임베디드
신호처리 실습

Convolution 실습 결과보고서

전자공학부 임베디드시스템

2014146004 김 민 섭

MATLAB

1-1 DFT 식을 기반으로 하여 N-Point DFT를 계산하는 함수를 작성하라.

DFT 식은 다음과 같다.

$$X_k = \sum_{n=0}^{N-1} x[n]e^{-j2\pi(k/N)n}, k = 0,1,2...N-1$$

입력과 출력은 다음과 같다.

입력

x : 이산신호 x[n] , 신호의 길이는 N, n = 0,1,2 .... N-1

출력

f_hat : 이산주파수 f^ , f^ = 0, 1/N , 2/N, …. (N-1)/N
Xk : 스펙트럼 Xk, 복소수이며 길이는 N
N_mult : 곱셈연산의 횟수

DFT를 MATLAB으로 구현하기란 정말 쉽다. 위에 있는 식을 그대로 MATLAB의 문법에 적용시켜 주면 끝나게 된다. 추가로 진행될 실습에서 이산주파수가 -1/2 ~ 1/2 에 대하여 그려야 하니 이를 적용시키는 것만 실행되면 된다. 이는 DFT가 주기적인 성질을 가지고 있어 위 부분만 알면 전체를 알 수 있기 때문이다. 이제부터 코드를 보면서 어떻게 구현했는지 알아보도록 하자.

```
function [f_hat , Xk,N_mult] = myfun_N_Point_DFT(x)

x_len = length(x);
N_to_Sum = 0 : 1:(x_len-1);
Xk_mid = [zeros(1,x_len)];
f_mid =  [zeros(1,x_len)];
```

먼저 초기 설정부터 알아보도록 하자. 입력으로 들어온 x 값의 길이를 알아내었다. 이를 통해 신호의 길이인 N을 알아 낼 수 있었다. n은 0 ~ N-1 이므로 ’ N_to_Sum = 0 : 1:(x_len-1); ’으로 구현해 주었다. 이는 매틀랩의 배열이 여타 다른 프로그래밍 언어들과는 달리 ‘0’번째 행렬이 없고 ‘1’번째 행렬부터 시작되기 때문이다. 뒤에 있는 mid d값들은 중간값으로 연산을 보다 확실히 하고자 넣어주었다.

```
for k=1 : x_len
    for n=1:x_len
    Xk_mid(k) =Xk_mid(k) + x(n)*exp(-1i*2*pi*(N_to_Sum(k)/ x_len)*N_to_Sum(n));
    end

    if k < (x_len/2 +1)
    f_mid(k) = N_to_Sum(k)/x_len;
    else
    f_mid(k) = N_to_Sum(k)/x_len-1;
    end
end
```

 이 부분은 실제로 DFT를 구현해주는 과정이다. For 문 안의 첫번째 for문은 DFT식을 메틀랩 함수로 표현한 것이다. 시그마 하는 과정으로 이중 포문으로 구현하였다.
아래 부분에 있는 if 문은 이산주파수의 범위를 설정하기 위해서 넣어준 부분이다. 이 부분을 좀 해석해 보도록 하겠다. 간단하게 전체 길이의 절반 이전까지는 그대로 이산주파수의 값을 넣어주고 그 이후부터는 -1 을 해줌으로 평행이동 시켜주었다.

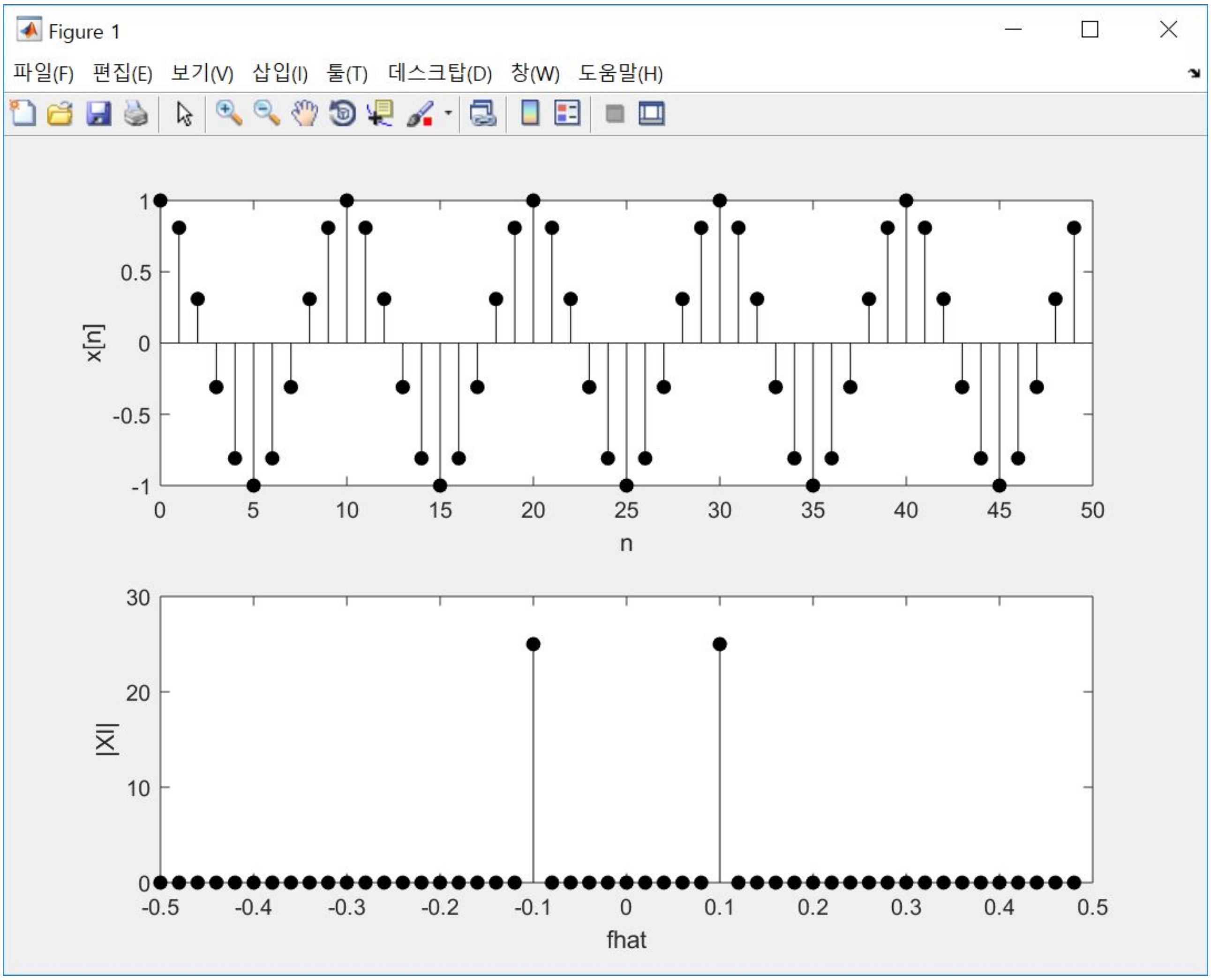
```
f_hat = f_mid;
Xk = Xk_mid;
N_mult =(x_len).^2
```

마지막 코드 부분이다. Mid 값들은 그대로 출력값으로 출력되고 총 곱셈연산한 횟수는 N의 제곱 번이다. 제곱번인 이유는 코드를 보면 알 수 있듯이 이중 포문으로 한번 돌때 N번 도는 것이 N번 돌기 때문이다.
전체적인 코드는 다음과 같다.

```
function [f_hat , Xk,N_mult] = myfun_N_Point_DFT(x)
x_len = length(x);
N_to_Sum = 0 : 1:(x_len-1);
Xk_mid = [zeros(1,x_len)];
f_mid =  [zeros(1,x_len)];

for k=1 : x_len
    for n=1:x_len
    Xk_mid(k) =Xk_mid(k) + x(n)*exp(-1i*2*pi*(N_to_Sum(k)/ x_len)*N_to_Sum(n));
    end

    if k < (x_len/2 +1)
    f_mid(k) = N_to_Sum(k)/x_len;
    else
    f_mid(k) = N_to_Sum(k)/x_len-1;
    end
end

f_hat = f_mid;
Xk = Xk_mid;
N_mult =(x_len).^2
```

1-2 위에서 구현한 N-Point DFT를 이용해 다음 이산신호 x[n]의 크기 스펙트럼을 구하고 그래프에 표시하여라.

- $x[n] = cos(2\pi\hat{f}_0 n), n = 0,1,2...,N-1$
- $\hat{f}_0 = 0.1, N = 50$



위 신호에 대해서 우리가 만든 DFT 함수를 통해서 변환시켜보니 다음과 같은 결과가 나오게 되었다. $x[n] = cos(2\pi\hat{f}_0 n)$ 에 대해서 푸리에 변환을 해보면 다음과 같은 식이 나오게 된다. $X(f) = 1/2[\delta(f-\hat{f}_0) + \delta(f+\hat{f}_0)]$ 이는 연속 함수인 cos 인 식에 대한 푸리에 변환이다. 그러나 이를 표본화 시킨 것 처럼 잘개 쪼갠 것이 DFT이니 이를 참고하여 식을 그려보면 -0.1 과 0.1에서 값을 가지는게 된다. 이외의 값에는 모두 ‘0’ 의 값을 가지게 된다. 그리고 0.1과 -0.1 에서 N/2 만큼인 값인 25만큼의 값이 찍히게 된다.

1-3 위 신호를 DFT 하는데 필요한 곱셈 연산의 횟수는 얼마인가? 손으로 계산한 값과 실험을 통해 측정한 값을 비교하고 이유를 설명하라.

먼저 손으로 계산한 DFT의 연산량은 다음과 같다.

$$X_K = \sum_{n=0}^{N-1} x[n] e^{-j2\pi(\frac{k}{N})n}, \quad K=0,1,2 \cdots N-1$$

$$X[0] = \sum_{n=0}^{N-1} x[n] = x[0] + x[1] + \cdots x[N-1]$$

$$X[1] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}n} = x[0] + x[1]e^{-j\frac{2\pi}{N}} + x[2]e^{-j\frac{4\pi}{N}} + \cdots x[N-1]e^{-j\frac{2\pi}{N}(N-1)}$$

$$\vdots$$

$$X[N-1] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2(N-1)\pi}{N}n} = x[0] + x[1]e^{-j\frac{2(N-1)}{N}\pi} + \cdots x[N-1]e^{-j\frac{(N-1)^2}{N}2\pi}$$

$$\begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{-j\frac{2\pi}{N}} & e^{-j\frac{4\pi}{N}} & & e^{-j\frac{(N-1)}{N}2\pi} \\ \vdots & & & & \\ 1 & e^{-j\frac{(N-1)}{N}\pi} & e^{-j\frac{(N-1)}{N}2\pi} & \cdots & e^{-j\frac{(N-1)^2}{N}2\pi} \end{bmatrix}}_{N개} \left.\begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}\right\} N개$$

총 N^2번의 곱셈이 필요하다.

위와 같은 방식을 통한 결과 총 N^2 번의 곱셈 연산이 필요하다는 사실을 알 수 있었다.

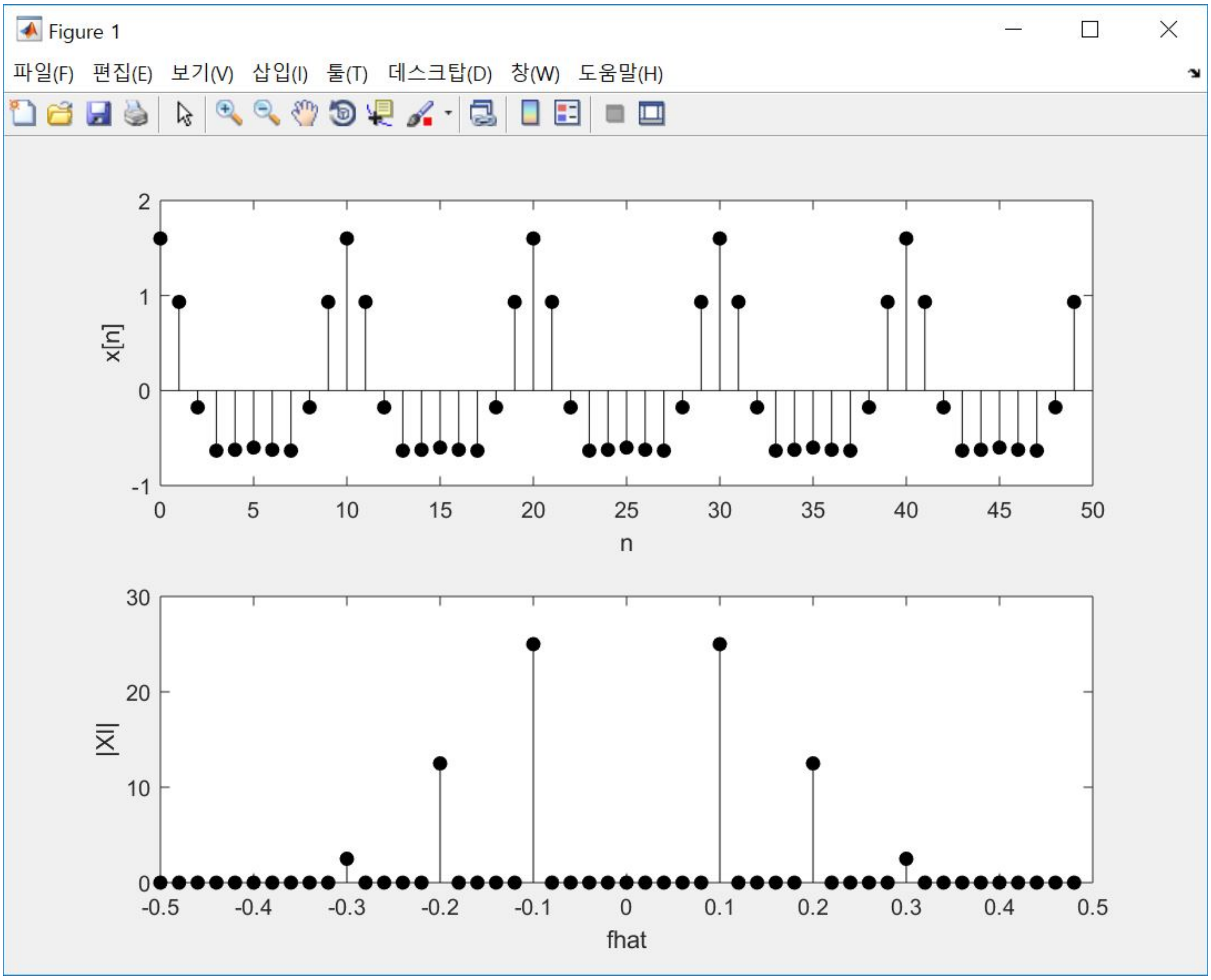
```
for k=1 : x_len
   for n=1:x_len
   Xk_mid(k) =Xk_mid(k) + x(n)*exp(-1i*2*pi*(N_to_Sum(k)/ x_len)*N_to_Sum(n));
   end
   end
end
```

실험에 사용된 곱셈 연산과 관련된 코드 부분이다. x_len이 N이고 이중 포문으로 구성되어 손으로 계산한 것과 마찬가지로 N^2번의 곱셈 연산이 나오게 된다. 이것으로 DFT연산시에 필요한 곱셈 연산 횟수는 N^2 이라는 사실을 알 수 있었다.

2-1 다음 이산신호 x[n]을 N - Point DFT하여 주파수 성분을 분석하라.

- $x[n] = x_1[n] + x_2[n] + x_3[n], n = 0,1,2...,N-1$
- $x_1[n] = cos(2\pi\hat{f}_1 n), \hat{f}_1 = 0.1$
- $x_2[n] = 0.5cos(2\pi\hat{f}_2 n), \hat{f}_2 = 0.2$
- $x_3[n] = 0.1cos(2\pi\hat{f}_3 n), \hat{f}_3 = 0.3$

위 식을 MATLAB을 통하여 그려보면 다음과 같이 그려지게 된다.



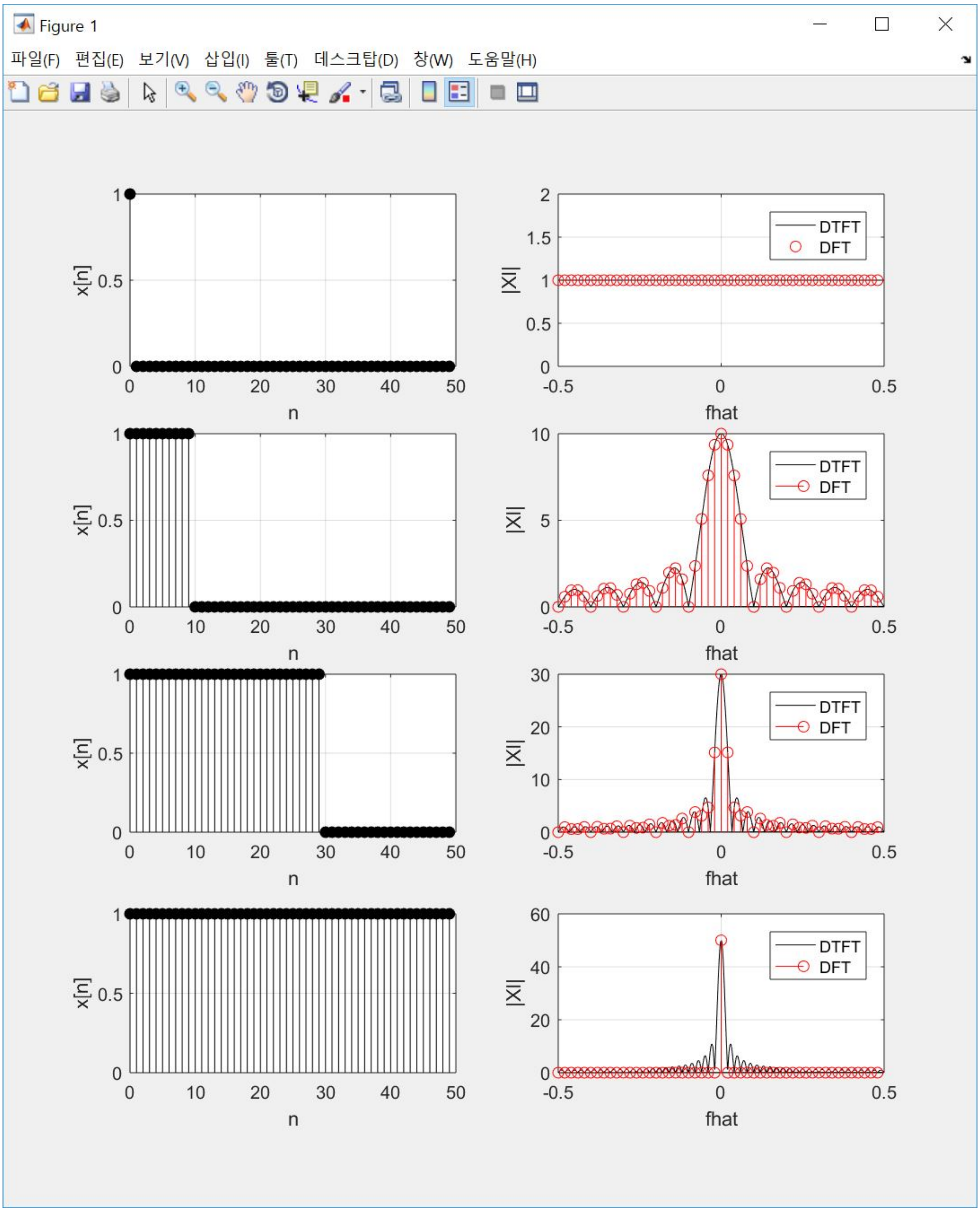
정형파의 합은 또 다른 정형파로 표현될 수 있으며 이는 x[n] 이 정형파의 모습을 띄게 한다. 주파수 영역으로. 가져왔을 때는 더해지기 전 각각의 이산화 주파수 성분들에서 값이 나오게 되며 이 성분들의 크기는 이산화 하기 전의 정형파만큼 곱혀져서 나오게 된다. 그 결과 위와 같은 그래프가 그려 질 수 있었다.

2-2 다음 파라메터에 대해 식(6)에 정의된 신호를 N-Point DFT 하여 크기 크기 스펙트럼을 구하고 그래프에 표시하여라.

$$p_L[n] = \begin{cases} 1 & , n = 0, 1, 2, \cdots, L-1 \\ 0 & , n = L, L+1, \cdots, N-1 \end{cases} \quad , N \geq L \geq 0 \qquad (6)$$

1. L = 1,   N = 50
2. L = 10, N = 50
3. L = 30, N = 50
4. L = 50, N = 50

위 파라미터와 식에 의해서 그려지는 그래프는 다음과 같다.



위 식과 그래프를 분석하여 보니 L =1 일때는 임펄스 함수를 의미하고 이후 부터는 구형파를 의미한다. 이로 인해 임펄스 함수를 Fourier transform 한 1의 값이 DFT에 나오게 되었다. 나머지 구형파들에서는 각각 주기에 따라서 sinc 함수가 잘 그려지고 있다. 다만 제로페딩이 전혀 이루어지지 않은 L = 50 인 영역에서는 x[n] = 1 인것이 푸리에 변환된 모습이다. 그래서 DFT를 실시하였을때에 0 에서만 50이라는 값을 가지는 임펄스 함수 같은 모습이 되었다.

2-3 N-Point DFT 결과를 손으로 계산한 DTFT와 비교하고 본 실습의 의의를 설명하라.

$$x_1[n] = \delta[n] \qquad \mathcal{F}\{x_1[n]\} = 1$$

$$x_2[n] = p_{10}[n] \qquad \mathcal{F}\{x_2[n]\} = 10\,\mathrm{sinc}\left(\frac{10 \cdot 2\pi f_0}{2\pi}\right) = 10\,\mathrm{sinc}\,10 f_0$$

$$x_3[n] = p_{30}[n] \qquad \mathcal{F}\{x_3[n]\} = 30\,\mathrm{sinc}\,30 f_0$$

$$x_4[n] = p_{50}[n] \qquad \mathcal{F}\{x_4[n]\} = 50\,\mathrm{sinc}\,50 f_0$$

2-2의 식을 손으로 구한 DTFT이다. 푸리에 변환 표를 활용하여 구하였다. DTFT와 DFT의 가장 큰 차이점은 연속과 이산이다. 두 변환 방식 모두 이산신호를 입력으로 받는다. 하지만 DTFT는 결과값이 연속으로 나온다. 반면 DFT는 이산값으로 나오고 있다. 이것이 가장 큰 첫번째 차이점이다.
두번째로 큰 차이점은 제로페딩의 중요성 유무이다. DTFT는 제로페딩이 되어있든 안되어있든 상관없이 제대로된 결과값을 뽑아 낸다. 반면 DFT의 경우 제로페딩이 아에 안되어 있을 경우 의도했던 결과값이 안나올 수 있다. 2-2의 4번 식을 바라보자 1의 값이 50개이고 50-point DFT이다. 즉 제로패딩이 되어 있지 않다. 이는 구형파를 DFT 한 결과라기 보다는 ‘1’을 DFT 한 결과처럼 나오게 된다.
반면 DTFT는 같은 식을 변환하여도 구형파로 인식하고 결과 값인 sinc 함수를 잘 출력하였다.
추가로 DFT 할적에 너무 많은 제로패딩을 하게되면 연산량이 많아지게 됨으로 적절한 제로패딩이 필요하다. 이로서 본 실습의 의의는 두가지로 정리할 수 있다.

- DFT 와 DTFT의 결과값은 각각 이산값과 연속적인 값으로 출력되는 것을 알 수 있다.
- DFT 를 실시할때는 적절한 제로패딩이 필요하다.