REPORT

DFT Program

과목명	신호및시스템
담당교수	이석필 교수님
학과	융합전자공학과
학년	2학년
학번	201910906
이름	이학민
제출일	2020.06.10

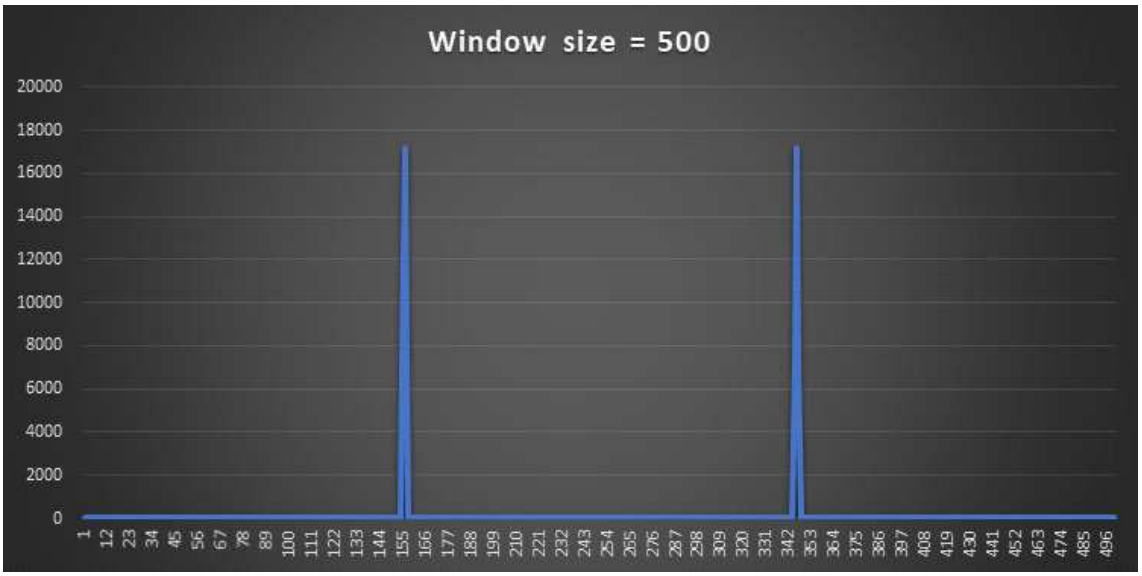
상명대학교
SANGMYUNG UNIVERSITY

1. Source Program

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define PI 3.141592653589793238

typedef struct complex{
    double Re, Im;
}COMPLEX;

int main(void){
    FILE *fp;
    fp = fopen("sample1.snd", "rb");
    if(fp == NULL){
        fprintf(stderr, "sample1.snd 파일을 열 수 없습니다.\n");
        return 1;
    }               // sample1.snd 파일을 열 수 없을 때 에러 메시지 출력 후 종료.

    int WINDOW_SIZE = 0;
    printf("Window size를 입력하시오 : ");
    scanf("%d", &WINDOW_SIZE);

    COMPLEX* complex;
    int k=0,n=0;
    char original_file[WINDOW_SIZE];                        // signed 8bit 파일이므로 1byte크기를
    double* mul_window = NULL;
    int choose_window = 0;
    double spectrum_size[WINDOW_SIZE];
    int max_k = 0;
    double max_freq = 0;

    mul_window = (double*)malloc(sizeof(double) * WINDOW_SIZE);
    complex = (COMPLEX*)malloc(sizeof(COMPLEX) * WINDOW_SIZE);      // 동적 할당.

    printf("어떤 Window를 씌울지 고르시오(Hamming Window : 1 , Rectangular Window : 2) : ");
    scanf("%d", &choose_window);

    switch(choose_window)
    {
        case 1 :                // Hamming Window 적용
            printf("\nHamming Window를 사용합니다.\n");
            for(n=0; n<WINDOW_SIZE; n++){
                fread(original_file , sizeof(char), WINDOW_SIZE, fp);       // sample1.snd 파일의
                mul_window[n] = (double)original_file[n] * (0.54-0.46*cos((2*PI*n)/WINDOW_SIZE));
                complex[n].Re = 0;                      // 실수부 초기화
                complex[n].Im = 0;                      // 허수부 초기화
            }
            break;

        case 2 :                // Rectangular Window 적용
            printf("\nRectangular Window를 사용합니다.\n");
            for(n=0; n<WINDOW_SIZE; n++){
                fread(original_file , sizeof(char), WINDOW_SIZE, fp);       // sample1.snd 파일의
                mul_window[n] = (double)original_file[n] * 1;       // 원래 값에 rectangular windo
                complex[n].Re = 0;                      // 실수부 초기화
                complex[n].Im = 0;                      // 허수부 초기화
            }
            break;

        default :
            printf("\n잘못 입력하였습니다. 프로그램을 종료합니다.\n");
            return 2;
    }
```

```
63
64
65      for(k=0; k<WINDOW_SIZE; k++)
66          for(n=0; n<WINDOW_SIZE; n++){
67              complex[k].Re += mul_window[n] * (cos((2*PI*k*n)/WINDOW_SIZE));                  // cos
68              complex[k].Im += mul_window[n] * (-(sin((2*PI*k*n)/WINDOW_SIZE)));               // sin
69          }
70
71      for(k=0; k<WINDOW_SIZE; k++)
72          spectrum_size[k] = sqrt(pow(complex[k].Re, 2) + pow(complex[k].Im, 2));             // 스펙
73
74      for(k=0; k<(WINDOW_SIZE)/2; k++)                // 대칭이므로 전체 WINDOW_SIZE에서 절반만 확인
75          if(spectrum_size[k]>max_freq){
76              max_freq = spectrum_size[k];
77              max_k = k;
78          }
79
80      max_freq = 4000*(max_k/(0.5*WINDOW_SIZE));                  // 최대 크기를 가지는 주파수 값을
81
82      printf("\n%d points DFT 수행\n", WINDOW_SIZE);
83      printf("\n최대 크기를 가지는 주파수 값 : %0.2lf\n\n\n", max_freq);
84      printf("\n      k                              X[k]                                    |X[k]|\n");
85      printf("\n   ==========================================================================================\n\n");
86
87      for(k=0; k<WINDOW_SIZE; k++)
88          printf("   %4d  %lf+j%lf %lf\n", k, complex[k].Re, complex[k].Im, spectrum_size[k]);
89
90      fclose(fp);
91      free(mul_window);
92      free(complex);
93
94      return 0;
95  }
```

2. Spectrum (Hamming Window 적용)



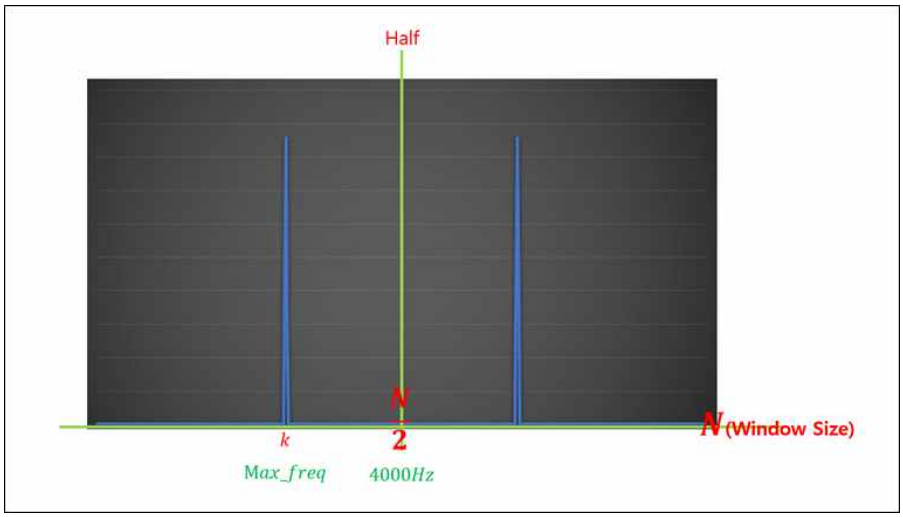
3. Maximum Frequency

```
max_freq = 4000*(max_k/(0.5*WINDOW_SIZE));          // 최대 크기를 가지는 주파수 값을 구하는 식
```

```
C:₩Users₩hakmi₩OneDrive₩바탕 화면₩2020₩2-1₩전공₩신호및시스템₩과제₩window₩window.exe
Window size를 입력하시오 : 500
어떤 Window를 씌울지 고르시오(Hamming Window : 1 , Rectangular Window : 2) : 1

Hamming Window를 사용합니다.

500 points DFT 수행

최대 크기를 가지는 주파수 값 : 2480.00 (Hz)


     k                          X[k]                              |X[k]|
  ==========================================================================
```

Window Size(500, 1000, 2000)와 Window의 종류(Hamming, Rectangular)에 상관없이 Maximum Frequency는 2480.00 (Hz)의 값을 갖는다.

max_freq를 구하는 식은 다음과 같은 과정을 통해 얻었다.



$$k : \frac{N}{2} = Max\_freq : 4000$$

$$Max\_freq = \frac{4000k}{\frac{N}{2}}$$

이 때, $k = Max_k$이고, $N = Window\ Size$ 이므로

$$Max\_freq = 4000 * \frac{Max\_k}{0.5Window\_Size}$$ 가 된다.