디지털통신시스템설계

Digital Communication System Capstone Design

ICE4009-001

5 주차 실습과제

정보통신공학과

12191765

박승재

Introduction

아날로그 신호를 샘플링할 때, Sampling Rate 를 신호가 가진 최고 주파수 성분의 2 배 이상으로 설정해야 원 신호의 손실이 발생하지 않는다. 이것을 Nyquist sampling theorem 이라 한다.

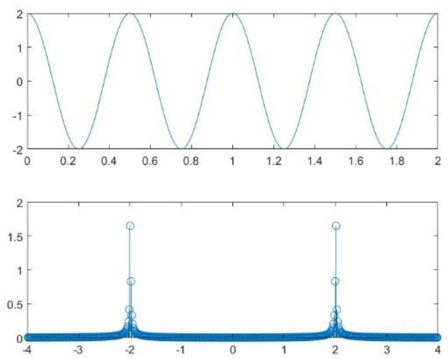
```
% Original Signal
f_s = 1000; % sampling frequency
t_start = 0; % sampling start time
t_end = 20 * pi; % sampling end time

t_s = 1 / f_s; % sampling interval
t = t_start:t_s:t_end;

x = 2 * cos(2 * pi * 2 * t); % original signal
```

Matlab 에서는 연속적인 신호를 만들 수 없기 때문에 f=1000 으로 잡고 매우 촘촘히 샘플링해 아날로그 신호를 흉내낼 수 있다.

```
X = abs(fftshift(fft(x))); % X(f) = FT{x(t)}
X_f = linspace(-f_s / 2, f_s / 2, length(x));
```

fft 는 푸리에 변환을 수행하는 함수이고, fftshift 는 반환 값의 순서를 바꾸는 함수이다. fft 를 수행하면 주파수는 오름차순이 아니라 양수->음수 순으로 나오게 되는데, fftshift 는 이것을 오름차순으로 바꿔준다. FFT 는 복소수 형태로 값이 나오는데 abs 는 허수 부분을 없애고 실수 부분만 나타낼 수 있도록 변환한다.

```
figure()
subplot(2, 1, 1)
plot(t, x)
axis([0 2 -2 2])
subplot(2, 1, 2)
stem(X_f, 2 * X / length(X))
xlim([-4 4])
```



출력된 그래프를 확인해보면 f=2 와 f=-2 에서 피크가 찍히는 것을 확인할 수 있다. $2\cos 4\pi t$에서 f=2 이기 때문에 ± 2에서 피크가 찍힌다.

```
% Sampled Signal
f2_s = 10; % sampling frequency

t2_s = 1 / f2_s; % sampling interval
t2 = t_start:t2_s:t_end;

x2 = x(1:floor(t2_s / t_s):length(t)); % Sampled signal
```

```
x3 = zeros(size(x)); % Sampled signal (same length)
for idx = 1:length(x)
    if mod(idx, floor(t2_s / t_s)) == 1
        x3(idx) = x(idx);
    end
end
```

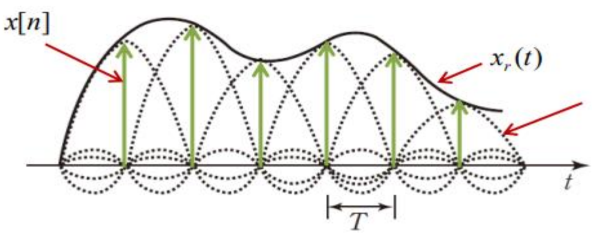
x 를 10Hz 로 샘플링한 x2 와 x3 이다. x2 는 x 에서 특정 시각에서의 값들만 뽑아오는 형태로 샘플링 했고, x3 는 샘플링 되지 않은 부분은 0 으로 입력하여 x 와 크기가 같다.

```
X3 = abs(fftshift(fft(x3))); % X3(f) = FT{x3(t)}
X3_f = linspace(-f2_s / 2, f2_s / 2, length(x3));
```

푸리에 변환은 x 와 동일하게 진행하면 된다. FFT 에 넣을 대상은 샘플링 되지 않은 부분은 0 으로 입력된 x3 이다.

```
% Reconstructed Signal
y = zeros(length(t2), length(t));
for idx = 1:length(t2)
    y(idx, :) = x2(idx) * sinc((t - (t(1) + (idx - 1) * t2_s)) / t2_s);
end
y = sum(y);

Y = abs(fftshift(fft(y))); % Y(f) = FT{y(t)}
Y_f = linspace(-f_s / 2, f_s / 2, length(y));
```

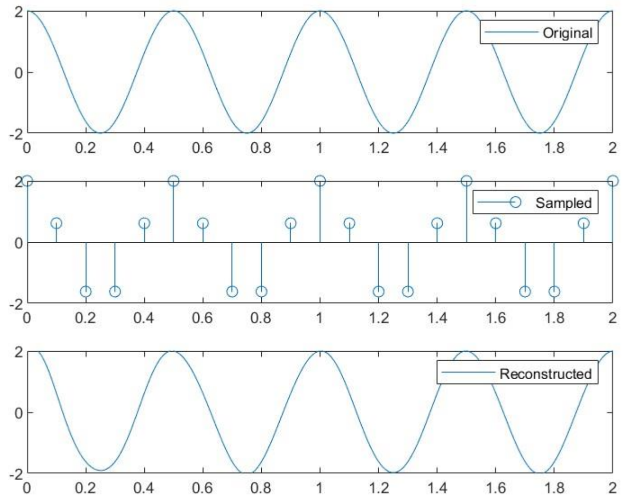
신호의 복원은 sinc 함수를 통해 수행한다. $y(t) = \sum_{n=-\infty}^{\infty} x_s(n)\, sinc\left(\frac{(t-nT_s)}{T_s}\right)$ 임을 이용하여 아래 사진과 같이 sinc 결과를 모두 합치게 되면 원래의 신호를 얻어낼 수 있다.



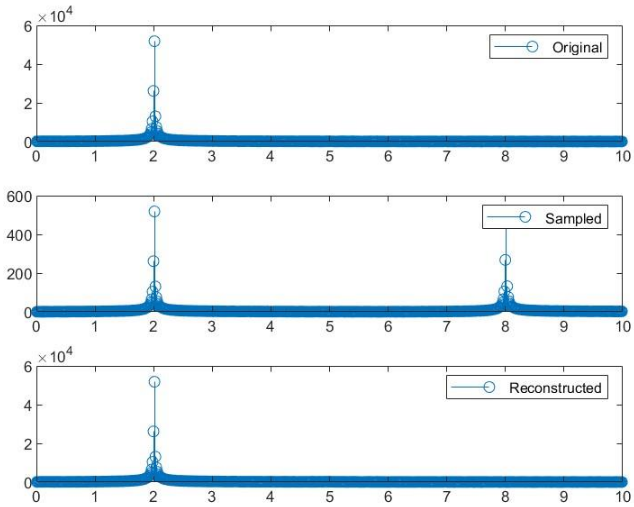
sinc 함수는 T 간격으로 0 을 지나며 출렁이기 때문에 다른 x_s(n)에서는 0 을 가지게 된다. 따라서 샘플 값에서는 해당 값만 반영되고 다른 값은 모두 0 이며, 샘플값 사이에서는 모든 sinc 값이 합해서 Interpolation 이 된다.

```
figure()
subplot(3, 1, 1)
plot(t, x)
axis([0 2 -2 2])
legend('Original')
subplot(3, 1, 2)
stem(t2, x2)
axis([0 2 -2 2])
legend('Sampled')
subplot(3, 1, 3)
plot(t, y)
axis([0 2 -2 2])
legend('Reconstructed')
```

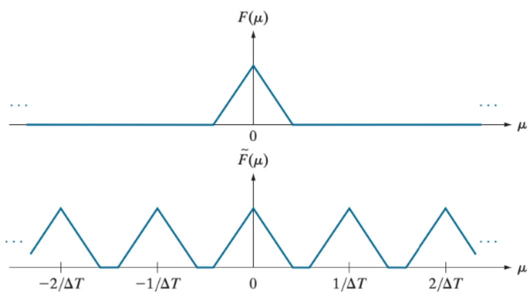
```
figure()
subplot(3, 1, 1)
stem(X_f, X)
legend('Original')
xlim([0 10])
subplot(3, 1, 2)
stem(X_f, X3)
legend('Sampled')
xlim([0 10])
subplot(3, 1, 3)
stem(Y_f, Y)
legend('Reconstructed')
xlim([0 10])
```



시간 영역에서의 신호이다. 원 신호와 복원된 신호가 거의 일치하는 것을 확인할 수 있다. 복원된 신호의 시작 부분이 원 신호와 조금 다른 것은 t<0 부분의 정보가 없기 때문에 sinc 합산에서 그 부분은 반영되지 않아서 오차가 커진 것으로 보인다. 복원된 신호의 첫 부분을 제외하면 원 신호와 복원된 신호는 거의 일치한다.



주파수 영역에서의 신호이다. 샘플링된 신호를 보면 f=2 와 f=8 에서 피크가 찍히는 것을 확인할 수 있다. 원 신호는 f=2 와 f=-2 에서 피크가 찍히는데, 신호를 샘플링하게 되면 샘플링 주파수 간격만큼 신호가 복사->이동되어 추가적으로 찍히게 된다. f=8 에 피크가 찍히는 이유는 f=-2 에서 찍히는 피크가 +10 만큼 이동되어 f=8 에 찍혔기 때문이다. 샘플링된 신호를 토대로 복원한 신호 그래프에서는 문제없이 f=2 에서만 피크가 찍히는 것을 확인할 수 있다.



Problem

다음의 원신호를 샘플링하고, 원신호를 복원하여라.

원신호: 연속된 시간 범위 [0, 2]에서의 두 개의 cosine 신호의 합으로 이루어진 신호

$x(t) = 2\cos 6\pi t + \cos 8\pi t$ (원신호의 샘플링 주파수: 1000Hz 로 설정)

1. 나이퀴스트 주파수를 제시하여라.
2. 나이퀴스트 주파수 이상으로 샘플링 할 때와 그 미만으로 샘플링 할 때의 샘플링 주파수를 구분하여 원신호를 복원하여라.

   원본 신호, 샘플링 신호, 복원 신호를 시간영역에서 그려라.

   원본 신호, 샘플링 신호, 복원 신호를 주파수영역에서 그려라.

Result

Problem 1

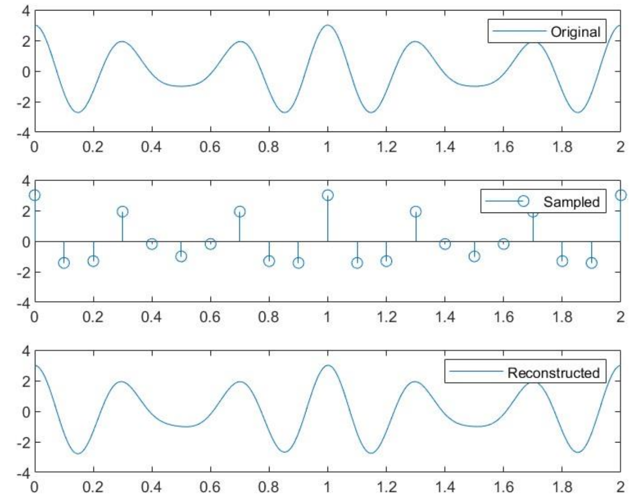
$x(t) = 2\cos 6\pi t + \cos 8\pi t$의 최대 주파수는 4Hz 이다. Nyquist 주파수는 원 신호의 최대 주파수보다 2 배 이상 커야 하므로 8Hz 이다.

Problem 2

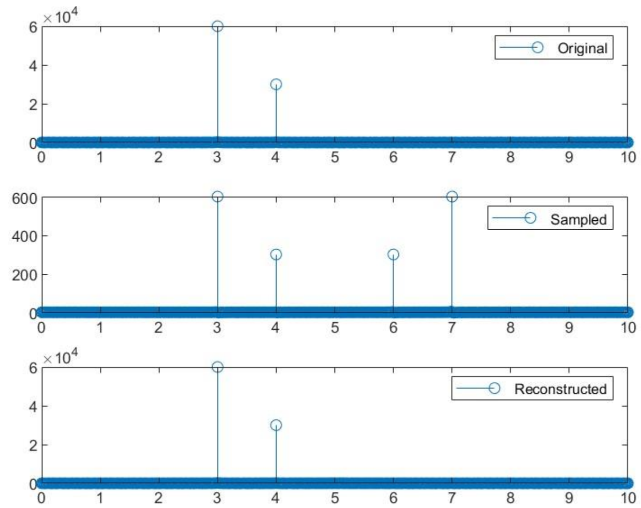
```
% Original Signal
f_s = 1000; % sampling frequency
t_start = 0; % sampling start time
t_end = 60; % sampling end time

t_s = 1 / f_s; % sampling interval
t = t_start:t_s:t_end;

x = 2 * cos(6 * pi * t) + cos(8 * pi * t); % original signal

X = abs(fftshift(fft(x))); % X(f) = FT{x(t)}
X_f = linspace(-f_s / 2, f_s / 2, length(x));

% Sampled Signal
f2_s = 10; % sampling frequency

t2_s = 1 / f2_s; % sampling interval
```

```
t2 = t_start:t2_s:t_end;

x2 = x(1:floor(t2_s / t_s):length(t)); % Sampled signal

x3 = zeros(size(x)); % Sampled signal (same length)
for idx = 1:length(x)
    if mod(idx, floor(t2_s / t_s)) == 1
        x3(idx) = x(idx);
    end
end

X3 = abs(fftshift(fft(x3))); % X3(f) = FT{x3(t)}
X3_f = linspace(-f2_s / 2, f2_s / 2, length(x3));

% Reconstructed Signal
y = zeros(length(t2), length(t));
for idx = 1:length(t2)
    y(idx, :) = x2(idx) * sinc((t - (t(1) + (idx - 1) * t2_s)) / t2_s);
end
y = sum(y);

Y = abs(fftshift(fft(y))); % Y(f) = FT{y(t)}
Y_f = linspace(-f_s / 2, f_s / 2, length(y));

figure()
subplot(3, 1, 1)
plot(t, x)
axis([0 2 -4 4])
legend('Original')
subplot(3, 1, 2)
stem(t2, x2)
axis([0 2 -4 4])
legend('Sampled')
subplot(3, 1, 3)
plot(t, y)
axis([0 2 -4 4])
legend('Reconstructed')

figure()
subplot(3, 1, 1)
stem(X_f, X)
legend('Original')
xlim([0 10])
subplot(3, 1, 2)
stem(X_f, X3)
legend('Sampled')
xlim([0 10])
```

```
subplot(3, 1, 3)
stem(Y_f, Y)
legend('Reconstructed')
xlim([0 10])
```



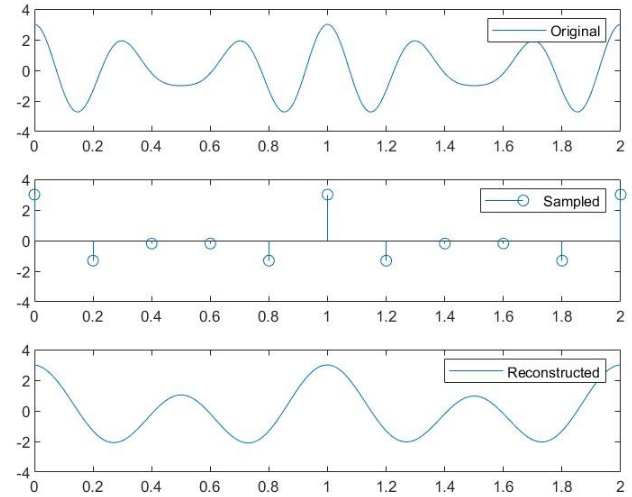
원 신호와 복원된 신호가 거의 일치하는 것을 확인할 수 있다.



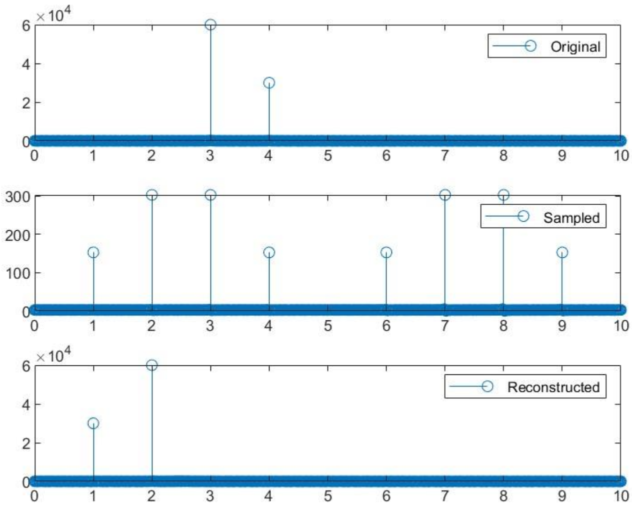
주파수 영역도 원 신호와 복원된 신호가 일치한다. $x(t) = 2\cos(2\times 3\pi t) + \cos(2\times 4\pi t)$ 이기 때문에 f=3 과 f=4 에서 피크가 찍히는 것도 확인할 수 있고, $\cos(2\times 3\pi t)$의 계수가 2 이기 때문에 f=3 이 f=4 보다 높게 그려지는 것도 확인할 수 있었다. 샘플링된 신호에서 f=6 은 f=-4 가 +10 만큼 이동해서 그려진 것이고, f=7 은 은 f=-3 이 +10 만큼 이동해서 그려진 것이다.

```
% Sampled Signal
f2_s = 5; % sampling frequency
```

f2_s 를 Nyquist 주파수 이하인 5Hz 로 내리고 다시 그래프를 그려보았다.



이번에는 원 신호와 복원된 신호가 다르게 나오는 것을 확인할 수 있다.



주파수 영역도 마찬가지로 다르게 나온다. 샘플링된 신호에서는 f=1 과 f=2 에서 피크가 추가로 찍히고, 그 결과 복원된 신호에는 f=1 과 f=2 에서만 피크가 찍히게 된다. 이는 원 신호와 복원된 신호의 주파수 성분이 다르다는 것을 의미한다.

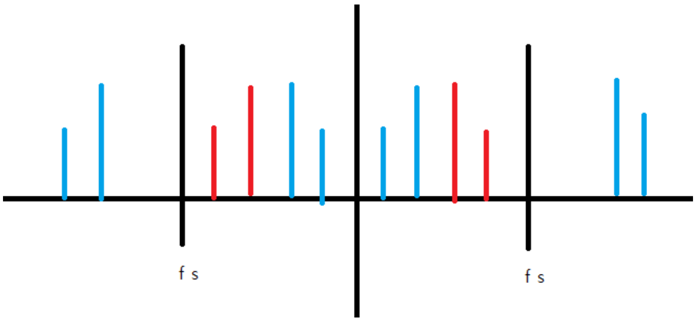
Conclusion

Matlab 을 통해 원 신호를 만들어 샘플링하고, 신호를 복원하면서 원 신호와 복원된 신호가 같은지 확인하는 실습이었다. fft 함수가 어떻게 동작하는지 시각적으로 확인할 수 있었고, 주파수 영역에서의 비교를 통해 신호가 어떻게 달라지는지 관찰할 수 있었다. 신호를 샘플링하게 되면,

주파수 영역에서는 샘플링 주파수만큼 신호가 이동되어 추가로 그려지는데 이것은 실습에서 f=8, 과제에서 f=3 과 f=4 에서 피크가 찍히는 것으로 확인할 수 있었다.

해당 신호를 sinc 함수를 이용해 복원하게 되면, 다시 원래의 신호 성분만 그려지는 것을 확인할 수 있다. 만약 샘플링 주파수가 원 신호의 최대 주파수의 2 배보다 작다면, 신호를 복원했을 때 원래의 신호가 나오지 않는데 이것을 Nyquist Theorem 이라 한다. 원래의 신호가 나오지 않는 이유는 원래 자리에 있던 신호와 주파수만큼 평행 이동한 신호가 겹치기 때문이다. 과제에서 5Hz 로 샘플링 했을 때 f=1 과 f=2 에서 피크가 찍힌 이유는, 원래 신호는 f=-4,-3,3,4 에서 피크가 찍히지만, f=-4,-3 에서 +5(샘플링 주파수)를 하게 되면 f=1,2 가 되기 때문이다.

원 신호의 최대 주파수의 2 배만큼 샘플링을 하면 샘플링한 신호들끼리 서로 겹치지 않는다.



하지만, 2 배 이하의 주파수로 샘플링하면 신호들끼리 서로 겹친다. 이 상태에서 신호를 복원하게 되면 원 신호와 다른 신호가 나오게 된다.