3

EXPERIMENT

Linear and Circular Convolution

AIM

To Write MATLAB programs to,

Part I: Linear Convolution

(a) Compute the linear convolution of two finite duration sequences by directly evaluating the mathematical expression for linear convolution

(b) Compute the linear convolution of two finite duration sequences by using the Toeplitz matrix method

(c) Verify the results using the MATLAB inbuilt function `conv`

Part II: Circular Convolution

(a) Compute the circular convolution of two finite duration sequences by directly evaluating the mathematical expression for circular convolution

(b) Compute the circular convolution of two finite duration sequences by using the Circulant Toeplitz matrix method

(c) Verify the results using the MATLAB inbuilt function `cconv`

Part III: Obtain the linear convolution of two finite duration sequences by means of circular convolution

Part IV: Verify the circular convolution property of DFT

THEORY

Linear convolution

The Linear convolution of two sequences $x_1[n]$ and $x_2[n]$ is computed as,

$$x_1[n] * x_2[n] = \sum_{m=-\infty}^{\infty} x_1[m]x_2[n-m]$$

If both the sequences are of finite duration and the length of $x_1[n] = p$ & the length of $x_2[n] = q$, then their linear convolution, $x_1[n] * x_2[n]$, will also be a finite duration sequence and its length will be $p+q-1$

Linear convolution using matrix method

The linear convolution operation can be viewed as a matrix multiplication, where one of the inputs represented as a row vector is multiplied with the other input converted into a Toeplitz matrix.

The linear convolution of two finite duration sequences $h[n]$ and $x[n]$, having lengths p and q respectively, can be formulated as:

$$\mathbf{xH_T} = \begin{bmatrix} x_1 & x_2 & \dots & x_{p-1} & x_p \end{bmatrix} \begin{bmatrix} h_1 & h_2 & \dots & h_q & 0 & 0 & \dots & 0 \\ 0 & h_1 & h2 & \dots & h_q & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & h_1 & \dots & \dots & h_q \end{bmatrix}$$

Where, $\mathbf{x}$ is the row vector representation of the sequence $x[n]$
$\mathbf{H_T}$ is the Toeoplitz matrix of order $p \times (p+q-1)$, obtained using the sequence $h[n]$

For example, let $x[n] = \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}$ and $h[n] = \begin{bmatrix} 5 & 6 & 7 \end{bmatrix}$
The Toeplitz convolution matrix for $h[n]$ is,

$$\mathbf{H_T} = \begin{bmatrix} 5 & 6 & 7 & 0 & 0 & 0 \\ 0 & 5 & 6 & 7 & 0 & 0 \\ 0 & 0 & 5 & 6 & 7 & 0 \\ 0 & 0 & 0 & 5 & 6 & 7 \end{bmatrix}$$

Hence, the linear convolution of sequences $x[n]$ and $h[n]$ is,

$$x[n] * h[n] = \mathbf{xH_T} = \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 & 7 & 0 & 0 & 0 \\ 0 & 5 & 6 & 7 & 0 & 0 \\ 0 & 0 & 5 & 6 & 7 & 0 \\ 0 & 0 & 0 & 5 & 6 & 7 \end{bmatrix} = \begin{bmatrix} 5 & 16 & 34 & 52 & 45 & 28 \end{bmatrix}$$

Toeplitz matrix

A **Toeplitz matrix** is a diagonal-constant matrix, *i.e.*, a matrix in which each descending diagonal from left to right is constant.

Example:

$$\begin{bmatrix} a & b & c & d & e \\ f & a & b & c & d \\ g & f & a & b & c \\ h & g & f & a & b \\ i & h & g & f & a \end{bmatrix}$$

An $n \times n$ Toeplitz matrix may be defined as a matrix A where $A_{i,j} = c_{i-j}$, for constants $c_{1-n}, \ldots, c_{n-1}$.

A Toeplitz matrix need not necessarily be square.

Circular convolution

The N-point circular convolution of two finite duration sequences $x_1[n]$ and $x_2[n]$ is another finite duration sequence of length N, given by,

$$x_3 = x_1[n] \textcircled{N} x_2[n] = \sum_{m=0}^{N-1} x_1[m] x_2[((n-m))_N]; \; 0 \le n \le N-1$$

Circular convolution using matrix method

The linear convolution operation can be viewed as a matrix multiplication, in which both the inputs are padded with zeros to make them have the same length, and then one of the inputs represented as a row vector and is multiplied with the other input converted into a Circulant matrix.

The N-point circular convolution of two finite duration sequences $x[n]$ and $h[n]$ having the same length N can be formulated as,

$$\mathbf{xH_C} = \begin{bmatrix} x_1 & x_2 & \ldots & x_{N-1} & x_N \end{bmatrix} \begin{bmatrix} h_1 & h_2 & \ldots & h_{N-1} & h_N \\ h_N & h_1 & \ldots & h_{N-2} & h_{N-1} \\ h_{N-1} & h_N & \ldots & h_{N-3} & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ h_2 & h_3 & \ldots & h_N & h_1 \end{bmatrix}$$

For example, let $x[n] = \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}$ and $h[n] = \begin{bmatrix} 5 & 6 & 7 & 8 \end{bmatrix}$

The Circulant convolution matrix for $h[n]$ is,

$$\mathbf{H_C} = \begin{bmatrix} 5 & 6 & 7 & 8 \\ 8 & 5 & 6 & 7 \\ 7 & 8 & 5 & 6 \\ 6 & 7 & 8 & 5 \end{bmatrix}$$

Hence, the 4-point circular convolution of sequences $x[n]$ and $h[n]$ is,

$$x[n] \textcircled{4} h[n] = \mathbf{x}\mathbf{H_C} = \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 & 7 & 8 \\ 8 & 5 & 6 & 7 \\ 7 & 8 & 5 & 6 \\ 6 & 7 & 8 & 5 \end{bmatrix} = \begin{bmatrix} 66 & 68 & 66 & 60 \end{bmatrix}$$

Circulant matrix

A circulant matrix is a square matrix in which all row vectors are composed of the same elements and each row vector is rotated one element to the right relative to the preceding row vector. It is a particular kind of Toeplitz matrix.

Example:

$$\mathbf{C} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 1 & 2 & 3 & 4 \\ 4 & 5 & 1 & 2 & 3 \\ 3 & 4 & 5 & 1 & 2 \\ 2 & 3 & 4 & 5 & 1 \end{bmatrix}$$

Note: A circulant matrix is fully specified by either specifying the first row vector or the first column vector.

Circular convolution property of DFT

Circular convolution property of DFT

if,

$$x_1[n] \xleftrightarrow[N]{DFT} X_1[k] \text{ and } x_2[n] \xleftrightarrow[N]{DFT} X_2[k]$$

Then,

$$x_1[n] \textcircled{N} x_2[n] \xleftrightarrow[N]{DFT} X_1[k]X_2[k]$$

Linear convolution from circular convolution

If $x_1[n]$ & $x_2[n]$ has lengths p and q respectively, their linear convolution will have a length of $p+q-1$. Thus to obtain their linear convolution from their circular convolution we pad $x_1[n]$ & $x_2[n]$ with zeros to make both their lengths equal to p+q-1. Then their (p+q-1 point) circular convolution will be the same as the linear convolution of the original sequences, $x_1[n]$ & $x_2[n]$

MATLAB FUNCTIONS USED

conv

Convolution and polynomial multiplication

```
w = conv(u,v) returns the convolution of vectors u and v
```

cconv

Modulo-n circular convolution

```
c = cconv(a,b,n) circularly convolves vectors a and b.
n is the length of the resulting vector.
```

toeplitz

Toeplitz matrix

```
T = toeplitz(c,r) returns a nonsymmetric Toeplitz matrix with
c as its first column and r as its first row.
```

fliplr

Flip array left to right

```
B = fliplr(A) returns A with its columns flipped in the
left-right direction (that is, about a vertical axis).
If A is a row vector, then fliplr(A) returns a vector of the
same length with the order of its elements reversed.  If A is
a column vector, then fliplr(A) simply returns A.
```

convmtx

Convolution matrix

```
A = convmtx(h,n) returns the convolution matrix, A, such
that the product of A and an n-element vector, x, is the
convolution of h and x.
```

ALGORITHM

Part 1: Linear Convolution

Step 1. Start

Step 2. Read/input the sequences $x[n]$ and $h[n]$

Step 3. Obtain the lengths of $x[n]$ and $h[n]$ as p and q respectively

Step 4. Compute the length of linear convolution, $L = p + q - 1$

Step 5. Compute the linear convolution by evaluating the expression

$$x[n] * h[n] = \sum_{m=-\infty}^{\infty} x[m]h[n-m]$$

Step 6. Compute the linear convolution of $x[n]$ and $h[n]$ using matrix method after forming the Toeplitz matrix for $h[n]$. The MATLAB function `toeplitz` can be used for forming the Topelitz matrix.

Step 7. Compute the linear convolution using the MATLAB inbuilt function `conv`

Step 8. Verify that all three results are the same sequence

Step 9. Stop

Part 2: Circular Convolution

Step 1. Start

Step 2. Read/input the sequences $x[n]$ and $h[n]$

Step 3. Obtain the lengths of $x[n]$ and $h[n]$ as p and q respectively

Step 4. Compute $N = max(p,q)$

Step 5. Compute the N-point circular convolution by evaluating the expression

$$x[n] \text{Ⓝ} h[n] = \sum_{m=0}^{N-1} x[m]h[((n-m))_N]; \text{ for } 0 \le n \le N-1$$

Step 6. Compute the N-point circular convolution of $x[n]$ and $h[n]$ using matrix method after forming the Circulant matrix for $h[n]$. The MATLAB function `toeplitz` can be used for forming the Circulant Topelitz matrix.

Step 7. Compute the circular convolution using the MATLAB inbuilt function `cconv`

Step 8. Verify that all three results are the same sequence

Step 9. Stop

Part 3: Linear Convolution through Circular Convolution

Step 1. Start

Step 2. Read/input the sequences $x[n]$ and $h[n]$

Step 3. Obtain the lengths of $x[n]$ and $h[n]$ as p and q respectively

Step 4. Compute the length of linear convolution, $L = p + q - 1$

Step 5. Compute the L point circular convolution of $x[n]$ and $h[n]$ with or without using the inbuilt function `cconv`

Step 6. Verify that the above result is the same as the linear convolution of $x[n]$ and $h[n]$, by using the MATLAB inbuilt function `conv`

Step 7. Stop

Part 4: Circular Convolution Property of DFT

Step 1. Start

Step 2. Read/input the sequences $x[n]$ and $h[n]$

Step 3. Obtain the lengths of $x[n]$ and $h[n]$ as p and q respectively

Step 4. Compute $N = max(p,q)$

Step 5. By using the inbuilt function `fft`, compute the N point DFTs of the two sequences $x[n]$ and $h[n]$

Step 6. Multiply the DFT coefficients of these two sequences together and take the N-point IDFT of the product sequence using the inbuilt function `ifft`

Step 7. Verify that the sequence obtained in the previous step is the same as the N-point circular convolution of $x[n]$ and $h[n]$ by using the inbuilt function `cconv`

Step 8. Stop

PROGRAM

Part 1: Linear Convolution

```

% Title: Program to
%1)Compute the linear convolution of two finite duration ... sequences by directly evaluating the mathematical expression ... for linear convolution
%2)Compute the linear convolution of two finite duration ... sequences by using the Toeplitz matrix method
%3)Verify the results using the inbuilt function conv

%Author: Sreejesh K V, Dept. of ECE, GCEK
%Date: 23/09/2022

clc;
clear;
close all;

% -- Define the sequences -- %
x=[-3 2 5 4 -1 8];
h=[1 2 -1 3];
p=length(x);
q=length(h);

% -- Length of linear convolution -- %
L=p+q-1;%length of linear convolution sequence

% direct evaluation of linear convolution
```

```
lc=zeros(1,L);
for n=1:L
for m=max([1 n-q+1]):min([n p])
lc(n)=lc(n)+x(m)*h(n-m+1);
end
end
disp('linear convolution result by direct evaluation is');
disp(lc);

%-- Linear convolution using matrix method --- %

%--Forming the row and column vectors for the Toeplitz matrix
c = [h(1) zeros(1, length(x) - 1)];
r = [h zeros(1, L - length(h))];
H=toeplitz(c,r);%dimesions of H: length(x) x L

%--displaying the Toeplitz matrix
disp('The Toeplitz matrix is');
disp(H);

%--verifying the Toeplitz matrix using inbuilt function convmtx
Hc=convmtx(h,length(x));
disp(['The Convolution matrix of h computed using inbuilt ...
    function convmtx for length=' num2str(length(x)) ' is']);
disp(H);

%---perform linear convolution by multiplying row vector x with ...
    the Topelitz matrix of h
lcmat=x*H;
disp('Linear convolution computed using Toeplitz matrix method is');
disp(lcmat);

%--verifying the linear convolution result using inbuilt conv ...
    function
lcon=conv(x,h);
disp('Linear convolution computed using inbuilt conv function is');
disp(lcon);
```

Part 2: Circular Convolution

```
% Title: Program to
%1)Compute the circular convolution of two finite duration ...
    sequences by directly evaluating the mathematical expression ...
    for circular convolution
%2)Compute the circular convolution of two finite duration ...
    sequences by using the Circulant Toeplitz matrix method
%3)Verify the results using the inbuilt function cconv

%Author: Sreejesh K V, Dept. of ECE, GCEK
```

```
%Date: 23/09/2022

clc;
clear;
close all;

% -- Define the sequences -- %
x=[-3 2 5 4 -1 8];
h=[1 2 3 4 5];
p=length(x);
q=length(h);

N = max(p,q);%length of circular convolution sequence

% zero padding to make both sequences of same (N) length
xpad=[x zeros(1,N-p)];
hpad=[h zeros(1,N-q)];

% -- Circular convolution using direct evaluation of expression ...
    -- %
cc = zeros(1,N);
for n =1:N
for m = 1:N
k = mod((n-m),N);
cc(n) = cc(n) + xpad(m)*hpad(k+1);
end
end
disp('Circular convolution result by direct evaluation is');
disp(cc);


%-- Circular convolution using circulant matrix method ---%
% -- Creating the circulant matrix for evaluatiing circular ...
    convolution -- %
c=[hpad(1) fliplr(hpad(2:N))];
r=hpad;
Hcirc=toeplitz(c,r);%The circulant Toeplitz matrix for h of dim NxN

%-- Display the NxN Circulant Topelitz matrix for h
disp('The Circulant Toeplitz matrix is');
disp(Hcirc);

%---perform Circular convolution by multiplying padded vector x ...
    with the Circulant matrix of h
ccmat=xpad*Hcirc;
disp('Circular convolution computed using Toeplitz matrix method ...
    is');
disp(ccmat);

%--verifying the Circular convolution result using inbuilt cconv ...
    function
```

```
ccinbuilt=cconv(x,h,N);
disp('Circular convolution computed using inbuilt cconv function ...
    is');
disp(ccinbuilt);
```

Part 3: Linear Convolution through Circular Convolution

```
% Title: Program to Obtain the linear convolution of two finite ...
    duration sequences through circular convolution

%Author: Sreejesh K V, Dept. of ECE, GCEK
%Date: 23/09/2022

clc;
clear;
close all;

% -- Define the sequences -- %
x=[-3 2 5 4 -1 8];
h=[1 2 3 4 5];

% -- linear convolution using circular convolution -- %
L=length(x)+length(h)-1;
lcusingcc = cconv(x,h,L);
disp('linear convolution result using circular convolution is ');
disp(lcusingcc);

% -- verification using conv -- %
lc=conv(x,h);
disp('linear convolution result using conv is ');
disp(lc);
```

Part 4: Circular Convolution Property of DFT

```
% Title: Program to verify the circular convolution property of DFT

%Author: Sreejesh K V, Dept. of ECE, GCEK
%Date: 23/09/2022

clc;
clear;
close all;

% -- Define the sequences -- %
x=[-3 2 5 4 -1 8];
h=[1 2 3 4 5];
p=length(x);
```

```
q=length(h);

N = max(p,q);%length of circular convolution sequence

% -- Computing the circular convolution using convolution ...
    property of DFT -- %
X=fft(x,N);
H=fft(h,N);
Y=X.*H;
ccfromdft=ifft(Y,N);
disp('IDFT of the products of N-point DFTs of x[n] and h[n]');
disp(ccfromdft);

%--verifying the Circular convolution result using inbuilt cconv ...
    function
ccinbuilt=cconv(x,h,N);
disp('Circular convolution computed using inbuilt cconv function ...
    is');
disp(ccinbuilt);
```

OUTPUT & OBSERVATIONS

Part 1: Linear Convolution

Command Window Output:

```
linear convolution result by direct evaluation is
    -3    -4    12     3     8    17    29   -11    24

The Toeplitz matrix is
     1     2    -1     3     0     0     0     0     0
     0     1     2    -1     3     0     0     0     0
     0     0     1     2    -1     3     0     0     0
     0     0     0     1     2    -1     3     0     0
     0     0     0     0     1     2    -1     3     0
     0     0     0     0     0     1     2    -1     3
```

```
The Convolution matrix of h computed using inbuilt function convmtx for length=6 is
     1     2    -1     3     0     0     0     0     0
     0     1     2    -1     3     0     0     0     0
     0     0     1     2    -1     3     0     0     0
     0     0     0     1     2    -1     3     0     0
     0     0     0     0     1     2    -1     3     0
     0     0     0     0     0     1     2    -1     3

Linear convolution computed using Toeplitz matrix method is
    -3    -4    12     3     8    17    29   -11    24

Linear convolution computed using inbuilt conv function is
    -3    -4    12     3     8    17    29   -11    24
```

Part 2: Circular Convolution

Command Window Output:

```
Circular convolution result by direct evaluation is
    51    36    27    48    15    48

The Circulant Toeplitz matrix is
     1     2     3     4     5     0
     0     1     2     3     4     5
     5     0     1     2     3     4
     4     5     0     1     2     3
     3     4     5     0     1     2
     2     3     4     5     0     1

Circular convolution computed using Toeplitz matrix method is
    51    36    27    48    15    48

Circular convolution computed using inbuilt cconv function is
    51    36    27    48    15    48
```

Part 3: Linear Convolution through Circular Convolution

Command Window Output:

```
linear convolution result using circular convolution is
   -3.0000   -4.0000    0.0000    8.0000   15.0000   48.0000   54.0000   40.0000   27.0000   40.0000

linear convolution result using conv is
    -3    -4     0     8    15    48    54    40    27    40
```

Part 4: Circular Convolution Property of DFT

Command Window Output:

```
IDFT of the products of N-point DFTs of x[n] and h[n]
    51    36    27    48    15    48

Circular convolution computed using inbuilt cconv function is
    51    36    27    48    15    48
```

RESULTS

Part I: Linear Convolution

(a) The linear convolution of two finite duration sequences was computed by directly evaluating the mathematical expression for linear convolution

(b) The linear convolution of two finite duration sequences was computed by using the Toeplitz matrix method

(c) The results were verified using the MATLAB inbuilt function `conv`

Part II: Circular Convolution

(a) The circular convolution of two finite duration sequences was computed by directly evaluating the mathematical expression for circular convolution

(b) The circular convolution of two finite duration sequences was computed by using the Circulant Toeplitz matrix method

(c) The results were verified using the MATLAB inbuilt function `cconv`

Part III: The linear convolution of two finite duration sequences was obtained by means of circular convolution

Part IV: The circular convolution property of DFT was verified