DSP Lab. Week 1
Drawing sinusoidal waves

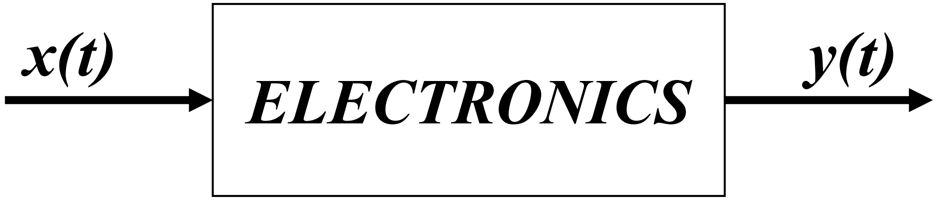
Kyuheon Kim
Media Lab. Rm567
kyuheonkim@khu.ac.kr

Last update : August 24, 2019

Sampling

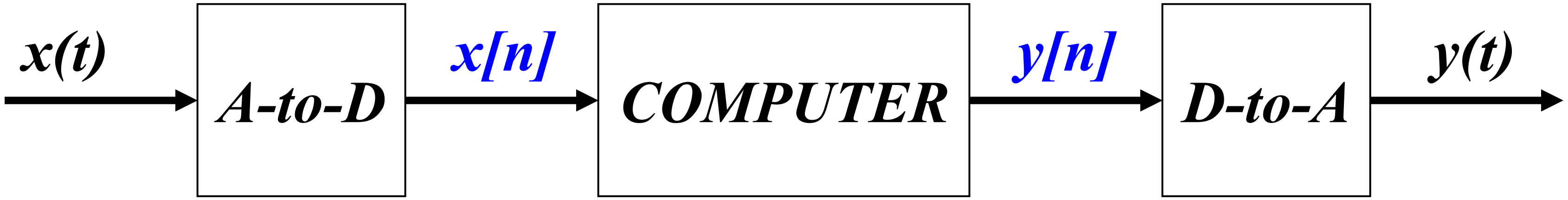
❖ ANALOG/ELECTRONIC:

- ✓ Circuits: resistors, capacitors, op-amps

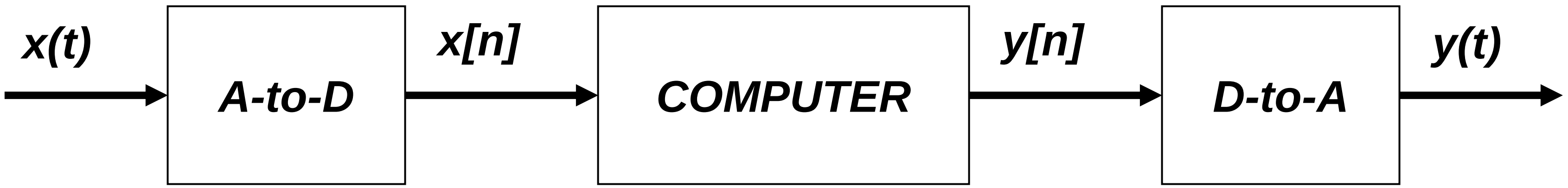


❖ DIGITAL/MICROPROCESSOR

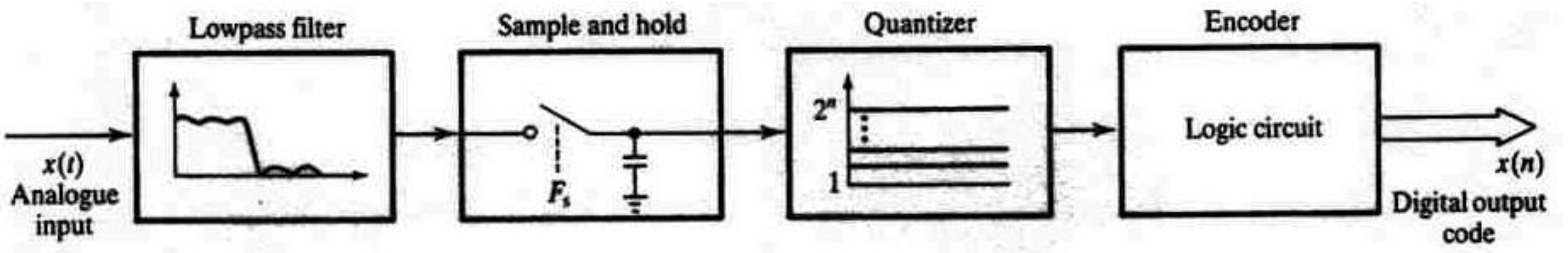
- ✓ Convert x(t) to numbers stored in memory



Sampling



- ❖ **A-to-D**
  - ✓ Convert x(t) to numbers stored in memory
- ❖ **D-to-A**
  - ✓ Convert y[n] back to a “continuous-time” signal, y(t)
  - ✓ y[n] is called a “discrete-time” signal



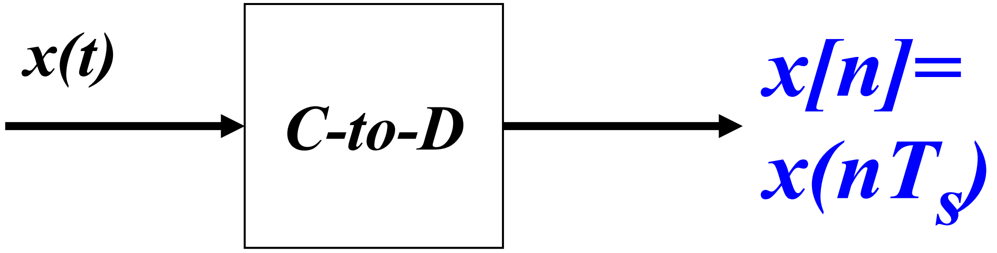
Sampling

❖ SAMPLING PROCESS

- ✓ Convert x(t) to numbers x[n]
- ✓ "n" is an integer; x[n] is a sequence of values
- ✓ Think of "n" as the storage address in memory

❖ UNIFORM SAMPLING at t = nTs

- ✓ IDEAL: x[n] = x(nTs)



❖ SAMPLING RATE (fs)

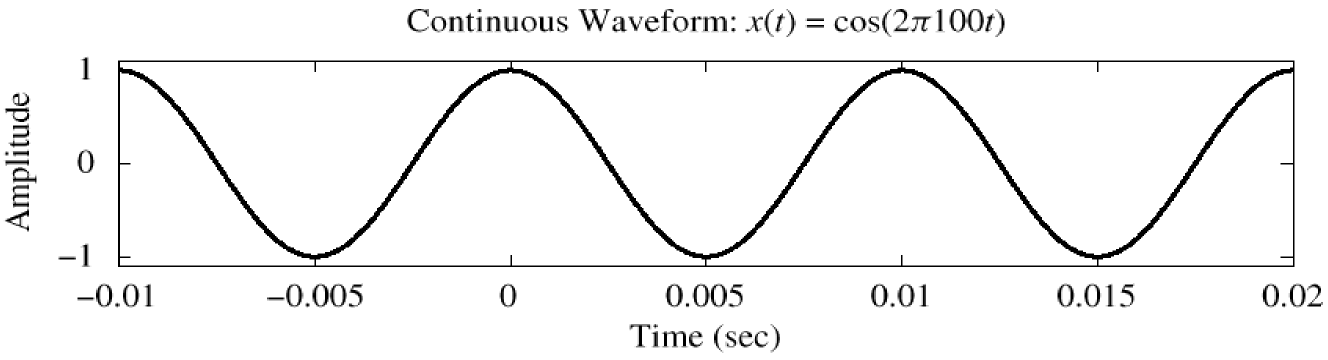
- fs =1/Ts
  - ✓ NUMBER of SAMPLES PER SECOND
- Ts = 125 microsec → fs = 8000 samples/sec
  - UNITS ARE HERTZ: 8000 Hz

❖ UNIFORM SAMPLING at t = nTs = n/fs

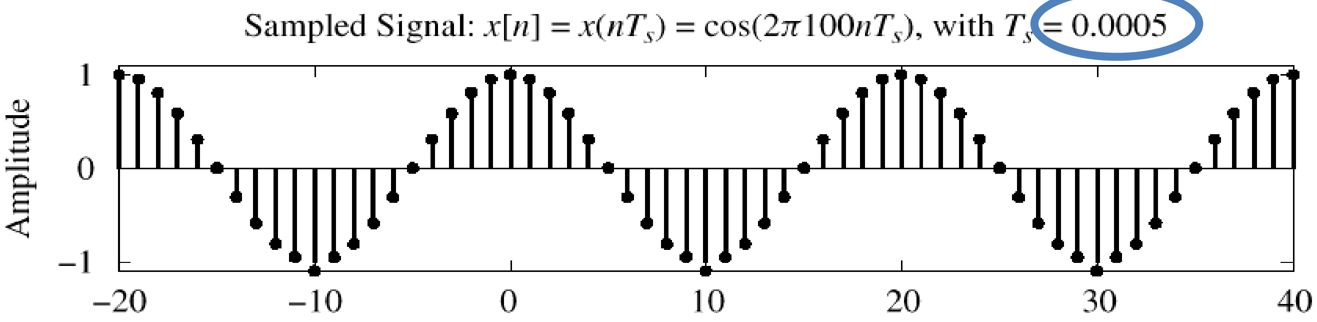
- IDEAL: x[n] = x(nTs)=x(n/fs)

Sampling

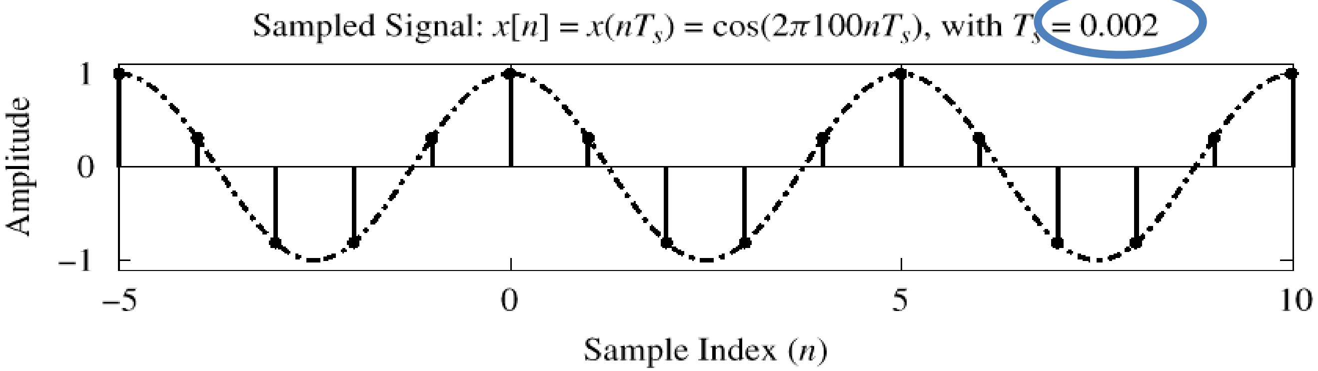
$f = 100\text{Hz}$



$f_s = 2\,\text{kHz}$



$f_s = 500\text{Hz}$



f_s : number of samples per a second

$n = T / T_s$: number of samples per a period

Sampling

- HOW OFTEN ?
  - DEPENDS on FREQUENCY of SINUSOID
  - ANSWERED by SHANNON/NYQUIST Theorem
  - ALSO DEPENDS on "<u>RECONSTRUCTION</u>"

Shannon Sampling Theorem

A continuous-time signal $x(t)$ with frequencies no higher than $f_{\max}$ can be reconstructed exactly from its samples $x[n] = x(nT_s)$, if the samples are taken at a rate $f_s = 1/T_s$ that is greater than $2f_{\max}$.

Sampling

- x[n] is a SAMPLED SINUSOID
  - A list of numbers stored in memory
- EXAMPLE: audio CD
- CD rate is 44,100 samples per second
  - 16-bit samples
  - Stereo uses 2 channels
- Number of bytes for 1 minute is
  - 2 X (16/8) X 60 X 44100 = 10.584 Mbytes

Sampling

- ❖ **Change x(t) into x[n]**

$$x(t) = A\cos(\omega t + \varphi)$$

$$x[n] = x(nT_s) = A\cos(\omega n T_s + \varphi)$$

$$x[n] = A\cos((\omega T_s)n + \varphi)$$

$$x[n] = A\cos(\hat{\omega} n + \varphi)$$

$$\hat{\omega} = \omega T_s = \frac{\omega}{f_s}$$

DEFINE DIGITAL FREQUENCY
Called as Normalised Radian frequency

Sampling

- ❖ $\hat{\omega}$ VARIES from 0 to 2π, as f varies from 0 to the sampling frequency
- ❖ UNITS are radians, not rad/sec
  - DIGITAL FREQUENCY is NORMALIZED

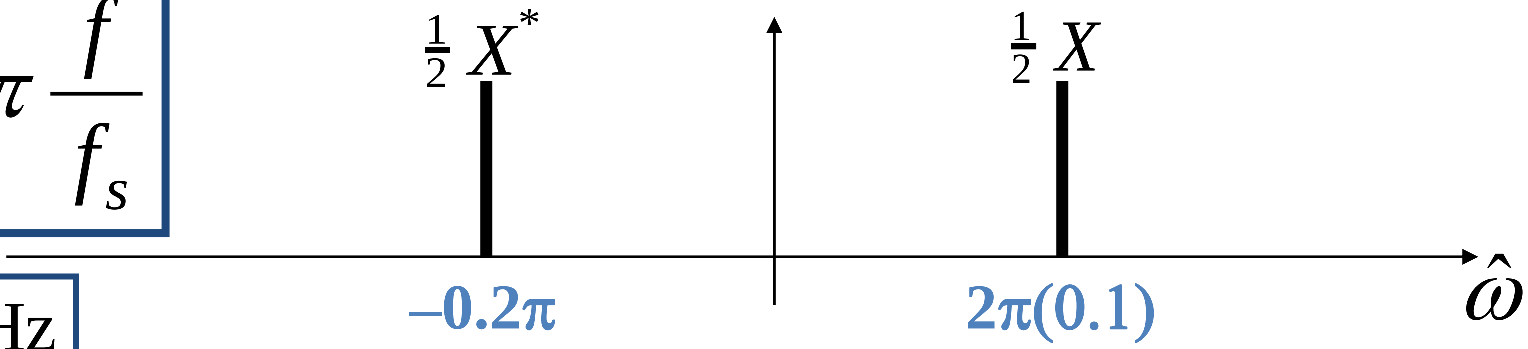
$$\boxed{\hat{\omega} = \omega T_s = \frac{2\pi f}{f_s}}$$

$$\boxed{f_s : \text{number of samples per a second}}$$

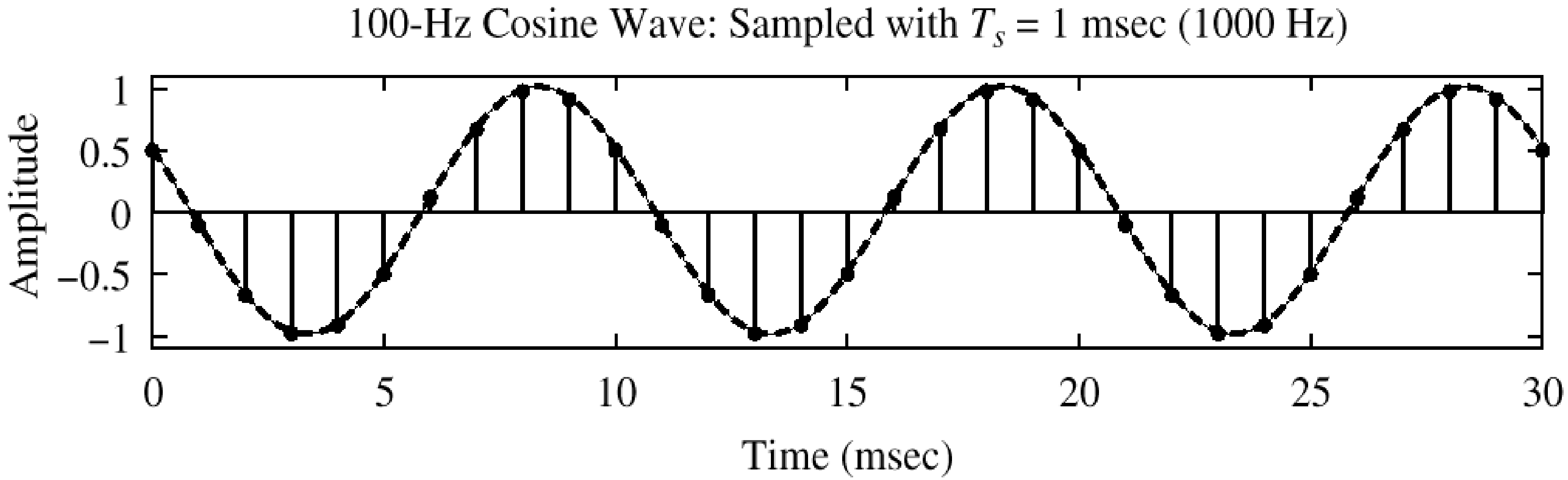
Sampling

$$\boxed{\hat{\omega} = 2\pi \frac{f}{f_s}}$$

$$\boxed{f_s = 1\,\text{kHz}}$$



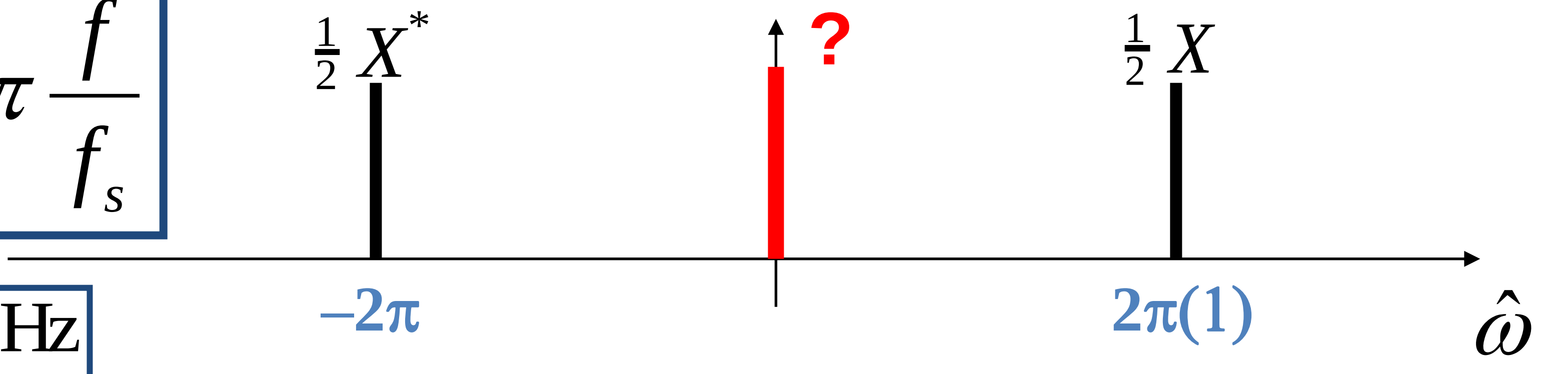
$$x[n] = A\cos(2\pi(100)(n/1000) + \varphi)$$



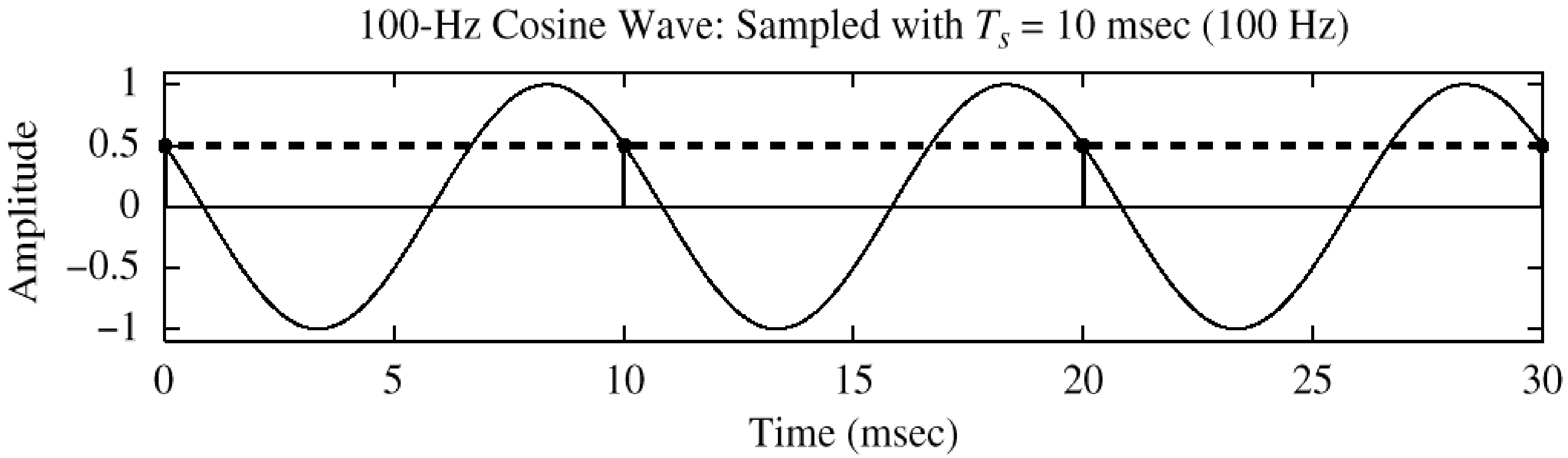
Sampling

$$\boxed{\hat{\omega} = 2\pi \frac{f}{f_s}}$$

$$\boxed{f_s = 100\ \text{Hz}}$$



$$x[n] = A\cos(2\pi(100)(n/100) + \varphi)$$



Sampling

- Spectrum of x[n] has more than one line for each complex exponential
  - Called <u>ALIASING</u>
  - <u>MANY SPECTRAL LINES</u>

- SPECTRUM is PERIODIC with period = 2π
  - Because

$$A\cos(\hat{\omega}n+\varphi)=A\cos((\hat{\omega}+2\pi)n+\varphi)$$

Sampling

❖ **Other Frequencies give the same**

$$x(t) = A\cos(2pft + f)$$

$$x[n] = x(nT_s) = A\cos(2pfnT_s + f)$$

$$y(t) = A\cos(2p(f + \ell f_s)t + f)$$

$$y[n] = y(nT_s) = A\cos(2p(f + \ell f_s)nT_s + f)$$

$$= A\cos((2pfT_s)n + (2p\ell f_s T_s)n + f)$$

$$= A\cos((2pfT_s)n + 2p\ell n + f)$$

$$y[n] = A\cos((2pfT_s)n + f) = A\cos(\hat{w}n + f) = x[n] \qquad \hat{w} = 2pfT_s = \frac{2pf}{f_s}$$

❖ **Aliases of the frequency f with respect to the sampling frequency fs**

Sampling

- Other Frequencies give the same $\hat{\omega}$

$$x_1(t) = \cos(400\pi t) \quad \text{sampled at } f_s = 1000 \text{ Hz}$$

$$x_1[n] = \cos(400\pi \tfrac{n}{1000}) = \cos(0.4\pi n)$$

$$x_2(t) = \cos(2400\pi t) \quad \text{sampled at } f_s = 1000 \text{ Hz}$$

$$x_2[n] = \cos(2400\pi \tfrac{n}{1000}) = \cos(2.4\pi n)$$

$$x_2[n] = \cos(2.4\pi n) = \cos(0.4\pi n + 2\pi n) = \cos(0.4\pi n)$$

$$\Rightarrow x_2[n] = x_1[n]$$

$$2400\pi - 400\pi = 2\pi(1000)$$

Sampling

$$w(t) = A\cos(2\pi(-f_0 + \ell f_s)t - \phi)$$

$$\begin{aligned}
w[n] = w(nT_s) &= A\cos(2\pi(-f_0 + \ell f_s)nT_s - \phi) \\
&= A\cos(-2\pi f_0 n T_s + 2\pi \ell f_s T_s - \phi) \\
&= A\cos(-2\pi f_0 n T_s + 2\pi \ell - \phi) \\
&= A\cos(2\pi f_0 n T_s + \phi) \\
&= x[n]
\end{aligned}$$

Sampling

- **ADDING f_s or $2f_s$ or $-f_s$ to the FREQ of x(t) gives exactly the same x[n]**
  - **The samples, x[n] = x(n/f_s) are EXACTLY THE SAME VALUES**

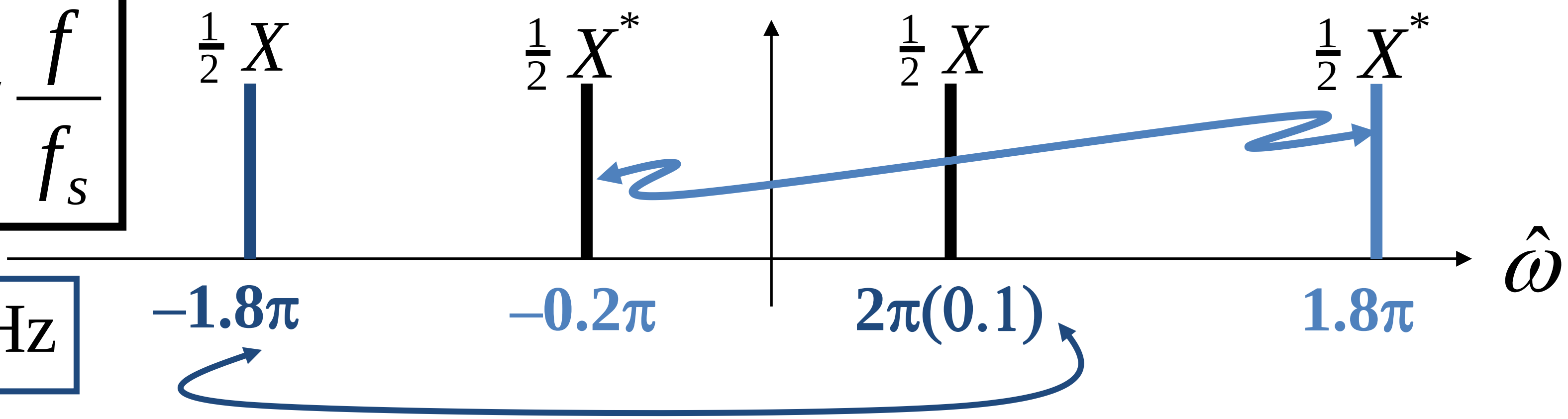
- **GIVEN x[n], WE CAN'T DISTINGUISH f_o FROM ($f_o + f_s$) or ($f_o + 2f_s$)**

- **The frequency are callled aliases of the frequency f with respect to the sampling frequency f_s**

Sampling

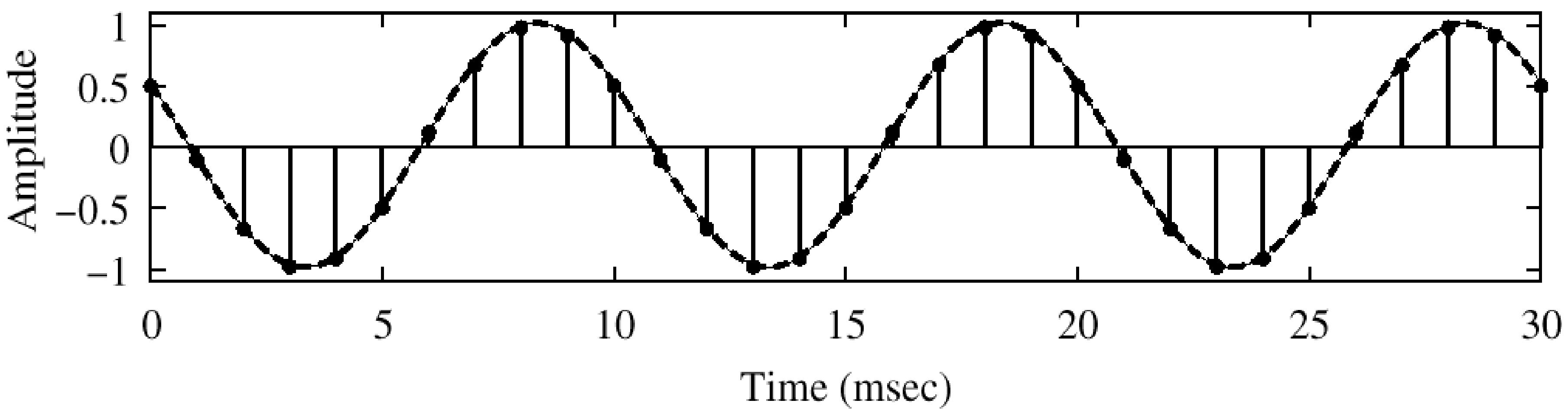
$$\boxed{\hat{\omega} = 2\pi \frac{f}{f_s}}$$

$$\boxed{f_s = 1\,\text{kHz}}$$



$$x[n] = A\cos(2\pi(100)(n/1000) + \varphi)$$

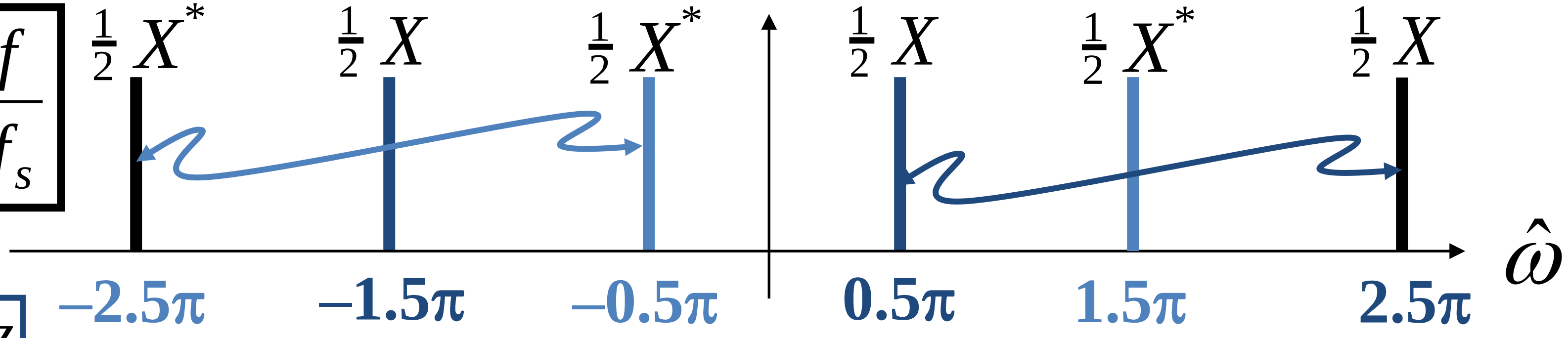
100-Hz Cosine Wave: Sampled with T_s = 1 msec (1000 Hz)



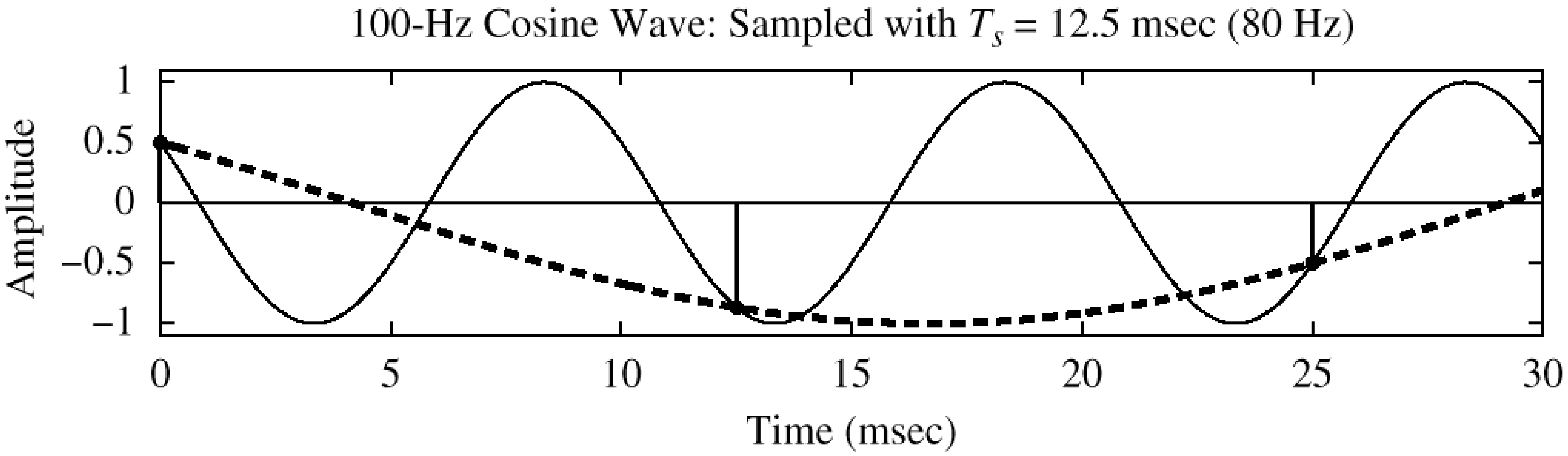
Sampling

$$\boxed{\hat{\omega} = 2\pi \frac{f}{f_s}}$$

$$\boxed{f_s = 80\,\text{Hz}}$$

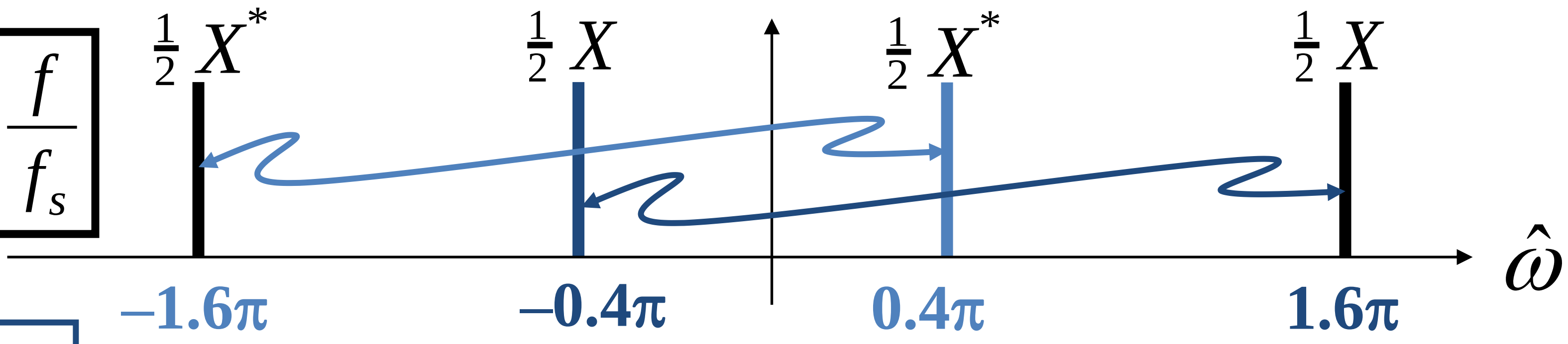


$$x[n] = A\cos(2\pi(100)(n/80) + \varphi)$$



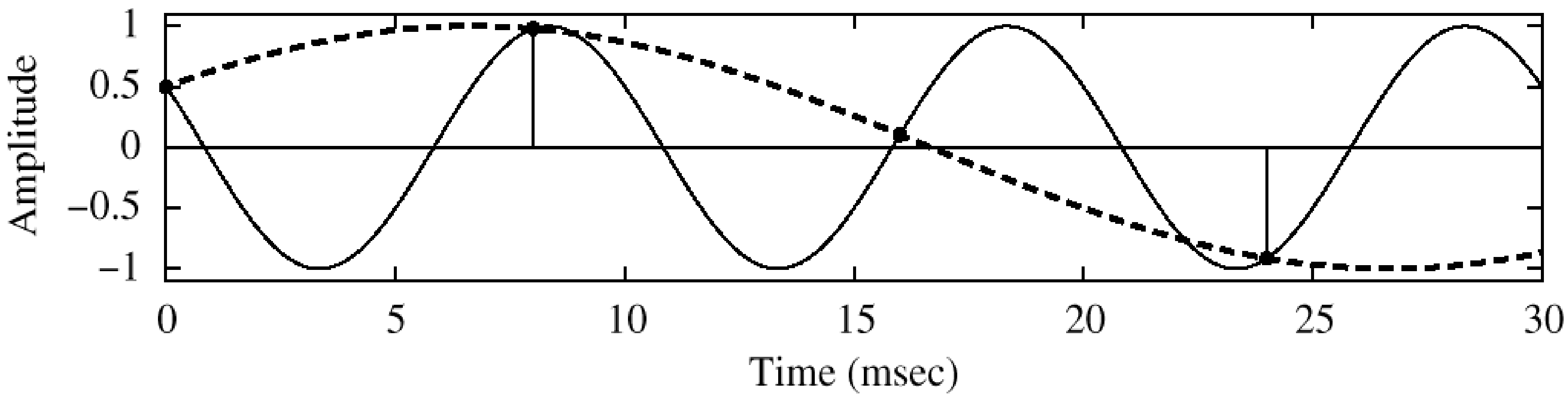
Sampling

$$\hat{\omega} = 2\pi \frac{f}{f_s}$$



$$f_s = 125 \text{ Hz}$$

$$x[n] = A\cos(2\pi(100)(n/125) + \varphi)$$



C-program (Sinusoidal wave) ex1

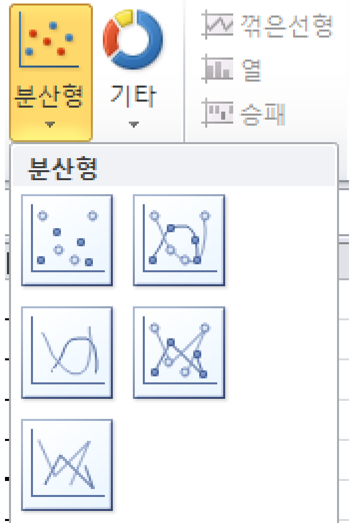
```
#include <iostream> // cout, cin
#include <fstream>  // ifstream, ofstream 입출력 파일 라이브러리
using namespace std;
#define PHI 3.141592
int main(){
        ofstream outFile;                      // 출력 파일 선언.
        outFile.open("data.txt", ios::out); // 출력 파일 data.txt 열기.
        float t, dt, f0;
        t = 0;
        dt = 1./44000.;  // fs = 44000Hz smapling frequency
        f0 = 440;          // 440Hz signal

        for(int i=0;i<400;i++,t+=dt)
             outFile << t << " " << sin(2.*PHI*f0*t) << endl;

        outFile.close();
        return 0;
}
```

Return type

main Function



// 프로그램을 만드는 폴더에서 data.txt를 읽어서 excel로 그래프 그린다.
// 두 개의 열을 drag한 후, "삽입➔분산형"으로 그래프를 그린다.

C-program (Sinusoidal wave) ex2

$$x(t) = 2\cos\left(2\pi(50)t + \frac{\pi}{2}\right) + \cos(2\pi(150)t)$$

$$f_0 = 50Hz, T_0 = 0.02$$
$$f_s = 300Hz, T_s = \frac{1}{300}$$

```
#include <iostream> // cout, cin
#include <fstream>  // ifstream, ofstream 입출력 파일 라이브러리
using namespace std;
#define PHI 3.141592
int main()
{
  ofstream outFile;                          // 출력 파일 선언.
  outFile.open("data.txt", ios::out);        // 출력 파일 data.txt 열기.
  float t = 0, fs = 300., dt = 1. / fs;      // 시간, 샘플링 주파수, 샘플링 주기
  int f0 = 50,n=3,smp_cnt;                  // 기본주파수, n개 주기 파형, 신호 샘플 개수
  smp_cnt = (fs / f0)*n;
  for (int i = 0; i <= smp_cnt; i++, t += dt)
    outFile << t << " " << 2*cos(2.*PHI*50*t+0.5*PHI)+ cos(2.*PHI*150*t)<< endl;

  outFile.close();
  return 0;
}
```

n주기 파형, $n \times T_0 =$ (신호 샘플 개수) $\times T_s$

Week 1 assignment

1. 어떤 주기 함수가 다음 식
$x(t) = 2 + 4\cos\left(30\pi t - \frac{1}{5}\pi\right) + 3\sin(40\pi t) + 4\cos\left(60\pi t - \frac{1}{3}\pi\right)$ 으로 주어졌다.
기본 주파수 f_0, 샘플링 주파수 f_s를 결정하여 3주기의 x(t)를 출력하라.

2. 변조(Modulation): 신호 $\mathrm{x(t)} = \cos(2\pi f_0 t)\sin(2\pi f_c t)$ 를 그려라. 이 그림은 base band frequency $\mathrm{f}_0 = 200\mathrm{Hz}$인 신호를 carrier frequency $\mathrm{f}_\mathrm{c} = 1600\mathrm{Hz}$로 modulation한 신호이다.

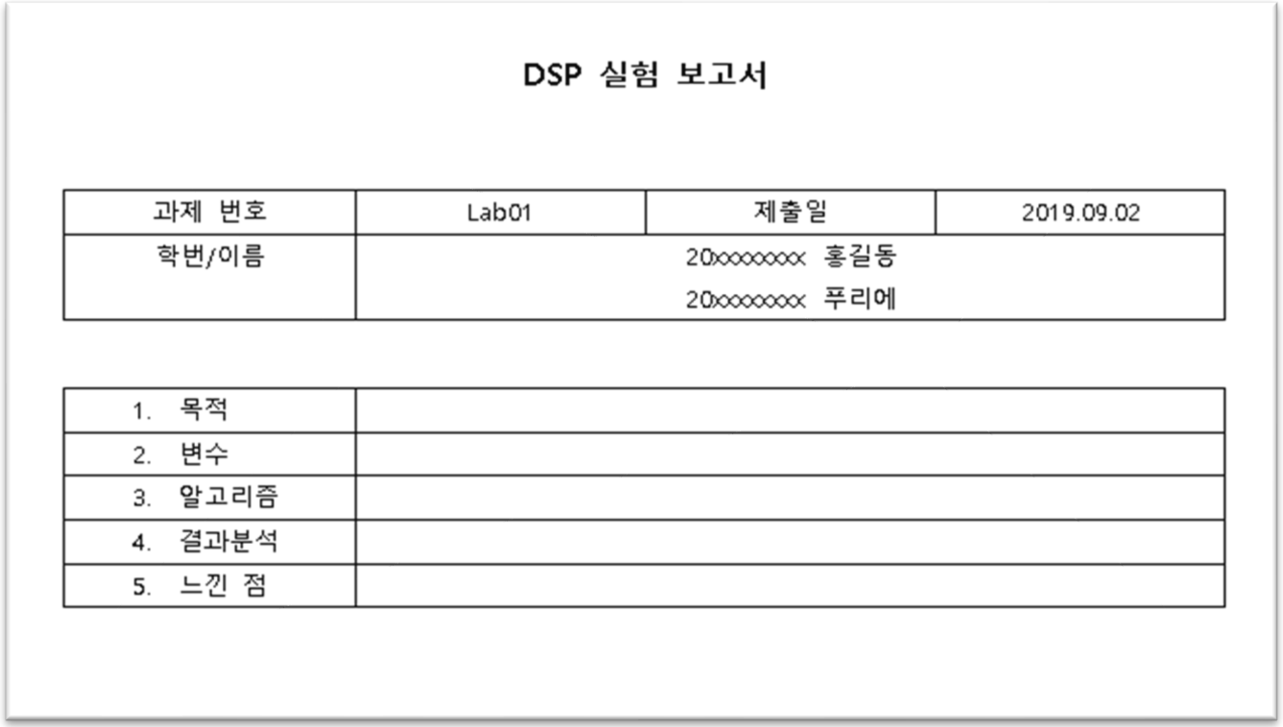
Week 1 assignment

"KLAS에 제출할 때 다음 사항을 꼭 지켜주세요"

1. 파일명 : "Lab00_요일_대표자이름.zip"

Ex) Lab01_목_홍길동.zip **(압축 툴은 자유롭게 사용)**

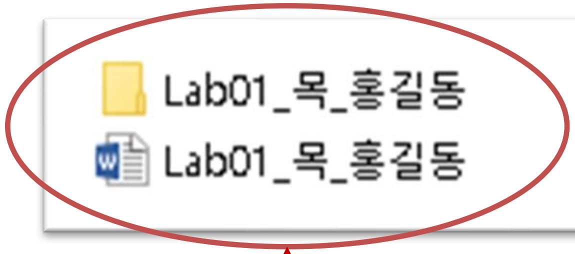
2. 제출 파일 **(보고서와 프로그램을 압축해서 제출)**
   - 보고서 파일 (hwp, word): 이름, 학번, 목적, 변수, 알고리즘(순서), 결과 분석, 느낀 점
   - 프로그램



DSP 실험 보고서

과제 번호	Lab01	제출일	2019.09.02
학번/이름	20xxxxxxxx 홍길동 20xxxxxxxx 푸리에		

1. 목적	
2. 변수	
3. 알고리즘	
4. 결과분석	
5. 느낀 점	



C++ 프로그램 파일
보고서

압축하여 제출