통신시스템 Matlab Project #5 (만점: 15점)

컴퓨터정보통신공학부 연세대학교 원주캠퍼스
담당교수: 김재권 (jaekwon@yonsei.ac.kr)

○ 목표: SSB Modulation/demodulation을 실험

○ 제공되는 Matlab Program: BPF_design2.m

○ 실습내용

1. 주어진 message신호 Matlab Project #4와 동일 (샘플링주파수: 8kHz)
2. USSB와 LSSB를 각각 구현
3. 각 단계에서의 신호를 time domain 및 frequency domain에서 관찰 (3장 슬라이드 54~57 참조)
4. Carrier 주파수는 적절히 선택. Coherent detection에서 phase shift는 없는 것으로 설정. 즉, $\phi = 0$.

○ 제출물 및 평가

- 제출일: 6월 24일 목요일 오후 3시까지 이메일 제출(jaekwon@yonsei.ac.kr)
- 제출물: 1. matlab 파일들을 프린트 후 손글씨로 자세히 설명
  2. 실험결과에 대한 해석
- 유의사항: Matlab Program을 제외한 모든 글은 손글씨로 작성

```matlab
%filename:AM_modulation.m
clear all;
close all;

load message2.dat; %fs=500, t=0:1/fs:511*1/fs
message=message2;
fs=400;%[samples/sec],[Hz]
N=1024;
message=[message zeros(1,N-length(message))];
t=0:1/fs:(N-1)*1/fs;
ka=0.7;
Ac=1.0;
fc=50;%[Hz]

m=message;
M=dft_new(m,N);
% time-domain message signal
figure, subplot(2,1,1), plot(t,m);
xlabel('time [s]');
ylabel('m(t)');
grid;
%axis([-5 5 -0.1 1.1]);
% frequency-domain message signal
subplot(2,1,2), plot([-fs/2:fs*1/N:fs*(N/2-1)/N],[abs(M(N/2+1:N)) abs(M(1:N/2))]);
xlabel('frequency [Hz]');
ylabel('|M(f)|');
grid;
%axis([-fs/2 fs/2 -10 100]);

% AM modulation
s=(1+ka*m).*cos(2*pi*fc*t);
S=dft_new(s,N);

% time-domain modulated signal
```

```
figure, subplot(2,1,1), plot(t,s);
xlabel('time [s]');
ylabel('s(t)=[1+k_a\times m(t)]cos(2\pi 100t)');
grid;
%axis([-5 5 -0.1 1.1]);
% frequency-domain modulated signal
subplot(2,1,2), plot([-fs/2:fs*1/N:fs*(N/2-1)/N],[abs(S(N/2+1:N)) abs(S(1:N/2))]);
xlabel('frequency [Hz]');
ylabel('|S(f)|');
grid;
%axis([-fs/2 fs/2 -10 100]);

% envelope detection
s_temp=abs(s);
S_temp=dft_new(s_temp,N);
% time-domain modulated signal
figure, subplot(2,1,1), plot(t,s_temp);
xlabel('time [s]');
ylabel('|s_{AM}(t)|^2');
grid;
%axis([-5 5 -0.1 1.1]);
% frequency-domain modulated signal
subplot(2,1,2), plot([-fs/2:fs*1/N:fs*(N/2-1)/N],[abs(S_temp(N/2+1:N)) abs(S_temp(1:N/2))]);
xlabel('frequency [Hz]');
ylabel('|S_{AM}(f)|');
grid;
%axis([-fs/2 fs/2 -10 100]);

%LPF
%LPF design
w_LPF=LPF_design;
W_LPF=dft_new(w_LPF,N);
% time-domain impulse response
figure, subplot(2,1,1), plot([0:length(w_LPF)-1]/fs,w_LPF);
xlabel('time [s]');
ylabel('w_{LPF}(t)');
```

```
grid;
%axis([-5 5 -0.1 1.1]);
% frequency-domain transfer function
subplot(2,1,2), plot([-fs/2:fs*1/N:fs*(N/2-1)/N],[abs(W_LPF(N/2+1:N)) abs(W_LPF(1:N/2))]);
xlabel('frequency [Hz]');
ylabel('|W_{LPF}(f)|');
grid;
%axis([-fs/2 fs/2 -10 100]);

s_LPF=conv_new(s_temp,w_LPF);
%removing the delay due to the LPF, the delay is 1024-800+1=225 samples
D=225;%samples
m_hat=s_LPF(D:D+N-1);
M_hat=dft_new(m_hat,N);
% time-domain recovered message signal
figure, subplot(2,1,1), plot(t,m_hat);
xlabel('time [s]');
ylabel('m_{hat}(t)');
grid;
%axis([-5 5 -0.1 1.1]);
% frequency-domain recovered message signal
subplot(2,1,2), plot([-fs/2:fs*1/N:fs*(N/2-1)/N],[abs(M_hat(N/2+1:N)) abs(M_hat(1:N/2))]);
xlabel('frequency [Hz]');
ylabel('|M_{hat}(f)|');
grid;
```

Matlab code 1. AM modulation

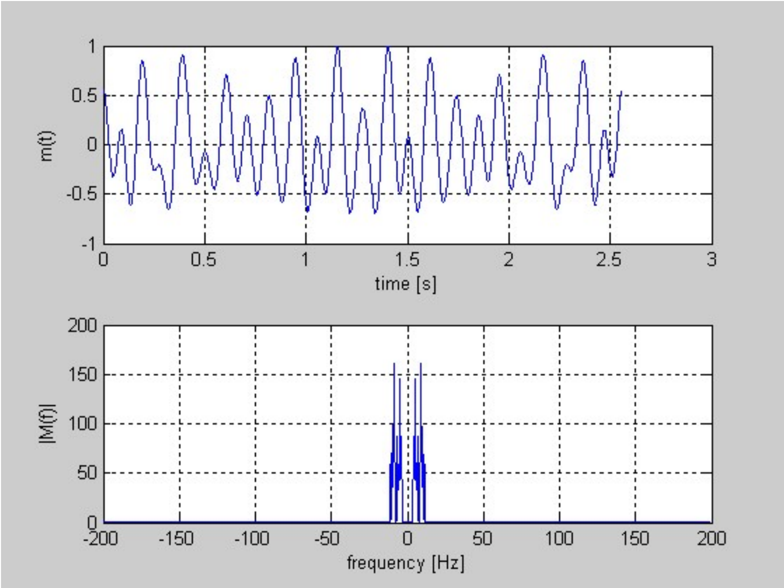

Fig. 1-1. message signal

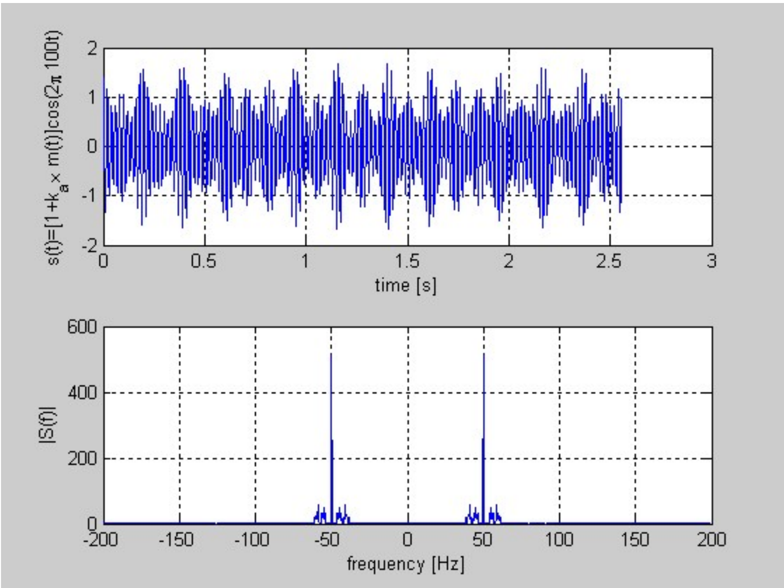

Fig. 1-2. AM modulated signal

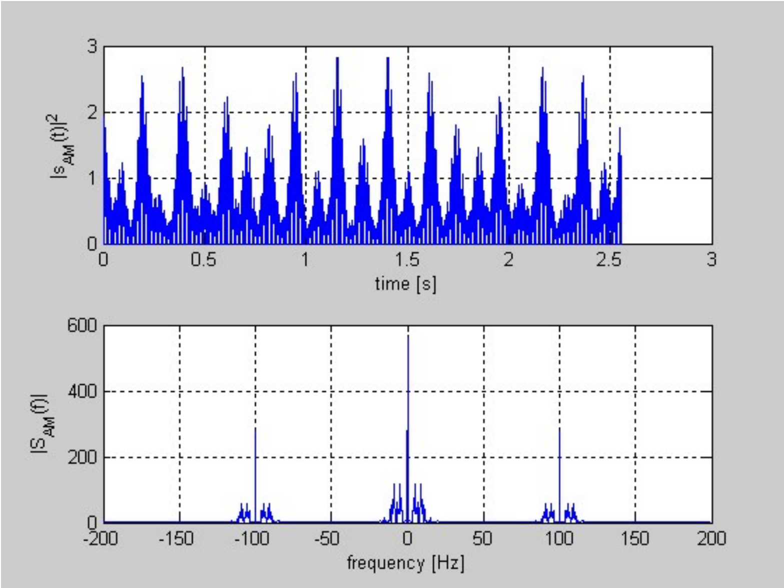

Fig. 1-3. squaring at the receiver

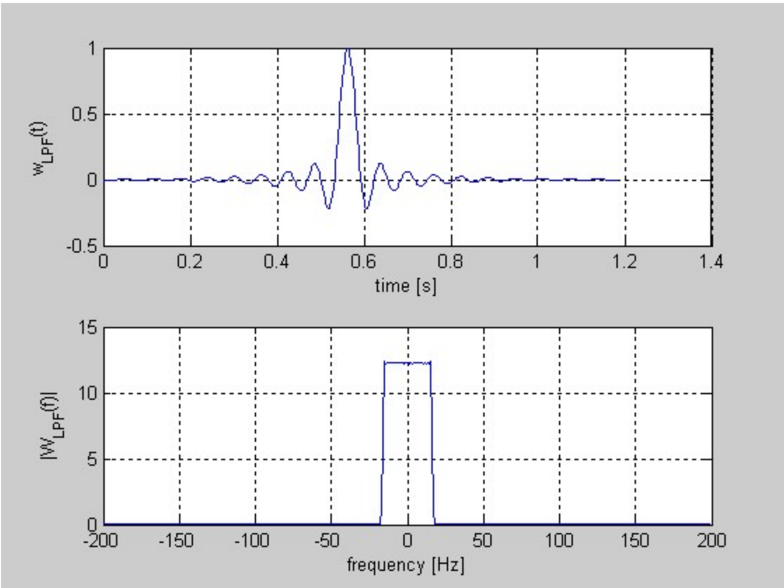

Fig. 1-4. LPF at the receiver

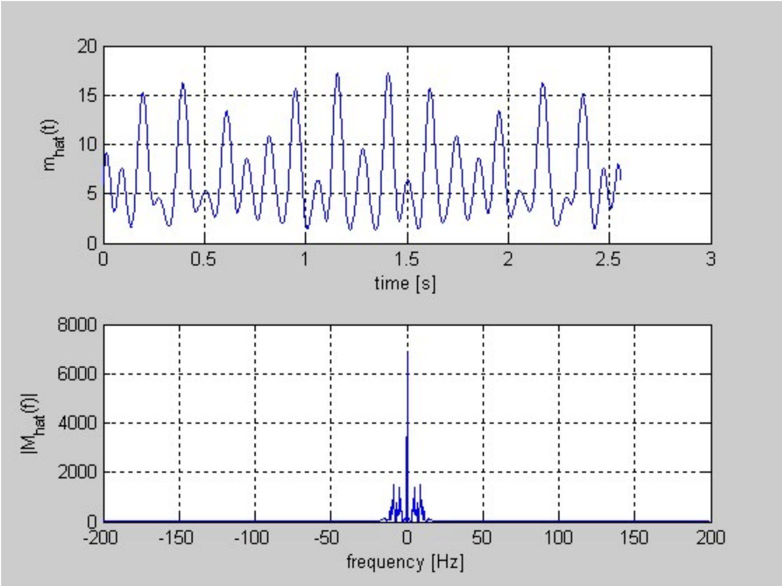

Fig. 1-5. recovered signal

```
%filename:DSB_SC.m
clear all;
close all;

load message2.dat; %fs=500, t=0:1/fs:511*1/fs,¦°¡Á¤ÇÔ
message=message2;
fs=400;%[samples/sec],[Hz]
N=1024;
message=[message zeros(1,N-length(message))];
t=0:1/fs:(N-1)*1/fs;
ka=0.7;
Ac=1.0;
fc=50;%[Hz]

m=message;
M=dft_new(m,N);

% time-domain message signal
figure, subplot(2,1,1), plot(t,m);
xlabel('time [s]');
ylabel('m(t)');
grid;
%axis([-5 5 -0.1 1.1]);
% frequency-domain message signal
subplot(2,1,2), plot([-fs/2:fs*1/N:fs*(N/2-1)/N],[abs(M(N/2+1:N)) abs(M(1:N/2))]);
xlabel('frequency [Hz]');
ylabel('|M(f)|');
grid;
%axis([-fs/2 fs/2 -10 100]);

s=m.*cos(2*pi*fc*t);
S=dft_new(s,N);

% time-domain modulated signal
figure, subplot(2,1,1), plot(t,s);
```

```
xlabel('time [s]');
ylabel('s_{DSB-SC}(t)=m(t)*cos(2\pi 100t)');
grid;
%axis([-5 5 -0.1 1.1]);
% frequency-domain modulated signal
subplot(2,1,2), plot([-fs/2:fs*1/N:fs*(N/2-1)/N],[abs(S(N/2+1:N)) abs(S(1:N/2))]);
xlabel('frequency [Hz]');
ylabel('|S_{DSB-SC}(f)|');
grid;
%axis([-fs/2 fs/2 -10 100]);

% coherent detection
s_temp=s.*cos(2*pi*fc*t);
S_temp=dft_new(s_temp,N);
% time-domain modulated signal
figure, subplot(2,1,1), plot(t,s_temp);
xlabel('time [s]');
ylabel('s_{DSB-SC}(t)*cos(2\pi 50t)');
grid;
%axis([-5 5 -0.1 1.1]);
% frequency-domain modulated signal
subplot(2,1,2), plot([-fs/2:fs*1/N:fs*(N/2-1)/N],[abs(S_temp(N/2+1:N)) abs(S_temp(1:N/2))]);
xlabel('frequency [Hz]');
ylabel('|S_{temp}(f)|');
grid;
%axis([-fs/2 fs/2 -10 100]);

%LPF
%LPF design
w_LPF=LPF_design;
W_LPF=dft_new(w_LPF,N);
% time-domain impulse response
figure, subplot(2,1,1), plot([0:length(w_LPF)-1]/fs,w_LPF);
xlabel('time [s]');
ylabel('w_{LPF}(t)');
grid;
```

```
%axis([-5 5 -0.1 1.1]);
% frequency-domain transfer function
subplot(2,1,2), plot([-fs/2:fs*1/N:fs*(N/2-1)/N],[abs(W_LPF(N/2+1:N)) abs(W_LPF(1:N/2))]);
xlabel('frequency [Hz]');
ylabel('|W_{LPF}(f)|');
grid;
%axis([-fs/2 fs/2 -10 100]);

s_LPF=conv_new(s_temp,w_LPF);
%removing the delay due to the LPF, the delay is 1024-800+1=225 samples
D=225;%samples
m_hat=s_LPF(D:D+N-1);
M_hat=dft_new(m_hat,N);
% time-domain recovered message signal
figure, subplot(2,1,1), plot(t,m_hat);
xlabel('time [s]');
ylabel('m_{hat}(t)');
grid;
%axis([-5 5 -0.1 1.1]);
% frequency-domain recovered message signal
subplot(2,1,2), plot([-fs/2:fs*1/N:fs*(N/2-1)/N],[abs(M_hat(N/2+1:N)) abs(M_hat(1:N/2))]);
xlabel('frequency [Hz]');
ylabel('|M_{hat}(f)|');
grid;
```

Matlab code 2. DSB-SC

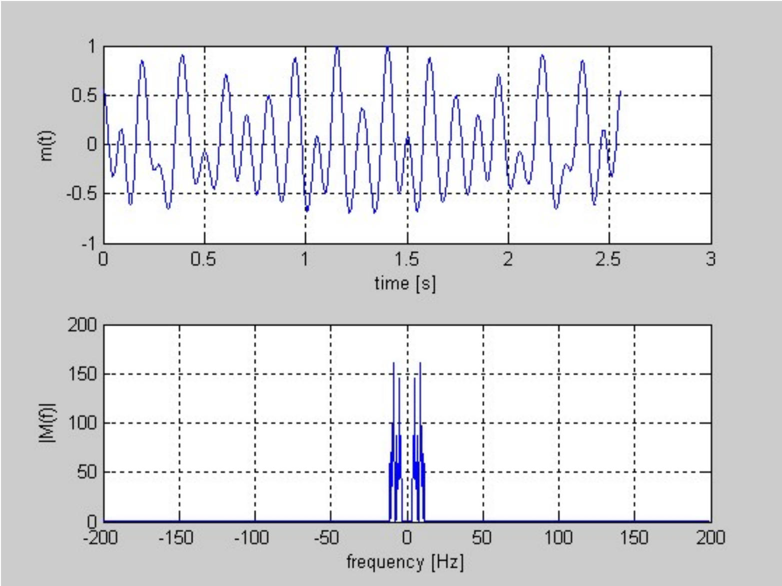

Fig. 2-1. message signal

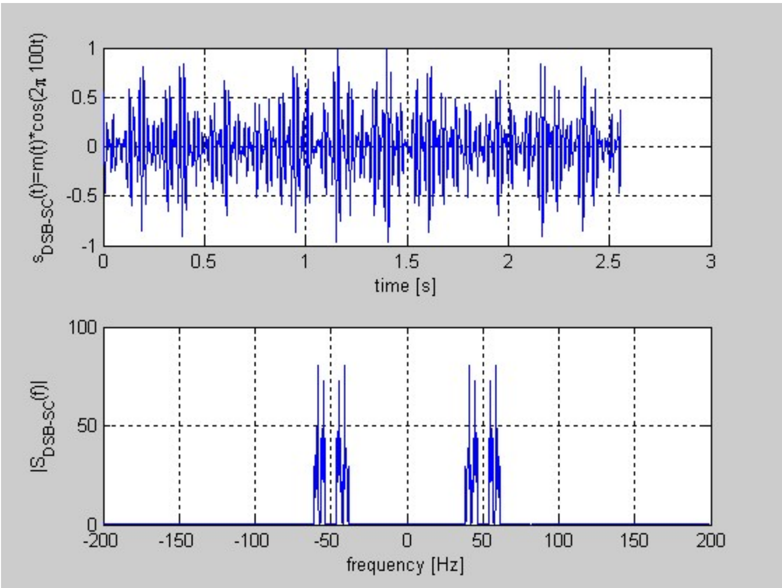

Fig. 2-2. DSB-SC modulated signal

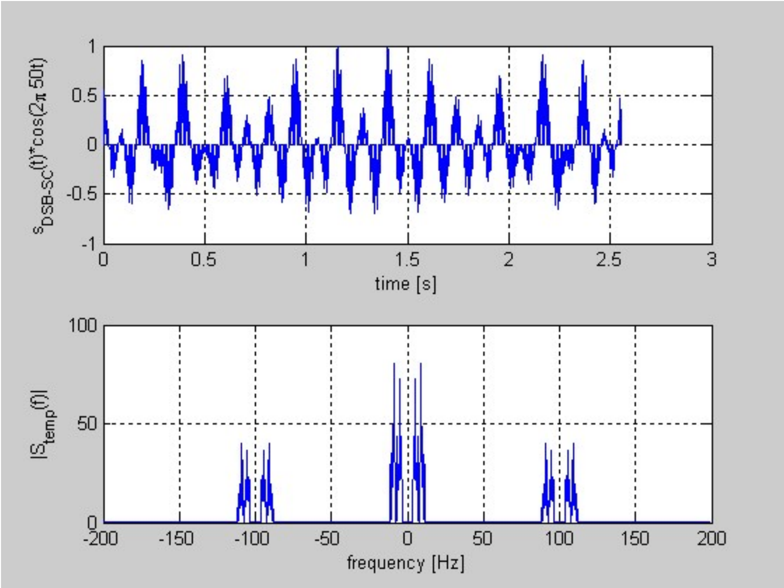

Fig. 2-3. frequency translation at the coherent detector

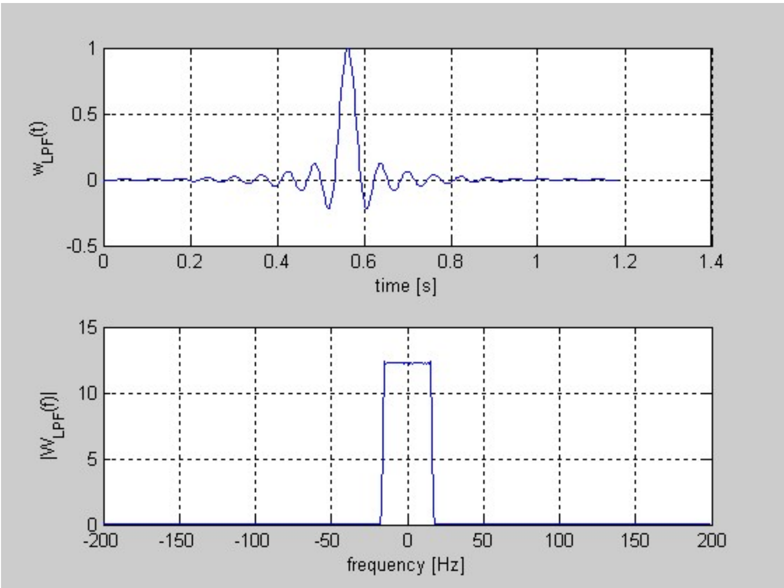

Fig. 2-4. LPF at the coherent detector

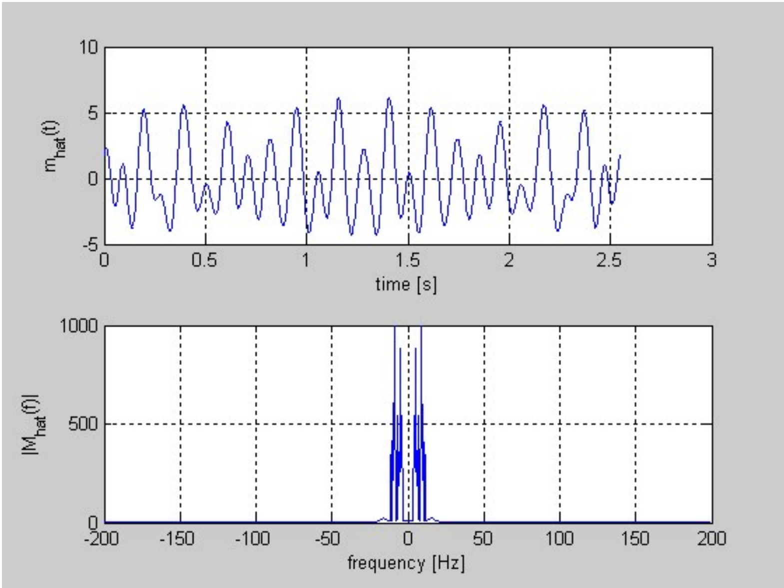

Fig. 2-5. recovered message

```
%filename:USSB.m
clear all;
close all;

load message2.dat; %fs=500, t=0:1/fs:511*1/fs,¦ °¡Á¤ÇÔ
message=message2;
fs=400;%[samples/sec],[Hz]
N=1024;
message=[message zeros(1,N-length(message))];
t=0:1/fs:(N-1)*1/fs;
ka=0.7;
Ac=1.0;
fc=50;%[Hz]

m=message;
M=dft_new(m,N);
% time-domain message signal
figure, subplot(2,1,1), plot(t,m);
xlabel('time [s]');
ylabel('m(t)');
grid;
%axis([-5 5 -0.1 1.1]);
% frequency-domain message signal
subplot(2,1,2), plot([-fs/2:fs*1/N:fs*(N/2-1)/N],[abs(M(N/2+1:N)) abs(M(1:N/2))]);
xlabel('frequency [Hz]');
ylabel('|M(f)|');
grid;
%axis([-fs/2 fs/2 -10 100]);

%USSB modulation
s_temp=m.*cos(2*pi*fc*t);
S_temp=dft_new(s_temp,N);
% time-domain modulated signal
figure, subplot(2,1,1), plot(t,s_temp);
xlabel('time [s]');
```

```
ylabel('s_{temp}(t)=m(t)*cos(2\pi 100t)');
grid;
%axis([-5 5 -0.1 1.1]);
% frequency-domain modulated signal
subplot(2,1,2), plot([-fs/2:fs*1/N:fs*(N/2-1)/N],[abs(S_temp(N/2+1:N)) abs(S_temp(1:N/2))]);
xlabel('frequency [Hz]');
ylabel('|S_{temp}(f)|');
grid;
%axis([-fs/2 fs/2 -10 100]);

% Tx pulse shape filter design
w_BPF=BPF_design;
W_BPF=dft_new(w_BPF,N);
% time-domain impulse response
figure, subplot(2,1,1), plot([0:length(w_BPF)-1]/fs,w_BPF);
xlabel('time [s]');
ylabel('w_{BPF}(t)');
grid;
%axis([-5 5 -0.1 1.1]);

% frequency-domain transfer function
subplot(2,1,2), plot([-fs/2:fs*1/N:fs*(N/2-1)/N],[abs(W_BPF(N/2+1:N)) abs(W_BPF(1:N/2))]);
xlabel('frequency [Hz]');
ylabel('|W_{BPF}(f)|');
grid;
%axis([-fs/2 fs/2 -10 100]);
s_temp=conv_new(s_temp,w_BPF);
%removing the delay due to the LPF, the delay is 1024-800+1=225 samples
D=225;%samples
s=s_temp(D:D+N-1);
S=dft_new(s,N);

% time-domain modulated signal
figure, subplot(2,1,1), plot(t,s);
xlabel('time [s]');
ylabel('s_{USSB}(t)');
```

```
grid;
%axis([-5 5 -0.1 1.1]);
% frequency-domain modulated signal
subplot(2,1,2), plot([-fs/2:fs*1/N:fs*(N/2-1)/N],[abs(S(N/2+1:N)) abs(S(1:N/2))]);
xlabel('frequency [Hz]');
ylabel('|S_{USSB}(f)|');
grid;
%axis([-fs/2 fs/2 -10 100]);

% coherent detection
s_temp=s.*cos(2*pi*fc*t);
S_temp=dft_new(s_temp,N);
% time-domain modulated signal
figure, subplot(2,1,1), plot(t,s_temp);
xlabel('time [s]');
ylabel('s_{USSB}(t)*cos(2\pi 50t)');
grid;
%axis([-5 5 -0.1 1.1]);
% frequency-domain modulated signal
subplot(2,1,2), plot([-fs/2:fs*1/N:fs*(N/2-1)/N],[abs(S_temp(N/2+1:N)) abs(S_temp(1:N/2))]);
xlabel('frequency [Hz]');
ylabel('|S_{temp}(f)|');
grid;
%axis([-fs/2 fs/2 -10 100]);

%LPF
% LPF design
w_LPF=LPF_design;
W_LPF=dft_new(w_LPF,N);
% time-domain impulse response
figure, subplot(2,1,1), plot([0:length(w_LPF)-1]/fs,w_LPF);
xlabel('time [s]');
ylabel('w_{LPF}(t)');
grid;
%axis([-5 5 -0.1 1.1]);
% frequency-domain transfer function
```

```matlab
subplot(2,1,2), plot([-fs/2:fs*1/N:fs*(N/2-1)/N],[abs(W_LPF(N/2+1:N)) abs(W_LPF(1:N/2))]);
xlabel('frequency [Hz]');
ylabel('|W_{LPF}(f)|');
grid;
%axis([-fs/2 fs/2 -10 100]);

s_LPF=conv_new(s_temp,w_LPF);
%removing the delay due to the LPF, the delay is 1024-800+1=225 samples
D=225;%samples
m_hat=s_LPF(D:D+N-1);
M_hat=dft_new(m_hat,N);
% time-domain recovered message signal
figure, subplot(2,1,1), plot(t,m_hat);
xlabel('time [s]');
ylabel('m_{hat}(t)');
grid;
%axis([-5 5 -0.1 1.1]);
% frequency-domain recovered message signal
subplot(2,1,2), plot([-fs/2:fs*1/N:fs*(N/2-1)/N],[abs(M_hat(N/2+1:N)) abs(M_hat(1:N/2))]);
xlabel('frequency [Hz]');
ylabel('|M_{hat}(f)|');
grid;
```
Matlab code 3. USSB

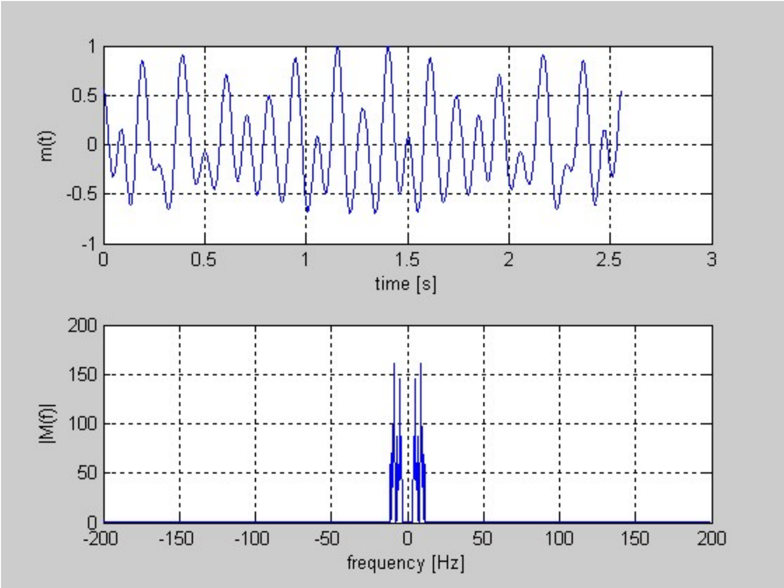

Fig. 3-1. message signal

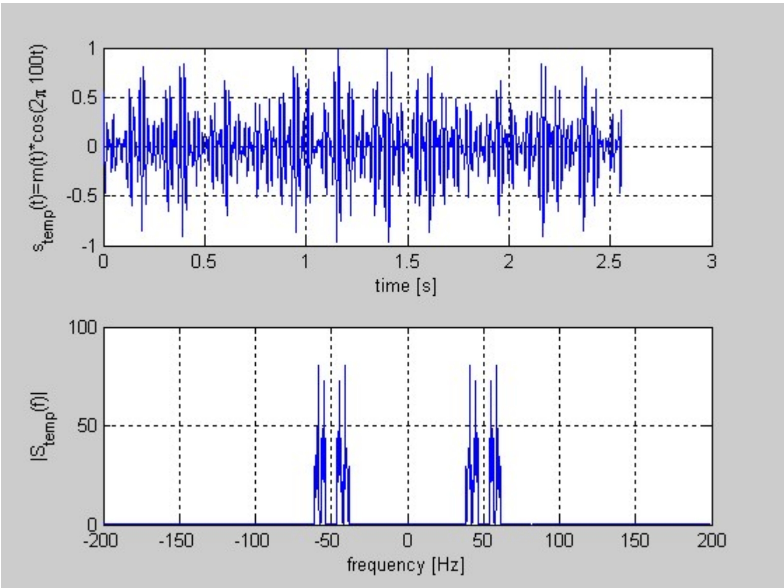

Fig. 3-2. product modulation at the Tx

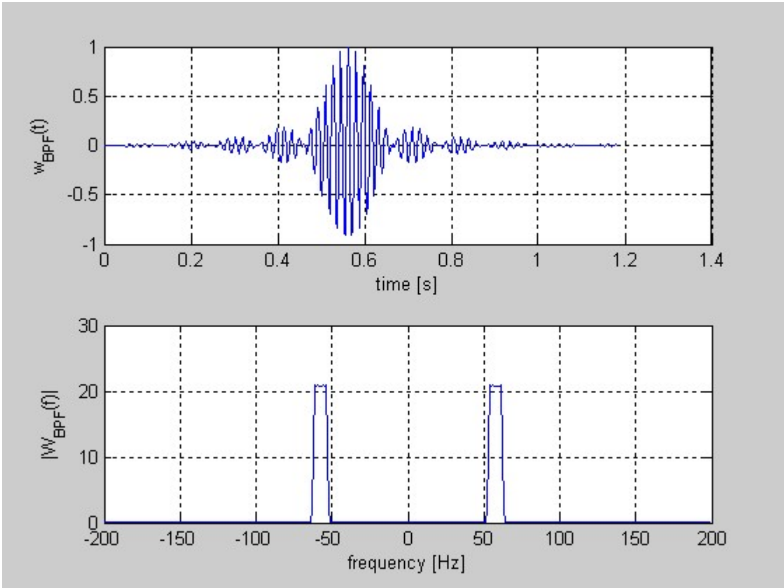

Fig. 3-3. BPF/ pulse shaping filter at the Tx

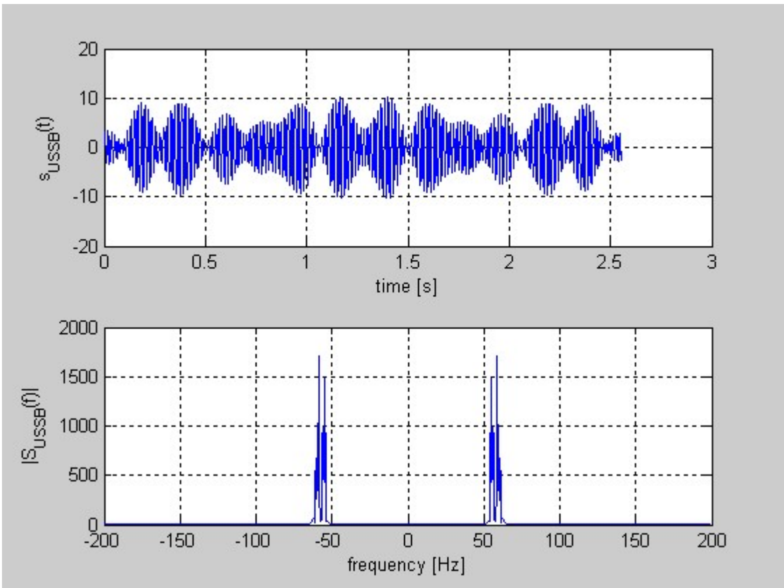

Fig. 3-4. USSB modulated signal

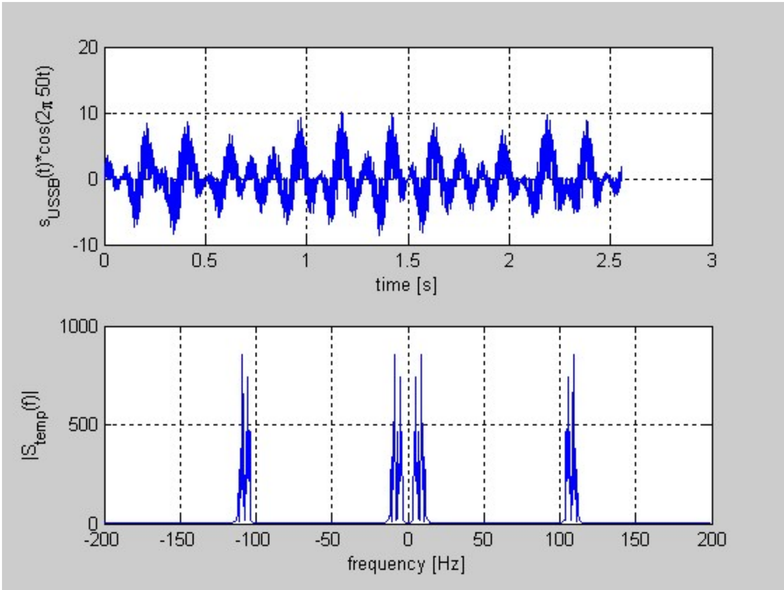

Fig. 3-5 product modulation/ frequency translation at Rx

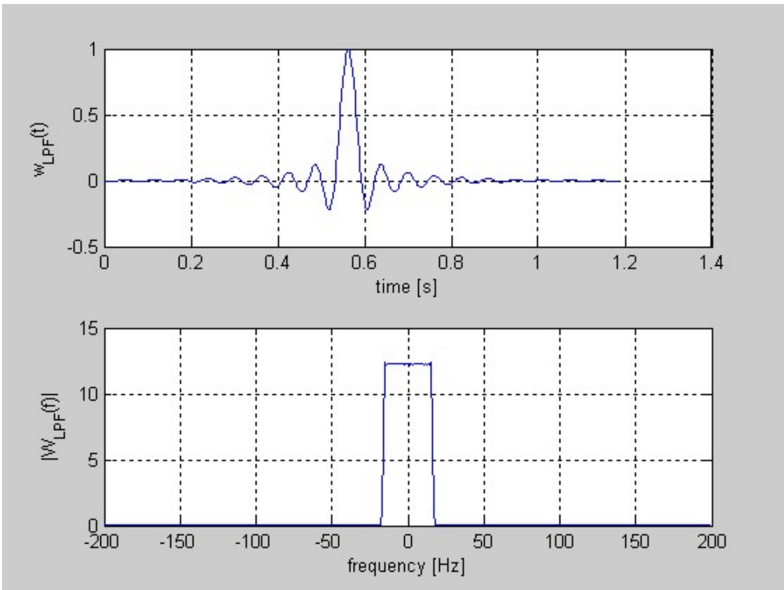

Fig. 3-6. LPF at Rx (coherent detector)

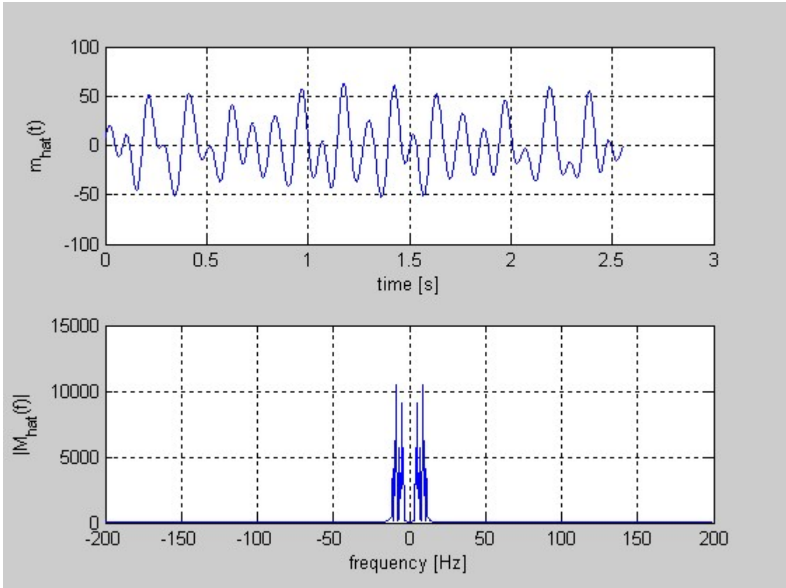

Fig. 3-7. recovered message

```
%filename:LPF_design.m
function w=LPF_design();
W=[ones(1,40) 0.9 0.7 0.4 0.2 0.1 0.01 0.001 0 0 0];
W2=[W zeros(1,1024-2*length(W)+1) wrev(W(2:length(W)))];
w=idft_new(W2,1024);
w=real(w);
w=w/max(abs(w));
w=[w(800:1024) w(1:250)];
```

Matlab code 4. LPF design

```
%filename:BPF_design.m
function w_BPF=BPF_design();
W=[zeros(1,128) 0 0 0 0.001 0.01 0.1 0.2 0.4 0.7 0.9 ones(1,20) 0.9 0.7 0.4 0.2 0.1 0.01 0.001 0 0 0];
W2=[W zeros(1,1024-2*length(W)+1) wrev(W(2:length(W)))];
w=idft_new(W2,1024);
w=real(w);
w=w/max(abs(w));
w_BPF=[w(800:1024) w(1:250)];
```

Matlab code 5. BPF design