Assignment 2

Name	Section	BN	Student code
Abdelrahman Mohamed Salem Hassan	1	38	9202794
Ahmed Fawzy Mohamed Ibrahim	1	8	9202153

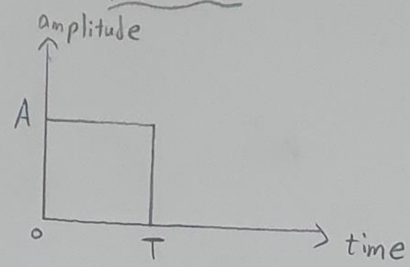
Table of Contents

Table of Contents

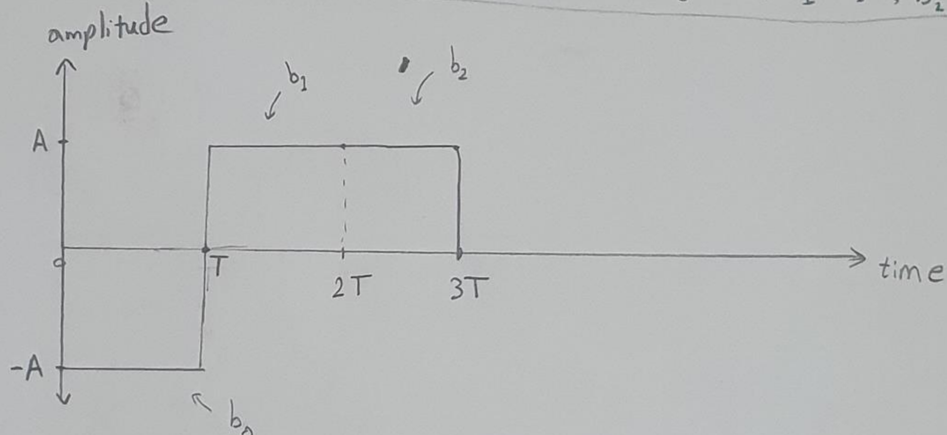
Part1 Solution

part 1

$s(t) \Rightarrow$

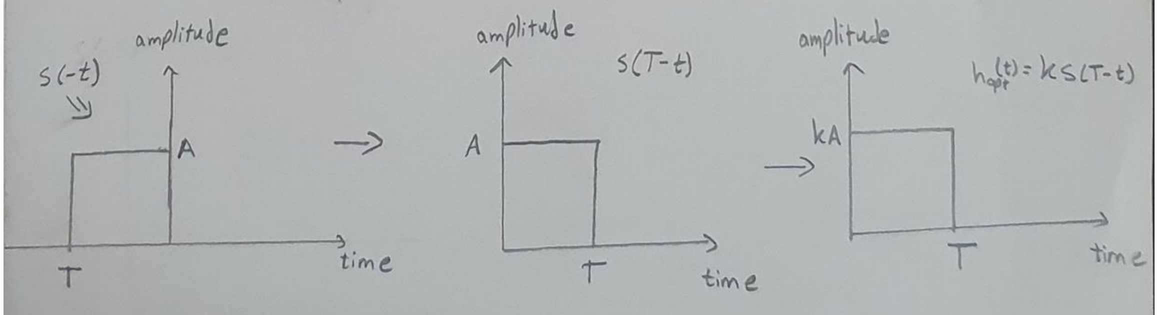


ⓐ transmitted baseband for bit sequence: $b_0 = '0'$, $b_1 = '1'$, $b_2 = '1'$



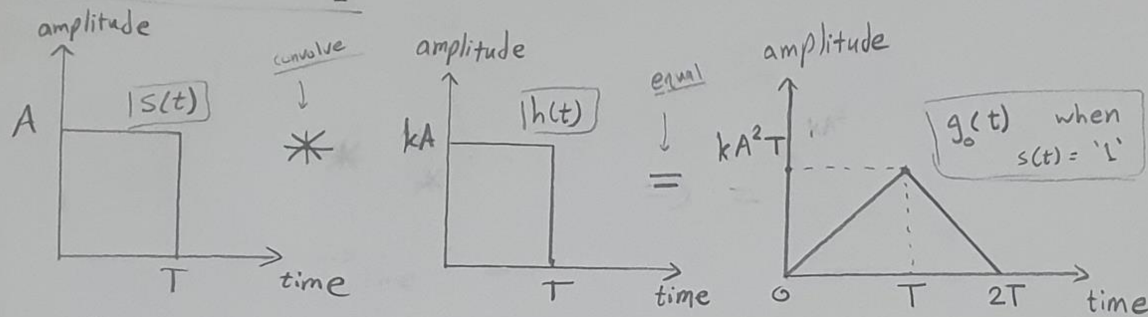
ⓑ output of matched filter

first we get $h_{opt}(t) = k\,s(T-t)$



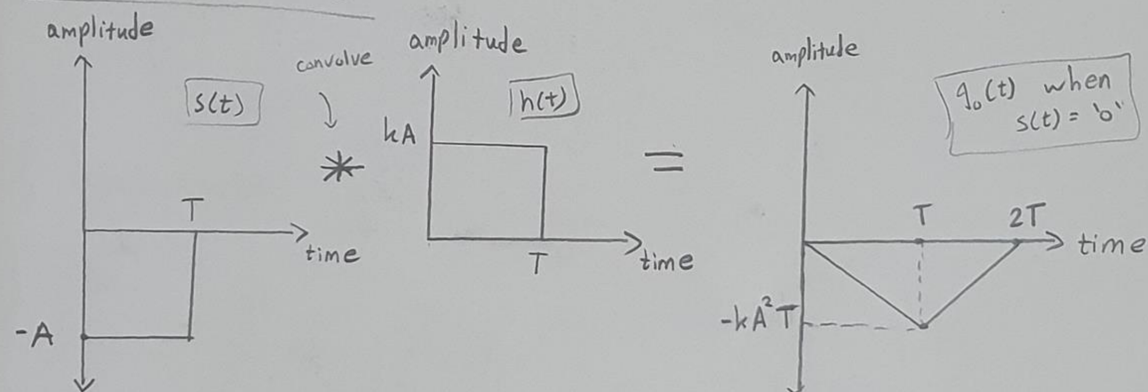
second we get convolution of $h_{opt}(t)$ with $s(t)$

case $s(t)$ was '1'



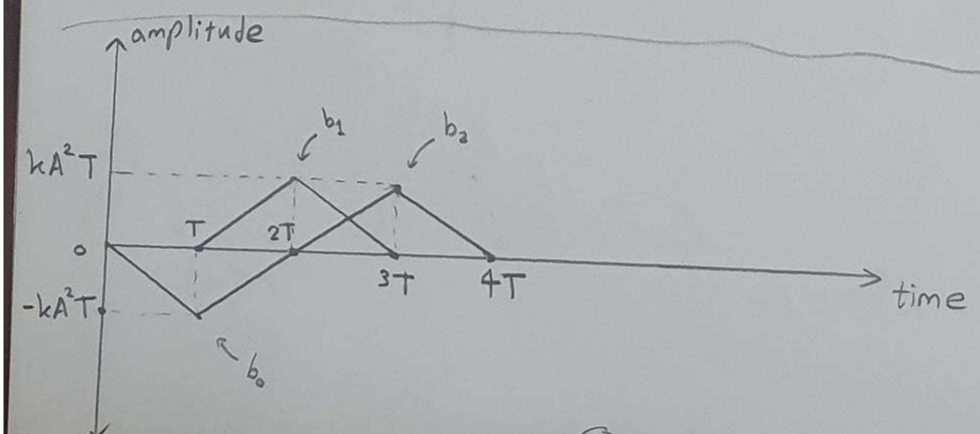
P.S: $\int_{-\infty}^{\infty} s(\tau) . h_{opt}(t-\tau)\, d\tau = \int_{0}^{T} (A)(kA)\, d\tau = [kA^2 \tau]_0^T = kA^2T$

case $s(t)$ was '0'

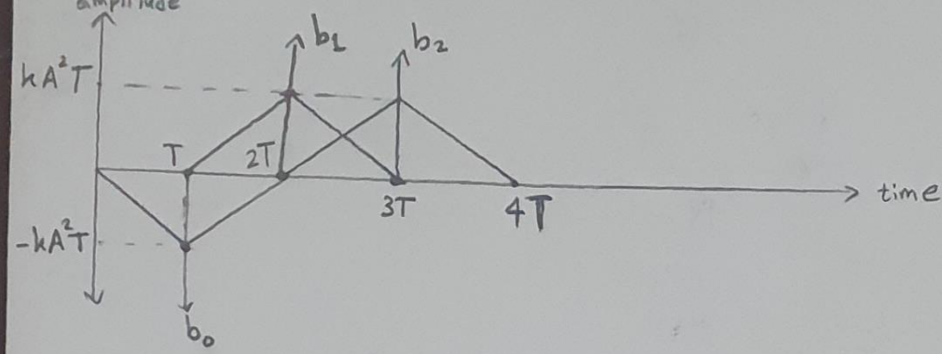


P.S: $\int_{-\infty}^{\infty} s(\tau) . h_{opt}(t-\tau)\, d\tau = \int_{0}^{T} (-A)(kA)\, d\tau = [-kA^2]_0^T = -kA^2T$

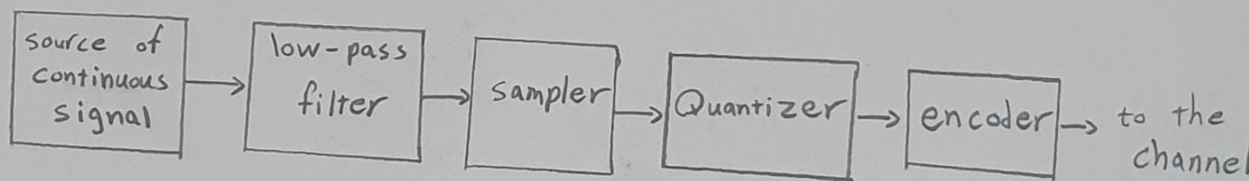
then the output signal of matched filter for bit sequence '011' is as follows



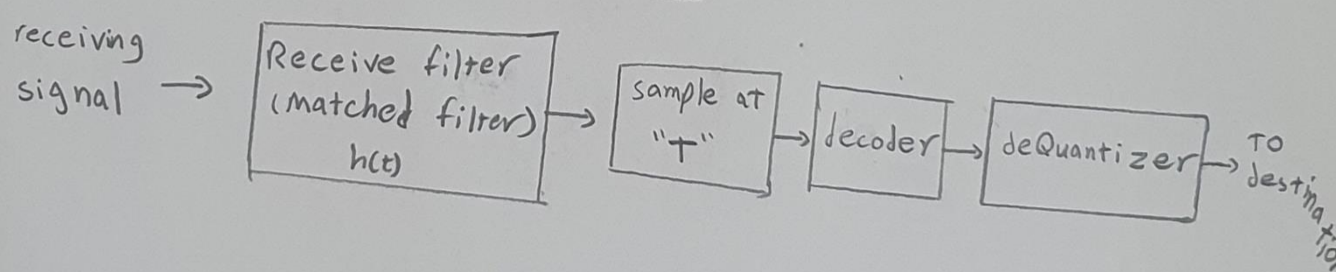
(c) the sampling instances will be at mk, where for b_0 it's at T, for b_1 it's at 2T, for b_2 it's at 3T



(d) block diagram for the transmitter:



(e) block diagram of the receiver:



Part2 Solution

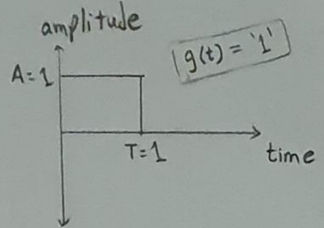
Q) solution is as follows:

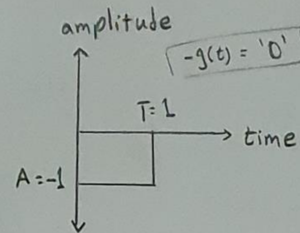
part 2

1

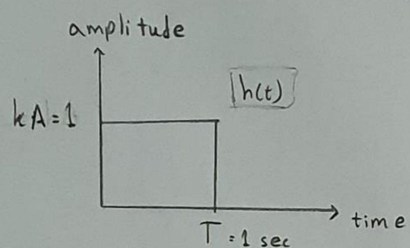
→ noise is AWGN with mean = 0, $\sigma^2 = \frac{N_0}{2}$ → let noise be $w(t)$

→ channel is ideal





ⓐ case h(t) is a unit energy matched filter



$\because kAT = 1$

and from $g(t) \rightarrow T = 1$ and $A = 1$

then $k \times 1 \times 1 = 1$

$\therefore k = 1$

then assume $r(t)$ is the input signal to matched filter and $w(t)$ is the noise added to original signal

$$\therefore r(t) = \begin{cases} A + w(t) & \text{for '1'}, \quad 0 \le t \le 1 \\ -A + w(t) & \text{for '0'}, \quad 0 \le t \le 1 \end{cases}$$

assume $y(t)$ is the output of the matched filter

$$\therefore y(T) = \int_{-\infty}^{\infty} r(t)\, h(t)\, dt = \int_{0}^{T \Rightarrow 1} \underset{1}{k}\underset{1}{A}\, r(t)\, dt = \int_0^1 r(t)\, dt$$

$$\therefore y(T) = \pm A + n(t) \quad \text{where } n(t) = \int_0^1 w(t)\, dt$$

from gaussian equation : $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-u)^2}{2\sigma^2}}$

for $y = \begin{cases} A+n(t) \text{ then } E[y] = E[A+n(t)] = A + E[\cancel{n(t)}]_{0} = A \\ -A+n(t) \text{ then } E[y] = E[-A+n(t)] = -A + E[\cancel{n(t)}]_{0} = -A \end{cases}$

note that $E[n(t)] = 0$ as given in the question that the mean of noise is 0

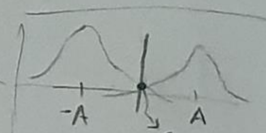
and $VAR[n(t)] = \frac{N_0}{2}$ as given in the question

for varience : $var(x) = E[x^2] - (E[x])^2$

for $y = \begin{cases} A+n(t),\ \sigma^2 = VAR[A+n(t)] = VAR[n(t)] = \frac{N_0}{2} \\ -A+n(t),\ \sigma^2 = VAR[-A+n(t)] = VAR[n(t)] = \frac{N_0}{2} \end{cases}$

$$\therefore p(y|'1') = \frac{1 \times e^{-\frac{(y-A)^2}{2\times\frac{N_0}{2}}}}{\sqrt{2\pi}\times\sqrt{\frac{N_0}{2}}} = \frac{e^{-\frac{(y-A)^2}{N_0}}}{\sqrt{N_0\pi}}$$

$$p(y|'0') = \frac{1 \times e^{-\frac{(y+A)^2}{2\times\frac{N_0}{2}}}}{\sqrt{2\pi}\times\sqrt{\frac{N_0}{2}}} = \frac{e^{-\frac{(y+A)^2}{N_0}}}{\sqrt{N_0\pi}}$$



$\lambda = \frac{A-A}{2} = 0$

$$\therefore p(e|'1') = \int_{-\infty}^{\lambda} p(y|'1')\,dy = \int_{-\infty}^{\lambda} \frac{e^{-\frac{(y-A)^2}{N_0}}}{\sqrt{N_0\pi}}\,dy = \frac{1}{\sqrt{N_0\pi}} \int_{-\infty}^{\lambda} e^{-\frac{(y-A)^2}{N_0}}\,dy$$

let $z = -\frac{(y-A)}{\sqrt{N_0}}$ then $dz = \frac{-1}{\sqrt{N_0}}dy \rightarrow dy = -\sqrt{N_0}\,dz$

when $y = \lambda$ then $z = \frac{-\lambda+A}{\sqrt{N_0}}$

when $y = -\infty$ then $z = \infty$

$$p(e|'1') = \frac{1}{\sqrt{\pi N_0}} \int_{\infty}^{\frac{-\lambda+A}{\sqrt{N_0}}} exp[-z^2] \times (-\sqrt{N_0})\,dz = \frac{-\sqrt{N_0}}{\sqrt{\pi N_0}} \int_{\infty}^{\frac{-\lambda+A}{\sqrt{N_0}}} exp[-z^2]\,dz$$

$$= \frac{1}{\sqrt{\pi}} \int_{\frac{-\lambda+A}{\sqrt{N_0}}}^{\infty} exp[-z^2]\,dz = \frac{1}{2}\, erfc\left(\frac{-\lambda+A}{\sqrt{N_0}}\right)$$

$\boxed{\lambda = 0}$

$$\boxed{\therefore p(e|'1') = \frac{1}{2}\, erfc\left(\frac{A}{\sqrt{N_0}}\right)}$$

$$p(e|'0') = \int_{\lambda}^{\infty} p(y|'0')\,dy = \int_{\lambda}^{\infty} \frac{e^{-\frac{(y+A)^2}{N_0}}}{\sqrt{N_0\pi}}\,dy = \frac{1}{\sqrt{N_0\pi}} \int_{\lambda}^{\infty} e^{-\frac{(y+A)^2}{N_0}}\,dy$$

let $z = \frac{y+A}{\sqrt{N_0}}$ then $dz = \frac{1}{\sqrt{N_0}}\,dy \rightarrow dy = \sqrt{N_0}\,dz$

when $y = \lambda$ then $z = \frac{\lambda+A}{\sqrt{N_0}}$

when $y = \infty$ then $z = \infty$

$$\therefore p(e|'0') = \frac{1}{\sqrt{N_0\pi}} \int_{\frac{\lambda+A}{\sqrt{N_0}}}^{\infty} \exp[-z^2]\sqrt{N_0}\,dz = \frac{\sqrt{N_0}}{\sqrt{N_0\pi}} \int_{\frac{\lambda+A}{\sqrt{N_0}}}^{\infty} \exp[-z^2]\,dz$$

$$= \frac{1}{\sqrt{\pi}} \int_{\frac{\lambda+A}{\sqrt{N_0}}}^{\infty} \exp[-z^2]\,dz = \frac{1}{2}\,\mathrm{erfc}\left(\frac{\lambda+A}{\sqrt{N_0}}\right)$$

$\boxed{\lambda = 0}$

$$\boxed{\therefore p(e|'0') = \frac{1}{2}\,\mathrm{erfc}\left(\frac{A}{\sqrt{N_0}}\right)}$$

for the probability of error:

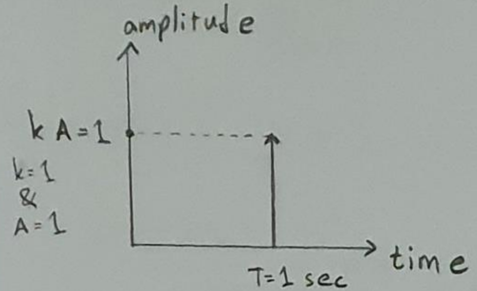
$$p(e) = p(e|'0')\underbrace{p('0')}_{\frac{1}{2}} + p(e|'1')\underbrace{p('1')}_{\frac{1}{2}}$$

$$\therefore p(e) = \frac{1}{2} \times \left[\frac{1}{2}\,\mathrm{erfc}\left(\frac{A}{\sqrt{N_0}}\right)\right] + \frac{1}{2} \times \left[\frac{1}{2}\,\mathrm{erfc}\left(\frac{A}{\sqrt{N_0}}\right)\right]$$

$$= \frac{1}{2}\,\mathrm{erfc}\left(\frac{A}{\sqrt{N_0}}\right) \qquad \boxed{A = 1}$$

$$\boxed{\therefore p(e) = \frac{1}{2}\,\mathrm{erfc}\left(\frac{1}{\sqrt{N_0}}\right)} \longrightarrow \text{Q.E.D}$$

ⓑ case $h(t)$ is not existant ($h(t) = \delta(t)$)



assume $r(t)$ is the input signal to matched filter and $w(t)$ is the noise added to original signal

$$\therefore r(t) = \begin{cases} A + w(t) & \text{for '1'}, \quad 0 \le t \le 1 \\ -A + w(t) & \text{for '0'}, \quad 0 \le t \le 1 \end{cases}$$

assume $y(t)$ is the output of the matched filter

$$\therefore y(T) = r(t) = \pm A + w(t)$$

from guassian equation : $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

for $y = \begin{cases} A + w(t) & \text{then } E[y] = E[A + w(t)] = A + \underbrace{E[w(t)]}_{0} = A \\ -A + w(t) & \text{then } E[y] = E[-A + w(t)] = -A + \underbrace{E[w(t)]}_{0} = -A \end{cases}$

notice that $E[w(t)] = 0$ as given in the question that the mean $= 0$

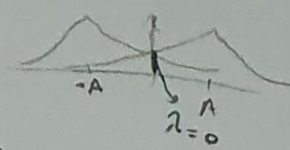
and $VAR[w(t)] = \frac{N_0}{2}$ as given in the question that variance $= \frac{N_0}{2}$

for varience : $Var(x) = E[x^2] - (E[x])^2$

for $y = \begin{cases} A + w(t), \ \sigma^2 = VAR[A + w(t)] = \frac{N_0}{2} \\ -A + w(t), \ \sigma^2 = VAR[-A + w(t)] = \frac{N_0}{2} \end{cases}$

$\because \; p(y|'1') = \dfrac{e^{\frac{-(y-A)^2}{N_0}}}{\sqrt{N_0 \pi}}$, $p(y|'0') = \dfrac{e^{\frac{-(y+A)^2}{N_0}}}{\sqrt{N_0 \pi}}$

$\because \; p('1') = p('0') = 0.5$ then $\lambda = 0$



$$p(e|'1') = \int_{-\infty}^{\lambda} p(y|'1')\,dy = \frac{1}{\sqrt{N_0 \pi}} \int_{-\infty}^{\lambda} e^{\frac{-(y-A)^2}{N_0}}\,dy$$

let $z = \dfrac{-(y-A)}{\sqrt{N_0}}$ then $dz = \dfrac{-1}{\sqrt{N_0}}\,dy \rightarrow dy = -\sqrt{N_0}\,dz$

when $y = \lambda$ then $z = \dfrac{-\lambda + A}{\sqrt{N_0}}$

when $y = -\infty$ then $z = \infty$

$$p(e|'1') = \frac{1}{\sqrt{\pi}} \int_{\frac{-\lambda+A}{\sqrt{N_0}}}^{\infty} \exp[-z^2]\,dz = \frac{1}{2}\,\mathrm{erfc}\left(\frac{-\lambda+A}{\sqrt{N_0}}\right)$$

given $\lambda = 0$

$$\boxed{\therefore \; p(e|'1') = \frac{1}{2}\,\mathrm{erfc}\left(\frac{A}{\sqrt{N_0}}\right)}$$

$$p(e|'0') = \int_{\lambda}^{\infty} p(y|'0')\,dy = \frac{1}{\sqrt{N_0 \pi}} \int_{\lambda}^{\infty} e^{-\frac{(y+A)^2}{N_0}}\,dy$$

let $z = \dfrac{y+A}{\sqrt{N_0}}$ then $dz = \dfrac{1}{\sqrt{N_0}}\,dy \rightarrow dy = \sqrt{N_0}\,dz$

when $y = \lambda$ then $z = \dfrac{\lambda + A}{\sqrt{N_0}}$

when $y = \infty$ then $z = \infty$

$$p(e|'0') = \frac{1}{\sqrt{\pi}} \int_{\frac{\lambda+A}{\sqrt{N_0}}}^{\infty} \exp[-z^2]\,dz = \frac{1}{2}\,\mathrm{erfc}\left(\frac{\lambda+A}{\sqrt{N_0}}\right)$$

given $\lambda = 0$

$$\boxed{p(e|'0') = \frac{1}{2}\,\mathrm{erfc}\left(\frac{A}{\sqrt{N_0}}\right)}$$

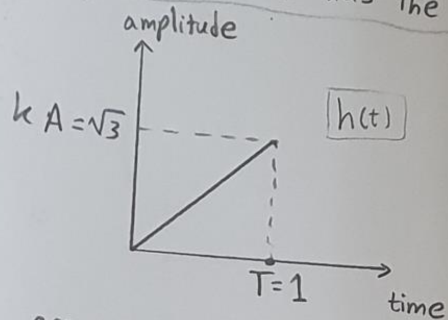
for the probability of error:

$$p(e) = p(e|'0')\,p('0') + p(e|'1')\,p('1') = \frac{1}{2}\,\text{erfc}\left(\frac{A}{\sqrt{N_0}}\right)$$

given $A = 1$

$$\boxed{\therefore P(e) = \frac{1}{2}\,\text{erfc}\left(\frac{1}{\sqrt{N_0}}\right)} \longrightarrow \text{Q.E.D}$$

ⓒ case $h(t)$ has the following impulse response:



$$\Rightarrow h(t) = \frac{\sqrt{3}}{1}t + 0 = \sqrt{3}\,t$$

assume $r(t)$ is the input signal to matched filter and $w(t)$ is the noise, since $r(t) = g(t) + w(t)$ then $r(t) * h(t) = \underbrace{g(t) * h(t)}_{\text{call it } g_0(t)} + \underbrace{w(t) * h(t)}_{\text{call it } n_0(t)}$

$$\therefore r(t) = \begin{cases} A + w(t) & \text{for '1'},\ 0 \le t \le 1 \\ -A + w(t) & \text{for '0'},\ 0 \le t \le 1 \end{cases}$$

$$\therefore g_0(t) = \begin{cases} \int_0^T A \times \sqrt{3}\,t\,dt = \sqrt{3}A\left[\frac{t^2}{2}\right]_0^T = \frac{\sqrt{3}AT^2}{2}, & A = 1 \\ \int_0^T -A \times \sqrt{3}\,t\,dt = -\sqrt{3}A\left[\frac{t^2}{2}\right]_0^T = -\frac{\sqrt{3}AT^2}{2}, & A = 1 \end{cases}$$

$$\therefore g_0(T=1) = \begin{cases} \frac{\sqrt{3}}{2} & \text{for '1'} \\ -\frac{\sqrt{3}}{2} & \text{for '0'} \end{cases}$$

$$n(t) = h(t) * w(t) = \int_0^T w(t).h(T-t)\,dt = \int_0^T w(t)\,g(t)\,dt = \int_0^T A\,w(t)\,dt$$

$$= \int_0^T w(t)\,dt$$

let's find mean and varience of n(t):

$$E[n(t)] = \int_0^T E[\cancel{w(t)}]\,dt = 0$$

$$VAR[n(t)] = E[n^2(T)] - (E[\cancel{n(t)}])^2 = E[n^2(T)]$$

$$= E[n(T).n(T)] = \int_{-\infty}^{\infty} S_w(f)\,|H(f)|^2\,df$$

$$= \frac{N_0}{2}\underbrace{\int_{-\infty}^{\infty} |h(t)|^2\,dt}_{\text{area of } \left[\left(\frac{\sqrt{3}t}{1}\right)^2\right]} = \frac{N_0}{2} \times \int_{-\infty}^{\infty} (\sqrt{3}t)^2\,dt = \frac{N_0}{2}[t^3]_0^1 = \frac{N_0}{2}$$

$$VAR[n(t)] = \frac{N_0}{2}$$

<u>thus</u>

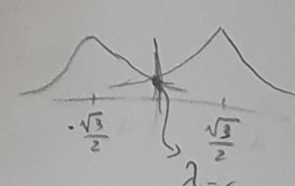
given gaussian distribution function: $f(x) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{\frac{-(x-u)^2}{2\sigma^2}}$

$$\therefore y = \begin{cases} \frac{\sqrt{3}}{2} + n(t) \\ \frac{-\sqrt{3}}{2} + n(t) \end{cases}$$

$$p(y|'1') = \frac{1}{\sqrt{2\pi} \times \sqrt{\frac{N_0}{2}}} \times e^{\frac{-(x-\frac{\sqrt{3}}{2})^2}{2\times\frac{N_0}{2}}} = \frac{e^{\frac{-(x-\frac{\sqrt{3}}{2})^2}{N_0}}}{\sqrt{\pi N_0}}$$

$$p(y|'0') = \frac{1}{\sqrt{2\pi} \times \sqrt{\frac{N_0}{2}}} \times e^{\frac{-(x+\frac{\sqrt{3}}{2})^2}{2\times\frac{N_0}{2}}} = \frac{e^{\frac{-(x+\frac{\sqrt{3}}{2})^2}{N_0}}}{\sqrt{N_0\pi}}$$

$\because p('0') = p('1')$ then $\lambda = 0$ $\left(\frac{\frac{\sqrt{3}}{2} - \frac{\sqrt{3}}{2}}{2} = 0\right)$



$$p(e|'1') = \int_{-\infty}^{\lambda} p(y|'1')\,dy = \frac{1}{\sqrt{\pi N_0}} \int_{-\infty}^{\lambda} e^{\frac{-(y-\frac{\sqrt{3}}{2})^2}{N_0}}\,dy$$

let $z = \frac{-(y-\frac{\sqrt{3}}{2})}{\sqrt{N_0}}$ then $dz = \frac{-1}{\sqrt{N_0}}dy \rightarrow dy = -\sqrt{N_0}\,dz$

when $y = \lambda \rightarrow z = \frac{-\lambda+\sqrt{3}}{\sqrt{N_0}}$

when $y = -\infty \rightarrow z = \infty$

$$\therefore p(e|'1') = \frac{\sqrt{N_0}}{\sqrt{\pi N_0}} \int_{\frac{-\lambda+\frac{\sqrt{3}}{2}}{\sqrt{N_0}}}^{\infty} \exp[-z^2]\,dz = \frac{1}{2}\,\mathrm{erfc}\left(\frac{-\lambda+\frac{\sqrt{3}}{2}}{\sqrt{N_0}}\right)$$

$\boxed{\text{given } \lambda = 0}$

$$\boxed{\therefore p(e|'1') = \frac{1}{2}\,\mathrm{erfc}\left(\frac{\frac{\sqrt{3}}{2}}{\sqrt{N_0}}\right)}$$

from symmetry of graphs : $p(e|'1') = p(e|'0')$

$$\therefore p(e) = p(e|'1')\underbrace{p('1')}_{0.5} + p(e|'0')\underbrace{p('0')}_{0.5}$$

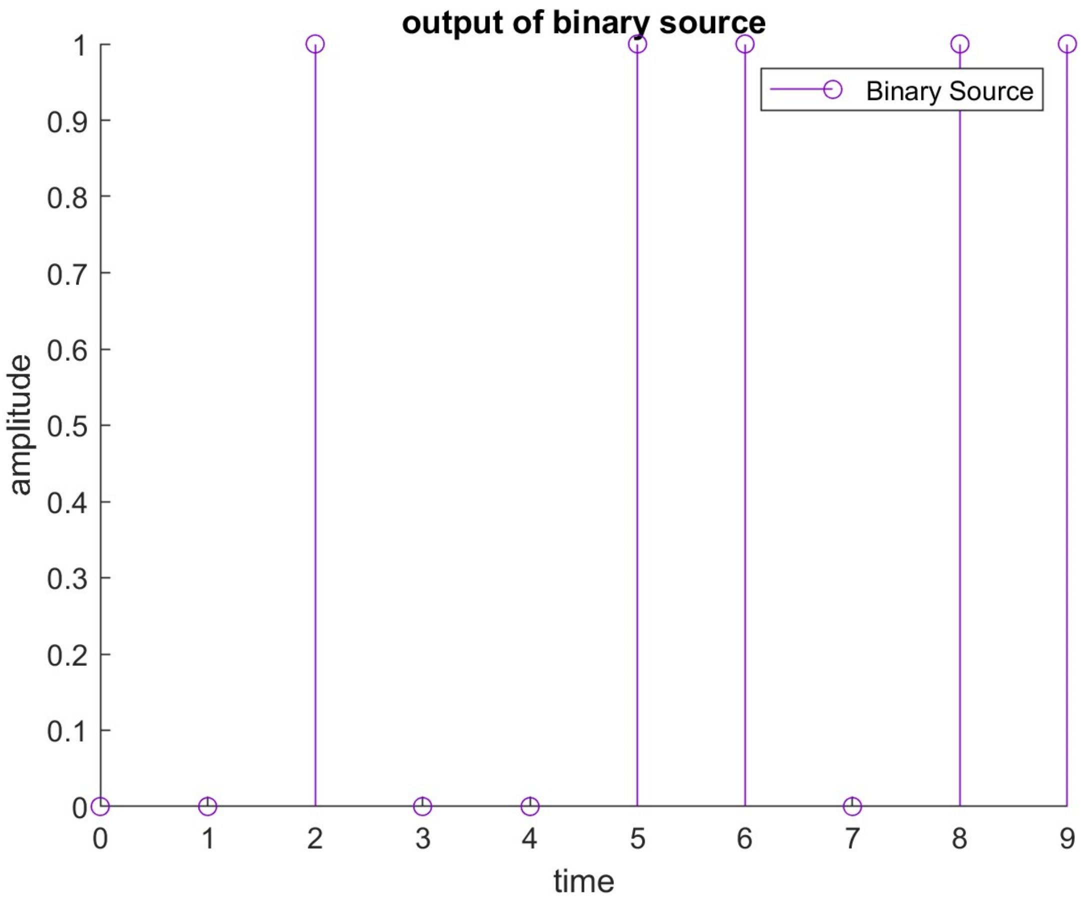
$$= p(e|'1')$$

$$p(e) = \frac{1}{2}\,\mathrm{erfc}\left(\frac{\frac{\sqrt{3}}{2}}{\sqrt{N_0}}\right)$$

Simulation of the system

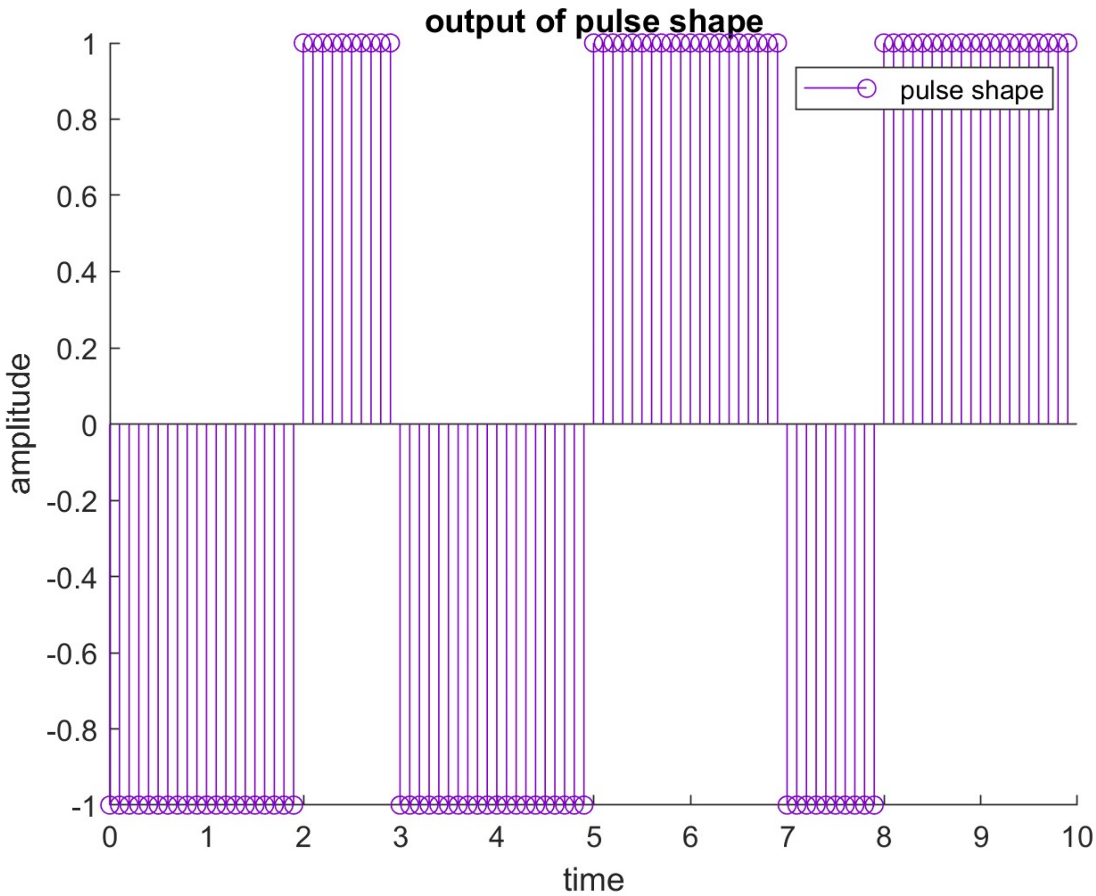
Suppose we have, a system where number of bits to be sent = 10 bits and for Noise $N_0 = 2$, then this is the output of each stage

output of the binary source



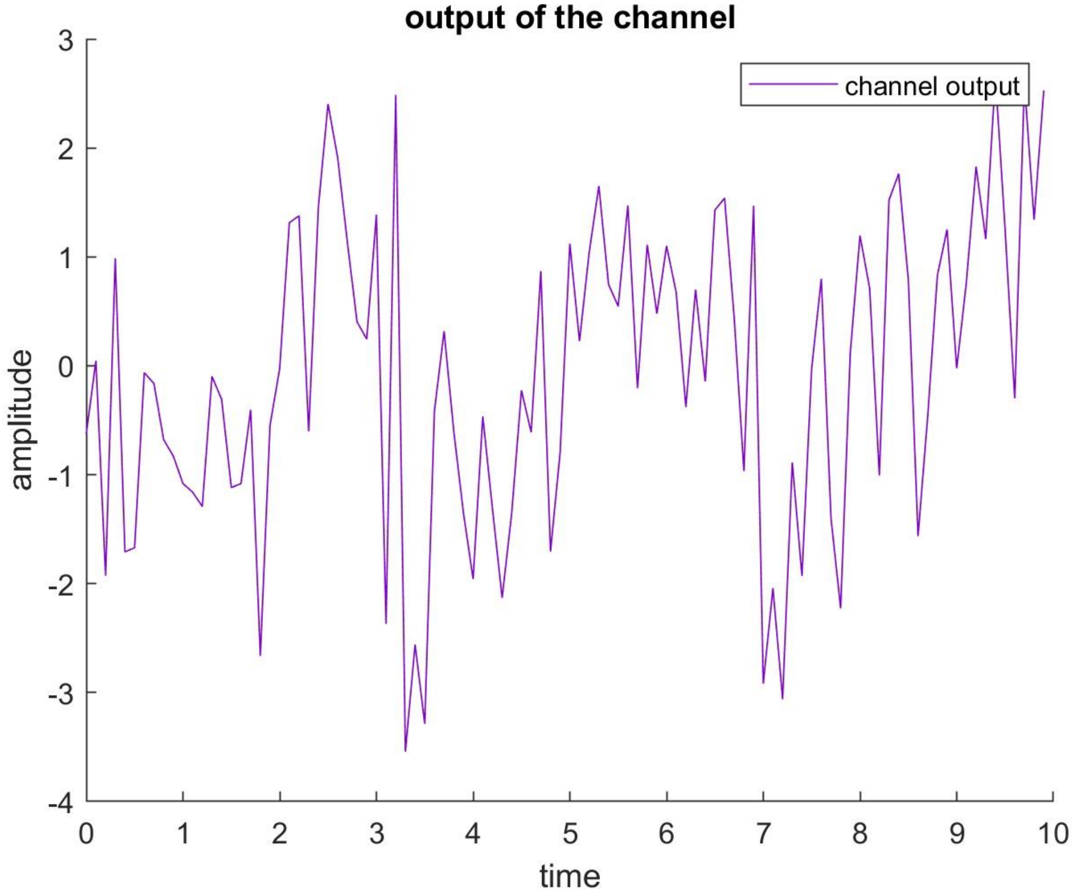
as you see the binary source generated bits to be sent in the following manner: 0010011011

output of the pulse shape



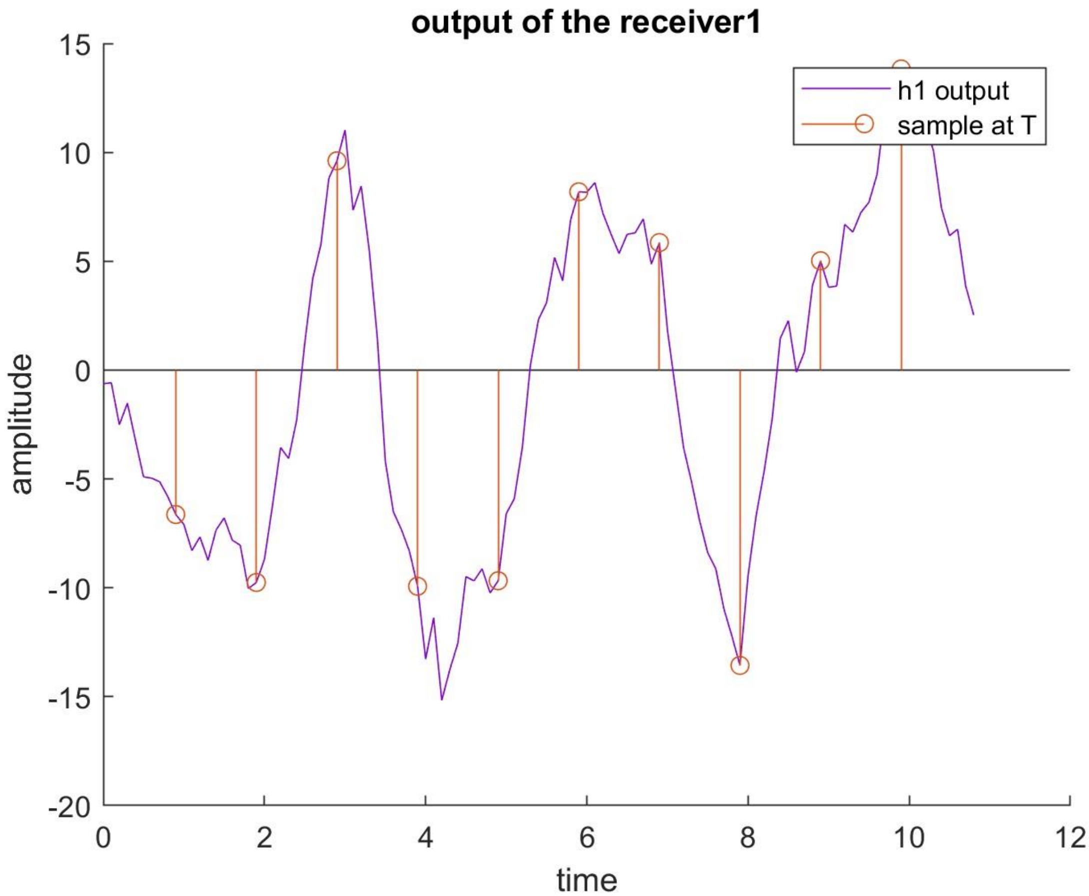
as you see, we repeat each bit 10 times and map 0 to -1 and 1 to 1 so the output of the pulse shape is 100 pulse where each pulse is either 1 or -1

output of the channel

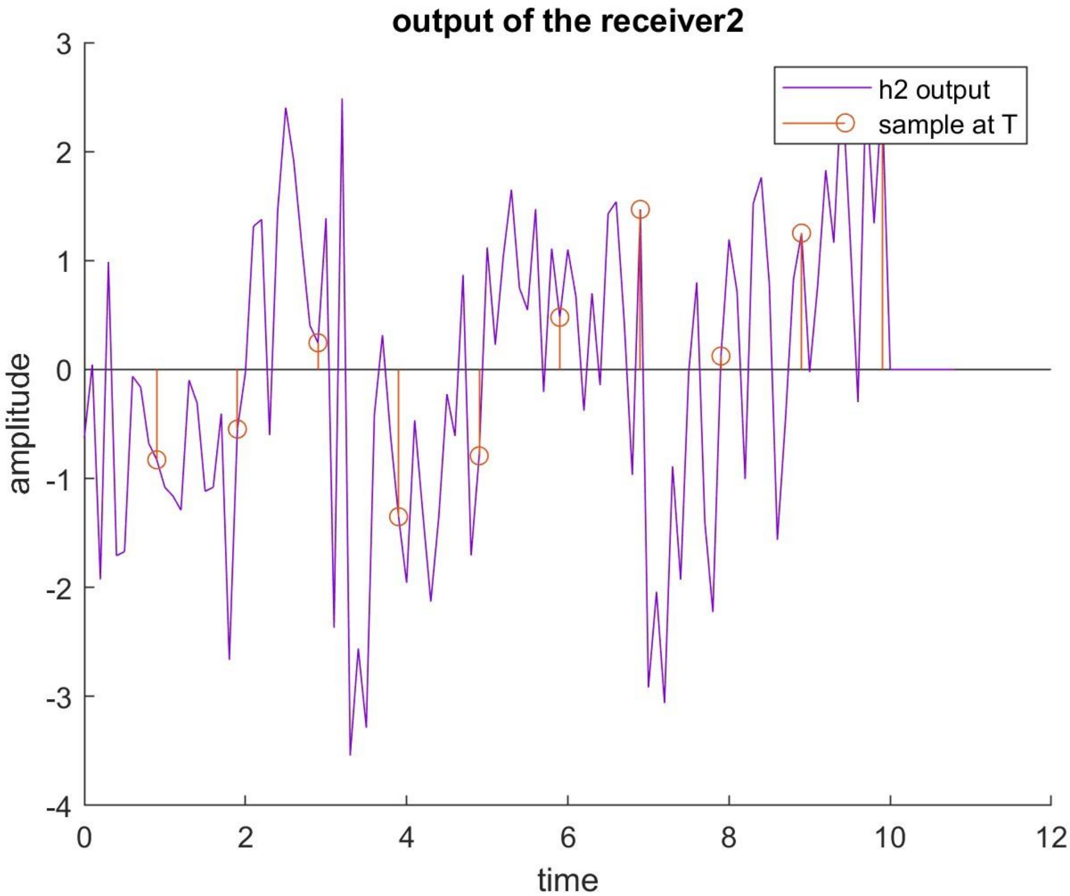


this is how our signal will look like in the channel after adding a noise with mean = 0 and variance = 0.5 (N_0 / 2) where the noise follows gaussian distribution.

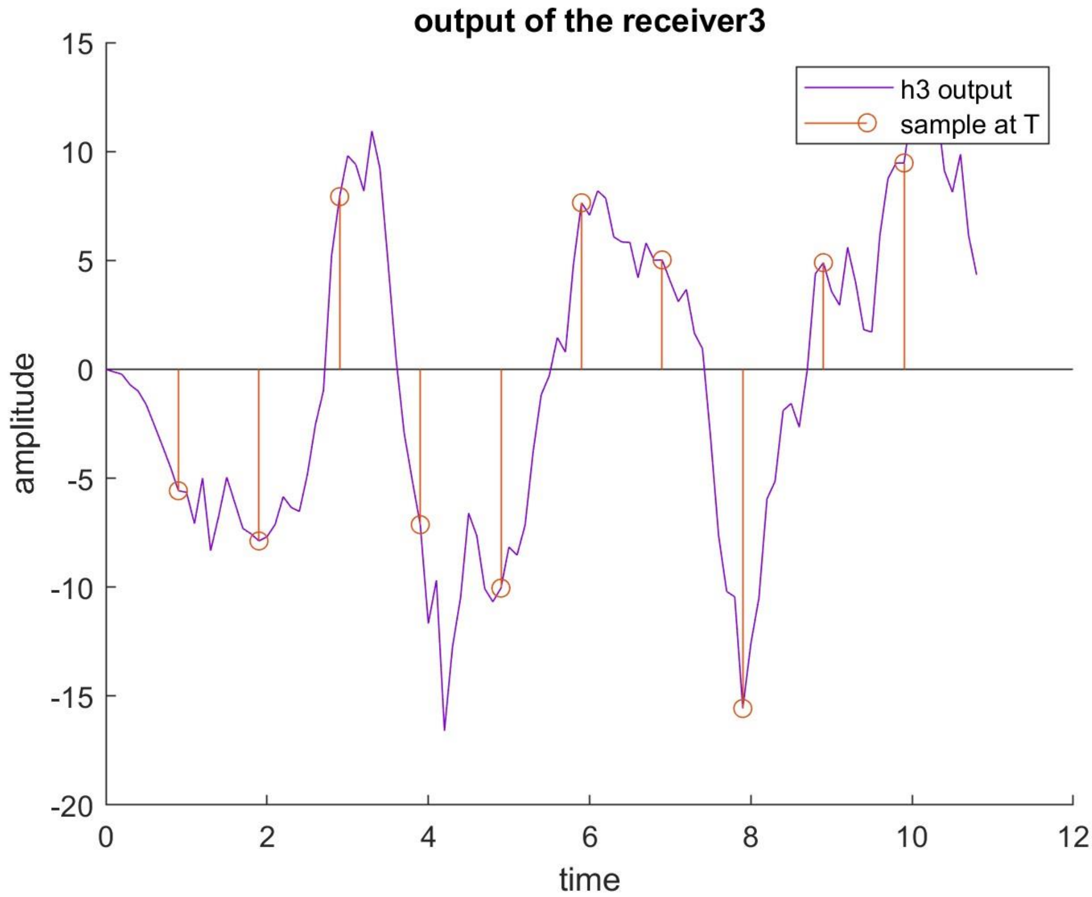
output of the receivers



this is the output signal from the matched filter with unit energy (1st case) along with time of sample with period = 1

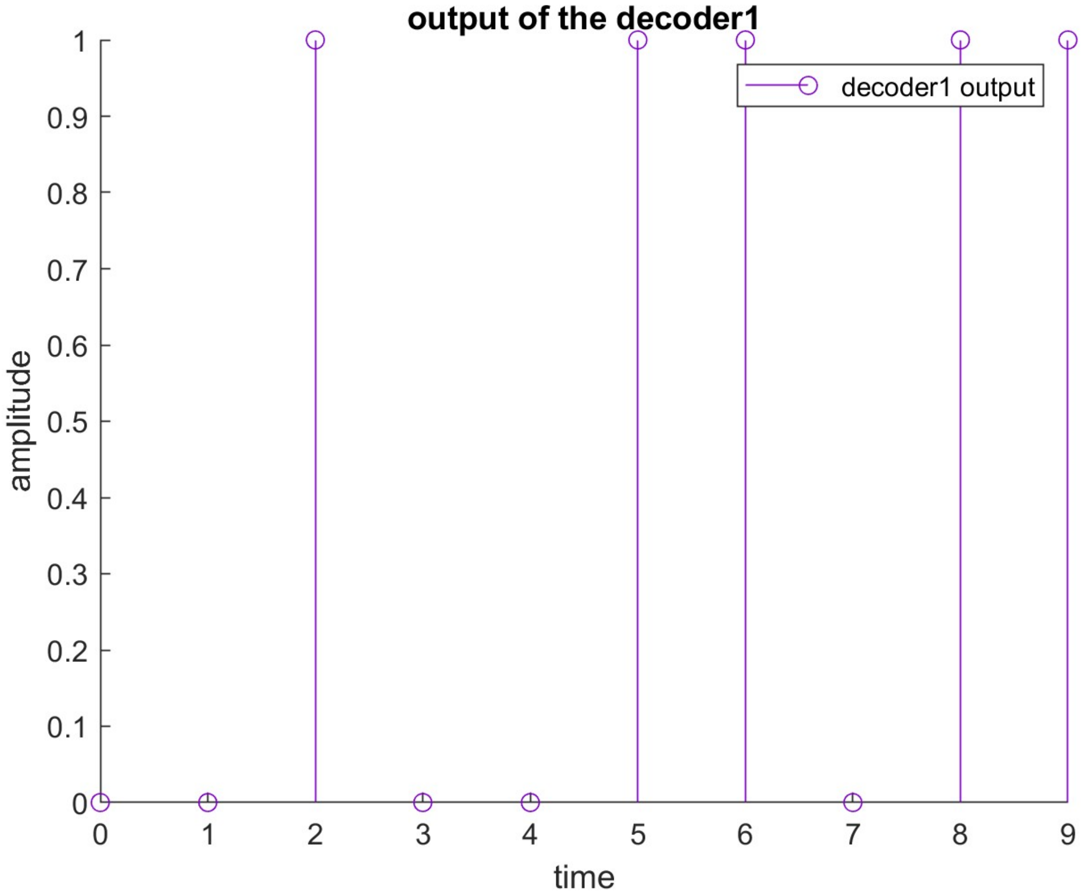


this is the output signal from the receiver when there is no matched filter (2nd case) along with time of sample with period = 1

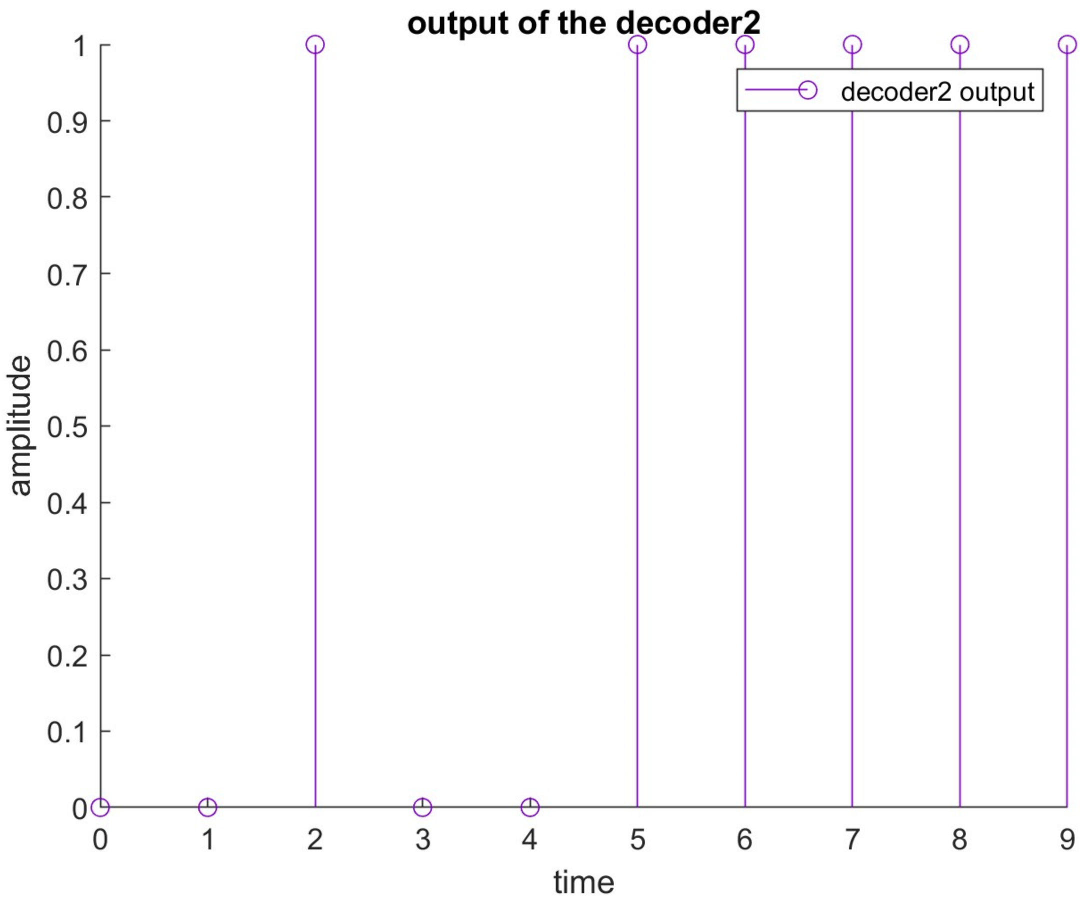


this is the output signal from the receiver when the matched filter has a triangle pulse response with equation $\sqrt{3}\ t$ (3rd case) along with time of sample with period = 1

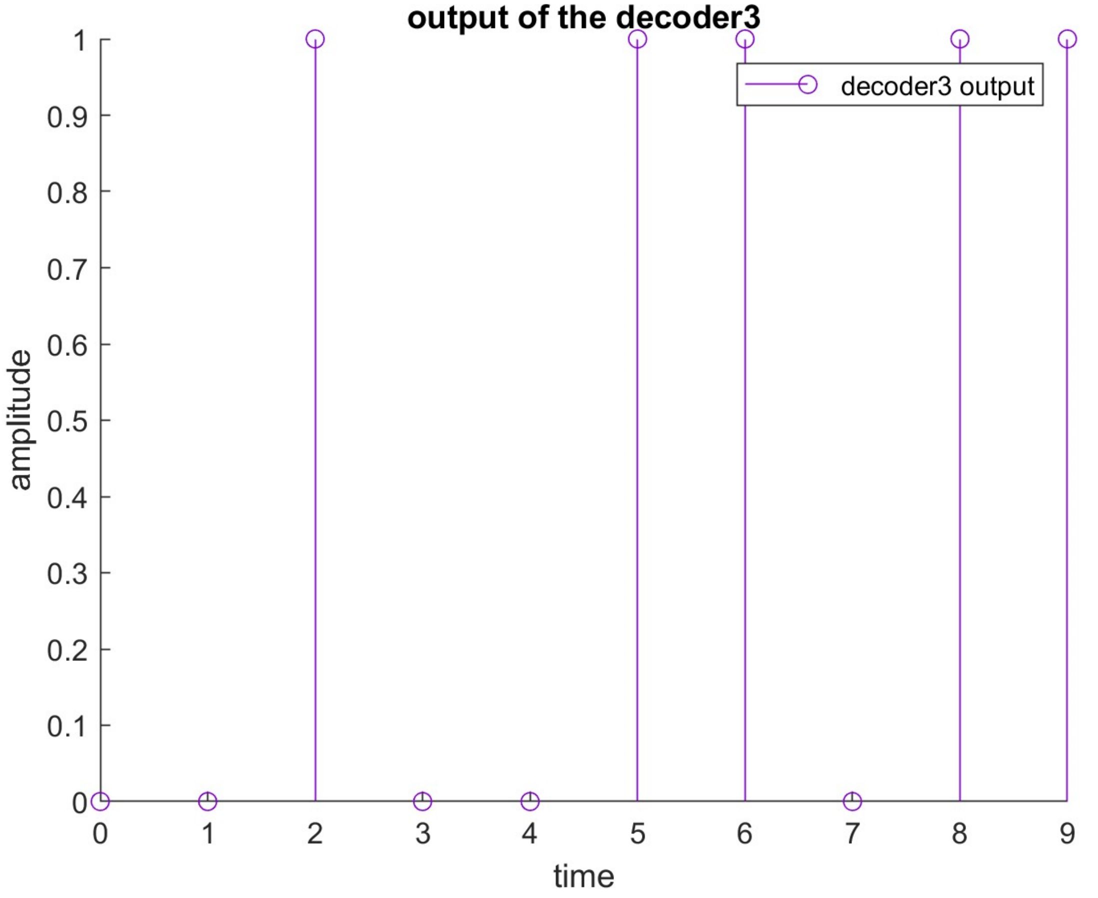
output of the decoders



this is the decoded signal when using a matched filter with unit energy (1st case) where the decoded signal is 0010011011 and the original signal was 0010011011 so no error occurred.

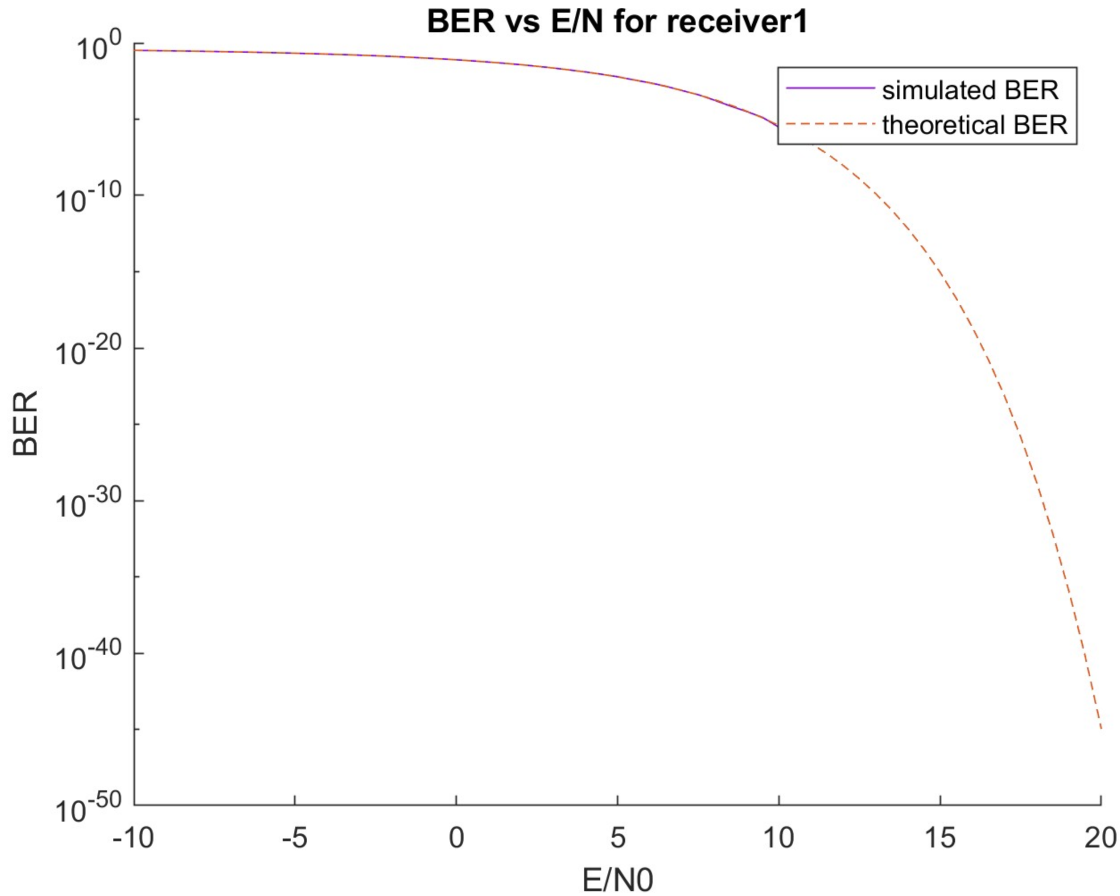


this is the decoded signal when no matched filter was used (2nd case) where the decoded signal is 0010011111 and the original signal was 0010011011 so error occurred at only 1 bit.

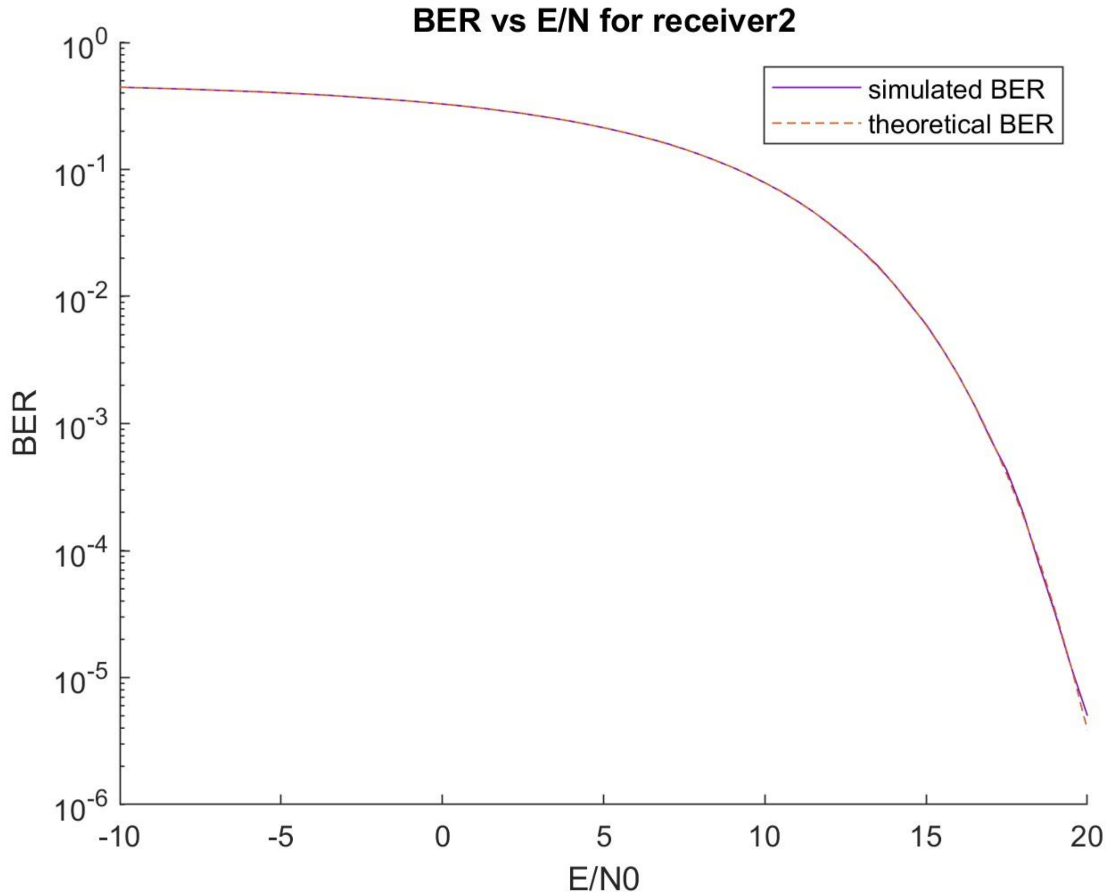


this is the decoded signal when a matched filter with impulse response equation $\sqrt{3}\ t$ was used (3^{rd} case) where the decoded signal is 0010011011 and the original signal was 0010011011 so no error occurred.

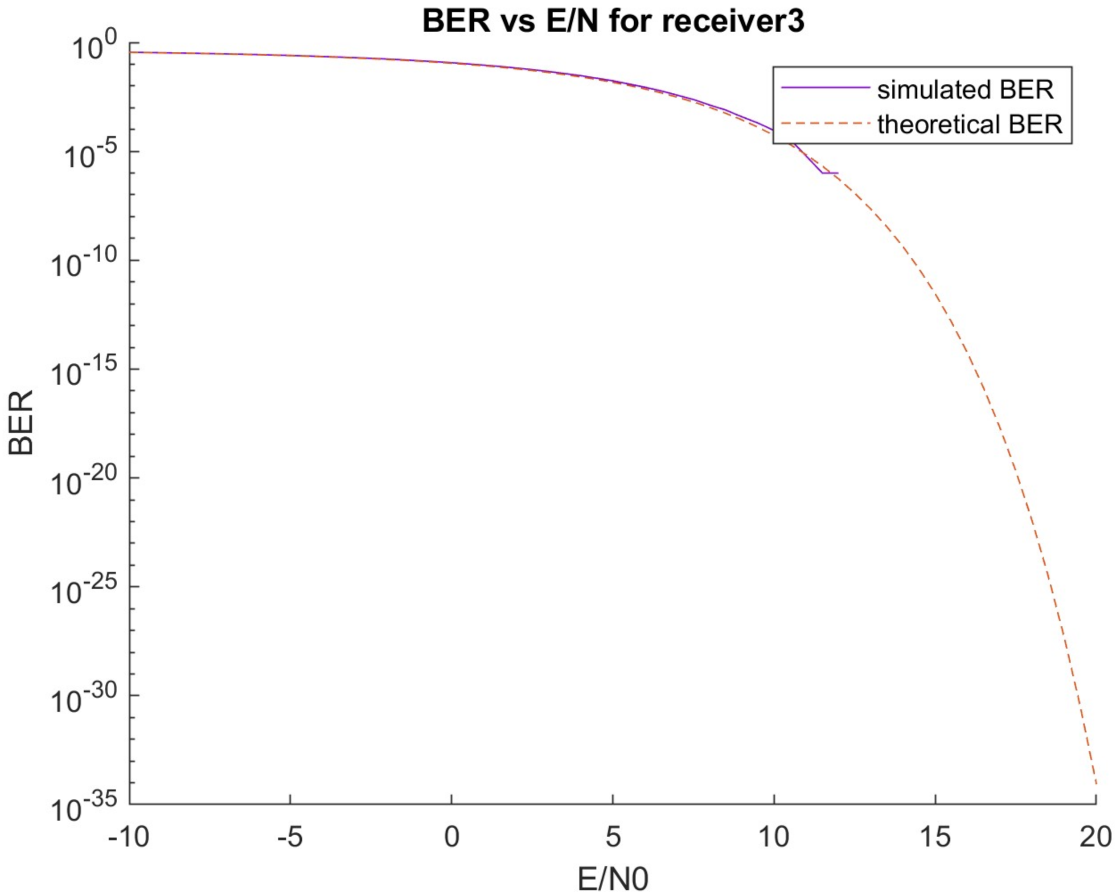
Simulated and theoretical BER vs E/N_0



In case of using a matched filter with unit energy (1st case), the simulated BER is becoming 0 after a specific E/N_0 where after that point the signal is received correctly with no problem at all, this is supposed to be the most optimal matched filter where it follows the equation Kg(T-t) and this matched filter impulse response has the same shape as the input signal so it's supposed to be the most optimal one (Note that in the simulated BER, it discontinues after a specific E/N_0 as BER=0 and so log(0) can't be plot as it tends to -infinity)



in case of not using any matched filter (2nd case), the simulated BER is nearly equal the theoretical BER as it appears on the graph and it's the worst BER as there is no matched filter (Note that in the simulated BER, it discontinues after a specific E/N_0 as BER=0 and so log(0) can't be plot as it tends to -infinity)



in case of using a matched filter with impulse response equation $\sqrt{3}\ t$ (3rd case) , the BER is better than not using matched but worse than the optimal filter used in 1st case,

Q5) the BER is a decreasing function in E/N_0 as E/N_0 represents energy of the original signal to that of the noise as the ratio increases it means that the energy of the original signal is more dominant and can be received successfully as the rate of error of receiving bits (BER) will decrease.

Q6) the case which have the lowest BER is the 1st case as the impulse response of the matched filter is the optimal filter following the equation KS(T-t) with K =1 as the impulse response shape of the matched filter is same shape as the input signal S(t) so it’s the most optimal filter with the lowest BER.

Total Code

```
% ------------------------------
% needed variables
% ------------------------------

% this variable is used to switch between plotting the output of each stage
% given N0 = 2 and plotting BER vs E/N0 for different values of N0
displayBlocksOutput = false;

% this variable is used to define the number of bits to be generated
numOfBits = 10000;

% energy of one symbol = 1
E = 1;

% generate different values for N0 where E/N0  ranges from -10db to 20db
E_over_N0 = -10:0.5:20;   % values in db
N0_arr = E_over_N0 ./ 10;

% select the values of N0 whether it's for plotting blocks output signals
% or plotting BER vs E/N0 for different values of N0
if displayBlocksOutput == true
    N0_arr = 2;
    numOfBits = 10;
else
    N0_arr = E ./ (10 .^ N0_arr);    % getting the actual value of N0
end


simulatedProbabilityOfError1 = zeros(1, length(E_over_N0));  % simulated pobability of error
case 1
simulatedProbabilityOfError2 = zeros(1, length(E_over_N0));  % simulated pobability of error
case 2
simulatedProbabilityOfError3 = zeros(1, length(E_over_N0));  % simulated pobability of error
case 3

% importat note: in the attached paper I proved that theoriticalProbabilityOfError2
% for receiver 2 is equal to 0.5*erfc(1/sqrt(N0)) but the problem in my
% prove is that I made the area of the impulse signal to be 1, so I
% made a width of the impulse to be 1/10 of the rectangle so the area
% of the impulse is 1/10, so after pluginging it in the equations, the
% theoriticalProbabilityOfError2 = 0.5*erfc(1/sqrt(N0*10))

% calculate the theoretical probability of error
theoriticalProbabilityOfError1(i) = 0.5*erfc(1/sqrt(N0));
theoriticalProbabilityOfError2(i) = 0.5*erfc(1/sqrt(N0*10));
theoriticalProbabilityOfError3(i) = 0.5*erfc(sqrt(3)/(2*sqrt(N0)));
```

```matlab
% ------------------------------
% BLOCK1 : Binary Source
% ------------------------------

% this is the array that holds our generated pulse
BinarySource = randi([0, 1], 1, numOfBits);


% ------------------------------
% BLOCK2 : pulse shape
% ------------------------------
pulseShapedSignal = pulseShape(BinarySource, 10);

% ------------------------------
% BLOCK3 : channel
% ------------------------------

for i = 1:length(N0_arr)

    % get the current N0
    N0 = N0_arr(i);

    % get the result signal after adding the noise
    R_t = channel(pulseShapedSignal, N0);

    % ------------------------------
    % BLOCK4 : Receive filter
    % ------------------------------

    % h(t) in case of unit energy filter (case 1)
    h1_t = ones(1, 10);
    convolutedSgn1 = receiveFilter(h1_t, R_t);

    % h(t) in case of matched filter doesn't exit (case 2)
    h2_t = [];
    convolutedSgn2 = receiveFilter(h2_t, R_t);


    % h(t) in case of triangular respose (case 3)
    T = 0:0.11:1;
    h3_t = sqrt(3) .* T;
    convolutedSgn3 = receiveFilter(h3_t, R_t);

    % ------------------------------
    % BLOCK5 : sample at T
    % ------------------------------

    % this is in case 1 for our matched filter
    sampledData1 = sampler(convolutedSgn1, 10);


    % this is in case 2 for our matched filter
    sampledData2 = sampler(convolutedSgn2, 10);
```

```matlab
    % this is in case 3 for our matched filter
    sampledData3 = sampler(convolutedSgn3, 10);



    % -----------------------------
    % BLOCK6 : decode to 0 and 1
    % -----------------------------


    % this is in case 1 for our matched filter
    decodedVals1 = decode(sampledData1);

    % this is in case 2 for our matched filter
    decodedVals2 = decode(sampledData2);

    % this is in case 3 for our matched filter
    decodedVals3 = decode(sampledData3);


    % -----------------------------
    % collecting data
    % -----------------------------

    % calculate the simulated probability of error
    simulatedProbabilityOfError1(i) = sum(decodedVals1 ~= BinarySource) / numOfBits;
    simulatedProbabilityOfError2(i) = sum(decodedVals2 ~= BinarySource) / numOfBits;
    simulatedProbabilityOfError3(i) = sum(decodedVals3 ~= BinarySource) / numOfBits;

    % calculate the theoretical probability of error
    theoriticalProbabilityOfError1(i) = 0.5*erfc(1/sqrt(N0));
    theoriticalProbabilityOfError2(i) = 0.5*erfc(1/sqrt(N0));
    theoriticalProbabilityOfError3(i) = 0.5*erfc(sqrt(3)/(2*sqrt(N0)));

    if displayBlocksOutput ==  true
        % plot the ouput from each stage

        % plotting the output of binary source
        T = 0:1:numOfBits-1;    % constructing the time signal
        figure
        title('output of binary source');
        hold on
        xlabel('time');
        ylabel('amplitude');
        h = stem(T, BinarySource, 'DisplayName', 'Binary Source');
        h.Color = [0.5 0 0.8];
        legend
        hold off

        % plotting the output of pulse shape
        T = 0:0.1:(numOfBits-0.1);    % constructing the time signal
        figure
        title('output of pulse shape');
        hold on
        xlabel('time');
```

```
ylabel('amplitude');
h = stem(T, pulseShapedSignal, 'DisplayName', 'pulse shape');
h.Color = [0.5 0 0.8];
legend
hold off

% plotting the output of channel
T = 0:0.1:(numOfBits-0.1);    % constructing the time signal
figure
title('output of the channel');
hold on
xlabel('time');
ylabel('amplitude');
h = plot(T, R_t, 'DisplayName', 'channel output');
h.Color = [0.5 0 0.8];
legend
hold off


% plotting the output of receive filter along with sampling
T = 0:0.1:(numOfBits+0.8);    % constructing the time signal
indexes = NaN(1, numOfBits*11 - 1);
for i = 10:10:(numOfBits*11 - 1)
    indexes(i) = 1;
end

% case 1
figure
title('output of the receiver1');
hold on
xlabel('time');
ylabel('amplitude');
h = plot(T, convolutedSgn1, 'DisplayName', 'h1 output');
h.Color = [0.5 0 0.8];
% plotting sampling time
sampleFunc = indexes .* convolutedSgn1;
stem(T, sampleFunc, 'DisplayName', 'sample at T' );
legend
hold off

% case 2
figure
title('output of the receiver2');
hold on
xlabel('time');
ylabel('amplitude');
h = plot(T, convolutedSgn2, 'DisplayName', 'h2 output');
h.Color = [0.5 0 0.8];
% plotting sampling time
sampleFunc = indexes .* convolutedSgn2;
stem(T, sampleFunc, 'DisplayName', 'sample at T' );
legend
hold off
```

```
        % case 3
        figure
        title('output of the receiver3');
        hold on
        xlabel('time');
        ylabel('amplitude');
        h = plot(T, convolutedSgn3, 'DisplayName', 'h3 output');
        h.Color = [0.5 0 0.8];
        % plotting sampling time
        sampleFunc = indexes .* convolutedSgn3;
        stem(T, sampleFunc, 'DisplayName', 'sample at T' );
        legend
        hold off

        % plotting the output of decoders
        T = 0:1:numOfBits-1;    % constructing the time signal

        % case 1
        figure
        title('output of the decoder1');
        hold on
        xlabel('time');
        ylabel('amplitude');
        h = stem(T, decodedVals1, 'DisplayName', 'decoder1 output');
        h.Color = [0.5 0 0.8];
        legend
        hold off

        % case 2
        figure
        title('output of the decoder2');
        hold on
        xlabel('time');
        ylabel('amplitude');
        h = stem(T, decodedVals2, 'DisplayName', 'decoder2 output');
        h.Color = [0.5 0 0.8];
        legend
        hold off

        % case 3
        figure
        title('output of the decoder3');
        hold on
        xlabel('time');
        ylabel('amplitude');
        h = stem(T, decodedVals3, 'DisplayName', 'decoder3 output');
        h.Color = [0.5 0 0.8];
        legend
        hold off

    end

end
```

```matlab
% ------------------------------
% plotting BER vs E/N0 for different values of N0
% ------------------------------
if displayBlocksOutput == false

    % plotting the theoretical and simulated Bit Error Rate (BER) Vs
    % E/No for receiver 1
    figure
    title('BER vs E/N for receiver1');
    hold on
    xlabel('E/N0');
    ylabel('BER');
    set(gca, 'YScale', 'log')
    h = plot(E_over_N0, simulatedProbabilityOfError1, 'DisplayName', 'simulated BER');
    h.Color = [0.5 0 0.8];
    plot(E_over_N0, theoriticalProbabilityOfError1, '--', 'DisplayName', 'theoretical BER');
    legend
    hold off

    % plotting the theoretical and simulated Bit Error Rate (BER) Vs
    % E/No for receiver 2
    figure
    title('BER vs E/N for receiver2');
    hold on
    xlabel('E/N0');
    ylabel('BER');
    set(gca, 'YScale', 'log')
    h = plot(E_over_N0, simulatedProbabilityOfError2, 'DisplayName', 'simulated BER');
    h.Color = [0.5 0 0.8];
    plot(E_over_N0, theoriticalProbabilityOfError2, '--', 'DisplayName', 'theoretical BER');
    legend
    hold off

    % plotting the theoretical and simulated Bit Error Rate (BER) Vs
    % E/No for receiver 3
    figure
    title('BER vs E/N for receiver3');
    hold on
    xlabel('E/N0');
    ylabel('BER');
    set(gca, 'YScale', 'log')
    h = plot(E_over_N0, simulatedProbabilityOfError3, 'DisplayName', 'simulated BER');
    h.Color = [0.5 0 0.8];
    plot(E_over_N0, theoriticalProbabilityOfError3, '--', 'DisplayName', 'theoretical BER');
    legend
    hold off


end
```

```
% this function is to convert pulses to (+ and -) rectangle pulses (pulse
% shape blocl)
function [G_t] = pulseShape(binaryBitsArr, numOfPulsesPerOnePulse)

    % needed varaibles for our for loop
    T = 0:1:numOfPulsesPerOnePulse-1;
    T = T / numOfPulsesPerOnePulse;
    V = [];

    % repeat each pulse 10 times to make a rectangle shape pulse
    for bit = binaryBitsArr
        % convert each signal to +1 and -1 insteal of 0 and 1
        if bit == 0
            V = cat(2, V, -rectpuls(T, numOfPulsesPerOnePulse));
        else
            V = cat(2, V, rectpuls(T, numOfPulsesPerOnePulse));
        end
    end

    % return the result
    G_t = V;
end


% this function is to add noise to the channel (N0)
function [R_t] = channel(originalSignal, N0)

    % generate the noise from gaussian distribution given mean = 0 and
    % varience = N0 / 2, refer to https://ch.mathworks.com/help/matlab/math/random-numbers-
with-specific-mean-and-variance.html
    noise = sqrt(N0 / 2) .* randn([1, length(originalSignal)]);

    % add the noise to the original signal (sqrt(10) is to normalized the length of the one
signal pulse)
    R_t = originalSignal/sqrt(10) + noise;

end

% this function is used to convolute the matched filter with the signal
% output from the channel
function [convolutedSgn] = receiveFilter(H_t, R_t)

    if(isempty(H_t))
        % the matched filter doesn't exist
        convolutedSgn = cat(2, R_t, zeros(1, 9));
    else
        % convulte the signal output from the channel with matched filter
        % response
        convolutedSgn = conv(H_t, R_t);
    end

end
```

```
% this function is used to sample at time T ( sample every N values)
function [sampledData] = sampler(data, N)

    % temporary vector
    tempVec = zeros(1, floor(length(data)/N));

    % sample every N times
    for i = 10:N:length(data)
        tempVec(i/10) = data(i);
    end

    % return the result
    sampledData = tempVec;

end



% this function is used to decode the results to either 0 or 1
function [decodedVals] = decode(data)
    % create temporary vector
    tempVec = data;

    % map the values to either 1 or 0
    tempVec(tempVec <= 0) = 0;
    tempVec(tempVec > 0) = 1;

    % return the result
    decodedVals = tempVec;
end
```