EECS 455: Solutions to Problem Set 4

1. A data signal consists of an infinite sequence of rectangular pulses of duration T. That is

$$s(t) = \sum_{l=-\infty}^{\infty} b_l p_T(t - lT)$$

where $p_T(t)$ is 1 for $0 \le t \le T$ and zero elsewhere. The data is represented by b_l and is either +1 or -1. The signal is filtered by a low pass RC filter with impulse response

$$h(t) = \alpha e^{-\alpha t} u(t)$$

where $u(t)$ is one for $t > 0$ and is 0 otherwise. The filter output is sampled every T seconds.

(a) Find an expression for the output of the filter at time T in terms of $b_0, b_{-1}, b_{-2}, \ldots$.

(b) Suppose that $b_0 = +1$. Find the largest and smallest possible value (over all possible data sequences except b_0) of the sampled output

Solution:

The output of the receiver filter at the sampling time T is

$$v(T) = b_0(1 - e^{-\alpha T}) + \sum_{m=1}^{\infty} b_{-m}(1 - e^{-\alpha T})e^{-m\alpha T}$$

Analytical derivation of solution (Not necessary if you know how a RC filter charges and discharges from EECS 215):

$$\begin{aligned}
v(T) &= \int_{-\infty}^{\infty} h(T - \tau)s(\tau)d\tau \\
&= \int_{-\infty}^{T} \alpha e^{-\alpha(T-\tau)} s(\tau)d\tau \\
&= \int_{-\infty}^{T} \alpha e^{-\alpha(T-\tau)} \sum_{l=-\infty}^{\infty} b_l p_T(\tau - lT)d\tau \\
&= \sum_{l=-\infty}^{\infty} b_l \int_{-\infty}^{T} \alpha e^{-\alpha(T-\tau)} p_T(\tau - lT)d\tau
\end{aligned}$$

Now $p_T(\tau - lT)$ is nonzero only when $0 \le \tau - lT \le T$ or $lT \le \tau \le (l+1)T$. Thus

$$\begin{aligned}
\int_{-\infty}^{T} \alpha e^{-\alpha(T-\tau)} p_T(\tau - lT)d\tau &= \begin{cases} 0, & l > 0 \\ \int_0^T \alpha e^{-\alpha(T-\tau)} d\tau, & l = 0 \\ \int_{lT}^{(l+1)T} \alpha e^{-\alpha(T-\tau)} d\tau, & l \le -1 \end{cases} \\
&= \begin{cases} 0, & l > 0 \\ e^{-\alpha(T-\tau)}|_0^T, & l = 0 \\ e^{-\alpha(T-\tau)}|_{lT}^{(l+1)T}, & l \le -1 \end{cases}
\end{aligned}$$

$$= \begin{cases} 0, & l > 0 \\ 1 - e^{-\alpha T}, & l = 0 \\ e^{-\alpha(T-(l+1)T)} - e^{-\alpha(T-lT)}, & l \leq -1 \end{cases}$$

$$= \begin{cases} 0, & l > 0 \\ 1 - e^{-\alpha T}, & l = 0 \\ e^{-\alpha(-lT)}[1 - e^{-\alpha T}], & l \leq -1 \end{cases}$$

Now it is easy to see that the answer above is correct.

The largest possible value is obtained when the data sequence is all ones. The filter output corresponding to that data sequence is

$$\begin{aligned} \hat{s}(T) &= \int_{-\infty}^{T} h(T-\tau)d\tau \\ &= 1 \end{aligned}$$

The smallest possible positive value is obtained when the data sequence is all negative except the last bit. The output at time 0 due to a constant negative pulse starting at time $-\infty$ and ending at time 0 is -1. At time T the output due to this pulse is $-e^{-\alpha T}$. The output at time T due to a positive pulse starting at time 0 and ending at time T is $1 - e^{-\alpha T}$. The total output is then the sum which is $1 - 2e^{-\alpha T}$.

2. Consider a communication system that transmits an infinite sequence of data bits $\{b_l\}_{l=-\infty}^{\infty}$ using two signals of duration T: $s_0(t) = -s_1(t) = Ap_T(t)$. Thus

$$s(t) = \sum_{l=-\infty}^{\infty} b_l A p_T(t - lT)$$

The signal $s(t)$ is transmitted over an additive white Gaussian noise channel with spectral density $N_0/2$. The receiver consists of a filter $h(t)$ the output of which is sampled at time iT and compared with a threshold of 0. If the output at time $iT > 0$ the receiver decides $b_{i-1} = +1$ otherwise the receiver decides $b_{i-1} = -1$. It is know that the filter is such that $\int_{-\infty}^{\infty} h^2(t)dt = 16$. It is also known that if the input to the filter is $p_T(t)$ ($X(t) = 0$) then the output at time iT is

$$Z(iT) = \begin{cases} 8 & i = 1 \\ 2 & i = 2 \\ 1 & i = 3 \\ 0 & i < 1,\ i > 3 \end{cases}$$

(a) Find the upper and lower bounds for the $E[Z(iT)]$ if $b_{i-1} = +1$. Find upper and lower bounds on the probability of error for data bit b_{i-1} given that $b_{i-1} = +1$.

Solution:

If $b_{i-} = +1$ then $E[Z(iT)]$ can be as large as 8+2+1=11 while $E[Z(iT)]$ can be as small as 8-2-1=5.

(b) Give an expression for the average probability of error for the data bit b_{i-1} if each data bit is equally likely to be +1 or -1 independently of all other data bits.

Solution: There are four possible outputs given $b_{i-1}+1$ (in the absence of noise). These are 11, 9, 7, and 5. These are equally likely. The probability of error is then

$$P_{e,}\bar{+}1 = \frac{1}{4}\left[Q(\frac{11}{\sigma})+Q(\frac{9}{\sigma})+Q(\frac{7}{\sigma})+Q(\frac{5}{\sigma})\right]$$

3. A communication system transmits one of three signals:

$$s_0(t) = A\cos(2\pi f_c t)p_T(t)$$

$$s_1(t) = 0$$

$$s_2(t) = -A\cos(2\pi f_c t)p_T(t)$$

over an additive white Gaussian noise channel with spectral density $N_0/2$. Let $r(t)$ denote the received signal ($r(t) = s_i(t)+n(t)$). The receiver computes the quantity

$$Z = \int_0^T r(t)\cos(2\pi f_c t)dt.$$

Assume $2\pi f_c T = 2\pi n$ for some integer n. Z is compared with a threshold γ and a threshold $-\gamma$. If $Z > \gamma$, the decision is made that $s_0(t)$ was sent. If $Z < -\gamma$, the decision is made that $s_2(t)$ was sent. If $-\gamma < Z < \gamma$ the the decision is made in favor of $s_1(t)$

(a) Determine the three conditional probabilities of error: $P_{e,0} =$ probability of error given s_0 sent, $P_{e,1}$ =probability of error given s_1 sent, and $P_{e,2}$

Solution: Assume signal 0 is transmitted. The decision variable is

$$\begin{aligned}
Z &= \int_0^T r(t)\cos(2\pi f_c t)dt. \\
Z &= \int_0^T (s_0(t)+n(t))\cos(2\pi f_c t)dt. \\
&= \int_0^T A\cos(2\pi f_c t)\cos(2\pi f_c t)dt+\eta. \\
&= \int_0^T A[1/2+1/2\cos(2\pi f_c t)]dt+\eta. \\
&= \int_0^T A[1/2+1/2\cos(2\pi f_c t)]dt+\eta. \\
&= AT/2+\eta.
\end{aligned}$$

where η is a Gaussian random variable. The mean of η is zero and the variance of η is calculated as

$$\sigma^2 = \mathrm{Var}\{\eta\} = E[\int_0^T n(t)\cos(2\pi f_c t)dt \int_0^T n(s)\cos(2\pi f_c s)ds]$$

$$
\begin{aligned}
&= \int_0^T \int_0^T E[n(t)n(s)] \cos(2\pi f_c t)\cos(2\pi f_c s)dtds \\
&= \int_0^T \int_0^T \frac{N_0}{2}\delta(t-s)\cos(2\pi f_c t)\cos(2\pi f_c s)dtds \\
&= \int_0^T \frac{N_0}{2}\cos^2(2\pi f_c t)dt \\
&= \frac{N_0}{2}\int_0^T [1/2+1/2\cos(2\pi f_c t)]dt \\
&= \frac{N_0}{2}T/2 \\
&= \frac{N_0 T}{4}
\end{aligned}
$$

The probability of error given signal 0 transmitted is then

$$
\begin{aligned}
P_{e,0} &= P\{AT/2+\eta<\gamma\} \\
&= P\{\eta<\gamma-AT/2\} \\
&= \int_{-\infty}^{\gamma-AT/2} \frac{1}{\sqrt{2\pi}\sigma} e^{-u^2/(2\sigma^2)}du \\
&= Q(\frac{AT/2-\gamma}{\sigma})
\end{aligned}
$$

Similarly

$$
\begin{aligned}
P_{e,2} &= P\{-AT/2+\eta>\gamma\} \\
&= Q(\frac{AT/2-\gamma}{\sigma}).
\end{aligned}
$$

Finally

$$
\begin{aligned}
P_{e,1} &= 1-P\{-\gamma<\eta<\gamma\} \\
&= 1-[\Phi(\gamma/\sigma)-\Phi(-\gamma/\sigma)] \\
&= 1-Q(-\gamma/\sigma)+Q(\gamma/\sigma) \\
&= 2Q(\gamma/\sigma)
\end{aligned}
$$

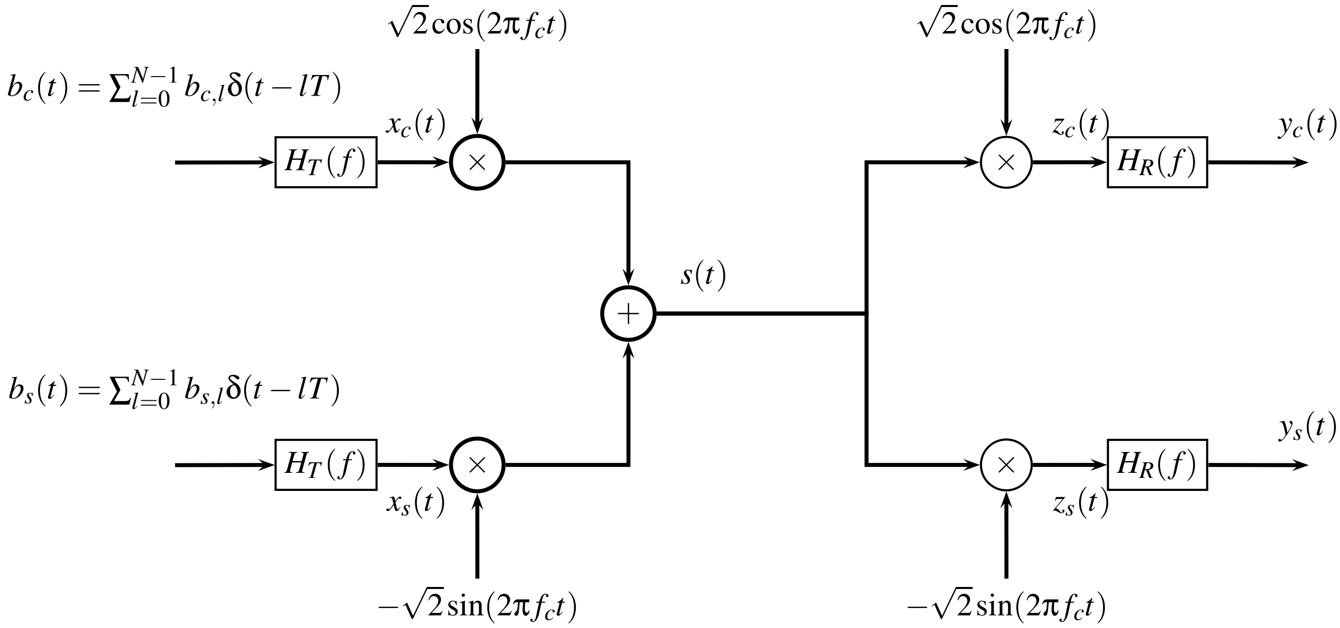
(b) Determine the average error probability assuming that all three signals are equally probable of being transmitted.

Solution: The average error probability is

$$\bar{P}_e = \frac{1}{3}P_{e,0}+\frac{1}{3}P_{e,1}+\frac{1}{3}P_{e,2}$$

4. Simulate the communication system shown below. The system uses square-root raised cosine pulses with $\alpha = 0.35$ data amplitude ± 1 to transmit data on the cosine and sine branch with data bit duration $T = 1$ second. Show the data waveform, the transmitted signal in the

time domain $s(t)$ and frequency domain $S(f)$. Plot the output of the mixers at the receiver in the frequency domain (show the double frequency terms). The receiver uses matched filters on each branch (so it is also square-root raised cosine). Plot the waveform at the output of the receiver filters for a sequence of 8 bits. Make an eye diagram for one of the outputs (use 512 bits for the eye diagram). Assume $T = 1$ and $f_c = 4$ for the simulation. Use a maximum frequency for the simulation of 16Hz.



Solution:

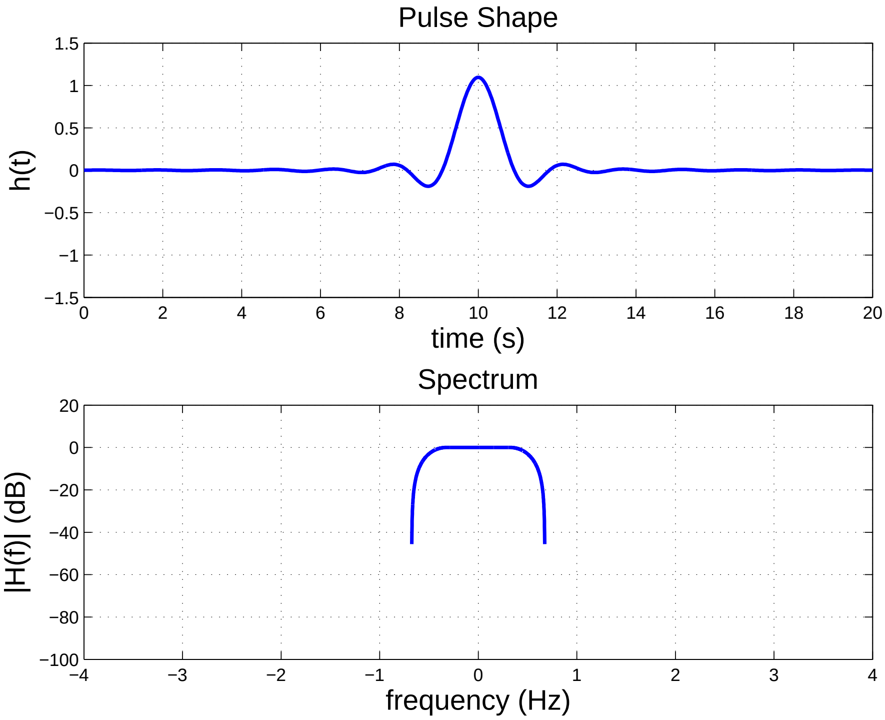


Figure 1: Inpulse Response and Frequency Response of Filter

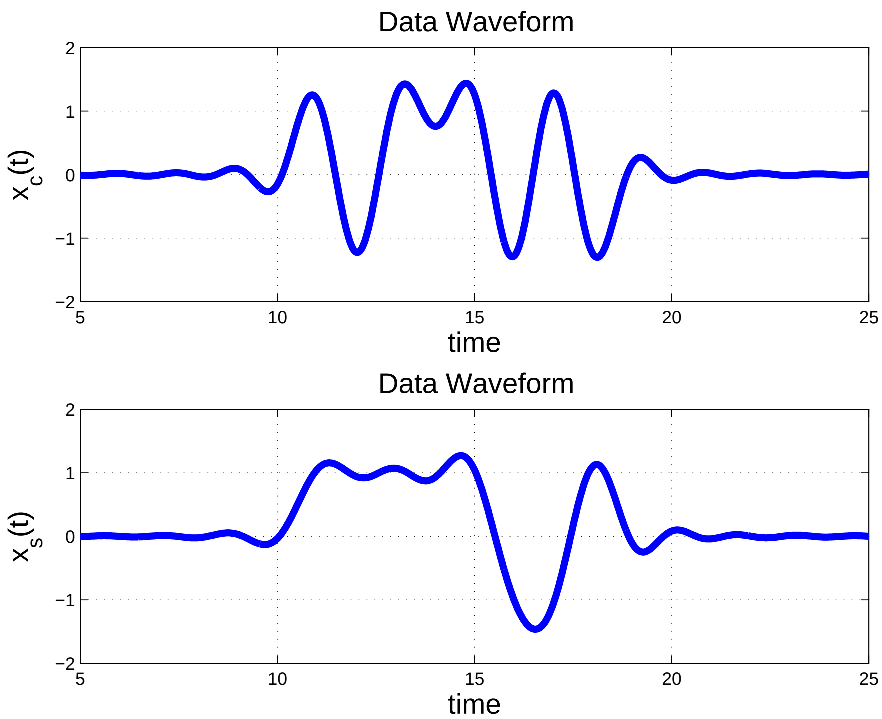


Figure 2: Output of Filters

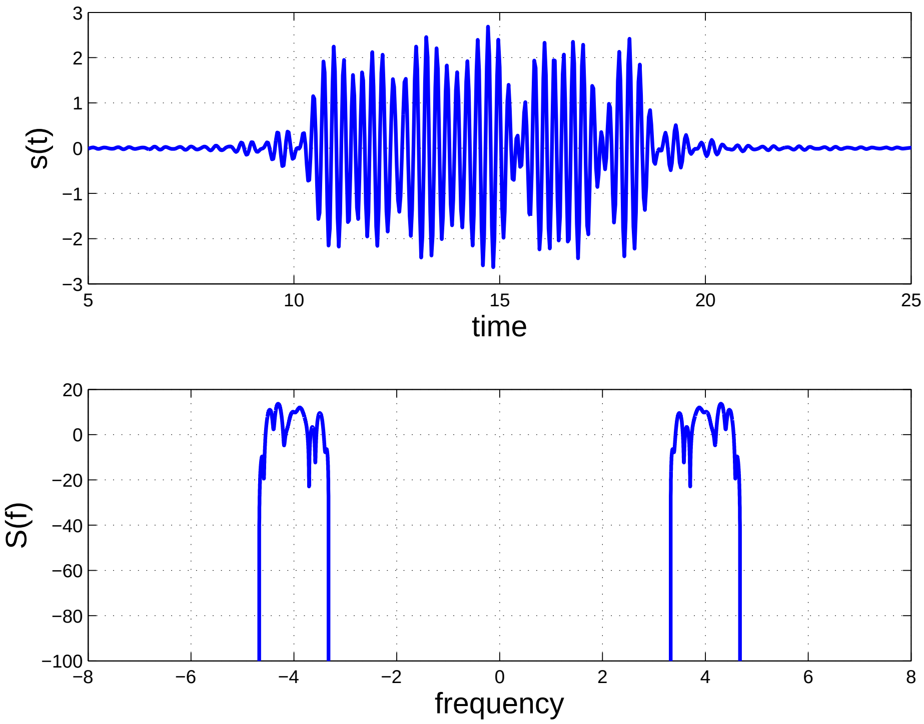


Figure 3: Upconverted Signal

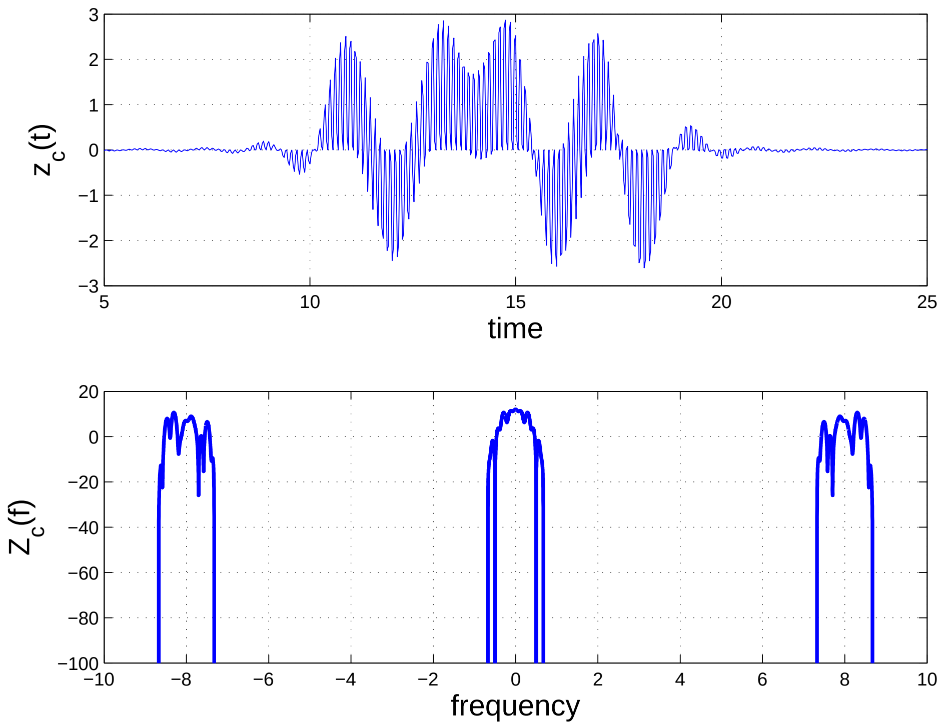


Figure 4: Signal after mixing (containing double frequency components)

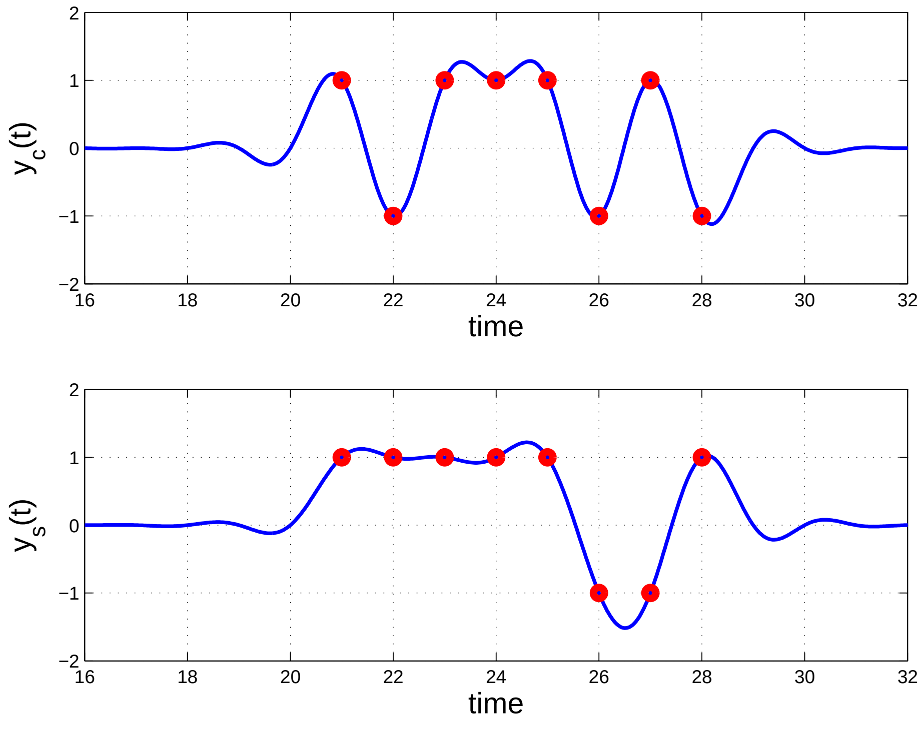


Figure 5: Signal after filtering

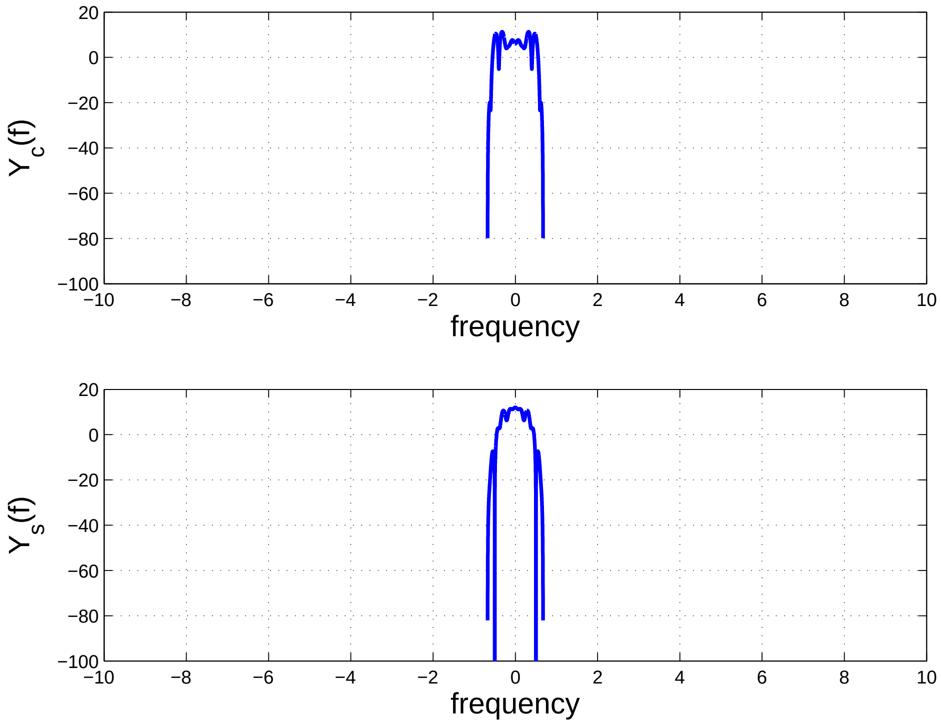


Figure 6: Signal after filtering

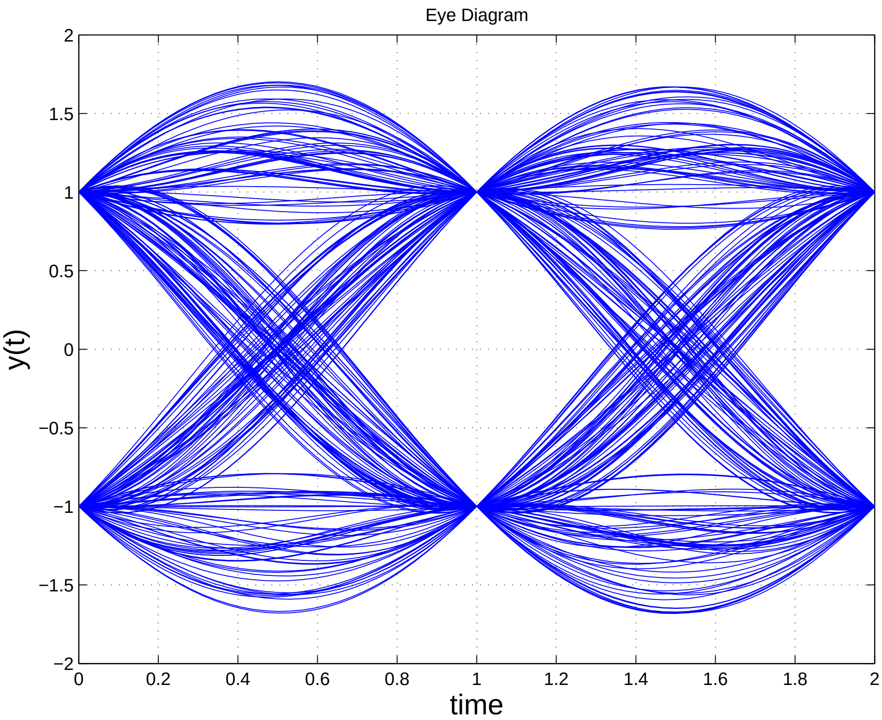


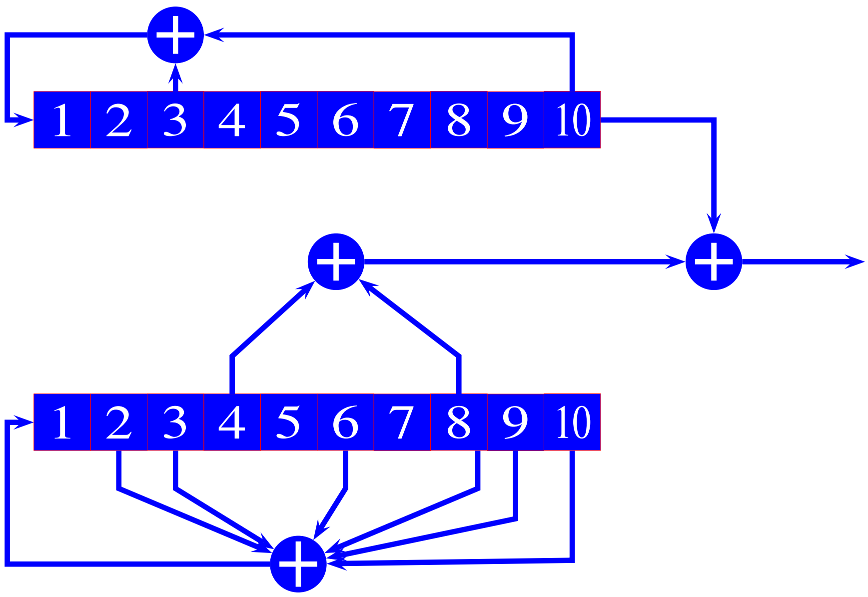
Figure 7: Eye Diagram

```
%  This program simulates a raised cosine pulse shape signal
%  filtered by a raised cosine filter.

clear;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                       %
%                Simulation Parameters                  %
%                                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


Tb=1;                     % Bit duration of data
%EbN0dB=input('E_b/N_0 (dB)= ');
%EbN0=10^(EbN0dB/10);
%P=1;                       % Received Power
%Eb=P*Tb;                  % Received Energy
%N0=Eb/EbN0;               % Noise power
%beta=input('Raised Cosine Filter Parameter=');
alpha=0.35
%Nb=input('Number of bits simulated= ');
Nb=8
%fmax=input('Maximum frequency of simulation= ');
fmax=16
%N=input('Number of samples = ');
N=2*16384
fs=2*fmax;                     % Sampling Frequency
N0=0.0004
df=2*fmax/N                    % Frequency spacing
dt=1./(df.*N)                  % Time spacing
t=(1:N)*dt-dt;                 % Time samples
Tmax=N*dt                      % Simulation time
f(1:N/2)=(1:N/2)*df-df;        % Nonnegative frequency samples
f(N/2+1:N)=-fmax+(0:N/2-1)*df; % Negative frequency samples
f2=fftshift(f);
Nsb=Tb/dt
rb=1/Tb;                  % Data rate
fc=4                      % Center Frequency

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                       %
%                Generate Signals                       %
%                                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N1=round((1-alpha)/(2*Tb)/df)   % This is the index corresponding to freq. (1-beta)/(2Tb)
N2=round((1+alpha)/(2*Tb)/df)   % This is the index corresponding to freq. (1+beta)/(2Tb)
h1f(1:N)=zeros(1,N);
h1f(1:N1)=sqrt(Tb*ones(1,N1));
h1f(N1:N2)=sqrt(0.5*Tb*(1-sin(pi*Tb/alpha*(f(N1:N2)-1/(2*Tb)))));;
h1f(1:N2)=h1f(1:N2).*exp(-j*2*pi*f(1:N2)*10*Tb);      %Include a delay to make it nearly causal
h1f(N/2+2:N)=conj(fliplr(h1f(2:N/2)));
h1t=real(ifft(h1f)./dt);

figure(1)
subplot(2,1,1), plot(t/Tb,h1t,'LineWidth',2);
grid
axis([0 20*Tb -1.5 1.5])
xlabel('time (s)','FontSize',16);
ylabel('h(t)','FontSize',16);
title('Pulse Shape','FontSize',16)
subplot(2,1,2), plot(f2(1:N),20*log10(fftshift(abs(h1f(1:N)))),'LineWidth',2);
axis([-fmax/4 fmax/4 -100 20])
grid
xlabel('frequency (Hz)','FontSize',16);
ylabel('|H(f)| (dB)','FontSize',16);
title('Spectrum','FontSize',16)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                       %
%          Generate data for cosine branch              %
%                                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

bc=sign(rand(1,Nb)-0.5);
bcf=zeros(1,N);
for k=1:Nb
  bcf=bcf+bc(k)*exp(-j*2*pi*f*k*Tb);
end

xcf=h1f.*bcf;
xct=real(ifft(xcf)./dt);

figure(2)
subplot(2,1,1)
    hold off
    plot(t/Tb,xct,'LineWidth',4);
grid
axis([5 25 -2 2])
xlabel('time','FontSize',16);
ylabel('x_c(t)','FontSize',16);
title('Data Waveform','FontSize',16)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%                                                         %
%          Generate data for sine branch                  %
%                                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

bs=sign(rand(1,Nb)-0.5);
bsf=zeros(1,N);
for k=1:Nb
bsf=bsf+bs(k)*exp(-j*2*pi*f*k*Tb);
end

xsf=h1f.*bsf;
xst=real(ifft(xsf)./dt);

figure(2)
subplot(2,1,2)
hold off
plot(t/Tb,xst,'LineWidth',4);
grid
axis([5 25 -2 2])
xlabel('time','FontSize',16);
ylabel('x_s(t)','FontSize',16);
title('Data Waveform','FontSize',16)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                         %
%                Mix the signal to a carrier              %
%                                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

st=sqrt(2)*(xct.*cos(2*pi*fc*t)-xst.*sin(2*pi*fc*t));
sf=fft(st)*dt;

figure(3)
subplot(2,1,1), plot(t/Tb,st,'LineWidth',2)
xlabel('time','FontSize',16)
ylabel('s(t)','FontSize',16)
grid
axis([5 25 -3 3])
subplot(2,1,2), plot(f2,20*log10(fftshift(abs(sf))),'LineWidth',2)
grid
xlabel('frequency','FontSize',16)
ylabel('S(f)','FontSize',16)
axis([-8 8 -100 20])
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                         %
%                Mix the signal back to baseband          %
%                                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

zct=sqrt(2)*st.*cos(2*pi*fc*t);
zcf=fft(zct)*dt;
zst=-sqrt(2)*st.*sin(2*pi*fc*t);
zsf=fft(zst)*dt;

figure(4)
hold off
title('Output of Mixer at Receiver')
subplot(2,1,1), plot(t/Tb,zct)
axis([5 25 -3 3])
grid on
xlabel('time','FontSize',16)
ylabel('z_c(t)','FontSize',16)
subplot(2,1,2), plot(f2,20*log10(fftshift(abs(zsf))),'LineWidth',2)
grid on
axis([-10 10 -100 20])
xlabel('frequency','FontSize',16)
ylabel('Z_c(f)','FontSize',16)



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                         %
%                Filter the signal                        %
%                                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

ycf=zcf.*h1f;
ysf=zsf.*h1f;
yct=real(ifft(ycf)./dt);
yst=real(ifft(ysf)./dt);


figure(5)
subplot(2,1,1)
plot(t/Tb,yct,'b','LineWidth',2)   %Advance the signal to match transmitted for plotting
grid on
xlabel('time','FontSize',16)
ylabel('y_c(t)','FontSize',16)
axis([16 32 -2 2])
t_first_data=21*Nsb;
tsample=t_first_data+1:Nsb:t_first_data+7*32+1;
```

```
hold on
plot(t(tsample),yct(tsample),'or','LineWidth',4)
hold off
subplot(2,1,2)
plot(t/Tb,yst,'b','LineWidth',2)
hold on
plot(t(tsample),yst(tsample),'or','LineWidth',4)
hold off
grid on
xlabel('time','FontSize',16)
ylabel('y_s(t)','FontSize',16)
axis([16 32 -2 2])


figure(6)
for k=2:floor(Nb/2)-11
plot(t(1:2*Nsb+1)/Tb,yct((k*2+20)*Nsb+1:(k*2+22)*Nsb+1))
hold on
end
xlabel('time','FontSize',16);
ylabel('y(t)','FontSize',16);
title('Eye Diagram')
grid on
hold off
```

5. A signal $s(t)$ of duration $T = 1ms$ from one of the GPS satellites consists of 1023 consecutive pulses (of duration $T/1023$) of amplitude ± 1. The sequence of amplitudes are determined by a pair of 10 stage shift registers shown below. Every $T/1023$ the binary (0 or 1) contents are shifted right and a new bit is produced at the output.

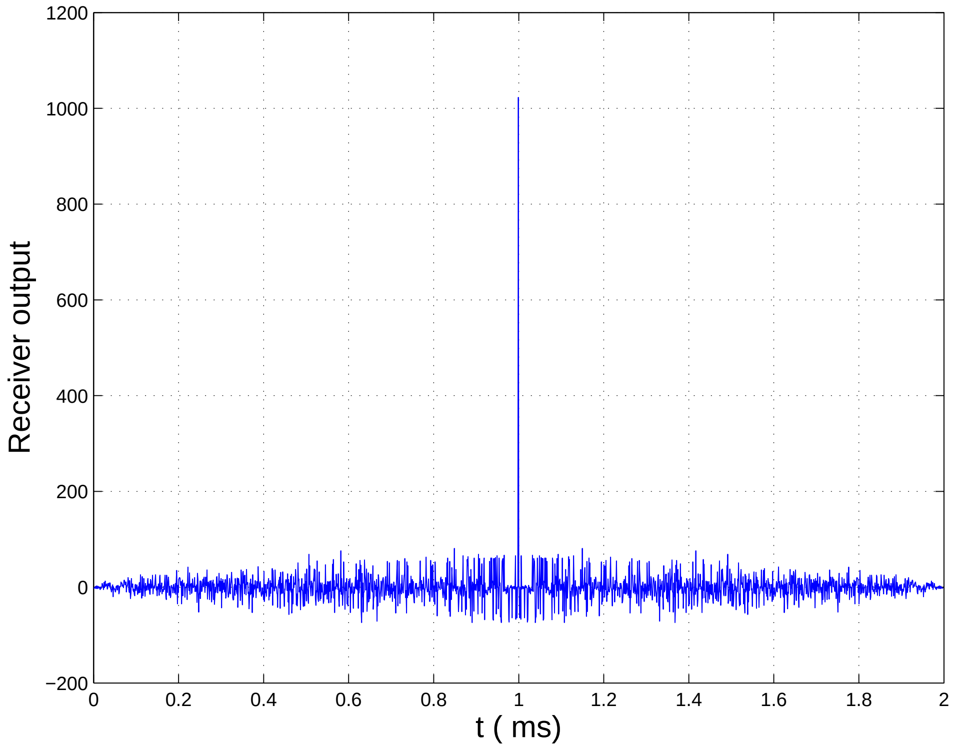


(a) The contents of the shift register at the start (time 0) are both all ones. The addition is done modulo 2 (0+0=0,0+1=1,1+0=1,1+1=0). Generate the sequence of bits from this shift register. The first 5 bits are 11110. What are the first 20 bits of the output? How many zeros and how many ones are produced in the sequence of 1023 bits.

Solution: The first 20 bits are 11110010001110001000. There are 512 ones and 511 zeros.

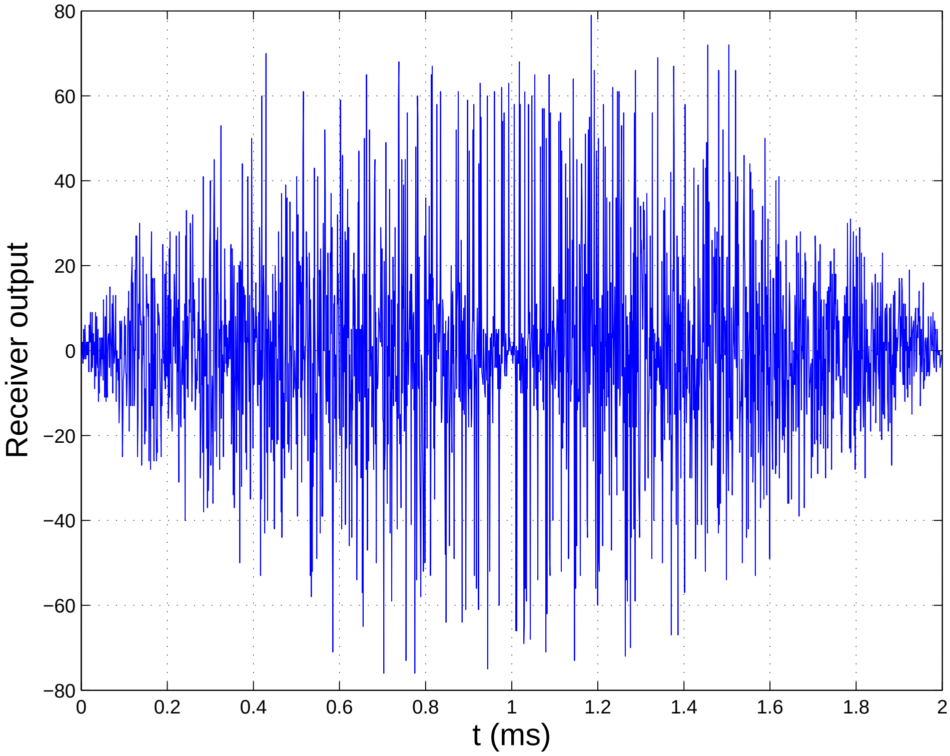
(b) Assume that that a signal consisting of pulses with amplitude determined by this shift register generates a signal $s(t)$. A zero bit at the output is mapped to $+1$ while a one bit at the output is mapped to -1. This signal is input to a linear time-invariant system (filter) with impulse response $h(t) = s(T - t)$. Find (plot) the output of the filter. (Use Matlab).

Solution:



(c) Now consider a signal from a different satellite whereby the sequence is produced as before except that contents of the second and sixth stage of the bottom shift register are added (exclusive or) with the output of the top shift register to produce the sequence. Determine (plot) the output of the same filter as in part (b) to the new signal. Determine the maximum (positive) output and the minimum (negative) output.

Solution: The maximum positive output is 79 and the minimum (negative) output is -76.




```
%==============================================%
%                                              %
%              GPS Gold Codes                  %
%           Generates different codes          %
%==============================================%
tap1=[2 3 4 5 1 2  1 2 3  2 3 5 6 7 8 9  1 2 3 4 5 6 1 4 5 6 7 8  1 2 3 4 ];
tap2=[6 7 8 9 9 10 8 9 10 3 4 6 7 8 9 10 4 5 6 7 8 9 3 6 7 8 9 10 6 7 8 9];
% These are the various taps used in GPS satellites

Nc=1023  %Number of chips per bit
a1(1:10)=[1 1 1 1 1 1 1 1 1 1];   %Starting phases
a2(1:10)=[1 1 1 1 1 1 1 1 1 1];
for i=11:Nc
    a1(i)=rem(a1(i-10)+a1(i-3),2);
    a2(i)=rem(a2(i-10)+a2(i-9)+a2(i-8)+a2(i-6)+a2(i-3)+a2(i-2),2);
end

for h=1:32
    h
    a3=rem(circshift(a2',tap1(h)-10)'+circshift(a2',tap2(h )-10)',2);
    gzo(h,:)=rem(a1+a3,2);
end
g=(-1).^gzo;
```

```
gzo(3,1:20)           % The third sequence is the desired one for this homew

%=============================================%
%         Part a                              %
%=============================================%

numberones=sum(gzo(3,:))    % Number of ones in the third sequence
    % (1023-this is number of zeros)
nmbrzeros=1023-numberones


%=============================================%
%         Part b                              %
%=============================================%

h=fliplr(g(3,:));             % Filter
yb=conv(h,g(3,:));
figure(1)
t=(0:length(yb)-1)/1023        % Time in microseconds
plot(t,yb)
xlabel('t (ms)','FontSize',16)
ylabel('Receiver output','FontSize',16)
grid on

%=============================================%
%         Part c                              %
%=============================================%


yc=conv(h,g(1,:));             % New received signal
figure(2)
plot(t,yc)
xlabel('t ( ms)','FontSize',16)
ylabel('Receiver output','FontSize',16)
grid on

max(yc)
min(yc)
%=============================================%
%         Not Assigned                        %
%=============================================%

yd=yb+yc;             % New received signal
figure(3)
plot(t,yd)
xlabel('t (ms)','FontSize',16)
ylabel('Receiver output','FontSize',16)
```