

제 4 장 Sampling과 Aliasing

1. 목 적

3장에서는 PC/AT와 A/D Converter Interface를 이용하여 Analog 신호를 Digital Data로 변환하는 것을 배웠다. 그러면 Sampling한 Digital Data를 Graph용지에 그리든가, Monitor에 Plot하여 보면 원래의 Analog 신호와 유사해 보일까? 물론 Sampling 시간 간격을 무한히 작게 하면 Analog 신호와 모양이 같아지겠지만 실제적으로는 그렇게 할 수 없다. 첫째로는, A/D Converter의 Time Resolution에 있어서 한계가 있기 때문이고, 둘째로는, 시간간격이 작아질수록 data갯수가 지나치게 많아지든가, 반대로 Sampling 총 시간이 짧아지기 때문이다. 그러면 어떤 신호를 A/D변환하여 Sampling할 때, Sampling time interval은 어떻게 하여야 할까?

본 장에서는 Analog 신호를 Digital data로 변환하는 과정에서 생기는 Aliasing 현상과 이를 관찰하기 위하여 Fast Fourier Transform(FFT)을 이용하는 방법을 공부한다. 아울러 이를 통하여 Computer와 A/D Converter Interface를 이용한 A/D 변환을 복습한다.

2. 예습부문

- (1) Random data : Analysis and Measurement Procedures(2nd Ed.), J. S. Bendat & A. G. Piersol, pp.325-342
- (2) Mechanical Measurements (4th Ed.) T. G. Beckwith & R. D. Marangoni Chapter 4. (pp.95-121)

3. 실험장치

- (1) Oscilloscope, FG, DMM, Frequency Counter
- (2) PC/AT, Printer
- (3) A/D Converter Board (DT2814)

4. 이 론

4.1 Fourier Transform

우리들은 응용해석학 시간에 Fourier Transform을 배웠으며, 따라서 시간과

진폭으로 나타난 여러 주기함수들을 주파수와 진폭으로 나타낼 수 있음을 알 것이다. 다시 한번 개념을 정리하기 위하여 몇가지 예를 들어본다.

Fig.4.1은 두가지의 주파수로 나타나는 함수에서 상대적인 진폭의 변화에 따른 영향을 본 것이다.

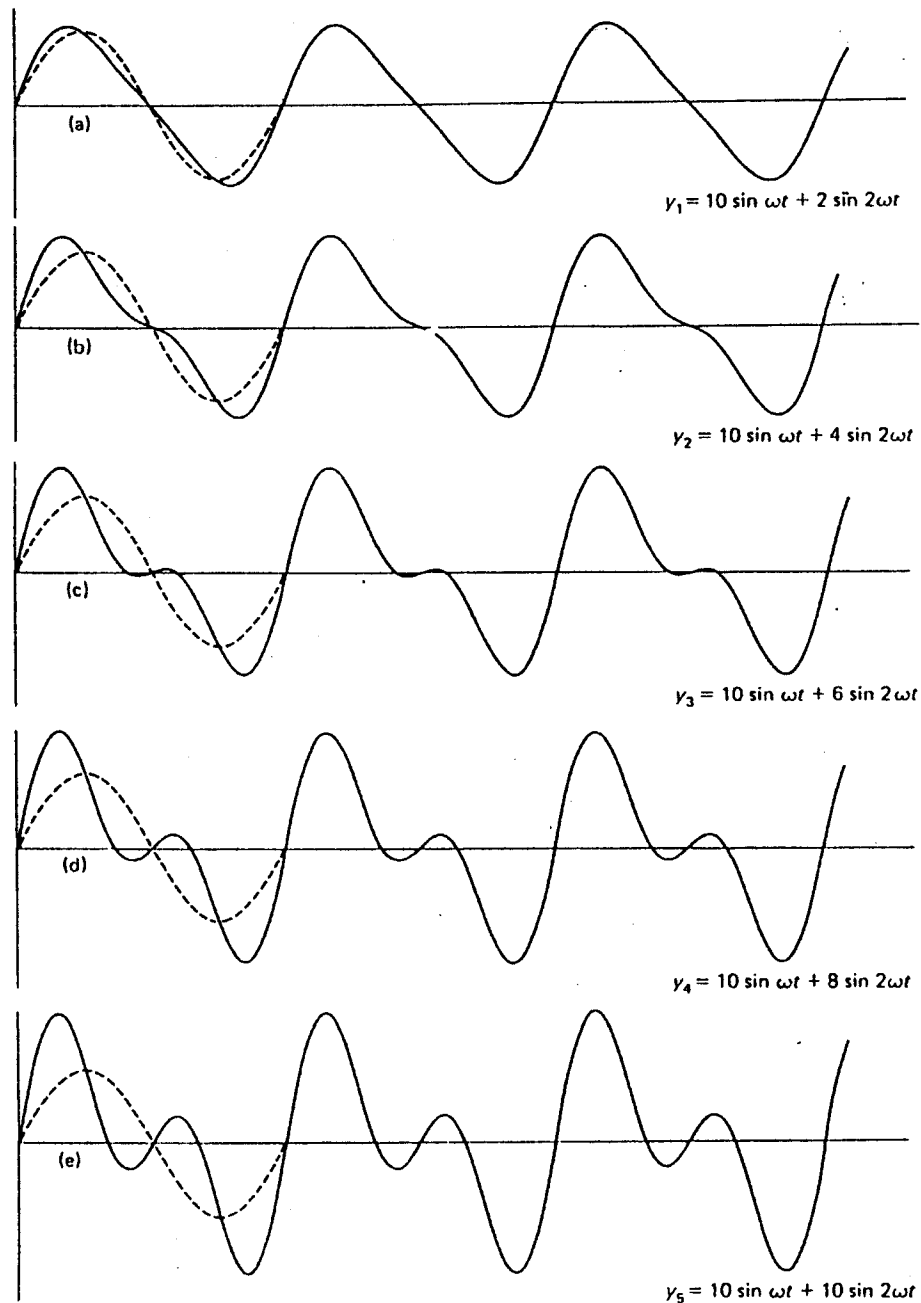


Fig.4.1 Examples of two component waveforms with second harmonic component of various relative amplitudes.

Fig.4.2는 진폭은 일정하면서 주파수 변화에 따른 파형을 모양변화를 본 것이다.

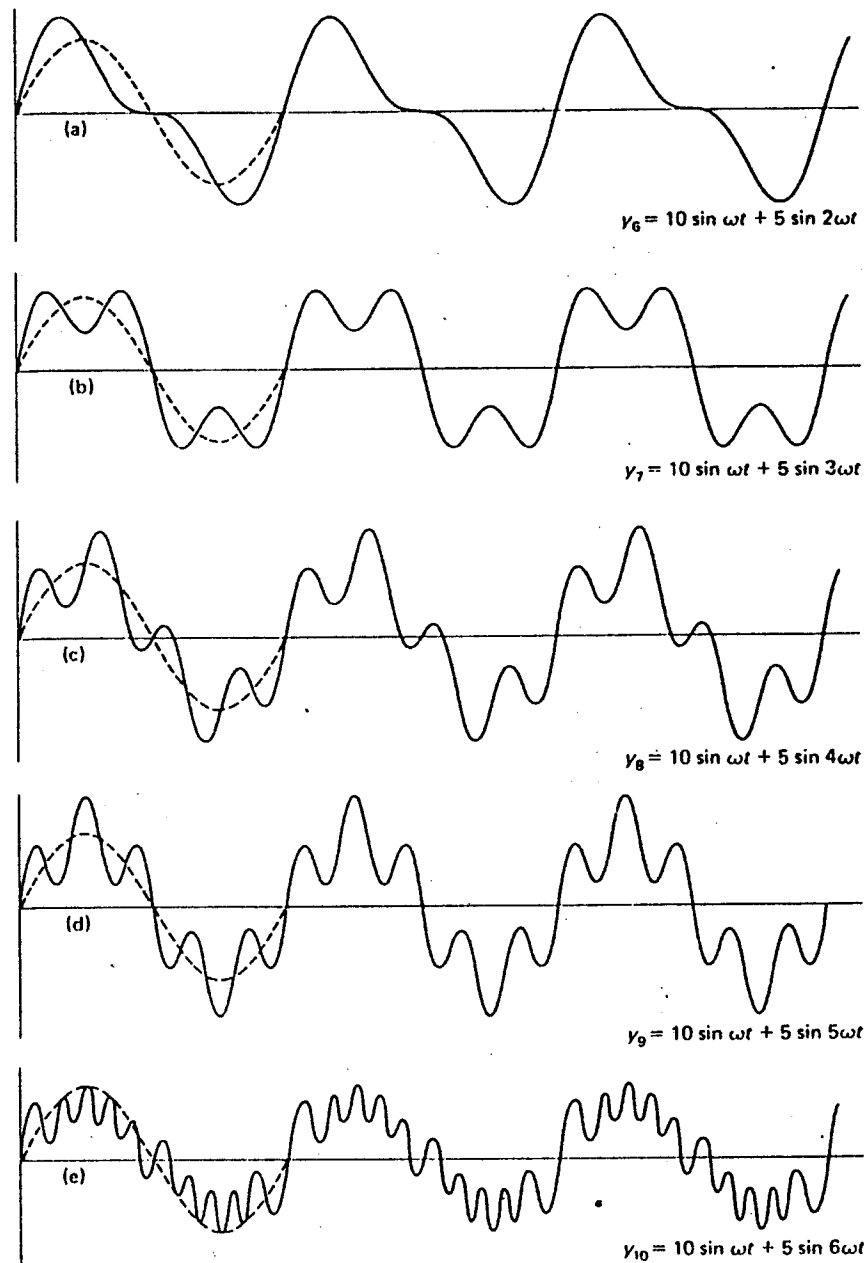


Fig.4.2 Examples of two component waveforms with second term of various relative frequencies.

Fig.4.3은 Phase의 변화에 따른 영향을 본것이다.

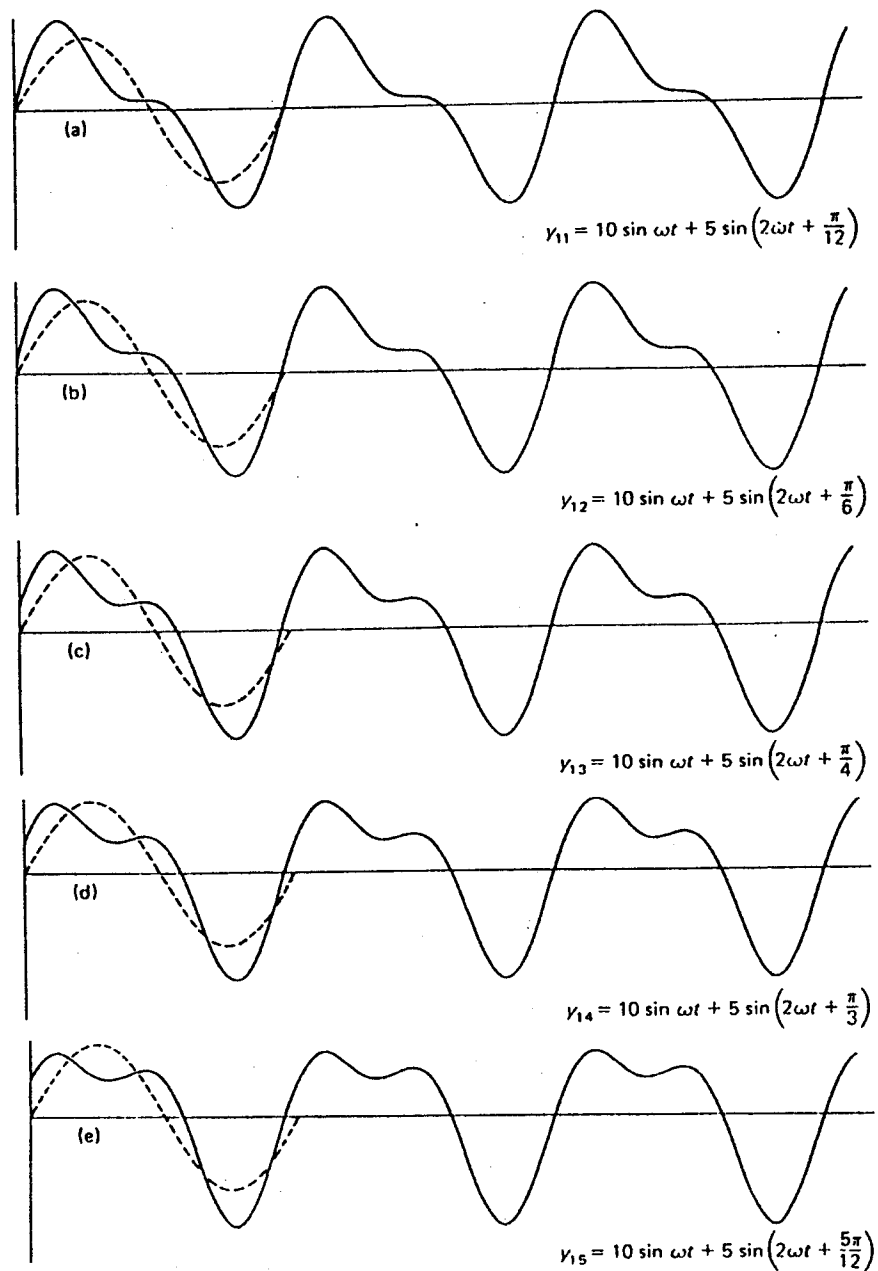


Fig.4.3 Examples of two component waveforms with the second harmonic having various degrees of phase shift

Fig.4.4는 주파수의 차가 현저할때 나타나는 현상으로서 진폭의 변화에 대한 영향도 나타나 있다.

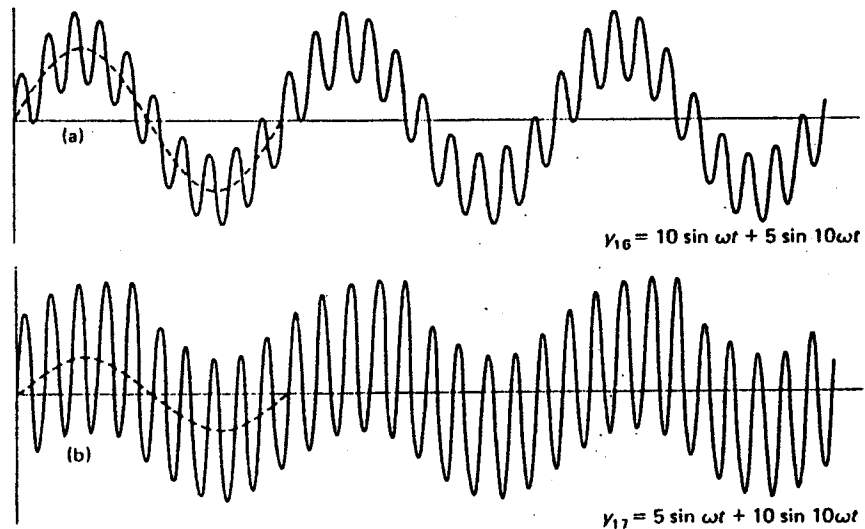


Fig.4.4 Examples of waveforms with the two components having considerably different frequencies

Fig.4.5는 주파수의 차가 근소할 경우에 나타나는 특이한 현상이다.

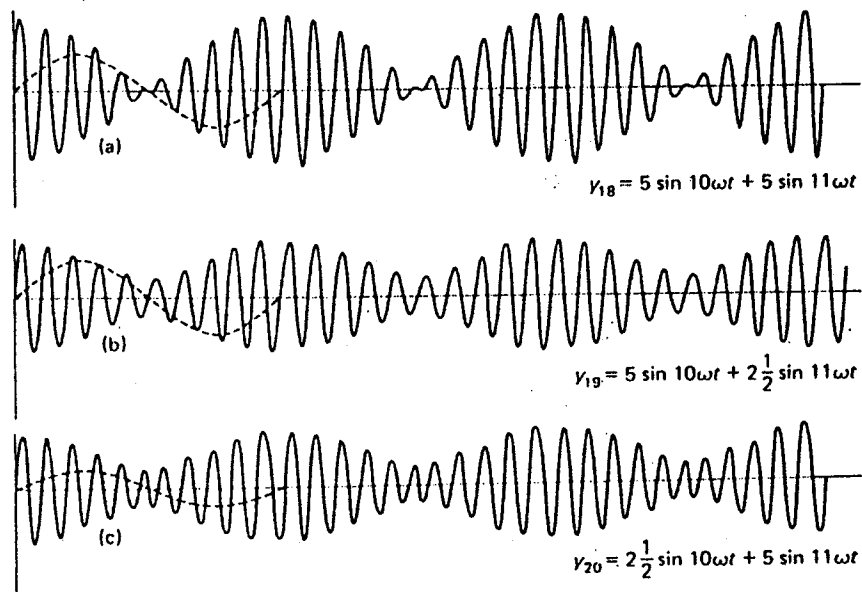
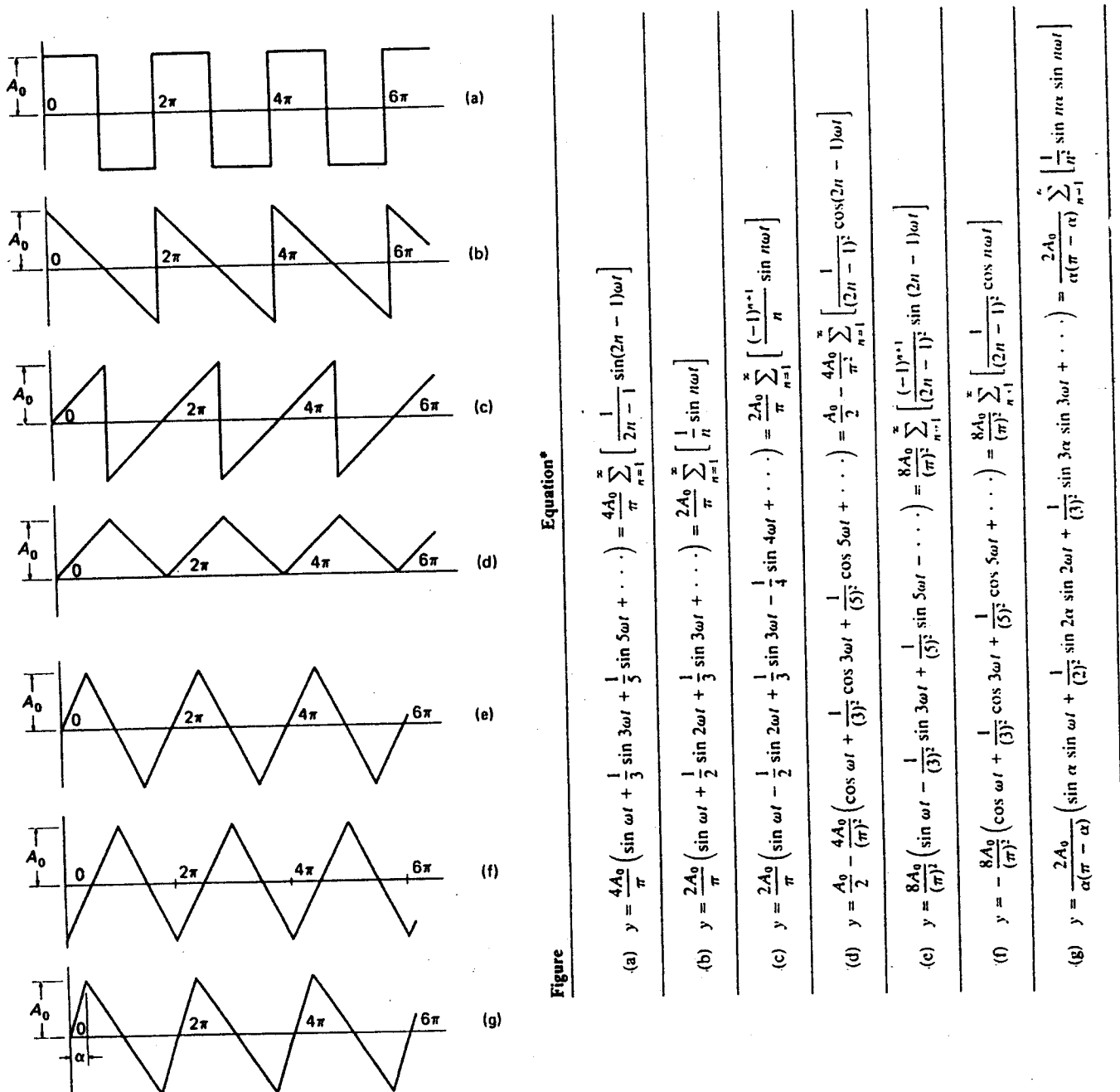


Fig.4.5 Examples of waveforms with two components having frequencies that are very nearly the same

현실적으로 Fig.4.4와 Fig.4.5의 경우를 많이 관찰하게 되는데, 신호측정시에 Line Noise(Power line의 60Hz Noise)의 영향이 관찰하려는 신호보다 큰 경우에 Fig.4.4(b)과 같은 현상이 나타나며, 주위의 고주파 Noise가 관찰하려는 신호에 결합될 경우 Fig.4.4(a)와 같은 모양이 된다. Fig.4.5의 경우는 「Beat(맥놀이)」로 불리는 현상으로서 많이 들었을 것이다. 다음 Fig.4.6은 여러가지 형상의 주기함수들을 Fourier Transform시킨것이다.



* n as used in these equations does not necessarily represent the harmonic order.

Fig.4.6 Various special waveforms of harmonic nature.

Fig.4.7은 Square wave의 경우, 급수항의 갯수를 바꾸면서 그 영향을 살펴본 것이다. Fig.4.6(a) 오른쪽의 급수항들 중에서 3개, 5개, 8개까지를 포함하는 것을 Fig.4.7의 (a), (b), (c)에 나타내었다.

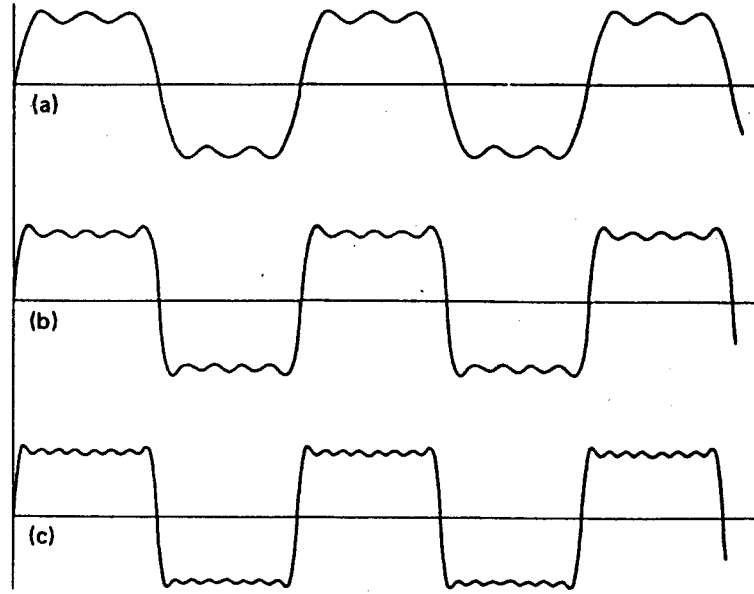


Fig.4.7 Plot of square wave function

- (a) Plot of first three terms only (includes the fifth harmonic)
- (b) Plot of the first five terms (includes the ninth harmonic)
- (c) Plot of the first eight terms (includes the fifteenth harmonic)

4.2 Frequency spectrum

지금까지 2가지 이상의 주파수를 가지는 함수들에서 주파수 및 진폭, Phase의 영향들을 살펴보았으며, 여러가지 함수들이 Fourier Transform에 의해 주파수와 진폭의 향으로 표시됨을 보았다.

여기서는 주파수와 진폭의 정보를 더욱 일목요연하게 볼 수 있도록 그래프로 나타내는데, 이를 Frequency spectrum이라 한다. Fig.4.8과 Fig.4.9에서는 앞의 여러 예들에 대한 Frequency Spectrum을 나타낸 것이며, Fig.4.10는 Square-wave에 대한 것이다. 여기서 특히 Square-wave는 System의 특성시험에 많이 쓰이고 있다.

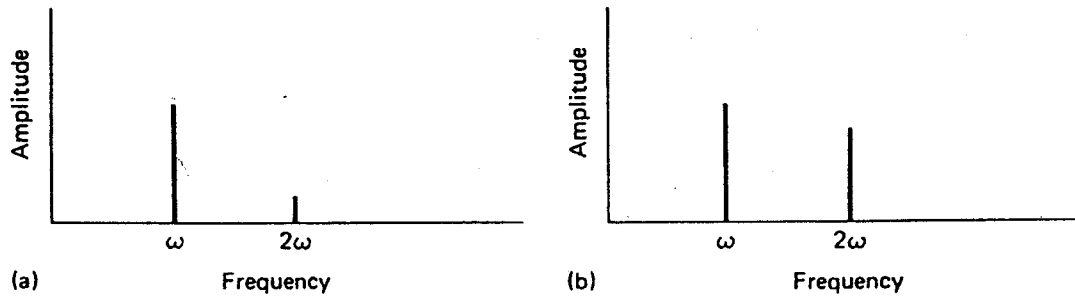


Fig. 4.8 Examples of Frequency Spectrum
 (a) Frequency Spectrum Corresponding to Fig. 4.1 (a)
 (b) Frequency Spectrum Corresponding to Fig. 4.1 (d)

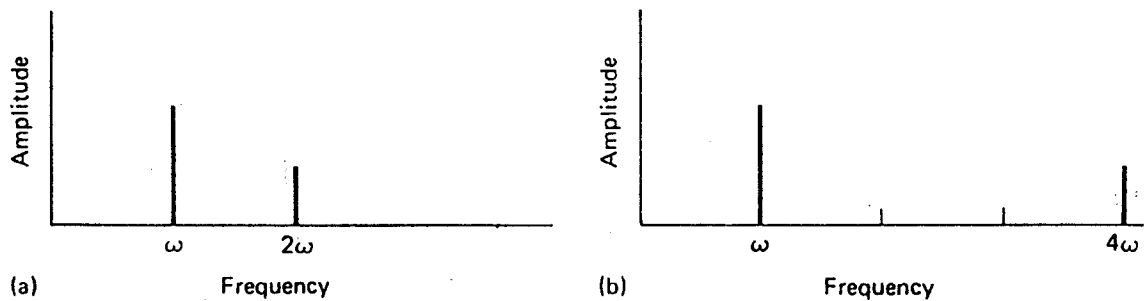


Fig. 4.9 Examples of Frequency Spectrum
 (a) Frequency Spectrum Corresponding to Fig. 4.2 (a)
 (b) Frequency Spectrum Corresponding to Fig. 4.2 (c)

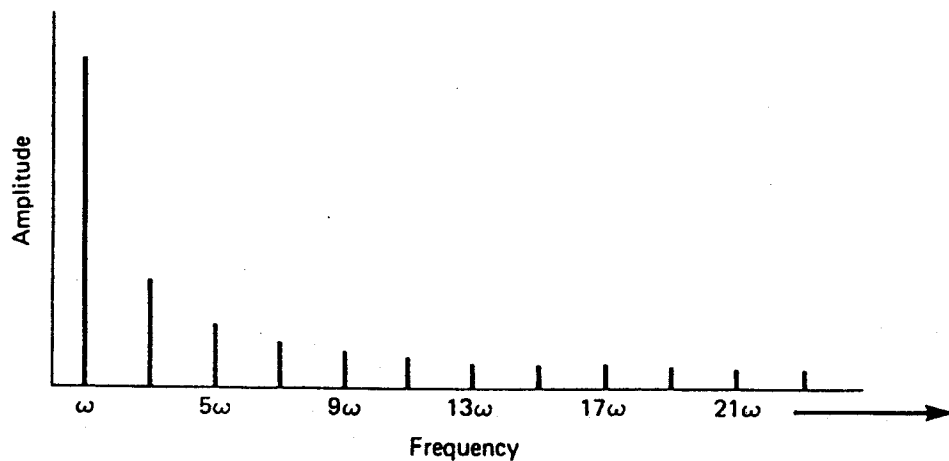


Fig. 4.10 Frequency spectrum for a square wave, Fig. 4.6(a)

4.3 Sampling Frequency와 Aliasing

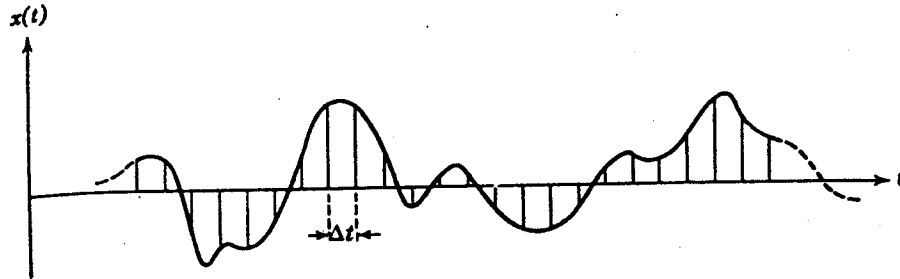


Fig.4.11 Sampling of a continuous record

아날로그 신호를 디지털 신호로 바꾸기 위하여 Sampling을 하는 경우, 일반적으로 같은 시간간격으로 데이터를 읽는 경우가 많다. 위 그림에서도 알 수 있듯이 h 가 작으면 작을수록 원형에 가까운 디지털신호를 얻을 수 있으나, 그만큼 비능률적일 것이다. 반면에 h 가 커지면 원래의 신호의 형상과 전혀 다른 엉뚱한 형상의 신호가 되어버릴 수 있는데, 이러한 현상을 Aliasing이라 한다. 달리는 마차바퀴의 방향이 반대로 보인다거나, 선풍기의 회전방향이 반복된다거나, 고속회전물체의 속도감각이 없다거나 하는 등의 현상이 이에 해당한다.

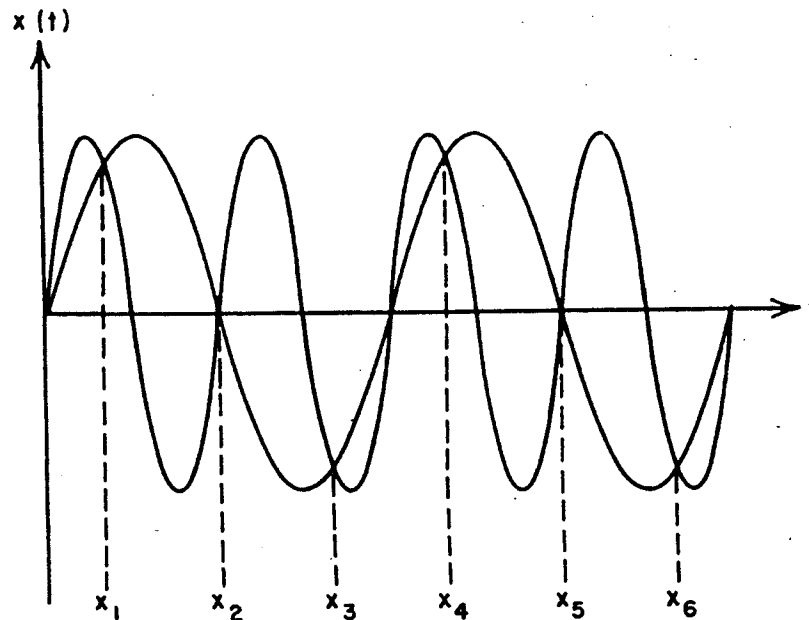


Fig.4.12 Illustration of aliasing problem

Fig.4.12을 보며 Aliasing을 논하여 보라. 그러면 어떤 신호를 해석하기 위

해서는 Sampling 주파수를 얼마로 해야 할 것인가? 최소한 신호 주파수의 두배는 되어야 한다. Sampling 시간간격을 h 라 할 때

$$f_c = \frac{1}{2h}$$

이고, 이때 f_c 를 Cutoff Frequency, Nyquist frequency, Folding Frequency 라 한다. 이 주파수 이상의 주파수에 대해서는 Fig.4.13와 같은 'Folding Down' 현상이 나타나며, Power Spectrum을 구하여 보면 Fig.4.14와 같은 'Aliased Spectrum'이 나타나게된다. Fig.4.13를 보면서 다음을 증명하여 보라.

$$\cos 2\pi(2nf_c \pm f)t = \cos 2\pi ft$$

$$\sin 2\pi(2nf_c \pm f)t = \sin 2\pi ft$$

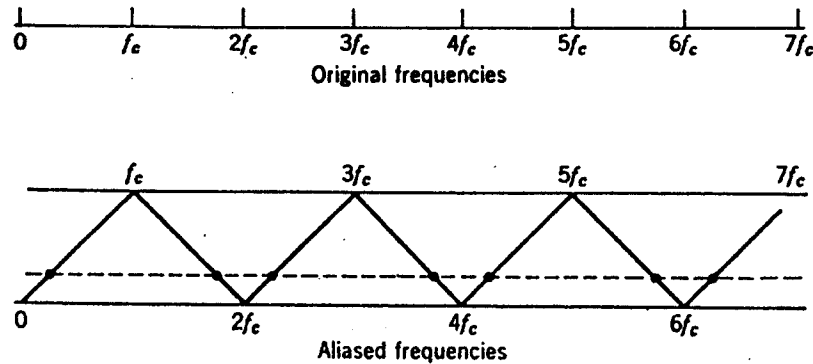


Fig.4.13 Folding about the Nyquist cutoff frequency f_c

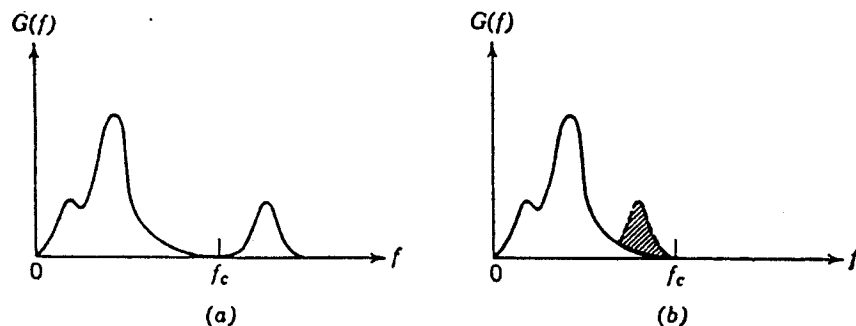
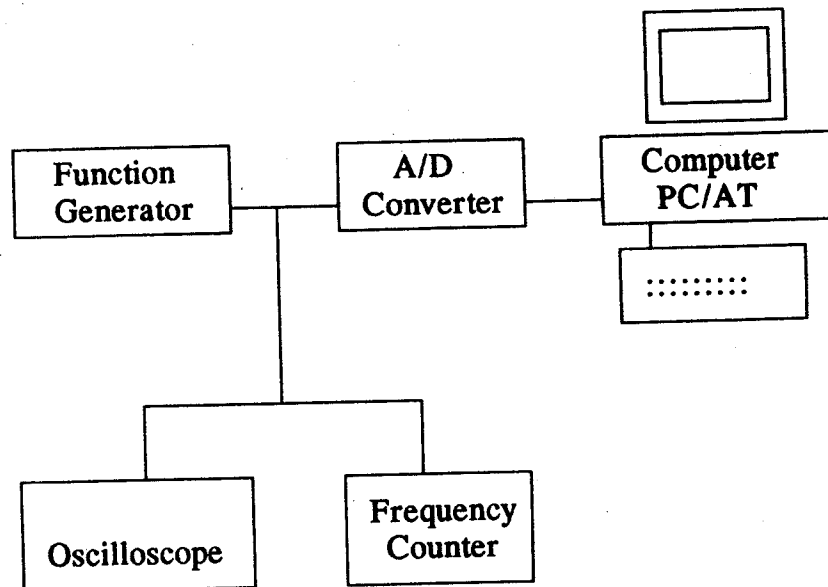


Fig.4.14 Aliased power spectrum due to folding
(a) True spectrum (b) Aliased spectrum

그러나 실제로 측정하려는 신호의 주파수를 일률적으로 말하기가 곤란하다면 어떻게 할 것인가? 먼저 대체적인 예상주파수(Turbulent인 경우에는 흔히 10kHz정도)의 두배 정도로 Sampling 주파수를 선정하고, 그 이상은 Filter를 써서 제거하여야 한다. 측정된 신호를 관찰하면서 Sampling 주파수와 Low Pass Filter의 Cut-off Frequency를 조정한다.

실 험 4 : Sampling과 Aliasing

1. Function Generator를 이용하는 경우



1-1. Function Generator의 Low Voltage 출력단자를 Computer의 A/D Converter 입력단자와 연결

1-2. Δt (Sampling Interval) 측정

(1) BASIC로 들어간다.

(2) ADC-DATA.BAS를 load 시킨다.

(3) ADC-DATA.BAS program 사이에 다음과 같은 program을 추가한다. 또한 추가된 program의 의미를 생각한다.

```

1025    TIME$ = "0 "
1045    FOR J=1 TO 30
1095    NEXT J
1112    PRINT TIME$
1113    STOP
  
```

(4) ADC-DATA.BAS를 RUN 시킨다. 몇분간을 기다리면 화면에 시간이 표시 될 것이다. 이로부터 Δt 를 구한다. [$\Delta t = t/(256 \times 30)$]

(5) 추가된 program을 지운다.

1-3. Signal sampling

- (1) BASIC로 들어간다.
- (2) F.G.에서 적당한(Cutoff Frequency보다 작은) sine wave를 발생시킨다.
(peak-to-peak 1~2 Volts , Frequency 10~60Hz 정도가 적당)
- (3) ADC-DATA.BAS를 load시킨다. (이 program은 Analog signal를 digital signal로 변환시킨다.)
- (4) ADC-DATA.BAS를 RUN시킨다. 이때 screen 상에 다음과 같은 화면이 나타난다.

Type output file name
?

이때 결과 DATA를 저장할 file name을 입력시킨다.

- (5) System으로 빠져 나온다.
- (6) FFT를 실행시킨다. FFT는 (4)번 file name에 저장되어 있는 time domain signal을 frequency domain으로 바꾸어주는 program이다. 이때 화면에 다음과 같이 나타난다.

TYPE INPUT FILE NAME
?

(4)번의 file name을 입력시킨다. 그러면 다음과 같은 화면이 나타난다.

TYPE OUTPUT FILE NAME
?

이때는 결과를 저장할 file name을 입력시킨다.

- (7) NUMBER OF DATA = 256
- (8) SAMPLING INTERVAL = Δt
- (9) peak value의 Frequency를 찾아낸다. (FFT 실행결과를 print로 출력해서 관찰)

$$\text{peak value} = \sqrt{\text{REAL}^2 + \text{IMAG}^2}$$

(10) Counter에 나타난 Frequency와 FFT 실행결과에서 얻은 Frequency를 비교한다.

(11) F.G.에서 발생시키는 Wave의 주파수를 바꾸어 가면서 위 실험을 반복한다.

(12) Input signal의 주파수 對 FFT-peak 주파수 관계의 graph를 그린다.
* sampling frequency는 고정

1-4. peak value frequency의 관찰

※ peak value frequency를 graph 상에서 관찰한다.

(1) File 중에 CGA.BAT과 PLOT.BAS라는 File이 있는가를 확인한다.

(2) System 상에서 다음과 같이 Type한다.

```
C:\USR\MON> CGA
```

(3) BASIC 안에서 PLOT.BAS를 load 시킨다.

(4) PLOT.BAS를 RUN시킨다. 그러면 다음과 같은 명령어가 나타난다.

```
INPUT YOUR DATA file NAME
?
```

이때 FFT를 실행시킨 결과를 저장한 file name을 입력시킨다.

(5) 화면에 graph와 peak-value frequency가 나타날 것이다. 이때 다른 FFT 실행 결과의 graph를 관찰하고 싶으면 'DO YOU WANT ANOTHER?'이라는 명령어 뒤에 'y'를 Type 한다.

1-5. Sampling wave의 관찰

샘플된 Data를 보고 싶으면, 즉 FFT를 실행시키기전의 파형을 보고자하면 다음과 같이 하면 볼 수 있다.

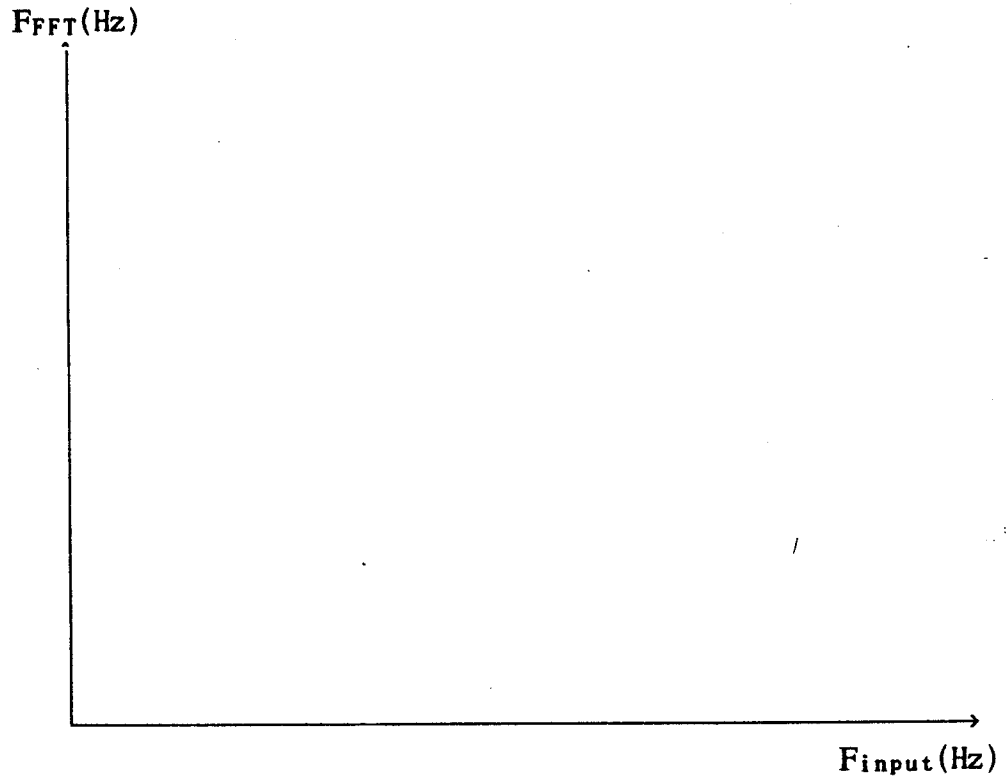
(1) BASIC안에서 WAVE.BAS를 load시킨다.

(2) WAVE.BAS를 RUN시킨다. 이때 input data file은 1-3.실험에서의 signal sampling을 한 결과를 저장한 file 즉 FFT를 실행시키기 전의 data file을 입력시킨다.

(3) Sampling wave가 화면에 나타날 것이다. 화면에 나타난 wave를 FFT한 결과가 1-4.번 실험이다.

* sine wave 말고도 삼각파, 방형파 등도 위의 실험결과를 화면으로 보면 재미있을 것이다.

Input Frequency				
FFT Frequency				



2. Software signal을 이용하는 경우

(1) PC의 BASIC로 들어간다.

(2) 『LOAD DATGEN』를 Type 한다.

(3) 『LIST DATGEN』를 Type 한다.

(4) Program Datagen를 대충 훑어 본다. 여기서 Δt 를 알아야 한다. Program 중에는 sine or cosine function으로 Output를 발생시키는 Program line을 발견할 수 있다. 만약 $SS = \sin(\omega t)$ 라면 signal발생 time interval Δt 는 t 가 될 것이다. 이 실험에서는

```
FOR I = 1 to 256
```

$$SS = \sin\left(2\pi f_1 \frac{I}{256}\right) + \sin\left(2\pi f_2 \frac{I}{256}\right)$$

이므로 $\Delta t = 1/256$ 의 간격으로 output 256개를 발생시킨다.

(5) 이 program을 RUN 시키면 output file name을 물어볼 것이다. output file name을 입력시키고 input frequency f_1, f_2 를 입력시킨다. 또한 실험1과 같이 WAVE.BAS를 이용하여 파형을 관찰하자.

(6) System으로 빠져나와 FFT를 실행시킨다.

* 여기서 time interval은 '0.003906' (=1/256)을 입력시켜야 한다.

(7) Program FFT의 output에서 peak value $\sqrt{(\text{Re}^2 + \text{Im}^2)}$ 에 해당되는 frequency f_1, f_2 를 구할 수 있을 것이다. 또한 PLOT.BAS를 이용하여 화면에 FFT결과를 볼 수 있을 것이다.

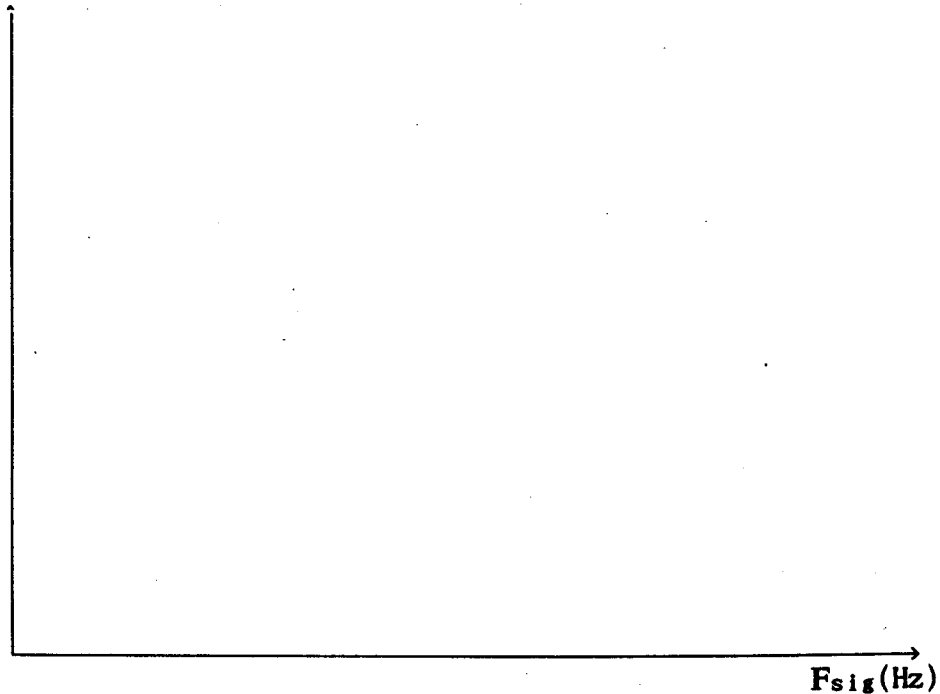
(8) (5)에서 $f_c = 1/(2\Delta t)$ 이므로 f_c 를 전후로 여러개의 frequency를 입력시켜본다.

(9) 위의 결과로부터 Aliasing을 관찰한다.

(10) f_{FFT} 對 f_{sig} 의 graph를 그려본다.

$f_c =$	$\Delta t =$			
Signal Frequency				
FFT Frequency				

$F_{\text{FFT}}(\text{Hz})$



DATGEN.BAS

```
10 PRINT "TYPE OUTPUT FILE NAME"
20 INPUT D$
30 OPEN D$ FOR OUTPUT AS #1
40 INPUT "TYPE INPUT FREQUENCIES, F1, F2 (Hz)"; DF1,DF2
50 FOR I = 1 TO 256
60 SS=SIN(I*3.1415965#*2*DF1/256)+SIN(I*3.1415965#*2*DF2/256)
70 PRINT SS
80 PRINT#1, SS
90 NEXT I
100 END
```

PLOT.BAS

```
1000 CLS:BEEP
2000 SCREEN 2
3000 KEY OFF
4000 DIM A(256),REAL(256),IMAG(256),Z(256)
5000 CLS:BEEP
6000 PRINT "INPUT YOUR DATA FILE NAME"
7000 INPUT A$
8000 OPEN A$ FOR INPUT AS #1
9000 INPUT#1,A$,S$,D$
10000 FOR I=1 TO 126
11000 INPUT#1,A(I),REAL(I),IMAG(I)
12000 NEXT I
13000 CLOSE#1
14000 FOR J= 1 TO 126
15000 Q=REAL(J)*REAL(J)
16000 W=IMAG(J)*IMAG(J)
17000 Z(J)=SQR(Q+W)*1.3
18000 NEXT J
19000 CLS
20000 LINE(10,2)-(10,170)
21000 LINE(10,170)-(580,170)
22000 LINE(580,170)-(580,2)
23000 LINE(580,2)-(10,2)
23100 FOR I=1 TO 23
23110 J=I*24+10
23120 LINE(J,170)-(J,172)
23150 NEXT I
```



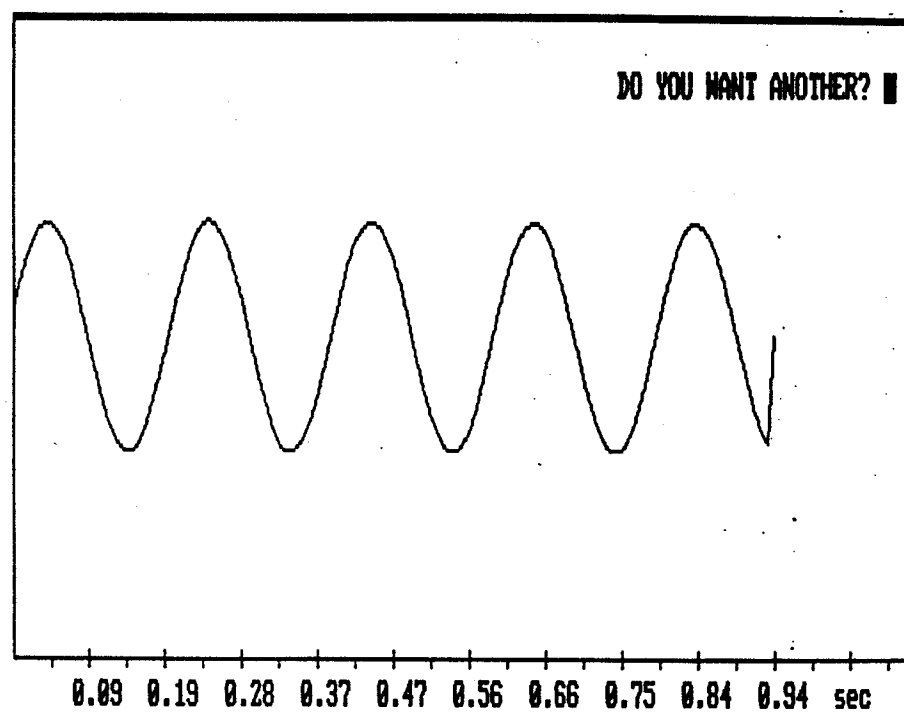
```
23151 FOR I=1 TO 11
23152 K=I*48+10
23153 LINE(K,168)-(K,172)
23154 NEXT I
23190 LOCATE 23,70 :PRINT"Hz"
23200 FOR I=1 TO 10
23205 K=I*12:FREQ=INT(A(K))
23210 J=I*6 :LOCATE 23,J:PRINT FREQ
23500 NEXT I
24000 K=5
25000 LOCATE K,50 :PRINT"PEAK-VALUE FREQ."
26000 FOR I=1 TO 125
27000 Z1=INT(Z(I))
28000 Z2=INT(Z(I+1))
29000 X1=I*4+10
30000 Y1=170-Z1
31000 X2=X1+4
32000 Y2=170-Z2
33000 LINE(X1,Y1)-(X2,Y2)
34000 IF Y1<80 THEN K=K+1:GOTO 41000
35000 NEXT I
36000 LOCATE 10,50 :PRINT "DO YOU WANT ANOTHER"
37000 LOCATE 10,69
38000 INPUT A$
39000 IF (A$="y")OR(A$="Y")THEN GOTO 5000
40000 SCREEN 0,0,0
40100 END
41000 LOCATE K+1,52
42000 PRINT A(I); "Hz"
43000 GOTO 35000
```

WAVE. BAS

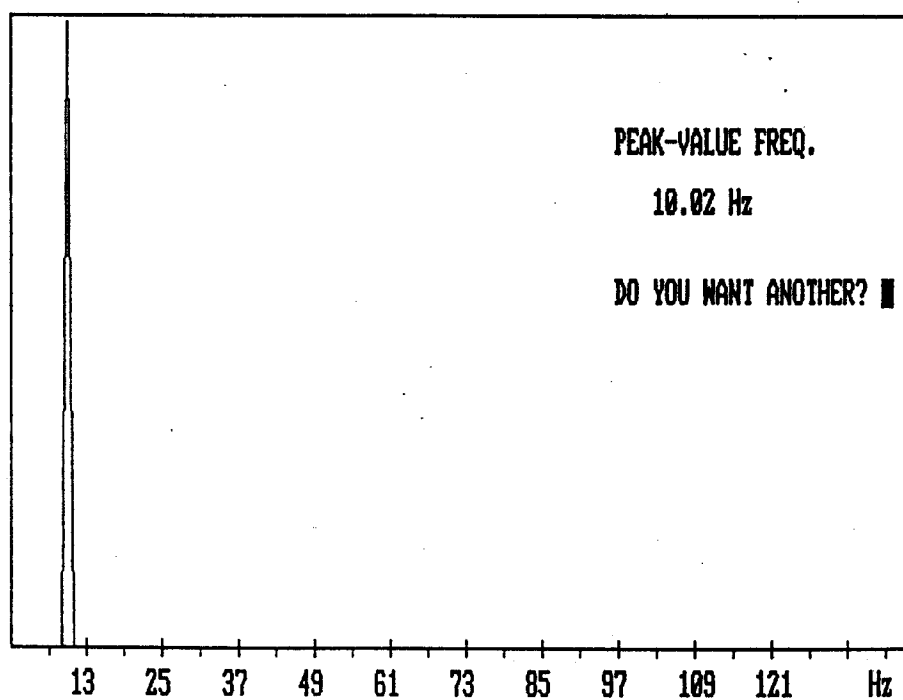
```
1000 CLS:BEEP
2000 SCREEN 2
3000 KEY OFF
4000 DIM TEMP(256)
5000 CLS:BEEP
6000 PRINT "INPUT YOUR DATA FILE NAME"
7000 INPUT A$
7100 PRINT:PRINT:PRINT
```

```
7500 PRINT "INPUT TIME INTERVAL"
7600 INPUT TIME
8000 OPEN A$ FOR INPUT AS #1
10000 FOR I=1 TO 120
11000 INPUT#1,TEMP(I)
12000 NEXT I
13000 CLOSE#1
19000 CLS
20000 LINE(10,2)-(10,170)
21000 LINE(10,170)-(580,170)
22000 LINE(580,170)-(580,2)
23000 LINE(580,2)-(10,2)
23050 LINE(580,1)-(10,1)
23100 FOR I=1 TO 23
23110 J=I*24+10
23120 LINE(J,170)-(J,172)
23150 NEXT I
23152 FOR I=1 TO 11
23153 K=I*48+10
23154 LINE(K,168)-(K,172)
23155 NEXT I
23190 LOCATE 23,67 :PRINT"sec"
23200 FOR I=1 TO 10
23205 K=I*12:TIME1=TIME*24*I
23210 J=I*6 :LOCATE 23,J:PRINT USING"##.##";TIME1
23500 NEXT I
26000 FOR I=1 TO 120
27000 TEM1=INT(TEMP(I)*30)
29000 X1=4*I+6
30000 Y1=85-TEM1
30100 X2=X1+4:TEM2=INT(TEMP(I+1)*30): Y2=85-TEM2
31000 LINE(X1,Y1)-(X2,Y2)
35000 NEXT I
36000 LOCATE 3,50 :PRINT "DO YOU WANT ANOTHER"
37000 LOCATE 3,69
38000 INPUT A$
39000 IF (A$="y")OR(A$="Y")THEN GOTO 5000
40000 SCREEN 0,0,0
40100 END
```

참고 4.1 Frequency Spectrum의 예, $f_1 = 10$ Hz, $f_2 = 0$ Hz

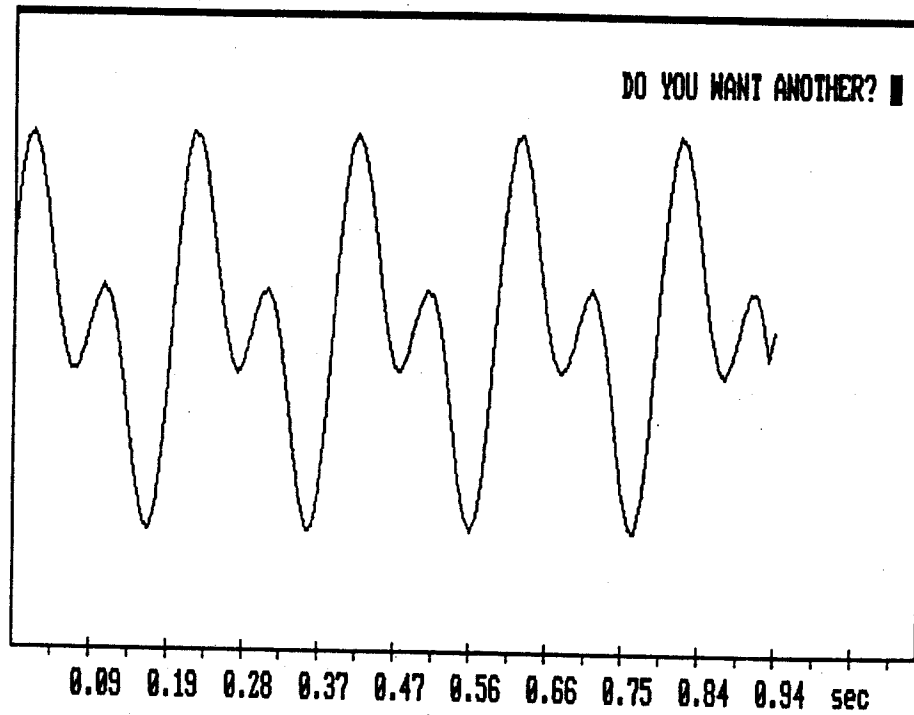


(a) Sampling Signal

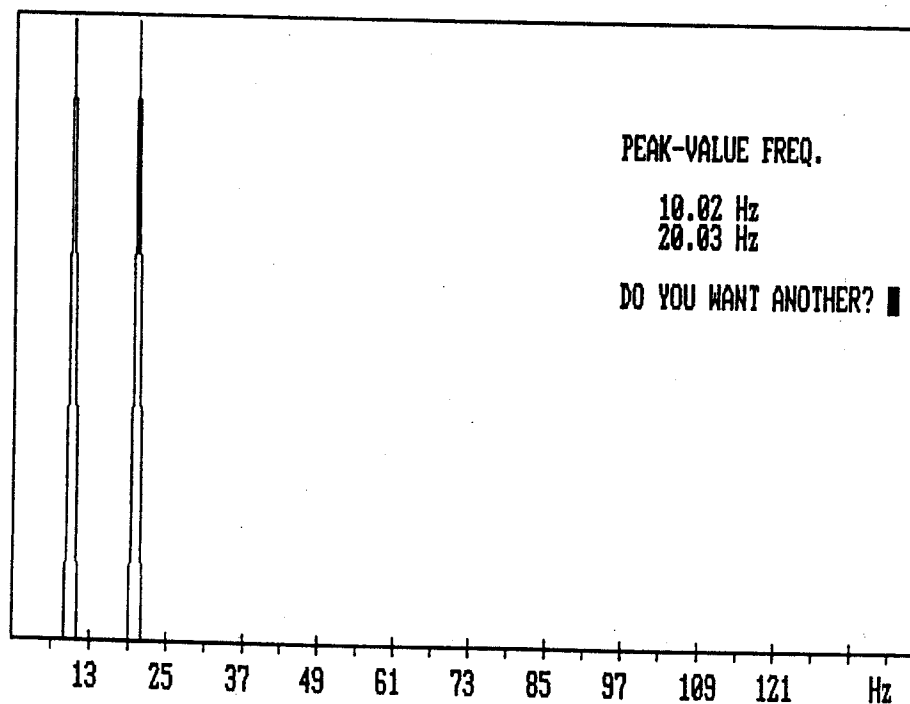


(b) FFT Output

참고 4.2 Frequency Spectrum의 예, $f_1 = 10 \text{ Hz}$, $f_2 = 20 \text{ Hz}$



(a) Sampling Signal



(b) FFT Output