**PRACTICE 3 – Z-TRANSFORM AND CAUSAL LTI DISCRETE -TIME SYSTEMS DESCRIBED AS LDECC**

**OBJECTIVES:**

1. Calculate the z-transform of special sequences x[n].
2. Calculate the inverse z-transform of rational functions.
3. Obtain the system function H(z) of causal LTI discrete-time systems described as LDEcc.
4. Obtain the output y[n] of a causal LTI discrete-time system described as a LDEcc for a finite input x[n]

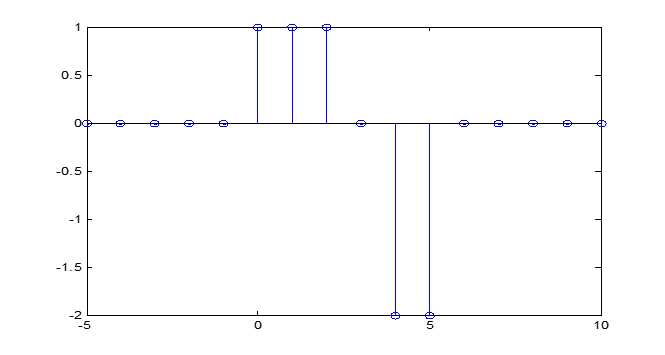
|  |  |
| --- | --- |
| **Name 1:** | Alejandro Serrano López |
| **Name 2:** | Mihnea-Iulian Popescu |

1. **Z-transform of special sequences.**

The Z-transform of a sequence x[n] is defined as:

The set of values of z for which the infinite sum converges is called the *region of convergence* (ROC).

1. Calculate the Z-transform and ROC of the following finite sequence x[n]:



>> n = -5:10;

>> x = [0, 0, 0, 0, 0, 1, 1, 1, 0, -2, -2, 0, 0, 0, 0, 0];

>> stem(n,x)

|  |
| --- |
| 1 + 1/z + 1/z^2 - 2/z^4 - 2/z^5  ROC = {z ≠ 0} |

1. Define a MATLAB function with input parameters, the finite sequence x[n], and output parameter, the Z-transform symbolic function X(z).

Remember that the MATLAB instruction “syms” creates symbolic variables and functions. For example:

syms X(z) % Create the symbolic function X(z)

X(z) = z/(z-1) % Specify the formula for X

>> X(z) = z/(z - 1)

To evaluate the function for some z=z0, write X(z0).

function [X] = z\_trans\_finite(x,n)

% x is the input sequence and n is the range of the discrete independent variable

|  |
| --- |
| **z\_trans\_finite.m** |
| function [X] = z\_trans\_finite(x,n) % x is the input sequence and n is the range of the discrete independent variable  syms z  X(z) = sum(x.\* z.^(-n))  end |

1. Try to find the z-transforms of the sequences below by using the MATLAB function “ztrans”, defining first the symbolic variable n (syms n). Specify the ROC too.

impulse = n==0;

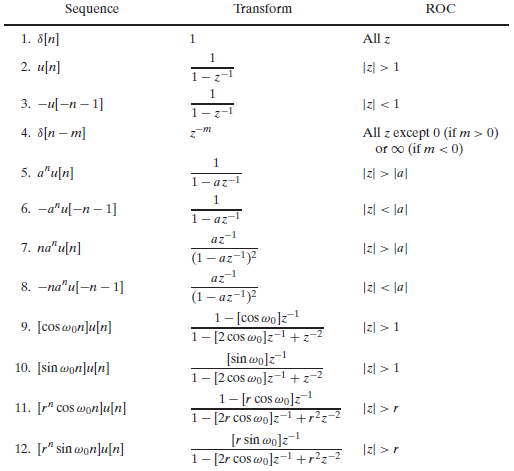
unitstep = n>=0;

ramp = n.\*unitstep;

quad = n.^2.\*unitstep;

|  |  |  |  |
| --- | --- | --- | --- |
| **Sequence** | **Using ztrans** | **Theoretical z-transform using tables or properties** | **Domain** |
| x[n] = 0.3n·u[n] | syms n  X=ztrans((0.3)^n);  >> X = z/(z - 3/10) |  | |z|>0.3 |
| x[n] = (-1)n-1·u[n-1] = δ[n] - (-1)n·u[n] | >> syms n  >> X=ztrans(kroneckerDelta(n) -(-1.0)^n)  X = 1 - z/(z + 1) |  | |z|> |-1| |
| x[n]=u[n]-2n·u[-n-1] | WE CANNOT USE ZTRANS  (u[-n-1] cannot be computed by ztrans, since that function only computes right-sided functions while that one is two-sided) |  | |z|>|1|∩ |z|< |2| = 1<|z|<|2| |
| x[n]=u[n]-u[n-3] = δ[n]+ δ[n-1]+ δ[n-2] | >> syms n  >> X=ztrans(kroneckerDelta(n,0) + kroneckerDelta(n,1) + kroneckerDelta(n,2))    X =    1/z + 1/z^2 + 1 |  | z ≠ 0 |
| x=-2n·u[n] = -2n·1n·u[n] | >> syms n  >> X=ztrans(-2.\*n)    X =    (2\*z)/(z - 1)^2 |  | |z|>1 |
| x=cos(nπ/2)·u[n] | >> syms n  >> X=ztrans(cos(n.\*pi/2))    X =    z^2/(z^2 + 1) | = | |z|>1 |

**SOME COMMON Z-TRANSFORM PAIRS**



**PROPERTIES OF THE Z-TRANSFORM**

Una captura de pantalla de un celular con letras

Descripción generada automáticamente con confianza media

1. **The inverse Z-transform of rational functions.**
2. For the following rational z-transforms X(z):
3. Calculate X(z) as a ratio of polynomials in z-1.
4. Obtain the pole-zero plots.

'zplane' requires one of the following:

DSP System Toolbox

Signal Processing Toolbox

|  |  |  |
| --- | --- | --- |
| **X(z)** | **z-1 power expression** | **Pole-zero plot** |
|  |  | zplane([0,0,1],[1,-3,2]) |
|  |  | >> zplane([1,1,-3, 2],[1,-2,0,0])  Gráfico, Diagrama  Descripción generada automáticamente con confianza media |
|  |  | >> zplane([0,1],[1,-2])  Diagrama  Descripción generada automáticamente |
|  |  | >> zplane([0,0,1,-1],[8,-6,1])  Diagrama  Descripción generada automáticamente |
|  |  | >> zplane([0,0,1],[1,-2,1])  Diagrama  Descripción generada automáticamente |
|  |  | >> zplane([0,0,1],[1,0,4])  Gráfico, Diagrama  Descripción generada automáticamente |

1. Knowing that the z-transform of causal sequences x[n] are the following X(z), find the inverse z-transforms x[n]:
2. Using the MATLAB function “iztrans”.
3. Using the partial fraction expansion method. Review the MATLAB function “residuez” in the MATLAB documentation. Write: >> help residuez

residuez Z-transform partial-fraction expansion.

[R,P,K] = residuez(B,A) finds the residues, poles and direct terms

of the partial-fraction expansion of B(z)/A(z),

B(z) r(1) r(n)

---- = ------------ +... ------------ + k(1) + k(2)z^(-1) ...

A(z) 1-p(1)z^(-1) 1-p(n)z^(-1)

B and A are the numerator and denominator polynomial coefficients,

respectively, in ascending powers of z^(-1). R and P are column

vectors containing the residues and poles, respectively. K contains

the direct terms in a row vector. The number of poles is

n = length(A)-1 = length(R) = length(P)

The direct term coefficient vector is empty if length(B) < length(A);

otherwise,

length(K) = length(B)-length(A)+1

If P(j) = ... = P(j+m-1) is a pole of multiplicity m, then the

expansion includes terms of the form

R(j) R(j+1) R(j+m-1)

-------------- + ------------------ + ... + ------------------

1 - P(j)z^(-1) (1 - P(j)z^(-1))^2 (1 - P(j)z^(-1))^m

[B,A] = residuez(R,P,K) converts the partial-fraction expansion back

to B/A form.

Warning: Numerically, the partial fraction expansion of a ratio of

polynomials represents an ill-posed problem. If the denominator

polynomial, A(s), is near a polynomial with multiple roots, then small

changes in the data, including roundoff errors, can make arbitrarily

large changes in the resulting poles and residues. Problem

formulations making use of state-space or zero-pole representations

are preferable.

% Example:

% Compute the partial fraction expansion of the following transfer

% function H(z) = (1 + 2z^-1) / (1 - z^-1 + 2z^-2).

num = [1 2]; % Numerator coefficients

den = [1 -1 2]; % Denominator coefficients

[r,p] = residuez(num,den) % H(z) = r(1)/(1-p(1)z^-1) + ...

% r(2)/(1-p(2)z^-1)

See also residue, prony, poly, roots, ss2tf, tf2ss, tf2zp and zp2ss.

Documentation for residuez

**NOTE:** symbolic in MATLAB is: with “syms n”.

|  |  |  |  |
| --- | --- | --- | --- |
| **X(z)** | **Using iztrans** | **Partial fractions expansion (z-1 powers)** | **Inverse z-transform**  **(See common z-transform pairs)** |
|  | syms n  syms z  X(z)= 1/(z^2-3\*z+2)  x(n)= iztrans(X)  >> x (n) = 2^n/2 + kroneckerDelta(n, 0)/2 – 1 | [R,p,k]= residuez([0 0 1] , [1 -3 2])  R = [0.5 -1]  p = [2 1]  k = 0.5 | x[n] = 0.5\* kroneckerDelta(n, 0) + 0.5\*2^n\*u[n] – u[n] |
|  | >> syms n  >> syms z  >> X(z) = 1/(z) - 1/(z.^2) + z/(z-2);  >> x(n) = iztrans(X(z))    x(n) =    kroneckerDelta(n - 1, 0) - kroneckerDelta(n - 2, 0) + 2^n | >> [R,p,k]= residuez([1 1 -3 2] , [1 -2])  R = 1  p = 2  k = 0 1 -1 | x[n] = kroneckerDelta(n -1, 0) - kroneckerDelta(n -2, 0) – 2^n \* unitStep(n) |
|  | >> syms n  >> syms z  >> X(z) = 1/(z-2);  >> x(n) = iztrans(X(z))    x(n) =    2^n/2 - kroneckerDelta(n, 0)/2 | >> [R,p,k]= residuez([0 1] , [1 -2])  R = 0.5000  p = 2  k = -0.5000 | x[n] = 2^(n-1) u(n-1) |
|  | >> syms n  >> syms z  >> X(z) = (z - 1)/(8.\*z.^3 -6.\*z.^2 +z);  >> x(n) = iztrans(X(z))    x(n) =    6\*(1/4)^n - (1/2)^n - kroneckerDelta(n - 1, 0) - 5\*kroneckerDelta(n, 0) | >> [R,p,k]= residuez([0 0 1 1] , [8 -6 1])  R = 3 -10  p = 0.5000 0.2500  k = 7 1 | x[n] = 7 \* delta(n) + delta(n-1) + 3\*0.5^n\*u(n) -10 \* 0.25^n\*u(n) |
|  | >> syms n  >> syms z  >> X(z) = 1/(z.^2-2.\*z+1);  >> x(n) = iztrans(X(z))    x(n) =    n + kroneckerDelta(n, 0) - 1 | >> [R,p,k]= residuez([0 0 1] , [1 -2 1])  R =  -2.0000  1.0000  p =  1  1  **NOTE THE MULTIPLICITY > 0**  k =  1 | x[n] = delta(n) – 2\* u(n) + (n+1) \* u(n+1) = delta(n) + n -1 |
|  | >> syms n  >> syms z  *>> X(z) = 1/(z.^2+4);*  *>> x(n) = iztrans(X(z))*    *x(n) =*    *((-2i)^(n - 1)\*1i)/4 - (2i^(n - 1)\*1i)/4 + kroneckerDelta(n, 0)/4* | >> [R,p,k]= residuez([0 0 1] , [1 0 4])  R =  -0.1250  -0.1250  p =  0.0000 + 2.0000i  0.0000 - 2.0000i  k =  0.2500 | x[n] = 0.25 \* delta(n) -0.125 \* (2i)^n\*u(n) -0.125 \* (-2i)^n\*u(n) |

1. **Obtain the System Function H(z) of causal LTI discrete-time systems described as LDEcc**

The Linear Difference Equation with constant coefficients (LDEcc):

or

implements a causal LTI system with input sequence x[n] if the auxiliary conditions are the initial-rest conditions.

**Initial-rest conditions**: if the input x[n] is zero for n less than some time , then the output y[n] is constrained to be zero for n less than .

1. How can you obtain the impulse response h[n] of this kind of systems without using the z-transform?

|  |
| --- |
| RECURSIVE METHOD  You get the polynomial equation and characteristic polynomial in order to obtain the general solution, after that we solve an equation system where we define x(n) = R(kroneckerDelta(n, 0)) (basically we replace x(n) by delta(n), x(n) = kroneckerDelta(n, 0)) , and y(n) = h(n), as well as h(0), h(1)… h(m) in initial-rest conditions, with m being the number of unknown coefficients in the general solution, solving an equation of the shape:  h(0) = generalSolution(0)  h(1) = generalSolution(1)  …  Which will later give us the values of all the coefficients to replace in the general solution. In the end, h(n) will be equal to the general solution with all the coefficients replaced.  EXPLICIT METHOD  It can also be solved by a mathematical method to obtain an explicit form. |

1. Using the z-transform, obtain the impulse response h[n] of the following causal LTI systems defined by LDEcc with initial-rest conditions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Causal LTI system** | **System Function H(z)** | **Pole-zero plot, ROC and stability**  (IN A CAUSAL LDEcc:  A ROC must be beyond the furthest pole from the center (outer pole)  It is stable if the unit circunference is contained in the ROC) | **h[n]** |
|  |  | zplane([1],[1,-1])    |z|>1, unstable. | h[n]=u[n] |
|  |  | >> zplane([1],[1,-0.5])  Diagrama  Descripción generada automáticamente  |z| > 0.5, stable. | >> X(z) = 1/(1-0.5\*z^(-1));  >> x(n) = iztrans(X(z))  x(n) =    (1/2)^n  Knowing that it has initial-rest conditions:  h[n]=0.5n \*u[n] |
|  |  | >> zplane([0,0,1],[1,-3, 2])    |z|> 2, unstable. | >> X(z) = z.^(-2)/(1-3.\*z.^(-1)+2.\*z.^(-2));  >> x(n) = iztrans(X(z))    x(n) =    2^n/2 + kroneckerDelta(n, 0)/2 – 1  Knowing that it has initial-rest conditions:  h[n]= (2n/2 +δ(n)/2 -1) \*u[n] |
|  |  | >> zplane([0,0,1/8, -1/8],[1,-6/8, 1/8])    |z|> 0.5, stable. | >> X(z) = (1/8\*z^(-2)-1/8\*z^(-3))/(1-6/8\*z^(-1)+1/8\*z^(-2));  >> x(n) = iztrans(X(z))    x(n) =    6\*(1/4)^n - (1/2)^n - kroneckerDelta(n - 1, 0) - 5\*kroneckerDelta(n, 0)  h[n]= (6\*(1/4)^n - (1/2)^n - kroneckerDelta(n - 1, 0) - 5\*kroneckerDelta(n, 0))\*u[n] |
|  |  | >> zplane([0,0,1],[1,-2, 2])  Diagrama  Descripción generada automáticamente  |z|> |sqrt(2)|, unstable. | >> X(z) = z^(-2)/(1-2\*z^(-1)+2\*z^(-2));  >> x(n) = iztrans(X(z))    x(n) =    ((-1)^n\*(- 1 - 1i)^(n - 1)\*1i)/2 - ((-1)^n\*(- 1 + 1i)^(n - 1)\*1i)/2 + kroneckerDelta(n, 0)/2 |
|  |  | >> zplane([1,-1,1, -1],[1, 0, 0.25])  Diagrama  Descripción generada automáticamente  |z|> 0.5, stable. | >> H(z) = (1-z.^(-1)+z.^(-2)-z.^(-3))/(1+ 0.25\*z.^(-2));  >> x(n) = iztrans(H(z))    x(n) =    (-1i/2)^(n - 1)\*(3/2 + 3i/4) + (1i/2)^(n - 1)\*(3/2 - 3i/4) - 4\*kroneckerDelta(n - 1, 0) + 4\*kroneckerDelta(n, 0) |

1. **Obtain the output y[n] of a causal LTI discrete-time system described as a LDEcc for a finite input x[n]**

To obtain the output sequence y[n] for LDEcc systems:

you can use the MATLAB function “filter(b,a,x)”. Review the MATLAB function “filter” in the MATLAB documentation. Write in the command window: help filter.

>> help filter

filter One-dimensional digital filter.

Y = filter(B,A,X) filters the data in vector X with the

filter described by vectors A and B to create the filtered

data Y. The filter is a "Direct Form II Transposed"

implementation of the standard difference equation:

a(1)\*y(n) = b(1)\*x(n) + b(2)\*x(n-1) + ... + b(nb+1)\*x(n-nb)

- a(2)\*y(n-1) - ... - a(na+1)\*y(n-na)

If a(1) is not equal to 1, filter normalizes the filter

coefficients by a(1).

filter always operates along the first non-singleton dimension,

namely dimension 1 for column vectors and non-trivial matrices,

and dimension 2 for row vectors.

[Y,Zf] = filter(B,A,X,Zi) gives access to initial and final

conditions, Zi and Zf, of the delays. Zi is a vector of length

MAX(LENGTH(A),LENGTH(B))-1, or an array with the leading dimension

of size MAX(LENGTH(A),LENGTH(B))-1 and with remaining dimensions

matching those of X.

filter(B,A,X,[],DIM) or filter(B,A,X,Zi,DIM) operates along the

dimension DIM.

Tip: If you have the Signal Processing Toolbox, you can design a

filter, D, using DESIGNFILT. Then you can use Y = filter(D,X) to

filter your data.

See also filter2, filtfilt, filtic, designfilt.

Note: FILTFILT, FILTIC and DESIGNFILT are in the Signal Processing

Toolbox.

Plot the output for the following causal LTI systems (set initial-rest conditions):

|  |  |  |
| --- | --- | --- |
| **Causal LTI system** | **Input x[n]** | **Input - output plot** |
|  | % unit step  [x,n] = step(7,0,40); | y = filter([1] , [1 -1] , x)  subplot(2,1,1); stem(n,x)  subplot(2,1,2); stem(n,y)  Gráfico, Histograma  Descripción generada automáticamente |
|  | % unit step  [x,n] = step(7,0,40); | >> y = filter([1] , [1 -0.7] , x);  >> subplot(2,1,1); stem(n,x)  >> subplot(2,1,2); stem(n,y)  Gráfico, Histograma  Descripción generada automáticamente |
|  | % sine sequence with gaussian noise  n=0:40;  x1= 5\*sin(2\*n\*pi/20);  x2= randn(1,41);  x=x1+x2; | >> y = filter([1 1 1 1 1/5], [1] , x);  >> subplot(2,1,1); stem(n,x)  >> subplot(2,1,2); stem(n,y) |
|  | % pulse with ramp  [x,n] = rectan(7,15,0,40);  x1=16\*x(1:16);  x2=16:-1:3;  x=[x1 x2 zeros(1,11)] | >> y = filter([1 -1 1 -1], [1 0 -0.25] , x);  >> subplot(2,1,1); stem(n,x)  >> subplot(2,1,2); stem(n,y)  Histograma  Descripción generada automáticamente |
|  | % pulse with ramp  [x,n] = rectan(7,15,0,40);  x1=16\*x(1:16);  x2=16:-1:0;  x=[x1 x2 zeros(1,8)] | >> y = filter([1 -1], [1] , x);  >> subplot(2,1,1); stem(n,x)  >> subplot(2,1,2); stem(n,y)  Gráfico, Histograma  Descripción generada automáticamente |