
Εισαγωγή στην
αναπαράσταση
και
επεξεργασία
σημάτων
συνεχούς και
διακριτού
χρόνου με
Matlab

Τμήμα Πληροφορικής και
Τηλεπικοινωνιών

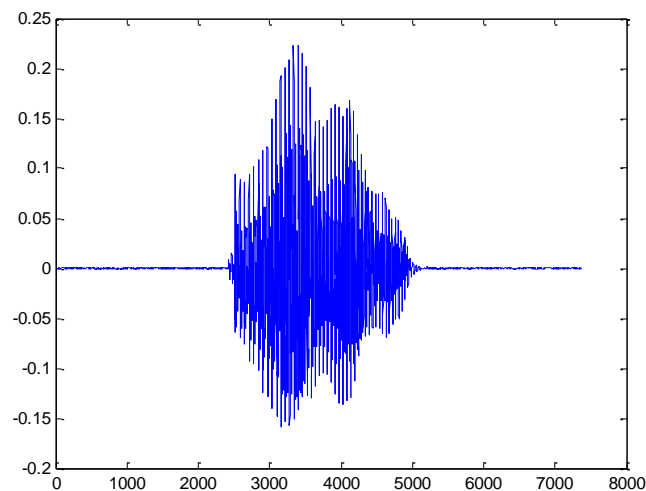
Κολοβού Αθανασία, ΕΔΙΠ .

Σήματα συνεχούς χρόνου

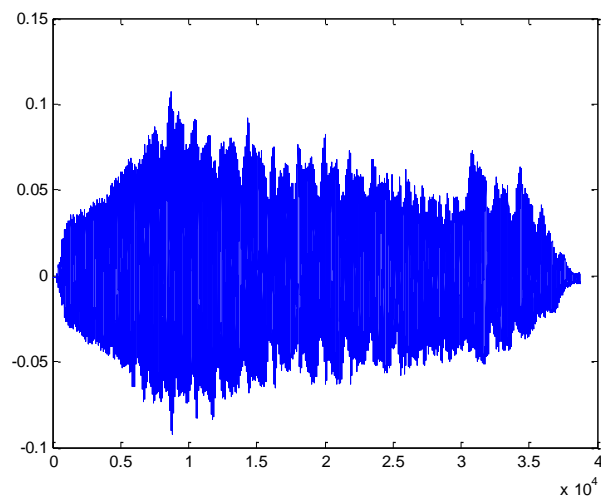
Τα σήματα ταξινομούνται σε σήματα συνεχούς χρόνου και σήματα διακριτού χρόνου. Συνήθως η ανεξάρτητη μεταβλητή είναι ο χρόνος, χωρίς να αποκλείεται η μεταβλητή αυτή να είναι μία άλλη ποσότητα, όπως η θερμοκρασία, η απόσταση κτλ.

Σήματα συνεχούς χρόνου καλείται οποιαδήποτε φυσική ποσότητα που μεταβάλλεται με το χρόνο. Υπάρχουν σήματα φωνής ή σήματα ήχου, όπως βλέπουμε στη συνέχεια.

Το φώνημα «οη» σε μία λέξη (από το TIDIGITS corpus)

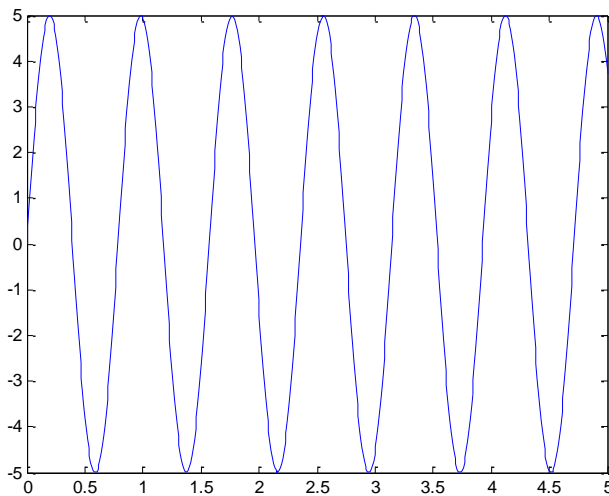


Ένα σήμα από μουσική ενός βιολιού (από [McGill Master Samples](#))



Υπάρχουν επίσης σήματα τα οποία περιγράφονται με μία μαθηματική συνάρτηση. Παράδειγμα η συνάρτηση $X(t)=5*\sin(8*t)$, την οποία θα σχεδιάσουμε στο διάστημα $0 \leq t \leq 5$. Στο Matlab γράφουμε

```
>> t=linspace(0,5,500);  
>> x=5*sin(8*t);  
>> plot(t,x)
```

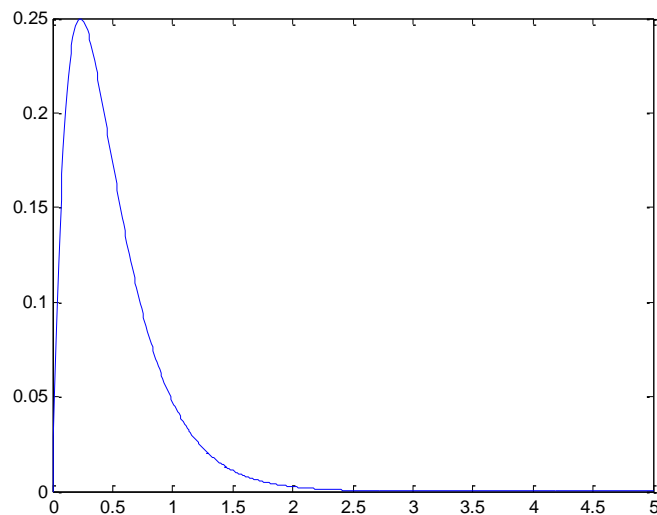


Η για παράδειγμα η παρακάτω συνάρτηση, την οποία θα σχεδιάσουμε $0 \leq t \leq 5$

$$x(t) = \begin{cases} e^{-3t} - e^{-6t}, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

Στο Matlab γράφουμε τις παρακάτω εντολές

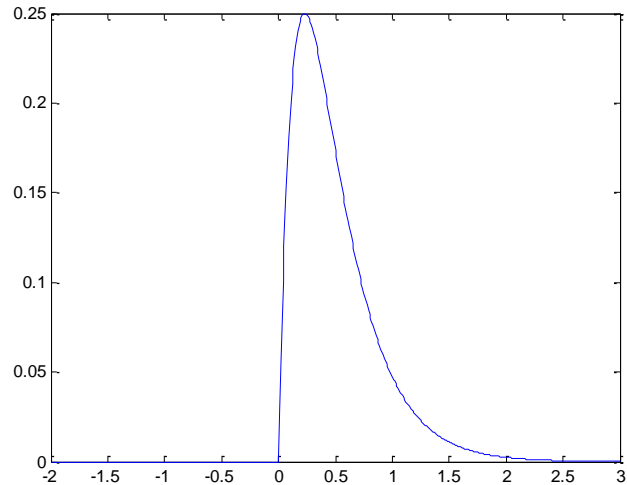
```
>> t=[0:0.01:5];
>> x2=exp(-3*t)-exp(-6*t);
>> plot(t,x2)
```



Ας σχεδιάσουμε στη συνέχεια την $x(t)$ στο διάστημα $-2 \leq t \leq 3$ σε χρονικά διαστήματα $\Delta t=0.01s$. Αν παρατηρήσουμε καλύτερα την $x(t)$, βλέπουμε ότι το σήμα αυτό έχει τιμή 0 για τιμές $t < 0$.

Για να μεριμνήσουμε για αυτό το γεγονός στο γράφημα που θα σχεδιάσουμε, θα χρησιμοποιήσουμε το λογικό τελεστή «<=» με την παρακάτω μορφή. (με την πράξη «<=» που γράφουμε εδώ, φτιάχνουμε ένα νέο πίνακα w , ο οποίος είναι το αποτέλεσμα της λογικής πράξης σύγκρισης το t με το 0, και παράγεται ένα πίνακας ο οποίος έχει τιμές 0 (true) και 1(false)).

```
>> t=[-2:0.01:3];
>> w=(t>=0); %w is a logical
operator
>> x2=(exp(-3*t)-exp(-6*t)).*w;
>> plot(t,x2)
```

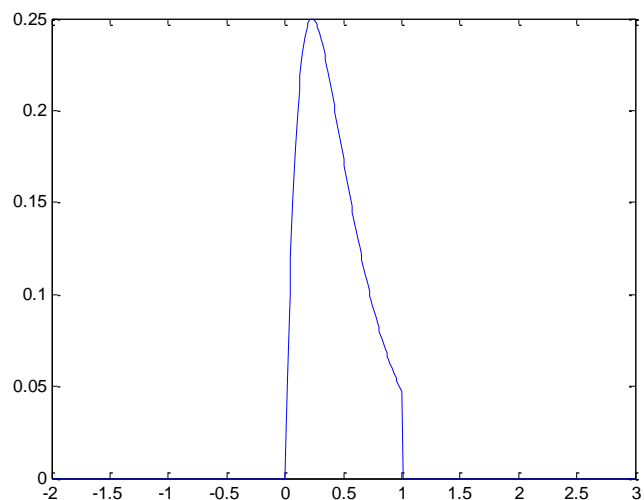


Έστω το σήμα αυτό το οποίο θέλουμε να σχεδιάσουμε στο χρονικό διάστημα $-2 \leq t \leq 3$ το σήμα

$$x(t) = \begin{cases} e^{-3t} - e^{-6t}, & 0 \leq t \leq 1 \\ 0, & \text{αλλιού} \end{cases}$$

Το σήμα είναι το ίδιο με αυτό που είχαμε πριν απλά τώρα $x(t)=0$ για $t>1$. Στο Matlab γράφουμε

```
>> t=[-2:0.01:3];
>> w=(t>=0) & (t<=1);
>> x2=(exp(-3*t)-exp(-6*t)).*w;
>> plot(t,x2)
```



Βασικά σήματα

Ένας αριθμός βασικών σημάτων συνεχούς (η διακριτού) χρόνου πολύ συχνά εμφανίζονται στην ψηφιακή επεξεργασία σήματος και η βασική χρήση τους είναι η δημιουργία άλλων πιο πολύπλοκων σημάτων. Μερικά από αυτά είναι:

* Μοναδιαία ή κρουστική συνάρτηση: $\delta(t) = \begin{cases} 1 & \alpha \nu, t = 0 \\ 0 & \alpha \lambda \lambda \omicron \upsilon \end{cases}$

* Βηματική συνάρτηση $u(t) = \begin{cases} 1 & \alpha \nu, t \geq 0 \\ 0 & \alpha \lambda \lambda \omicron \upsilon \end{cases}$

* Μιγαδική εκθετική ακολουθία $x(t)=e^{j\omega t}$, όπου ω η γωνιακή συχνότητα στο διάστημα $[0, 2\pi]$.

* Εκθετικά σήματα $x(t)=\alpha^t$ και $x(t)=e^{\alpha t}$

* Μοναδιαία Συνάρτηση ράμπας (αντιπροσωπεύεται από μια μονότονα αύξουσα συνάρτηση που μοιάζει με μια ράμπα)

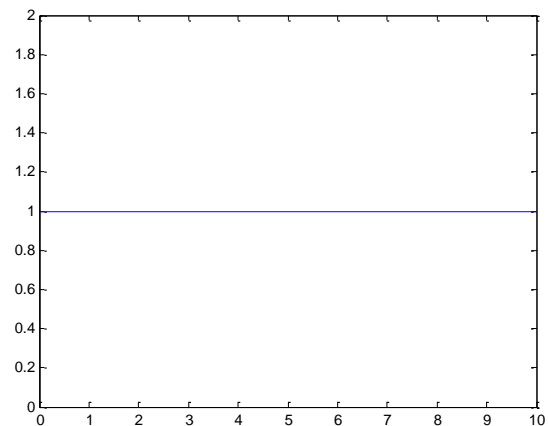
$r(t) = \begin{cases} t & \alpha \nu, t \geq 0 \\ 0 & \alpha \nu, t < 0 \end{cases}$

Θα γράψουμε στη συνέχεια συναρτήσεις στο Matlab που υλοποιούν τα παραπάνω βασικά σήματα. Δημιουργούμε αρχικά ένα **function** στο matlab και στην συνέχεια βλέπουμε τον τρόπο με τον οποίο καλούμε την **function** για να δούμε και γραφικά την συνάρτηση αυτή. Με αυτόν τον τρόπο μπορούμε να καλούμε την `step_function` με διαφορετικές παραμέτρους κάθε φορά.

A. Βηματική συνάρτηση

```
function x = step_function(t)
x = 1*(t>=0);
```

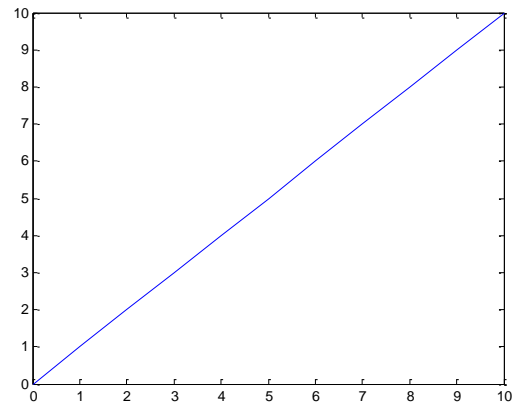
```
>> t=[0:1:10];
>> x = step_function(t);
>> plot(t,x)
```



B. Συνάρτηση ράμπας

```
function x = ramp_function(t)
x = t.*(t>=0);
```

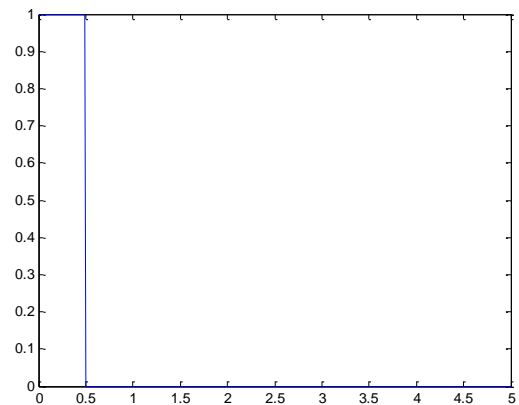
```
>> t=[0:1:10];
>> x = ramp_function(t);
>> plot(t,x)
```



Γ. Χρησιμοποιούμε την βηματική για να φτιάξουμε ένα τετραγωνικό παλμό

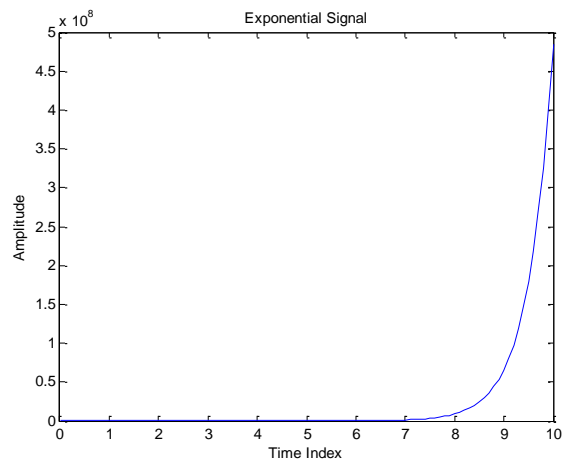
```
function x = pulse(t)
x = step_function(t+0.5)-step_function(t-0.5);
```

```
>> t=[0:0.01:5];
>> x=pulse(t);
>> plot(t,x)
```

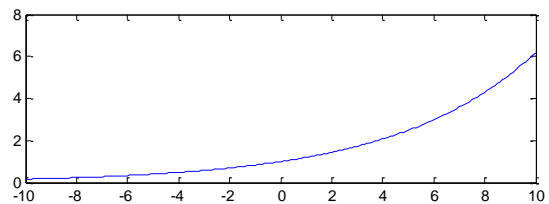
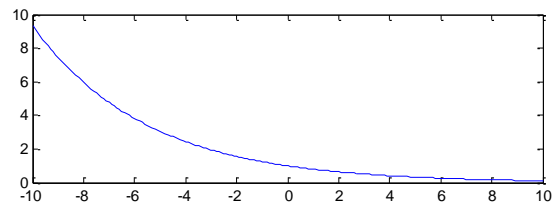


Δ. Εκθετικά σήματα

```
>> t=0:0.1:10;
>> y=exp(2*t);
>> plot(t,y);
>> ylabel ('Amplitude');
>> xlabel ('Time Index');
>> title ('Exponential Signal');
```



```
>> t=-10:0.1:10;
>> a1=0.8;
>> a2=1.2;
>> subplot (211);plot (t,a1.^t)
>> subplot (212);plot (t,a2.^t)
```

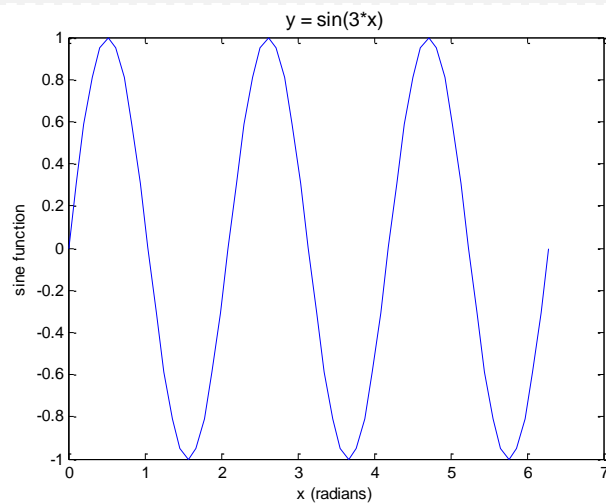


Ημιτονοειδή σήματα συνεχούς χρόνου

Ένα ημιτονοειδές σήμα συνεχούς χρόνου, μπορεί να εκφραστεί ως εξής $x(t)=A*\cos(\Omega*t+\theta)$, όπου A =πλάτος, Ω =η συχνότητα (rad/sec) και θ =φάση (rad). Ή μπορεί να γραφτεί και ως εξής $x(t)=A*\cos(2*\pi*F*t+\theta)$, όπου $\Omega=2\pi F$ και F είναι η συχνότητα σε Hz.

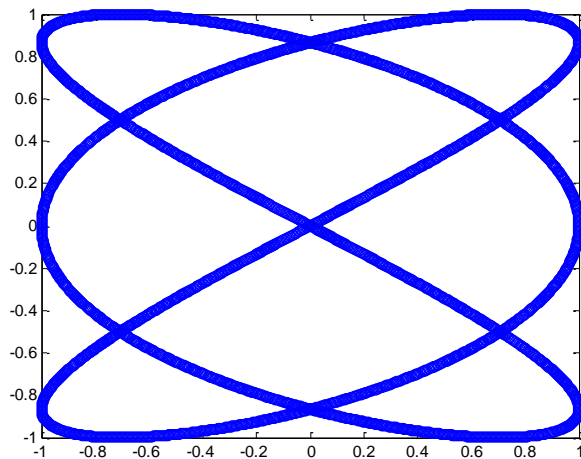
Παραδείγματα

```
>> x = 0:pi/30:2*pi; % x vector, 0 <= x <= 2*pi, increments of pi/30
>> y = sin(3*x); % vector of y values
>> plot(x,y) % create the plot
>> xlabel('x (radians)'); % label the x-axis
>> ylabel('sine function'); % label the y-axis
>> title('y = sin(3*x)', 'FontSize', 12);
```



Συνεχίζοντας μπορούμε να γράψουμε και το παρακάτω

```
>> plot(cos(3*t), sin(2*t), 'o'); % blue (default) circles
```



Σημείωση: περισσότερες πληροφορίες για την εντολή `plot`, μπορείτε να δείτε με `help plot`.

Σήματα διακριτού χρόνου

Όταν η εξαρτημένη μεταβλητή παίρνει διακριτές τιμές, τότε τα σήματα λέγονται διακριτού χρόνου. Συμβολίζονται με $x(n)$ όπου n ακέραιος αριθμός

- Μοναδιαία ή κρουστική συνάρτηση: $\delta(n) = \begin{cases} 1 & \alpha \nu, n = 0 \\ 0 & \alpha \lambda \lambda \omicron \upsilon \end{cases}$
- Βηματική συνάρτηση $u(n) = \begin{cases} 1 & \alpha \nu, n \geq 0 \\ 0 & \alpha \lambda \lambda \omicron \upsilon \end{cases}$
- Μιγαδική εκθετική ακολουθία $x(n)=e^{j\omega n}$, όπου ω η γωνιακή συχνότητα στο διάστημα $[0, 2\pi]$.

Παράδειγμα 1.

- Δημιουργούμε την ακολουθία $u(n)$ για $n:0, 1, \dots, 30$ και δίνουμε την γραφική της παράσταση.
- Δημιουργούμε την ακολουθία $u(n-10)$ για $n: 1, 2, 3, \dots, 30$ και δίνουμε την γραφική της παράσταση.
- Δημιουργούμε την ακολουθία $u(n)-u(n-5)$ για $n: 1, 2, 3, \dots, 10$ και δίνουμε την γραφική της παράσταση.
- Δημιουργούμε την ακολουθία $y=10\cos(50n)$ για $n:0, 1, \dots, 50$.

Όπως και πριν, στα σήματα συνεχούς χρόνου, θα δημιουργήσουμε συναρτήσεις στο Matlab, για να μπορούμε να δημιουργούμε και άλλα σήματα χρησιμοποιώντας τα βασικά σήματα.

Συνάρτηση $\delta(n)$

Α τρόπος: Στην γραμμή « $n-n0==0$ » γίνεται το εξής. Αρχικά υπολογίζεται η πράξη $n-n0$ και το αποτέλεσμα είναι ένα διάνυσμα μεγέθους n . Κάθε στοιχείο του διανύσματος αυτού συγκρίνεται με το 0, και το αποτέλεσμα της σύγκρισης (το οποίο είναι 0 αν η σύγκριση είναι ψευδής και 1 αν είναι αληθής) γράφεται στο διάνυσμα x .

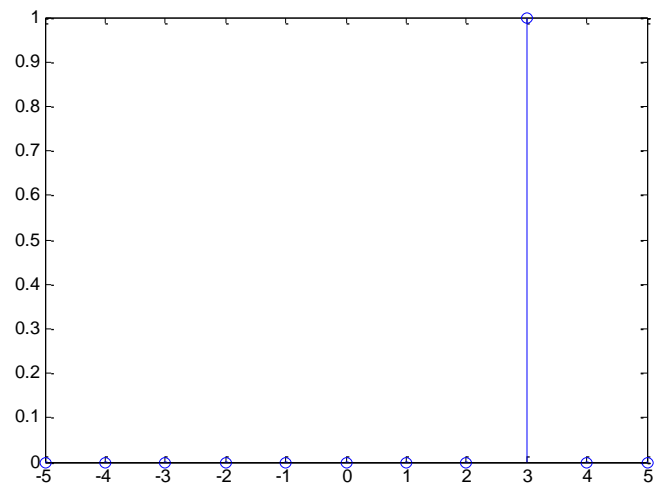
```
function [x,n] = imp(n0,n1,n2)
%implulse function
%using logical operators

n=n1:n2;
x=n-n0==0;

end
```

Α τρόπος

```
>> [x,n]=imp(3,-5,5);  
>> stem(n,x)
```

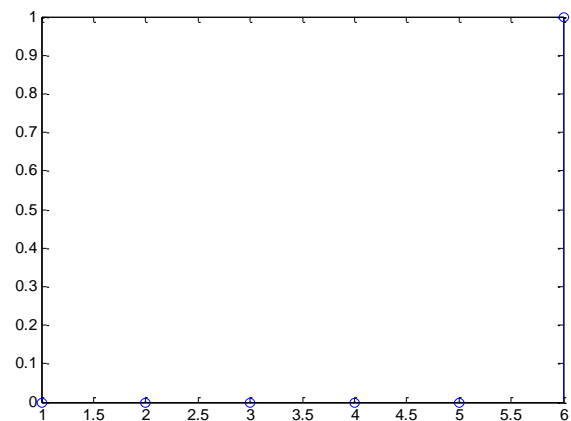


Β τρόπος

Εδώ φτιάχνουμε ένα πίνακα με μηδενικά, και το σημείο k+1

```
function y=dn(1,k)  
d=zeros(1,1);  
d(k+1)=1;  
y=d;  
end
```

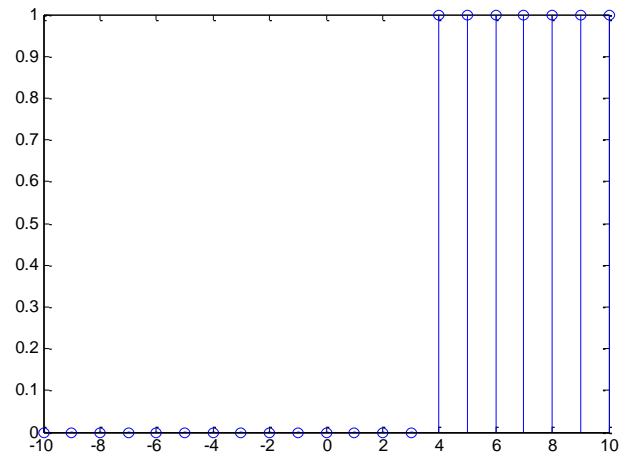
```
>> y=dn(5,5);  
>> stem(y)
```



Βηματική συνάρτηση

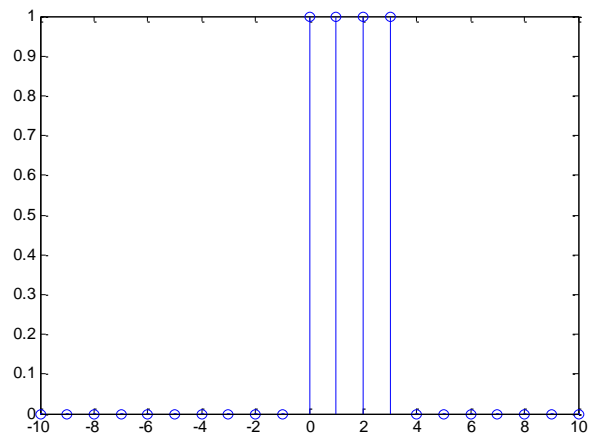
```
function [x,n]=un(n0,n1,n2)  
n=n1:n2;  
x=(n-n0)>=0;  
end
```

```
>> [x,n]=un(7,-10,10);  
>> stem(n,x)
```



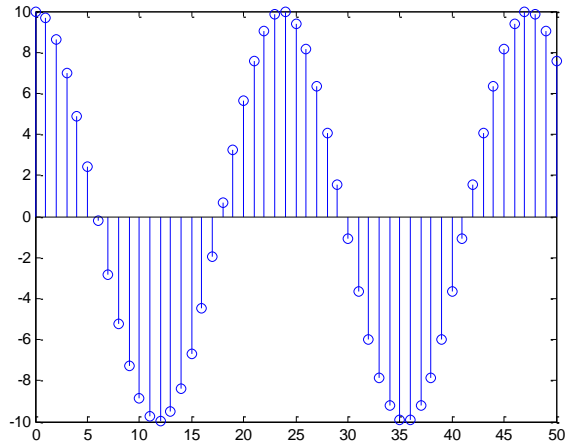
Ολίσθηση σημάτων

```
>> [x1,n]=un(0,-10,10);  
>> [x2,n]=un(4,-10,10);  
>> x=x1-x2;  
>> stem(n,x)
```



Ημιτονοειδής ακολουθία

```
>> n=0:1:50;
>> y=10*cos(50*n);
>> stem(n,y)
```



Διακριτός μετασχηματισμός Fourier.

Αν $x(n)$ ένα διακριτό δάνυσμα το οποίο είναι μη μηδενικό στο διάστημα $0 \leq n \leq N-1$ ο διακριτός μετασχηματισμός Fourier (D.F.T.) καθορίζεται ως εξής:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N} \quad \kappa=0, 1, \dots, N-1.$$

Αν μας δοθεί ο DFT το σήμα μπορεί να ανακατασκευασθεί χρησιμοποιώντας τον αντίστροφο μετασχηματισμό Fourier:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi nk/N} \quad n=0, \dots, N-1.$$

Παραδείγματα

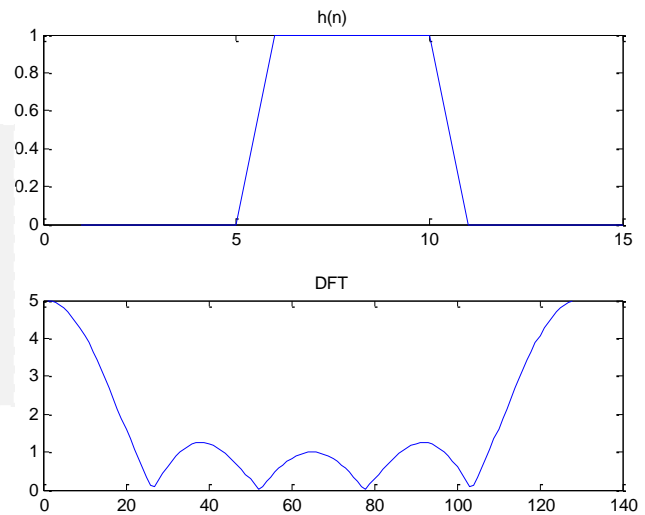
- Δημιουργούμε ένα διακριτό δάνυσμα της επιλογής μας που να έχει τουλάχιστον 20 δείγματα και υπολογίστε τον D.F.T η σημείων.
- Απο τον D.F.T. που προέκυψε ανακατασκευάζουμε το αρχικό σας δάνυσμα.
- Κάνουμε το ίδιο παράδειγμα με ένα μιγαδικό δάνυσμα.

Θα χρησιμοποιήσουμε την εντολή `fft()` του Matlab

```

h=[0 0 0 0 0 1 1 1 1 1 0 0 0 0 0];
[H]=fft(h,128);
subplot(211);
plot(h);title('h(n)')
subplot(212);
plot(abs(H));title('DFT')

```



Στο παραπάνω παράδειγμα χρησιμοποιήσαμε την εντολή `fft` με την οποία υπολογίζετε ο DFT.

Αλλά πριν μιλήσουμε για αυτό ας περιγράψουμε την σχέση το DTF με τον DTFT. Ας δούμε για αρχή έναν τετραγωνικό παλμό $x(n)=1, 0 < n < M$. Ο DTFT αυτής της συνάρτησης είναι

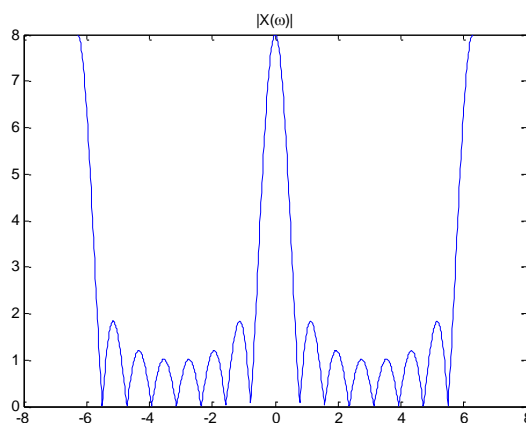
$$X(\omega) = \frac{\sin(\omega M/2)}{\sin(\omega/2)} e^{-j\omega(M-1)/2}$$

Ας σχεδιάσουμε την $X(\omega)$ για $M=8$

```

M = 8;
w = linspace(-2*pi, 2*pi, 800);
X_dtft = (sin(w*M/2) ./ sin(w/2)) .* exp(-1j * w * (M-1) / 2);
plot(w, abs(X_dtft))
title('|X(\omega)|')

```



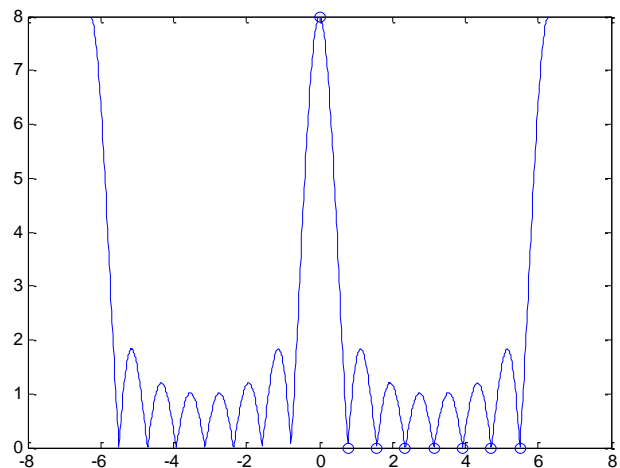
Η συνάρτηση `fft` του Matlab υπολογίζει τον DFT. Στην $x(n)$ δηλαδή θα είχαμε

```
x = ones(1, M);
X = fft(x)
```

```
X =
     8     0     0     0     0     0     0     0
```

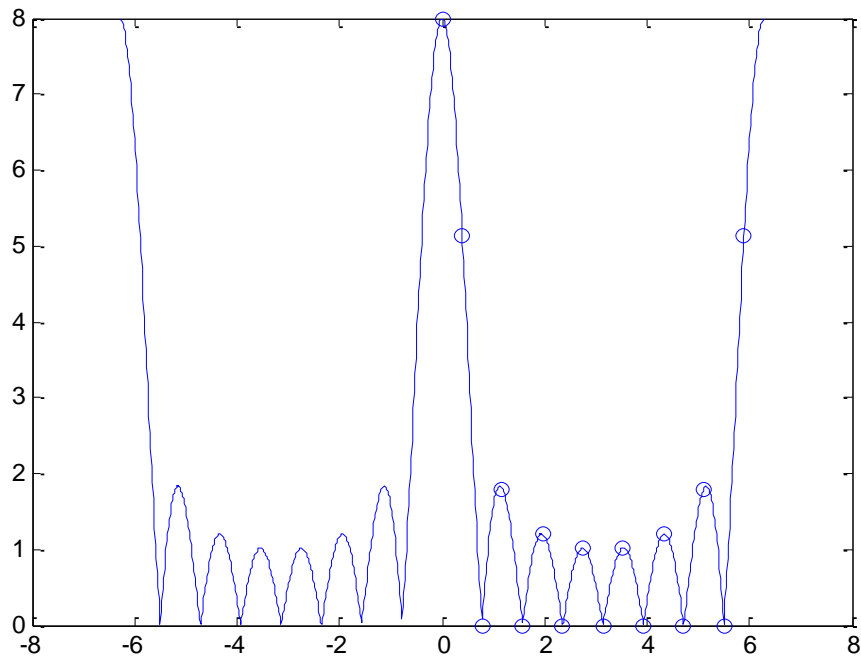
Δηλαδή το dft είναι μόνο 8 και τα υπόλοιπα είναι μηδενικά ? Αυτό δεν μοιάζει και πολύ σωστό! Αν όμως υπολογίσουμε τον fft στα σωστά σημεία τότε ίσως πάρουμε κάτι πιο σωστό...

```
P = 8;
w_k = (0:P-1) * (2*pi/P);
X = fft(x);
plot(w, abs(X_dtft))
hold on
plot(w_k, abs(X), 'o')
hold off
```



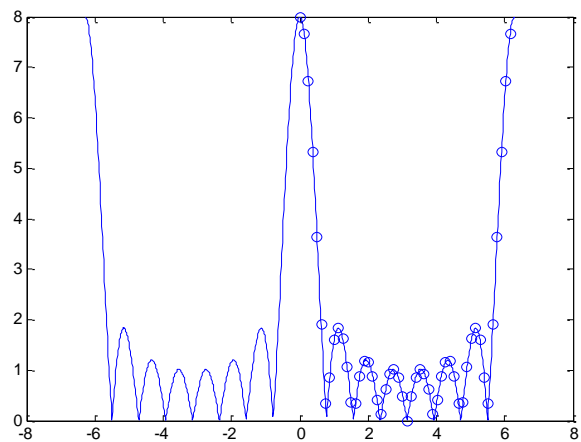
Στο γράφημα έχουμε σημειώσει τα 7 μηδενικά του fft τα οποία αντιστοιχούν σε 7 σημεία στα οποία ο DTFT είναι μηδέν. Μπορούμε να υπολογίσουμε περισσότερα σημεία αλλά αυξάνοντας το P. Ένας τρόπος να γίνει αυτό είναι με zero-padding.

```
>> x16 = [x, zeros(1, 8)]
x16 =
  Columns 1 through 11
     1     1     1     1     1     1     1     1     0     0
  0
  Columns 12 through 16
     0     0     0     0     0
>> P = 16;
>> X16 = fft(x16);
>> w_k = (0:P-1) * (2*pi/P);
>> X = fft(x);
>> plot(w, abs(X_dtft))
>> hold on
>> plot(w_k, abs(X16), 'o')
>> hold off
```



Ένας άλλος τρόπος για να αυξήσουμε το P είναι να χρησιμοποιήσουμε τον εναλλακτικό τρόπο της συνάρτησης $\text{fft}(x, P)$. Όταν χρησιμοποιούμε την συνάρτηση με αυτή τη μορφή προσθέτονται τα αντίστοιχα μηδενικά (zero-padding), και υπολογίζεται ο n -point DFT.

```
P = 50;
Xp = fft(x, P);
w_k = (0:P-1) * (2*pi/P);
X = fft(x);
plot(w, abs(X_dtft))
hold on
plot(w_k, abs(Xp), 'o')
hold off
```



Αν λοιπόν ποτέ είχατε αναρωτηθεί τι εννοούμε με το zero-padding στο Fourier transform, τώρα ξέρετε! Όταν προσθέτουμε μηδενικά σε μία ακολουθία και στη συνέχεια υπολογίζουμε τον DFT, τότε απλά παίρνουμε περισσότερα δείγματα από τον DTFT της αρχικής ακολουθίας.

Δειγματοληψία

Εστω $x_a(t)$ ένα αναλογικό σήμα. Με τη δειγματοληψία επιδιώκουμε την δημιουργία του διακριτού σήματος $x(n)$:

$$x(n) = x_a(nT)$$

T: περίοδος δειγματοληψίας

$$f_s = \frac{1}{T} : \text{συχνότητα δειγματοληψίας}$$

Ο συνεχής μετασχηματισμός Fourier του $x_a(t)$ είναι:

$$X_a(j\Omega) = \int_{-\infty}^{\infty} x_a(t) e^{-j\Omega t} d\Omega$$

Ο μετασχηματισμός Fourier του $x(n)$ είναι:

$$X(e^{j\omega}) = \sum_{-\infty}^{\infty} x(n) e^{-j\omega n}$$

Οι δύο μετασχηματισμοί ακολουθούν την εξής σχέση:

$$X(e^{j\omega}) = \frac{1}{T} \sum_{-\infty}^{\infty} X_a\left(\frac{j\omega + j2\pi k}{T}\right)$$

Ο $X(e^{j\omega})$ είναι άθροισμα των $X_a(j\omega/T + j2\pi k/T)$ για $k = -\infty, \dots, +\infty$. Για να αποφύγουμε επικαλύψεις των αθροιζόμενων $X_a(j\omega/T + j2\pi k/T)$ πρέπει να ισχύει:

$$f_s \geq 2f_{\max}$$

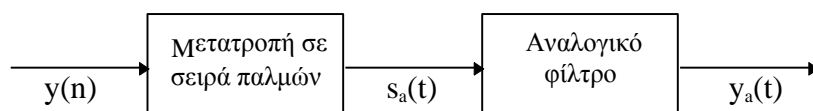
με $2f_{\max}$: Συχνότητα Nyquist.

f_{\max} : η μέγιστη συχνότητα για την οποία το $X_a(j\Omega) \neq 0$.

Δηλαδή για $X_a(j\Omega) = 0$ για $|f| > f_{\max}$.

Ανακατασκευή

Η μετατροπή από διακριτό σήμα σε αναλογικό φαίνεται στο επόμενο σχήμα:



Για την σειρά παλμών $s_a(t)$ ισχύει:

$$s_a(t) = \sum_k y(k)\delta(t - kT)$$

Θεωρώντας τους μετασχηματισμούς Fourier των παραπάνω σημάτων θα ισχύει:

$$Y_a(\Omega) = S_a(\Omega)H_a(\Omega)$$

Το H_a πρέπει να είναι ένα ιδανικό κατωδιαβατό φίλτρο με κέρδος T και συχνότητα αποκοπής $\Omega_c = \pi/T$.

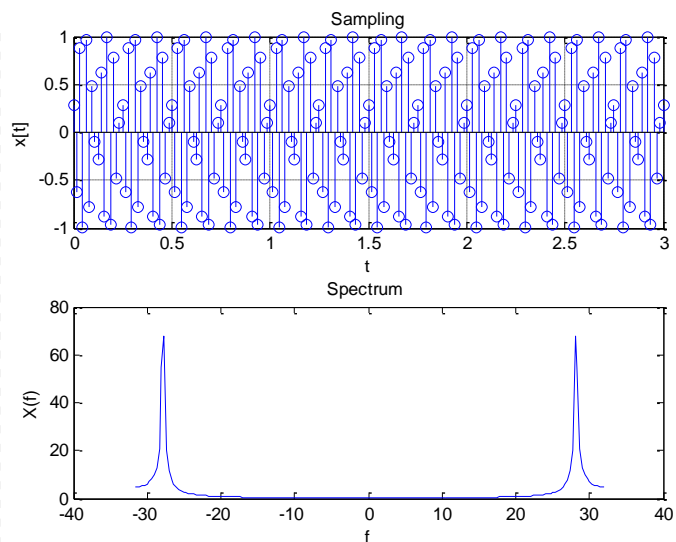
Το ιδανικό αυτό φίλτρο έχει τη μορφή: $h_a = \frac{\sin(\pi t/T)}{\pi t/T}$

■ Αντί για ιδανικό φίλτρο οι εμπορικοί D/A μετατροπείς εργάζονται με τα επόμενα φίλτρα:
zero-order-hold: $h_a(t) = u(t) - u(t-T)$

και

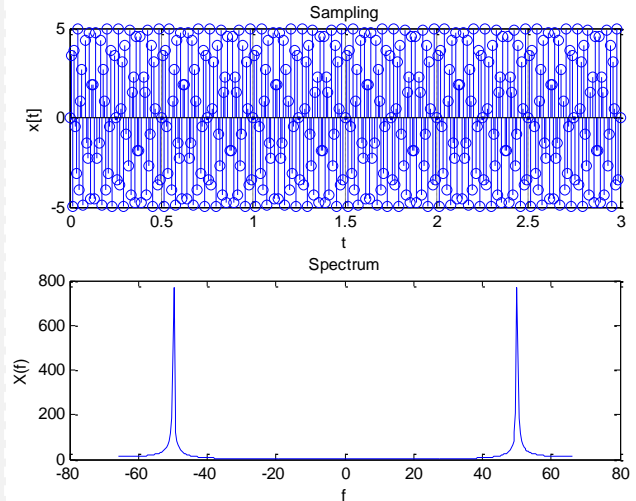
$$\text{first-order-hold: } h_a(t) = \begin{cases} \frac{t}{T} + 1 & \text{αν } -T \leq t \leq 0 \\ -\frac{t}{T} + 1 & \text{αν } 0 \leq t \leq T \\ 0 & \text{αλλού.} \end{cases}$$

```
%Deigmatolipsia
subplot(2,1,1);
T=1/64;
n=0:T:3;
x=cos(2*pi*100*n+5);
stem(n,x);
title('Sampling');
xlabel('t');
ylabel('x[t]');
subplot(2,1,2);
N=length(fft(x));
f=1/T*((1:N)-N/2)/N;
plot(f,abs(fftshift(fft(x))));
title('Spectrum ');
xlabel('f');
ylabel('X(f)');
```



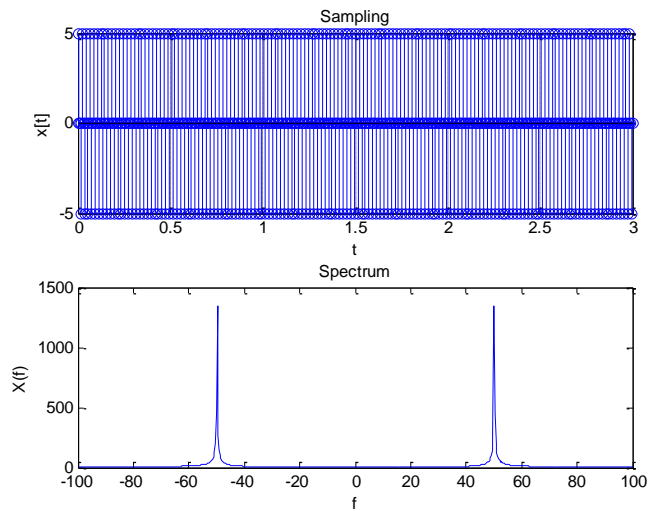
Γα T=132

```
subplot(2,1,1);
T=1/132;
n=0:T:3;
x1=5*sin(2*pi*50*n);
stem(n,x1);
title('Sampling');
xlabel('t');
ylabel('x[t]');
grid on;
subplot(2,1,2);
N=length(fft(x1));
f=1/T*((1:N)-N/2)/N;
plot(f,abs(fftshift(fft(x1))));
title('Spectrum');
xlabel('f');
ylabel('X(f)');
```



```
subplot(2,1,1);
T=1/200;
n=0:T:3;
x1=5*sin(2*pi*50*n);
stem(n,x1);
title('Sampling');
xlabel('t');
ylabel('x[t]');
grid on;
subplot(2,1,2);
N=length(fft(x1));
f=1/T*((1:N)-N/2)/N;
plot(f,abs(fftshift(fft(x1))));
title('Spectrum');
xlabel('f');
ylabel('X(f)');
```

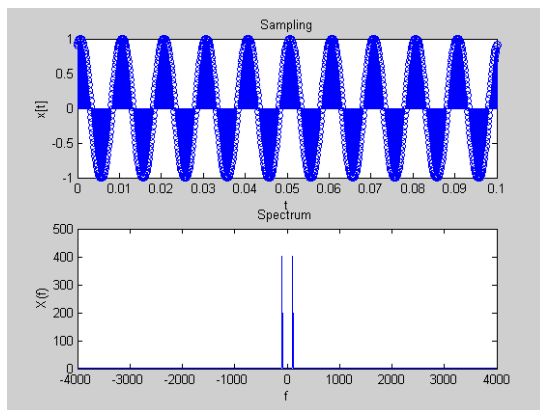
Γα T=200



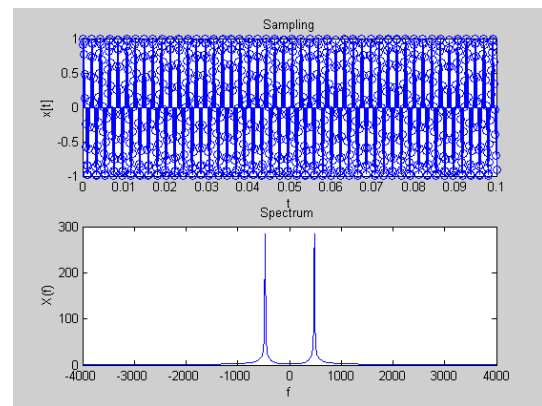
Έχουμε το σήμα $x(t)=\sin(2\pi f_0 t+\phi)$, συχνότητα δειγματοληψίας $f_s=8\text{KHz}$ και ϕ 20 μοίρες. Για τις συχνότητες 100,225,350,475 Hz και 7525,7650,7775,7900 στη θέση του f_0 παίρνουμε για το σήμα και το φάσμα του:

```
subplot(2,1,1);
f0=100;
Ts=1/8000;
t=0:Ts:0.1;
x=sin(2*pi*f0*t+20);
stem(t,x);
title('Δειγματοληψία ');
xlabel('t');
ylabel('x[t]');
grid on;
subplot(2,1,2);
N=length(fft(x));
f=1/Ts*((1:N)-N/2)/N;
plot(f,abs(fftshift(fft(x)))));
title('Φάσμα του σήματος με
δειγματοληψία ');
xlabel('f');
ylabel('X(f)');
```

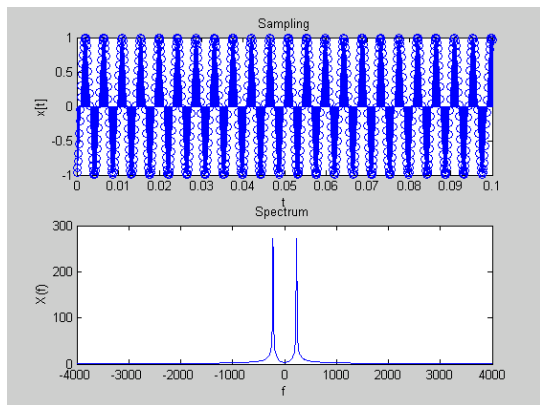
f0=100;



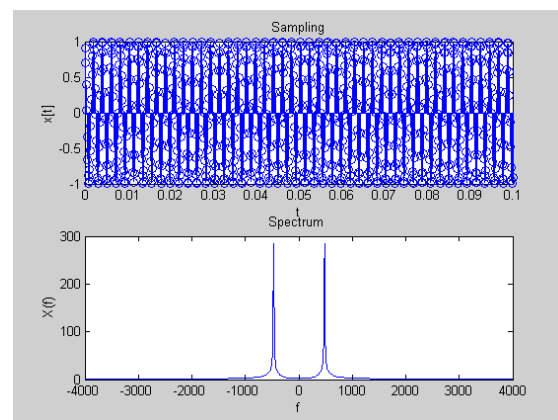
f0=475;



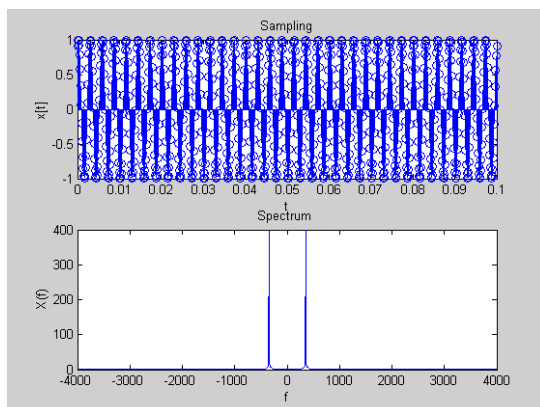
f0=225;



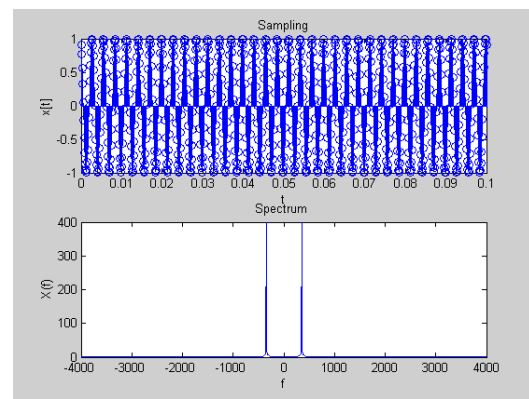
f0=7525;



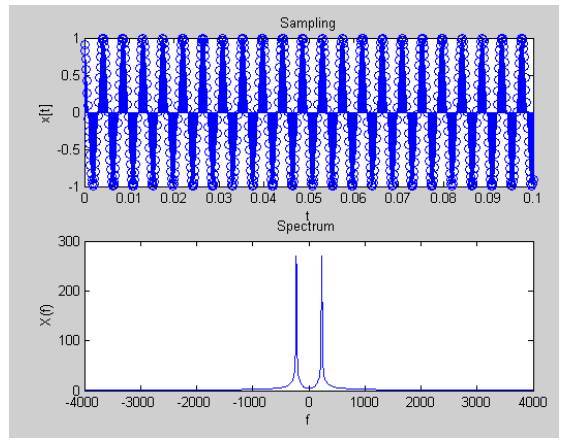
f0=350;



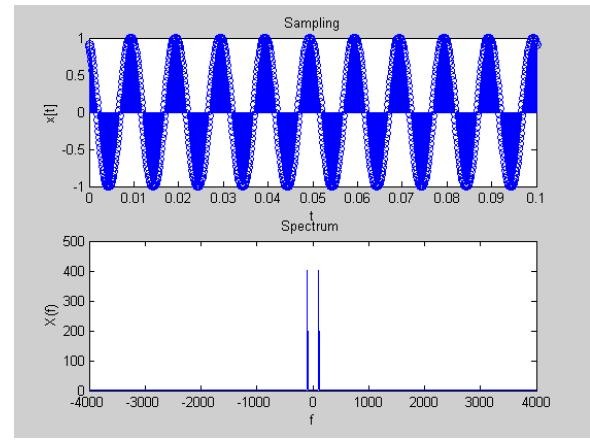
f0=7650;



f0=7775;



f0=7900;



Παράδειγμα (συχνότητα Nyquist)

Έστω ότι $x(t)=10\cos(2\pi 20t)+4\sin(2\pi 40t+5)$. Η συχνότητα Nyquist FN είναι $\max(20,40) \cdot 2 = 80\text{Hz}$

$$1. \frac{dx(t)}{dt} = -400\pi \sin(2\pi 20t) + 320\pi \cos(2\pi 40t + 5)$$

Η συχνότητα Nyquist F_s είναι $\max(20,40) \cdot 2 = 80\text{Hz}$

$$2. x(2t) = 10\cos(2\pi 40t) + 4\sin(2\pi 80t + 5)$$

Η συχνότητα Nyquist F_s είναι $\max(40,80) \cdot 2 = 160\text{Hz}$

(διπλάσια της FN).

$$3. x^2(t) = 100\cos^2(2\pi 20t) + 80\cos(2\pi 20t) \cdot \sin(2\pi 40t + 5) + 16\sin^2(2\pi 40t + 5) = 50 + 50\cos(2\pi 40t) + 40\sin(2\pi 20t + 5) + 40\sin(2\pi 60t + 5) + 8 - 8\cos(2\pi 80t + 10)$$

Η συχνότητα Nyquist είναι 160Hz (διπλάσια της FN).

4.

$$x^2(2t) = 50 + 50\cos(2\pi 80t) + 40\sin(2\pi 40t + 5) + 40\sin(2\pi 120t + 5) + 8 - 8\cos(2\pi 160t + 10)$$

Η συχνότητα Nyquist για αυτό το σήμα είναι 320Hz = 4*FN.

Παρατηρούμε ότι η συχνότητα Nyquist του $x^n(kt)$, είναι ίση με $(n+k)FN$.

Τα παραπάνω αποτελέσματα επιβεβαιώνονται και στο MatLab. (ο κώδικας που τρέχουμε είναι ο παρακάτω μόνο που κάθε φορά αλλάζουμε το χ ή το T)

```
T=1/128;  
t=0:1/T*10;  
x=10*cos(2*pi*20*T*t)+4*sin(2*pi*  
40*T*t+5);  
y=fft(x);  
N=length(y);  
f=1/T*((1:N)-N/2)/N;  
plot(f,abs(fftshift(y)));
```

