Πανεπιστήμιο Δυτικής Αττικής
Σχολή Μηχανικών
Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

ΕΡΓΑΣΤΗΡΙΟ ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΣΗΜΑΤΟΣ

ΦΙΛΙΠΠΟΣ ΠΑΠΑΓΕΩΡΓΙΟΥ - 21390174

ΗΜΕΡΟΜΗΝΙΑ ΠΑΡΑΔΟΣΗΣ:

Τετάρτη 31 Ιανουαρίου 2024 - 11:59 μ.μ.

ΟΜΑΔΑ ΕΡΓΑΣΤΗΡΙΟΥ:

Τμήμα 2 Παρασκεύη 11:00-13:00

Υπευθύνος Ομάδας:

ΜΠΡΑΤΣΟΛΗΣ ΕΜΜΑΝΟΥΗΛ

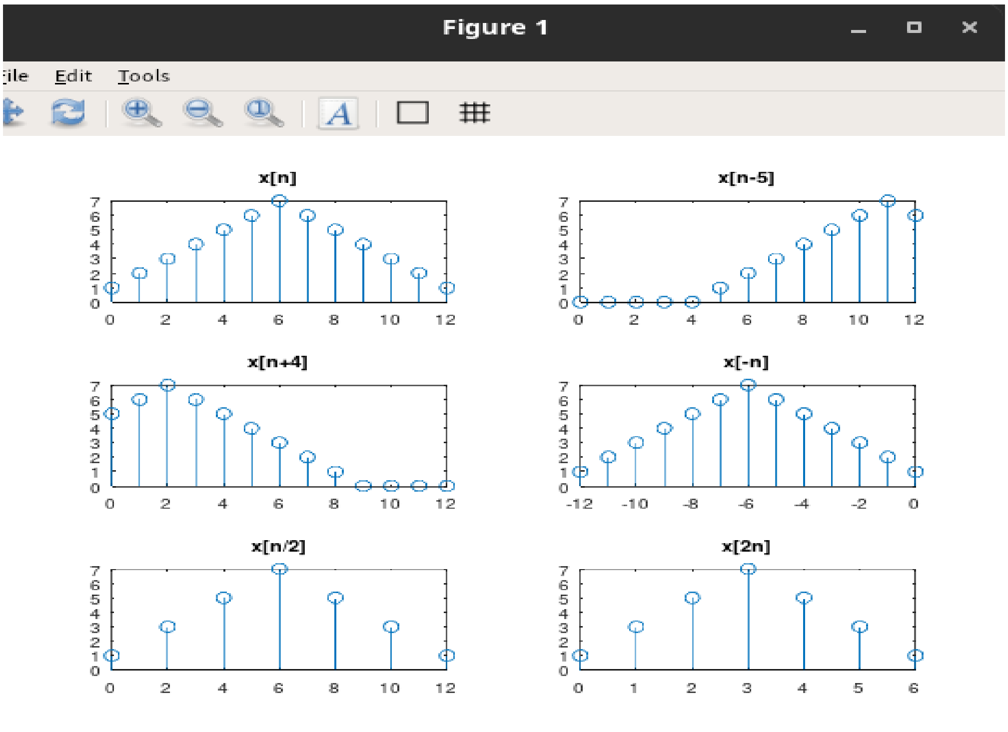
ΤΙ ΘΑ ΔΟΥΜΕ ΣΤΗΝ ΕΡΓΑΣΙΑ

Η ασκήσεις της εργασίας που μας εχετέ αναθέσει καλυπτούν ένα ευρύ φάσμα θεματών που σχετιζονταί με την επεξεργασία σηματός.Δηλαδή συμπεριλαμβάνονται η δημιουργία και ο χειρισμός σηματών διακριτού χρόνου,της ανάλυσης απόκρισης συστηματός, της συνέλιξης και της ανάλυσης απόκρισης συχνότητας.Επισής τα εργαλεία που θα χρησιμοποιησούμε είναι το MatLab ή Octave στην δική μου περιπτώση θα χρησιμοποιήσω το octave. Συνοπτικά οι ασκήσεις θα περιεχούν ακομή βασικές αρχές της ψηφιακής επεξεργασίας σηματός οπώς γραμμικότητα , η συνέλιξη και ο μετασχηματιμσός Fourier. Τέλος στης ασκήσης μου θα αναλύω ποίο πολύ την σκέψη μου για την ασκήση πάρα τον κωδικά καθώς θα είναι πλήρης σχολιασμένος.

1 . Να δημιουργηθεί το σήμα διακριτού χρόνου □[□] = [1 2 □ ↑ 4 5 6 7 6 5 4 3 2 1] Σε έξι υποπαράθυρα του ίδιου γραφικού παράθυρου να δημιουργηθούν οι γραφικές παραστάσεις των : □) □[□], □) □[□ − 5], □) □[□ + 4], □) □[−□], □) □[□/2], □) □[2□].

exercise1.m

```
% Ορίστε το σήμα x[n]
x = [1 2 3 4 5 6 7 6 5 4 3 2 1];

% Ορίστε το εύρος του n
n = 0:length(x)-1;


subplot(3,2,1);
stem(n, x);
title('x[n]');

subplot(3,2,2);
stem(n, [zeros(1,5), x(1:end-5)]);
title('x[n-5]');

subplot(3,2,3);
stem(n, [x(5:end), zeros(1,4)]);
title('x[n+4]');

subplot(3,2,4);
stem(-n, x);
title('x[-n]');

subplot(3,2,5);
n_half = 0:2:length(x)-1;
stem(n_half, x(1:2:end));
title('x[n/2]');


subplot(3,2,6);
n_double = 0:(length(x)-1)/2;
stem(n_double, x(1:2:end));
title('x[2n]');
```



Επεξεργασία σκέψης άσκησης:

Ορίζω πρώτα το σήμα x[n]. Το εύρος του n ορίζεται σύμφωνα με το μήκος του x. Το subplot χρησιμοποιείται για τη δημιουργία έξι υποπαραθύρων σε ένα ενιαίο γραφικό παράθυρο. Κάθε συνάρτηση απεικονίζει έναν διαφορετικό μετασχηματισμό του x[n] σε ένα υποπαράθυρο. Οι μετασχηματισμοί περιλαμβάνουν χρονική μετατόπιση, χρονική αντιστροφή και χρονική κλιμάκωση του αρχικού σήματος. Αυτός ο κώδικας θα δημιουργήσει ένα γραφικό παράθυρο με έξι υποπαράθυρα, καθένα από τα οποία θα απεικονίζει έναν διαφορετικό μετασχηματισμό του σήματος x[n].

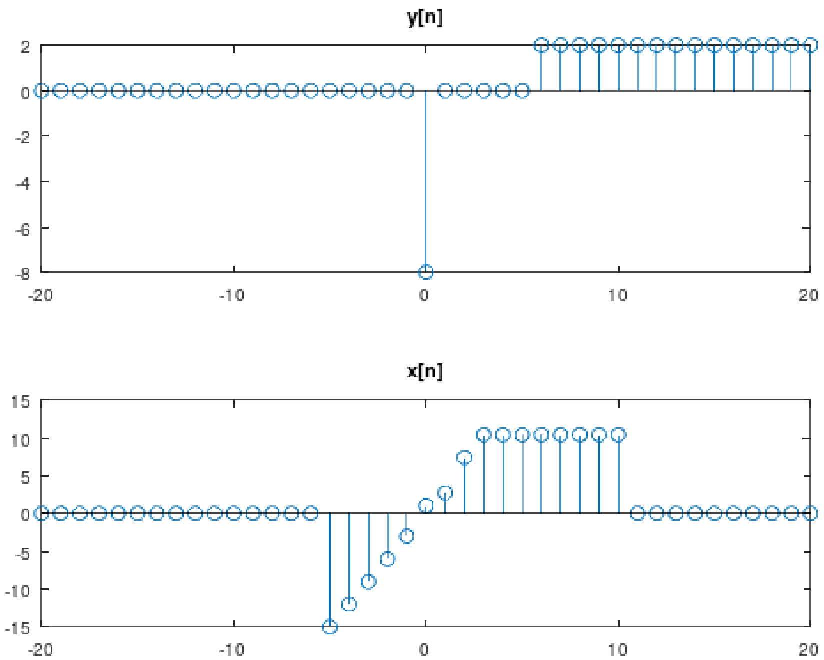
2. Να σχεδιαστούν τα σήματα διακριτού χρόνου α) □[□] = 2 * □[□ − (2 + □□□ □□□ 5)] − 8 * □[□ − □□□ □□□ 4], για □ = −20 − (□□□ □□□ 2): 20 + (□□□ □□□ 2) β) □[□] = { 3 * □, − 5 ≤ □ ≤ −1 □ □ , − 1 < □ ≤ 2 √4 * □□□(□□), 2 < □ ≤ 10 Όπου AM ο αριθμός μητρώου σας, □□□(□□) το μονοψήφιο άθροισμα των ψηφίων του, AML το τελευταίο ψηφίο του AM σας, και u[n] και δ[n] η μοναδιαία βηματική και η μοναδιαία κρουστική ακολουθία αντίστοιχα.

Επεξεργασία σκέψης άσκησης:

Αυτός ο κώδικας υπολογίζει τα y[n] και x[n] με βάση τους δεδομένους τύπους. Το u[n] είναι μια συνάρτηση για τη μοναδιαία βηματική ακολουθία και το \delta[n] για τη μοναδιαία παλμική ακολουθία. Το εύρος του n ορίζεται σύμφωνα με τις δεδομένες συνθήκες. Το x[n] υπολογίζεται με τη χρήση arrayfun για την εφαρμογή της συνάρτησης κατά τεμάχια σε όλο το εύρος του n.

Τέλος, τα σήματα απεικονίζονται με τη χρήση stem σε διάφορα υποδιαγράμματα. Αυτός ο κώδικας θα δημιουργήσει μια γραφική αναπαράσταση τόσο του y[n] όσο και του x[n], επιτρέποντάς σας να οπτικοποιήσετε αυτά τα σήματα.

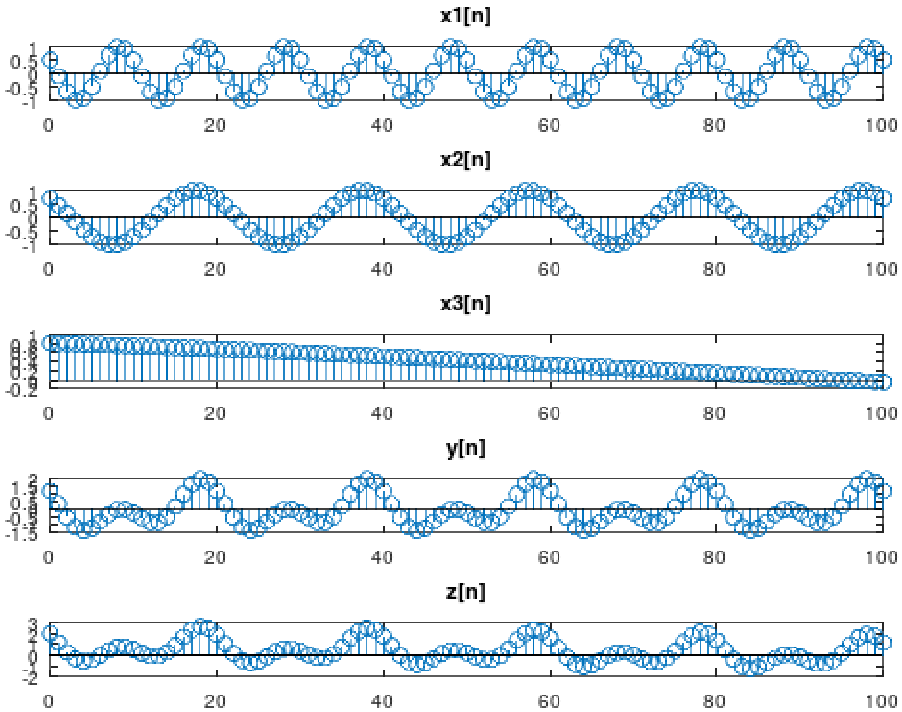
```
% Ορισμός της AM και των σχετικών υπολογισμών
AM = 21390174;
sum_AM = sum(int2str(AM)-'0'); % Άθροισμα των ψηφίων του AM
AML = mod(AM, 10); % Τελευταίο ψηφίο AM

% Ορίστε το εύρος του n
n = -20 - mod(AML, 2) : 20 + mod(AML, 2);

% Ορίστε u[n] και δ[n]
u = @(n) n >= 0;
delta = @(n) n == 0;

% Ορισμός y[n]
y = 2 * u(n - (2 + mod(AML, 5))) - 8 * delta(n - mod(AML, 4));

% Ορισμός x[n]
x = arrayfun(@(n) (n >= -5 && n <= -1) * (3 * n) + ...
                  (n > -1 && n <= 2) * exp(n) + ...
                  (n > 2 && n <= 10) * sqrt(4 * sum_AM), n);

%Σχεδιάστε τα y[n] και x[n]
subplot(2,1,1);
stem(n, y);
title('y[n]');

subplot(2,1,2);
stem(n, x);
title('x[n]');
```



3. Να σχεδιαστούν τα παρακάτω σήματα και για όσα είναι περιοδικά να προσδιοριστεί η θεμελιώδης περίοδός τους

και να γίνει γραφική επαλήθευση της περιοδικότητάς τους. □1[□] = □□□(0.2□ □ + □/3) □2[□] = □□□(0.1□ □ + □/4) □3[□] = □□□(0.01 □ + □/5) □[□] = □1 [□] + □2[□] □[□] = □1 [□] + □2 [□] + □3[□]

```
% Ορίστε το εύρος του n
n = 0:100;  % Ρυθμίστε το εύρος ανάλογα με τις ανάγκες

% Καθορίστε τα σήματα
x1 = cos(0.2 * pi * n + pi / 3);
x2 = cos(0.1 * pi * n + pi / 4);
x3 = cos(0.01 * n + pi / 5);
y = x1 + x2;
z = x1 + x2 + x3;

% Σχεδιάστε τα σήματα
subplot(5,1,1);
stem(n, x1);
title('x1[n]');

subplot(5,1,2);
stem(n, x2);
title('x2[n]');

subplot(5,1,3);
stem(n, x3);
title('x3[n]');

subplot(5,1,4);
stem(n, y);
title('y[n]');

subplot(5,1,5);
stem(n, z);
title('z[n]');
```



Επεξεργασία σκέψης άσκησης:

Προσδιορισμός της θεμελιώδους περιόδου: Για ένα διακριτού χρόνου συνημιτονικό σήμα cos(ωn + φ)cos(ωn+φ), η θεμελιώδης περίοδος N μπορεί να βρεθεί από τη σχέση: N =2π /|ω|, με την προϋπόθεση ότι η N είναι ακέραιος αριθμός.

Για x1[n], ω=0.2π οπότε η θεμελιώδης περίοδος είναι N = 2π\0.2π = 10N=

Για x2[n], ω = 0.1π, οπότε η θεμελιώδης περίοδος είναι N = 2π/0.1π = 20N

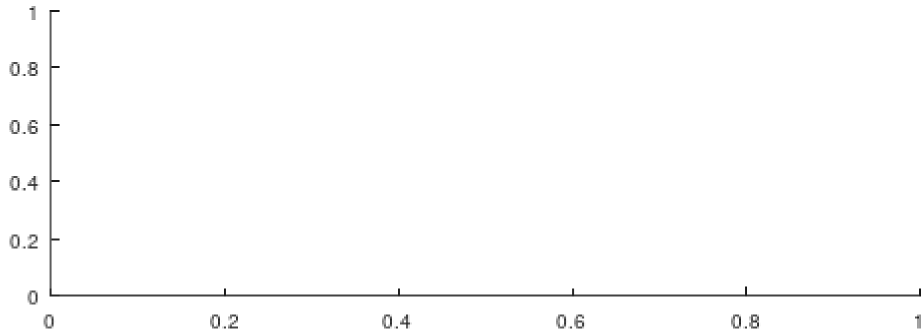
Το x3[n] δεν έχει ακέραια θεμελιώδη περίοδο, αφού ω = 0,01 οδηγεί σε μη ακέραια περίοδο.

Τα y[n] και z[n] είναι αθροίσματα περιοδικών σημάτων με διαφορετικές περιόδους, οπότε η περιοδικότητά τους θα εξαρτηθεί από τα κοινά πολλαπλάσια των επιμέρους περιόδων.

4. Να γραφεί συνάρτηση (function) που να δέχεται σαν όρισμα σήμα διακριτού χρόνου και το χρονικό διάστημα στο οποίο ορίζεται και να επιστρέφει το άρτιο και περιττό μέρος του σήματος, καθώς και το χρονικό άξονα στον οποίο ορίζονται. Τέλος η συνάρτηση να φτιάχνει τη γραφική παράσταση των δύο σημάτων εξόδου, στο σωστό χρονικό άξονα. Π.χ. [xe,xo,m]=ev od(x,n). Στη συνέχεια με χρήση της συνάρτησης, να υπολογιστεί το άρτιο και περιττό μέρος της ακολουθίας του σήματος x[n] του ερωτήματος 2β. Να γίνουν οι γραφικές παραστάσεις.

```
function [xe, xo, m] = ev_od(x, n)

% Ελέγξτε ότι τα x και n έχουν το ίδιο μήκος
if length(x) ~= length(n)
    error('The length of x must be the same as the length of n.');
end

% Δημιουργήστε έναν άξονα χρόνου συμμετρικό ως προς το μηδέν
% Εξασφαλίστε ότι ο άξονας χρόνου καλύπτει τόσο το -n όσο και το n
m = -fliplr(n);
m = unique([m, n]);

% Υπολογισμός ζυγών και περιττών μερών
xe = 0.5 * (x + fliplr(x));
xo = 0.5 * (x - fliplr(x));

% Σχεδιάστε το ζυγό μέρος
figure; % Άνοιγμα νέου παραθύρου σχήματος
subplot(2,1,1);
stem(m, xe);
title('Even Part');

% Σχεδιάστε το περίεργο μέρος
subplot(2,1,2);
stem(m, xo);
title('Odd Part');
end
```

```
AM = 21390174;
sum_AM = sum(int2str(AM)-'0');
n = -5:10;


x = arrayfun(@(n) (n >= -5 && n <= -1) * (3 * n) + ...
                  (n > -1 && n <= 2) * exp(n) + ...
                  (n > 2 && n <= 10) * sqrt(4 * sum_AM), n);


[xe, xo, m] = ev_od(x, n);
```

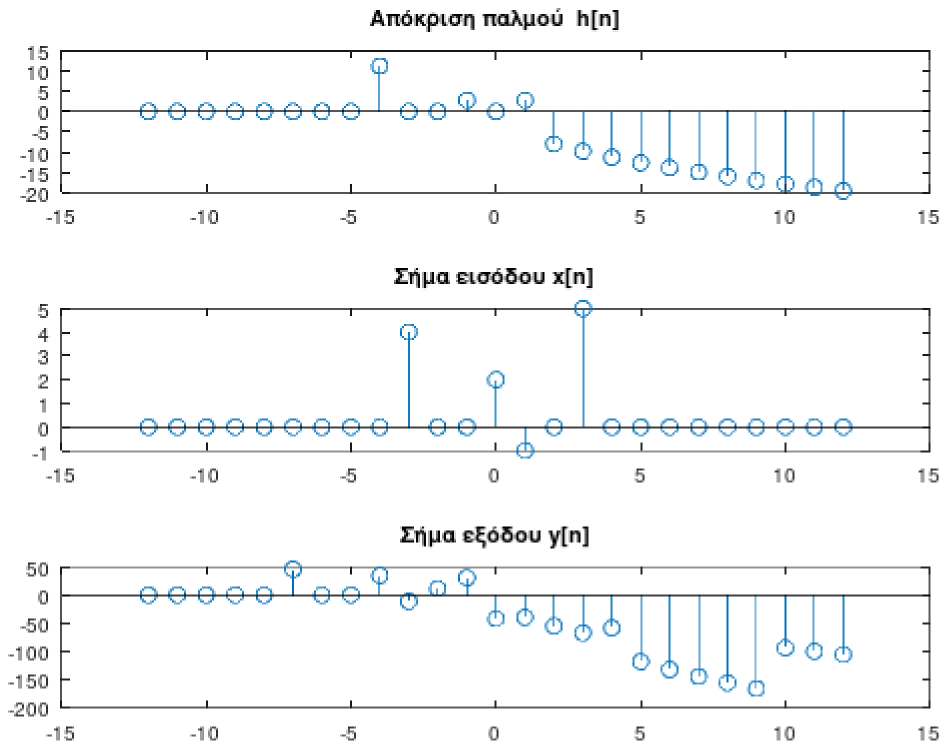


Επεξεργασία σκέψης ασκήσης:

Η συνάρτηση που κατασεκύασα θα επιστρέψει το ζυγό μέρος xe , το περιτό μέρος xo ακι τον χρονικό άξονα m στον οποίον έχουν οριστεί. Θα σχεδιάσει άκομα το αρτίο και το περιτό μέρος του σήματος.Η συνάρτηση θα δεχτεί ως όρισμα ένα σήμα και το χρονικό του διάστημα. Θα δημιουργήσει ενά συμμετρικό άξονα που καλύπτει τόσο τους θετικούς όσο και τους αρνητικούς χρόνους. *(Αυτή ήταν η σκέψη της ασκήσης άλλα δεν μου εμφανίζει ο κώδικας το αποτελεσμά που θέλω).

5. Θεωρήστε σύστημα διακριτού χρόνου, γραμμικό χρονικά αναλλοίωτο, το οποίο χαρακτηρίζεται από κρουστική απόκριση : h[□] = √8|□|(2□[□ + 4] + (□[□ + 1] − 3□[□ − 2])). Να υπολογιστεί η έξοδος του συστήματος όταν η είσοδος είναι : □[□] = 4□[□ + 3] + 2□[□] − □[□ − 1] + 5□[□ − 3], θέτωντας −12 ≤ □ ≤ 12. Να γίνει η γραφική παράσταση των σημάτων της κρουστικής απόκρισης, της εισόδου και της εξόδου του συστήματος.

```
% Ορίστε το εύρος του n
n = -12:12;

% Ορίστε την κρουστική απόκριση h[n]
delta = @(n) n == 0; % Συνάρτηση παλμού
u = @(n) n >= 0; % Συνάρτηση βήματος
h = sqrt(8*abs(n)) .* (2*delta(n + 4) + (u(n + 1) - 3*u(n - 2))),

% Ορισμός της εισόδου x[n]
x = 4*delta(n + 3) + 2*delta(n) - delta(n - 1) + 5*delta(n - 3),

% Υπολογισμός της εξόδου y[n] με τη συνέλιξη του x[n] με το h[n]
y = conv(x, h, 'same'),

% Σχεδιάστε την κρουστική απόκριση h[n]
subplot(3,1,1),
stem(n, h),
title('Απόκριση παλμού  h[n]'),

% Απεικόνιση του σήματος εισόδου x[n]
subplot(3,1,2),
stem(n, x),
title('Σήμα εισόδου x[n]'),

% Απεικόνιση του σήματος εξόδου y[n]
subplot(3,1,3),
stem(n, y),
title('Σήμα εξόδου y[n]'),
```



Επεξεργασία σκέψης άσκησης:

Για να υπολογίσουμε την έξοδο ενός γραμμικά χρονικά αναλλοίωτου συστήματος με δεδομένη την κρουστική απόκριση h[n] και την είσοδο x[n] , θα χρησιμοποίησω την πράξη της συνέλιξης , ορίζω πρώτα τα h[n] και x[n] στο εύρος n άπο -12 - 12 και στην συνέχεια εκτελούμε τη συνέλιξη για να βρούμε την έξοδο y[n].

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

```
function y = conv_toeplitz(x, h)
    % x είναι η πρώτη ακολουθία (σήμα εισόδου)
    % h είναι η δεύτερη ακολουθία (κρουστική απόκριση)

    % Δημιουργία του πίνακα Toeplitz για την κρουστική απόκριση
    h_pad = [h, zeros(1, length(x) - 1)] % Μηδενική συμπλήρωση
    T = toeplitz(h_pad, zeros(size(h_pad))),

    % Αναδιαμόρφωση του x σε διάνυσμα στήλης
    x_col = x(:),

    % Εκτέλεση της συνέλιξης ως γινόμενο πίνακα-διανύσματος
    y = T * x_col,

    % Κοπή της εξόδου στο σωστό μήκος (length(x) + length(h) - 1)
    y = y(1:length(x) + length(h) - 1),
end
```

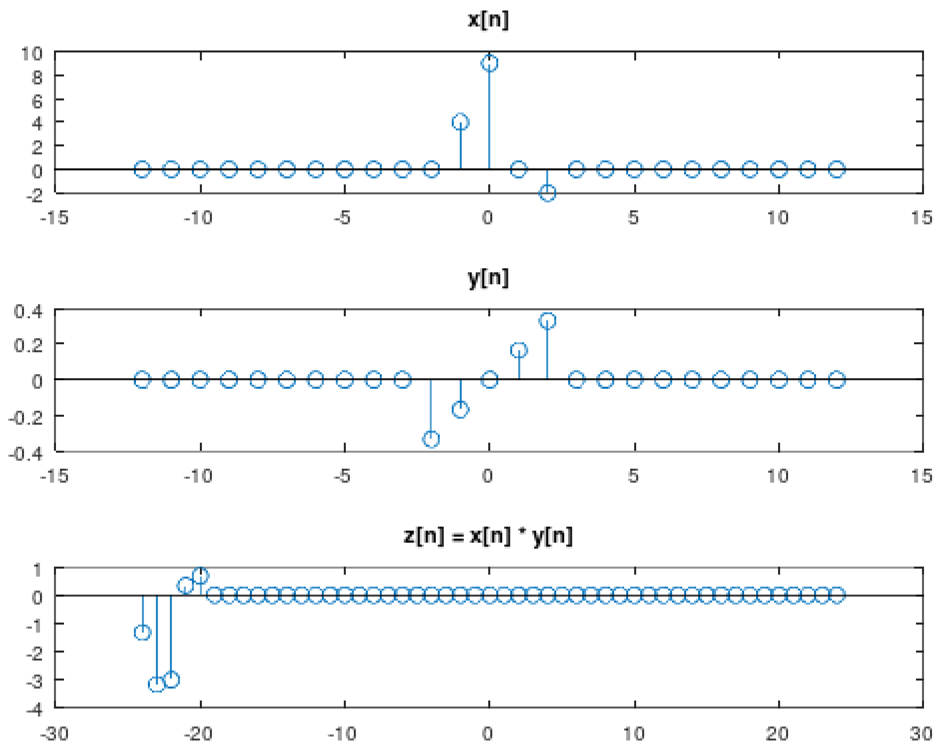
Επεξεργασία σκέψης άσκησης:

Για να δημιουργησούμε ένα διανύσμα ίδιου μήκους με το χ. Κατασκεύαζω έναν πίνακα Toeplitz από την παλμική απόκριση με μηδενική συμπληρώση. Αναδιαμορφώνω την ακολουθία εισόδου x σε διάνυσμα στήλης εάν δεν είναι ήδη. Πολλαπλασιάζω τον πίνακα Toeplitz με το διάνυσμα της ακολουθίας εισόδου για να εκτελέσει τη συνέλιξη. Κόβω το διάνυσμα y που προκύπτει στο αναμενόμενο μήκος της εξόδου της συνέλιξης, το οποίο είναι το μήκος του x συν το μήκος του h μείον ένα.

7. Με χρήση της συνάρτησης του ερωτήματος 6 να υπολογιστεί η συνέλιξη των ακολουθιών z[n]=x[n]* y[n]

όπου: □[□] = 4□[□ + 1] + 9□[□] − 2□[□ − 2] □[□] = (□ 6) (□(□ + 2) − □(□ − 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().

```
% Ορισμός της συνάρτησης δέλτα
delta = @(n) n == 0;

% Ορισμός της συνάρτησης μοναδιαίου βήματος
u = @(n) n >= 0;

% Ορισμός του εύρους του n
n = -12:12; % Τροποποίηση ανάλογα με τις ανάγκες

% Ορισμός των ακολουθιών x[n] και y[n]
x_n = 4 * delta(n + 1) + 9 * delta(n) - 2 * delta(n - 2);
y_n = (n / 6) .* (u(n + 2) - u(n - 3));

% Υπολογισμός της συνέλιξης με τη χρήση της προσαρμοσμένης συνάρτησης
z_n = conv_toeplitz(x_n, y_n);

% Σχεδιάστε τα x[n], y[n] και z[n].
subplot(3,1,1);
stem(n, x_n);
title('x[n]');

subplot(3,1,2);
stem(n, y_n);
title('y[n]');

subplot(3,1,3);
% Εφόσον η συνέλιξη επεκτείνει το εύρος, προσαρμόστε το n για το z[n]
n_z = -24:24; % Αυτό μπορεί να χρειαστεί να προσαρμοστεί
stem(n_z, z_n);
title('z[n] = x[n] * y[n]');
```

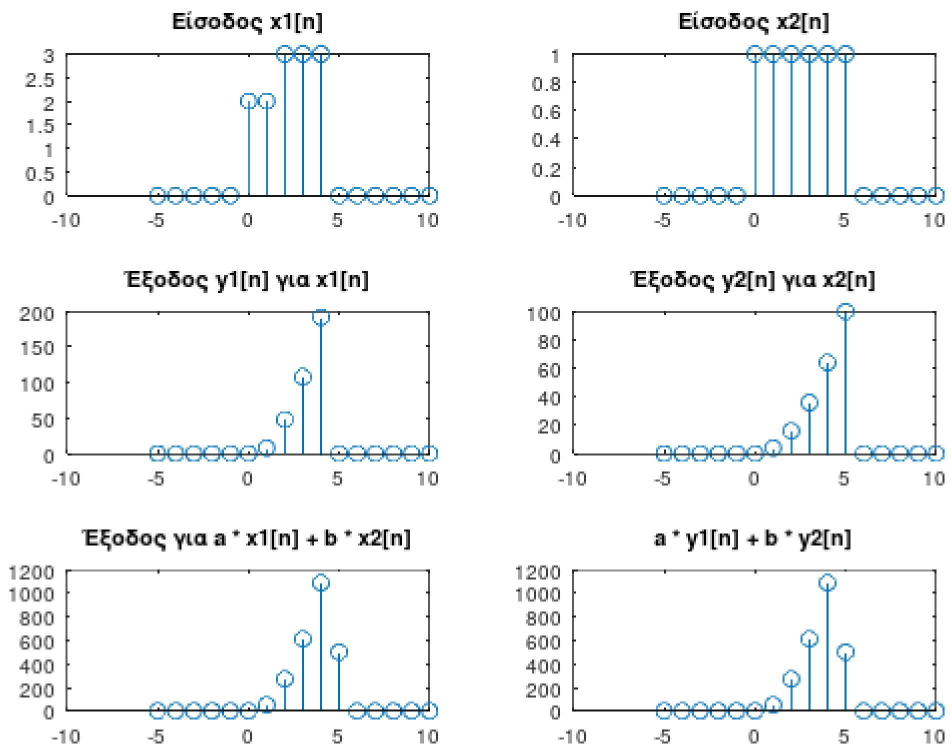


Επεξεργασία σκέψης ασκήσης:

Για να υπολογίσουμε τη συνέλιξη των ακολουθιών z[n] = x[n]* y[n] χρησιμοποίωντας τη συνάρτηση που φτιαξάμε πριν. Θα ορίσουμε της ακολουθίες x[n] και y[n] συμφώνα με τους ορισμούς καλούμε μέτα την προσαρμοσμένη σύναρτηση της συνέλιξης για να λαβέτε z[n].

8. Ας θεωρηθεί το σύστημα που περιγράφεται από τη σχέση εισόδου εξόδου: □[□] = 4□ 2 □[□] Να διερευνηθεί με γραφικό τρόπο εάν το σύστημα είναι γραμμικό. Η διερεύνηση να γίνει στο διάστημα n = [-5 :10] ενώ οι βοηθητικές είσοδοι είναι : x1[n]=[2 2 3 3 4], 0 ≤ n ≤ 4, x2[n]=u[n]-u[n-6] και οι σταθερές : α=4 και β=5. Σε υποπαράθυρα του ίδιου γραφικού παραθύρου να σχεδιαστούν οι είσοδοι x1, x2 και οι δύο έξοδοι που σχετίζονται με τη διερεύνηση της γραμμικότητας

```
% Ορίστε το εύρος του n
n = -5:10;

% Ορισμός των εισόδων x1[n] και x2[n]
x1 = zeros(size(n)),
x1_idx = (n >= 0) & (n <= 4) % Λογικός δείκτης για 0 <= n <= 4
x1_values = [2 2 3 3 3 4] % Τιμές για x1 στο εύρος 0 έως 4
x1(x1_idx) = x1_values(1:sum(x1_idx)); % Ανάθεση τιμών στο x1

u = @(n) n >= 0,
x2 = u(n) - u(n - 6),

% Εξίσωση συστήματος
sys_eq = @(x, n) 4*n.^2 .* x,

% Υπολογισμός των εξόδων y1[n] και y2[n]
y1 = sys_eq(x1, n),
y2 = sys_eq(x2, n),

% Σταθερές
a = 4;
b = 5;

% Συνδυασμένη είσοδος και έξοδος
x_combined = a * x1 + b * x2,
y_combined = sys_eq(x_combined, n),

% Έλεγχος γραμμικότητας (υπέρθεση)
y_test = a * y1 + b * y2,

% Απεικόνιση των εισόδων και εξόδων
subplot(3,2,1),
stem(n, x1),
title('Είσοδος x1[n]'),

subplot(3,2,2),
stem(n, x2),
title('Είσοδος x2[n]'),

subplot(3,2,3),
stem(n, y1),
title('Έξοδος y1[n] για x1[n]'),

subplot(3,2,4),
stem(n, y2),
title('Έξοδος y2[n] για x2[n]'),


subplot(3,2,5),
stem(n, y_combined),
title('Έξοδος για a * x1[n] + b * x2[n]'),

subplot(3,2,6),
stem(n, y_test),
title('a * y1[n] + b * y2[n]'),
```

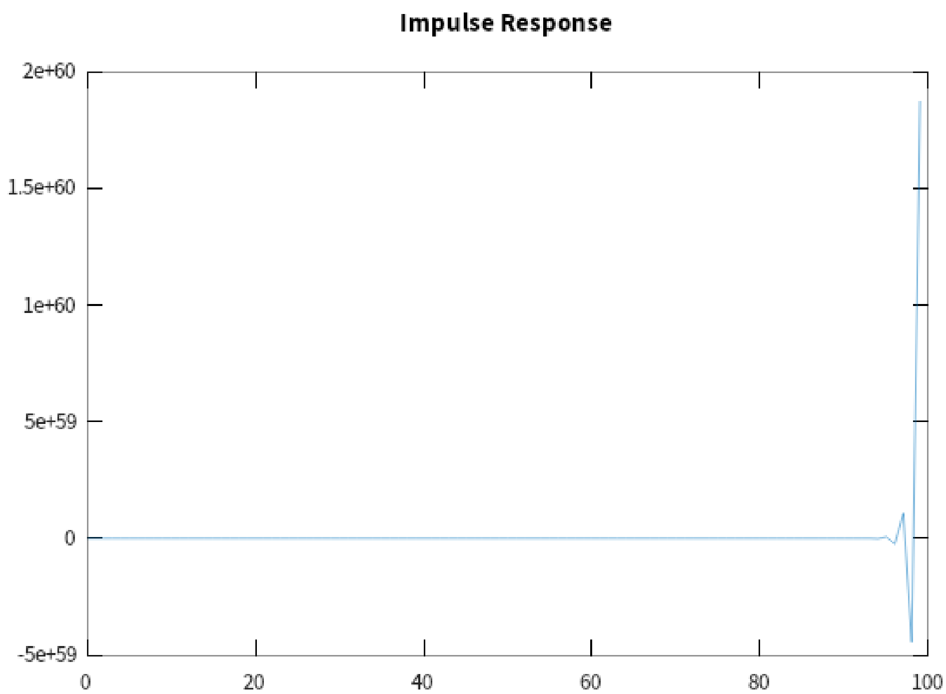


9. Δίνεται η εξίσωση διαφορών ενός συστήματος:□[□] + 3□[□ − 1] − 5□[□ − 2] = 0.5□[□] − □[□ − 2]α) Να υπολογιστεί και να γίνει η γραφική παράσταση της κρουστικής απόκρισης.β) Απ’ αυτή τη γραφική παράσταση να εξηγήσετε αν το σύστημα είναι ΒΙΒΟευσταθές ή όχι.γ) Να υπολογιστεί η έξοδος για είσοδο την x=[2 1 1 1 0 0 1 1 0 0 2] χρησιμοποιώντας τη filterκαι την conv και να εξηγήσετε αυτά που βλέπετε σε σχέση με αυτά που σχολιάσαμε στοφυλλάδιο.δ) Επαναλάβατε το ερώτημα (γ) για είσοδο x[n] = 2cos(3π n).

Επεξεργασία σκέψης άσκησης:

A) Για να υπολογισούμε την κρουστική απόκριση, πρέπει να προσομοιώσουμε την απόκριση του συστήματος σε μια κρουστική είσοδο \delta[n] η οποία είναι 1 στο n=0 και 0 άλλου

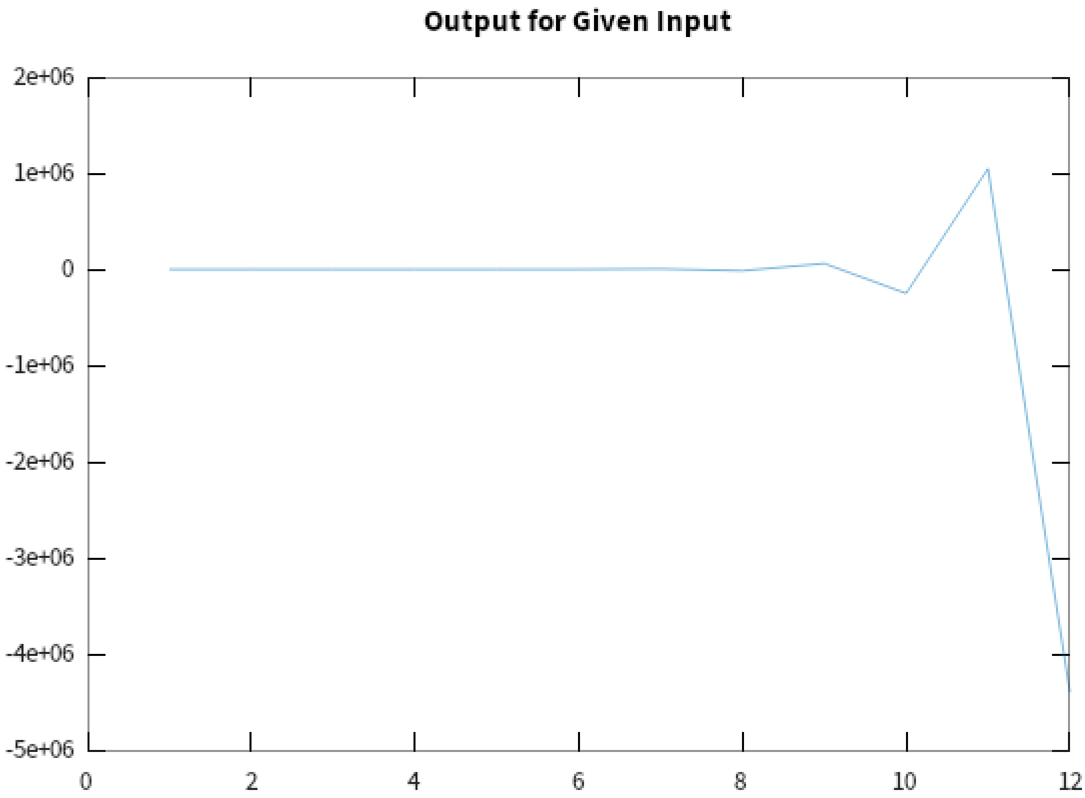
```
% Ορισμός του παλμικού σήματος
N = 100; % Αριθμός δειγμάτων, μπορεί να ρυθμιστεί
x = [1, zeros(1, N-1)]; % Εισαγωγή παλμού

% Αρχικοποίηση y
y = zeros(1, N),

% Εφαρμογή της εξίσωσης διαφοράς
for n = 3:N
    y(n) = 0,5 * x(n) - x(n-2) - 3 * y(n-1) + 5 * y(n-2),
end

% Αναγκάζει τη δημιουργία ενός νέου παραθύρου σχήματος
figure,

% Απεικόνιση της κρουστικής απόκρισης
plot(0:N-1, y),
title('Impulse Response'),


```



B) Για να προσδιορίσουμε την ευστάθεια BIBO από την κρουστική απόκριση , ελέγχουμε αν η κρουστική απόκριση είναι απόλυτως αθροιστική , έαν δεν συγκλίνει στο 0 ή αύξανεται στο απείρο δεν είναι ευσταθές BIBO.

Γ)Για την είσοδο x = [2, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 2] μπορούμε να χρησιμοποιήσουμε τη συνάρτηση filter

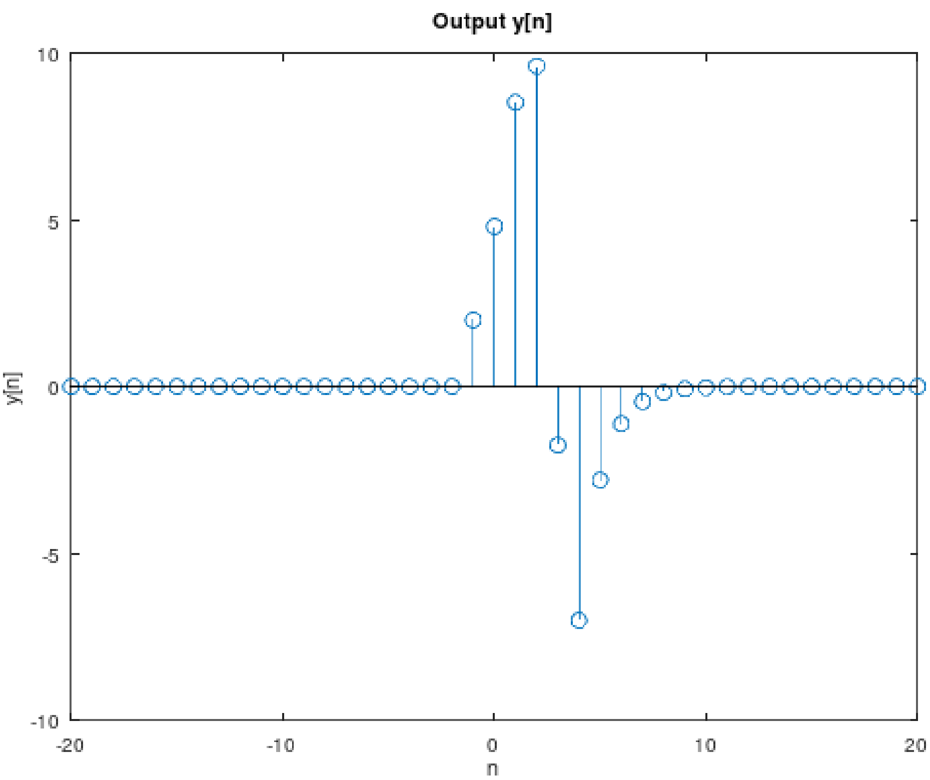
```
% Ορίστε την είσοδο
x = [2, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 2];

% Ορίστε τους συντελεστές της εξίσωσης διαφορών
b = [0.5, 0, -1] % Συντελεστές του x
a = [1, 3, -5] % Συντελεστές του y

% Υπολογισμός της εξόδου
y = filter(b, a, x),

% Απεικόνιση της εξόδου
plot(y),
title('Output for Given Input'),

```



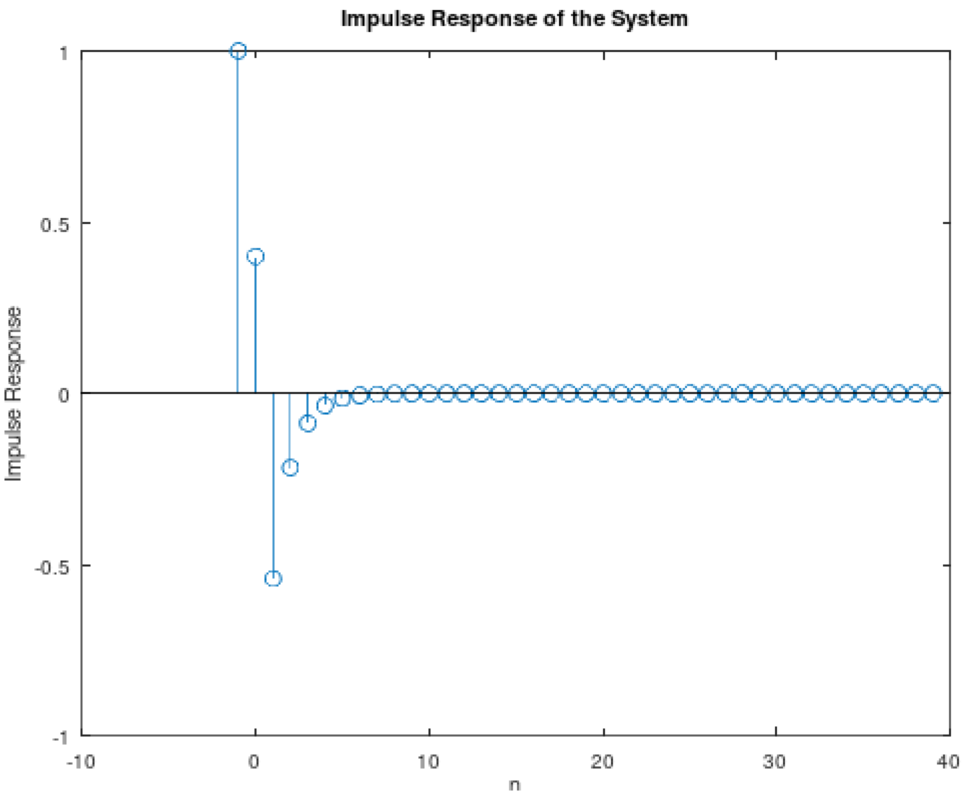
<u>**10. Δίνεται σύστημα που περιγράφεται από την ΕΔ □[□] = 0.4□[□ − 1] + □[□] − 0.7□[□ − 2] a) Να βρεθεί και να σχεδιαστεί η έξοδος για χρόνο n=[-20 : 20] όταν η είσοδος είναι □[□] = 2□[□ + 1] + 4□[□] + 8□[□ − 1] + 9□[□ − 2] b) Να βρεθεί και να σχεδιαστεί για n=[-20 : 20]**</u>

κρουστική απόκριση του συστήματος με χρήση της συνάρτησης impz (help impz για σύνταξη).

```
% Ορίστε το εύρος του n
n = -20:20;

% Ορισμός της εισόδου x[n]
x = 2 * (n == -1) + 4 * (n == 0) + 8 * (n == 1) + 9 * (n == 2),

% Αρχικοποίηση y[n]
y = zeros(size(n)),

% Εφαρμογή της εξίσωσης διαφορών
for i = 3:length(n)
    y(i) = 0,4 * y(i - 1) + x(i) - 0,7 * (i-2 >= 1) * x(i - 2),
end

% Απεικόνιση της εξόδου
stem(n, y),
title('Output y[n]'),
xlabel('n'),
ylabel('y[n]'),
```



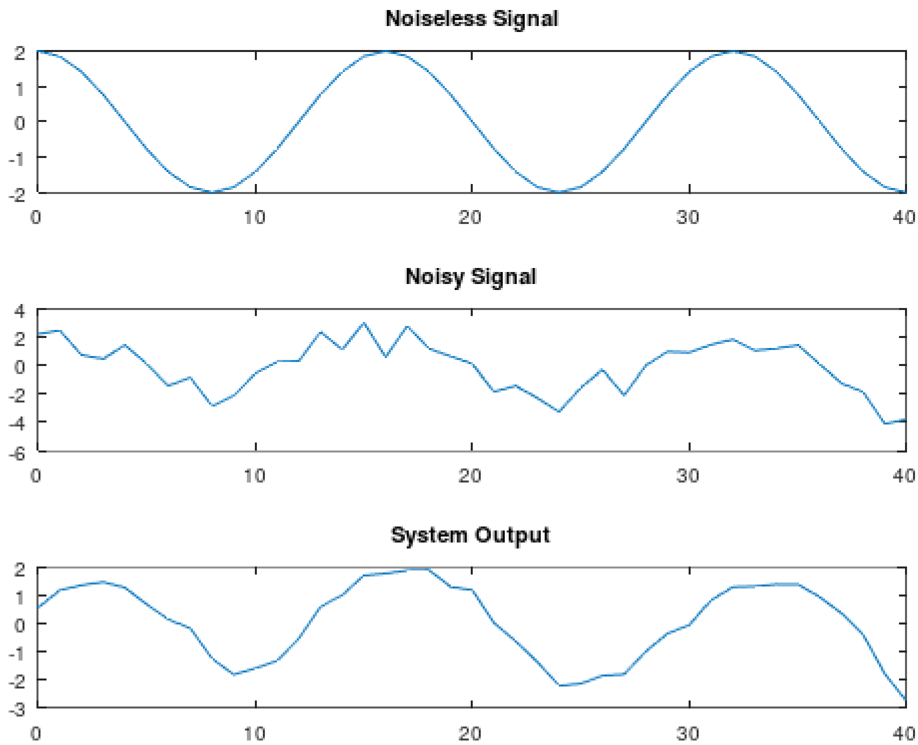
```
% Συντελεστές της εξίσωσης διαφοράς
b = [1, 0, -0.7]; % Συντελεστές για x[n]
a = [1, -0.4]; % Συντελεστές για y[n]

% Καθορισμός του εύρους του n για τη γραφική παράσταση
n = -20:20;

% Απεικόνιση της κρουστικής απόκρισης με τη χρήση impz
[impulse_response, response_n] = impz(b, a, length(n)),
stem(response_n - 1, impulse_response);
title('Impulse Response of the System'),
xlabel('n'),
ylabel('Impulse Response'),
```

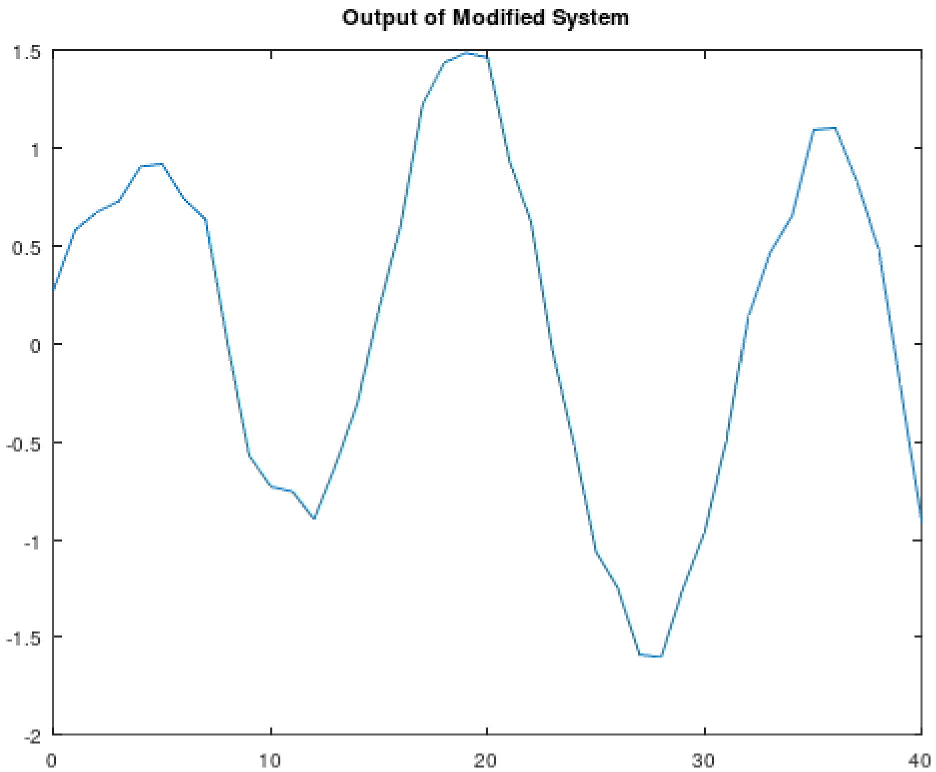


11 . Σύστημα διακριτού χρόνου περιγράφεται από την εξίσωση διαφορών: □[□] = 1 4 ∑ □[□ − □]. 3 □=0 Να βρεθεί η έξοδος του συστήματος όταν η είσοδός του είναι το σήμα □[□] = 2 ∗ cos (□□ 8) + □[□], □ = 0: 40. Οπου w[n] είναι θόρυβος Gaussian που προστίθεται στο ημιτονοειδές σήμα (να χρησιμοποιηθεί η συνάρτηση randn()). Να γίνουν οι γραφικές παραστάσεις του αθόρυβου σήματος, του ενθόρυβου σήματος και της εξόδου του συστήματος. Ποια πιστεύετε οτι είναι η δράση του συστήματος στο ενθόρυβο σήμα? Επαναλάβετε τη

διαδικασία για το σύστημα □[□] = 1 8 ∑ □[□ − □] 7 □=0 . Συγκρίνετε τα δύο συστήματα.

```
n = 0:40;

% Ορισμός του σήματος εισόδου
x = 2 * cos(pi * n / 8),

% Προσθήκη γκαουσιανού θορύβου
w = randn(size(n)),
x_noisy = x + w,

% Υπολογισμός της εξόδου του συστήματος
y = filter(ones(1, 4) / 4, 1, x_noisy),

% Απεικόνιση των σημάτων
figure,
subplot(3, 1, 1),
plot(n, x),
title('Noiseless Signal'),

subplot(3, 1, 2),
plot(n, x_noisy),
title('Noisy Signal'),

subplot(3, 1, 3),
plot(n, y),
title('System Output'),

% Επανάληψη για τροποποιημένο σύστημα (y[n] = 1/8 * άθροισμα x[n - m], m=0:7)
y_modified = filter(ones(1, 8) / 8, 1, x_noisy),

% Σχεδιάστε την έξοδο του τροποποιημένου συστήματος
figure,
plot(n, y_modified),
title('Output of Modified System'),
```



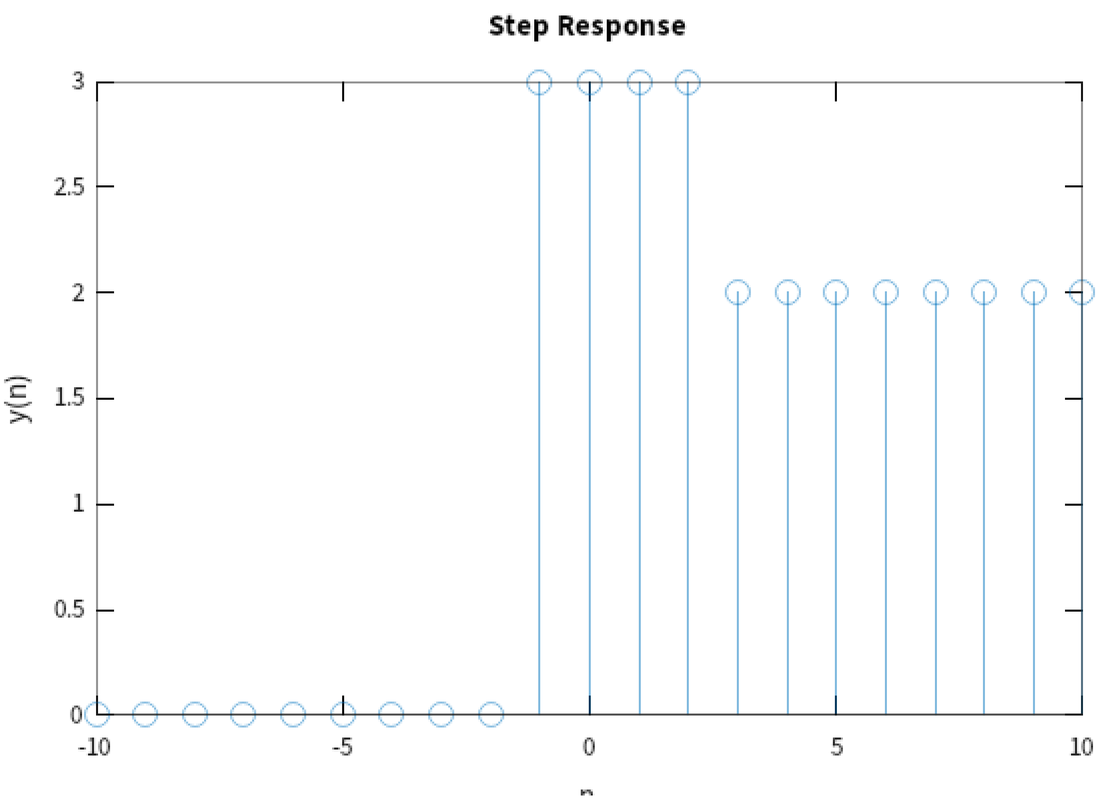


Επεξεργασία σκέψης άσκησης:

Για να βρούμε την έξοδο του συγκεκριμένου συστήματος και να ανλύσουμε την επίδραση του στο θορυβώδες σήμα. Αρχικά θα ορίσω το σήμα εισόδου , δημιουργώ ένα ημητονειδές σήμα και προσθέστω σε αυτό γκαουσιανό θόρυβο . Ο θόρυβος μπορεί να παραχθεί χρησιμοποιώντας τη συνάρτηση randn() στο Octave.

12. Δίνεται το σύστημα διακριτού χρόνου που περιγράφεται από την ακόλουθη εξίσωση διαφορών: □(□) = 3□(□ + 1) − □(□ − 3) Να σχεδιαστεί η απόκρισή του στη μοναδιαία βηματική ακολουθία (βηματική απόκριση) στο διάστημα −10 ≤ □ ≤ 10.

```
% Ορίστε το εύρος του n
n = -10:10;

% Ορισμός της συνάρτησης μοναδιαίου βήματος
u = @(n) n >= 0;

% Αρχικοποίηση y
y = zeros(size(n));

% Υπολογισμός της απόκρισης
for i = 1:length(n)
    y(i) = 3 * u(n(i) + 1) - u(n(i) - 3);
end

% Απεικόνιση της βηματικής απόκρισης
stem(n, y);
title('Step Response');
xlabel('n');
ylabel('y(n)');
```



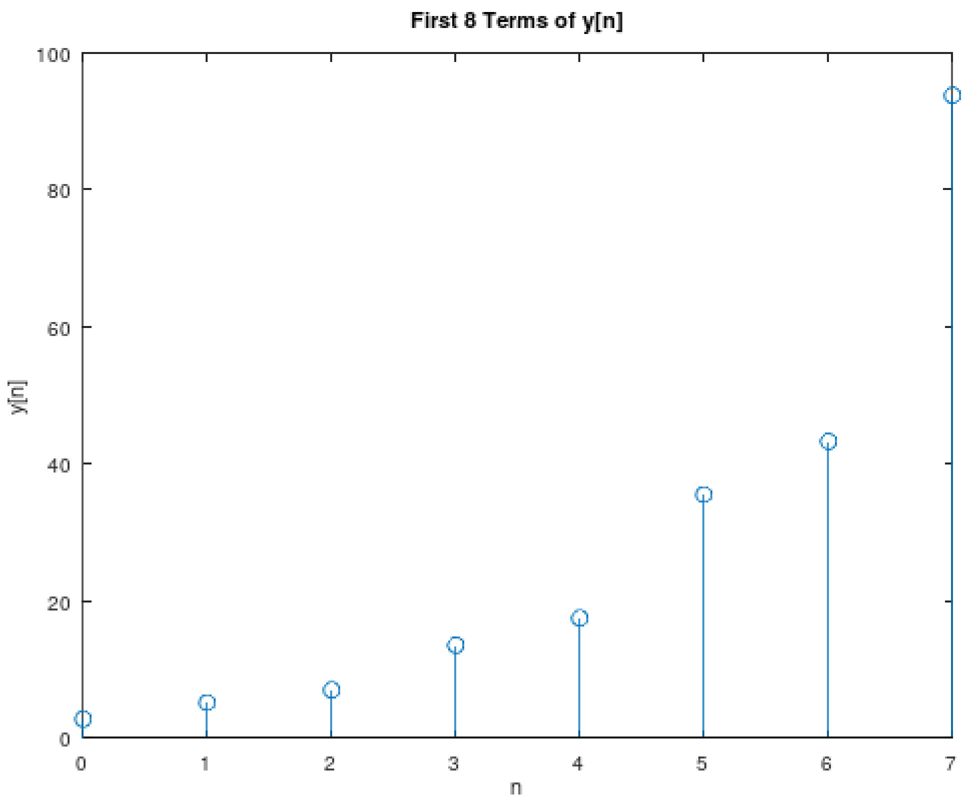
Επεξεργασία σκέψης άσκησης:

Η βηματική απόκριση είναι η εξόδος όταν η είσοδος είναι μια μονοδιαία βηματική συνάρτηση.Υπολογίζω το y[n]χρησιμοποιώντας τη δεδομένη εξίσωση διαφορών όπου το x(n) αντικαθίσταται από το u(n).

13.Να βρεθούν οι ακολουθίες που έχουν τους ακόλουθους μετασχηματισμούς Z, και στη συνέχεια να υπολογιστούν οι

8 πρώτοι όροι τους □(□) = □ □ − 1/3 □(□) = □ 2 + 1.6□ (□ + 1.8)(0.5□ − 0.8)

```
% Ορίστε τους συντελεστές του Y(z)
b = [1, 1.6, 0] % Συντελεστές αριθμητή
a = conv([1, 1.8], [0.5, -0.8]); % Συντελεστές παρονομαστή

% Μερική αποσύνθεση κλάσματος
[r, p, k] = residue(b, a),

% Αριθμός όρων προς υπολογισμό
N = 8;
n = 0:N-1,

% Αρχικοποίηση y[n]
y = zeros(1, N),

% Υπολογισμός του y[n] με τη χρήση του αναπτύγματος μερικών κλασμάτων
for i = 1:length(r)
    y = y + r(i) * p(i).^n,
end

% Σχεδιάστε το y[n]
stem(n, y),
title('First 8 Terms of y[n]'),
xlabel('n'),
ylabel('y[n]'),
```



Επεξεργασία σκέψης άσκησης:

Για τον υπολογίσμο των πρώτων 8 ορών του χ[n] και Y(z) θα πρέπει να εκτελέσω μερική αποσύνθεση κλασμάτος και στη να βρούμε τον ανίστροφο μετασχηματισμό Z.

14. Με χρήση των συναρτήσεων poly() και residue() να αναλυθεί σε άθροισμα απλών κλασμάτων η ρητή συνάρτηση : □(□) = □ 2 + z + 1 □(□ + 0.4)(□ + 0.5)

Επεξεργασία σκέψης ασκήσης:

Για να ανλύσουμε τη συνάρτηση X(z) χρησιμοποιώντας τις συναρτήσεις poly() και residue() θα εκφράσουμε πρώτα τον αριθμήτη και τον παρονομάστη ως πολύωνημα και στη συνέχεια θα εφαρμόσουμε μερική αποσύνθεση κλασμάτων.

Για τον παρονομάστη χρησιμοποιήουμε poly() για να λάβουμε την πολυωνυμική αναπαράσταση των ριζών [0, -0,4, -0,5].Χρησιμοποιήω την residue() για να αναλύσω τη συνάρτηση σε μερικά κλάσματα.

```
% Ορισμός των πολυωνύμων αριθμητή και παρονομαστή
num = [1, 1, 1],
den = poly([0, -0.4, -0.5]),

% Μερική αποσύνθεση κλάσματος
[r, p, k] = residue(num, den),

% Εμφάνιση των αποτελεσμάτων
disp('Residues:'),
disp(r),
disp('Poles:'),
disp(p),
disp('Direct Terms:'),
disp(k),

```

```
Residues:
   15
  -19
    5
Poles:
  -0.5000
  -0.4000
        0
Direct Terms:
[](0x0)
>> exercise9a

Residues:
   15
  -19
    5
Poles:
  -0.5000
  -0.4000
        0
Direct Terms:
[](0x0)
>> exercise9a

num =

   1   1   1

den =

   1.0000   0.9000   0.2000        0

r =

   15
  -19
    5

p =

  -0.5000
  -0.4000
        0

k = [](0x0)
Residues:
   15
  -19
    5
Poles:
  -0.5000
  -0.4000
        0
Direct Terms:
[](0x0)
```

15 . Δίνεται σύστημα ΓΧΑ με κρουστική απόκριση h(n)=0.9n u(n). Να βρεθεί η συνάρτηση μεταφοράς του, να διερευνηθεί αν είναι ευσταθές, να γίνει το διάγραμμα πόλων μηδενικών.

Επεξεργασία σκέψης άσκησης:

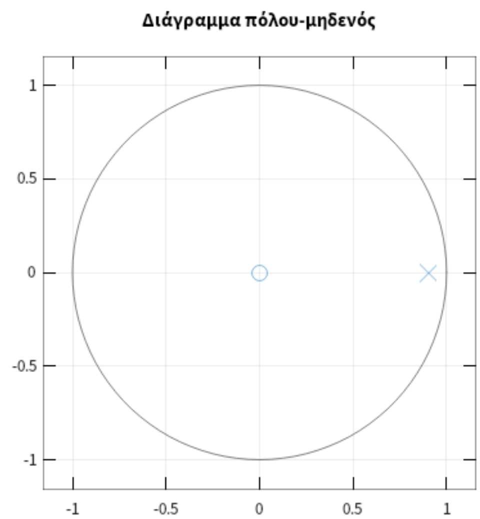
Για ένα σύστημα με κρουστική απόκριση h(n) όπου u(n) είναι η μοναδίαι βημαική συνάρτηση , βρισκώ τη σύναρτηση μεταφοράς και να αναλύσω την ευσταθεία της.

Συνάρτηση μεταφοράς: Η συνάρτηση μεταφοράς H(z) ενός συστήματος διακριτού χρόνου με κρουστική απόκριση h(n) είναι ο μετασχηματισμός Z του h(n).

Σταθερότητα: Ένα σύστημα είναι ευσταθές εάν η περιοχή σύγκλισης (ROC) της συνάρτησης μεταφοράς του περιλαμβάνει τον μοναδιαίο κύκλο. Στην περίπτωση αυτή, η ROC είναι |z| > 0.9 η οποία περιλαμβάνει τον μοναδιαίο κύκλο (αφού οι πόλοι του συστήματος βρίσκονται μέσα στον μοναδιαίο κύκλο). Επομένως, το σύστημα είναι ευσταθές.

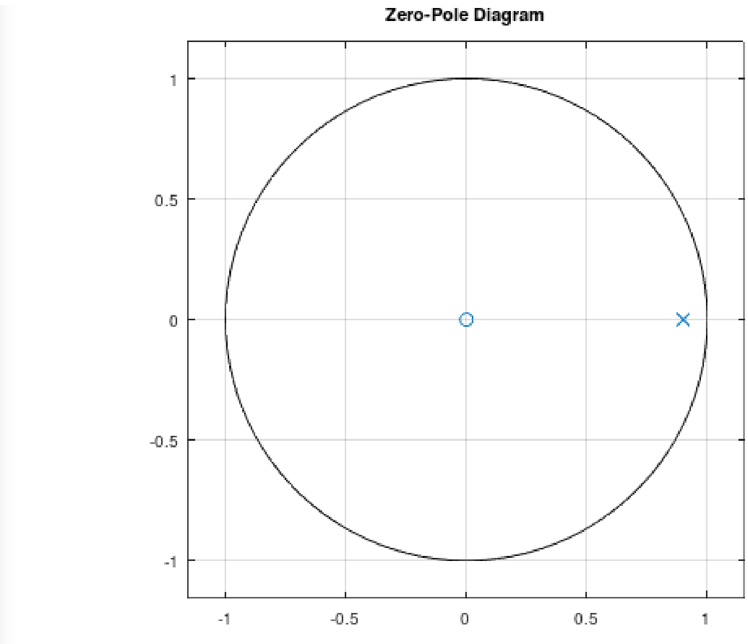
Το παραπάνω προγράμμα εμφάνιζει το διαγράμμα πόλων-μηδενικών για τη δεδομένη συνάρτηση μεταφοράς, βοηθώντας στην οπτικοποίηση του συστήματοςπόλους και τα μηδενικά του συστήματος και επιβεβαιώνει περαιτέρω τα χαρακτηριστικά ευστάθειάς του.

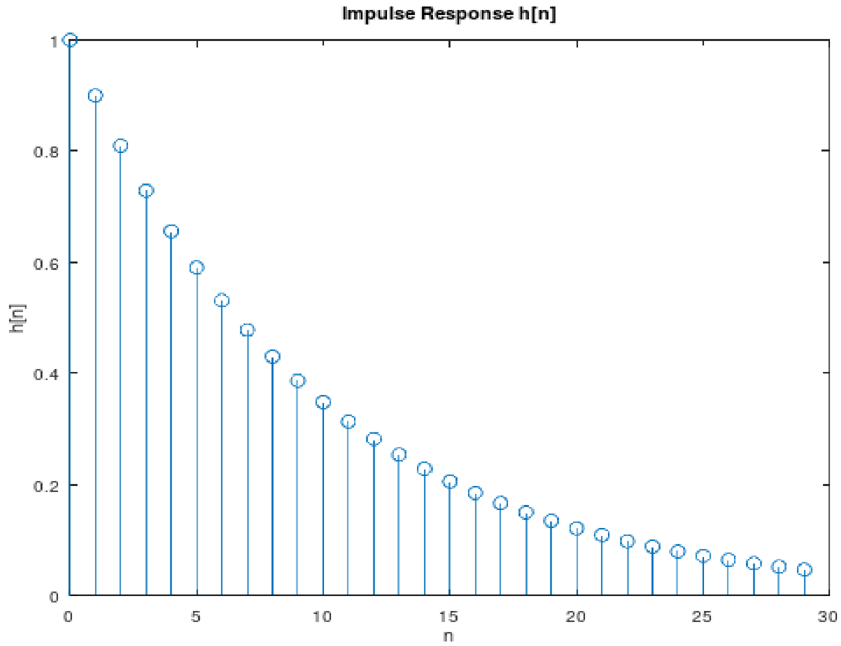
```
% Καθορισμός των συντελεστών της συνάρτησης μεταφοράς
b = 1; % Συντελεστές αριθμητή
a = [1, -0.9]; % Συντελεστές παρονομαστή

% Διάγραμμα πόλου-μηδενός
zplane(b, a),
title('Διάγραμμα πόλου-μηδενός'),

```



17 . Να υπολογιστεί η συνάρτηση μεταφοράς και η κρουστική απόκριση του αιτιατού ΓΧΑ συστήματος διακριτού χρόνου πρώτης τάξης, το οποίο περιγράφεται από την εξίσωση διαφορών: □(□) − 0.9□(□ − 1) = □(□) Να γίνει το διάγραμμα πόλων μηδενικών και να προσδιοριστεί η ΠΣ.

```
% Συντελεστές συνάρτησης μεταφοράς
b = 1; % Συντελεστής αριθμητή
a = [1, -0.9]; % Συντελεστές παρονομαστή

% Δημιουργία συνάρτησης μεταφοράς
H = tf(b, a, -1); % Το '-1' δηλώνει σύστημα διακριτού χρόνου

% Εμφάνιση της συνάρτησης μεταφοράς
disp('Transfer Function H(z):'),
disp(H),

% Υπολογισμός της κρουστικής απόκρισης
N = 30; % Αριθμός δειγμάτων για την κρουστική απόκριση
impulse_response = filter(b, a, [1, zeros(1, N-1)]),

% Απεικόνιση της κρουστικής απόκρισης
Σχήμα,
stem(0:N-1, impulse_response),
title('Impulse Response h[n]'),
xlabel('n'),
ylabel('h[n]'),

% Διάγραμμα μηδενικού πόλου
Σχήμα,
zplane(b, a),
title('Zero-Pole Diagram'),
```



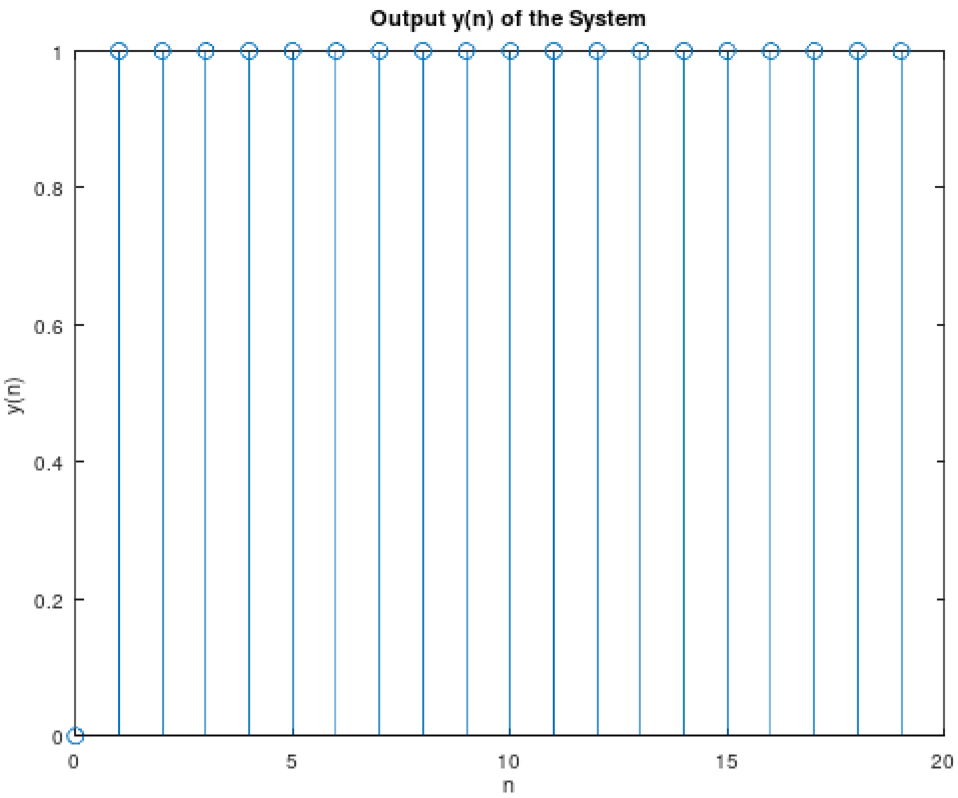


Επεξεργασία σκέψης άσκησης:

Αυτός ο κώδικας υπολογίζει τη συνάρτηση μεταφοράς, σχεδιάζει την κρουστική απόκριση και το διάγραμμα μηδενικού πόλου. Είναι σημαντικό να σημειωθεί ότι αυτό το σύστημα έχει πόλο στο 0,9, ο οποίος βρίσκεται εντός του μοναδιαίου κύκλου, υποδεικνύοντας σταθερότητα. Η κρουστική απόκριση, όπως υπολογίζεται από τη συνάρτηση φίλτρου, θα δείξει την απόκριση του συστήματος με την πάροδο του χρόνου.

18.Δίνεται το αιτιατό ΓΧΑ σύστημα διακριτού χρόνου του οποίου η είσοδος και η έξοδος συνδέονται από την εξίσωση διαφορών □(□) − 0.5□(□ − 1) = □(□) Να υπολογιστεί η έξοδος του συστήματος αν το σήμα εισόδου είναι x(n) = u(n) και το σύστημα βρίσκεται σε ηρεμία.

```
%το μηκος της ακολουθιας
N = 20;

% Αρχικοποίηση των x(n) και y(n)
x = ones(1, N) % Συνάρτηση μοναδιαίου βήματος u(n)
y = zeros(1, N); % Το σύστημα αρχικά σε ηρεμία

% Εφαρμογή της εξίσωσης διαφορών
for n = 2:N % Ξεκινώντας από το 2 αφού χρειαζόμαστε το y(n-1)
    y(n) = x(n) + 0,5 * y(n - 1),
end

% Απεικόνιση της εξόδου
stem(0:N-1, y),
title('Output y(n) of the System'),
xlabel('n'),
ylabel('y(n)'),

```



Επεξεργασία σκέψης άσκησης:

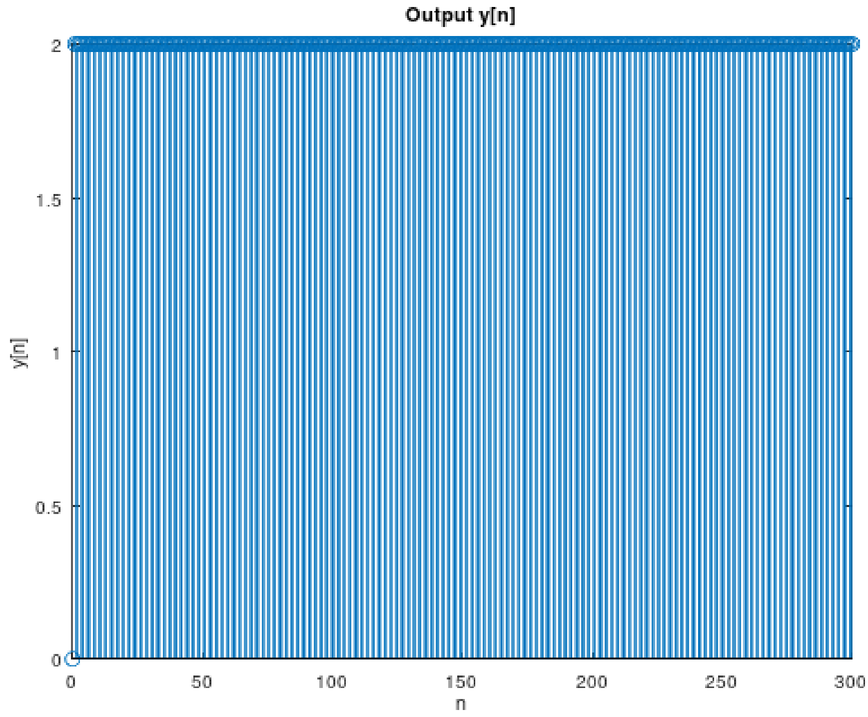
Αύτος ο κωδικάς υπολογίζει και σχεδίαζει την εξόδος του συστήματος για μια είσοδο μονοδίαιου βήματος σε εύρος n εώς N-1.Το συστήμα θεωρείται οτί βρίσκεται ηρεμία , οπότε το y(n) ξεκινά άπο το 0.

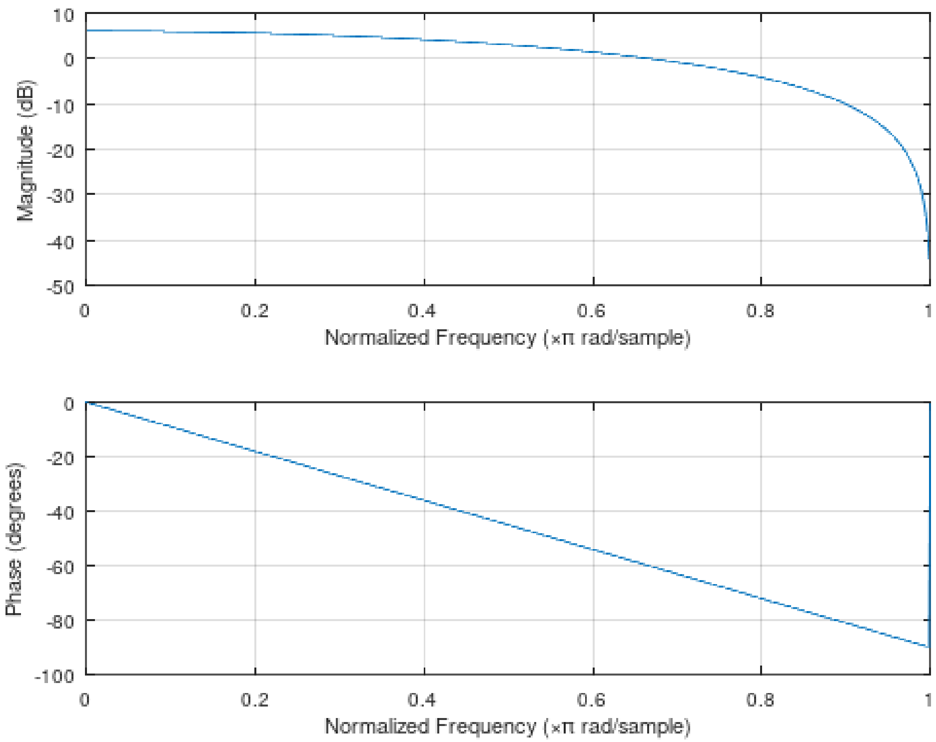
20. Σε σύστημα που χαρακτηρίζεται από την εξίσωση διαφορών: □[□] = □[□] + □[□ − 1] εφαρμόζεται είσοδος □[□] = □[□]. Να σχεδιαστεί η έξοδος y[n] για 0 ≤ □ ≤ 300. Με χρήση της freqz() να αναπαρασταθεί η απόκριση συχνότητας του συστήματος.

Επεξεργασία σκέψης άσκησης:

Αυτός ο κώδικας θα δημιουργήσει δύο διαγράμματα: ένα για την έξοδο y[n] και ένα για την απόκριση συχνότητας του συστήματος. Το διάγραμμα απόκρισης συχνότητας θα δείξει πώς ενισχύονται ή εξασθενούν οι διάφορες συνιστώσες συχνότητας στο σήμα εισόδου από το σύστημα.

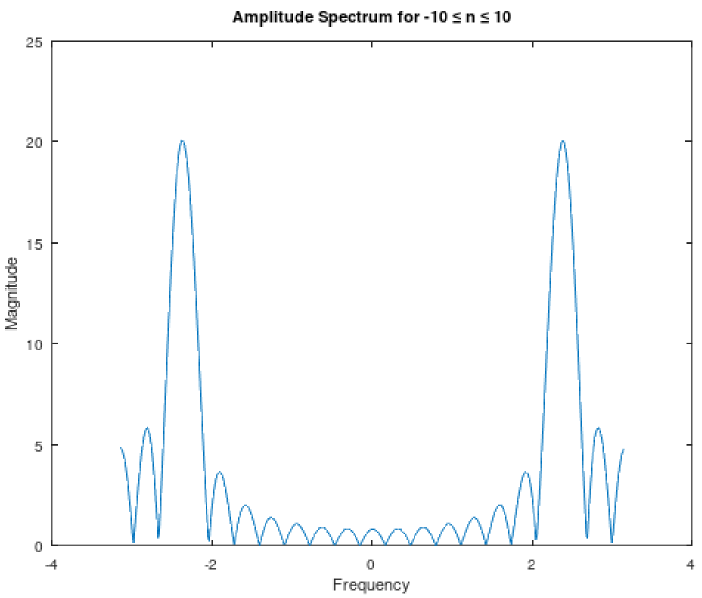
```
% Ορίστε το εύρος του n
n = 0:300;

% Δημιουργία του σήματος εισόδου x[n]
x = (n >= 0) % Συνάρτηση βήματος μονάδας

% Αρχικοποίηση y[n]
y = zeros(size(n)),

% Εφαρμογή της εξίσωσης διαφοράς
for i = 2:length(n)
    y(i) = x(i) + x(i - 1),
end

% Απεικόνιση της εξόδου
figure,
stem(n, y),
title('Output y[n]'),
xlabel('n'),
ylabel('y[n]'),

% Απόκριση συχνότητας με χρήση freqz
b = [1, 1]; % Συντελεστές του φίλτρου (αριθμητής)
a = 1; % Συντελεστής του φίλτρου (παρονομαστής)
figure,
freqz(b, a),
```

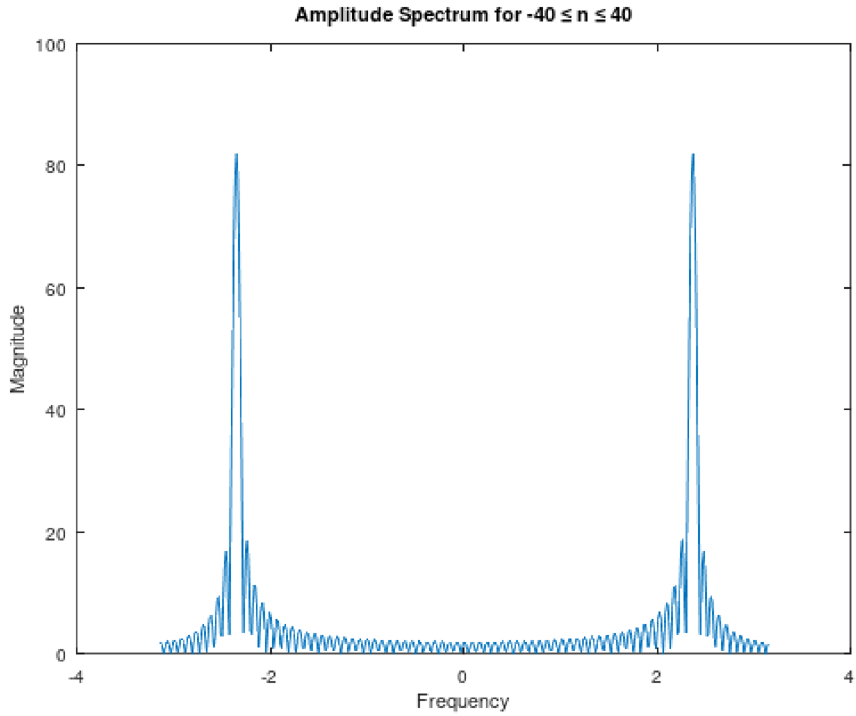

Output y[n]
2
1.5
1
0.5
0
y[n]
0
50
100
150
200
250
300
n


10
0
-10
-20
-30
-40
-50
Magnitude (dB)
0
0.2
0.4
0.6
0.8
1
Normalized Frequency (×π rad/sample)
0
-20
-40
-60
-80
-100
Phase (degrees)
0
0.2
0.4
0.6
0.8
1
Normalized Frequency (×π rad/sample)

23. Ας θεωρηθεί το σήμα □[□] = 2□□□(3□□/4), -10 ≤ □ ≤ 10.Να βρεθεί το φάσμα του σήματος αυτού με χρήση της fft σε 512 σημεία. Αναπαραστήστε τοφάσμα πλάτους σε γραφικό παράθυρο. Παρατηρήστε τη διασπορά συχνοτήτων σε εύροςμεγαλύτερο από τις αναμενόμενες θεωρητικά φασματικές γραμμές του συνημιτόνου. Τοφαινόμενο αυτό οφείλεται στον περιορισμό της άπειρης διάρκειας του συνημιτόνου καιονομάζεται φασματική διασπορά.Επαλαβετε την ίδια διαδικασία για το σήμα x[n], θεωρώντας αυτή τη φορά -40 ≤ □ ≤ 40.Σχολιάστε το φάσμα που προκύπτει.

```
% Σήμα για -10 ≤ n ≤ 10
n1 = -10:10;
x1 = 2 * cos(3 * pi * n1 / 4),
X1 = fft(x1, 512),
f1 = linspace(-pi, pi, 512),

% Απεικόνιση του φάσματος πλάτους για -10 ≤ n ≤ 10
figure,
plot(f1, abs(fftshift(X1))),
title('Amplitude Spectrum for -10 ≤ n ≤ 10'),
xlabel('Frequency'),
ylabel('Magnitude'),

% Σήμα για -40 ≤ n ≤ 40
n2 = -40:40;
x2 = 2 * cos(3 * pi * n2 / 4),
X2 = fft(x2, 512),
f2 = linspace(-pi, pi, 512),

% Απεικόνιση φάσματος πλάτους για -40 ≤ n ≤ 40
figure,
plot(f2, abs(fftshift(X2))),
title('Amplitude Spectrum for -40 ≤ n ≤ 40'),
xlabel('Frequency'),
ylabel('Magnitude'),
```





Επεξεργασία σκέψης άσκησης:

Δημιουργώ το σήμα x[n] για το εύρος -10 μέχρι 10 χρησιμοποίο την συνάρτηση FFT για να υπολογίσω σε φάσμα 512 σημεία.Θα σχεδίασω το φάσμα πλάτους για -40<=n<=40. Η φασματική διάσπορα εμφανίζεται λογώ της πεπερασμένης διάρκειας του σήματος.Η φασματική διασπορά εμφανίζεται λογώ της πεπερασμένης διάρκειας του σήματος.Άπο τα δύο σήματα διαφορετικής διάρκειας η μεγαλύτερη διάρκεια επήρεαζει τη φασματικλη αναλύση.

<u>29. Να υπολογιστεί η ισχύς της μοναδιαίας βηματικής ακολουθίας:□[□], □□□ − 5000 ≤ □ ≤ 5000.Συγκρίνετε το αποτέλεσμα που βρήκατε με το αποτέλεσμα από τη θεωρητική επίλυση.</u>

Επεξεργασία σκέψης άσκησης:

Για να υπολογίσω την ισχύ της ακολούθιας μονοδίαου βήματος u[n] για -5000 εώς 5000 θα χρησιμοποιήσω τον ορισμό της ισχύος, ο οποίος είναι ο μέσος όρος του τετραγώνου του σήματος με τη πάροδο του χρόνουΔεδομένου ότι η συνάρτηση μοναδιαίου βήματος u[n] είναι 1 για n≥0 και 0 διαφορετικά, το τετράγωνό της είναι επίσης 1 για 0n≥0. Δεδομένου του εύρους από -5000 έως 5000, υπάρχουν 5001 δείγματα όπου u[n] = 1.

Power=1/10001

5000

∑n=-5000 u[n]^2=5001/10001

```
% Ορίστε το εύρος του n
n = -5000:5000;

% Ορισμός της ακολουθίας μοναδιαίων βημάτων u[n]
u = (n >= 0),

% Υπολογισμός της ισχύος
power_calculated = sum(u.^2) / length(n),

% Υπολογισμός θεωρητικής ισχύος
% Για τη μισή διάρκεια, u[n] = 1, επομένως η ισχύς είναι η μισή του 1^2
power_theoretical = 0,5,

% Εμφάνιση των αποτελεσμάτων
disp(['Calculated Power: ', num2str(power_calculated)]),
disp(['Θεωρητική ισχύς: ', num2str(power_theoretical)]),

```

```
Calculated Power: 0.50005
Theoretical Power: 0.5
```

Αυτός ο κώδικας υπολογίζει την ισχύ της ακολουθίας βημάτων της μονάδας στο καθορισμένο εύρος και τη συγκρίνει με τη θεωρητική ισχύ. Η θεωρητική ισχύς αναμένεται να είναι 0,5 αφού η συνάρτηση μοναδιαίου βήματος είναι 1 για το μισό χρόνο (θετικό n) και 0 διαφορετικά.

ΤΕΛΟΣ ΕΡΓΑΣΙΑΣ

ΧΡΗΣΙΜΕΣ ΠΗΓΕΣ:

- **ΨΕΣ/ECLASS/ΕΡΓΑΣΤΗΡΙΟ/ΕΡΓΑΣΙΕΣ**
- **ΕΡΓΑΣΤΗΡΙΑΚΑ ΜΑΘΗΜΑΤΑ**
- **ΜΑΘΗΜΑΤΑ ΘΕΩΡΙΑΣ**
- **Octave Documentation**
- **ΒΙΒΛΙΟ "ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΣΗΜΑΤΟΣ"**

ΕΡΩΤΗΜΑΤΑ ΠΟΥ ΔΕΝ ΥΛΟΠΟΙΗΘΗΚΑΝ:

- **16,19,21,22,24,25,26,27,28,30.**