디지털 신호 처리 HW#2 Guide

Kyung Hee Univ.
HanJun Lee
2025.11.26.

Contents

❖ Example Guide

- 1D Z-transform(예시)
- 1D FIR filter(예시)

❖ Assignment Guide

- [1번 문제] : Compute the Z-transform of the finite-length signal $x[n] = a^n \cdot u[n]$ (n = 0, 1, … , 50), and compare it with the closed-form expression $X(z) = \frac{1}{1-a\cdot z^{-1}}$(과제)
- [2번 문제] : Apply an FIR Filter to a Grayscale Image and Observe How the Image Changes (과제)

❖ Submission Guide

❖ 1D Z-transform (Example)

Fourier Transform

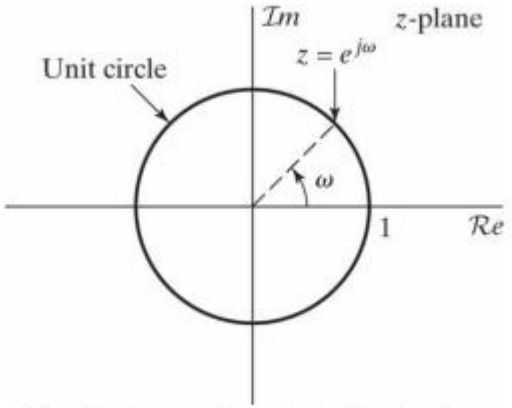
$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

$$z = e^{j\omega}$$

Z-transform

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

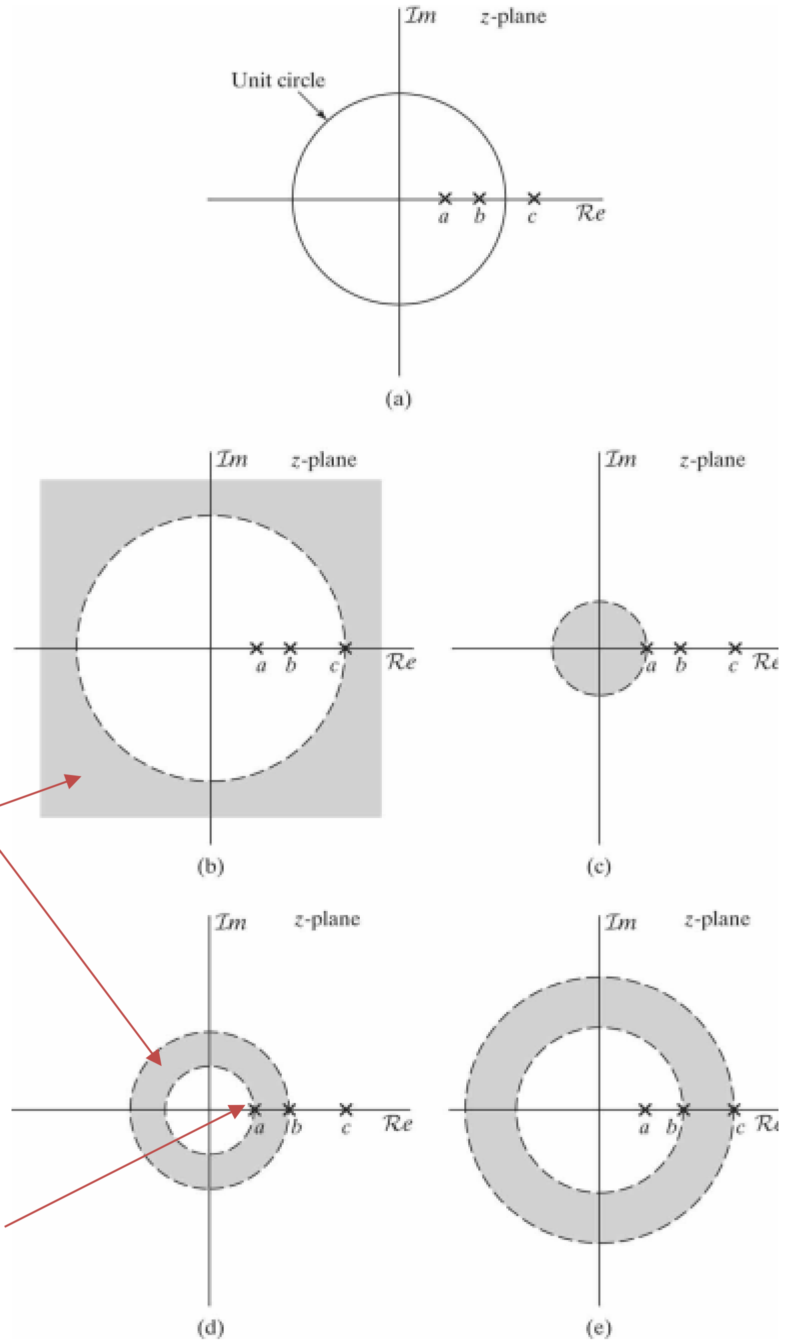
$$z = r \cdot e^{j\omega}$$



The inherent periodicity in frequency of the FT is captured

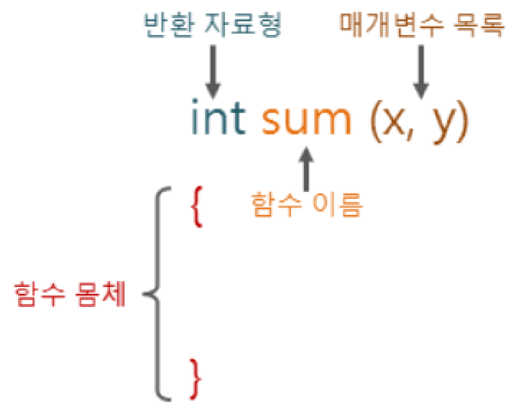
ROC (Region of Convergence)

Poles



❖ 1D Z-transform (Example)

- function




```cpp
#include <iostream>
#include <vector>
#include <complex>
#include <cmath>
#define PI 3.141592

using namespace std;

complex<double> zTransform1D(const vector<double>& x, complex<double> z)
{
    complex<double> X(0.0, 0.0); // X(z)=0 + j0 로 초기화

    complex<double> z_pow(1.0, 0.0); // 처음에는 z^{-0}=1

    // n=0, 1, ... , x.size()-1 까지 반복
    for (size_t n = 0; n < x.size(); ++n) {
        X += x[n] * z_pow; // x[n]*z^{-n}
        z_pow /= z; // z_pow = z_pow / z;
    }

    return X;
}

void printComplex(const complex<double>& z)
{
    double a = z.real();
    double b = z.imag();

    cout << a;

    if (b < 0) {
        cout << " - j" << -b;
    }
    else {
        cout << " + j" << b;
    }
}
```

Compile시 실제 동작

```cpp
int main()
{
    // z = r * e^{jw}
    double r = 0.9;
    double w = PI / 4.0; // π/4

    complex<double> z = r * complex<double>(cos(w), sin(w));

    // 예제 1 신호: x[n] = δ[n] = {1.0}
    vector<double> x_delta = { 1.0 };
    complex<double> X_delta = zTransform1D(x_delta, z);

    cout << "예제 1 : X(z) = "; printComplex(X_delta);
    cout << endl << endl;

    // 예제 2 신호: x[n] = u[n], 길이 5
    vector<double> x_step = { 1.0, 1.0, 1.0, 1.0, 1.0 };
    complex<double> X_step = zTransform1D(x_step, z);

    cout << "예제 2 : X(z) = "; printComplex(X_step);
    cout << endl << endl;

    return 0;
}
```

❖ 1D Z-transform (Example)

```cpp
#include <iostream>
#include <vector>
#include <complex>
#include <cmath>
#define PI 3.141592

using namespace std;

complex<double> zTransform1D(const vector<double>& x, complex<double> z)
{
    complex<double> X(0.0, 0.0); // X(z)=0 + j0 로 초기화

    complex<double> z_pow(1.0, 0.0); // 처음에는 z^{-0}=1

    // n=0, 1, ... , x.size()-1 까지 반복
    for (size_t n = 0; n < x.size(); ++n) {
        X += x[n] * z_pow; // x[n]*z^{-n}
        z_pow /= z; // z_pow = z_pow / z;
    }

    return X;
}

void printComplex(const complex<double>& z)
{
    double a = z.real();
    double b = z.imag();

    cout << a;

    if (b < 0) {
        cout << " - j" << -b;
    }
    else {
        cout << " + j" << b;
    }
}
```

1. PI = 3.141592 define

2. zTransform1D function
   - Input : vector $x[n]$, complex z
   - Output : $Z\{x[n]\}$
   - $X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$

     $= x[0]z^0 + x[1]z^{-1} + x[2]z^{-2} + \cdots$

3. printComplex function
   - $X(z) = Real + j\,Imag$꼴로 출력하기 위한 함수

❖ 1D Z-transform (Example)

```cpp
int main()
{
    // z = r * e^{jw}
    double r = 0.9;
    double w = PI / 4.0; // π/4

    complex<double> z = r * complex<double>(cos(w), sin(w));

    // 예제 1 신호: x[n] = δ[n] = {1.0}
    vector<double> x_delta = { 1.0 };
    complex<double> X_delta = zTransform1D(x_delta, z);

    cout << "예제 1 : X(z) = "; printComplex(X_delta);
    cout << endl << endl;

    // 예제 2 신호: x[n] = u[n], 길이 5
    vector<double> x_step = { 1.0, 1.0, 1.0, 1.0, 1.0 };
    complex<double> X_step = zTransform1D(x_step, z);

    cout << "예제 2 : X(z) = "; printComplex(X_step);
    cout << endl << endl;

    return 0;
}
```

$$z = r \cdot e^{j\omega} \rightarrow r \cdot (cos(\omega) + jsin(\omega))$$

Example 1. $x[n] = \delta[n]$일 때, $X(z)$ 구하기
- $x[n] = \{1\}$

Example 2. $x[n] = u[n]$일 때, $X(z)$ 구하기 (길이 5)
-$x[n] = \{1, 1, 1, 1, 1\}$

Terminal 실행 [$z = (0.9) \cdot e^{j\frac{\pi}{4}}$]

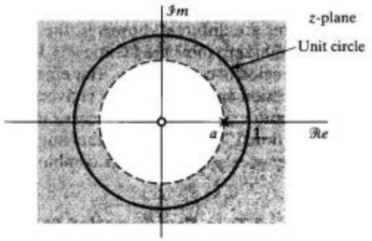
```
예제 1 : X(z) = 1 + j0

예제 2 : X(z) = -0.708451 - j2.99021
```

❖ 1D Z-transform (Assignment)

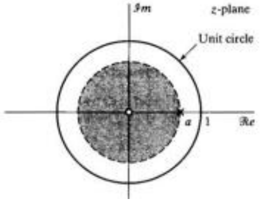
- [1번 문제]: Compute the Z-transform of the finite-length signal $x[n] = a^n \cdot u[n]$ (n = 0, 1, … , 50), and compare it with the closed-form expression $X(z) = \frac{1}{1-a\cdot z^{-1}}$

What is the Z transform of the following signals?

$$x[n] = a^n u[n]$$



$$x[n] = -a^n u[-n-1]$$



$$x[n] = a^n \cdot u[n]$$

1

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] \cdot z^{-n}$$

2

$$= \sum_{n=0}^{\infty} a^n \cdot z^{-n} = \frac{1}{1 - a \cdot z^{-1}}$$

무한 등비급수의 합

$$\sum_{n=1}^{\infty} a \cdot r^{n-1} = \frac{a}{1-r}$$

TABLE 3.1 SOME COMMON z-TRANSFORM PAIRS

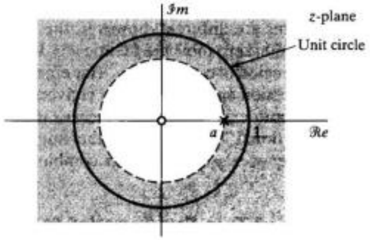
Sequence	Transform	ROC
1. $\delta[n]$	1	All z
2. $u[n]$	$\frac{1}{1-z^{-1}}$	$\lvert z\rvert > 1$
3. $-u[-n-1]$	$\frac{1}{1-z^{-1}}$	$\lvert z\rvert < 1$
4. $\delta[n-m]$	z^{-m}	All z except 0 (if $m>0$) or ∞ (if $m<0$)
5. $a^n u[n]$	$\frac{1}{1-az^{-1}}$	$\lvert z\rvert > \lvert a\rvert$
6. $-a^n u[-n-1]$	$\frac{1}{1-az^{-1}}$	$\lvert z\rvert < \lvert a\rvert$
7. $na^n u[n]$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$\lvert z\rvert > \lvert a\rvert$
8. $-na^n u[-n-1]$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$\lvert z\rvert < \lvert a\rvert$
9. $[\cos \omega_0 n]u[n]$	$\frac{1-[\cos\omega_0]z^{-1}}{1-[2\cos\omega_0]z^{-1}+z^{-2}}$	$\lvert z\rvert > 1$
10. $[\sin \omega_0 n]u[n]$	$\frac{[\sin\omega_0]z^{-1}}{1-[2\cos\omega_0]z^{-1}+z^{-2}}$	$\lvert z\rvert > 1$
11. $[r^n \cos \omega_0 n]u[n]$	$\frac{1-[r\cos\omega_0]z^{-1}}{1-[2r\cos\omega_0]z^{-1}+r^2z^{-2}}$	$\lvert z\rvert > r$
12. $[r^n \sin \omega_0 n]u[n]$	$\frac{[r\sin\omega_0]z^{-1}}{1-[2r\cos\omega_0]z^{-1}+r^2z^{-2}}$	$\lvert z\rvert > r$
13. $\begin{cases} a^n, & 0 \le n \le N-1, \\ 0, & \text{otherwise} \end{cases}$	$\frac{1-a^Nz^{-N}}{1-az^{-1}}$	$\lvert z\rvert > 0$

❖ 1D Z-transform (Assignment)

- **[1번 문제]: Compute the Z-transform of the finite-length signal $x[n] = a^n \cdot u[n]$ (n = 0, 1, … , 50), and compare it with the closed-form expression $X(z) = \frac{1}{1-a \cdot z^{-1}}$**

What is the Z transform of the following signals?

$$x[n] = a^n u[n]$$



$$x[n] = -a^n u[-n-1]$$

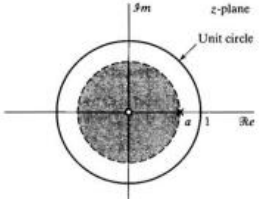


TABLE 3.1 SOME COMMON z-TRANSFORM PAIRS

Sequence	Transform	ROC
1. $\delta[n]$	1	All z
2. $u[n]$	$\frac{1}{1-z^{-1}}$	$\lvert z\rvert > 1$
3. $-u[-n-1]$	$\frac{1}{1-z^{-1}}$	$\lvert z\rvert < 1$
4. $\delta[n-m]$	z^{-m}	All z except 0 (if $m > 0$) or ∞ (if $m < 0$)
5. $a^n u[n]$	$\frac{1}{1-az^{-1}}$	$\lvert z\rvert > \lvert a\rvert$
6. $-a^n u[-n-1]$	$\frac{1}{1-az^{-1}}$	$\lvert z\rvert < \lvert a\rvert$
7. $na^n u[n]$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$\lvert z\rvert > \lvert a\rvert$
8. $-na^n u[-n-1]$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$\lvert z\rvert < \lvert a\rvert$
9. $[\cos \omega_0 n]u[n]$	$\frac{1-[\cos \omega_0]z^{-1}}{1-[2\cos \omega_0]z^{-1}+z^{-2}}$	$\lvert z\rvert > 1$
10. $[\sin \omega_0 n]u[n]$	$\frac{[\sin \omega_0]z^{-1}}{1-[2\cos \omega_0]z^{-1}+z^{-2}}$	$\lvert z\rvert > 1$
11. $[r^n \cos \omega_0 n]u[n]$	$\frac{1-[r\cos \omega_0]z^{-1}}{1-[2r\cos \omega_0]z^{-1}+r^2z^{-2}}$	$\lvert z\rvert > r$
n $\omega_0 n]u[n]$	$\frac{[r\sin \omega_0]z^{-1}}{1-[2r\cos \omega_0]z^{-1}+r^2z^{-2}}$	$\lvert z\rvert > r$
, $0 \le n \le N-1$, otherwise	$\frac{1-a^N z^{-N}}{1-az^{-1}}$	$\lvert z\rvert > 0$

$$x[n] = a^n \cdot u[n]$$

1 $$X(z) = \sum_{n=-\infty}^{\infty} x[n] \cdot z^{-n}$$

2 $$= \sum_{n=0}^{\infty} a^n \cdot z^{-n} = \frac{1}{1 - a \cdot z^{-1}}$$

무한 등비급수의 합

$$\sum^{\infty} a \cdot r^{n-1} = \frac{a}{\ldots}$$

```
1번 문제 : X(z) = 1.161 - j0.751169
이론식  : X(z) = 1.161 - j0.751169
```

❖ 1D Z-transform (Assignment)

- **[1번 문제]: Compute the Z-transform of the finite-length signal $x[n] = a^n \cdot u[n]$ (n = 0, 1, … , 50), and compare it with the closed-form expression $X(z) = \frac{1}{1-a\cdot z^{-1}}$**

  - **보고서 제출 항목**

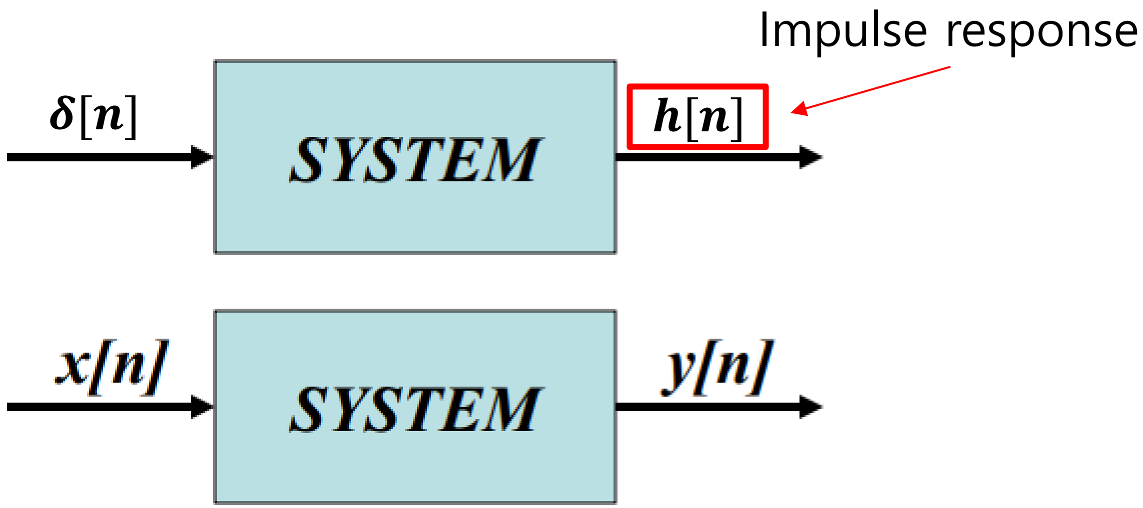
    ✓ $\sum_{n=-\infty}^{\infty} x[n] \cdot z^{-n}$ **로 계산한 출력 결과 코드**

    ✓ $\frac{1}{1-a\cdot z^{-1}}$**로 계산한 출력 결과 코드**

    $[a = 0.5\,, z = (0.9) \cdot e^{j\frac{\pi}{4}}]$ 일 때, $X(z)$를 구하시면 됩니다.

❖ 1D FIR filter (Example)

- **Finite Impulse Response (FIR) filter**

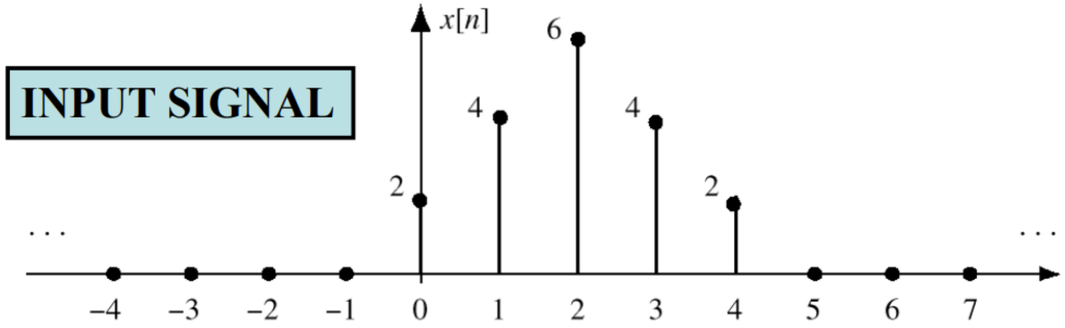


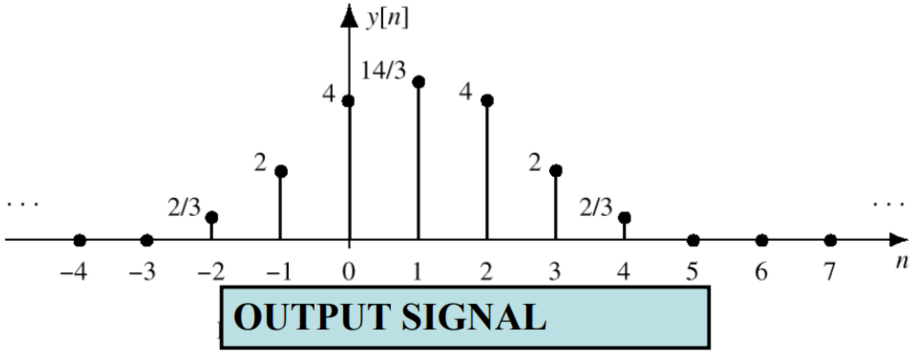
$$y[n] = system\left\{\sum_k x[k]\delta[n-k]\right\}$$

$$= \sum_k x[k] system\{\delta[n-k]\} \quad \text{(By Linearity)}$$

$$= \boxed{\sum_k x[k]h[n-k]} \quad \text{(By Time-Invariant)}$$

$$x[n] * h[n]$$





1D FIR filter (Example)

- **Convolution = Filter Definition**
  - **Filter Coeffs = Impulse Response**

n	$n<0$	0	1	2	3	$\dots$	M	$M+1$	$n>M+1$
$x[n]=\delta[n]$	0	1	0	0	0	0	0	0	0
$y[n]=h[n]$	0	b_0	b_1	b_2	b_3	$\dots$	b_M	0	0

$$y[n]=\sum_{k=0}^{M} b_k x[n-k]$$

$$y[n]=\sum_{k=0}^{M} h[k]x[n-k]$$

CONVOLUTION

- **FILTER COEFFICIENTS $\{b_k\}$**
  - **DEFINE THE FILTER**

$$y[n]=\sum_{k=0}^{M} b_k x[n-k]$$

  - **For example,** $b_k=\{3,-1,2,1\}$

$$\begin{aligned} y[n] &= \sum_{k=0}^{3} b_k x[n-k] \\ &= 3x[n]-x[n-1]+2x[n-2]+x[n-3] \end{aligned}$$

- **FILTER ORDER is M**
- **FILTER LENGTH is L = M+1**
  - **NUMBER of FILTER COEFFS is L**

❖ 1D FIR filter (Example)

```cpp
#include <iostream>
#include <vector>

using namespace std;

// 1D FIR filter: y[n] = sum_{k=0 ~ M} h[k]*x[n-k]

vector<double> firFilter1D(const vector<double>& x, const vector<double>& h)
{
    size_t N = x.size(); //입력 길이
    size_t L = h.size(); //필터 길이
    size_t M = L - 1;    //차수
    size_t Ny = N + L - 1; // 출력 길이

    vector<double> y(Ny, 0.0); // 출력 y 길이를 Ny로 하고 0으로 초기화

    for (size_t n = 0; n < Ny; ++n) { // n = 0, 1, ... ,Ny-1
        double acc = 0.0;             // y[n] 누적합

        for (size_t k = 0; k <= M; ++k) { // k = 0, 1, ... ,M
            if (n >= k && (n-k)<N) {      // x[n-k]의 인덱스가 양수일때만
                acc += h[k] * x[n - k];   // acc에 누적하고
            }
        }

        y[n] = acc; // 계산한 acc를 y[n]에 저장
    }

    return y;
}
```

1. firFilter1D function
   - Input : vector $x[n]$, vector $h[n]$
   - Output : $y[n]$

$$y[n] = \boxed{\sum_{k=0}^{M}} h[k]\,x[n-k]$$

$$y[n] = \{y[0], y[1], y[2], \ldots, y[\boxed{Ny} - 1]\}$$

2. $y[n]$의 길이 = convolution의 길이

   = $x[n]$의 길이 + $h[n]$의 길이 – 1
   = N + L – 1

3. for (y[n]의 index 반복문){
   for($\sum_{k=0}^{M} h[k]x[n-k]$ 반복문)

❖ 1D FIR filter (Example)

```cpp
int main()
{
    // 예제 입력: x[n] = {1, 2, 3, 4, 5}
    vector<double> x = { 1, 2, 3, 4, 5};

    // 예제 FIR 필터
    vector<double> h = { 1.0 / 3, 1.0 / 3, 1.0 / 3 };

    vector<double> y = firFilter1D(x, h);

    cout << "입력 x[n]: ";
    for (double v : x) cout << v << " ";
    cout << endl;

    cout << "필터 h[n]: ";
    for (double v : h) cout << v << " ";
    cout << endl;

    cout << "출력 y[n]: ";
    for (double v : y) cout << v << " ";
    cout << endl;

    return 0;
}
```

Example $x[n] = \{1, 2, 3, 4, 5\}$일 때, y$[n]$ 구하기

- $x[n] = \{1, 2, 3, 4, 5\}$
- $h[n] = \{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$

Terminal 실행

```
입력 x[n]: 1 2 3 4 5
필터 h[n]: 0.333333 0.333333 0.333333
출력 y[n]: 0.333333 1 2 3 4 3 1.66667
```

Range-based for문

```cpp
for (double v : x) cout << v << " ";
```

=

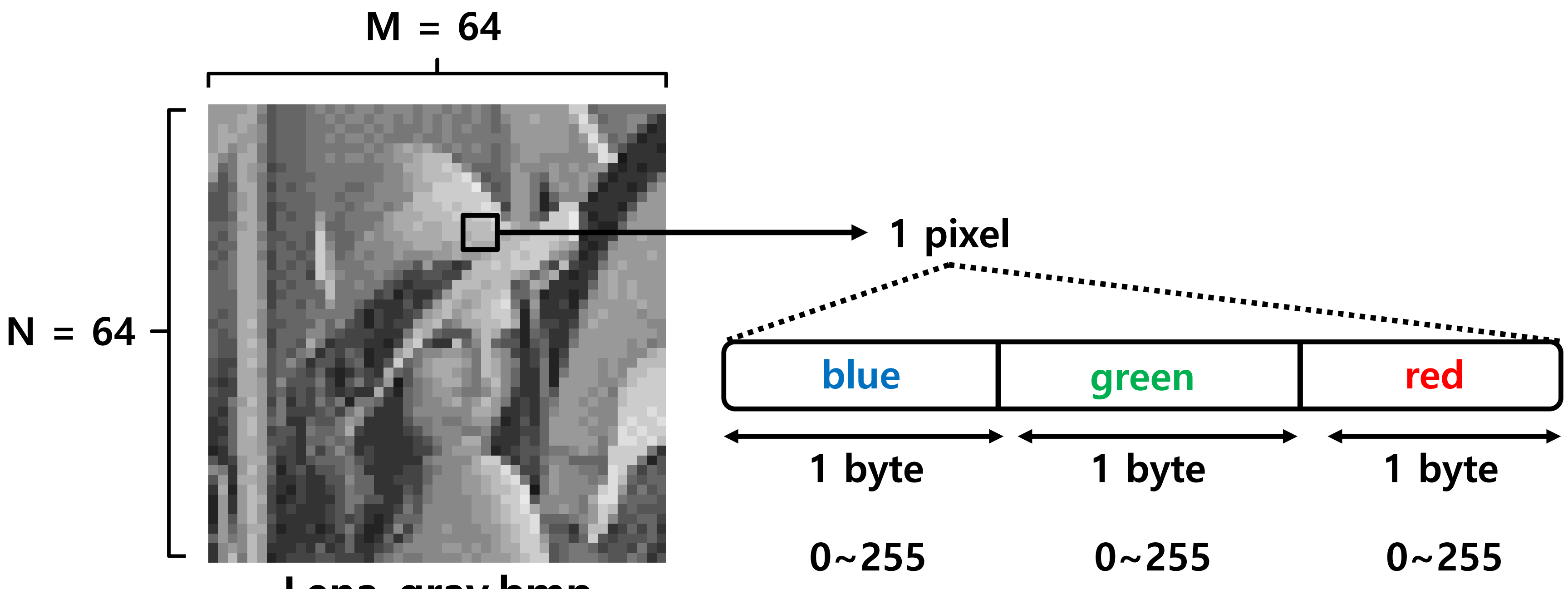
```cpp
for (size_t i = 0; i < x.size(); ++i) {
    double v = x[i];
    cout << v << " ";
}
```

❖ 2D FIR filter (Assignment)

- [2번 문제]: Apply an FIR Filter to a Grayscale Image(Lena_gray.bmp) and observe how the Image Changes

$$y[i,j] = \sum_{m}\sum_{n} \boxed{h[m,n]} \cdot \boxed{x[i-m,j-n]}$$

2차원에서의
FIR filter 수식



Lena_gray.bmp

❖ 2D FIR filter (Assignment)

- **[2번 문제]: Apply an FIR Filter to a Grayscale Image(Lena_gray.bmp) and observe how the Image Changes**

$$y[i,j] = \sum_{m}\sum_{n} \boxed{h[m,n]} \cdot \boxed{x[i-m, j-n]}$$

2차원에서의 FIR filter 수식

Filter 1. $h[3][3] = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$

Filter 2. $h[3][3] = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

Filter 3. $h[3][3] = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

Lena_gray.bmp에 Filter 1, 2, 3을 적용해 보고 이미지의 변화를 확인해보자

64 (pixel)

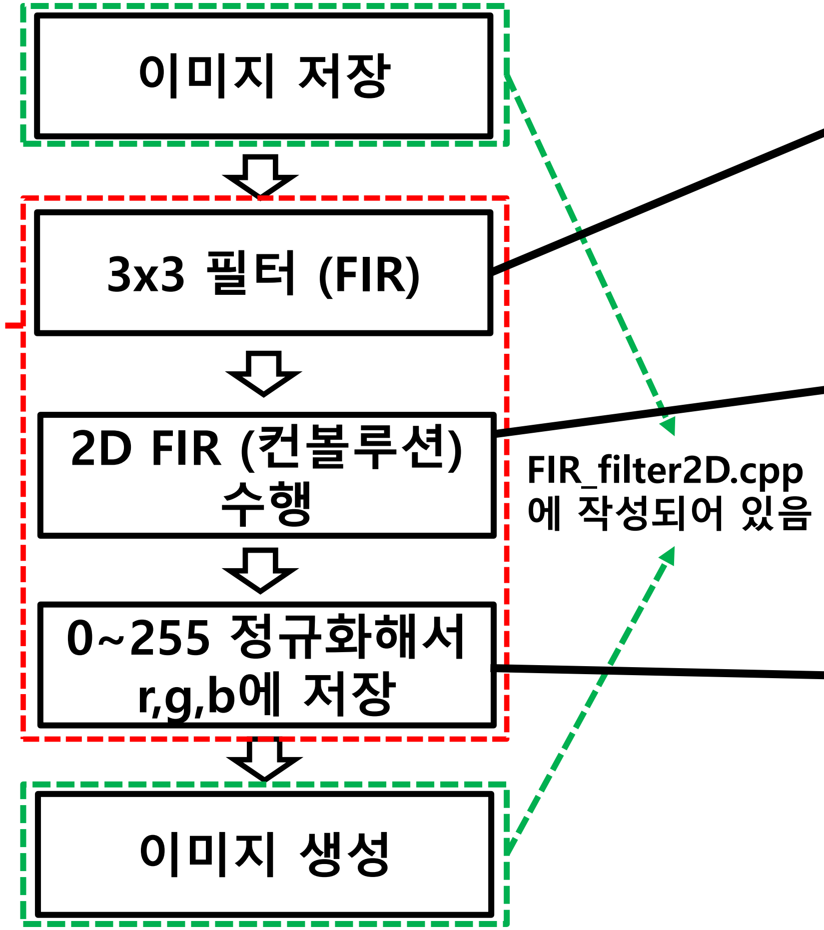
64

Lena_gray.bmp

❖ 2D FIR filter (Assignment)

- [2번 문제]: Apply an FIR Filter to a Grayscale Image(Lena_gray.bmp) and observe how the Image Changes

[코드 구성]



작성해야 하는 부분

```
/* 작성해야 하는 코드 부분 --------------------------------- */

// 3x3 평균 필터 커널 (FIR)

// 2D FIR (컨볼루션) 수행

// double 값을 BYTE(0~255)로 바꿔서 r,g,b에 다시 넣기
//     1. double 값을 반올림 (정수로 바꾸기)
//     2. 0보다 작으면 -> 0으로 , 255보다 크면 -> 255로
//     3. r, g, b에 이 값을 저장

/* ---------------------------------------------------------- */
```

Filter 1. $h[3][3] = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$ Filter 2. $h[3][3] = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

Filter 3. $h[3][3] = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

$$y[i,j] = \sum_{m}\sum_{n} h[m,n] \cdot x[i-m, j-n]$$

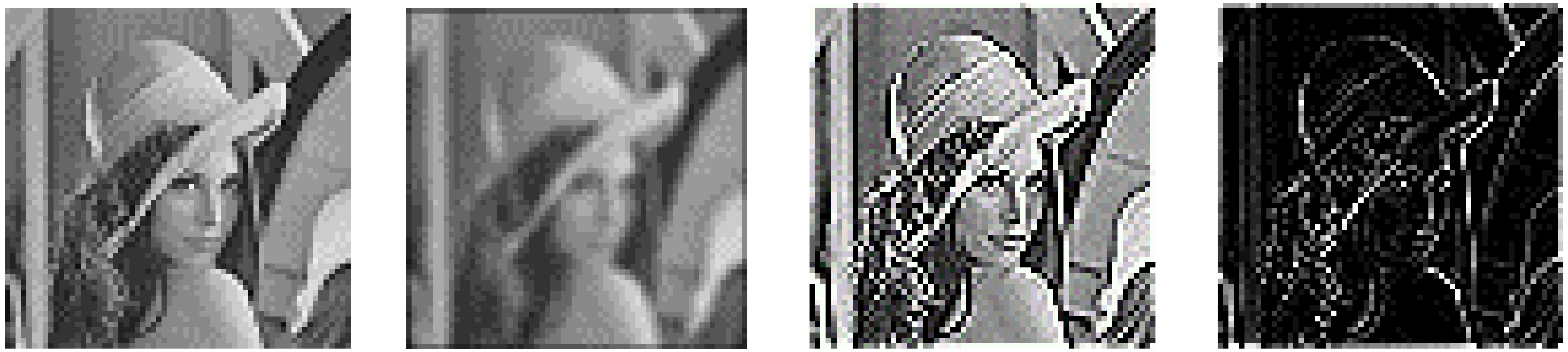
1. double 값을 반올림 (정수로 만들기)
2. FIR filter로 변형된 픽셀 값 = x

$$x = \begin{cases} 0 & \text{If } (\ x < 0) \\ 255 & \text{If } (\ x > 255) \\ x & \text{otherwise} \end{cases}$$

3. r, g, b에 이 값을 저장

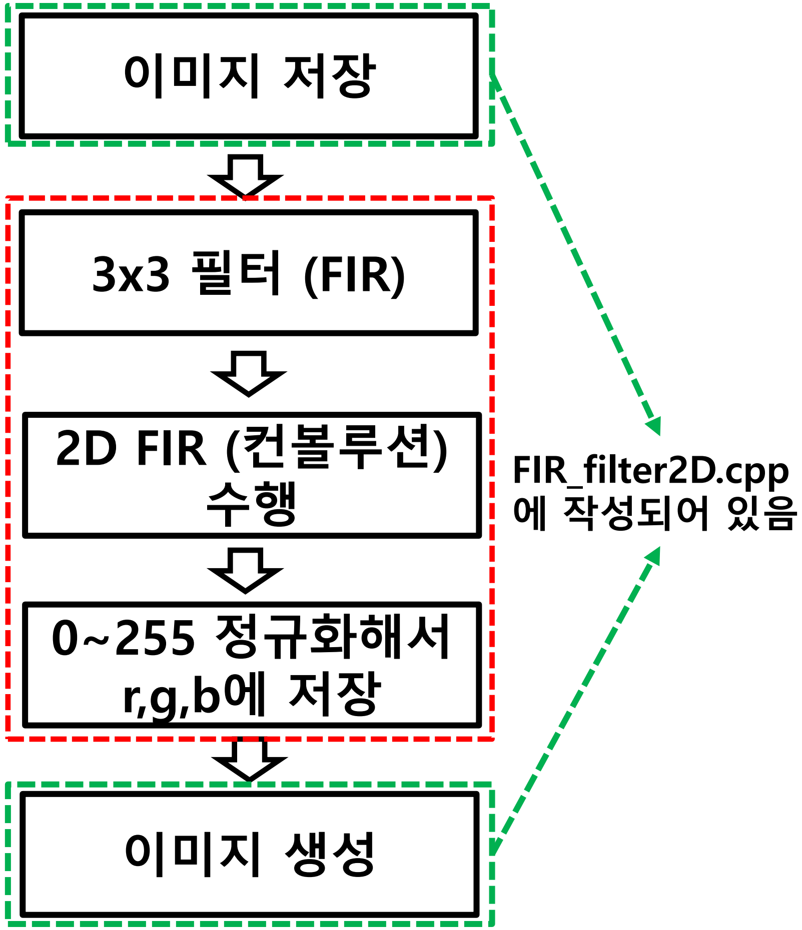
❖ 2D FIR filter (Assignment)

- [2번 문제]: Apply an FIR Filter to a Grayscale Image(Lena_gray.bmp) and observe how the Image Changes
  - 보고서 제출 항목
    - ✓ Lena_gray.bmp의 Filter 1 적용 이미지
    - ✓ Lena_gray.bmp의 Filter 2 적용 이미지
    - ✓ Lena_gray.bmp의 Filter 3 적용 이미지
    - ✓ FIR_filter2D code file



Original　　Filter 1　　Filter 2　　Filter 3

[코드 구성]



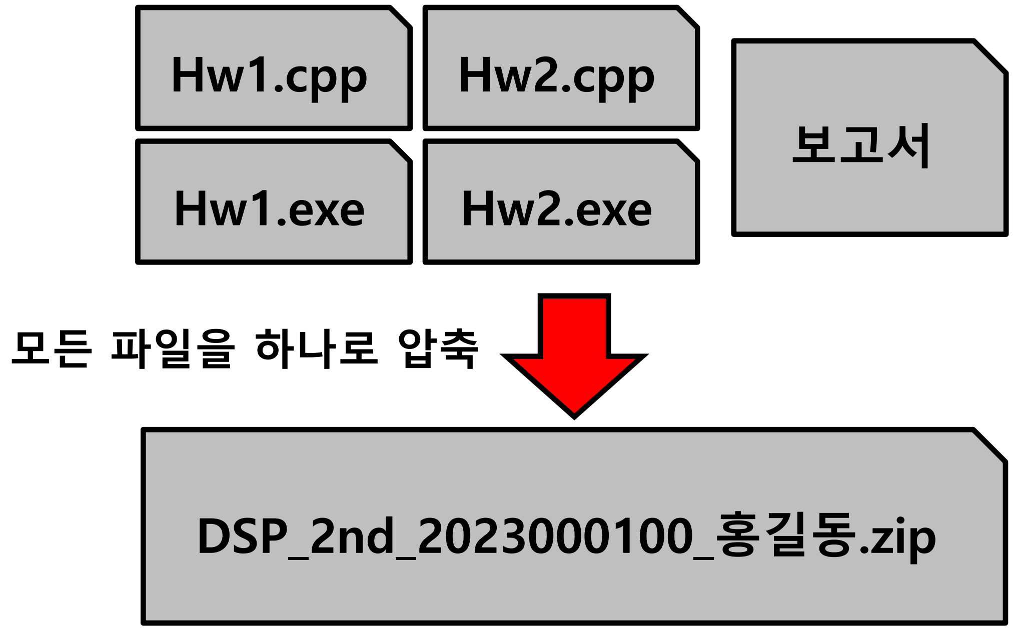
Submission Guide

Submission Guide

Lab 2.

제출일		학번	
전공		이름	

① 문제	
② 주요 변수	
③ 알고리즘	
④ 결과	



1. 파일명 : "DSP_2nd_학번_이름.zip"

EX) DSP_2nd_2025000100_홍길동.zip

2. 제출파일

- ✓ 보고서(hwp, word): 이름, 학번, 주요 변수, 알고리즘, 결과 분석, 느낀 점 작성
- ✓ 프로그램 파일(두 가지 언어 중 하나 선택)
  - C++ : (.cpp, .exe) 파일 2개
  - Python : (.py) 파일 1개

3. 기타 사항

- ✓ 제출이 늦을 경우 감점은 있지만, 학기 종료까지 제출하시면 점수가 있습니다

Thank you