EDDI
Electronic Design
Development Institute

에디로봇아카데미

임베디드 마스터 Lv2 과정

제 1기

2022. 05. 14

손표훈

CONTENTS

- DiscreteFourierTransform
  - 예제를 통한 DFT 수식 해석
  - DFT C code 구현

- 히스토그램(Box-Muller변환에 대해 좀 더 공부하고 작성 예정..)
  - 균등분포 히스토그램
  - Box-Muller 변환을 이용한 가우시안 분포 난수 생성
  - 가우시안 분포 히스토그램

Discrete Fourier Transform

➢ 예제를 통한 DFT 수식 해석

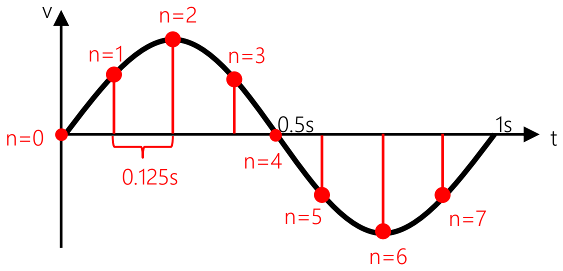
1. DFT 정의

$$X[k]=\sum_{n=0}^{N-1}x[n]e^{\frac{-j2\pi kn}{N}}\ (0\le k\le N-1)$$

→ N은 샘플된 신호의 총 개수
→ x[n]은 샘플된 신호의 크기
→ k는 주파수 스펙트럼의 인덱스 값이다
$F_k=\frac{kF_s}{N}$ (F_k: DFT 결과 주파수 축, F_s : 샘플링 주파수)

→ $e^{\frac{-j2\pi kn}{N}}$는 오일러 공식에 의해 다음과 같다
→ $e^{\frac{-j2\pi kn}{N}}=\cos\frac{2\pi kn}{N}-j\sin\frac{2\pi kn}{N}$

2. 예제 : 크기가 1이고, 1Hz의 사인파 신호를 8Hz의 샘플링을 통해 8점 DFT를 한다



$x[0]=0$
$x[1]=\sin(2\pi*1/8)$=0.707
$x[2]=\sin(2\pi*2/8)$=1
$x[3]=\sin(2\pi*3/8)$=0.707
$x[4]=\sin(2\pi*4/8)$ = 0
$x[5]=\sin(2\pi*5/8)$=-0.707
$x[6]=\sin(2\pi*6/8)$=-1
$x[7]=\sin(2\pi*7/8)$=-0.707

→ 1Hz의 신호에서 8개의 sample을 얻을 수 있다

→ 0.125s 마다 1개의 샘플을 얻는다

→ 각 주파수 스펙트럼 인덱스에 해당하는 DFT를 하면

$$X[0]=x[0]*1+x[1]*1+x[2]*1+\cdots+x[7]*1=0$$

$$X[1]=x[0]*\{\cos\left(\frac{2\pi*1*0}{8}\right)-j\sin(\frac{2\pi*1*0}{8})\}+x[1]*\{\cos\left(\frac{2\pi*1*1}{8}\right)-j\sin(\frac{2\pi*1*1}{8})\}$$
$$+x[2]*\{\cos\left(\frac{2\pi*1*2}{8}\right)-j\sin(\frac{2\pi*1*2}{8})\}+\cdots x[7]*\{\cos\left(\frac{2\pi*1*7}{8}\right)-j\sin(\frac{2\pi*1*7}{8})\}=-4j$$

$$X[2]=x[0]*\{\cos\left(\frac{2\pi*2*0}{8}\right)-j\sin(\frac{2\pi*2*0}{8})\}+x[1]*\{\cos\left(\frac{2\pi*2*1}{8}\right)-j\sin(\frac{2\pi*2*1}{8})\}$$
$$+x[2]*\{\cos\left(\frac{2\pi*2*2}{8}\right)-j\sin(\frac{2\pi*2*2}{8})\}+\cdots x[7]*\{\cos\left(\frac{2\pi*2*7}{8}\right)-j\sin(\frac{2\pi*2*7}{8})\}=0$$

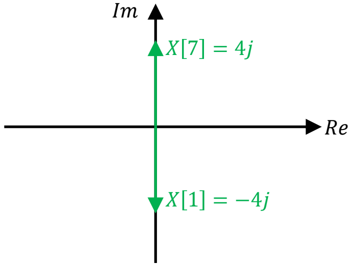
$$\vdots$$

$$X[7]=x[0]*\{\cos\left(\frac{2\pi*7*0}{8}\right)-j\sin(\frac{2\pi*7*0}{8})\}+x[1]*\{\cos\left(\frac{2\pi*7*1}{8}\right)-j\sin(\frac{2\pi*7*1}{8})\}$$
$$+x[2]*\{\cos\left(\frac{2\pi*7*2}{8}\right)-j\sin(\frac{2\pi*7*2}{8})\}+\cdots x[7]*\{\cos\left(\frac{2\pi*7*7}{8}\right)-j\sin(\frac{2\pi*7*7}{8})\}=4j$$

Discrete Fourier Transform

→ DFT의 결과를 복소평면상에서 표현하면 다음과 같다

$X[0] = 0$
$X[1] = 0 - 4j$
$X[2] = 0$
$X[3] = 0$
$X[4] = 0$
$X[5] = 0$
$X[6] = 0$
$X[7] = 0 + 4j$



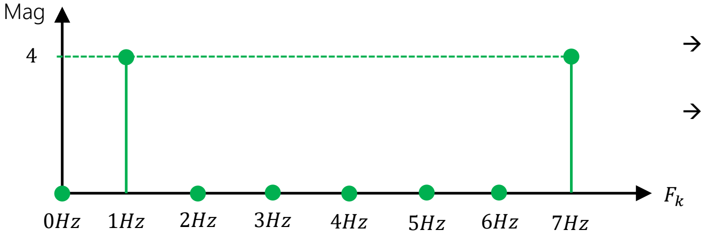
→ $X[1] = 0 - 4j$

복소평면상에서 크기 = $\sqrt{16} = 4$

→ $X[1] = 0 - 4j$

복소평면상에서 크기 = $\sqrt{16} = 4$

→ DFT의 주파수 스펙트럼을 그리면 다음과 같다



→ k는 주파수 스펙트럼의 인덱스 값이다

$F_k = \frac{kF_s}{N}$ (F_k: DFT 결과 주파수 축, F_s : 샘플링 주파수)

→ $F_k = \frac{k8}{8}$로 즉 DFT의 주기는 k1Hz가 된다

→ Q1. 자료를 찾다 보니 위 스펙트럼 그래프가 공액복소수라 대칭성은 이해가 갔는데 왜 magnitude가 double plot에서 single plot으로 변할 때 2배가 되는지 이해가 안됩니다

→ Q2. 4Hz 지점이 Fk가 Fs/2로 나이퀴스트 샘플링 주파수

Discrete Fourier Transform

➤ DFT C code 구현

```c
void draw_spectrum(void)
{
    float x = 0, x2 = 0, y2, cx, cy;
    float t, step = 0.0;
    int i, j, cnt = 0, cache = 0;
    float period, freq = 100.0;
    float res_real[32] = {0};
    float res_image[32] = {0};
    float y[32] = {0};
    c exp = {0};

    calc_period(&freq, &period);
    step = get_step(SLICE, period);

    for(i = 0; i < SLICE; i++)
    {
        for(j = 0; j < SLICE; j++)
        {
            exp.cosx[i][j] = cos(-(2 * M_PI * j * i) / SLICE);
            exp.isinx[i][j] = sin(-(2 * M_PI * j * i) / SLICE);
            printf("exp.cosx[%d][%d] = %f\n", i, j, exp.cosx[i][j]);
            //printf("exp.isinx[%d][%d] = %f\n", i, j, exp.isinx[i][j])
        }
    }
```

```c
void calc_period(float *freq, float *period)
{
    *period = 1 / (*freq);
}

void calc_angular_velocity(float *freq, float *ang_vel)
{
    *ang_vel = 2 * M_PI * (*freq);
}

float get_step(float slice, float period)
{
    return period / slice;
}
```

```c
    i = 0;
    t = 0.0;
    for(; i < SLICE; t += step)
    {
        //if(t > 3 * period)
        if(t > period)
        {
            break;
            t = 0.0;
        }

        y[i] = 10 * cos(200 * M_PI * t);
        printf("y[%d] = %f\n", i++, y[i]);
        //printf("exp.cosx[%d] = %f\n", i, exp.cosx[i]);
        //printf("exp.isinx[%d] = %f\n", i, exp.isinx[i]);
        //printf("res_real[%d] = %f\n", i, res_real[i]);
        //printf("res_real = %f\n", res_real);
        //printf("res_image = %f\n", res_image);
    }

    for(i = 0; i < SLICE; i++)
        for(j = 0; j < SLICE; j++)
        {
            res_real[i] += y[j] * exp.cosx[i][j];
            res_image[i] += y[j] * exp.isinx[i][j];
            //printf("res_real[%d] = %f\n", i, res_real[i]);
            printf("res_image[%d] = %f\n", i, res_image[i]);
        }

    //printf("OK");
```

```c
    for(i = 0; i < SLICE; i++)
    {
        glBegin(GL_POINTS);
        glVertex2f(i * 10, res_real[i] * 3);
        //glVertex2f(i * 10, res_image[i] * 3);
        glEnd();

        glBegin(GL_LINE_STRIP);
        glVertex2f(i * 10, res_real[i] * 3);
        glVertex2f(i * 10, 0);
        //glVertex2f(i * 10, res_image[i] * 3);
        //glVertex2f(i * 10, 0);
        glEnd();
    }
```

→ $X[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j2\pi kn}{N}}\ (0 \le k \le N-1)$에서 복소 함수부분을 오일러 공식을 이용하여 구한다

→ 입력 신호를 샘플링 한다. 샘플링 주파수는 $\frac{1}{\frac{period}{slice}} = 400Hz$ = 2.5ms 주기로 크기가 10이고 주파수가 100인 신호를 샘플링 한다.. 샘플링이 400Hz이고 4-point DFT를 수행하므로 주파수 스펙트럼은 100Hz 주기로 나타난다

→ $X[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j2\pi kn}{N}}$의 최종 DFT 연산을 한다

→ Q. 왜 실수부만 그래프를 그리는지 궁금합니다