ELC 433

Fall 2017

Lab 4A: Discrete Fourier Transform

Group Members:

Bryan Guner

Stephanie Fournier

Haley Scott

Submitted On:

November 11, 2017

Signal Processing Laboratory (ELC 433)
Lab 4A: Discrete Fourier Transform

Stephanie Fournier*
Bryan Guner*
Haley Scott*

* *Electrical and Computer Engineering Department, The College of New Jersey, (e-mail: fournis1@tcnj.edu gunerb1@tcnj.edu scotth3@tcnj.edu)*

Abstract: In this lab, students were required to use MATLAB to compute and study the frequency domain representation of discrete-time signals. Students will familiarize with this process, called the Discrete Fourier Transform, and the Inverse Discrete Fourier Transform.

---

1. INTRODUCTION

In this laboratory, our team implemented the N-point Discrete Fourier Transform (DFT) in MATLAB. We computed the DFT of two finite-length sequences, and utilized the results to evaluate the linear convolution. The process was to be verified through use of the Inverse Discrete Fourier Transform, by comparing the result of the Inverse Discrete Fourier Transform to the given sequence X[n]. Our team also implemented the L-point DFT, and the L-point IDFT for validation purposes. The results of the L-point DFT were also used to evaluate the linear convolution of the sequences.

2. METHODS

Our team utilized MATLAB in order to compute and plot the N-point DFT (X[k]) of a sequence x[n] of a given length, and subsequently compute the IDFT of X[k]. First, we utilized the sequence x[n] = {0, 1, 2, 3], $0 \leq n \leq 3$. In order to verify our DFT calculation, we computed the Inverse DFT (IDFT) to check that it matched the original sequence, x[n]. Next, we continued on to calculate the L-point DFT (X[k]) of a sequence x[n] of length L, where $L \geq N$. In order to verify this calculation we also computed the IDFT of X[k]. We implemented the 8-point DFT of our original sequence from Figure 1. Finally, our team developed a program to calculate the linear convolution of two sequences, g[n] = [1, 2, 3, 4, 5] where $0 \leq n \leq 4$, and g[n] = [2, 2, 0, 1, 1] where $0 \leq n < 4$.

3. RESULTS

The resulting DFT of the first sequence x[n], as well as the IDFT of X[k]. The sequence x[n] = [0, 1, 2, 3], where 0 < n < 3.

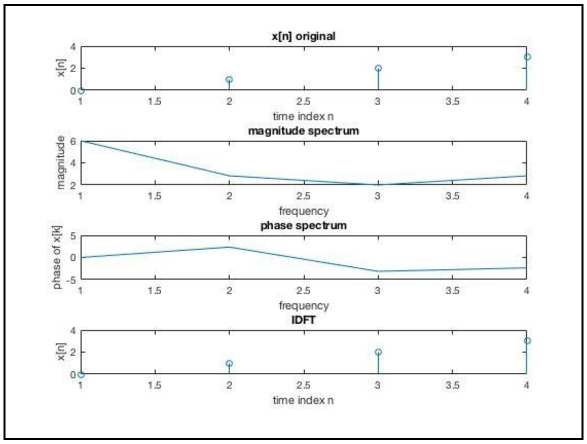

Figure 1: DFT of x[n] and IDFT of X[k]

The resulting DFT of the 8-point zero-padded sequence, x[n] = [1, 2, 3, 4, 0, 0, 0, 0], and the IDFT of the respective X[k].

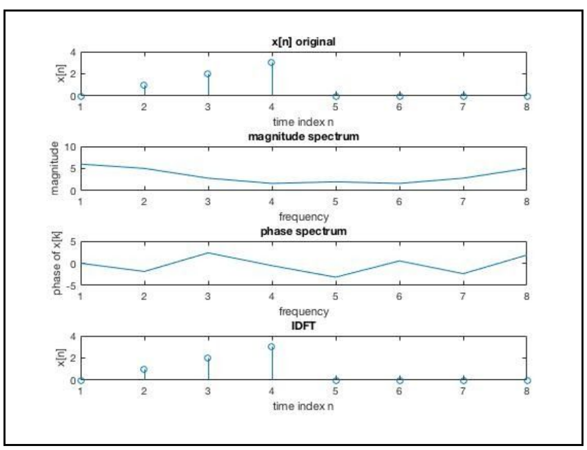

Figure 2: DFT of x[n] and IDFT of X[k]

The linear convolution through the DFT method resulted in the graphical convolution representation shown below in Figure 3. The two convolved sequences were g[n] = [1, 2, 3, 4, 5] $0 \leq n \leq 4$, and g[n] = [2, 2, 0, 1, 1] $0 \leq n \leq 4$.

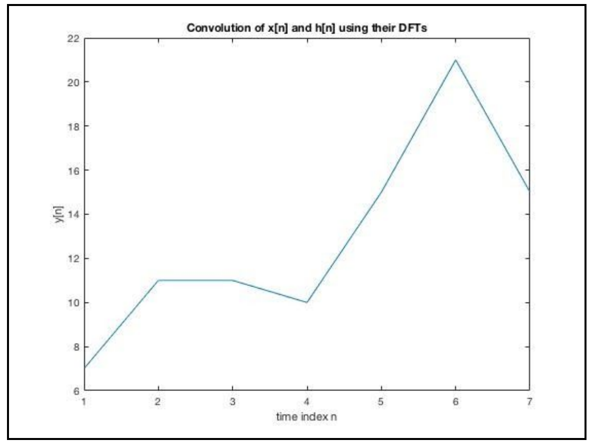

Figure 3: Linear Convolution of Two Sequences via DFT

4. DISCUSSION

Our team computed the results of both the DFT, IDFT, and linear convolution via the DFT. We were able to verify our results through numerous mathematical properties, namely that the IDFT of the DFT should result in the original sequence. We utilized this truth in parts one and two of our experiment, with the N-point DFT and the L-point DFT. Both results proves successful, as the IDFT graph exactly represented the sequence graph.

The DFT method allowed our team to compute the linear convolution of the sequences. We were able to provide an appropriate range of N = 7 in order to multiply the two DFT results for each sequence. Our program results in the Fourier Transform Matrix, and continues to plot the IDFT of the convolved results.

5. CONCLUSION

In this lab, we were successful in the utilization of MATLAB in order to build a code to conduct each step of the lab. We were able to properly compute and plot both the N-point Discrete Fourier Transform and Inverse Fourier Transform. Then we accurately computed the DFT and IDFT of a given sequence and the L-point from the code in the first step. Then we successfully developed a code to compute the linear convolution of two sequences through DFT and then, using this code, for the last step with the given sequences. Overall, this experiment demonstrated the many capabilities of MATLAB in the field of signal processing

6. REFERENCES

Winser, Alexander. Williams, Cranos. (2016). *Digital Signal Processing: Principles, Algorithms and System Design*. Academic Press, Cambridge, MA.

7. APPENDIX

```
xn=[0,1,2,3];
N=4;
for k=1:N
   for n=1:N
   D(k,n)=exp(-1i*(2*pi/N)*(k-1)*(n-1));
   end
end

Xk=D*xn';
Dinv=conj(dftmtx(N))/N;
xr=conj(dftmtx(N))*Xk/N;
subplot(4,1,1);
stem(xn);
title('x[n] original');
xlabel('time index n');
ylabel('x[n]');
subplot(4,1,2);
plot(abs(Xk));
title('magnitude spectrum');
xlabel('frequency');
ylabel('magnitude');
subplot(4,1,3);
plot(angle(Xk));
title('phase spectrum');
xlabel('frequency');
ylabel('phase of x[k]');
subplot(4,1,4);
stem(abs(xr));
title('IDFT');
xlabel('time index n');
ylabel('x[n]');
```

```
xn=[0,1,2,3,0,0,0,0];
N=8;
for k=1:N
   for n=1:N
   D(k,n)=exp(-1i*(2*pi/N)*(k-1)*(n-1));
   end
end

Xk=D*xn';
Dinv=conj(dftmtx(N))/N;
xr=conj(dftmtx(N))*Xk/N;

subplot(4,1,1);
stem(xn);
title('x[n] original');
xlabel('time index n');
ylabel('x[n]');
subplot(4,1,2);
plot(abs(Xk));
title('magnitude spectrum');
xlabel('frequency');
ylabel('magnitude');
subplot(4,1,3);
plot(angle(Xk));
title('phase spectrum');
xlabel('frequency');
ylabel('phase of x[k]');
subplot(4,1,4);
stem(abs(xr));
title('IDFT');
xlabel('time index n');
ylabel('x[n]');
```

```
xn=[0,1,2,3,4,5,0,0];
hn=[2,2,0,1,1,0,0,0];
N=7;
Xk=fft(xn,N);
Hk=fft(hn,N);
```

```
Yk=Xk.*Hk;
%fourier transform matrix
yn=ifft(Yk);

plot(yn);
title('Convolution of x[n] and h[n] using their DFTs');
xlabel('time index n');
ylabel('y[n]');
```