 \\ 0, & \alpha\lambda\lambda o\acute{\upsilon} \end{cases}$$



Στον παραπάνω κώδικα η γραμμή *r = n.*(n>=0);*
 μπορεί να αντικατασταθεί από:

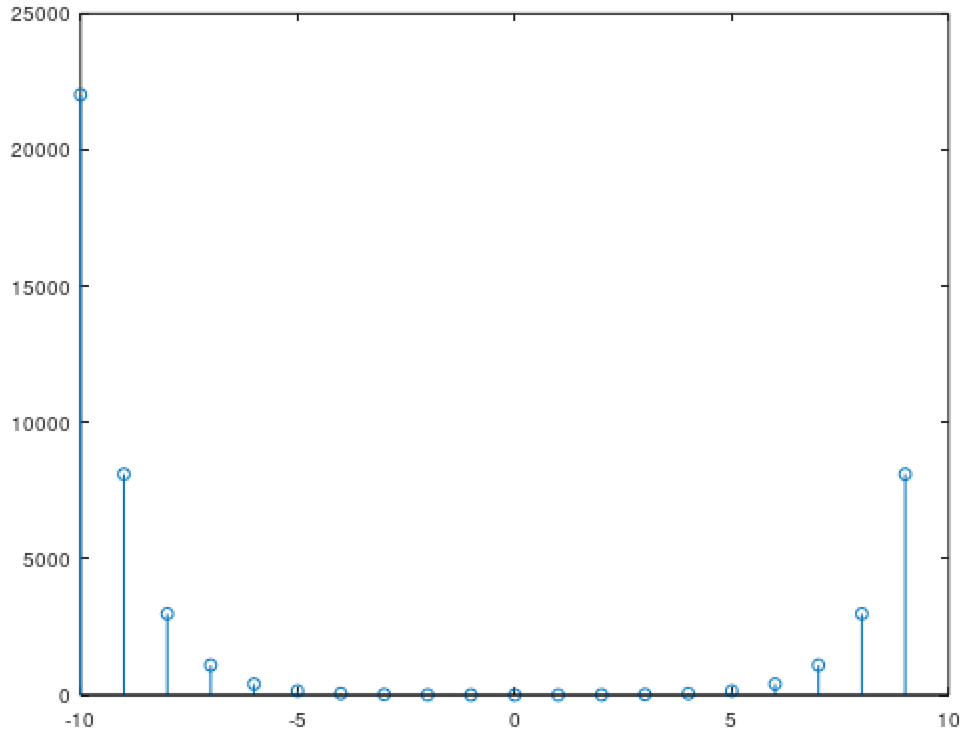
```
» for i = 1:length(n)
»    if n(i)<0
»       r(i) = 0;
»    else
»        r(i)=n(i);
»     end
» end
```

Σήματα διακριτού χρόνου

<u>Παράδειγμα 10</u>

- Εκθετικό σήμα e[n]
  - » n = [-10:9];
  - » e= exp(abs(n));
  - » figure(1)
  - » stem(n,e);

$$e[n] = a^{|n|}$$



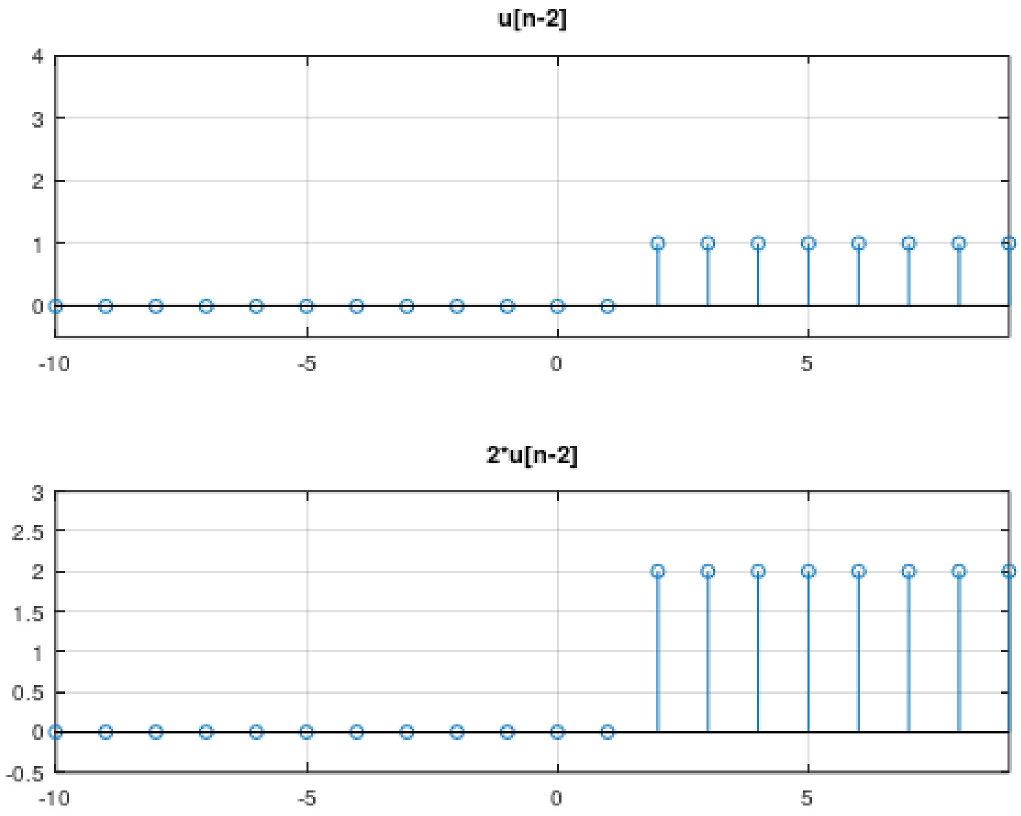
Πράξεις σημάτων

- Πράξεις στο σήμα:
  - Κλιμάκωση πλάτους $y[n] = Ax[n], \quad \forall n$
  - Άθροισμα $y[n] = x_1[n] + x_2[n], \quad \forall n$
  - Γινόμενο $y[n] = x_1[n] \cdot x_2[n], \quad \forall n$
- Πράξεις στο χρόνο:
  - Χρονική ολίσθηση (Μετατόπιση) $y[n] = x[n-k], \quad \forall n$
    - Αν k>0 τότε ολίσθηση στα δεξιά
    - Αν k<0 τότε ολίσθηση στα αριστερά
  - Ανάκλαση $y[n] = x[-n], \quad \forall n$
  - Συνδυασμός: $y[n] = x[k-n], \quad \forall n$
    - Αν k>0 τότε ολίσθηση στα δεξιά
    - Αν k<0 τότε ολίσθηση στα αριστερά

Σήματα διακριτού χρόνου

<u>Παράδειγμα 11</u>

- Μοναδιαίο βήμα u[n-2](μετατόπιση προς τα δεξιά) & με 2*u[n-2] (κλιμάκωση πλάτους)
  - n= [-10:9];
  - u = (n>=2); %If true == 1
  - u1=2*(n>=2);
  - figure(1)
  - subplot(2,1,1), stem(n,u);
  - title('u[n-2]');
  - axis([-10 9 -0.5 4])
  - grid
  - subplot(2,1,2), stem(n,u1);
  - title('2*u[n-2]');
  - axis([-10 9 -0.5 3])
  - grid

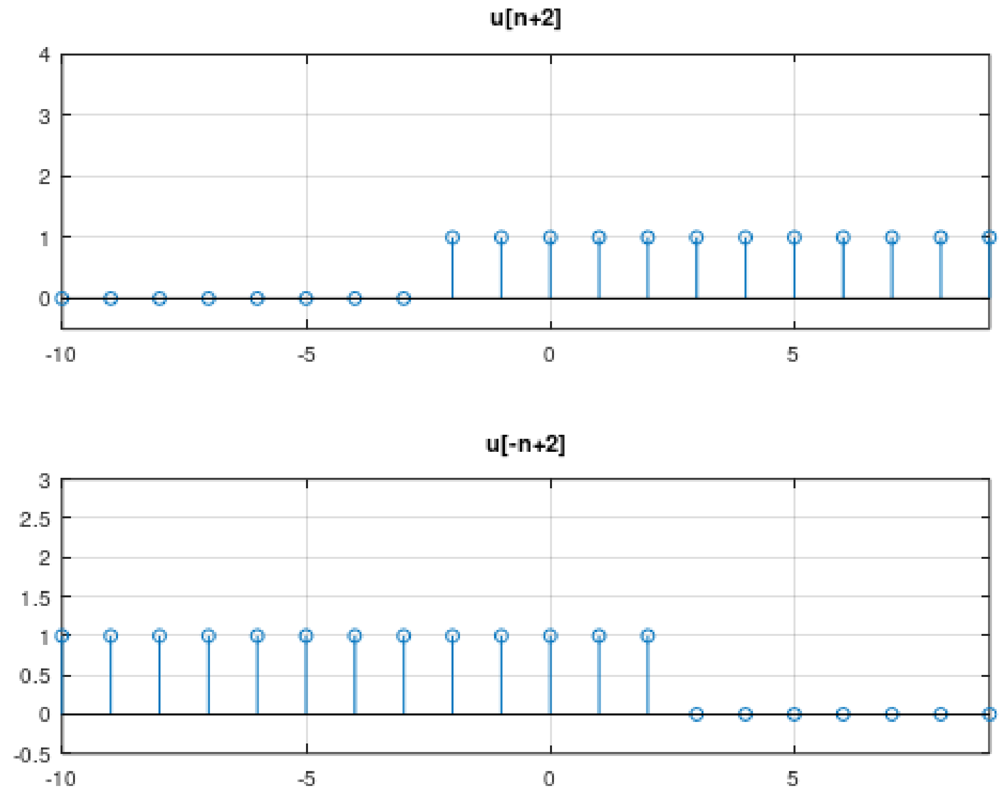


Σήματα διακριτού χρόνου

Παράδειγμα 12

- Μοναδιαίο βήμα u[n+2](μετατόπιση προς τα αριστερά) & με u[-n+2] (ανάκλαση)

```
» n= [-10:9];
» u = (n>=-2); %If true == 1
» u1=(-n>=-2);
» figure(1)
» subplot(2,1,1), stem(n,u);
» title('u[n+2]');
» axis([-10 9 -0.5 4])
» grid
» subplot(2,1,2), stem(n,u1);
» title('u[-n+2]');
» axis([-10 9 -0.5 3])
» grid
```



Πράξεις Σημάτων

- Η πρόσθεση δύο σημάτων x(n) = x1(n)+x2(n) γίνεται με την χρήση του “+” και την εντολή
  - x = x1+ x2

  Το νέο σήμα θα έχει πλάτος ίσο με το άθροισμα των πλατών των σημάτων που προστίθενται.

- Ο πολλαπλασιασμός δύο σημάτων x(n) = x1(n)*x2(n) γίνεται με την χρήση του “.*” και την εντολή
  - x = x1.* x2

  Το νέο σήμα θα έχει πλάτος ίσο με το γινόμενο των πλατών των σημάτων που πολλαπλασιάζονται.

Όταν δεν έχουν το ίδιο μήκος, ισχύει:

Αν το σήμα x1(n) έχει διάρκεια το διάστημα [A1:T1]με A1 ≤ T1 (όπου A1 και T1είναι ακέραιοι αριθμοί) και το σήμα x1(n) έχει διάρκεια το διάστημα [A2:T2]με A2 ≤ T2 (όπου A2 και T2 είναι ακέραιοι αριθμοί), τότε το άθροισμα x(n) = x1(n)+x2(n) έχει διάρκεια το διάστημα [A:T], όπου [A:T]=[min(A1,A2):max(T1,T2)] .

Πρόσθεση Σημάτων

Παράδειγμα 13

- Πραγματοποιήστε και απεικονίστε την πρόσθεση των σημάτων *x1(n)=n+2*, για *n στο [-1:4]* και x2(n)=[8,9,5,2] για *n στο [-2:1]*

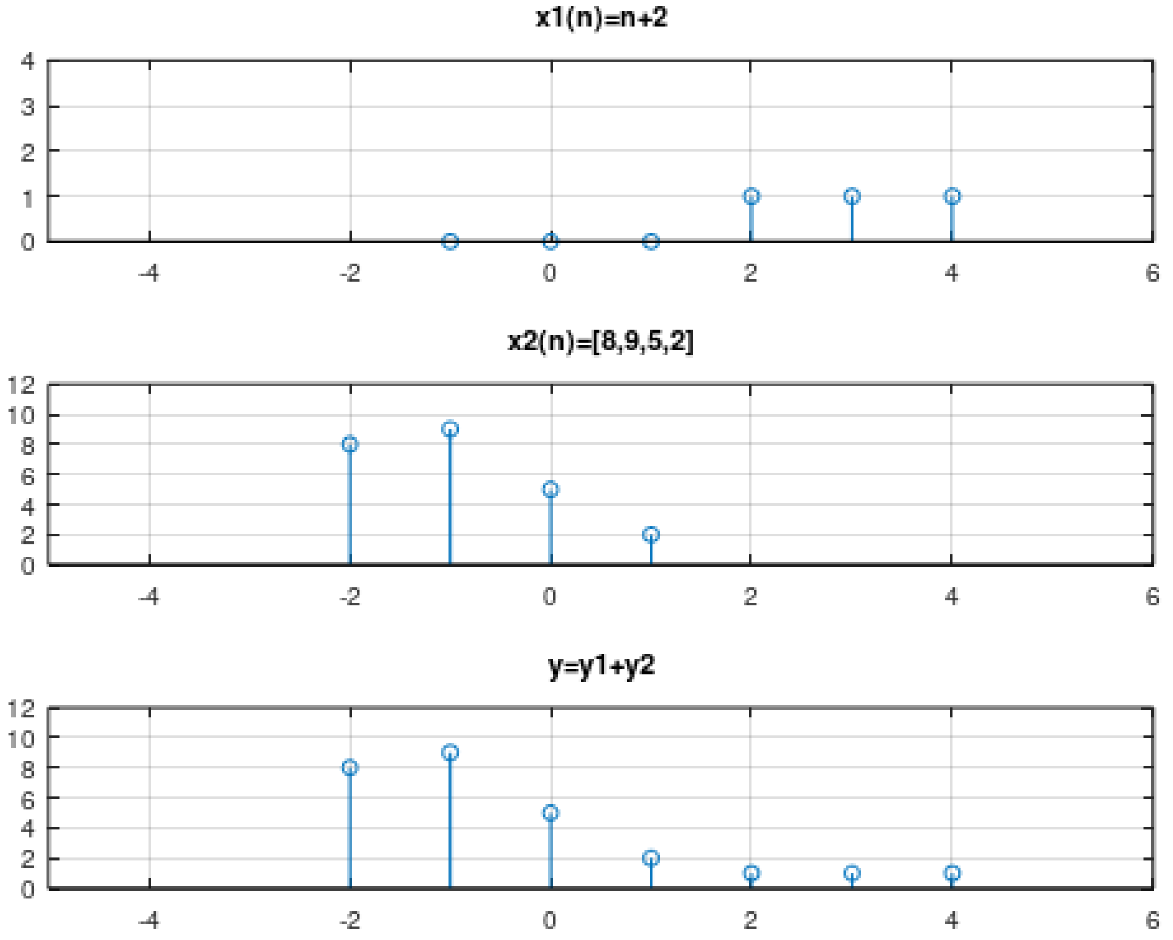
```
» clear all;
» clf;
» n1=[-1:4];
» n2=[-2:1];
» x1=(n1>=2); %x1(n)=n+2,όπου x1 = η πρώτη ακολουθία στο διάστημα n1
» x2=[8,9,5,2]; % x2(n)=[8,9,5,2] όπου  x2= η δεύτερη ακολουθία στο διάστημα n2
» n=[min(min(n1),min(n2)) : max(max(n1),max(n2))]; %Η διάρκεια της y(n)
» y1=zeros(1,length(n)); %γεμίζω με 0 όσο το μέγεθος n
» y2=zeros(1,length(n));
» y1(find((n>=min(n1))&(n<=max(n1))))=x1;  % x1 με διάρκεια n
» y2(find((n>=min(n2))&(n<=max(n2))))=x2; % x2 με διάρκεια n
» y=y1+y2;   %y = το άθροισμα των ακολουθιών στο διάστημα n
```

Πρόσθεση Σημάτων

» figure(1)
» subplot(3,1,1), stem(n1,x1);
» title('x1(n)=n+2');
» axis([-5 6 0 4])
» grid
» subplot(3,1,2), stem(n2,x2);
» title('x2(n)=[8,9,5,2]');
» axis([-5 6 0 12])
» grid
» subplot(3,1,3), stem(n,y);
» title('y=y1+y2');
» axis([-5 6 0 12])
» grid

Πρόσθεση Σημάτων

» figure(1)
» subplot(3,1,1), stem(n1,x1);
» title('x1(n)=n+2');
» axis([-5 6 0 4])
» grid
» subplot(3,1,2), stem(n2,x2);
» title('x2(n)=[8,9,5,2]');
» axis([-5 6 0 12])
» grid
» subplot(3,1,3), stem(n,y);
» title('y=y1+y2');
» axis([-5 6 0 12])
» grid



Πρόσθεση Σημάτων

Τον παραπάνω κώδικα για την πρόσθεση σημάτων μπορούμε να τον ελαχιστοποιήσουμε δημιουργώντας συνάρτηση για την πρόσθεση «signaladd», την οποια θα αποθηκεύσουμε σε ίδιο φάκελο με το M file ή θα την καλέσουμε από το command window δίνοντας τα κατάλληλα ορίσματα. Η συνάρτηση signaladd έχει εισόδους τις παραμέτρουςx1,n1, που είναι το πλάτος και ο χρόνος του πρώτου σήματος και τις παραμέτρουςx2,n2, που είναι το πλάτος και ο χρόνος του δεύτερου σήματος. Η συνάρτηση έχει εξόδους τις παραμέτρους y και n, που είναι το πλάτος και ο χρόνος του αθροίσματος. Δημιουργήσετε την συνάρτηση και δοκιμάστε την σύμφωνα με το παράδειγμα 9. Μπορείτε να κάνετε το ίδιο για τον πολλαπλασιασμό σημάτων ;!

```
» function [y,n]=signaladd(x1,n1,x2,n2)
» % addition y(n)=x1(n)+x2(n)
» n=min(min(n1),min(n2)):max(max(n1),max(n2));
» y1=zeros(1,length(n));
» y2=y1;
» y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;
» y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;
» y=y1+y2;
```

Αναφορές

Το παρών έγγραφο δημιουργήθηκε με βάση τα παρακάτω συγγράμματα, εκπαιδευτικά υλικά:

[1] Νικόλαος Ασημάκης, Μαρία Αδάμ, Σήματα και Συστήματα, ΣΥΝΔΕΣΜΟΣ ΕΛΛΗΝΙΚΩΝ ΑΚΑΔΗΜΑΪΚΩΝ ΒΙΒΛΙΟΘΗΚΩΝ- Εθνικό Μετσόβιο Πολυτεχνείο, 2015.

[2] Β. Διακολουκάς, "Σήματα και Συστήματα," Πολυτεχνείο Κρήτης, Χανιά.

[3] Σ. Καραμπογιάς, "Σήματα και Συστήματα," Αθήνα, 2009.

[4] Vinay K. Ingle and John G.Proakis, "Digital Signal Processing," Bookware Companion Series, 2003.