

INHA University

dept of Information & Communication
Engineering

ICE3009
통신이론 설계

Final Project

Multiplexing
Transmission & Receiving

12161725 류덕형 (Duckhyung Ryu)

Introduction	3
Design Illustration	3
Simulation Result	4
w1=100kHz w2=30kHz w3=65kHz	4
RC1=2e-4 RC2=2e-4 RC3=2e-4	4
w3=e2로 수정했을 때	6
RC1=2e-5로 수정했을 때	6
RC1=2e-3으로 수정했을 때	8
Reflection	8

Introduction

세개의 오디오 파일을 변조해서 하나에 신호로 전송하고, 이를 다시 따로따로 복조해 오디오 신호를 비교한다.

첨부파일:

final_project.m

Final_Report.pdf

result1.wav

result2.wav

result3.wav (result오디오파일은 첫번째 simulation결과로 나온(별 문제없이 복조가 된) 파일들)

Design Illustration

```
to=1/44100;    ts=to/8;
t=0:to:10;    t2=0:ts:10; N=length(t2);

%Transmission part
X=audioread('song1.wav'); xol=transpose(X(1:44100*10+1,1));
X=audioread('song2.wav'); xo2=transpose(X(44100*15+1:44100*25+1,1));
X=audioread('song3.wav'); xo3=transpose(X(44100*30+1:44100*40+1,1));

x1=interp1(t,xol,t2);
x2=interp1(t,xo2,t2);
x3=interp1(t,xo3,t2);

w1=100e3*pi;    w2=40e3*pi;    w3=65e3*pi;
xc1=x1.*cos(w1*t2);
xc2=x2.*cos(w2*t2);
xc3=x3.*cos(w3*t2);
xc=xc1+xc2+xc3;

%Receiving part from here
R1=xc.*cos(w1*t2);
R2=xc.*cos(w2*t2);
R3=xc.*cos(w3*t2);

RC1=2e-4;    RC2=2e-4;    RC3=2e-4;
y1=zeros(1,N);    y2=zeros(1,N);    y3=zeros(1,N);
for i=2:N %LPF
    y1(i)=(RC1-ts)/RC1*y1(i-1)+ts/RC1*R1(i-1);
    y2(i)=(RC2-ts)/RC2*y2(i-1)+ts/RC2*R2(i-1);
    y3(i)=(RC3-ts)/RC3*y3(i-1)+ts/RC3*R3(i-1);
end
y1=y1-mean(y1); %erase DC component
y2=y2-mean(y2);
y3=y3-mean(y3);

y1=interp1(t2,y1,t);
y2=interp1(t2,y2,t);
y3=interp1(t2,y3,t);

y1=y1/max(y1); %normalize
y2=y2/max(y2);
y3=y3/max(y3);

figure(1);
subplot(2,1,1); plot(t,xol); title('original');
subplot(2,1,2); plot(t,y1); title('received');
```

```
figure(2);
subplot(2,1,1); plot(t,xo2); title('original');
subplot(2,1,2); plot(t,y2); title('received');
figure(3);
subplot(2,1,1); plot(t,xo3); title('original');
subplot(2,1,2); plot(t,y3); title('received');

audiowrite('Result1.wav',y1,1/to);
audiowrite('Result2.wav',y2,1/to);
audiowrite('Result3.wav',y3,1/to);
```

Carrier주파수와 LPF의 RC값은 각각 w_1, w_2, w_3 , RC_1, RC_2, RC_3 값을 바꿈으로써 수정할 수 있다.

Song1은 처음 10초, 2는 15초~25초, 3은 30~40초 부분을 샘플링했으며 이는 원한다면 audioread 부분의 transpose함수 parameter를 바꿔서 수정할 수 있다.

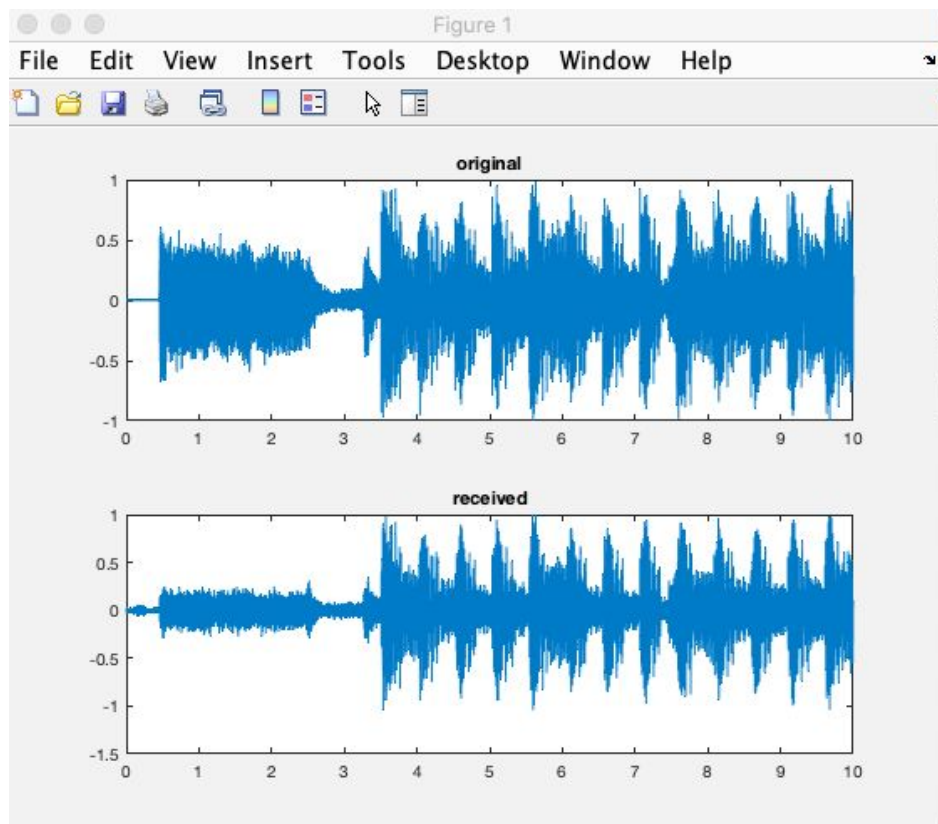
코드안에 주석대로 line 17까지는 Transmission, 그 다음부터는 Receiving 부분이다.

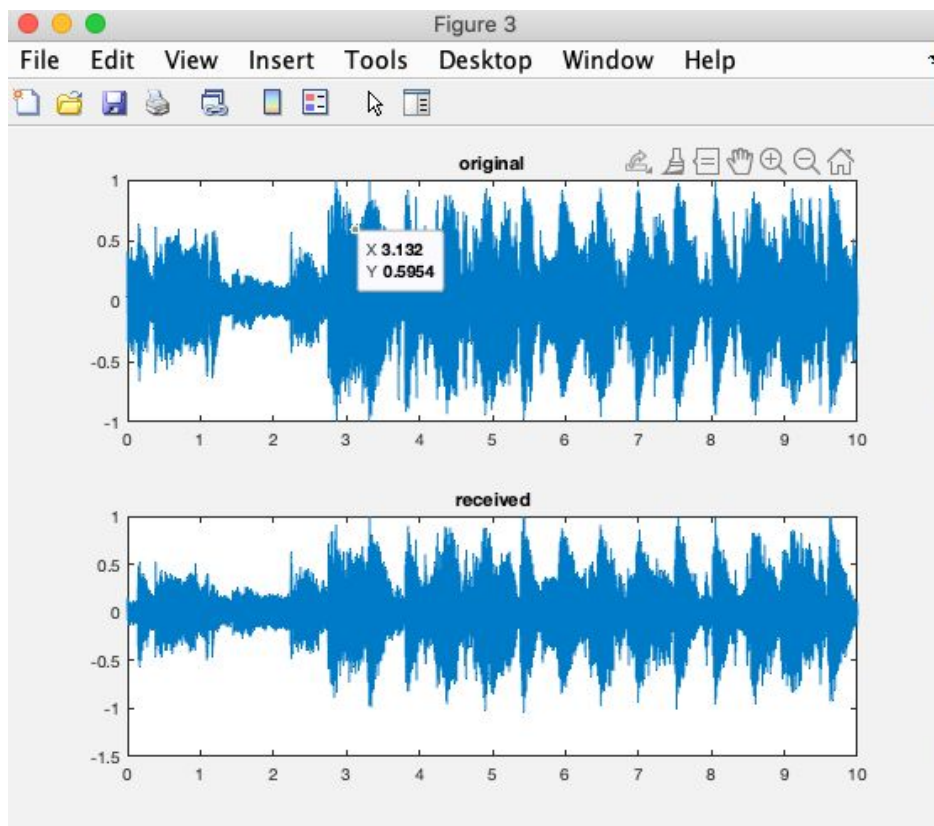
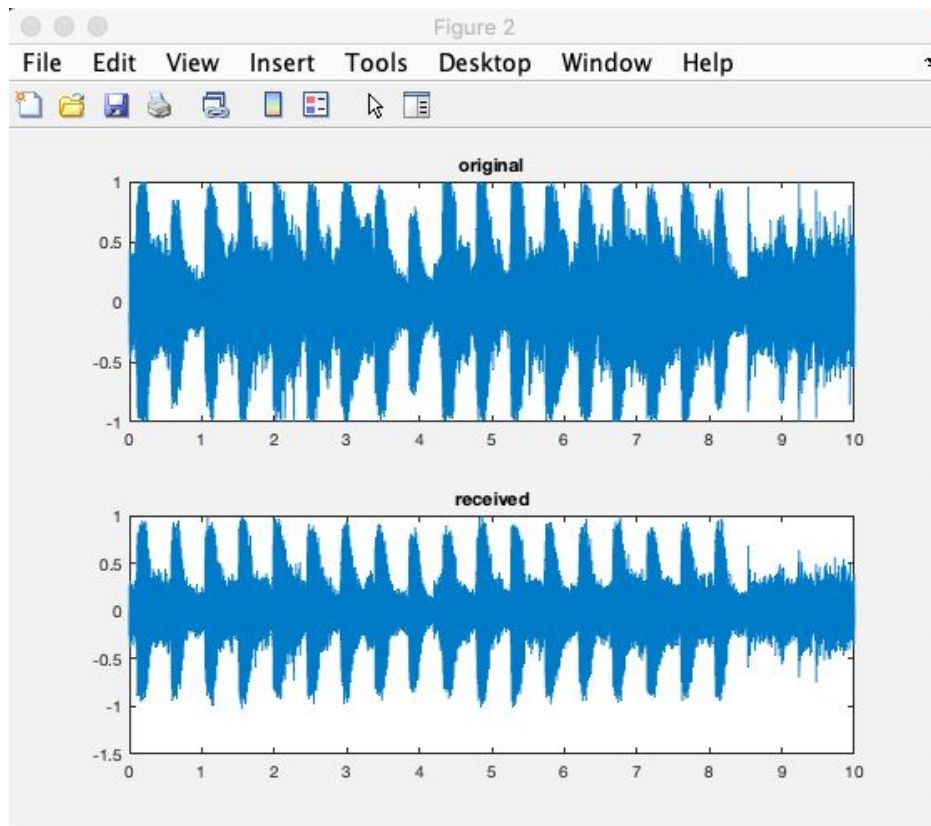
Figure1,2,3는 각각 song1,2,3에 대한 Original Signal 그래프, Received Signal 그래프를 보여준다.

Simulation Result

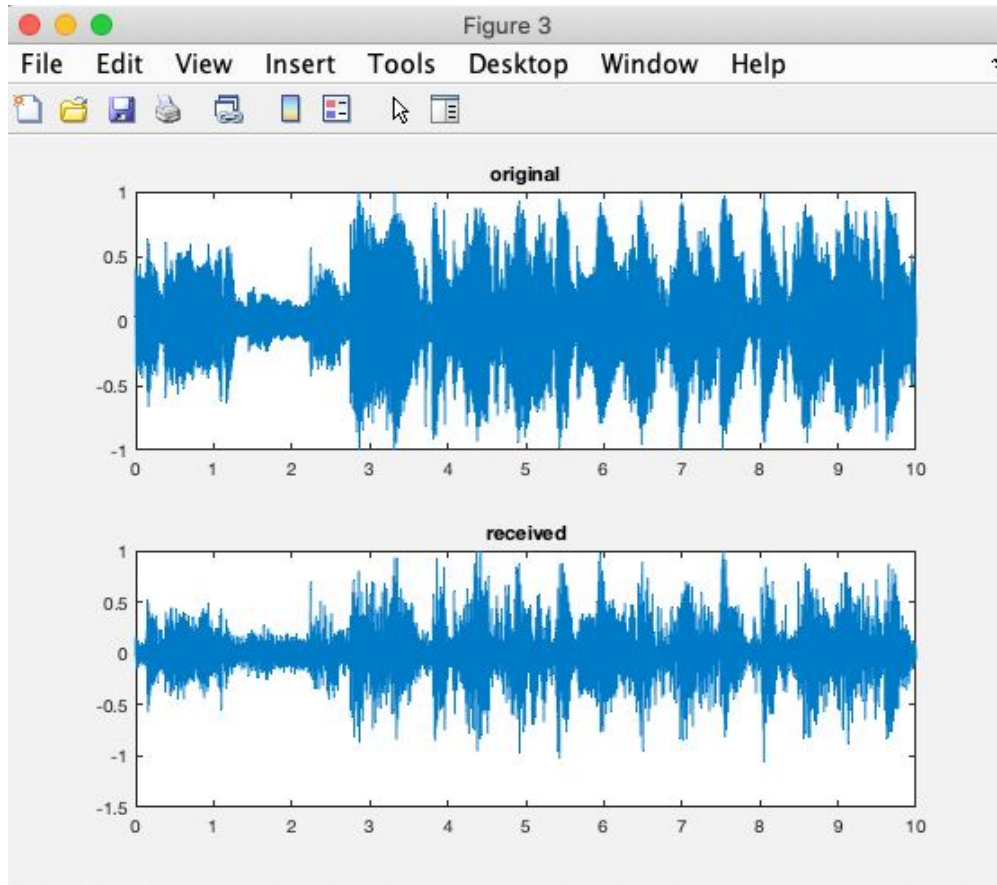
$w_1=100\text{kHz}$ $w_2=30\text{kHz}$ $w_3=65\text{kHz}$

$RC_1=2e-4$ $RC_2=2e-4$ $RC_3=2e-4$

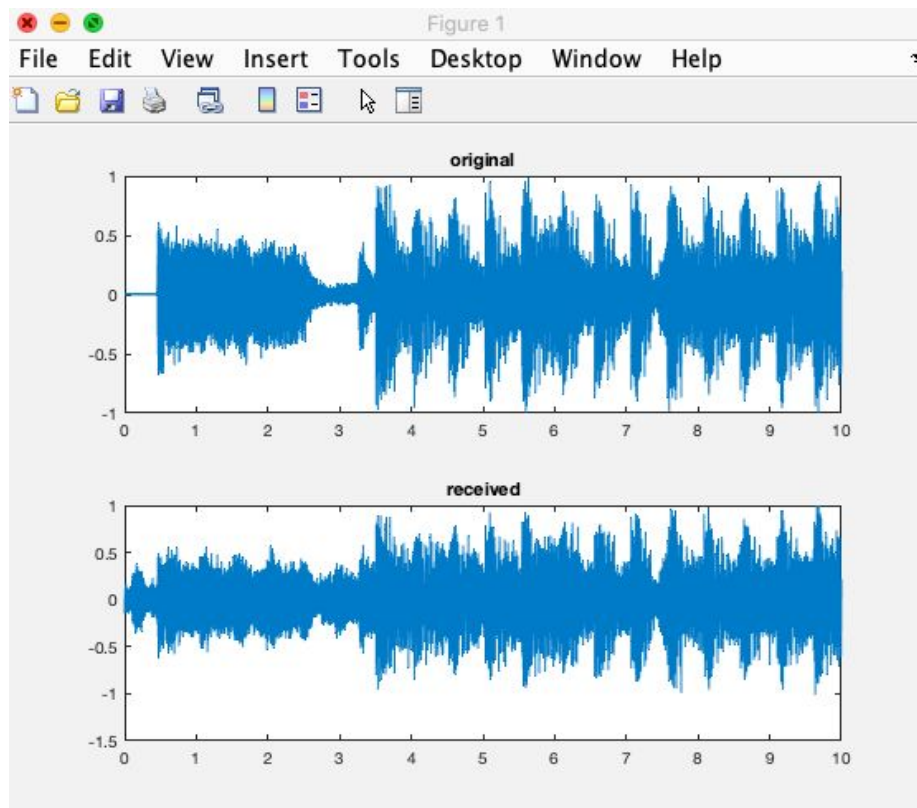




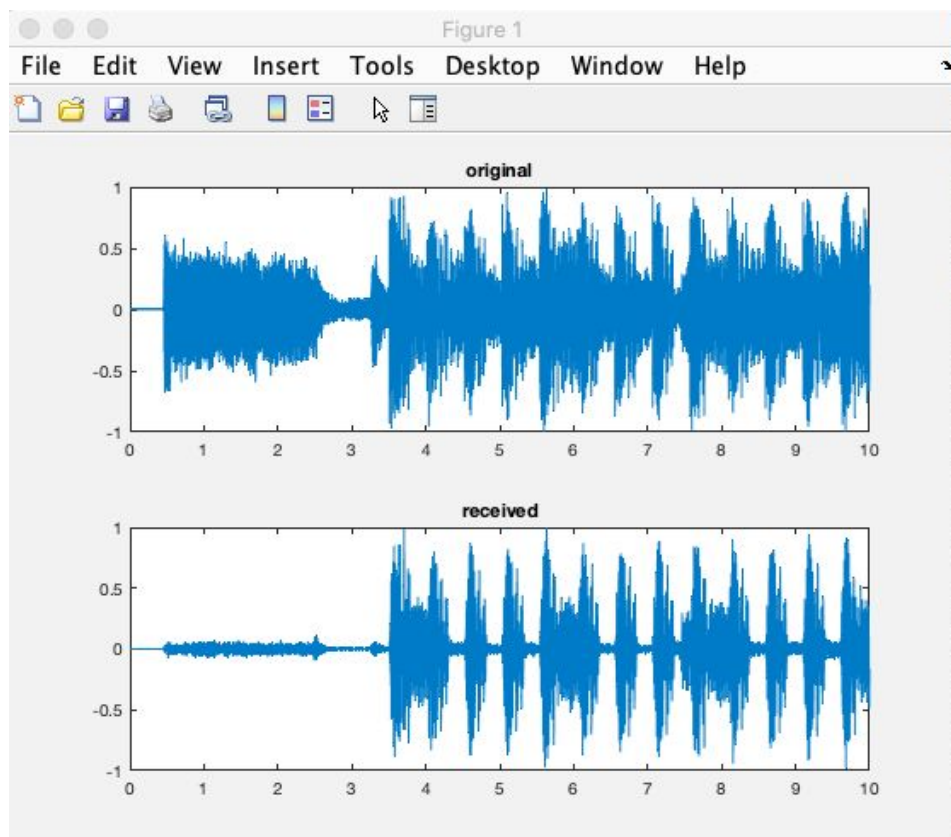
w3=e2로 수정했을 때
(Figure3만 바뀌므로 3만 첨부)



RC1=2e-5로 수정했을 때
(w3는 2e-4로 돌려놓음, Figure1만 바뀌므로 이것만 첨부)



RC1=2e-3으로 수정했을 때



Reflection

Carrier 주파수 W_1, W_2, W_3 은 서로 충분히 떨어져있지 않으면 서로 간섭이 생겨서 다른 음악소리가 들릴 수 있다. 음악파일은 보통 300~3400Hz의 주파수대역폭을 가지기 때문에 W_1, W_2, W_3 를 각각 100, 30, 65 kHz로 중심주파수 기준 35000 Hz의 간격을 두어 서로 충분히 떨어뜨려 놓았다.

Carrier frequency w 를 바꿔가며 시뮬레이션 해본 결과 $w_2=100(1e2)$ 로 상대적으로 낮게 했을 때 그래프에서는 신호가 약갈 열어보이는데 소리로 들어보면 낮은 음으로 목소리를 변조한것처럼 들리게 되는 것으로 봐서 Carrier 주파수가 충분히 높지 않으면 복조가 제대로 되지 않는것을 알 수 있었다.

LPF의 특성을 결정하는 RC값을 바꿔가며 시뮬레이션을 해본 결과 RC값이 너무 크면 3dB cutoff frequency가 작아져 소리가 조금 답답하게 들리고(그래프를 보면 원신호까지도 많이 깎인 모습), RC값이 작으면 3dB Frequency가 커져 noise가 생겨서(그래프를 보면 잡음도 LPF를 많이 통과해 추가되었음) 이 사이에 적절한 RC값을 선택해야함을 알 수 있었다.

이로써 여러개의 신호를 서로 겹쳐지지 않는 다른 주파수로 변조해 하나에 신호로 합쳐서 전송하고, 이를 복조하는 것을 실습해볼 수 있었고, 그래프로만 신호를 보는것이 아니라, 직접 복조되는 음악파일 소리를 들어보며 시뮬레이션을 해서 Multiplexing에 대해, 그리고 한학기동안 통신이론설계를 들으며 배운 많은 이론들에 대해서 더 실감나게 정리하며 공부할 수 있었다.