# Ψηφιακή Επεξεργασία Σήματος

## Ρόμπερτ Πολοβίνα – 23390338

### Εξάμηνο 7ο

Εργαστηριακή Εργασία Εξαμήνου

Τμήμα 4 Παρασκευή 15.00-17.00

2025-2026

## Άσκηση 1:

**Παραβολική ακολουθία μοναδιαίου βήματος.** Να δημιουργηθεί συνάρτηση που να παράγει την παραβολική ακολουθία μοναδιαίου βήματος η οποία ορίζεται ως ακολούθως:

$$p[n-k] = \begin{cases} (n-k)^2, & n \geq k \\ 0, & n < k \end{cases}$$

Η ίδια συνάρτηση να παράγει: με κωδικό 1 την παραβολική, με κωδικό 2 τη μοναδιαία κρουστική ακολουθία, με κωδικό 3 τη μοναδιαία βηματική ακολουθία και με κωδικό 4 τη μοναδιαία ράμπα. Στη συνέχεια, με χρήση της συνάρτησης να παραχθούν όλες οι παραπάνω ακολουθίες για $-10 \leq n \leq 10$, $k = 4$. Να δημιουργηθούν όλες οι αντίστοιχες γραφικές παραστάσεις.

**Σχολιασμένος κώδικας επίλυσης**:

**func.m:**

```
function x= func(n, k, input)
  % Αρχικοποίηση διανύσματος x (μεγέθους n) με μηδενικά
  x= zeros(size(n));

  % Επιλογή ακολουθίας με βάση το input
  if input== 1 %παραβολική
    %x[i]= (n-k)^2 όπου n>=k, αλλιώς 0
    for i= 1:length(n)
      if n(i)>= k
        x(i)= (n(i)-k)^2;
      endif
    endfor

  elseif input== 2 %μοναδιαία κρουστική
    %x[i]= 1 όπου n=k, αλλιώς 0
    x(n==k)= 1;

  elseif input== 3 %μοναδιαία βηματική
    %x[i]= 1 όπου n>=k, αλλιώς 0
    x(n>=k)= 1;

  elseif input== 4 %μοναδιαία ράμπα
    %x[i]= n-k όπου n>=k, αλλιώς 0
    for i= 1:length(n)
      if n(i)>= k
        x(i)= n(i)-k;
      endif
    endfor
  endif
endfunction
```

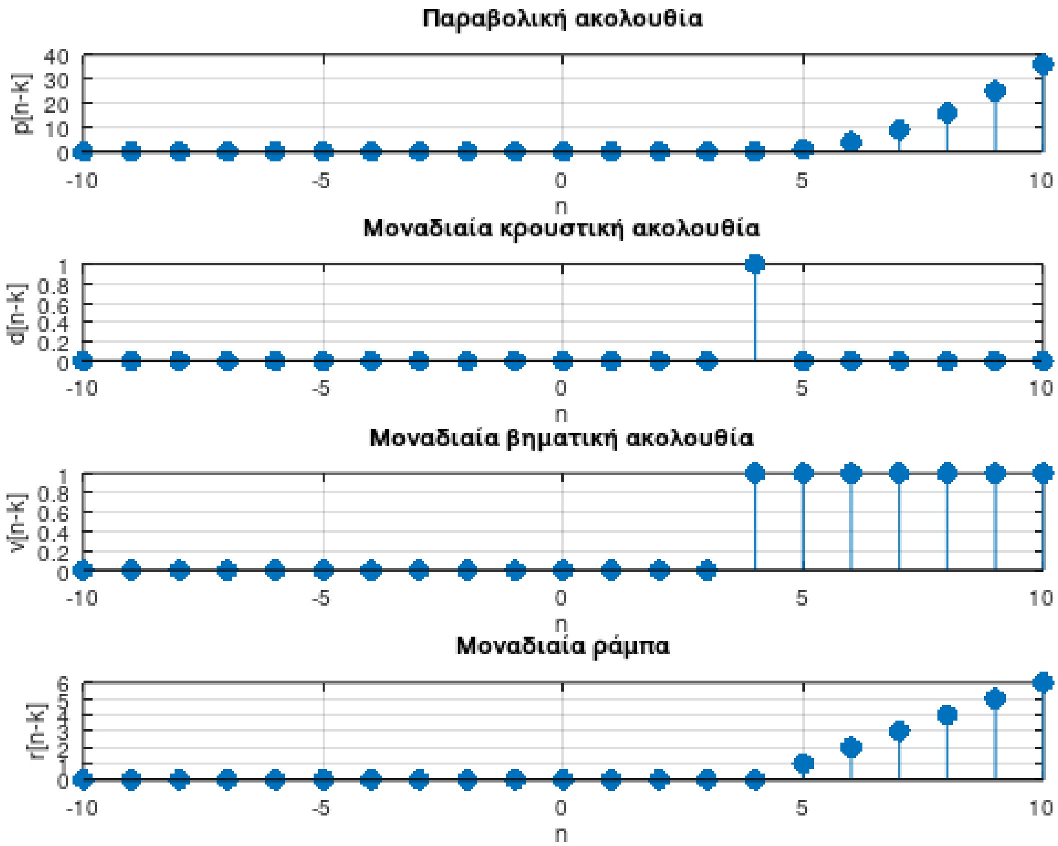
Ο υπολογισμός των ακολουθιών γίνεται με απλό τρόπο. Μηδενίζουμε έναν πίνακα *x* μεγέθους *n*, και όπου ισχύει η επιθυμητή σχέση των *n* και *k*, το αντίστοιχο στοιχείο του πίνακα *x* είτε γίνεται 1, είτε παίρνει ως τιμή το αποτέλεσμα ενός τύπου (αναλόγως την επιλεγμένη ακολουθία). Αλλιώς παραμένει 0.

**main.m:**

```
n= -10:10; %διάνυσμα n από -10 εώς 10
k= 4; %μετατόπιση k

%κλήση της func για κάθε ακολουθία
p= func(n, k, 1); %παραβολική
d= func(n, k, 2); %μοναδιαία κρουστική
v= func(n, k, 3); %μοναδιαία βηματική
r= func(n, k, 4); %ράμπα

figure;

%Σχεδίαση γραφημάτων
subplot(4,1,1);
stem(n, p, 'filled');
title('Παραβολική ακολουθία');
xlabel('n');
ylabel('p[n-k]');

subplot(4,1,2);
stem(n, d, 'filled');
title('Μοναδιαία κρουστική ακολουθία');
xlabel('n');
ylabel('d[n-k]');

subplot(4,1,3);
stem(n, v, 'filled');
title('Μοναδιαία βηματική ακολουθία');
xlabel('n');
ylabel('v[n-k]');

subplot (4,1,4);
stem(n, r, 'filled');
title('Μοναδιαία ράμπα');
xlabel('n');
ylabel('r[n-k]');
```

Στην **main** περνάμε το διάνυσμα *n*, την μετατόπιση *k* και την επιλογή της ακολουθίας στην **func**, και έπειτα σχεδιάζουμε τις αντίστοιχες γραφικές παραστάσεις.

**<u>Αποτελέσματα:</u>**




- Η **παραβολική ακολουθία** είναι μηδενική για n< k και αυξάνεται εκθετικά για n>= k.
- Η **κρουστική ακολουθία** έχει μία τίμη ίση με 1 για n= k και 0 οπουδήποτε αλλού.
- Η **βηματική ακολουθία** έχει την τιμή 0 για n< k και 1 για n>= k.
- Η **μοναδιαία ράμπα** έχει την τιμή 0 και αυξάνεται γραμμικά μετά από το σημείο που n=k.

Όλα τα σήματα δηλαδή, ενεργοποιούνται για *n*= *k*= 4.

## Άσκηση 2:

**Άρτια Περιττή συνιστώσα σήματος**. Να δημιουργηθεί το σήμα διακριτού χρόνου:

$$x[n] = u[n+3] + 5u[n-15] + 4u[n+10]\ , -20 \leq n \leq 20$$

όπου, u[n] η μοναδιαία βηματική ακολουθία.

Στη συνέχεια, να υπολογιστούν το άρτιο και περιττό μέρος της ακολουθίας του σήματος x[n]. Στο ίδιο γραφικό παράθυρο να γίνουν οι γραφικές παραστάσεις του σήματος x[n] καθώς και του άρτιου και περιττού μέρους του σήματος.

### Σχολιασμένος κώδικας επίλυσης:

### ustep.m:

```
function u= ustep(n)

  %u= 0, n< 0
  %u= 1, n>= 0
  u= (n>= 0);
endfunction
```

Ορίζουμε την συνάρτηση της μοναδιαίας βηματικής ακολουθίας ως εξής:

**u= 0 για n< 0**

**u= 1 για n>= 0**

### main.m:

```
n= -20:20;

%x[n]= u[n+3]+ 5u[n-15]+ 4u[n+10]
x= ustep(n+3)+ 5*ustep(n-15)+ 4*ustep(n+10);

%Λόγω συμμετρίας του n, x[-n]= αντεστραμμένο x
x_rev= fliplr(x);
x_even= (x+ x_rev)/2;
x_odd= (x- x_rev)/2;

figure;

%Σχεδίαση γραφημάτων
subplot(3,1,1);
stem(n, x, 'filled');
title('Αρχικό x[n]');
xlabel('n');
ylabel('x[n]');
```

```
subplot(3,1,2);
stem(n, x_even, 'filled');
title('Άρτιο μέρος x_even[n]');
xlabel('n');
ylabel('x_even[n]');

subplot(3,1,3);
stem(n, x_odd, 'filled');
title('Περιττό μέρος x_odd[n]');
xlabel('n');
ylabel('x_odd[n]');
```

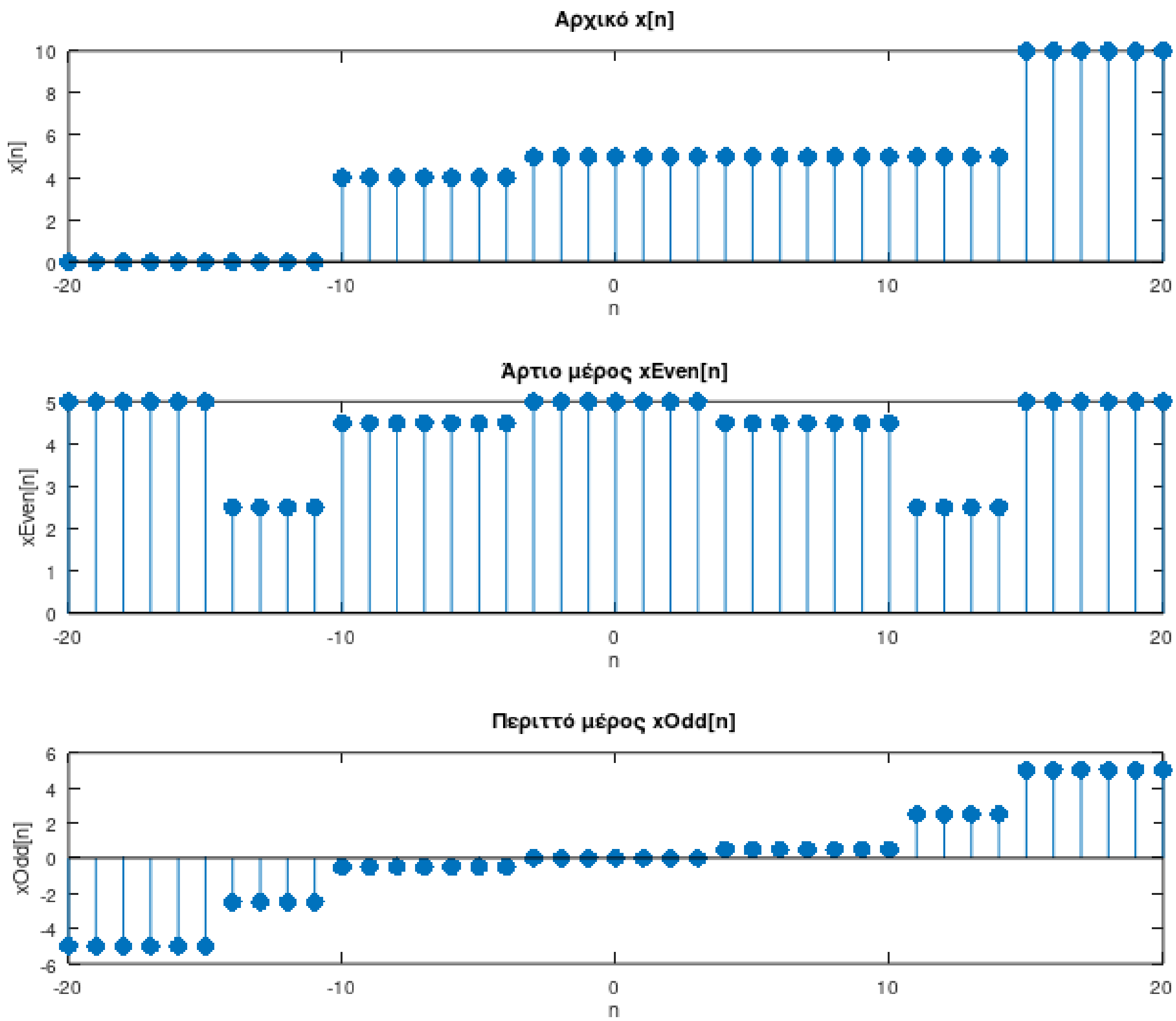
Στην *main* ορίζουμε το διάνυσμα *n* από το -20 εώς το 20 με βήμα 1, και το σήμα *x* χρησιμοποιώντας την συνάρτηση που υλοποιήσαμε παραπάνω. Λόγω της συμμετρίας του *n*, το αντεστραμμένο σήμα *x* υπολογίζεται εύκολα με την **flipr**. Το **άρτιο** και το **περιττό μέρος** του σήματος *x* υπολογίζονται αντίστοιχα από τους τύπους:

**x_even[n]= (x[n]+x[-n])/2**

**x_odd[n]= (x[n]-x[-n])/2**

Τέλος, εμφανίζουμε τις γραφικές παραστάσεις.

**<u>Αποτελέσματα:</u>**




<u>Αρχικό σήμα x[n]:</u>

- Για **n< -10**: κανένα τμήμα του σήματος **u(n+i)** δεν επιστρέφει τιμή >0, άρα τιμή= 0
- Για **-10<n<-3**: ενεργό είναι μόνο το **4u[n+10]**, άρα τιμή= 4
- Για **-3<n<15**: είναι ενεργά το **4u[n+10]** και το και το **u[n+3]**, άρα τιμή= 5
- Για **n>= 15**: είναι όλα τα τμήματα ενεργά, άρα τιμή= 10

<u>Άρτιο μέρος σήματος x_even[n]:</u>

Οι τιμές αριστερά και δεξιά είναι ίδιες, άρα βλέπουμε πως σωστά  είναι συμμετρικό ως προς το *n= 0*.

Περιττό μέρος σήματος x_odd[n]:

Το αριστερό και το δεξί τμήμα του σήματος αντικατοπτρίζονται και για *n= 0* το σήμα έχει τιμή 0, συνεπώς σωστά έχουμε περιττή συμμετρία ως προς το *n= 0*.

## Άσκηση 3:

**Ευστάθεια Συστήματος.** Θεωρήστε το σύστημα διακριτού χρόνου που περιγράφεται από τη Συνάρτηση Μεταφοράς:

$$H(z) = \frac{1 - 0.8z^{-1}}{1 + 1.15z^{-1} + 1.9z^{-2}}$$

Βρείτε και απεικονίστε γραφικά την κρουστική απόκριση του συστήματος. Διερευνήστε την ευστάθεια του συστήματος α) μέσω της κρουστικής απόκρισης (απολύτως αθροίσιμη) και β) μέσω των πόλων της Συνάρτησης μεταφοράς. Εμφανίστε τις αντίστοιχες γραφικές παραστάσεις.

Να εκτελέσετε τα ίδια βήματα για το σύστημα διακριτού χρόνου που περιγράφεται από τη Συνάρτηση Μεταφοράς:

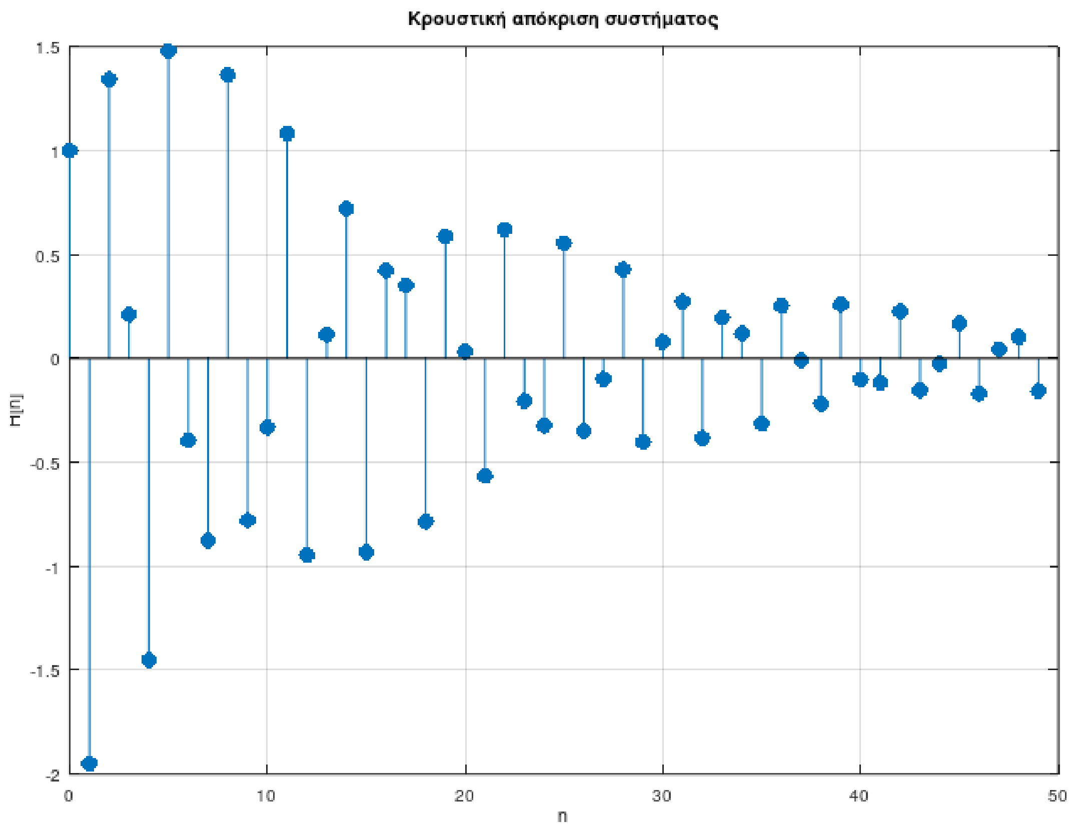
$$H(z) = \frac{1 - 0.8z^{-1}}{1 + 1.15z^{-1} + 1.9z^{-2}}$$

Ξεκινάμε να λύνουμε για το πρώτο σύστημα. Ακολουθεί το κομμάτι κώδικα με το οποίο φτιάχνουμε την κρουστική απόκριση του συστήματος και η γραφική της παράσταση.

```
%Συντελεστές H(z)
b= [1 -0.8];
a= [1 1.15 0.9];

N= 50; %μήκος κρουστικής
n= 0:N-1;

%Κρουστική απόκριση
x= [1 zeros(1, N-1)]; %δ(n)
h= filter(b, a, x); %κρουστική απόκριση

figure;
```

```
stem(n, h, 'filled');
xlabel('n');
ylabel('H[n]');
title('Κρουστική απόκριση συστήματος');
grid on;
```

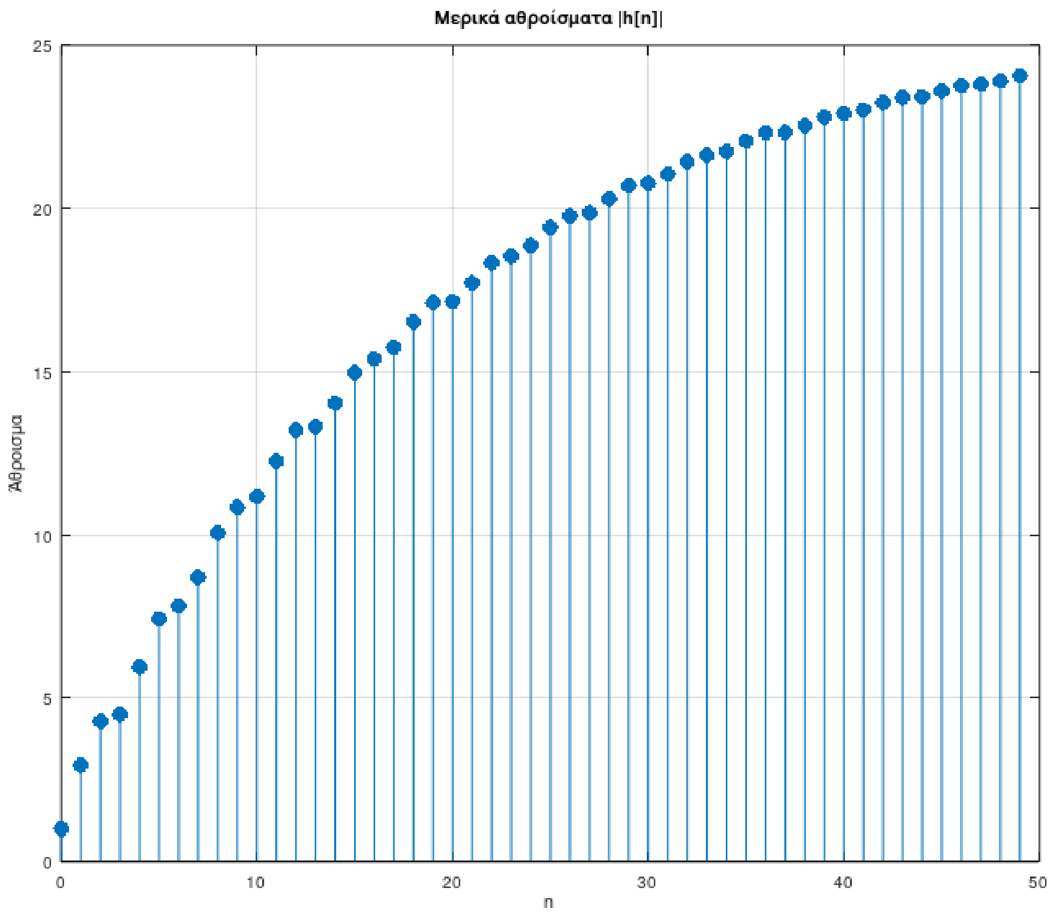
Η κρουστική απόκριση του συστήματος υπολογίζεται εύκολα αρχικοποιώντας με μηδενικά τον πίνακα x μεγέθους όσο το επιθυμητό μήκος της κρουστικής απόκρισης και χρησιμοποιώντας την συνάρτηση ***filter*** με ορίσματα τους συντελεστές του αριθμητή, του παρονομαστή και τον πίνακα x.




Για να είναι το σύστημα ευσταθές πρέπει η κρουστική απόκριση να είναι απολύτως αθροίσιμη, δηλαδή το άθροισμα των |h[n]| να είναι πεπερασμένο. Εκτελώντας την εντολή *sum(abs(h)* βλέπουμε ότι ισούται περίπου με 24, άρα είναι όντως πεπερασμένο και συνεπώς το σύστημα είναι ευσταθές με βάση την κρουστική απόκριση.
Για την γραφική απεικόνιση αυτό σημαίνει πως τα μερικά αθροίσματα πρέπει να συγκλίνουν στο αριθμητικό αποτέλεσμα του συνολικού αθροίσματος, δηλαδή στο 24.

Μέσω της συνάρτησης ***cumsum*** υπολογίζουμε τα μερικά αθρολισματα και δημιουργούμε το διάγραμμα τους.

```
%Ελεγχος ευστάθειας μέσω κρουστικής
sum(abs(h))

figure;
stem(n, cumsum(abs(h)), 'filled');
xlabel('n');
ylabel('Άθροισμα');
title('Μερικά αθροίσματα |h[n]|');
grid on;
```

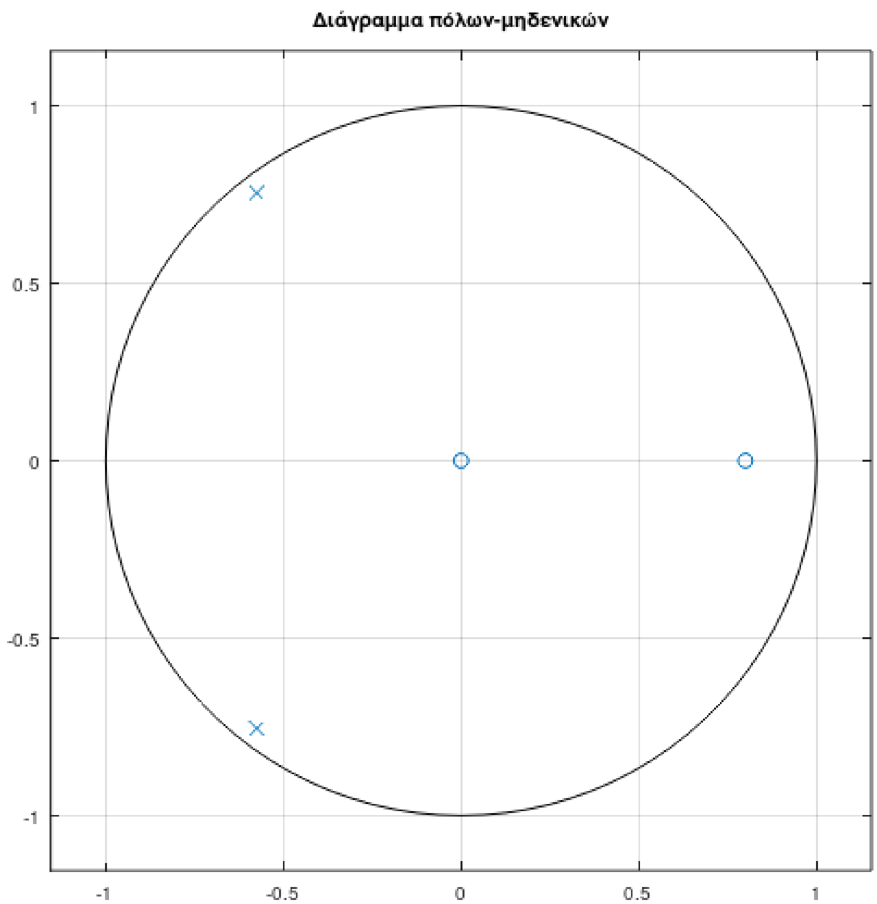



Παρατηρούμε ότι το μερικό άθροισμα πράγματι συγκλίνει στο 24.

Τέλος, θα εξετάσουμε την ευστάθεια μέσω των πόλων της συνάρτησης μεταφοράς. Ένα σύστημα είναι ευσταθές αν όλοι οι πόλοι του βρίσκονται μέσα στον μοναδιαίο κύκλο.

```
%Ελεγχος ευστάθειας μέσω πόλων
figure;
zplane(a, b);
title('Διάγραμμα πόλων-μηδενικών');
grid on;
```

Για την γραφική επίλυση αρκεί να χρησιμοποιήσουμε την συνάρτηση ***zplane*** με ορίσματα τον αριθμητή και τον παρονομαστή της συνάρτησης μεταφοράς.




Βλέπουμε ότι οι πόλοι της συνάρτησης μεταφοράς βρίσκονται εντός του μοναδιαίου κύκλου, άρα το σύστημα είναι ευσταθές και μέσω αυτού του τρόπου.

Συνεπώς ολόκληρομένος ο κώδικας ***main.m*** γίνεται:

```
%Συντελεστές H(z)
b= [1 -0.8];
a= [1 1.15 0.9];

N= 50; %μήκος κρουστικής
n= 0:N-1;

%Κρουστική απόκριση
x= [1 zeros(1, N-1)]; %δ(n)
h= filter(b, a, x); %κρουστική απόκριση

figure;
stem(n, h, 'filled');
xlabel('n');
ylabel('H[n]');
title('Κρουστική απόκριση συστήματος');
grid on;

%Ελεγχος ευστάθειας μέσω κρουστικής
sum(abs(h))
figure;
stem(n, cumsum(abs(h)), 'filled');
xlabel('n');
ylabel('Άθροισμα');
title('Μερικά αθροίσματα |h[n]|');
grid on;

%Ελεγχος ευστάθειας μέσω πόλων
figure;
zplane(b, a);
title('Διάγραμμα πόλων-μηδενικών');
grid on;
```

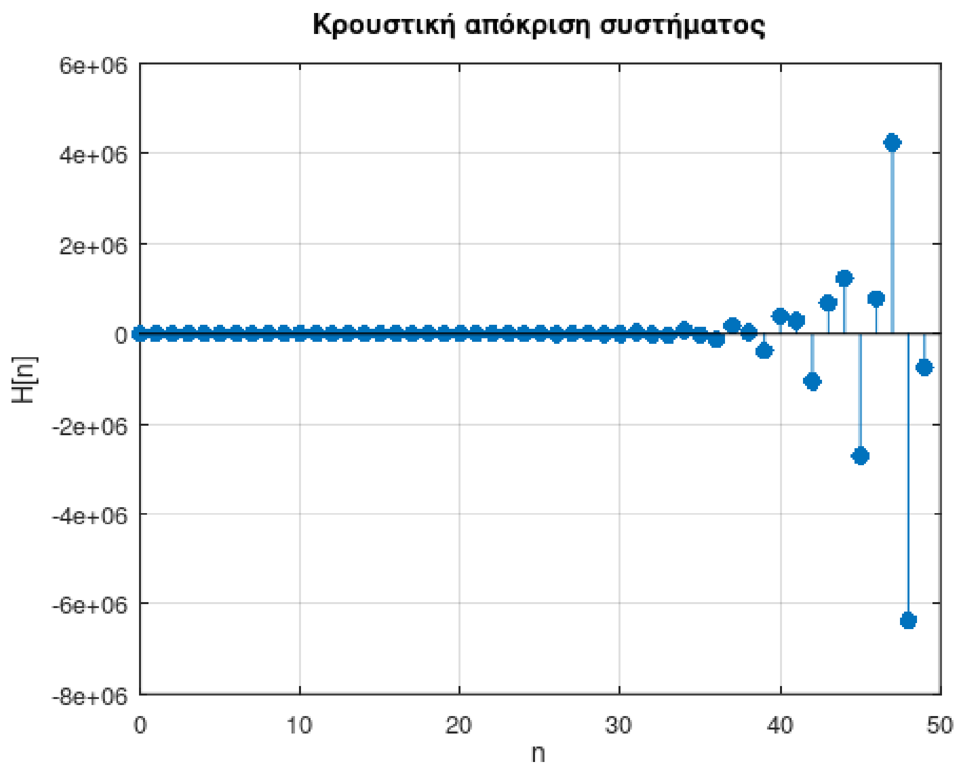
Θα προχωρήσουμε τώρα στην επίλυση των ίδιων ζητούμενων αλλά για την

$$H(z) = \frac{1 - 0.8z^{-1}}{1 + 1.15z^{-1} + 1.9z^{-2}}$$

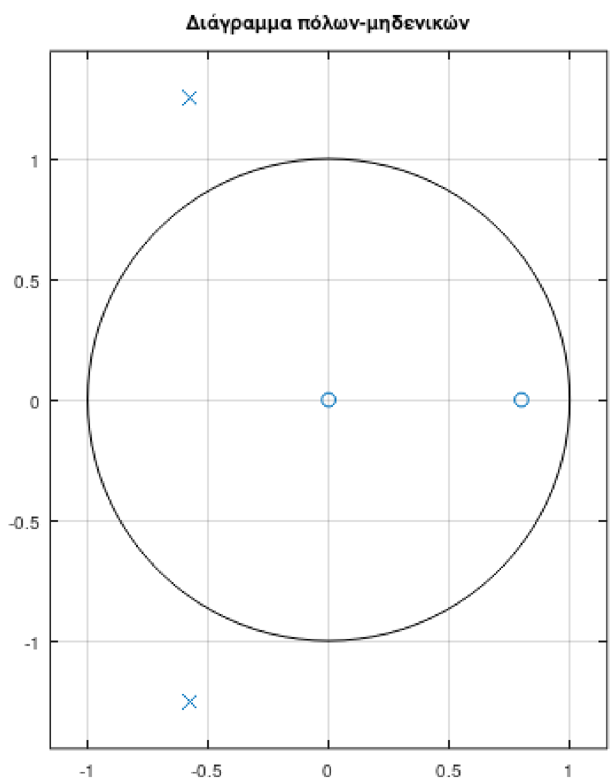
Αρκεί να αλλάξουμε τους συντελεστές στο πρόγραμμα μας για να βρούμε τα αποτελέσματα που ψάχνουμε. Συνεπώς για

```
a= [1 -0.8];
b= [1 1.15 1.9];
```

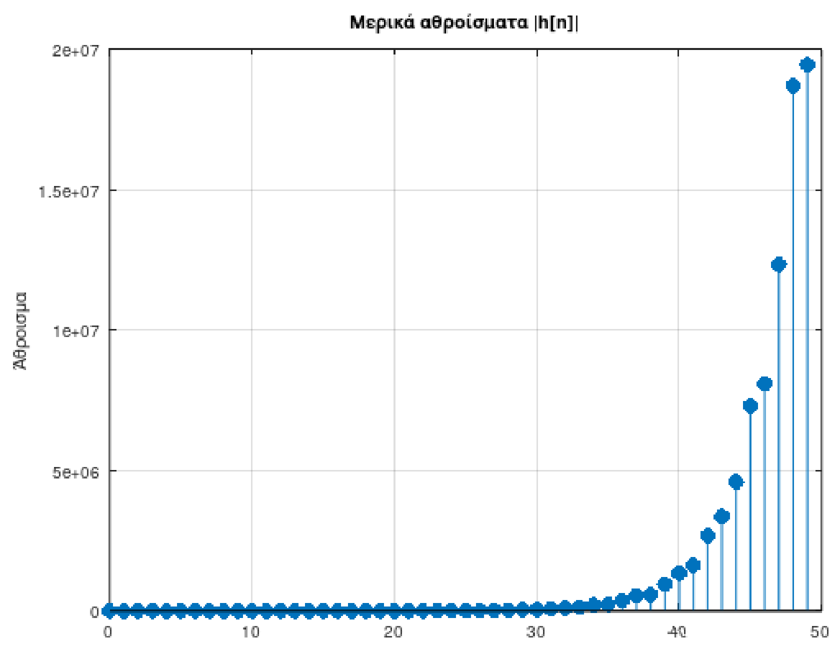
μπορούμε να καταλάβουμε από τα διαγράμματα των πόλων και των μερικών αθροισμάτων πως το σύστημα που περιγράφεται από την νέα συνάρτηση μεταφοράς <u>δεν</u> είναι ευσταθές.



Κρουστική απόκριση συστήματος
6e+06
4e+06
2e+06
0
-2e+06
-4e+06
-6e+06
-8e+06
H[n]
0
10
20
30
40
50
n

Διάγραμμα πόλων-μηδενικών
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1

Μερικά αθροίσματα |h[n]|
2e+07
1.5e+07
1e+07
5e+06
0
Άθροισμα
0
10
20
30
40
50
n

## <u>Άσκηση 4:</u>

**Κυκλική συνέλιξη.** Δίνονται τα σήματα διακριτού χρόνου:

$$x[n] = \delta[n] + 2\delta[n-1] + \delta[n-3]$$

$$h[n] = 2\delta[n] + 2\delta[n-1] + \delta[n-2] + \delta[n-3]$$

Χρησιμοποιώντας την ιδιότητα της κυκλικής συνέλιξης του DFT

$$DFT[x[n] \otimes h[n]] = X[k]H[k]$$

καθώς και την ακόλουθη σχέση για τον υπολογισμό του iDFT με χρήση του DFT

$$y[n] = idft(Y(k)) = 1/N\ dft(Y*(k))$$

(όπου $Y*(k)$ ο συζυγής μιγαδικός του $Y(k)$), υπολογίστε την κυκλική τους συνέλιξη, 4 σημείων α) στο πεδίο του διακριτού χρόνου β) στο πεδίο συχνοτήτων : β1) με χρήση του fft, β2) χωρίς χρήση fft αλλά με υπολογισμό του DFT με πολλαπλασιασμό με τον πίνακα παραγόντων στροφής 4x4. Επιβεβαιώστε οτι όλα τα αποτελέσματα συμπίπτουν.

α) <u>Κυκλική συνέλιξη στο πεδίο του χρόνου:</u>

```
%Συντελεστές σημάτων
x= [1 2 0 1];
h= [2 2 1 1];
N= 4;
%κυκλική συνέλιξη:

%α) στο πεδίο του χρόνου
y_time= zeros(1, N);

for n0= 0:N-1
  for m= 0:N-1
    k= mod(n0- m, N);
    y_time(n0+1)= y_time(n0+1)+ x(m+1)* h(k+1);
  endfor
endfor

y_time
```

Ο ορισμός είναι της κυκλικής συνέλιξης είναι:

$$\sum_{m=0}^{N-1} x_1(m)x_2((n-m))_N$$

Όπου $x_2((n-m))_N$ η κυκλικά ολισθημένη κατά m θέσεις ακολουθία.

Στον κώδικα επίλυσης ο εξωτερικός βρόχος ***for n0= 0:N-1*** επαναλαμβάνει για κάθε n0= 0, 1, 2, ..., N-1 και ο εσωτερικός βρόχος ***for m= 0:N-1*** προσαυξάνει το άθροισμα της συνέλιξης για το συγκεκριμένο n0.

Ο υπολογισμός ξεκινάει από τον υπολογισμό του υπολοίπου της διαίρεσης με το N, μέσω της εντολής ***k= mod(n0- m, N).*** Έπειτα, πολλαπλασιάζουμε τα στοιχεία ***x[m+1]*** και ***h[k+1]*** και τα αθροίζουμε για κάθε ***n0***, υλοποιώντας έτσι τον ορισμό της κυκλικής συνέλιξης που δώσαμε παραπάνω.

Τα αποτελέσματα που εμφανίζονται στο παράθυρο εντολών είναι

**y_time= [6 7 6 5]**

β) Κυκλική συνέλιξη στο πεδίο συχνοτήτων:

**β1) με χρήση του *fft:***

```
%β) στο πεδίο συχνοτήτων
%β1) με ftt

%μετασχηματισμός γραμμών και στηλών
X= fft(x, N);
H= fft(h, N);

Y= X.*H; %πολ/μος στοιχείων
y_fft= ifft(Y)
```

Για την κυκλική συνέλιξη στο πεδίο συχνοτήτων ακολουθούμε την ιδιότητα του DFT

$$DFT[x[n]\otimes h[n]] = X[k]H[k]$$

Ξεκινάμε υπολογίζοντας τον μετασχηματισμό Fourier για τα δύο σήματα μας με τις εντολές **X= fft(x, N) και H= fft(h, N)** αντίστοιχα. Προχωράμε στον πολλαπλασιασμό των στοιχείων των πινάκων που βρήκαμε από τους μεταχηματισμούς, με την εντολή **Y= X.*H .** Ως τελευταίο βήμα, κάνοντας αντίστροφο μετασχηματισμό Fourier χρησιμοποιώντας την συνάρτηση ***ifft*** βρίσκουμε την ζητούμενη κυκλική συνέλιξη η οποία τυπώνεται και στο παράθυρο εντολών:

**y_fft= [6 7 6 5]**

**β2) χωρίς χρήση fft**

```
%β2) χωρίς fft
%πίνακας DFT (W) 4x4
x_col= x.';
h_col= h.';

i= (0:N-1).'; %στήλες
j= 0:N-1;     %γραμμές
W= exp(-1j*2*pi/N * (i*j)); %W(k,n)= e^(-j2πkn/N)

X= W*x_col;
H= W*h_col;
Y= X.*H; %DFT[x]*DFT[h]

y= (1/N)*conj(W).'*Y
```

Για να πραγματοποιήσουμε την κυκλική συνέλιξη χωρίς την χρήση της ***fft*** ξεκινάμε με το να ορίσουμε τον πίνακα παραγόντων στροφής 4x4. Αφού δημιουργήσουμε τις γραμμές και τις στήλες του, εφαρμόζοντας τον τύπο ***e^(−j2πkn/N)*** από τον ορισμό της DFT φτιάχνουμε τον ζητούμενο πίνακα W για να εκτελέσουμε παρακάτω τον μετασχηματισμό σαν πολ/μο πινάκων.

Εφαρμόζουμε τον μετασχηματισμό DFT στις στήλες και στις γραμμές του πίνακα με τις εντολές **X= W*x_col** και **H= W*h_col** και προχωράμε στον πολλαπλασιασμό των στοιχείων του πίνακα στο πεδίο των συχνοτήτων σύμφωνα με τον ορισμό της συνέλιξης:

**DFT{x*h}= DFT{x}·DFT{h}**

Τέλος, σύμφωνα με την σχέση της εκφώνησης $\boldsymbol{y[n]= 1/N\ dft(Y*(k))}$, υπολογίζουμε τον αντίστροφο DFT (iDFT) με την εντολή

**y= (1/N)*conj(W)*Y**

Συγκεκριμένα, η ***conj(W)*** υπολογίζει τον συζυγή του πίνακα W και τον πολλαπλασιάζουμε με τον DFT του W * 1/N. Με την συνάρτηση ***real*** τυπώνουμε μόνο το πραγματικό μέρος και με τον τελεστή **.’** μετατρέπεται το αποτέλεσμα από στήλη σε γραμμή για να εμφανίζεται το αποτέλεσμα ομοιόμορφα με τα προηγούμενα.

Μετά την εκτέλεση των εντολών, βλέπουμε στην γραμμή εντολών να τυπώνεται πάλι

**y= [6 7 6 5]**

Ολόκληρο το αρχείο **main.m** τελικά είναι:

```
%Συντελεστές σημάτων
x= [1 2 0 1];
h= [2 2 1 1];
N= 4;

%κυκλική συνέλιξη:

%α) στο πεδίο του χρόνου
y_time= zeros(1, N);

for n0= 0:N-1
  for m= 0:N-1
    k= mod(n0- m, N);
    y_time(n0+1)= y_time(n0+1)+ x(m+1)* h(k+1);
  endfor
endfor

y_time

%β) στο πεδίο συχνοτήτων
%β1) με fft
%μετασχηματισμός γραμμών και στηλών
X= fft(x, N);
H= fft(h, N);

Y= X.*H; %πολ/μος στοιχείων
y_fft= ifft(Y)

%β2) χωρίς fft
%πίνακας DFT (W) 4x4
x_col= x.';
h_col= h.';

i= (0:N-1).'; %στήλες
j= 0:N-1;     %γραμμές
W= exp(-1j*2*pi/N * (i*j)); %W(k,n)= e^(-j2πkn/N)= iDFT

X= W*x_col;
H= W*h_col;
Y= X.*H; %DFT[x]*DFT[h]

y= real((1/N)*conj(W).'*Y).'
```

Και επιβεβαιώνουμε ότι τα αποτελέσματα συμπίπτουν:

```
y_time =

   6   7   6   5

y_fft =

   6   7   6   5

y =

   6   7   6   5
```

## Άσκηση 5:

**Απόκριση Συχνότητας.** Δίνεται σύστημα διακριτού χρόνου με Απόκριση Συχνότητας:

$$H(ei\omega) = (1 + 2e\hat{}i\omega)/(1 - 0.2e\hat{}i\omega)$$

Να σχεδιαστεί το πραγματικό, το φανταστικό μέρος, το μέτρο και η φάση της Απόκρισης Συχνότητας.

Ο κώδικας ξεκινάει με τον ορισμό της ανάλυσης του πλέγματος συχνοτήτων N. Μαζί με την επόμενη εντολή **W= linspace(-pi, pi, N)** δημιουργούμε ένα διάνυσμα [-π, π] με 512 τιμές. Το διάνυσμα αυτό θα αποτελέσει τον οριζόντιο άξονα των διαγραμμάτων.

Συνεχίζουμε με τον υπολογισμό του e^iω με την εντολή **eiw= exp(1i*W)**. Έτσι δημιουργούμε έναν πίνακα μιγαδικών αριθμών (eiw).

Ορίζουμε στην μεταβλητή H τον τύπο που δίνεται στην εκφώνηση και υπολογίζουμε τις ζητούμενες συνιστώσες με την σειρά που παρουσιάζονται στην εκφώνηση:

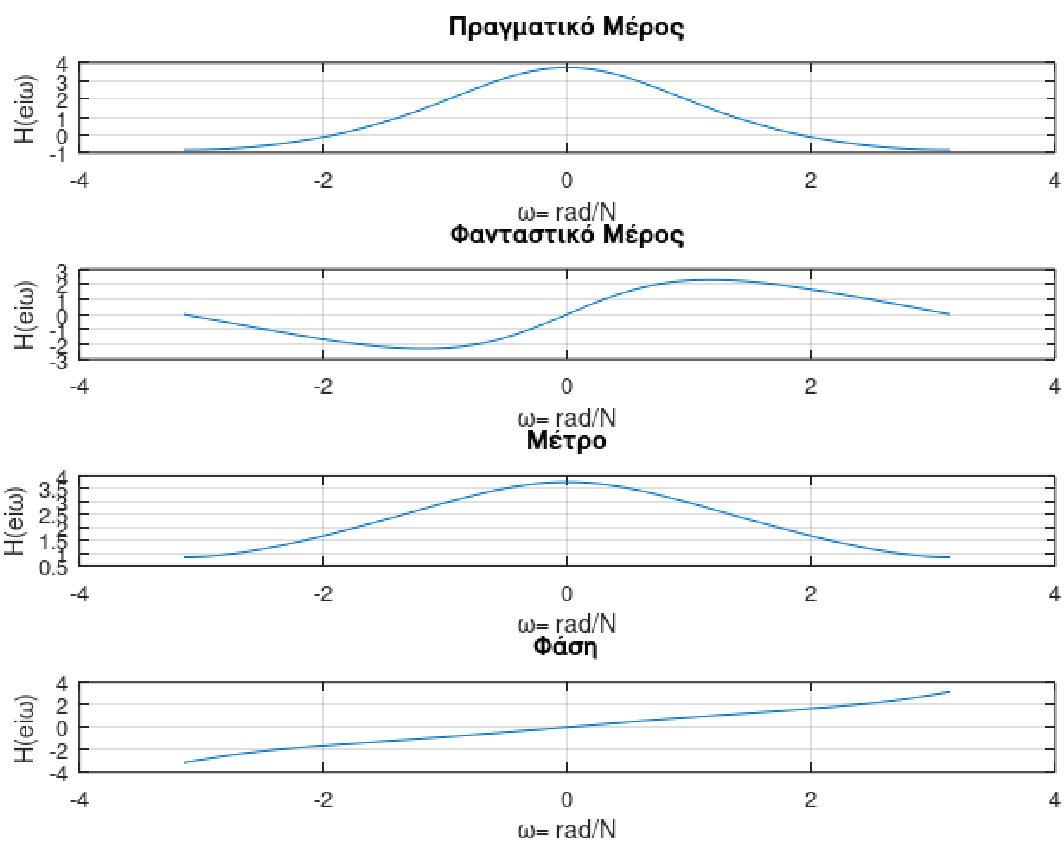
- H_real= **real**(H); *--πραγματικό μέρος*
- H_imag= **imag**(H); *--φανταστικό*
- H_met= **abs**(H); *--μέτρο*
- H_phase= **angle**(H); *--φάση*

Και τέλος, εμφανίζουμε τα διαγράμματα τους.

Κώδικας ***main.m***:

```
N= 512; %ανάλυση πλέγματος συχνοτήτων
W= linspace(-pi, pi, N); %ω= rad/N
eiw= exp(1i*W);

%εκφώνηση
H= (1+2*eiw)./(1-0.2*eiw);

%υπολογισμός μερών
H_real= real(H);
H_imag= imag(H);
H_met= abs(H);
H_phase= angle(H);

%γραφικές παραστάσεις
figure;
subplot(4,1,1);
plot(W, H_real);
```

```
xlabel('ω= rad/N');
ylabel('H(eiω)');
title('Πραγματικό Μέρος');

subplot(4,1,2);
plot(W, H_imag);
xlabel('ω= rad/N');
ylabel('H(eiω)');
title('Φανταστικό Μέρος');

subplot(4,1,3);
plot(W, H_met);
xlabel('ω= rad/N');
ylabel('H(eiω)');
title('Μέτρο');

subplot(4,1,4);
plot(W, H_phase);
xlabel('ω= rad/N');
ylabel('H(eiω)');
title('Φάση');
```



## Άσκηση 6:

**Ιδιότητα συνέλιξης DTFT.** Σύστημα διακριτού χρόνου χαρακτηρίζεται από την ακόλουθη κρουστική απόκριση :

$$h[n] = \delta[n] - 2\delta[n-1] + 3\delta[n-2] - 2\delta[n-3] + \delta[n-4].$$

Εαν η είσοδος του συστήματος είναι το σήμα :

$$x[n] = \delta[n] + 3\delta[n-1] + 5\delta[n-2] + 7\delta[n-3] + 5\delta[n-4] + \cdots$$

$$11\delta[n-5] + 13\delta[n-6] + 17\delta[n-7] + 18\delta[n-8] + \cdots$$

$$+21\delta[n-9] + 12\delta[n-10]$$

Να βρεθεί ο DTFT της εξόδου του συστήματος, α) με χρήση της συνέλιξης στο πεδίο του χρόνου και β) ως γινόμενο του DTFT της κρουστικής απόκρισης επί του DTFT της εισόδου. Να αναπαρασταθούν γραφικά και στις δυο περιπτώσεις τα φάσματα πλάτους και φάσης, και να συγκριθούν τα αποτελέσματα.

α) <u>με χρήση της συνέλιξης στο πεδίο του χρόνου:</u>

```
%συντελεστές
h= [1 -2 3 -2 1];
x= [1 3 5 7 5 11 13 17 18 21 12];
N= 512;
%με συνέλιξη στο πεδίο του χρόνου
y_time= conv(x, h);
Y= fft(y_time, N);
```

Ξεκινάμε με το να ορίσουμε του συντελεστές από την εκφώνηση και να υπολογίζουμε την συνέλιξη στον χρόνο με την συνάρτηση ***conv***, η οποία ισούται και με την έξοδο του συστήματος y[n]. Έπειτα κάνουμε μετασχηματισμό Fourier στην έξοδο.

β) <u>με το γινόμενο του DTFT της κρουστικής και του DTFT της εισόδου:</u>

```
%με γινόμενο
X= fft(x, N);
H= fft(h, N);

Y_dtft= X.*H;
```

Εδώ υπολογίζουμε τον μετασχηματισμό Fourier για την είσοδο του συστήματος X και την κρουστική απόκριση H, και έπειτα το γινόμενο τους.

```
%Άξονας συχνότητας
w= linspace(-pi, pi, N); %ω= rad/N
X= fftshift(X);
H= fftshift(H);
Y= fftshift(Y);
Y_dtft= fftshift(Y_dtft);

%Γραφική απεικόνιση πλάτους
figure;
plot(w, abs(Y));
hold on;
plot(w, abs(Y_dtft), '--');
xlabel('ω (rad/N)');
ylabel('Y(eiω)');
title('Πλάτος DTFT');

%Γραφική απεικόνιση φάσης
figure;
plot(w, angle(Y));
hold on;
plot(w, angle(Y_dtft), '--');
xlabel('ω= rad/N');
ylabel('Φάση (rad)');
title('Φάση DTFT');
```

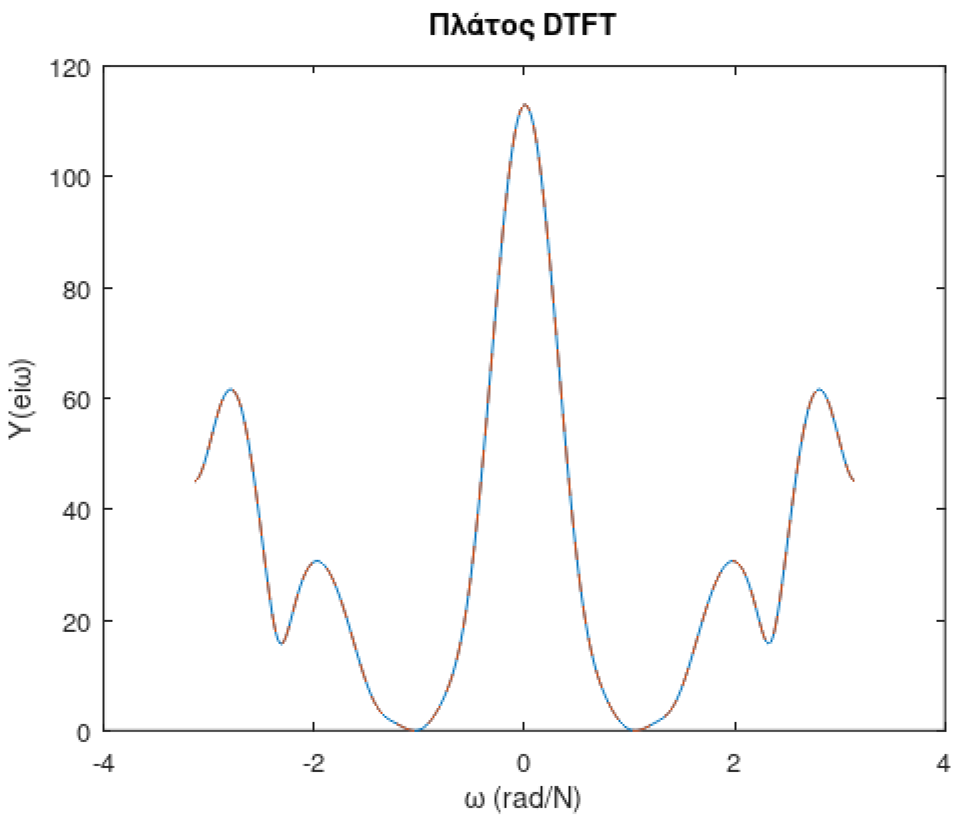
Τέλος, δημιουργούμε τις γραφικές παραστάσεις του πλάτους και της φάσης. Για το πλάτος και την φάση ξεχωριστά, αποτυπώνουμε τις καμπύλες των δύο περιπτώσεων στο ίδιο διάγραμμα όπου μπλέ για α) και καφέ για β).

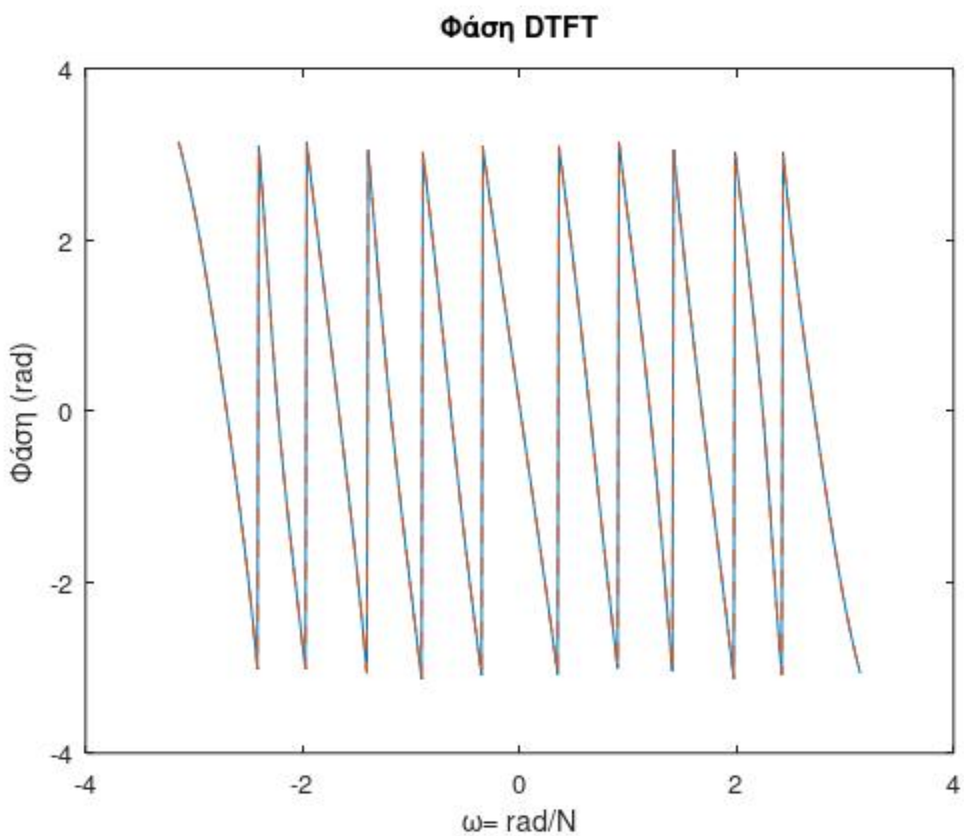
Έχει προηγηθεί και η χρήση της συνάρτησης ***fftshift*** με την οποία μετατοπίζουμε το φάσμα για την εμφάνιση του πλάτους στο διάστημα [-π, π].

Τελικά, ολόκληρο το αρχείο ***main.m*** γίνεται:

```
%συντελεστές
h= [1 -2 3 -2 1];
x= [1 3 5 7 5 11 13 17 18 21 12];

%με συνέλιξη στο πεδίο του χρόνου
y_time= conv(x, h);
Y= fft(y_time, N);

%με γινόμενο
N= 512;
H= fft(h, N);
X= fft(x, N);

Y_dtft= X.*H;

%Άξονας συχνότητας
w= linspace(-pi, pi, N); %ω= rad/N
X= fftshift(X);
H= fftshift(H);
Y= fftshift(Y);
Y_dtft= fftshift(Y_dtft);

%Γραφική απεικόνιση μέτρου
figure;
plot(w, abs(Y));
hold on;
plot(w, abs(Y_dtft), '--');
xlabel('ω (rad/N)');
ylabel('Y(eiω)');
title('Πλάτος DTFT');

%Γραφική απεικόνιση φάσης
figure;
plot(w, angle(Y));
hold on;
plot(w, angle(Y_dtft), '--');
xlabel('ω= rad/N');
ylabel('Φάση (rad)');
title('Φάση DTFT');
```

Και από την εκτέλεση του παίρνουμε τις παρακάτω γραφικές παραστάσεις. Παρατηρούμε ότι και στις δύο περιπτώσεις τα αποτελέσματα έχουμε πλήρη επικάλυψη.


Πλάτος DTFT
Y(ejω)
ω (rad/N)


Φάση DTFT
Φάση (rad)
ω= rad/N

## Άσκηση 7:

**DFT.** Δίνεται το σήμα διακριτού χρόνου :

$$x[n] = \delta[n] - \delta[n-1] + 2\delta[n-2] - 2\delta[n-3]$$

να υπολογιστεί ο DFT 4 σημείων, του σήματος α) με χρήση fft β) χωρίς τη χρήση fft, με πολλαπλασιασμό με τον πίνακα παραγόντων στροφής 4x4 (χρησιμοποιείστε την έτοιμη συνάρτηση dftmtx). Στη συνέχεια, υπολογίστε τον iDFT του DFT που βρήκατε (δηλαδή, το αποτέλεσμα θα είναι το ίδιο το σήμα) με 3 τρόπους α) με χρήση της συνάρτησης fft, β) χωρίς τη χρήση fft μέσω της σχέσης

$$x[n] = idft(X(k)) = \frac{1}{N} dft(X^*(k))$$

και γ) χωρίς τη χρήση fft μέσω της σχέσης :

$$x[n] = idft\big(X(k)\big) = \frac{1}{N} \; W_N^* \; dft(X(k))$$

(όπου *: συζυγής μιγαδικός)

Τέλος, υπολογίστε την ενέργεια του σήματος x[n] α) στο πεδίο του χρόνου και β) στο πεδίο της συχνότητας.

Επειδή δουλεύουμε στο Octave, ο κώδικας ξεκινάει με την φόρτωση του πακέτου ***signal*** το οποίο περιέχει την συνάρτηση ***dftmtx*** που θα χρειαστούμε αργότερα. Συνεχίζουμε με τον υπολογισμό του DFT 4 σημείων με *fft* και με πολλαπλασιασμό πίνακα παραγόντων.

```
pkg load signal;

x= [1 -1 2 -2];
N= 4;

%======== DFT ========
%α) με fft
x_dft= fft(x, N);

%β) με dftmax
M= dftmtx(N); %δημιουργία 4x4 πίνακα DFT
x_mtx= M*x.'; %x.' -> στήλη
```

Ορίζουμε τους συντελεστές του σήματος μας στον πίνακα ***x*** και το πλήθος τους. Για το πρώτο ερώτημα απλώς χρησιμοποιούμε την συνάρτηση ***fft*** για να κάνουμε μετασχηματισμό Fourier. Για το δεύτερο δημιουργούμε τον 4x4 πίνακα παραγόντων μέσω της συνάρτησης ***dftmax*** και κάνουμε πολ/μο των στοιχείων του πίνακα αυτού και του πίνακα ***x***.

Συνεχίζουμε με τον υπολογισμό του iDFT με χρήση της ***ifft*** και με χρήση των σχέσεων που βρίσκονται στην εκφώνηση:

```
%======== iDFT ========
%α) με ifft
x_idft= ifft(x_fft);

%β) χωρίς ifft μέσω πρώτης σχέσης
x_idft1= (1/N)* (fft(conj(x_fft), N));

%γ) χωρίς ifft μέσω δεύτερης σχέσης
xcol= x_fft.';
x_mtx_col= (1/N)*M'*xcol;
x_idft2= x_mtx_col.';
```

Ο υπολογισμός μέσω της ***ifft*** είναι εύκολος καθώς απλώς καλούμε την συνάρτηση για το σήμα που έχει υποστεί DFT από πριν.

Στην επόμενη περίπτωση «μεταφράζουμε» την σχέση της εκφώνησης σε κώδικα matlab. Βρίσκουμε δηλαδή τον iDFT εκτελώντας μετασχηματισμό Fourier στον συζυγή του ήδη μετασχηματισμένου σήματος ***x_fft***, και πολλαπλασιάζοντας το αποτέλεσμα με 1/N.

Στην τρίτη περίπτωση παραθέτουμε το μετασχηματισμένο σήμα ***x_fft*** ως στήλη και εφαρμόζουμε την σχέση της εκφώνησης. Βρίσκουμε δηλαδή το γινόνενο του σήματος με τον πίνακα παραγόντων που έχουμε ήδη δημιουργήσει και το 1/N. Ως τελευταία ενέργεια, παραθέτουμε το αντεστραμένο σήμα από στήλη σε γραμμή, για να τυπωθεί ομοιόμορφα με τα προηγούμενα.

Στο τελευταίο ερώτημα ο κώδικας είναι:

```
%======== Ενέργεια x[n]========
%α) στο πεδίο του χρόνου
E_time= sum(abs(x).^2);

%β) στο πεδίο της συχνότητας
E_freq= (1/N)* sum(abs(x_dft).^2);
```

Βρίσκουμε την ενέργεια του σήματος στο πεδίο του χρόνου υπολογίζοντας το άθροισμα του απολύτου των υψωμένων στο τετράγωνο συντελεστών του σήματος. Η ενέργεια στο πεδίο της συχνότητας υπολογίζεται από την εξίσωση του Parseval.

Το αρχείο ***main.m*** τελειώνει με την εκτύπωση όλων των αποτελέσματων στην γραμμή εντολών για να συγκρίνουμε τα αποτελέσματα. Ο κώδικας ολοκληρωμένος είναι:

```
pkg load signal;

x= [1 -1 2 -2];
N= 4;

%======== DFT ========
%α) με fft
x_dft= fft(x, N);

%β) με dftmax
M= dftmtx(N); %δημιουργία 4x4 πίνακα DFT
x_mtx= M*x.'; %x.' -> στήλη

%======== iDFT ========
%α) με ifft
x_idft= ifft(x_fft);

%β) χωρίς ifft μέσω πρώτης σχέσης
x_idft1= (1/N)* (fft(conj(x_fft), N));

%γ) χωρίς ifft μέσω δεύτερης σχέσης
xcol= x_fft.';
x_mtx_col= (1/N)*M'*xcol;
x_idft2= x_mtx_col.';

%======== Ενέργεια x[n]========
%α) στο πεδίο του χρόνου
E_time= sum(abs(x).^2);

%β) στο πεδίο της συχνότητας
E_freq= (1/N)* sum(abs(x_dft).^2);

%Αποτελέσματα
disp('α) DFT με fft:');
disp(real(x_dft));
disp('β) DFT με dftmx:');
disp(real(x_mtx.'));
disp('====================')

disp('α) iDFT με ifft:');
disp(real(x_idft));
disp('β) iDFT με πρώτη σχέση:');
disp(real(x_idft1));
disp('γ) iDFT με δεύτερη σχέση:');
disp(real(x_idft2));
disp('====================')

disp('α) Ενέργεια στον χρόνο:');
disp(E_time);
disp('β) Ενέργεια στην συχνότητα:');
disp(E_freq);
```

Και δίνει τα εξής αποτελέσματα:

```
>> main

α) DFT με fft:
   0  -1   6  -1
β) DFT με dftmx:
   0  -1   6  -1
====================
α) iDFT με ifft:
   1  -1   2  -2
β) iDFT με πρώτη σχέση:
   1  -1   2  -2
γ) iDFT με δεύτερη σχέση:
   1  -1   2  -2
====================
α) Ενέργεια στον χρόνο:
10
β) Ενέργεια στην συχνότητα:
10
```

Όπως βλέπουμε, ο μετασχηματισμός DFT και ο αντίστροφος iDFT που εκτελέσαμε χωρίς τις συναρτήσεις ***fft*** και ***ifft*** αντίστοιχα, οδηγούν στο ίδιο αποτέλεσμα με τις έτοιμες συναρτήσεις. Συνεπώς είναι σωστά.

Βλέπουμε επίσης πως η ενέργεια στον χρόνο ισούται με την ενέργεια στην συχνότητα, επαληθεύοντας έτσι την σχέση του Parseval.

## Άσκηση 8:

**Συνέλιξη Overlap – add method**. Στην περίπτωση που η μία από τις δύο ακολουθίες που εμπλέκονται στη συνέλιξη είναι πολύ μεγαλύτερη από την άλλη χρησιμοποιείται η τεχνική του κατακερματισμού της μεγάλης ακολουθίας σε τμήματα μήκους ίσου με αυτό της μικρής. Για την τμηματική εκτέλεση της συνέλιξης μία μέθοδος που χρησιμοποιείται είναι η Overlap –add method. Περιληπτικά, η μέθοδος αυτή περιγράφεται ως εξής:

Η μεγάλου μήκους ακολουθία τμηματοποιείται σε μικρότερες ακολουθίες μήκους ίσου με το μήκος της μικρότερης (έστω N). Πραγματοποιείται συνέλιξη της «μικρής» ακολουθίας με τα δημιουργηθέντα τμήματα της μεγάλης. Για να δημιουργηθεί η τελική έξοδος της συνέλιξης, προστίθενται τα τελευταία N- 1 δείγματα της καθεμίας συνέλιξης με τα N-1 πρώτα της επόμενής της τμηματικής συνέλιξης. Τα υπόλοιπα δείγματα μένουν ανέπαφα. Ολες αυτές οι έξοδοι συνθέτουν την τελική έξοδο (συνέλιξη) των δύο ακολουθιών.

Σαν απλή εφαρμογή της παραπάνω μεθόδου, θεωρήστε το σήμα («μεγάλη ακολουθία», για λόγους

κατανόησης και οπτικοποίησης, το μήκος της ακολουθίας δεν είναι μεγάλο !)

$$x[n] = [1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10],\ 0 \le n \le 9$$

και το σήμα («μικρή ακολουθία»

$$h[n] = [\ 1\ 2\ 3\ 4],\ 0 \le n \le 3$$

Γράψτε τον κατάλληλο κώδικα για την συνέλιξη των δύο ακολουθιών με τη μέθοδο Overlap–add και συγκρίνετε το αποτέλεσμα με το αποτέλεσμα της conv.

Μετατρέποντας την παραπάνω ανάλυση σε κώδικα Matlab δημιουργούμε το παρακάτω αρχείο ***main.m***:

```
x= [1 2 3 4 5 6 7 8 9 10];
h= [1 2 3 4];

Nx= length(x);
Nh= length(h);

%Μήκος τμήματος == μήκος h
L= Nh;
overlap= Nh-1;

%Μήκος εξόδου
N_total= Nx+ Nh- 1;
y_overlap= zeros(1, N_total);

i= 1; %θέση στο x

%Για κάθε τμήμα x, κάνουμε συνέλιξη με το σήμα h
%και τα ενώνουμε
while i<= Nx
  x_block= x(i:min(i+L-1, Nx));

  y_block= conv(x_block, h);

  out_start= i; %θέση αρχής τμήματος
  out_end= i+length(y_block)-1; %θέση τέλους τμήματος

  %ένωση των τμημάτων μετά την συνέλιξη
  y_overlap(out_start: out_end)= y_overlap(out_start: out_end)+ y_block;

  i= i+L; %μετατόπιση δείκτη στο επόμενο τμήμα
endwhile

%Απλή συνέλιξη (χωρίς overlap)
y_conv= conv(x, h);

%Αποτελέσματα
disp('Αποτέλεσμα Overlap-Add:');
disp(y_overlap);
disp('======================');
disp('Αποτέλεσμα conv(x,h):');
disp(y_conv);
```

Ξεκινάμε ορίζοντας τους συντελεστές των ακολουθιών και τα αντίστοιχα μεγέθη τους. Αρχικοποιούμε επίσης και τον πίνακα ***y_overlap***, στον οποίο θα προστεθούν τα διάφορα τμήματα για να συνδεθούν.

Δημιουργούμε τον βρόχο επανάληψης ο οποίος θα χωρίσει το μεγάλο σήμα ***x*** σε τμήματα, απομονόνωντας μέρος του ***x*** από την θέση ***i*** μέχρι την ***(i+L-1)***, όπου ***i*** ο μετρητής της επανάληψης και ***L*** το μήκος της μικρής ακολουθίας, ή μέχρι το μήκος της μεγάλης ακολουθίας ***Nx***, όποιο είναι μικρότερο (χρήση συνάρτησης ***min***).

Έπειτα εκτελούμε την συνέλιξη του τμήματος που δημιουργήσαμε (***y_block)*** με το σήμα ***h[n],*** και ορίζουμε τα όρια του τμήματος μετά την συνέλιξη με τον υπολογισμό ***(i+L(y_block)-1)*** όπως παραπάνω.

Τέλος, προχωράμε στην συννένωση των τμημάτων σε ένα διάνυσμα ***y_overlap*** προσαυξάνωντας σε αυτό κάθε τμήμα, υπολογίζουμε την συνέλιξη μόνο με χρήση ***conv*** και τυπώνουμε τα αποτελέσματα των δύο μεθόδων.

```
>> main

Αποτέλεσμα Overlap-Add:
     1     4    10    20    30    40    50    60    70    80    79    66    40
========================
Αποτέλεσμα conv(x,h):
     1     4    10    20    30    40    50    60    70    80    79    66    40
>>
```

Βλέπουμε πως η υλοποίηση μας της μεθόδου Overlap-add βγάζει τα ίδια αποτελέσματα με την συνάρτηση ***conv***, επομένως η υλοποίηση μας είναι σωστή.

## Άσκηση 9:

**Moving Average filter.** Ας θεωρηθεί το σύστημα διακριτού χρόνου που περιγράφεται από την ακόλουθη σχέση εισόδου εξόδου (Moving Average filter, 5th order)

$$y[n] = 1/5\,(x[n] + x[n-1] + x[n-2] + x[n-3] + x[n-4])$$

Να βρεθεί η Απόκριση Συχνότητας του συστήματος α) μέσω DTFT με χρήση της ιδιότητας ολίσθησης, β) μέσω μετασχηματισμού Z γ) με χρήση της συνάρτησης freqz. Σε όλες τις περιπτώσεις να δημιουργηθούν τα γραφήματα των φασμάτων πλάτους και φάσης. Να συγκριθούν τα αποτελέσματα. Να διαπιστωθεί, μέσω του φάσματος πλάτους, ότι το σύστημα λειτουργεί ως χαμηλοπερατό φίλτρο.

α) Ξεκινάμε αρχικοποιώντας τον πίνακα με τους συντελεστές του συστήματος και ορίζοντας τον οριζόντιο άξονα στο διάστημα [-π, π].

```
%Συντελεστές moving average filter 5 σημείων
a= (1/5)*ones(1,5); %[1/5 1/5 1/5 1/5 1/5]

N= 512;
omega= linspace(-pi, pi, N);

%α) H(e^{-jω}) από ορισμό DTFT
ejw= exp(-1j*omega);
H_dtft= (1/5)*(1+ejw+ejw.^2+ejw.^3+ejw.^4);
```

Για τον υπολογισμό DTFT μέσω της ιδιότητας ολίσθησης, υπολογίζουμε το e^(-iω) σύμφωνα με το τον ορισμό του DTFT: ***X(e^jω)=∑x[n]e^(−jωn)***.

β) Από τις σημειώσεις του εργαστηρίου πάνω στον μετασχηματισμό Z έχουμε ότι

$$F(e^{j\omega}) = \sum_{n=-\infty}^{\infty} f(n)e^{-j\omega n}$$

και

$$F(z) = \sum_{n=-\infty}^{\infty} f(n)z^{-n}$$

Άρα αντικαθιστώντας το z με e^jω καταλήγουμε από τον ΜΖ απευθείας στον DTFT. Συνεπώς με τον παραπάνω υπολογισμό δεν χρειάζεται να κάνουμε άλλον υπολογισμό για να απαντήσουμε στο ερώτημα β. Συνεπώς:

```
%β) H(e^{jω}) από μετασχηματισμό-Z
H_z= H_dtft;
```

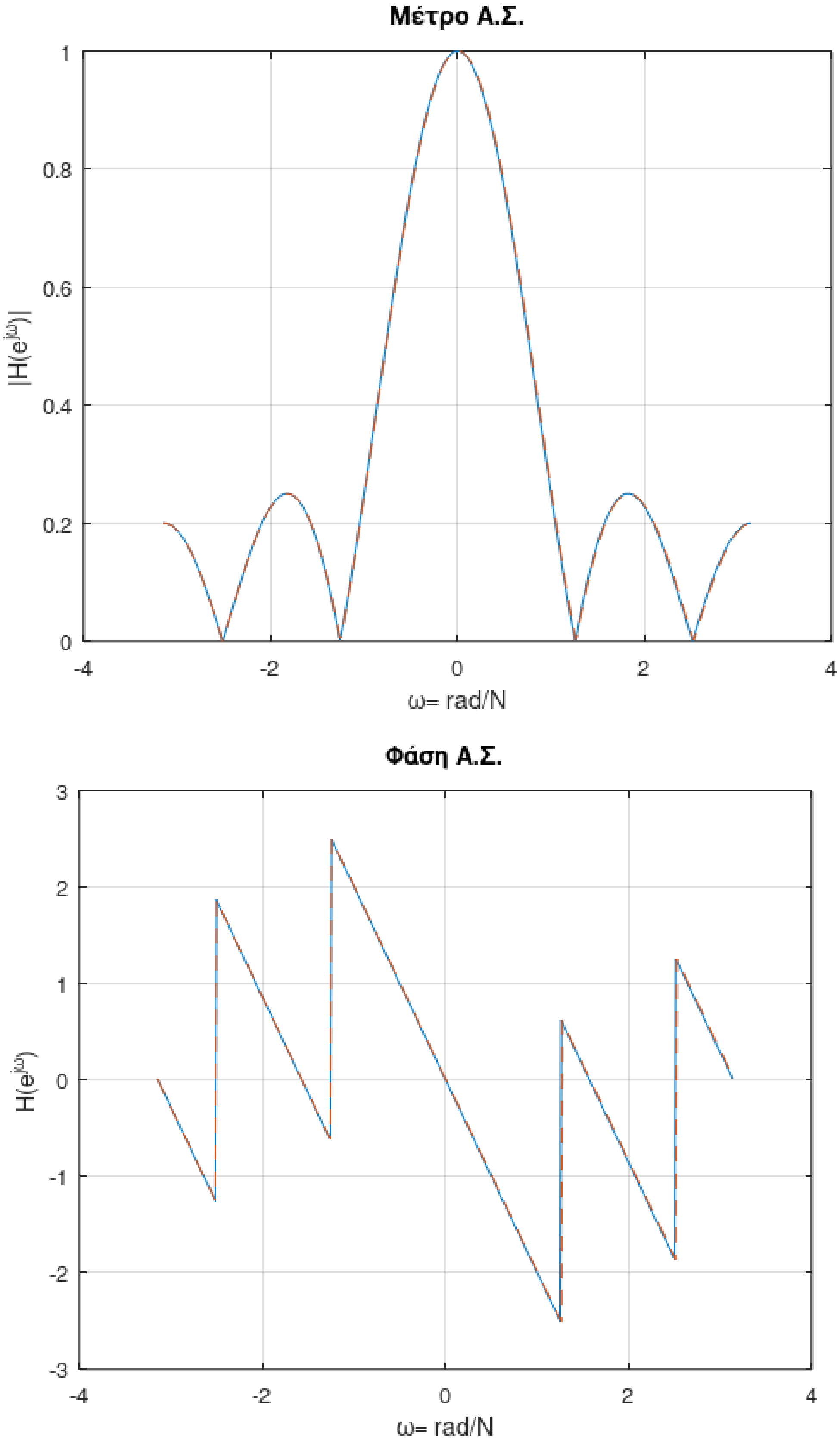
γ) Για την υλοποίηση με την συνάρτηση ***freqz*** παίρνουμε την μορφή ***[H_f, w_f]= freqz(a, 1, N, 'whole')*** η οποία υπολογίζει την Α.Σ. σε Ν σημεία και το διάστημα των συχνοτήτων ***w_f*** το οποίο ορίζεται στο διάστημα [0, π]. Τα ***a*** και ***1*** είναι οι συντελεστές του αριθμητή και του παρονομαστή της ***y[n]***. Με την προσθήκη του ***whole*** υπολογίζουμε την Α.Σ. για ολόκληρο τον μοναδιαίο κύκλο. Τέλος, με χρήση της ***fftshift***, μεταφέρουμε την ***H_f*** που βρήκαμε στο διάστημα [-π, π] για να εμφανιστεί το διάγραμμα με κέντρο το 0. Συνεπώς, ο κώδικας είναι:

```
%γ) H(e^{jω}) από freqz
w_f= 0:0.1:2*pi;
[H_f, w_f]= freqz(a, 1, N, 'whole');
H_fshift= fftshift(H_f);
```

Τέλος, ακολουθεί η δημιουργία και εμφάνιση των γραφικών παραστάσεων. Το ολοκληρωμένο αρχείο ***main.m*** γίνεται:

```
%Συντελεστές moving average filter 5 σημείων
a= (1/5)*ones(1,5); %[1/5 1/5 1/5 1/5 1/5]

N= 512;
omega= linspace(-pi, pi, N);

%α) H(e^{jω}) από ορισμό DTFT
ejw= exp(-1j*omega); %εδώ παίρνουμε -j
H_dtft= (1/5)*(1+ejw+ejw.^2+ejw.^3+ejw.^4);

%β) H(e^{jω}) από μετασχηματισμό-Z
H_z= H_dtft;

%γ) H(e^{jω}) από freqz
w_f= 0:0.1:2*pi;
[H_f, w_f]= freqz(a, 1, N, 'whole');
H_fshift= fftshift(H_f);

%Αποτελέσματα
figure;
plot(omega, abs(H_dtft));
hold on;
plot(omega, abs(H_fshift), '--');
xlabel('ω= rad/N');
ylabel('|H(e^{j\omega})|');
title('Μέτρο Α.Σ.');

figure;
plot(omega, angle(H_dtft));
hold on;
plot(w_fshift, angle(H_fshift), '--');
xlabel('ω= rad/N');
ylabel('H(e^{j\omega})');
title('Φάση Α.Σ.');
```

Και τα αποτελέσματα είναι:



Έχουμε εμφανίσει σε κάθε γράφημα την καμπύλη του μετασχηματισμού ***DTFT και Z***, μαζί με την καμπύλη της ***freqz.*** Όπως βλέπουμε όλες οι μέθοδοι μας δίνουν το ίδιο αποτέλεσμα, άρα η υλοποίηση μας είναι σωστή.

Από το φάσμα πλάτους (μέτρο) **|H(e^jω)|** βλέπουμε ότι:

- Για *ω= 0* έχουμε **H(e^j0)= 51(1+1+1+1+1)= 1**, δηλαδή η μηδενική συχνότητα περνάει χωρίς να επηρεαστεί.
- Για *ω> 0* ο όρος **1+e^(−jω)+e^(−j2ω)+e^(−j3ω)+e^(−j4ω)** δημιουργεί ακυρώσεις.
- Υπάρχουν επίσης και συχνότητες όπου το μέτρο μηδενίζεται, επηρεάζοντας σε μεγάλο βαθμό τις υψηλές συχνότητες.

Συνεπώς, το φίλτρο κρατάει τις χαμηλές συχνότητες και μειώνει τις υψηλές. Έχει, δηλαδή, την συμπεριφορά ενός χαμηλοπερατού φίλτρου.

## Άσκηση 10:

**Απόκριση συστήματος σε ημιτονοειδή είσοδο**. Ας θεωρηθεί το σύστημα διακριτού χρόνου με Απόκριση Συχνότητας :

$$H(ej\omega) = (2 - 3e{-}j\omega + 0.1e{-}j2\omega)/(1 - 0.3e{-}j\omega)$$

Να υπολογιστεί η έξοδος του συστήματος όταν η είσοδος είναι το σήμα :

$$x[n] = 3\cos(\pi n/5 + \pi/4)$$

Ποια θα είναι η σχέση του πλάτους του σήματος εισόδου με το πλάτος της εξόδου και της φάσης του σήματος εισόδου με τη φάση της εξόδου?

Από το σήμα της εισόδου αμέσως μπορούμε να πούμε ότι:

- **ω**= 5/4
- **Πλάτος εισόδου**= 3
- **Φάση Εισόδου**= π/4

Εφόσον μας δίνεται η απόκριση συχνότητας του συστήματος (Α.Σ.) καταλαβαίνουμε ότι πρόκειται για **ΓΧΑ** σύστημα. Αυτό σημαίνει ότι η **έξοδος** υπολογίζεται από την σχέση:

**y[n]= πλάτος_εξόδου* cos(ωn+ φάση_εξόδου)**

Το **πλάτος εξόδου** με την σειρά του υπολογίζεται από την σχέση:

**Αεξ= πλάτος_εισόδου*πλάτος_ΑΣ**

Και η **φάση εξόδου** από την σχέση:

**Φεξ= φάση_εισόδου + φάση_ΑΣ**

Για να βρούμε το πλάτος και την φάση της ΑΣ πρέπει πρώτα να την υπολογίσουμε. Βρίσκουμε αρχικά το *e^(-jω)* και με αυτό υπολογίζουμε την ΑΣ για *ω= π/5*.

Τώρα μπορούμε να βρούμε τα ζητούμενα **Αεξ** και **Φεξ** εφαρμόζοντας τις συναρτήσεις ***abs(H)*** και ***angle(H)*** αντίστοιχα και να υπολογίσουμε τελικά την έξοδο **y[n].**

Στο τέλος, τυπώνουμε τα αποτελέσματα των παραπάνω υπολογισμών.

Συνεπώς, ο κώδικας στο αρχείο **main.m** έχει ως ακολούθως:

```
fr_in= pi/5; %συχνότητα εισόδου
platos_in= 3; %πλάτος εισόδου
ph_in= pi/4; %φάση εισόδου
omega= pi/5; %ω= π/5

%e^{-jω}
ejw= exp(-1j*omega);

%Υπολογισμός Α.Σ.
H= (2-3*ejw+0.1*ejw.^2)/(1-0.3*ejw);

%πλάτος και φάση της Α.Σ.
platosH= abs(H);
phaseH= angle(H);

%πλάτος και φάση εξόδου
platos_out= platos_in* platosH;
phase_out= ph_in+ phaseH;

%έξοδος
n= 0:40; %δείγματα στον χρόνο
y_out= platos_out* cos(omega*n+ phase_out);

%Αποτελέσματα
disp('H(e^{-jω}= '); disp(H);
disp('Πλάτος H= '); disp(platosH);
disp('Φάση H= '); disp(phaseH);
disp('=============');
```

```
disp('Πλάτος εισόδου= '); disp(platos_in);
disp('Φάση εισόδου= '); disp(ph_in);
disp('==============');

disp('Πλάτος εξόδου= '); disp(platos_out);
disp('Φάση εξόδου= '); disp(phase_out);
disp('Έξοδος= '); disp(y_out.');
disp('==============');
```

## Άσκηση 11:

**Περιοδικότητα.** Να σχεδιαστούν τα παρακάτω σήματα και για όσα είναι περιοδικά να προσδιοριστεί η θεμελιώδης περίοδός τους και να γίνει γραφική επαλήθευση της περιοδικότητας.

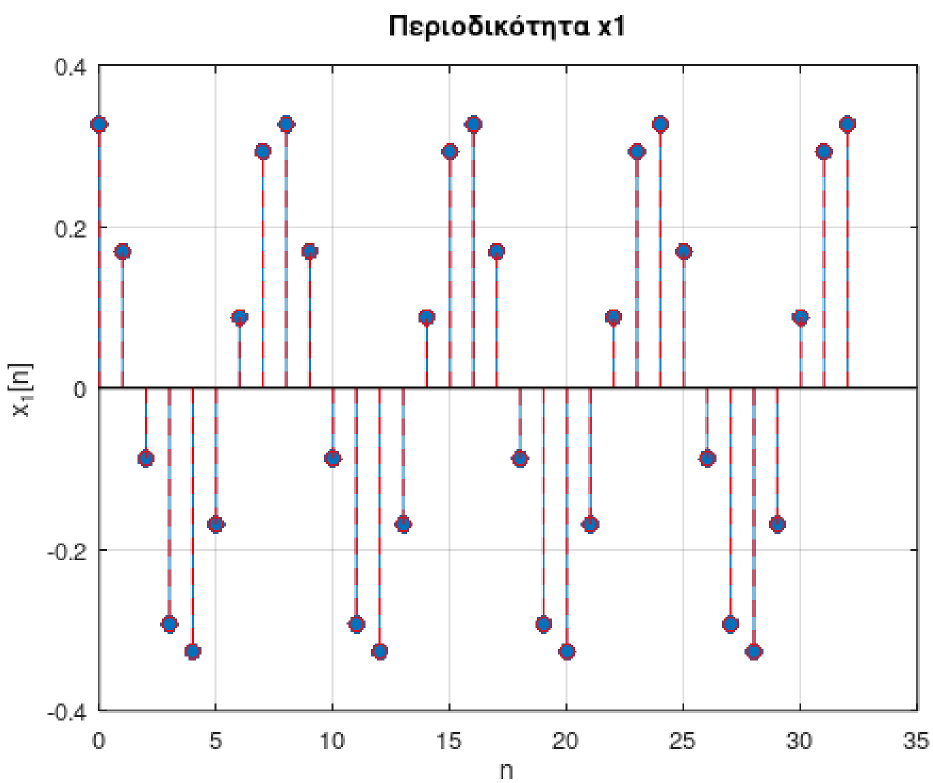
$$a)\quad x_1[n] = \frac{AM}{1000} cos(\frac{\pi n}{4} + \frac{\pi}{12})$$

$$b)\quad x_2[n] = 2\ cos^2(\frac{\pi}{6} n + \frac{\pi AM}{1000})$$

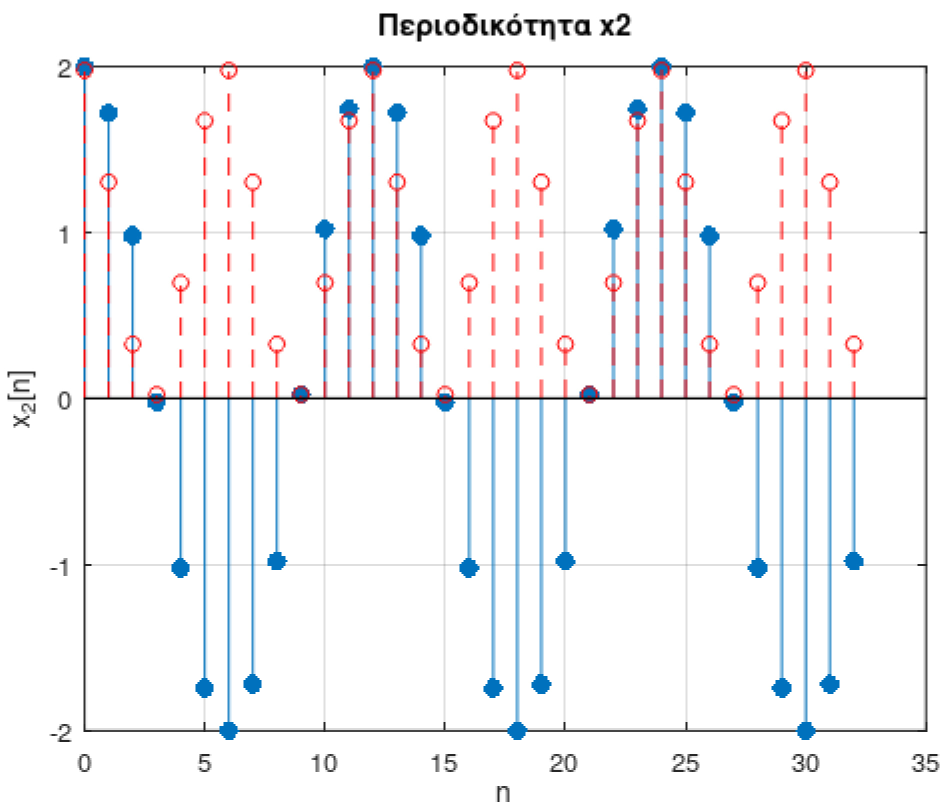
$$c)\quad x_3[n] = e^{j\pi n/5} + e^{-j\pi n/5}$$

Οπου ΑΜ τα 4 τελευταία ψηφία του Αριθμού Μητρώου σας *(0338)*.

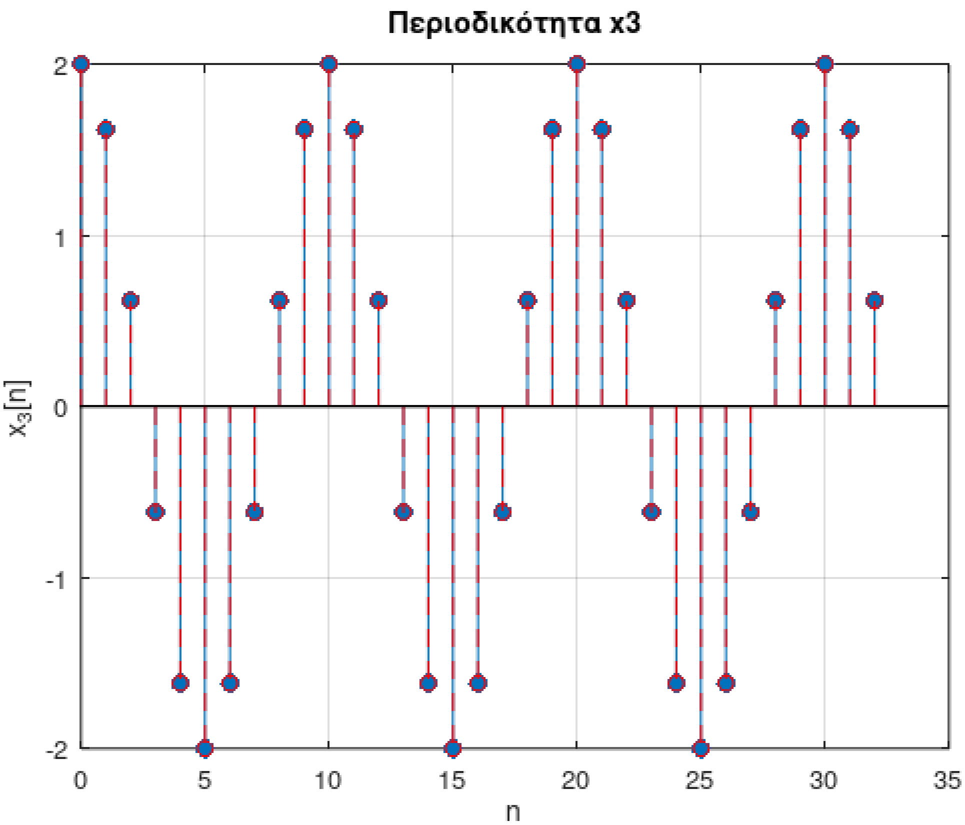
α) Για να είναι ένα διακριτό ημιτονοειδές σήμα ***x[n]=A*cos(ωn+φάση)*** περιοδικό, πρέπει να υπάρχει ακέραιο $N_0$ τέτοιο ώστε **$\omega_0 N_0$= 2kπ.** Στο σήμα x1 έχουμε $\omega_0$= π/4 => **$N_0$= 8k** άρα το x1 **είναι** περιοδικό με **θεμελιώδη περίοδο $N_0$= 8.** Βλέπουμε στο διάγραμμα ότι το σήμα πέφτει πάνω στο μετατοπισμένο κατά μία περίοδο σήμα, άρα η περιοδικότητα επαληθεύεται.



β) Το x2 είναι της μορφής ***x2[n]=2cos^2(ωn+φάση)*** με **ω0**= π/6 και **φάση**= 0.338π. Η συχνότητα της εξόδου είναι **$\omega_0$'=** $2\omega_0$= **π/3** και για να είναι ένα διακριτό συνημίτονο περιοδικό πρέπει **$\omega_0$'*$N_0$= 2kπ**. Άρα π/3N= 2kπ => **N= 6k.** Συνεπώς το σήμα είναι περιοδικό με **θεμελιώδη περίοδο** $N_0$= 6. Στο διάγραμμα μπορούμε εύκολα να διακρίνουμε και να επαληθεύσουμε την περιοδικότητα (**η φάση δεν την επηρεάζει**).



γ) Το $x3[n] = e^{(j\pi n/5)} + e^{(-j\pi n/5)}$ είναι άθροισμα δύο συζυγών φασικών εκθετικών. Θα είναι περιοδικό όταν $\omega_0 N_0 = 2k\pi$, άρα $(5/\pi)N = 2k\pi \Rightarrow N_0 = 10k$. Συνεπώς το σήμα είναι και αυτό περιοδικό με **θεμελιώδη περίοδο** $N_0 = 10$. Και εδώ το σήμα πέφτει πάνω στο μετατοπισμένο σήμα, άρα η περιοδικότητα επαληθεύεται.




Η διαδικασία υπολογισμού στο Octave είναι παρόμοια και για τις τρεις περιπτώσεις. Συνεπώς το αρχείο **main.m** περιλαμβάνει τις εξής εντολές:

```
n= 0:32; %0 έως 32 -> 4 περίοδοι

%α) x1[n] = 0338/1000*cos(πn/4 + π/12)
A= 0.338;
omega= pi/4;
phase= pi/12;

x1= AM*cos(omega*n+ phase);
N= 8; %θεμελιώδης περίοδος

%μετατόπιση σήματος κατά 1 περίοδο
%για επαλήθευση
x1_shift= A*cos(omega*(n+N1)+ phase);

figure;
stem(n, x1, 'filled');
```

```
hold on;
stem(n, x1_shift, 'r--');
grid on;

xlabel('n');
ylabel('x_1[n]');
title('Περιοδικότητα x1');

%β) x2[n] = 2 cos^2(π/6 n + π0338/1000)
A= 2;
omega= pi/6;
phase= 0.338/pi;

x2= A* cos(omega*n+ phase.^2);
N= 6;

%μετατόπιση για επαλήθευση
x2_shift= A*cos(omega*(n+N)+ phase).^2;

figure;
stem(n, x2, 'filled');
hold on;
stem(n, x2_shift, 'r--');
grid on;

xlabel('n');
ylabel('x_2[n]');
title('Περιοδικότητα x2');

%γ) x3[n] = e^(jπn/5) + e^(-jπn/5)
x3= exp(1j*pi*n/5)+ exp(-1j*pi*n/5);
N= 10;

x3_shift= exp(1j*pi*(n+N)/5)+ exp(-1j*pi*(n+N)/5);

figure;
stem(n, real(x3), 'filled');
hold on;
stem(n, real(x3_shift), 'r--');
grid on;

xlabel('n');
ylabel('x_3[n]');
title('Περιοδικότητα x3');
```

## Άσκηση 12:

**Συνέλιξη**. Να υπολογιστεί η έξοδος συστήματος LTI

α) με κρουστική απόκριση $h[n] = |n-3|(u[n] - u[n-6])$ και είσοδο

$$x[n] = \begin{cases} -1, & n = -1 \\ 1, & n = 0 \\ 2, & n = 1 \\ -1, & n = 2 \end{cases},$$

Θέτοντας $-10 \le n \le 10$.

β) με κρουστική απόκριση $h[n] = 0.6nu[n]$ και είσοδο $x[n] = u[n] - u[n-10]$

Θέτοντας $-10 \le n \le 30$

α) Ξεκινάμε υπολογίζοντας της κρουστική απόκριση h[n].

u[n]= 1, n>= 0 αλλιώς 0 =>

=> u[n-6= 1, n-6>=0 => n>= 6, αλλιώς 0

Συνεπώς ο όρος **u[n]- u[n-6]** είναι **1 για 0<= n<= 5 και 0 αλλού**.

Βάζοντας μέσα και τον παράγοντα |n-3| έχουμε ότι

**h[n]=** |n-3|(u[n]- u[n-6])= **|n-3| για 0<= n<= 5 και 0 αλλού**.

Τώρα θα υπολογίσουμε τις τιμές h[n] για n [0, 5]:

- h[0]= |0-3|= 3
- h[1]= |1-3|= 2
- h[2]= 1
- h[3]= 0
- h[4]= 1
- h[5]= 2

Τελευταίο βήμα του υπολογισμού είναι οι περιοχές στήριξης.

- x[n] =/= 0 για n= [-1, 2]
- h[n] =/= 0 για n= [0, 5]

Άρα η συνέλιξη y[n]= (x * h)[n] έχει μη μηδενικές τιμές για:

Nmin= -1+0= -1, Nmax= 2+5= 7

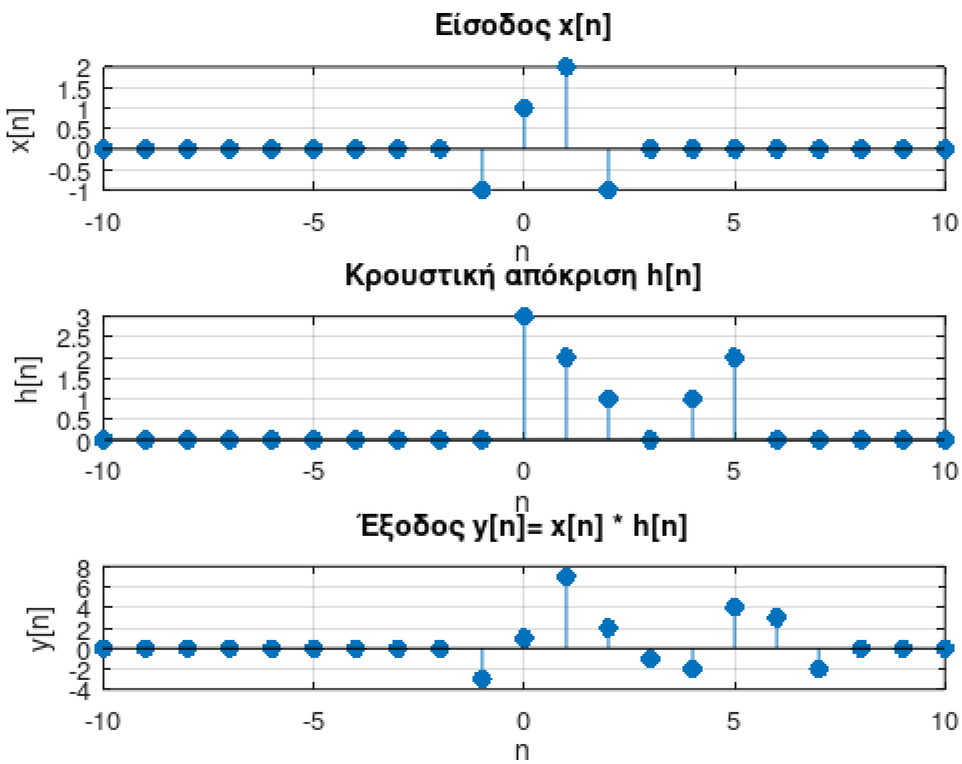
και γενικά y[n]= 0 εκτός του διαστήματος [-1, 7].

Έχοντας κάνει αυτούς τους υπολογισμούς, προχωράμε στην εύρεση της εξόδου του συστήματος μέσω κώδικα στο αρχείο **mainA.m:**

```
n= -10:10 %περιοχή του χρόνου

%Είσοδος x[n]
x= zeros(size(n));
x(n== -1)= -1;
x(n== 0)= 1;
x(n== 1)= 2;
x(n== 2)= -1;

%Κρουστική h[n]= |n-3|(u[n] - u[n-6])
h= zeros(size(n));
idx= (n>= 0) & (n<= 5);
h(idx)= abs(n(idx)- 3);

%Συνέλιξη y[n]= x[n] * h[n]
y_conv= conv(x, h);

%Αποτελέσματα
figure;
subplot(3,1,1);
stem(n, x, 'filled');
grid on;
xlabel('n');
ylabel('x[n]');
title('Είσοδος x[n]');

subplot(3,1,2);
stem(n, h, 'filled');
grid on;
xlabel('n');
ylabel('h[n]');
title('Κρουστική απόκριση h[n]');

subplot(3,1,3);
stem(n, y, 'filled');
grid on;
xlabel('n');
ylabel('y[n]');
title('Έξοδος y[n]= x[n] * h[n]');
```

- Τα x, h ορίζονται για n= [-10, 10]
- idx είναι ένας λογικός πίνακας που είναι αληθής μόνο για n= [0, 5]
- Στις θέσεις αυτές του idx βάζουμε |n-3|
- Οι υπόλοιπες παραμένουν 0
- Η conv(x, h) υπολογίζει την συνέλιξη

Αποτέλεσμα είναι οι παρακάτω γραφικές παραστάσεις:



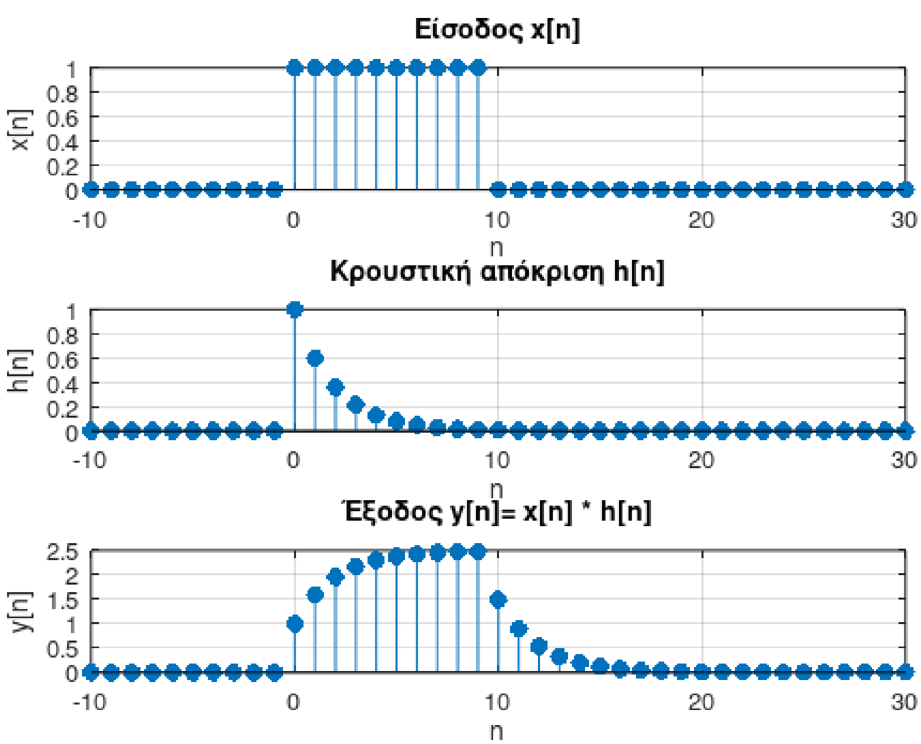

β) Εδώ έχουμε μία αιτιατή εκθετική κρουστική απόκριση η οποία μηδενίζεται για n< 0, και για είσοδο έναν ορθογώνιο παλμό μήκους 10, δηλαδή x[n]= 1 για ν= [0, 9] και 0 αλλού. Υπολογίζουμε παρακάτω μέσω κώδικα την ζητούμενη έξοδο για n= [-10, 10].

**<u>mainB.m:</u>**

```
n= -10:30;

%Είσοδος x[n]= u[n] - u[n-10]
x= zeros(size(n));
x((n>= 0) & (n<= 9))= 1; %1 για 0<=n<=9, αλλιώς 0

%Κρουστική απόκριση h[n]= 0.6^n u[n]
```

```
h= (0.6.^n) .* (n>= 0); %0.6^n για n>=0, αλλιώς 0

%y[n]= x[n] * h[n]
y_conv= conv(x, h);

figure;
subplot(3,1,1);
stem(n, x, 'filled');
grid on;
xlabel('n');
ylabel('x[n]');
title('Είσοδος x[n]');

subplot(3,1,2);
stem(n, h, 'filled');
grid on;
xlabel('n');
ylabel('h[n]');
title('Κρουστική απόκριση h[n]');

subplot(3,1,3);
stem(n, y, 'filled');
grid on;
xlabel('n');
ylabel('y[n]');
title('Έξοδος y[n]= x[n] * h[n]');
```

Με αποτέλεσμα:



## **Άσκηση 13:**

**Ανάλυση σε απλά κλάσματα**. Να αναλυθεί σε άθροισμα απλών κλασμάτων η ρητή συνάρτηση:

$$X(z) = \frac{2z^2 + 3z - 1}{z^3 - 5z^2 + 8z - 4}$$

Να γίνει και η θεωρητική ανάλυση και να συγκριθούν τα αποτελέσματα.

Ξεκινάμε με την θεωρητική επίλυση. Παραγοντοποιώντας τον παρονομαστή το κλάσμα έρχεται στην μορφή

$$X(z)= A/(z-1) + B/(z-2) + Γ/(z-2)^2$$

Πολλαπλασιάζοντας με (z-2)^2*(z-1), γίνεται:

$$2z^2+ 3z- 1= A(z-2)^2 + B(z-1)(z-2)+ Γ(z-1)$$

Και έτσι έχουμε τις τρεις εξισώσεις που χρειαζόμαστε:

| | | |
|---|---|---|
| • A+B= 2 | | A= 4 |
| • -4A-3B+Γ=3 | => | B= -2 |
| • 4A+2B-Γ= -1 | | Γ= 13 |

Συνεπώς το κλάσμα αναλύεται σε:

$$\mathbf{X(z)= 4/(z-1) – 2(z-2) + 13(z-2)^2}$$

Η επίλυση στο Octave γίνεται με το παρακάτω αρχείο **main.m** με χρήση της ***residue***:

```
pkg load signal;

a= [0 2 3 -1];
b= [1 -5 8 -4];

[r, p, k]= residue(a, b);
disp('A, B, Γ: '); disp(r);
disp('Πόλοι: '); disp(p);
```

```
>> main

A, B, Γ:
   -2
   13
    4
Πόλοι:
   2
   2
   1
```

Βλέπουμε ότι τα αριθμητικά αποτελέσματα των A, B, Γ και των πόλων είναι τα ίδια.

## Άσκηση 14:

**Μετατροπές σήματος διακριτού χρόνου.** Να δημιουργηθεί το σήμα διακριτού χρόνου

$$x[n] = [1\ 2\ \underset{\uparrow}{\underbrace{3}}\ 4\ 5\ 6\ 7\ 6\ 5\ 4\ 3\ 2\ 1]$$

Σε έξι υποπαράθυρα του ίδιου γραφικού παράθυρου να δημιουργηθούν οι γραφικές παραστάσεις των: $a)\ x[n]$, $b)\ x[n-5]$, $c)\ x[n+4]$, $d)\ x[-n]$, $e)\ x[n/2]$, $f)\ x[2n]$

Βλέπουμε στην εκφώνηση ότι το βελάκι έχει ορίσει $n_0$= 3 άρα θα πάρουμε ως n το διάστημα [-2, 10] έτσι ώστε x[0]= 3.

Στον κώδικα δημιουργούμε 2 αρχεία, το κυρίως **main.m** και την συνάρτηση **local_x.m.** Η συνάρτηση μας επιστρέφει το x[n] αν το n βρίσκεται μέσα στο [-2, 10], αλλιώς μας επιστρέφει 0. Στην main.m κάνουμε τον απαραίτητο μετασχηματισμό για κάθε υποερώτημα και σχηματίζουμε τις γραφικές παραστάσεις. Συνεπώς:

**local_x.m:**

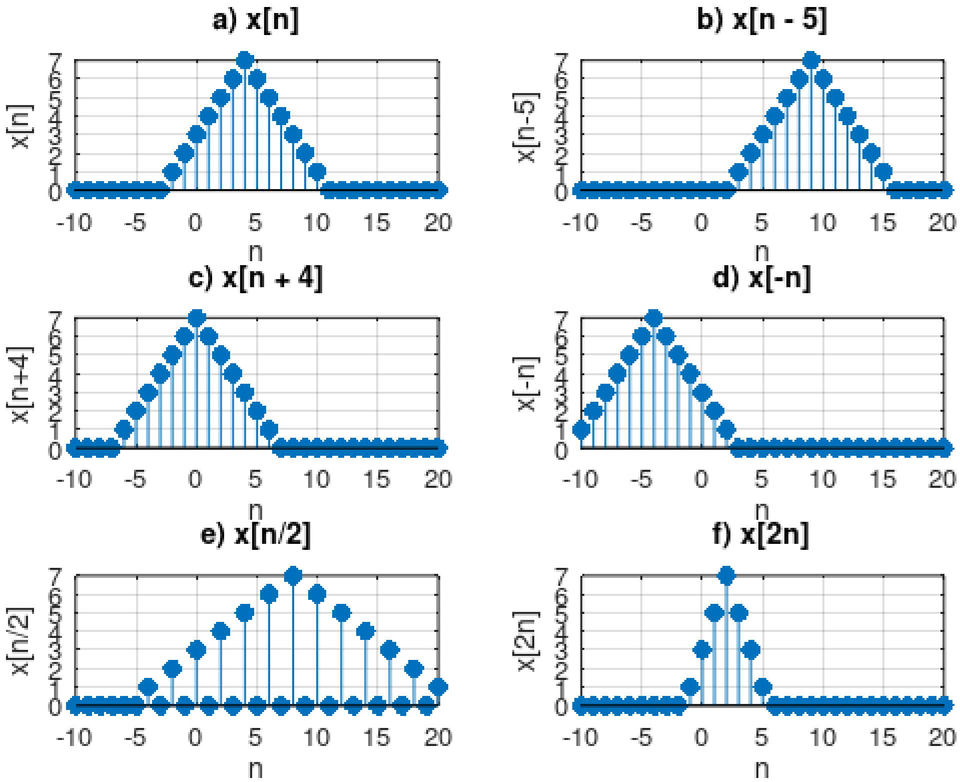
```
function y = local_x(m, n0, x0)
    %επιστρέφει x[m] αν m είναι μέσα στο n0, αλλιώς 0
    idx= find(n0 == m, 1);
    if isempty(idx)
        y= 0;
    else
        y= x0(idx);
    endif
endfunction
```

**main.m:**

```
%Αρχικό σήμα x[n] με n0 [-2, 10]
n0= -2:10;
x0= [1 2 3 4 5 6 7 6 5 4 3 2 1];

%Περιοχή χρόνου σχεδίασης
n= -10:20;

%Συνάρτηση που επιστρέφει το x[n] για οποιοδήποτε n (αν έξω από [-2,10] => 0)
xn= @(nn) arrayfun(@(m) local_x(m, n0, x0), nn);

%α) x[n]
xa= xn(n);

%β) x[n - 5]
xb= xn(n-5);
```

```
%γ) x[n + 4]
xc= xn(n+4);

%δ) x[-n]
xd= xn(-n);

%ε) x[n/2]  (για περιττά n, το n/2 δεν είναι ακέραιο => έξοδος 0)
xe= xn(n/2);

%στ) x[2n]
xf= xn(2*n);

%Σχεδίαση γραφικών παραστάσεων
figure;
subplot(3,2,1);
stem(n, xa, 'filled'); grid on;
title('a) x[n]');
xlabel('n'); ylabel('x[n]');

subplot(3,2,2);
stem(n, xb, 'filled'); grid on;
title('b) x[n - 5]');
xlabel('n'); ylabel('x[n-5]');

subplot(3,2,3);
stem(n, xc, 'filled'); grid on;
title('c) x[n + 4]');
xlabel('n'); ylabel('x[n+4]');

subplot(3,2,4);
stem(n, xd, 'filled'); grid on;
title('d) x[-n]');
xlabel('n'); ylabel('x[-n]');

subplot(3,2,5);
stem(n, xe, 'filled'); grid on;
title('e) x[n/2]');
xlabel('n'); ylabel('x[n/2]');

subplot(3,2,6);
stem(n, xf, 'filled'); grid on;
title('f) x[2n]');
xlabel('n'); ylabel('x[2n]');
```

Με αποτέλεσμα:



## Άσκηση 15:

**Συνέλιξη με πίνακα Toeplitz**. Να γραφεί συνάρτηση η οποία να υλοποιεί τη συνέλιξη δύο ακολουθιών υπολογίζοντας τη συνέλιξη ως γινόμενου διανυσματικού πίνακα Toeplitz επί διάνυσμα.

Με χρήση της συνάρτησης να υπολογιστεί η συνέλιξη των ακολουθιών $z[n] = x[n] * y[n]$, όπου:

$$x[n] = 4\delta[n+1] + 9\delta[n] - 2\delta[n-2]$$

$$y[n] = \left(\frac{n}{6}\right)(u(n+2) - u(n-3))$$

Να γίνει η γραφική παράσταση των $z[n]$, $x[n]$, $y[n]$ στους κατάλληλους άξονες. Τέλος, να επαληθευτεί το αποτέλεσμα με τη χρήση της conv().

Η θεωρία πίσω από την συνέλιξη με πίνακα Toeplitz μας λέει ότι έχοντας δύο ακολουθίες x[n] μήκους L και h[n] μήκους M, η συνέλιξη τους y[n] μήκους L+M-1 γράφεται ως

**y= Tx*h ή y= Th*x**

όπου πίνακας Toeplitz T, ένας πίνακας όπου κάθε διαγώνιος έχει ίδια τιμή.

Από την ακολουθία x[n] βρίσκουμε ότι

- x[−1] = 4
- x[0] = 9
- x[2] = −2
- αλλιώς 0

άρα θέτουμε περιοχή του χρόνο n= [-5, 10].

Από την ακολουθία y βρίσκουμε ότι u[n+2]−u[n−3]= 1 για n= [-2, 2], άρα

y[n]= n/6 για n= [-2, 2], αλλιώς 0.

Το αρχείο μας **main.m** είναι:

```
% Περιοχή χρόνου
n= -5:10;
nz= -10:20;

%x[n]
x= zeros(size(n));
x(n== -1)= 4;
x(n==  0)= 9;
x(n==  2)= -2;

%Ορισμός y[n]
y= zeros(size(n));
idx= (n >= -2) & (n <= 2);
y(idx)= n(idx)/6;

%Συνέλιξη με Toeplitz
Tx= toeplitz([x(:); zeros(length(y)-1,1)], [x(1) zeros(1,length(y)-1)]);
z_toe= Tx * y(:);
z_toe= z_toe(1:length(x)+length(y)-1);

%Επαλήθευση με conv
z_conv= conv(x, y);

%Αποτελέσματα
figure;
subplot(3,1,1);
stem(n, x, 'filled'); grid on;
title('x[n]');
xlabel('n'); ylabel('x');

subplot(3,1,2);
stem(n, y, 'filled'); grid on;
title('y[n]');
xlabel('n'); ylabel('y');

subplot(3,1,3);
stem(nz, z_toe, 'filled'); hold on;
stem(nz, z_conv, 'r--');
grid on;
title('z[n] = x[n] * y[n]');
xlabel('n'); ylabel('z');
```

Στην αρχή δημιουργούμε τις ακολουθίες x και y με τα αποτελέσματα που υπολογίσαμε παραπάνω. Έπειτα δημιουργούμε τον πίνακα Toeplitz ως εξής:

***Tx= toeplitz([x(:); zeros(length(y)-1,1)], [x(1) zeros(1,length(y)-1)]);***

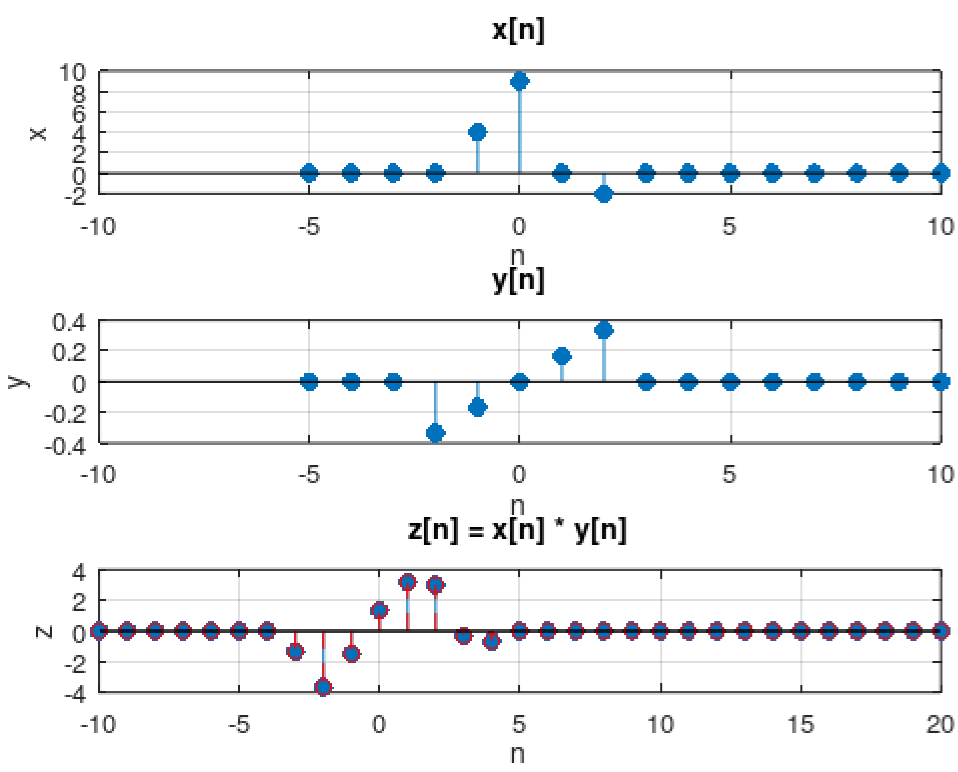
- [x(:); zeros(length(y)-1,1)]
  - η πρώτη στήλη του πίνακα παίρνει όλα τα στοιχεία του x σε μορφή στήλης
  - και μετά προσθέτει length(y)-1 μηδενικά
- [x(1) zeros(1,length(y)-1)]
  - η πρώτη γραμμή έχει ως πρώτο στοιχείο το x(1)
  - και τα υπόλοιπα 0

Με τον τρόπο αυτό εξασφαλίζουμε ότι ο πίνακας Toeplitz έχει L+M-1 γραμμές όπου

- L= length(x)
- M= length(y)

και ότι το σύστημα ξεκινάει από x[0] και οι επόμενες στήλες περιέχουν τις μετατοπίσεις του x. Ακολουθεί η συνέλιξη πολ/ζοντας τον πίνακα με το y σε μορφή στήλης.

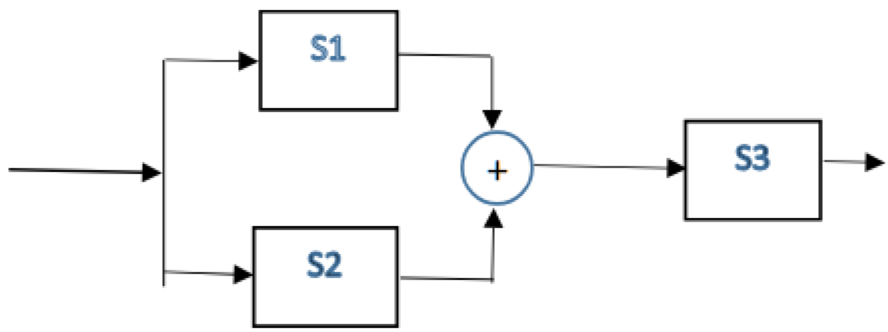
Αποτέλεσμα:




Όπως βλέπουμε η συνέλιξη μέσω Toeplitz ταιριάζει απόλυτα με την συνέλιξη μέσω conv στο διάγραμμα μας. Συνεπώς, το αποτέλεσμα της συνέλιξης επαληθεύεται.

## Άσκηση 16:

**Σύνδεση συστημάτων**. Δίνονται τα ΓΧΑ συστήματα S1, S2, S3 τα οποία είναι συνδεδεμένα μεταξύ τους όπως φαίνεται στο σχήμα:




και περιγράφονται από τις εξισώσεις διαφορών:

$$S1: 2y[n] + y[n-1] + 0.5y[n-2] = 0.1x[n-1] + 0.1x[n-3]$$

$$S2: y[n] = 2x[n] + x[n-2]$$

$$S3: y[n] - 0.5y[n-2] + 0.8y[n-3] = x[n] + 0.4x[n-1] - 1.4x[n-2]$$

Να βρεθεί η συνολική κρουστική απόκριση της διάταξης με χρήση της impz() και να αναπαρασταθεί γραφικά.

Από το σχήμα βλέπουμε ότ τα συστήματα S1 και S2 είναι συνδεδεμένα σε σειρά, άρα προστίθενται και μετά μπαίνουν στο S3. Συνεπώς:

Sολ= S3(S1+S2)

Για S1:

Μετασχηματισμός Z -> Αριθμητής b1= $0.1z^{-1} + 0.1z^{-3}$ = [0 0.1 0 0.1]

Παρονομαστής a1= $2+ z^{-1}+ 0.5z^{-2}$ = [2 1 0.5]

Για S2:

Μετασχηματισμός Z -> b2= [2 0 1]

a2= 1;

Για S3:

Μετασχηματισμός Z -> b3= [1 0.4 -1.4]

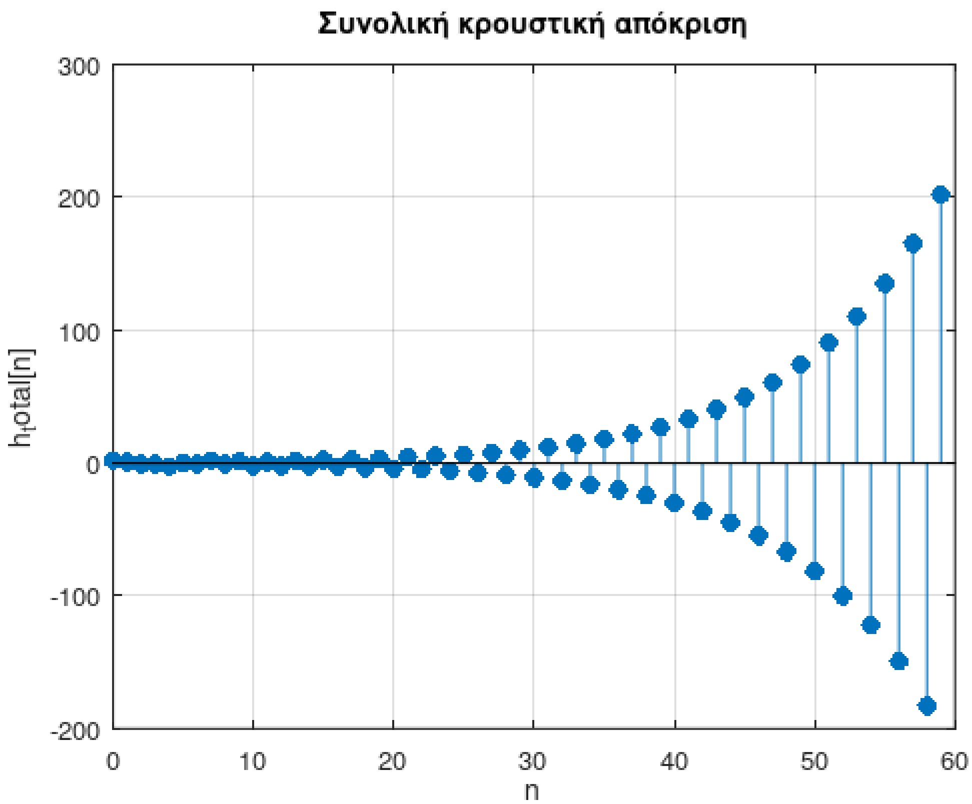
a3= [1 0 -0.5 0.8]

Για να κάνουμε την παράλληλη σύνδεση σύνδεση S1+S2 θα κάνουμε πρόσθεση των σημάτων με εξίσωση κοινών παρονομαστών. Μετά την συνέλιξη για την σύνδεση με το S3 κάνουμε σειριακή σύνδεση. Συνεπώς ο κώδικας μας είναι:

**main.m**:

```
%S1
b1= [0 0.1 0 0.1];
a1= [2 1 0.5];

%S2
b2= [2 0 1];
a2= 1;

%S3
b3= [1 0.4 -1.4];
a3= [1 0 -0.5 0.8];

%S1+S2 - παράλληλη σύνδεση
c1= conv(b1, a2);   % αριθμητής1
c2= conv(b2, a1);   % αριθμητής2

%μετατροπή στο ίδιο μήκος
L= max(length(c1), length(c2));
c1= [c1, zeros(1, L- length(c1))];
c2= [c2, zeros(1, L- length(c2))];

Bpar = c1 + c2;      %αριθμητής S1,2
Apar = conv(a1, a2); %παρονομαστής S1,2

%Σύνδεση S3(S1+S2)
b_total = conv(b3, Bpar);
a_total = conv(a3, Apar);

%Συνολική κρουστική απόκριση
N= 60;
[h_total, n]= impz(b_total, a_total, N);

stem(n, h_total, 'filled');
grid on;
xlabel('n');
ylabel('h_total[n]');
title('Συνολική κρουστική απόκριση');
```

Με αποτέλεσμα:



## Άσκηση 17:

**Φάσμα σήματος.** Ας θεωρηθεί το σήμα:

$$x[n] = 2cos(3\pi n/4),\ -10 \leq n \leq 10.$$

Να βρεθεί το φάσμα του σήματος αυτού με χρήση της fft σε 512 σημεία. Αναπαραστήστε το φάσμα πλάτους σε γραφικό παράθυρο. Παρατηρήστε τη διασπορά συχνοτήτων σε εύρος μεγαλύτερο από τις αναμενόμενες θεωρητικά φασματικές γραμμές του συνημιτόνου. Το φαινόμενο αυτό οφείλεται στον περιορισμό της άπειρης διάρκειας του συνημιτόνου και ονομάζεται φασματική διασπορά.

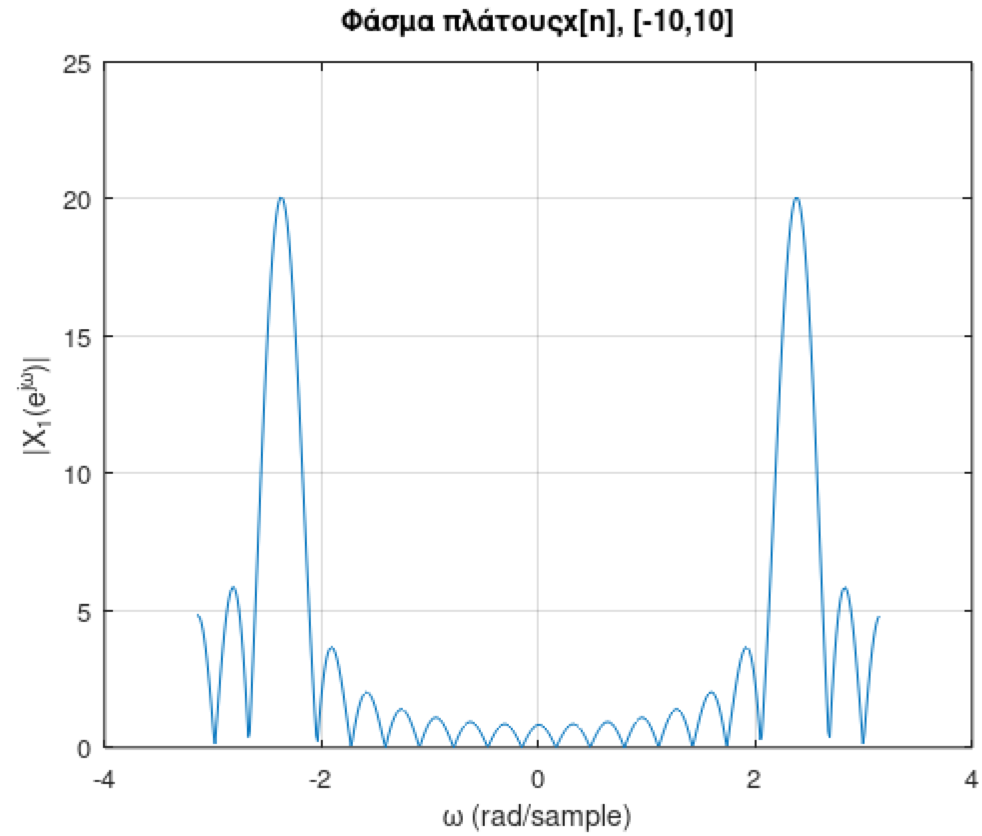
Επαναλάβετε την ίδια διαδικασία για το σήμα x[n], θεωρώντας αυτή τη φορά $-40 \leq n \leq 40$. Σχολιάστε το φάσμα που προκύπτει.

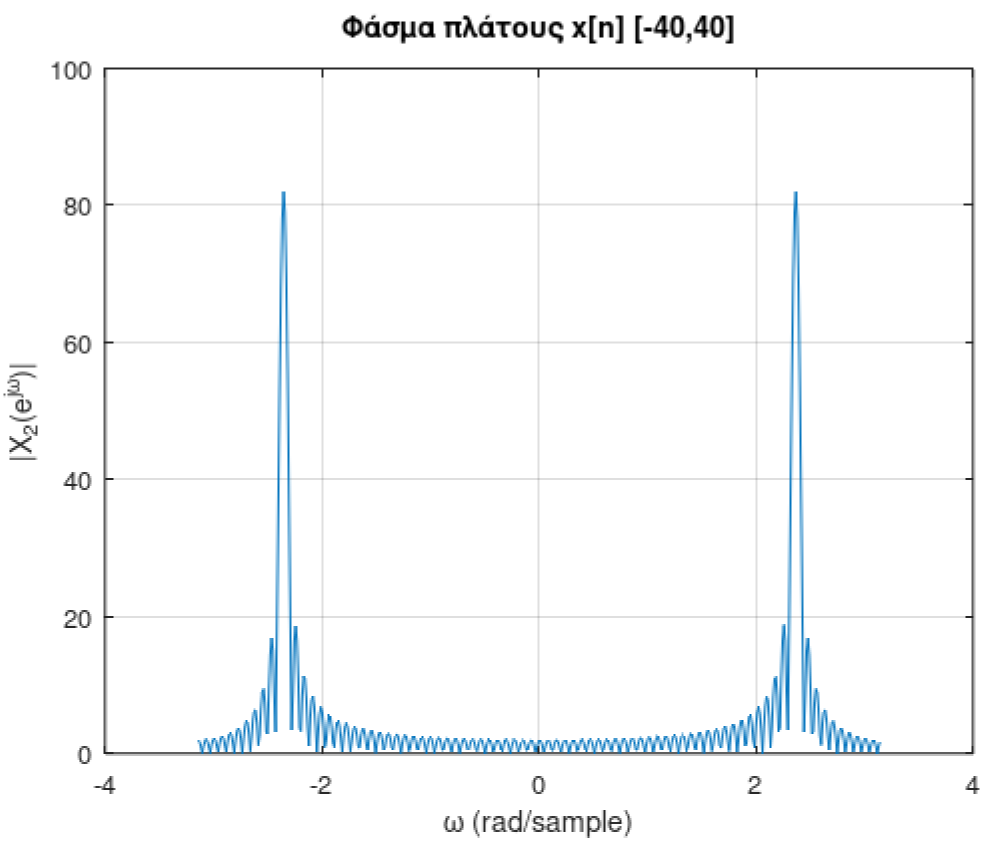
Ο κώδικας της άσκησης παρουσιάζεται ολόκληρος και για τα δύο n διανύσματα. Ο τρόπος εύρεσης του φάσματος είναι γνωστός και από προηγούμενες ασκήσεις, συνεπώς:

**main.m:**

```
Nfft= 512; % μέγεθος FFT

% -10 <= n <= 10
n1= -10:10;
x1= 2*cos(3*pi*n1/4); % x[n]

X1= fft(x1, Nfft);    % FFT με zero padding μέχρι 512
X1s= fftshift(X1);    % μεταφορά σε [-π, π]
w= linspace(-pi, pi, Nfft);

figure;
plot(w, abs(X1s)); grid on;
xlabel('\omega (rad/sample)');
ylabel('|X_1(e^{j\omega})|');
title('Φάσμα πλάτουςx[n], [-10,10]');

% -40 <= n <= 40
n2 = -40:40;
x2 = 2*cos(3*pi*n2/4);

X2 = fft(x2, Nfft);
X2s= fftshift(X2);

figure;
plot(w, abs(X2s)); grid on;
xlabel('\omega (rad/sample)');
ylabel('|X_2(e^{j\omega})|');
title('Φάσμα πλάτους x[n] [-40,40]');
```

**Αποτελέσματα:**



Όταν το διάστημα παρατήρησης αυξάνεται από [-10, 10] σε [-40, 40], το φάσμα συγκεντρώνεται περισσότερο κοντά στις ιδανικές συχνότητες. Οι δύο κορυφές στις ±3π/4 γίνονται πιο στενές και ταυτόχρονα πιο ψηλές ενώ οι υπόλοιπες κορυφές γίνονται πυκνότερες και μικρότερες.

Επίσης οσο μεγαλώνει η διάρκεια του σήματος το φάσμα γίνεται πιο στενό και συνεπώς η συνέλιξη με τα δέλτα συγκεντρώνει το φάσμα πιο κοντά στις ιδανικές γραμμές, δηλαδή η φασματική διασπορά μειώνεται.

## Άσκηση 18:

**Γραμμικότητα συστήματος.** Να διερευνηθεί με γραφικό τρόπο εάν το σύστημα $y[n]$ = $2x[n]$ είναι γραμμικό. Ως είσοδοι να θεωρηθούν τα σήματα $x1[n]$ = $0.8n$, $x2[n]$ = $cos[n]$, 0 ≤ $n$ ≤ 5. Ως σταθερές να θεωρηθούν οι α=2 και β=3. Σε υποπαράθυρα του ίδιου γραφικού παραθύρου να σχεδιαστούν οι είσοδοι x1, x2 και οι δύο έξοδοι.

Για να είναι το σύστημα γραμμικό πρέπει να ισχύει

$$\mathbf{y[n]=T(\alpha x1[n]+\beta x2[n]) = \alpha T(x1[n])+\beta T(x2[n])}$$

Αντικαθιστώντας παίρνουμε:

$$2(\alpha x1[n]+\beta x2[n]) = \alpha(2x1[n])+\beta(2x2[n])$$

το οποίο ισούται με:

$$\underline{2\alpha x1+2\beta x2=2\alpha x1+2\beta x2}$$

Συνεπώς το σύστημα είναι πράγματι γραμμικό.

Αυτό σε κώδικα αποδεικνύεται ως εξής:

**main.m:**

```
n= 0:5;

%Είσοδοι
x1= 0.8*n;
x2= cos(n);

%Συντελεστές
a= 2;
b= 3;

%Σύστημα y[n] = 2x[n]
y1= 2*x1;
y2= 2*x2;

%Ελεγχος γραμμικότητας
left= 2*(a*x1 + b*x2);   % T(a x1 + b x2)
right= a*y1 + b*y2;     % a T(x1) + b T(x2)

%Αποτελέσματα
figure;
subplot(3,1,1);
stem(n, x1, 'filled'); hold on;
stem(n, x2, 'r', 'filled');
title('Είσοδοι x_1[n] και x_2[n]');
legend('x_1[n]', 'x_2[n]');
grid on;

subplot(3,1,2);
stem(n, left, 'filled');
title('T(α x_1[n] + β x_2[n])');
grid on;

subplot(3,1,3);
stem(n, right, 'filled');
title('α T(x_1[n]) + β T(x_2[n])');
grid on;
```

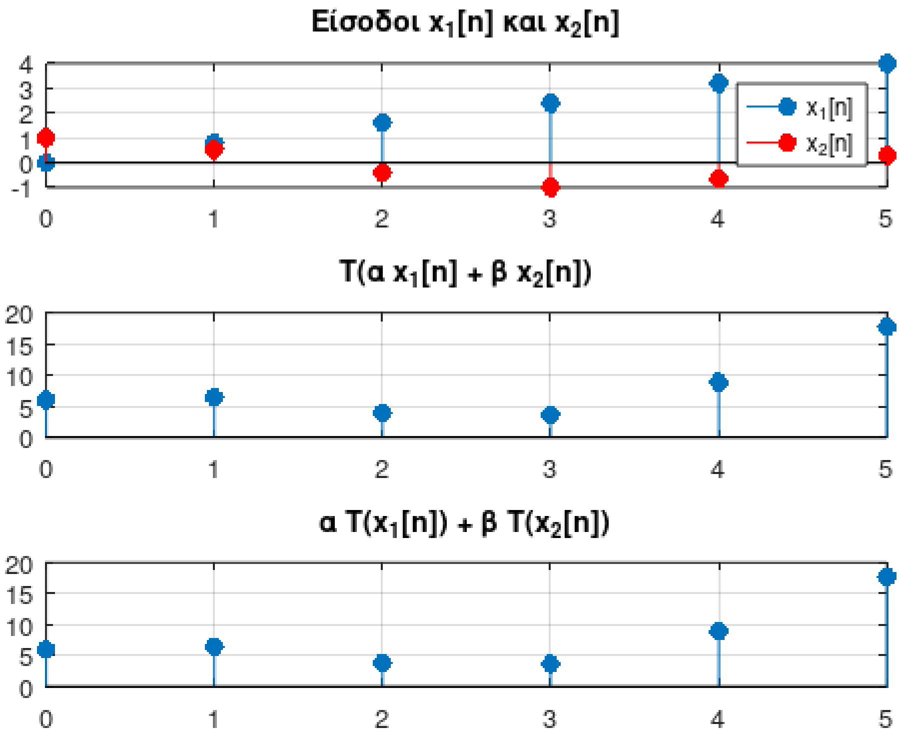
Για να αποδείξουμε την γραμμικότητα αρκεί να δούμε το 2° και το 3° subplot όπου οι τιμές τους είναι ακριβώς οι ίδιες. Συνεπώς ισχύει ότι:

**T(a x1 + b x2) = a T(x1) + b T(x2)**

**=>**

**T(x[n])=2x[n]**

Άρα το σύστημα είναι πράγματι γραμμικό.


Είσοδοι x1[n] και x2[n]
x1[n]
x2[n]
T(α x1[n] + β x2[n])
α T(x1[n]) + β T(x2[n])
4 3 2 1 0 -1
20 15 10 5 0
0 1 2 3 4 5

## <u>Άσκηση 19:</u>

**Ενέργεια σήματος.** Με βάση το θεώρημα Parseval για τον DFT η ενέργεια ενός σήματος διακριτού χρόνου μπορεί να υπολογιστεί από τον τύπο:

$$E = \frac{1}{N}\sum_{k=0}^{N-1} |X_k|^2$$

Θεωρήστε το σήμα $x[n] = 1/(n+1)$ , $0 \leq n \leq 10$ και επιβεβαιώστε ότι η ενέργεια του σήματος μπορεί να βρεθεί από τον παραπάνω τύπο.

**<u>main.m:</u>**

```
%x[n]
n= 0:10;
x= 1./(n+1);

N= length(x);

%Ενέργεια στο χρόνο
E_time= sum(abs(x).^2);

%Ενέργεια από DFT
X= fft(x);
E_freq= (1/N) * sum(abs(X).^2);

%Αποτελέσματα
disp('Ενέργεια στο χρόνο:');
disp(E_time);

disp('Ενέργεια από το DFT (Parseval):');
disp(E_freq);
```

Υπολογίζουμε την ενέργεια του σήματος της εκφώνησης και με τον τύπο υπολογισμού της στον χρόνο **Σ|x[n]|^2**, αλλά και με τον τύπο του Parseval όπως δίνεται παραπάνω. Εμφανίζοντας τα αποτελέσματα επιβεβαιώνουμε πως τα αποτελέσματα είναι ίσα:

```
Ενέργεια στο χρόνο:
1.5580
Ενέργεια από Parseval:
1.5580
```

## Άσκηση 20:

**Ευστάθεια συστήματος**. Δίνεται σύστημα Γραμμικό Χρονικά Αναλλοίωτο με κρουστική απόκριση $h[n] = 0.7^n\, u[n]$. Να βρεθεί η συνάρτηση μεταφοράς του, να διερευνηθεί αν είναι ευσταθές και να γίνει το διάγραμμα πόλων μηδενικών.

Η συνάρτηση μεταφοράς του συστήματος υπολογίζεται από τον ορισμό του μετασχηματισμού Z:

$$\sum_{n=0}^{\infty} f(n) z^{-n}$$

Αντικαθιστώντας το f(n) με 0.7^n, βρίσκουμε H(z)= 1/(1- 0.7z^-1) και πολλαπλασιάζοντας με z παίρνουμε την τελική μορφή της συνάρτησης μεταφοράς:

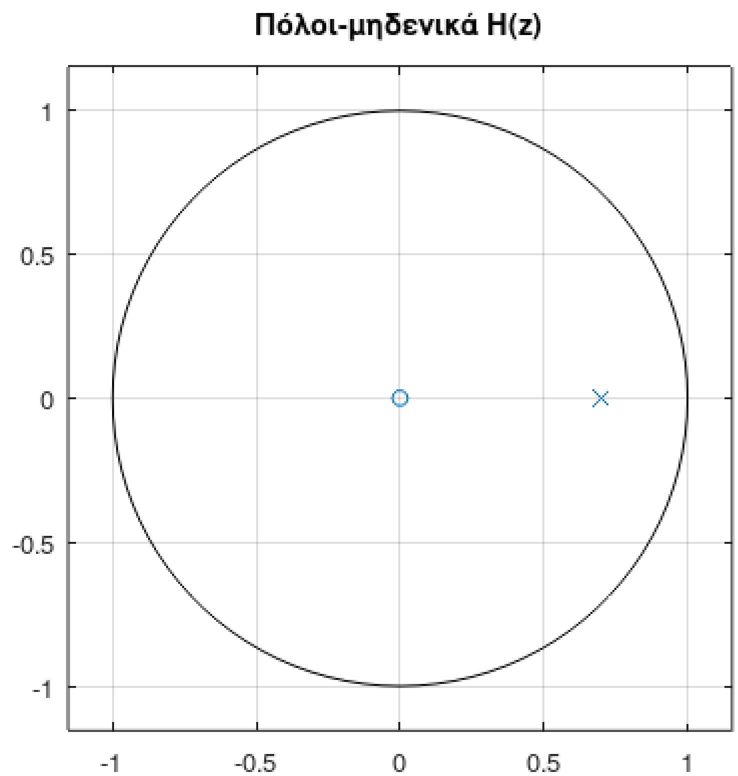
**H(z)= z/(z- 0.7)**

Τους πόλους του συστήματος και τον έλεγχο ευστάθειας τον πραγματοποιούμε μέσω της συνάρτησης zplane:

**main.m**:

```
%Κρουστική Απόκριση
%h[n]= 0.7^n * u[n]
N= 20;
n= 0:N;
h= (0.7).^n;

%Ελεγχος ευστάθειας - πόλοι
figure;
zplane(b, a);
title('Πόλοι-μηδενικά H(z)');
roots(a)
```

Για να είναι ευσταθές το σύστημα πρέπει οι πόλοι του να βρίσκονται εντός του μοναδιαίου κύκλου. Βλέπουμε από το παρακάτω διάγραμμα ότι αυτό ισχύει:




## <u>Άσκηση 21:</u>

**Μετασχηματισμός Ζ**. Έστω σήμα διακριτού χρόνου, το οποίο περιγράφεται από τη σχέση:

$$x[n] = (3 + 2n)\, u[n]$$

Να υπολογιστούν θεωρητικά : (i) ο μετασχηματισμός z του σήματος $x[n]$ (ii) οι πόλοι και τα μηδενικά του μετασχηματισμού z του σήματος. Να δημιουργηθεί κώδικας για τον υπολογισμό των (i) & (ii). Επίσης, να γίνει το διάγραμμα πόλων μηδενικών.

i) Για τον μετασχηματισμό Z θα υπολογίσουμε τον τύπο $Z\{f(n)\} = \sum_{n=0}^{\infty} f(n) z^{-n}$

Αντικαθιστώντας το f(n) με (3+2n), χωρίζοντας τον τύπο σε δύο αθροίσματα και εφαρμόζοντας τους τύπους της γεωμετρικής σειράς και της σειράς nz^-n στα αθροίσματα αντίστοιχα, καταλήγουμε στο

X(z)= 3(1/(1-z^1))+ 2(z^-1/(1- z^-1)^2))

Φέρνοντας την παραπάνω σχέση σε μορφή κοινού παρονομαστή καταλήγουμε στην τελική μορφή του μετασχηματισμού Z:

**X(z)= (3- z^-1)/((1- z^-1)^2), |z|> 1**

ii) Για τους πόλους και τα μηδενικά εργαζόμαστε ως εξής:

Στον αριθμητή του μετασχηματισμού έχουμε 3- z^-1 => z^-1= 3 => z= 1/3, άρα

**ένα μηδενικό στο z= 1/3**

Στον παρονομαστή έχουμε (1-z^1)^2= 0 => z^-1= 1 => z= 1, άρα

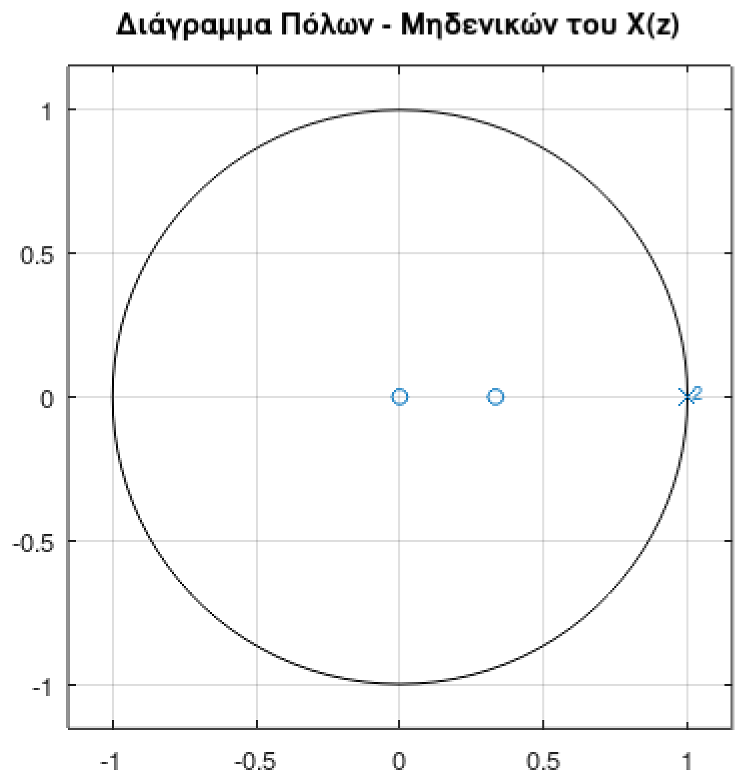
**διπλός πόλος στο z= 1**

Οι παραπάνω ενέργειες πραγματοποιούνται στον παρακάτω κώδικα:

**<u>main.m:</u>**

```
b= [3  -1];                 %αριθμητής
a= conv([1 -1],[1 -1]);   %(1 - z^-1)^2

disp('Αριθμητής X(z):');
disp(b);

disp('Παρονομαστής X(z):');
disp(a);

%Πόλοι και μηδενικά
Z= roots(b);
P= roots(a);

disp('Μηδενικά:');
disp(Z);

disp('Πόλοι:');
disp(P);

%Διάγραμμα πόλων
figure;
zplane(b, a);
title('Διάγραμμα Πόλων - Μηδενικών του X(z)');
grid on;
```

Ολοκληρώνουμε με το διάγραμμα των πόλων- μηδενικών:




Το διάγραμμα επαληθεύει την θεωρητική επίλυση.

## Άσκηση 22:

**Σχέση DTFT & DFT**. Όπως είναι γνωστό ο DFT μπορεί να θεωρηθεί ότι προέρχεται από τη δειγματολήπτιση του DTFT πάνω στις συχνότητες $\omega = 2\pi k/N$, $k = 0, \dots N-1$, δηλαδή σε N ισαπέχοντα σημεία (συχνότητες) στο διάστημα 0 έως 2π (μία περίοδος). Οσο περισσότερα δείγματα παίρνουμε τόσο πυκνώνουν οι γραμμές του διακριτού φάσματος.

Θεωρήστε το διακριτό σήμα $x[n] = n$, $0 \leq n \leq 15$ . Να σχεδιαστούν στο ίδιο σχήμα το μέτρο του DTFT για $0 \leq \omega \leq 2\pi$ και του DFT (με χρήση του αλγορίθμου fft) πάνω στις συχνότητες

$$\omega = 2\pi k/N,\ k = 0, \dots N-1 \text{ όπου N=16.}$$

Στο ίδιο σχήμα να σχεδιαστεί και το μέτρο του DFT του σήματος για N=32 σημεία.

Πρακτικά η άσκηση ζητάει:

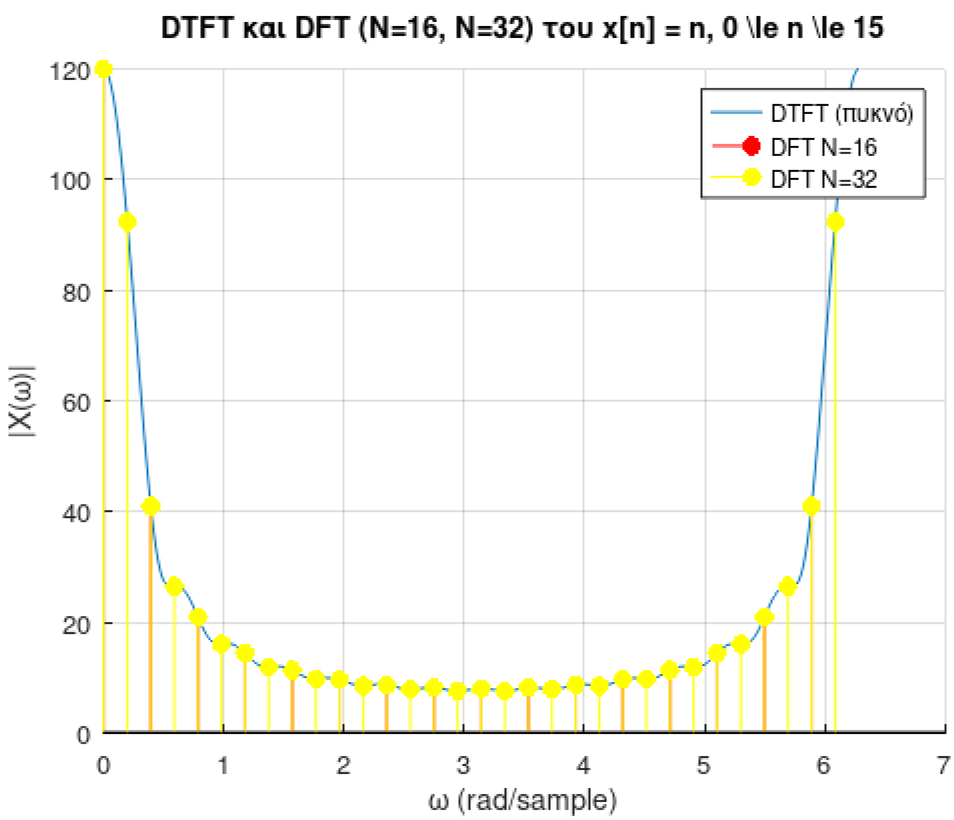
- DTFT του x[n]= n σε συνεχές πλέγμα συχνοτήτων 0<= ω<= 2π
- DTFT με N= 16 (ίσο με το μήκος του σήματος)
- DTFT με N= 32 για πιο πυκνή δειγματοληψία του DTFT
- Να εμφανιστούν όλα στο ίδιο σχήμα ως το μέτρο του DTFT (|X|)

Τα παραπάνω υλοποιούνται με τον παρακάτω κώδικα **main.m:**

```
%x[n]= n, 0<=n<=15
n= 0:15;
x= n;

%DTFT με πυκνή δειγματοληψία στο
Nw= 4000;   %πολύ πυκνό πλέγμα συχνοτήτων
w= linspace(0, 2*pi, Nw); %0<= ω<= 2π
X_dtft= zeros(size(w));

%Υπολογισμός DTFT: X(e^{jω})= Σ x[n] e^{-j ω n}
for k= 1:length(w)
    X_dtft(k)= sum(x .* exp(-1j*w(k)*n));
endfor

%DFT με N= 16
N1= 16;
X16= fft(x, N1);    % εδώ N1 = length(x), άρα χωρίς zero padding
w16= 2*pi*(0:N1-1)/N1;  % ω_k = 2πk/N1

%DFT με N= 32
N2= 32;
X32= fft(x, N2);  %zero padding, επειδή N2> length(x)
w32= 2*pi*(0:N2-1)/N2;
```

```
%Αποτελέσματα
figure;
hold on;
grid on;

%DTFT (συνεχής καμπύλη)
plot(w, abs(X_dtft));

%DFT N=16
stem(w16, abs(X16), 'r', 'filled');

%DFT N=32
stem(w32, abs(X32), 'y', 'filled');
xlabel('\omega (rad/sample)');
ylabel('|X(\omega)|');
title('DTFT και DFT (N=16, N=32) του x[n] = n, 0 \le n \le 15');
legend('DTFT (πυκνό)', 'DFT N=16', 'DFT N=32');
```

Ο οποίος μας δίνει το διάγραμμα που ακολουθεί:



Αναλύοντας το διάγραμμα, έχουμε ότι:

- Μπλε καμπύλη (DTFT): συνεχής συνάρτηση του ω= [0, 2π]
- Κόκκινα σημεία (N=16):
  - 16 σημεία πάνω στη μπλε καμπύλη
  - δείχνουν ότι ο DFT είναι δειγματοληψία της DTFT στις συχνότητες 2πk/16
- Κίτρινα σημεία (N=32):
  - 32 σημεία, διπλάσια σε πλήθος
  - ακολουθούν πιο “πυκνά” τη μπλε καμπύλη
  - δείχνουν ότι με μεγαλύτερο N παίρνουμε πιο πυκνή δειγματοληψία του DTFT

Επαληθεύουμε έτσι την πρόταση της εκφώνησης, ότι δηλαδή όσο περισσότερα δείγματα παίρνουμε τόσο πυκνώνουν οι γραμμές του διακριτού φάσματος.

## Άσκηση 23:

**Κυκλική και Γραμμική συνέλιξη.** Δίνονται οι ακολουθίες:

$$x[n] = 3\delta[n] + \delta[n-1] - \delta[n-5]$$

$$\kappa\alpha\iota\ w[n] = [1, 0,1, -2\ ],\ 0 \le n \le 3$$

Ζητούνται:

1) Να υπολογιστεί η κυκλική τους συνέλιξη α) στο πεδίο του χρόνου β) με χρήση fft(), ifft()

2) Να υπολογιστεί η κυκλική τους συνέλιξη 9 σημείων α) στο πεδίο του χρόνου β) με χρήση fft(), ifft()

3) Να βρεθεί η γραμμική συνέλιξη α) στο πεδίο του χρόνου β) με χρήση fft(), ifft()

1) Αρχικά γράφουμε και την ακολουθία x σε μορφή διανύσματος βλέπωντας που μηδενίζει η κρουστική δ[n]. Βρίσκουμε ότι:

- x[0] = 3
- x[1] = 1
- x[5] = −1
- αλλού 0

Συνεπώς το x γράφεται και ως **x = [3 1 0 0 0 -1 0 0], 0<= n<= 5.**

α) Έπειτα θα υπολογίσουμε την κυκλική συνέλιξη στο πεδίο του χρόνου μέσω του ορισμού της

$$\sum_{m=0}^{N-1} x_1(m) x_2((n-m))_N$$

Η σχέση αυτή υλοποιείται μέσα στο παρακάτω for-loop:

```
N= 6;

%x[n] και w[n] σε μορφή διανυσμάτων
x= [3 1 0 0 0 -1];
w= [1 0 1 -2 0 0];
```

```
%1)
disp('Κυκλική συνέλιξη:');
%α)στο πεδίο χρόνο
y_time= zeros(1, N);
for n= 0:N-1
  for m= 0:N-1
    k= mod(n-m, N);
    y_time(n+1)= y_time(n+1)+ x(m+1)*w(k+1);
  endfor
endfor

y_time
```

β) Ο υπολογισμός της κυκλικής συνέλιξης με χρήση ***fft*** είναι έυκολη διαδικασία, θα μετασχηματίσουμε και τα δύο διανύσματα, θα τα πολλαπλασιάσουμε και θα τα αντιστρέψουμε:

```
%β)με χρήση fft
X= fft(x, N);
W= fft(w, N);
Y= X.*W;
y_fft= round(real(ifft(Y)));

y_fft
```

Στρογγυλοποιούμε το **y_fft** διότι τα αποτελέσματα έμοιαζαν διαφορετικά λόγω floating point σφαλμάτων.

Τα αποτελέσματα και των δύο μεθόδων είναι ίσα.

```
Κυκλική συνέλιξη:
y_time =

   3   0   5  -5  -2  -1

y_fft =

   3   0   5  -5  -2  -1
```

2) Η διαδικασία υπολογισμού της κυκλικής συνέλιξης 9 σημείων είναι ακριβώς η ίδια με παραπάνω απλώς τα διανύσματα θα επεκταθούν (μέσω zero padding) στις 9 τιμές. Δηλαδή:

- N= 9;
- x= [3 1 0 0 0 -1 0 0 0];
- w= [1 0 1 -2 0 0 0 0 0];

Ακολουθεί ο κώδικας για τα ερωτήματα α και β και σύγκριση των αποτελεσμάτων τους.

```
%2)
disp('==========================');
disp('Για 9 σημεία:');
%α) στο πεδίο του χρόνου
%padding για να πάμε στα 9 σημεία
N= 9;
```

```
x= [3 1 0 0 0 -1 0 0 0];
w= [1 0 1 -2 0 0 0 0 0];

y_time= zeros(1, N);
for n= 0:N-1
  for m= 0:N-1
    k= mod(n-m, N);
    y_time(n+1)= y_time(n+1)+ x(m+1)*w(k+1);
  endfor
endfor

y_time

%β)με χρήση fft
X= fft(x, N);
W= fft(w, N);
Y= X.*W;
y_fft= round(real(ifft(Y)));

y_fft
```

Τα αποτελέσματα είναι και εδώ ίσα μεταξύ τους.

```
==========================
Για 9 σημεία:
y_time =

   3   1   3  -5  -2  -1   0  -1   2

y_fft =

   3   1   3  -5  -2  -1   0  -1   2
```

3) Τέλος υπολογίζουμε την γραμμική συνέλιξη των ακολουθιών. Εδώ τα διανύσματα δεν δέχονται κάποια επιμήκυνση (padding) αλλά υπολογίζονται στο μέγεθος L= Lx + Lw – 1, όπου Lx= length(x), Lw= length(w). Συνεπώς έχουμε:

```
%3)
disp('==========================');
disp('Γραμμική συνέλιξη:');
%α) στο πεδίο του χρόνου
x= [3 1 0 0 0 -1];
w= [1 0 1 -2];
Lx= length(x);
Lw= length(w);
L= Lx + Lw - 1;

y_time= zeros(1, L);

for n= 0:L-1
  for m= 0:Lx-1
    k= n - m;
    if k>= 0 && k< Lw
```

```
      y_time(n+1)= y_time(n+1) + x(m+1)*w(k+1);
    endif
  endfor
endfor

y_time

%β) με χρήση fft
N= 9;   %N= Lx+Lw-1

X= fft(x, N);
W= fft(w, N);

Y= X.*W;
y_fft= round(real(ifft(Y)))
```

Με αποτελέσματα τα ακόλουθα.

Και έδω βγάινουν ίσα συνεπώς είμαστε σωστοί για όλα τα ερωτήματα.

```
============================
Γραμμική συνέλιξη:
y_time =

   3   1   3  -5  -2  -1   0  -1   2

y_fft =

   3   1   3  -5  -2  -1   0  -1   2
```

Ακολουθεί ολόκληρος ο κώδικας της **main.m:**

```
N= 6;

%x[n] και w[n] σε μορφή διανυσμάτων
x= [3 1 0 0 0 -1];
w= [1 0 1 -2 0 0];

%1)
disp('Κυκλική συνέλιξη:');
%α)στο πεδίο χρόνο
y_time= zeros(1, N);
for n= 0:N-1
  for m= 0:N-1
    k= mod(n-m, N);
    y_time(n+1)= y_time(n+1)+ x(m+1)*w(k+1);
  endfor
endfor

y_time

%β)με χρήση fft
X= fft(x, N);
W= fft(w, N);
Y= X.*W;
```

```
y_fft= round(real(ifft(Y)));

y_fft

%2)
disp('===========================');
disp('Για 9 σημεία:');
%α) στο πεδίο του χρόνου
%padding για να πάμε στα 9 σημεία
N= 9;
x= [3 1 0 0 0 -1 0 0 0];
w= [1 0 1 -2 0 0 0 0 0];

y_time= zeros(1, N);
for n= 0:N-1
  for m= 0:N-1
    k= mod(n-m, N);
    y_time(n+1)= y_time(n+1)+ x(m+1)*w(k+1);
  endfor
endfor

y_time

%β)με χρήση fft
X= fft(x, N);
W= fft(w, N);
Y= X.*W;
y_fft= round(real(ifft(Y)));

y_fft

%3)
disp('===========================');
disp('Γραμμική συνέλιξη:');
%α) στο πεδίο του χρόνου
x= [3 1 0 0 0 -1];
w= [1 0 1 -2];
Lx= length(x);
Lw= length(w);
L= Lx + Lw - 1;

y_time= zeros(1, L);

for n= 0:L-1
  for m= 0:Lx-1
    k= n - m;
    if k>= 0 && k< Lw
      y_time(n+1)= y_time(n+1) + x(m+1)*w(k+1);
    endif
  endfor
endfor
```

```
y_time

%β) με χρήση fft
N= 9;   %N= Lx+Lw-1

X= fft(x, N);
W= fft(w, N);

Y= X.*W;
y_fft= round(real(ifft(Y)))
```

```
>> main

Κυκλική συνέλιξη:
y_time =

   3   0   5  -5  -2  -1

y_fft =

   3   0   5  -5  -2  -1

============================
Για 9 σημεία:
y_time =

   3   1   3  -5  -2  -1   0  -1   2

y_fft =

   3   1   3  -5  -2  -1   0  -1   2

============================
Γραμμική συνέλιξη:
y_time =

   3   1   3  -5  -2  -1   0  -1   2

y_fft =

   3   1   3  -5  -2  -1   0  -1   2

>> 
```

## **Άσκηση 24:**

Εκτίμηση περιόδου από αυτοσυσχέτιση. Ας θεωρηθούν τα σήματα:

$$x[n] = 3\cos(\frac{\pi n}{4} + \frac{\pi}{12})$$

$$y[n] = 2cos^2(\frac{\pi}{3}n)$$

Να βρεθεί η ακολουθία αυτοσυσχέτισης του κάθε σήματος με τη συνάρτηση xcorr. Να γίνει η γραφική παράσταση του κάθε σήματος καθώς και της ακολουθία αυτοσυσχέτισής του. Στη συνέχεια, με χρήση της xcorr να γίνει εκτίμηση της περιόδου του κάθε σήματος.

<u>Για το x[n]:</u>

Η συχνότητα είναι $\omega_0$= π/4

Για διακριτό χρόνο, $\omega_0 N_0$= 2πk ⇒ (4/π)$N_0$= 2πk ⇒ $N_0$=8k άρα

ελάχιστη περίοδος $N_0$= 8.

<u>Για το y[n]:</u>

$\cos^2(\theta)$= (1+ cos(2θ))/ 2 ⇒ y[n]= 1+cos(2πn/3) άρα

συχνότητα συνημιτόνου $\omega_0$= 2π/3

Ελάχιστη περίοδος $\omega_0 N_0$= 2πk ⇒ 32π$N_0$= 2πk ⇒ $N_0$= 3k ⇒ $N_0$= 3

Έστω διάστημα n= [0, 63] για 64 δείγματα, ο κώδικας μας σε matlab με συσχέτιση μέσω xcorr και εκτίμηση περιόδου γίνεται:

**main.m:**

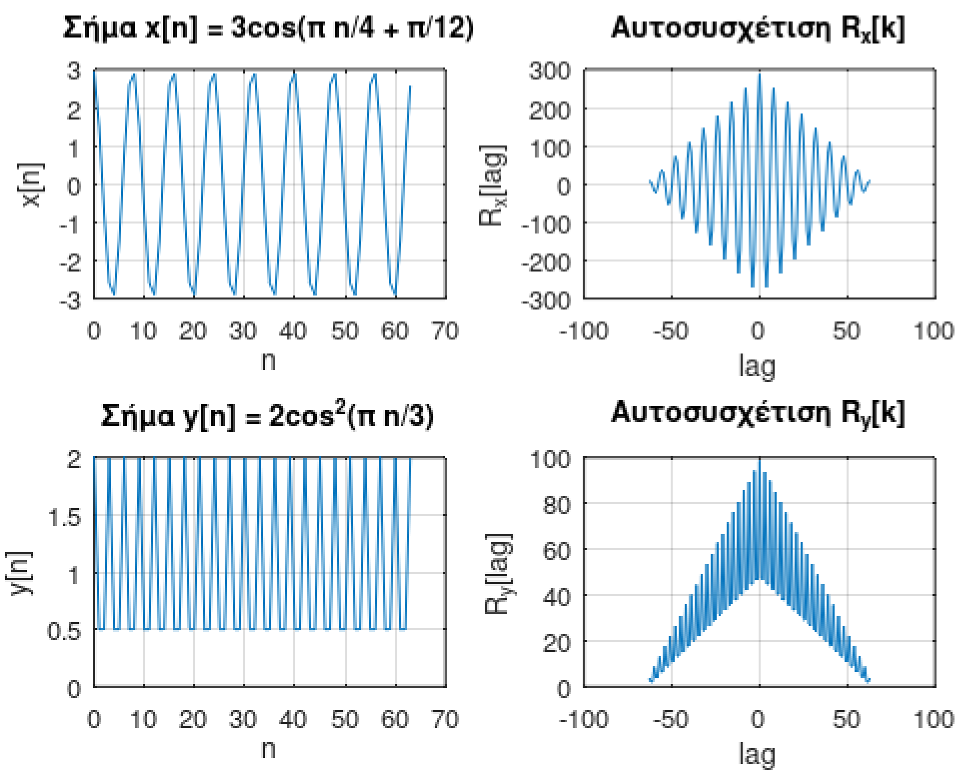
```
pkg load signal;
%Διάστημα δειγμάτων
N= 64;
n= 0:N-1;

%Σήματα
x = 3*cos(pi*n/4 + pi/12); % x[n]
y = 2*(cos(pi*n/3).^2);   % y[n]

%Αυτοσυσχέτιση με xcorr
[Rx, lags_x]= xcorr(x);
[Ry, lags_y]= xcorr(y);

% Εκτίμηση περιόδου από αυτοσυσχέτιση
%Για x[n]
[~, idx0_x]= max(Rx);
segment_x= Rx(idx0_x+1:end); %θέλουμε θετική μέγιστη μετατόπιση
[~, idx1_x]= max(segment_x);

%Για y[n]
[~, idx0_y]= max(Ry);
segment_y= Ry(idx0_y+1:end);
[~, idx1_y]= max(segment_y);

fprintf('Εκτιμώμενη περίοδος x[n]: %d δειγμάτα\n', idx1_x);
fprintf('Εκτιμώμενη περίοδος y[n]: %d δειγμάτα\n', idx1_y);

%Αποτελέσματα
figure;
subplot(2,2,1);
plot(n, x); grid on;
xlabel('n'); ylabel('x[n]');
title('Σήμα x[n] = 3cos(\pi n/4 + \pi/12)');

subplot(2,2,2);
plot(lags_x, Rx); grid on;
xlabel('lag'); ylabel('R_x[lag]');
title('Αυτοσυσχέτιση R_x[k]');

subplot(2,2,3);
plot(n, y); grid on;
xlabel('n'); ylabel('y[n]');
title('Σήμα y[n] = 2cos^2(\pi n/3)');

subplot(2,2,4);
plot(lags_y, Ry); grid on;
xlabel('lag'); ylabel('R_y[lag]');
title('Αυτοσυσχέτιση R_y[k]');
```

Στον παραπάνω κώδικα:

- idx0_x,y → η θέση της πρώτης μέγιστης συσχέτισης στον πίνακα των συσχετίσεων R
- Rx(idx0_x+1:end) → φτιάχνει νέο διάνυσμα  με αρχή αμέσως μετά το πρώτο μέγιστο
- idx1_x,y → η θέση της δεύτερης μέγιστης συσχέτισης από το νέο διάνυσμα

Στην εκτύπωση των αποτελεσμάτων βλέπουμε ότι για το x[n] έχουμε 8 δείγματα και για το y[n] 3 δείγματα. Τα αποτελέσματα αυτά ταιριάζουν με τις ελάχιστες περιόδους που βρήκαμε παραπάνω στην θεωρητική επίλυση.

```
>> main

Εκτιμώμενη περίοδος x[n]: 8 δειγμάτα
Εκτιμώμενη περίοδος y[n]: 3 δειγμάτα
>> |
```




## Άσκηση 25:

**Ετεροσυσχέτιση.** Να δημιουργηθεί το σήμα:

$$x_1[n] = \begin{cases} n, & 0 \leq n \leq 30 \\ 0, & 31 \leq n \leq 100 \end{cases}$$

Να δημιουργηθεί το σήμα $x2[n]$ από την κυκλική ολίσθηση του $x1[n]$ κατά 25 δείγματα (χρήση της circshift). Να βρεθεί η συσχέτιση των δύο σημάτων με τη συνάρτηση xcorr. Να γίνει η γραφική παράσταση του κάθε σήματος καθώς και της ακολουθία ετεροσυσχέτισής τους. Σε ποιο σημείο μεγιστοποιείται η ετεροσυσχέτιση;

Από την εκφώνηση διαπιστώνουμε ότι το x1[n] είναι σήμα 101 δειγμάτων (n= 0:100). Δημιουργούμε το $x_2[n]$ με circshift για 25 δείγματα: $x_2[n]= x_1[n-25]$. Υπολογίζουμε την ετεροσυσχέτιση μέσω της ***xcorr*** σύμφωνα με την σχέση $R_{x1x2}[k]= \sum x_1[n]x_2[n-k]$, όπου R οι τιμές της συσχέτισης για κάθε μετατόπιση και lags το αντίστοιχο διάνυσμα των μετατοπίσεων. Μας δείχνει δηλαδή για ποια μετατόπιση τα δύο σήματα ευθυγραμμίζονται.

Έπειτα, με την max(R) βρίσκουμε την μέγιστη τιμή της συσχέτισης και η idx_max κρατάει την θέση στον πίνακα R που αυτή βρέθηκε. Τέλος, από το διάνυσμα των μετατοπίσεων lags παίρνουμε την τιμή της μετατόπισης που αντιστοιχεί στο idx_max.

Η παραπάνω θεωρητική επίλυση πραγματοποιείται στο Octave με τον παρακάτω κώδικα:

**main.m:**

```
%x1[n]
n= 0:100; % συνολικό εύρος
x1= zeros(size(n));
x1(1:31)= 0:30;  % 0 ≤ n ≤ 30

%Κυκλική ολίσθηση κατά 25 δείγματα
x2= circshift(x1, 25);

%Ετεροσυσχέτιση
[R, lags]= xcorr(x1, x2);

%Θέση μέγιστου
[~, idx_max]= max(R);
lag_max= lags(idx_max);

fprintf('Η ετεροσυσχέτιση μεγ. για lag= %d\n', lag_max);

%Αποτελέσματα
figure;
subplot(3,1,1);
plot(n, x1);
title('Σήμα x_1[n]');
xlabel('n'); grid on;
```

```
subplot(3,1,2);
plot(n, x2);
title('Σήμα x_2[n] (25 δείγματα)');
xlabel('n'); grid on;

subplot(3,1,3);
plot(lags, R);
title('Ετεροσυσχέτιση R_{x_1x_2}[k]');
xlabel('lag'); ylabel('R'); grid on;
```

```
>> main

Η ετεροσυσχέτιση μεγ. για lag= -25
```

Πράγματι η ετεροσυσχέτιση μεγιστοποιείται στο -25 και σύμφωνα με τον κώδικα. Συνεπώς επιβεβαιώνεται η θεωρητική επίλυση. Ακολουθούν οι γραφικές παραστάσεις.