

---

## Table of Contents

Pulse Formation .....	1
Signal Formation and Up-Conversion .....	4
Up-Convert .....	5
Downconversion .....	9
Matched Filter .....	11

## Pulse Formation

### Pulse Shape 1 - Sinc Function

```
N = 30;
Ts = .1; % symbol period (rate 1/Ts)
dt = .01; % sample period
t = -5*Ts:dt:5*Ts; % time vector
sinc_func = sinc(t/Ts); % define sinc
sinc_func2 = cos(2*pi*10*t).*sinc(t/Ts);
sinc_func3 = cos(2*pi*20*t).*sinc(t/Ts);
sinc_func4 = cos(2*pi*30*t).*sinc(t/Ts);
c = 1;

figure
subplot(2,1,1), plot(t,sinc_func)
xlabel('time (s)'), ylabel('p_1(t)'), title('Truncated sinc')

fs = 1/dt; % sample frequency
Nfft = 1024; % length of fft
f = [0:fs/Nfft:fs-fs/Nfft];
subplot(2,1,2)
hold on
plot(f,abs(fft(sinc_func,Nfft)))
plot(f,abs(fft(sinc_func2,Nfft)))
plot(f,abs(fft(sinc_func3,Nfft)))
plot(f,abs(fft(sinc_func4,Nfft)))
xlabel('frequency (Hz)'), ylabel('|P_1(j\omega)|')

% Pulse Shape 2:
square = zeros(10,1);
square(4:7) = ones(4,1);
square2 = cos(2*pi*10*t).*square;
square3 = cos(2*pi*20*t).*square;
square4 = cos(2*pi*30*t).*square;

figure
subplot(2,1,1), plot(square)
xlabel('time (s)'), ylabel('p_2(t)'), title('Square wave')
```

---

```

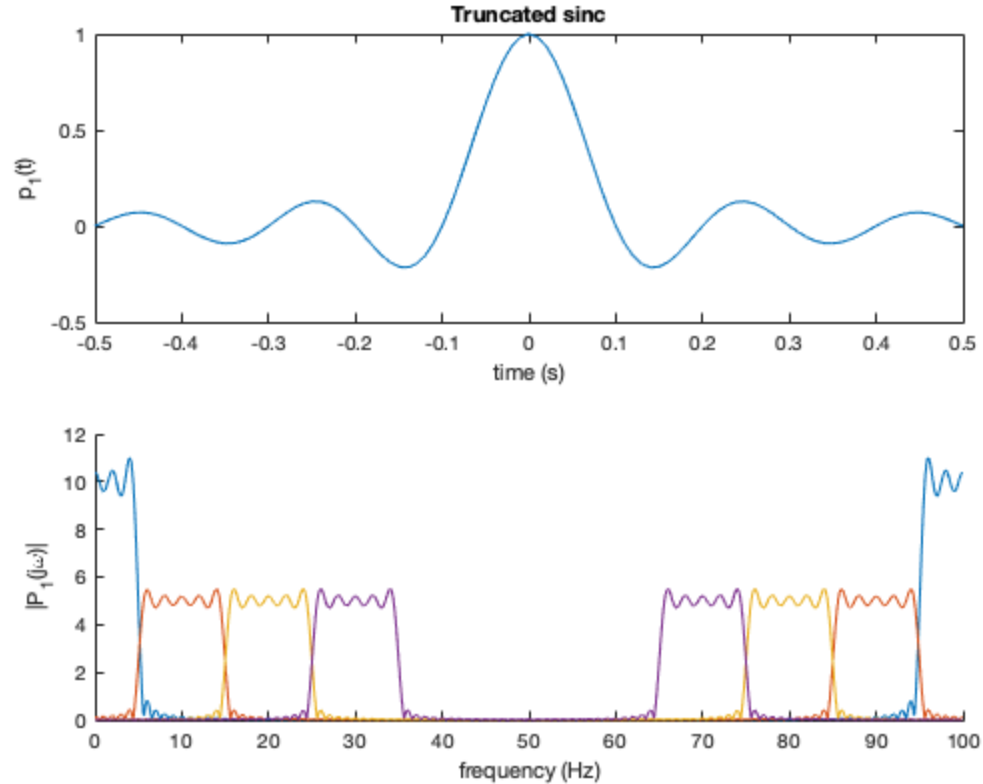
subplot(2,1,2)
hold on
plot(f,abs(fft(square,Nfft)))
plot(f,abs(fft(square2,Nfft)))
plot(f,abs(fft(square3,Nfft)))
plot(f,abs(fft(square4,Nfft)))
xlabel('frequency (Hz)'), ylabel('|P_2(j\omega)|')

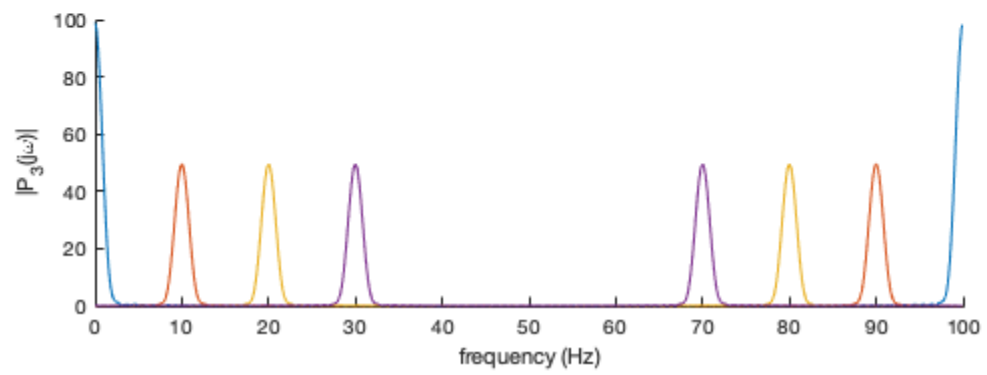
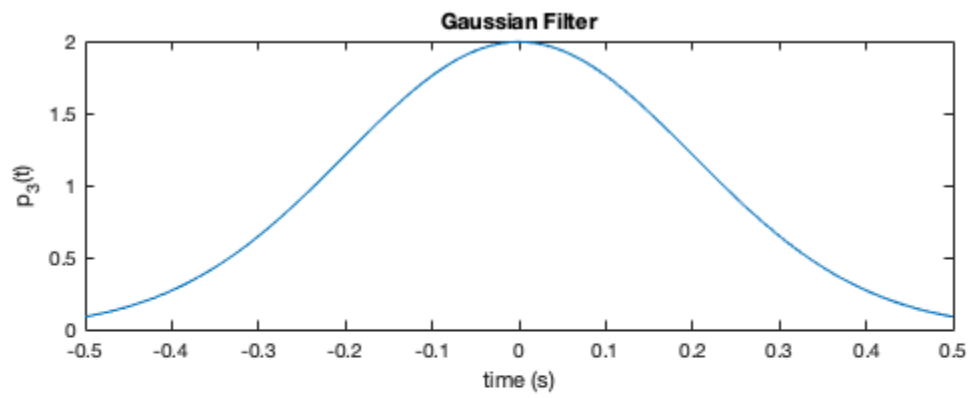
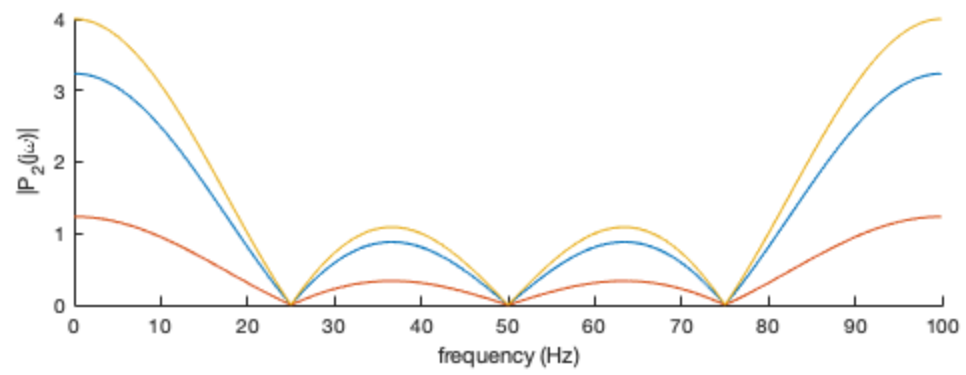
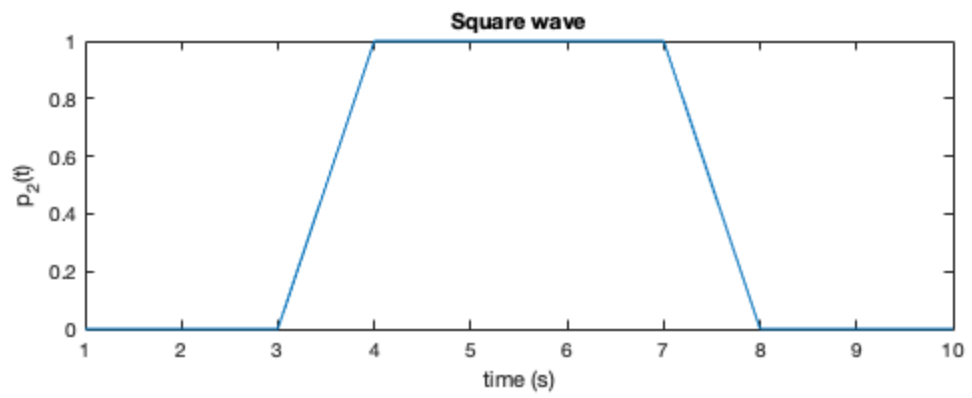
% Pulse Shape 3:
gauss = normpdf(t,0,.2);
gauss2 = cos(2*pi*10*t).*gauss;
gauss3 = cos(2*pi*20*t).*gauss;
gauss4 = cos(2*pi*30*t).*gauss;

figure
subplot(2,1,1), plot(t,gauss)
xlabel('time (s)'), ylabel('p_3(t)'), title('Gaussian Filter')

subplot(2,1,2)
hold on
plot(f,abs(fft(gauss,Nfft)))
plot(f,abs(fft(gauss2,Nfft)))
plot(f,abs(fft(gauss3,Nfft)))
plot(f,abs(fft(gauss4,Nfft)))
xlabel('frequency (Hz)'), ylabel('|P_3(j\omega)|')

```





---

# Signal Formation and Up-Conversion

```
p = sinc_func; % Change this to change symbol shape

% Message 1
message = 'I love ESE 351! :)';
binary = str2num(reshape(dec2bin(message)',1,[]));
%binary = 2*((rand(1,N)>0.5)-0.5);
vector1 = zeros(length(binary)*c/Ts,1);

for i = 1:length(binary)
    if binary(i)==0
        binary(i)=-1;
    end
end

for i = 1:length(vector1)
    if mod(i,c/Ts)==0
        vector1(i) = binary(i*Ts/c);
    end
end

signal = conv(vector1,p,'same');

% Message 2
message2 = 'Trobaugh = goat !';
binary2 = str2num(reshape(dec2bin(message2)',1,[]));
%binary2 = 2*((rand(1,N)>0.5)-0.5);
vector2 = zeros(length(binary2)*c/Ts,1);

for i = 1:length(binary2)
    if binary2(i)==0
        binary2(i)=-1;
    end
end

for i = 1:length(vector2)
    if mod(i,c/Ts)==0
        vector2(i) = binary2(i*Ts/c);
    end
end

signal2 = conv(vector2,p,'same');

% Message 3
message3 = 'Systems > Electrical!';
binary3 = str2num(reshape(dec2bin(message3)',1,[]));
%binary3 = 2*((rand(1,N)>0.5)-0.5);

vector3 = zeros(length(binary3)*c/Ts,1);

for i = 1:length(binary3)
    if binary3(i)==0
```

---

```

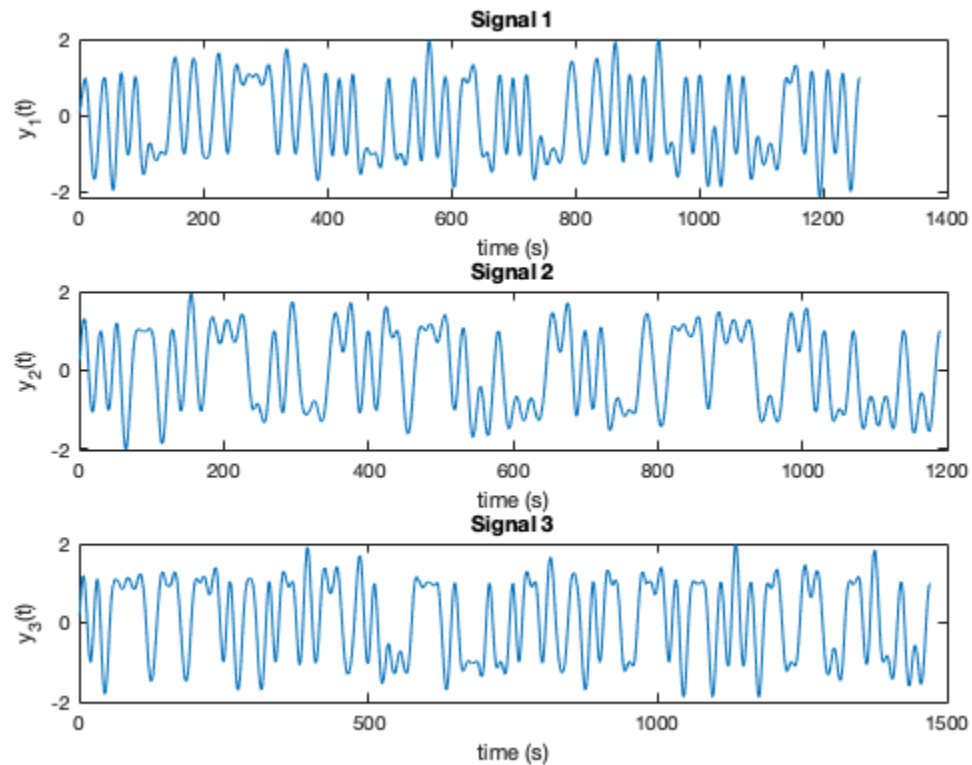
        binary3(i)=-1;
    end
end

for i = 1:length(vector3)
    if mod(i,c/Ts)==0
        vector3(i) = binary3(i*Ts/c);
    end
end

signal3 = conv(vector3,p,'same');

figure
subplot(3,1,1)
plot(signal)
xlabel('time (s)'), ylabel('y_1(t)'), title('Signal 1')
subplot(3,1,2)
plot(signal2)
xlabel('time (s)'), ylabel('y_2(t)'), title('Signal 2')
subplot(3,1,3)
plot(signal3)
xlabel('time (s)'), ylabel('y_3(t)'), title('Signal 3')

```



## Up-Convert

```

t = -length(signal)*dt/2:dt:length(signal)*dt/2-dt; % time vector

```

---

```

t2 = -length(signal2)*dt/2:dt:length(signal2)*dt/2-dt;
t3 = -length(signal3)*dt/2:dt:length(signal3)*dt/2-dt;

freq = [20 30 40];
wc = [2*pi*freq(1) 2*pi*freq(2) 2*pi*freq(3)];
y = signal.*cos(wc(1)*t');
y2 = signal2.*cos(wc(2)*t2');
y3 = signal3.*cos(wc(3)*t3');

figure
subplot(3,1,1), plot(t,y)
xlabel('time (s)'), ylabel('y(t)*cos(w_ct)'), title('Up-converted
    signal Y_1*cos(w_ct)')

subplot(3,1,2), plot(t2,y2)
xlabel('time (s)'), ylabel('y(t)*cos(w_ct)'), title('Up-converted
    signal Y_2*cos(w_ct)')

subplot(3,1,3), plot(t3,y3)
xlabel('time (s)'), ylabel('y(t)*cos(w_ct)'), title('Up-converted
    signal Y_3*cos(w_ct)')

combined = zeros(max([length(y) length(y2) length(y3)]),1);
for i = 1:length(y)
    combined(i) = y(i);
end
for i = 1:length(y2)
    combined(i) = combined(i)+y2(i);
end
for i = 1:length(y)
    combined(i) = combined(i)+y3(i);
end

% Adding Noise
sigma = 3/9;
noise = sigma*randn(1,length(combined));
combined = combined+noise';

tmax = -length(combined)*dt/2:dt:length(combined)*dt/2-dt;

figure
subplot(4,1,1), plot(f,abs(fft(y,Nfft)))
xlabel('frequency (Hz)'), ylabel('|Y_1(j\omega)|'), title('FT of up-
converted signal: fft(Y_1*cos(w_ct))');
subplot(4,1,2), plot(f,abs(fft(y2,Nfft)))
xlabel('frequency (Hz)'), ylabel('|Y_2(j\omega)|'), title('FT of up-
converted signal: fft(Y_2*cos(w_ct))');

subplot(4,1,3), plot(f,abs(fft(y3,Nfft)))
xlabel('frequency (Hz)'), ylabel('|Y_3(j\omega)|'), title('FT of up-
converted signal: fft(Y_3*cos(w_ct))');

```

---

---

```

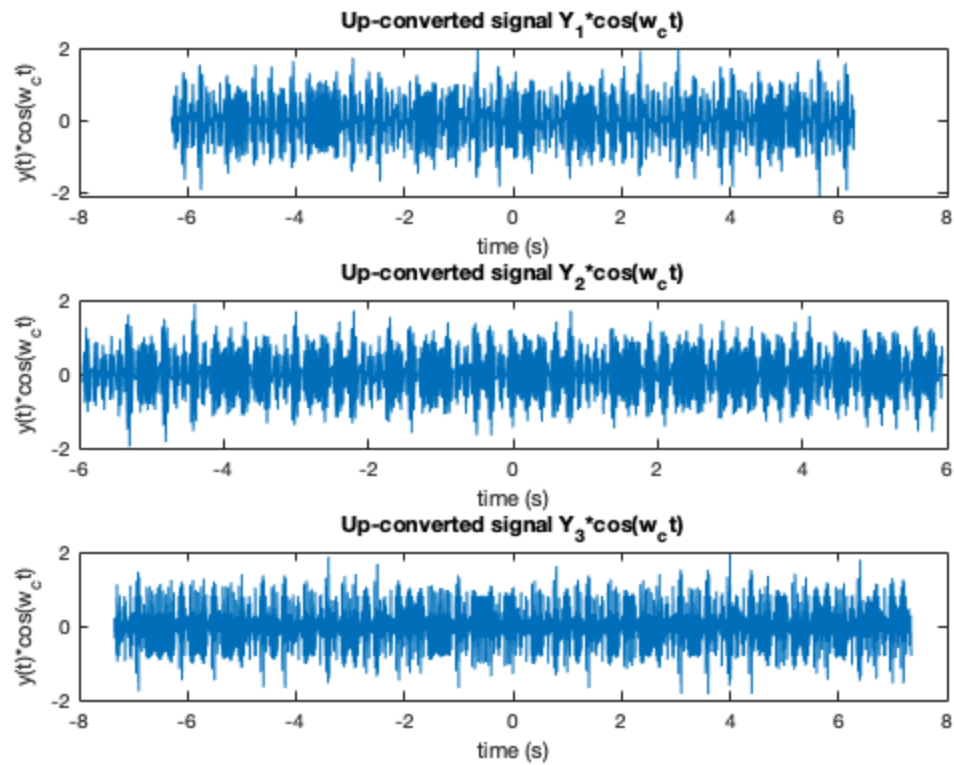
subplot(4,1,4), plot(f,abs(fft(combined,Nfft)))
xlabel('frequency (Hz)'), ylabel('|R(j\omega)|'), title('Combined (Y +
noise)')

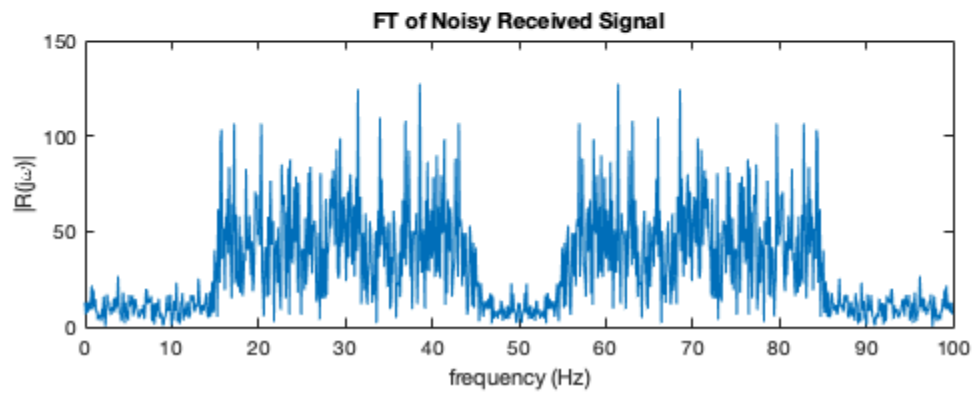
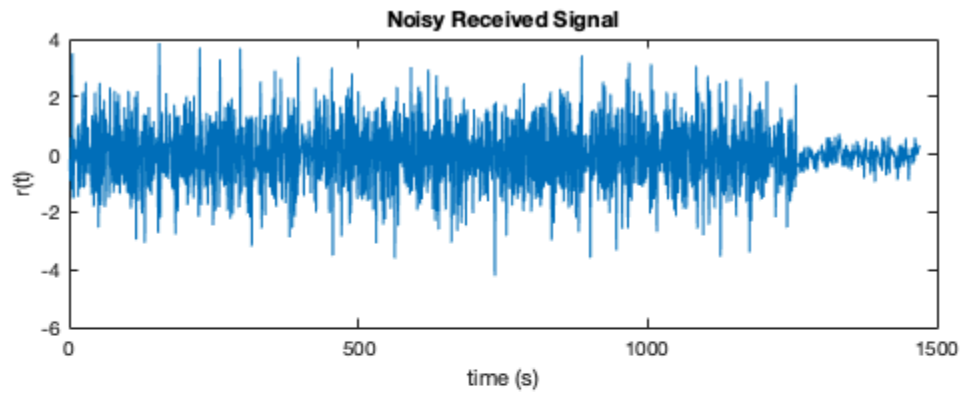
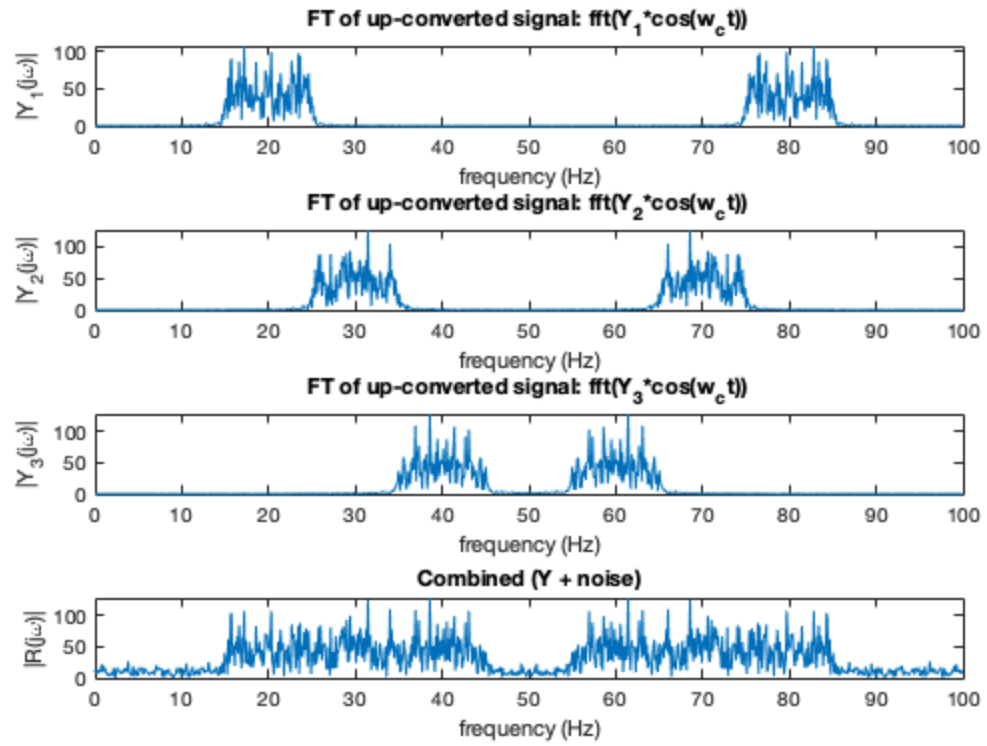
```

```

figure
subplot(2,1,1), plot(combined)
xlabel('time (s)'), ylabel('r(t)'), title('Noisy Received Signal')
subplot(2,1,2), plot(f,abs(fft(combined,Nfft)))
xlabel('frequency (Hz)'), ylabel('|R(j\omega)|'), title('FT of Noisy
Received Signal')

```







---

# Downconversion

```
xr = combined.*cos(wc(1)*tmax');
x2r = combined.*cos(wc(2)*tmax');
x3r = combined.*cos(wc(3)*tmax');

% Signal 1

xrl = lowpass(xr, 15, fs);

figure
subplot(2,1,1), plot(tmax,xrl)
xlabel('time (s)'), ylabel('y_r_e_c(t)'), title('Downconverted Signal 1')

subplot(2,1,2), plot(f,abs(fft(xrl,Nfft)))
xlabel('frequency (Hz)'), ylabel('|Y(j\omega)|'), title('FT of Downconverted Signal 1')

% Signal 2

x2rl = lowpass(x2r, 15, fs);

figure
subplot(2,1,1), plot(tmax,x2rl)
xlabel('time (s)'), ylabel('y_r_e_c(t)'), title('Downconverted Signal 2')

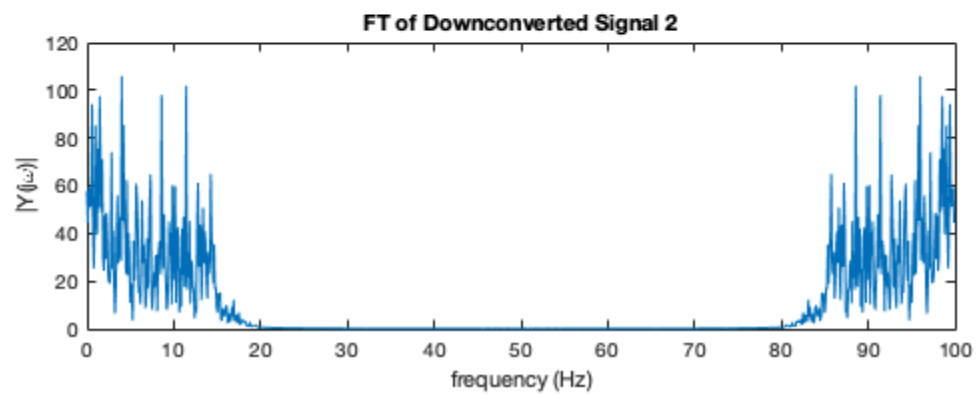
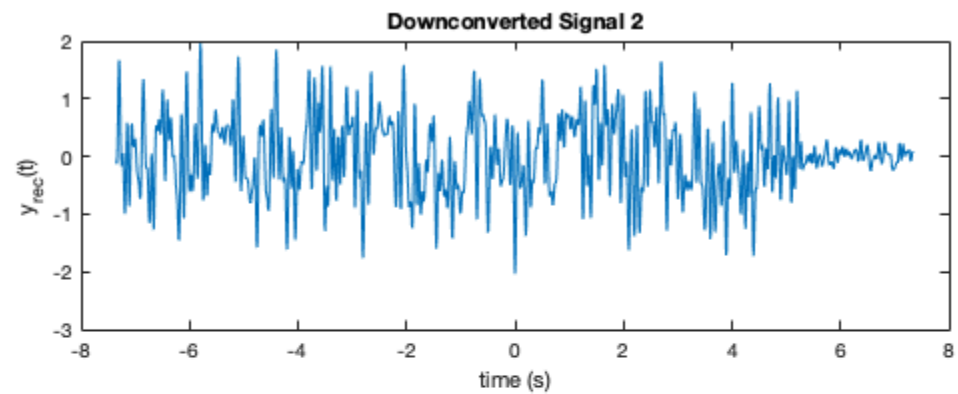
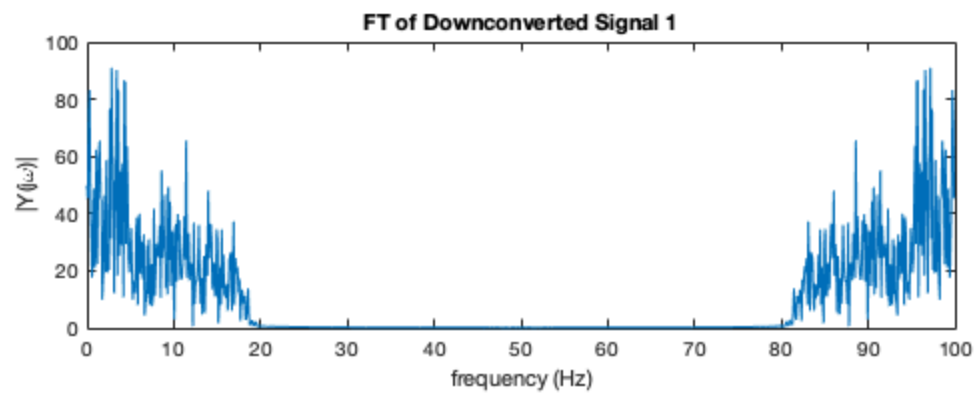
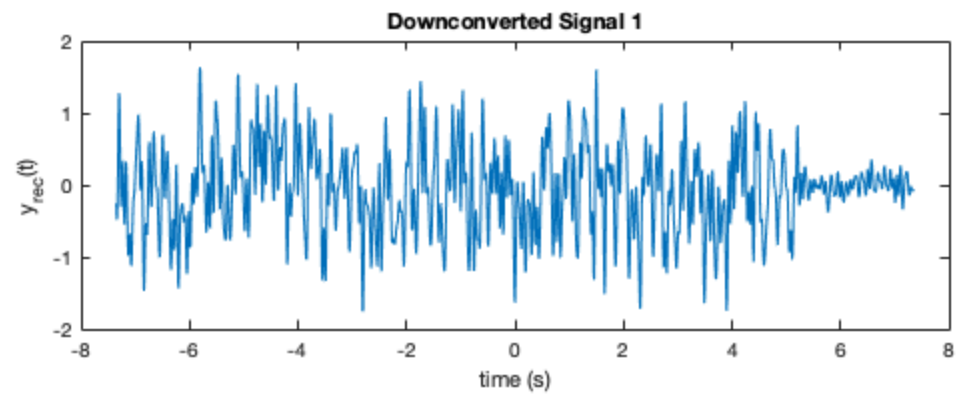
subplot(2,1,2), plot(f,abs(fft(x2rl,Nfft)))
xlabel('frequency (Hz)'), ylabel('|Y(j\omega)|'), title('FT of Downconverted Signal 2')

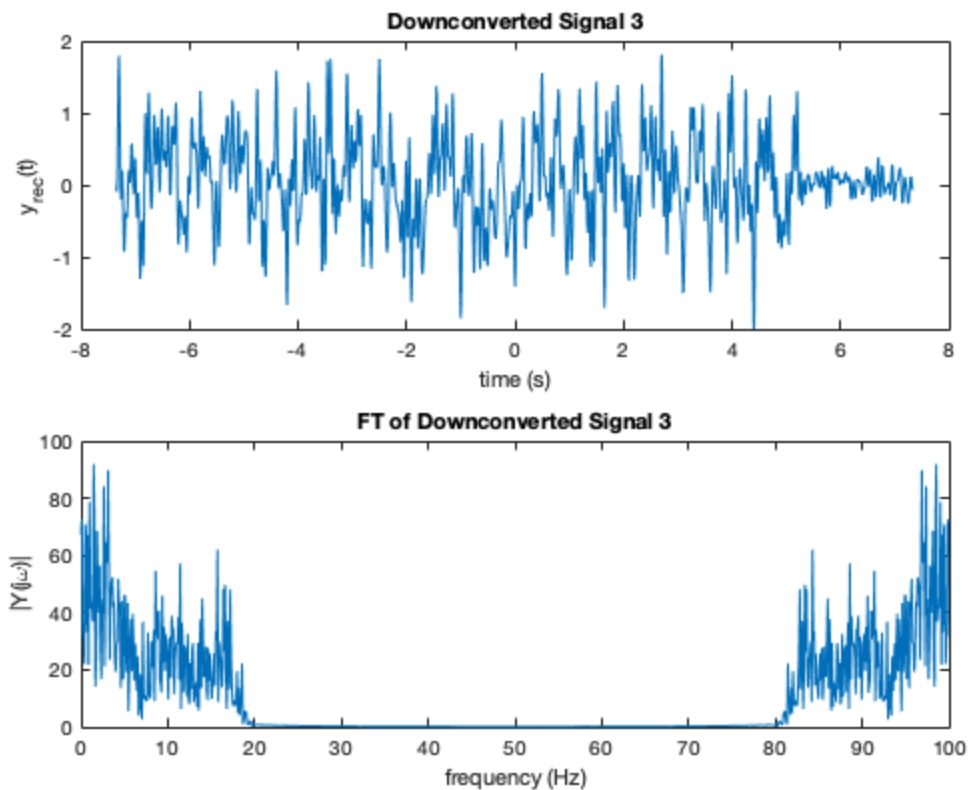
% Signal 3

x3rl = lowpass(x3r,15, fs);

figure
subplot(2,1,1), plot(tmax,x3rl)
xlabel('time (s)'), ylabel('y_r_e_c(t)'), title('Downconverted Signal 3')

subplot(2,1,2), plot(f,abs(fft(x3rl,Nfft)))
xlabel('frequency (Hz)'), ylabel('|Y(j\omega)|'), title('FT of Downconverted Signal 3')
```





## Matched Filter

```

z1 = conv(xr1,p, 'same');
z2 = conv(x2r1,p, 'same');
z3 = conv(x3r1,p, 'same');

x_hat = zeros(length(binary),1);
x2_hat = zeros(length(binary2),1);
x3_hat = zeros(length(binary3),1);

for i = 1:length(binary)
    if z1((i*c/Ts)) > 0
        x_hat(i) = 1;
    else
        x_hat(i) = -1;
    end
end

for i = 1:length(binary2)
    if z2((i*c/Ts)) > 0
        x2_hat(i) = 1;
    else
        x2_hat(i) = -1;
    end
end

```

---

```

        end
    end

    for i = 1:length(binary3)
        if z3((i*c/Ts)) > 0
            x3_hat(i) = 1;
        else
            x3_hat(i) = -1;
        end
    end

    totalErrors1 = 0;
    for i = 1:length(binary)
        if binary(i)~=x_hat(i)
            totalErrors1 = totalErrors1 + 1;
        end
    end
    errorRate = 100*totalErrors1/length(binary);

    totalErrors2 = 0;
    for i =1:length(binary2)
        if binary2(i)~=x2_hat(i)
            totalErrors2 = totalErrors2 + 1;
        end
    end
    errorRate2 = 100*totalErrors2/length(binary2);

    totalErrors3 = 0;
    for i =1:length(binary3)
        if binary3(i)~=x3_hat(i)
            totalErrors3 = totalErrors3 + 1;
        end
    end
    errorRate3 = 100*totalErrors3/length(binary3);

    errorRateAvg = mean([errorRate errorRate2 errorRate3]);
    % dataset(1,j,3) = errorRateAvg;

    figure

    subplot(3,1,1);
    hold on
    stem(binary);
    stem(x_hat);
    xlabel('bits')
    legend('x_n', 'decoded x_n');
    title('Matched-Filter Results');
    title("Matched-Filter 1 Results (error rate = " + errorRate + "%)");

    subplot(3,1,2);
    hold on
    stem(binary2);
    stem(x2_hat);
    xlabel('bits')

```

---

---

```

legend('x_n', 'decoded x_n');
title("Matched-Filter 2 Results (error rate = " + errorRate2 + "%)");

subplot(3,1,3);
hold on
stem(binary3);
stem(x3_hat);
xlabel('bits')
legend('x_n', 'decoded x_n');
title("Matched-Filter 3 Results (error rate = " + errorRate3 + "%)");

for i = 1:length(x_hat)
    if x_hat(i)==-1
        x_hat(i)=0;
    end
end
for i = 1:length(x2_hat)
    if x2_hat(i)==-1
        x2_hat(i)=0;
    end
end
for i = 1:length(x3_hat)
    if x3_hat(i)==-1
        x3_hat(i)=0;
    end
end

% Master Plot
figure
subplot(4,1,1);
plot(signal) % y(t)
title('Noise-Free PAM Signal, y_1(t)');
subplot(4,1,2);
plot(y) % y(t)cos(wt)
title('Up-converted signal, y(t)cos(w_ct)');
subplot(4,1,3);
plot(combined) % r(t) = y1 + y2 + y3 + noise
title('All 3 signals with noise');
subplot(4,1,4);
plot(xr1) % y - received
title('Received Signal 1');

messageOut1 = char(bin2dec(num2str(reshape(x_hat,7,[]))));
disp("Retrieved message 1: " + messageOut1);

messageOut2 = char(bin2dec(num2str(reshape(x2_hat,7,[]))));
disp("Retrieved message 2: " + messageOut2);

messageOut3 = char(bin2dec(num2str(reshape(x3_hat,7,[]))));
disp("Retrieved message 3: " + messageOut3);

```

---

---

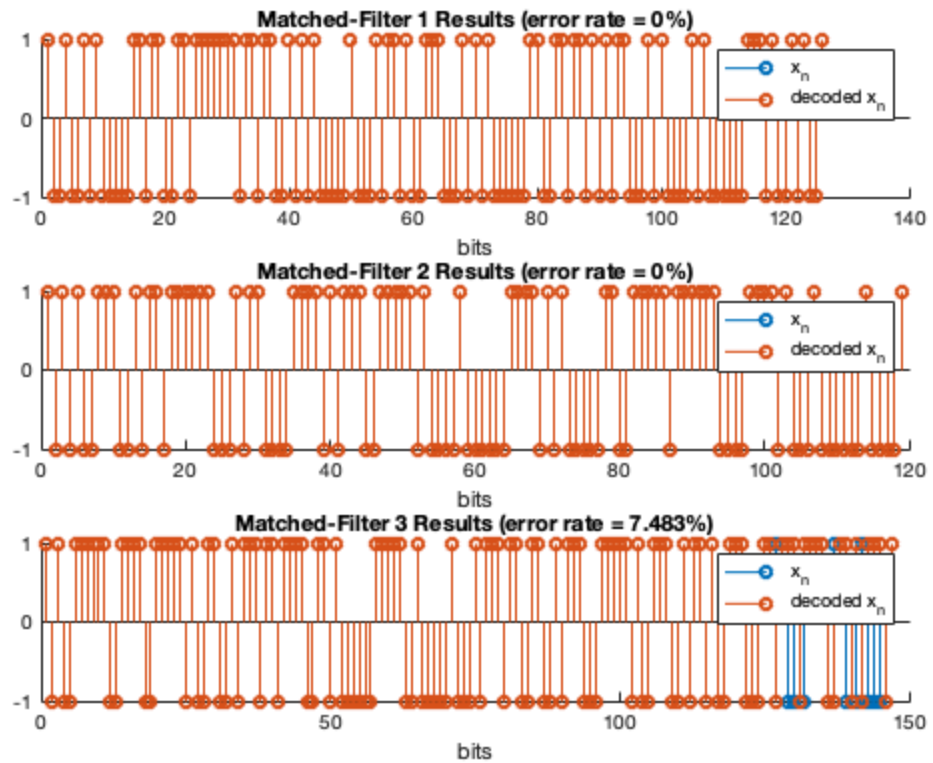
```
disp("Noise Level: " + sigma);
```

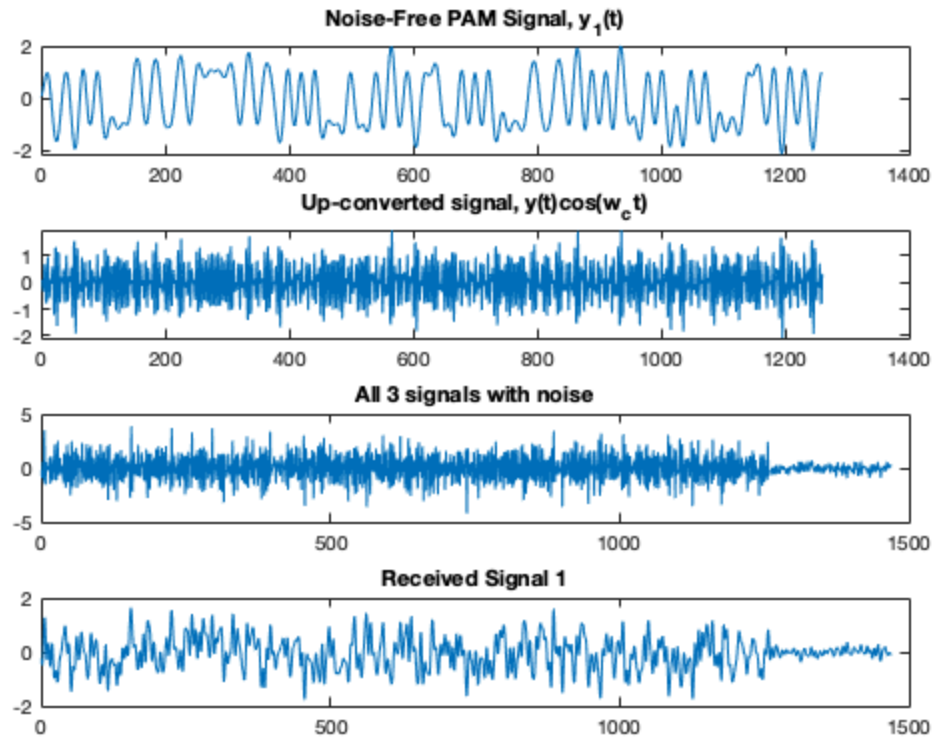
```
Retrieved message 1: I love ESE 351! :)
```

```
Retrieved message 2: Trobaugh = goat !
```

```
Retrieved message 3: Systems > Electric;f]
```

```
Noise Level: 0.33333
```





```
x = 0:9; figure hold on scatter(x,dataset(1,:,1)); scatter(x,dataset(2,:,1)); scatter(x,dataset(3,:,1)); ti-
tle('f=20/30/40'),xlabel('Noise'),ylabel('Error'),legend('Normal', 'Square', 'Sinc'); ylim([0 50]);
```

```
figure hold on scatter(x,dataset(1,:,2)); scatter(x,dataset(2,:,2)); scatter(x,dataset(3,:,2)); ti-
tle('f=15/30/45'),xlabel('Noise'),ylabel('Error'),legend('Normal', 'Square', 'Sinc'); ylim([0 50]);
```

```
figure hold on scatter(x,dataset(1,:,3)); scatter(x,dataset(2,:,3)); scatter(x,dataset(3,:,3)); ti-
tle('f=10/30/50'),xlabel('Noise'),ylabel('Error'),legend('Normal', 'Square', 'Sinc'); ylim([0 50]);
```

*Published with MATLAB® R2019a*