Communication Assignment 2

Name	Sec	BN	ID
Asmaa Adel Abdelhamed	1	14	9202285
Samaa Hazem Mohamed	1	32	9202660

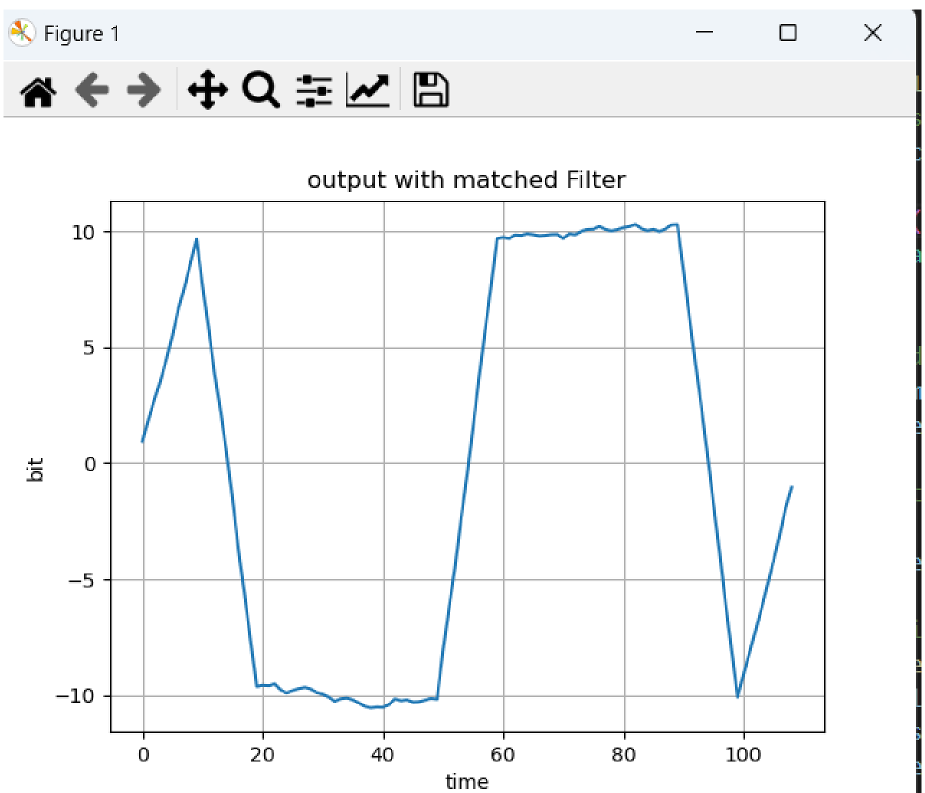
For theoretical solutions for question 1 and question 2, you will find it at the end of the report

Question 2:

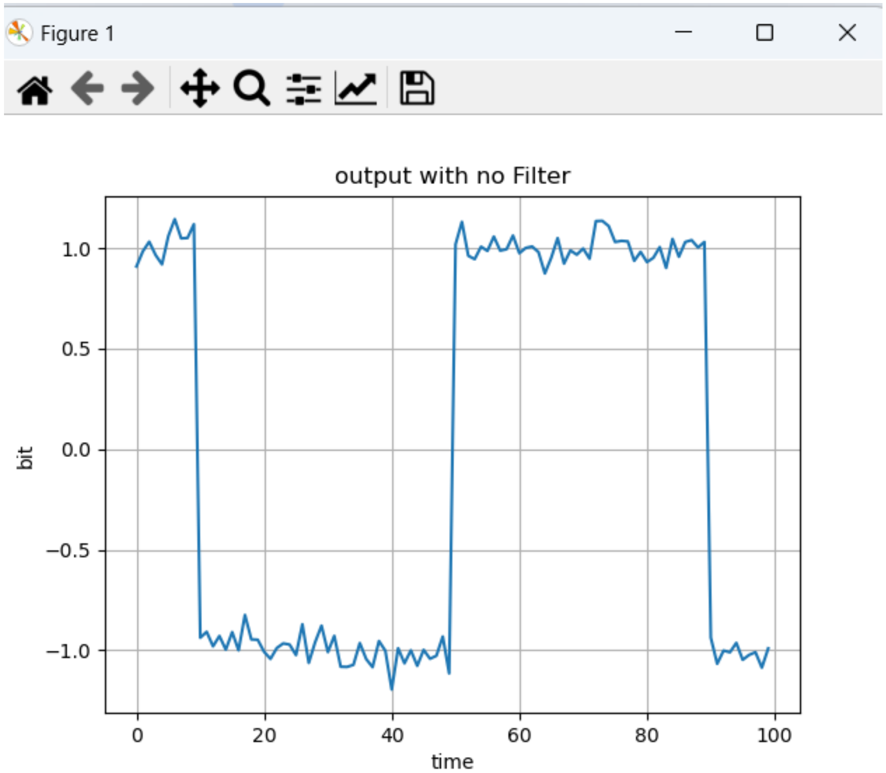
- **Plots for the outputs for each filter:**

  To make it visible, we made the number of bits = 10, and the number of samples =10

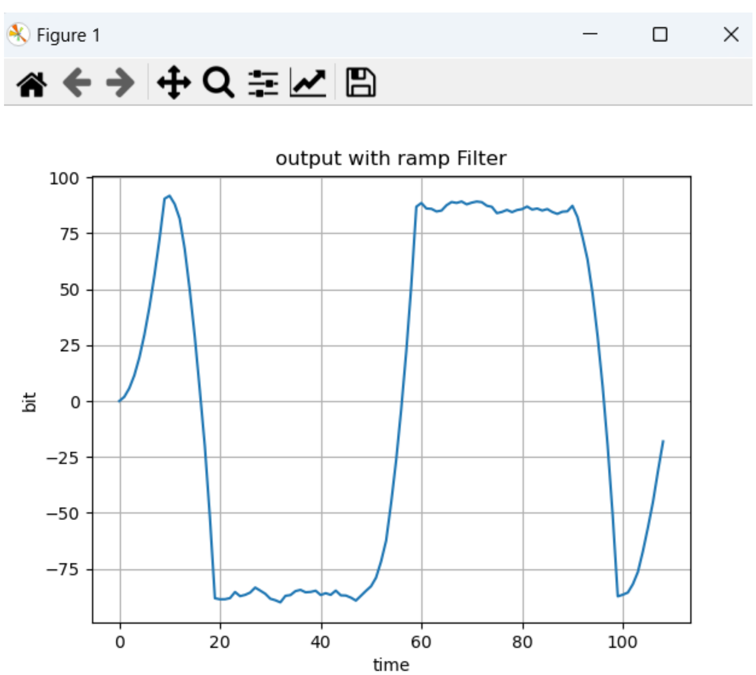
Filter 1 output:



Filter 2 output:

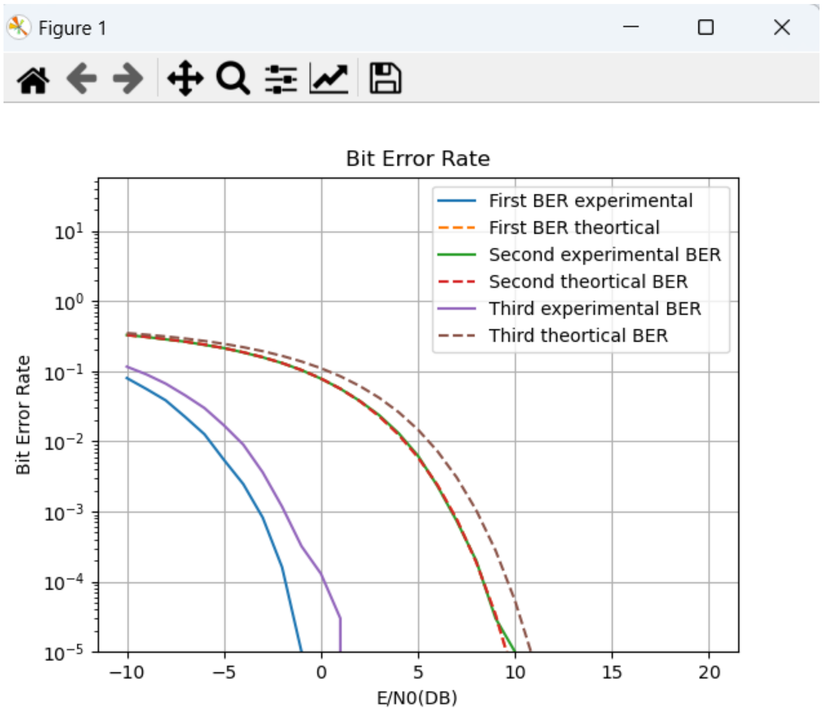


Filter 3 output:



- **The plot for theoretical and simulated Bit Error Rate (BER) Vs E/*N*o:**

  To make it visible, we made the number of bits = 10^5, and the number of samples =10



Comments for req 5 & req 6:

```
############################################################
############### Comments for 5 and 6 #######################
############################################################

####  5  ####
# given BER = 0.5*erfc(0.5*(A**2 * Tb/ 2*No)**0.5)
#           = 0.5*erfc(0.5*(E/No)**0.5)
# so if E/No increase , BER decrease, becauese of the nautral of erfc function

####  6  ####
# given BER(min) = 0.5*erfc(0.5*(SNR(max)/2)**0.5)
# First filter has the lowest BER as it has the max SNR
# as the SNR(max) is driven from Cauchy-Schwarz Inequality
# that if you want to get SNR(max) then the filter equation have to be
# h(t)opt = K g(T-t)
# which the first filter satisfied
```

The code:

```
############################################################
                    ############## Comments for 5 and 6
######################

############################################################

                    ####  5  ####
                    # given BER = 0.5*erfc(0.5*(A**2 * Tb/ 2*No)**0.5)
                    #           = 0.5*erfc(0.5*(E/No)**0.5)
                    # so if E/No increase , BER decrease, becauese of the
nautral of erfc function

                    ####  6  ####
                    # given BER(min) = 0.5*erfc(0.5*(SNR(max)/2)**0.5)
                    # First filter has the lowest BER as it has the max SNR
                    # as the SNR(max) is driven from Cauchy-Schwarz
Inequality
                    # that if you want to get SNR(max) then the filter
equation have to be
                    # h(t)opt = K g(T-t)
                    # which the first filter satisfied

import numpy as np
import matplotlib.pyplot as plt
import math

############################################################
################## Convolution Function ####################
############################################################
def applyConvolution(noisySamples, receivedFilterValues):

    convolutionResultSampledTp = np.zeros(numOfBits)

    if (receivedFilterValues is not None):
        convolutionResult = np.convolve(
            noisySamples.flatten(), receivedFilterValues)
    else:
```

```
        convolutionResult = noisySamples.flatten()
    for i in range(numOfBits):
        convolutionResultSampledTp[i] = convolutionResult[(numOfSamplesPerBit
- 1) + numOfSamplesPerBit * i]
    return convolutionResult, convolutionResultSampledTp



############################################################
######################## BER Simulated #####################
############################################################
def calculateBERSimulated(bits, recievedBits):
    receivedSamples = np.ones(numOfBits)
    receivedSamples = np.sign(recievedBits)
    # calculate probability of error
    error_probability = np.sum(receivedSamples != bits)
    error_probability /= numOfBits
    return error_probability


############################################################
######################## main function #####################
############################################################
def func(samples,receivedFilterMatched, type, plot):

    experimentalBER = []
    theorticalBER = []

    for EOverN0_db in range(-10, 21):
        EOverN0_dec = 10 ** (EOverN0_db / 10)

        # generating random noise
        generatedNoise = np.random.normal(
            # we want sigma of the noise t0 equal np.sqrt(No/2)
            # 0, E/(2*EOverN0_dec),
numOfBits*numOfSamplesPerBit).reshape((numOfBits, numOfSamplesPerBit))
            0, np.sqrt(1/(2*EOverN0_dec)),
numOfBits*numOfSamplesPerBit).reshape((numOfBits, numOfSamplesPerBit))
        # add noise to samples
        noisySamples = samples + generatedNoise
```

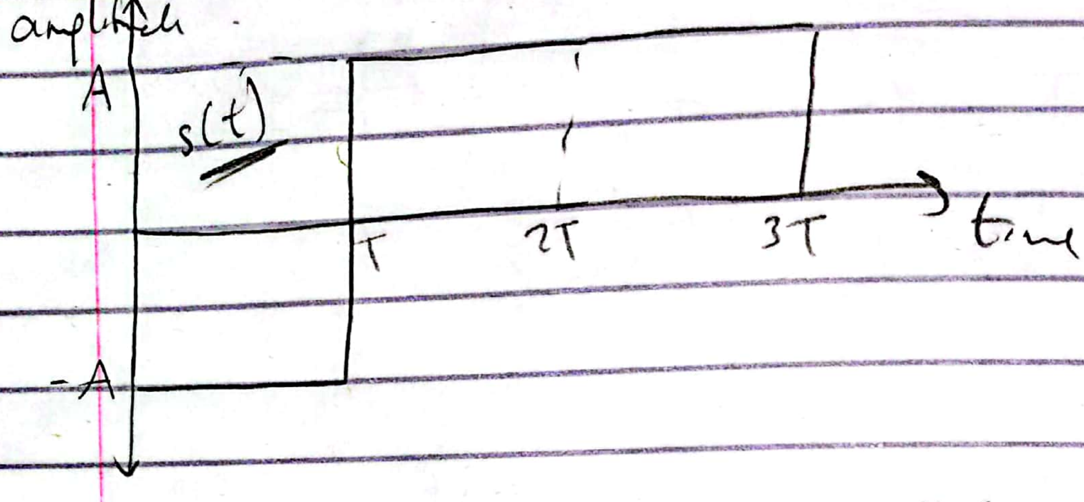
```
        # apply convolution for the noisy samples
        filteredSamples, recievedBits = applyConvolution(
            noisySamples, receivedFilterMatched)
        # append BER simulated for plotting
        experimentalBER.append(calculateBERSimulated(bits, recievedBits))
        # append BER theortical for plotting

        if type == 3:
            theorticalBER.append(0.5*math.erfc((3**0.5/2) * (EOverN0_dec **
0.5)))
        else:
          theorticalBER.append(0.5*math.erfc(EOverN0_dec ** 0.5))
    if plot == 1:

        # ploting
        plt.figure()
        plt.plot(range(0, filteredSamples.flatten(
        ).shape[0]), filteredSamples.flatten(), label="bit value")

        plt.xlabel('time')
        plt.ylabel('bit')
        if type == 1:
            plt.title('output with matched Filter')

        elif type == 2:
            plt.title('output with no Filter')

        else:
            plt.title('output with ramp Filter')

        plt.grid()
        plt.show()
    return experimentalBER, theorticalBER



############################################################
############## Generating bits and samples ###############
```

```
############################################################

#constants
numOfBits = 10
numOfSamplesPerBit = 10

# generate random bits with equal probability
# bits = np.asarray([(random.randint(0, 1)*2 - 1) for i in range(numOfBits)])
bits = np.random.choice([-1, 1], size=(numOfBits,), p=[1./2, 1./2])
# generate samples in range [numOfBits, numOfSamplesPerBit]
samples = (np.asarray([[bits[i] for i in range(numOfBits)]
           for _ in range(numOfSamplesPerBit)])).T


# receive with matched filter
filter_1 = np.ones(numOfSamplesPerBit)
experimentalBER_1, theorticalBER_1 = func(samples,filter_1, 1, 1)

# receive with no filter
filter_2 = None
experimentalBER_2, theorticalBER_2 = func(samples,filter_2, 2, 1)

# receive with ramp filter
filter_3 = np.linspace(0, 10, numOfSamplesPerBit)
for i in range(len(filter_3)):
    filter_3[i] = np.sqrt(3) * filter_3[i]
experimentalBER_3, theorticalBER_3 = func(samples,filter_3, 3, 1)


#############################################################################
###########################################


############################################################
############### Generating bits and samples ################
############################################################

# constants
numOfBits = 100000
numOfSamplesPerBit = 10
```

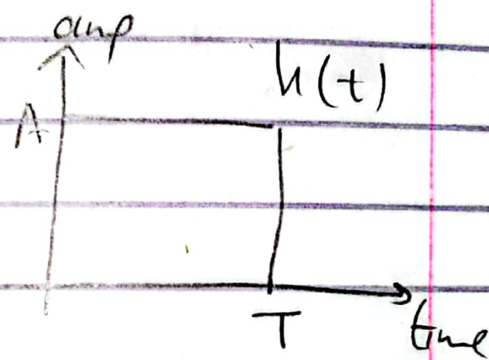
```
# generate random bits with equal probability
bits = np.random.choice([-1, 1], size=(numOfBits,), p=[1./2, 1./2])
# generate samples in range [numOfBits, numOfSamplesPerBit]
samples = (np.asarray([[bits[i] for i in range(numOfBits)]
           for _ in range(numOfSamplesPerBit)])).T


# receive with matched filter
filter_1 = np.ones(numOfSamplesPerBit)
experimentalBER_1, theorticalBER_1 = func(samples,filter_1, 1, 0)

# receive with no filter
filter_2 = None
experimentalBER_2, theorticalBER_2 = func(samples,filter_2, 2, 0)

# receive with ramp filter
filter_3 = np.linspace(0, 10, numOfSamplesPerBit)
for i in range(len(filter_3)):
    filter_3[i] = np.sqrt(3) * filter_3[i]
experimentalBER_3, theorticalBER_3 = func(samples,filter_3, 3, 0)


#ploting
plt.figure()
plt.plot(range(-10, 21), experimentalBER_1, label = "First BER experimental")
plt.plot(range(-10, 21), theorticalBER_1, "--", label = "First BER
theortical")

plt.plot(range(-10, 21), experimentalBER_2, label = "Second experimental
BER")
plt.plot(range(-10, 21), theorticalBER_2, "--", label = "Second theortical
BER")

plt.plot(range(-10, 21), experimentalBER_3, label = "Third experimental BER")
plt.plot(range(-10, 21), theorticalBER_3, "--", label = "Third theortical
BER")

plt.xlabel('E/N0(DB)')
plt.ylabel('Bit Error Rate')
plt.yscale('log')
```

```
plt.ylim(10**(-5))
plt.title('Bit Error Rate')

plt.legend()
plt.grid()
plt.show()
```

a) $s(t) = \begin{cases} -A & \text{for } (0) \\ A & \text{for } (1) \end{cases}$



b) $\because h(t) = k\, s(T-t)$



$\therefore$

$\because$ we will ignore the Noise

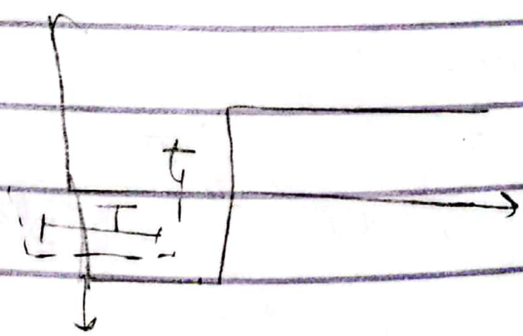
$\therefore y(t) = s(t) * h(t)$

$\cancel{\int s(t) h(t}$

we will have (4) possible regions :-

(1) $0 < t < T$

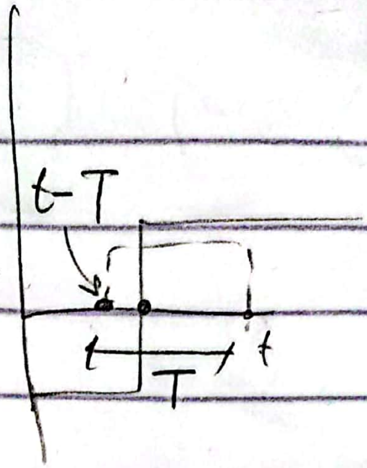
$$y(t) = \int_0^t -A \cdot A \, d\tau$$



$$= -A^2 \tau \Big|_0^t = \boxed{-A^2 t}$$

(2)

$$y(t)=\int_{t-T}^{t} = \int_{t-T}^{T} + \int_{T}^{t}$$



$$= \int_{t-T}^{T} -A^2\, d\tau + \int_{T}^{t} A^2\, d\tau$$

$$= (-A^2 \tau)\Big|_{t-T}^{T} + (A^2\tau)\Big|_{T}^{t}$$

$$= -A^2\left(T-(t-T)\right) + A^2(t-T)$$

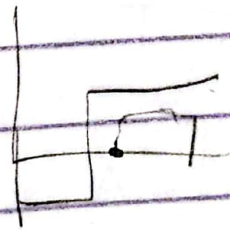
$$= -A^2(-t+2T) + A^2(t-T)$$

$$= +A^2 t - 2TA^2 - A^2T + A^2 t$$

$$\boxed{= +2A^2 t - 3A^2 T}$$

(3)

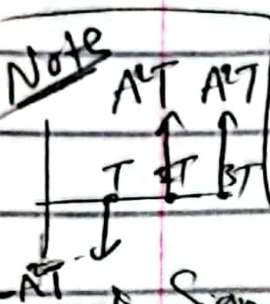
$$y(t)=\int_{t-T}^{t} A^2\, d\tau$$



$$= A^2 \tau\Big|_{t-T}^{t} = A^2(t-(t-T)) = \boxed{+A^2T}$$

④ $y(t) = \int_{t-T}^{3T} A^2 \, d\tau = A^2\tau \Big|_{t-T}^{3T}$

Note



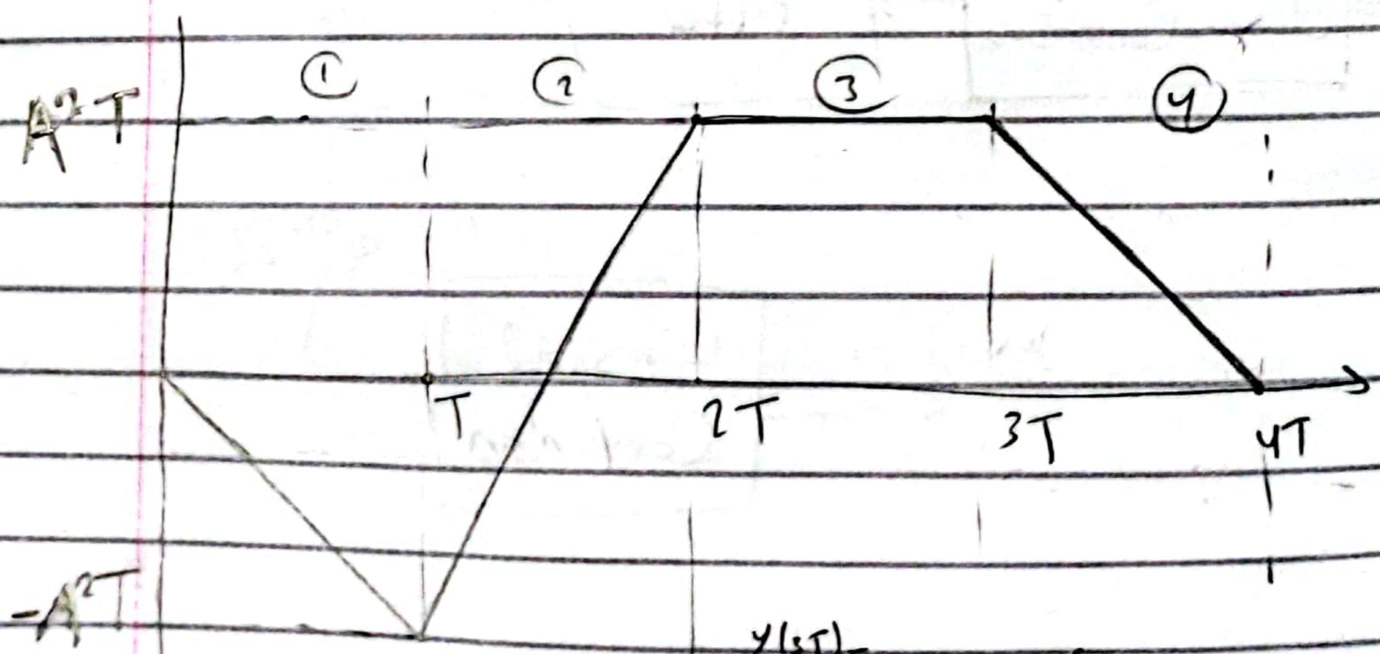
$= A^2(3T-(t-T)) = A^2(4T-t) = \boxed{-A^2t + 4A^2T}$

as → Signal will be Sampled at $T, 2T, 3T$ as following:-

b_0 $\boxed{0 \to y(T) = -A^2T}$, b_1 $\boxed{1 \to y(2T) = A^2T}$, b_2 $\boxed{1 \to y(3T) = A^2T}$

as $y(t)\big|_{total}$ equals:-

$$y(t) = \begin{cases} -A^2t & 0 < t < T \\ +2A^2t - 3A^2T & T < t < 2T \\ +A^2T & 2T < t < 3T \\ -A^2t + 4A^2T & t > 3T \end{cases}$$



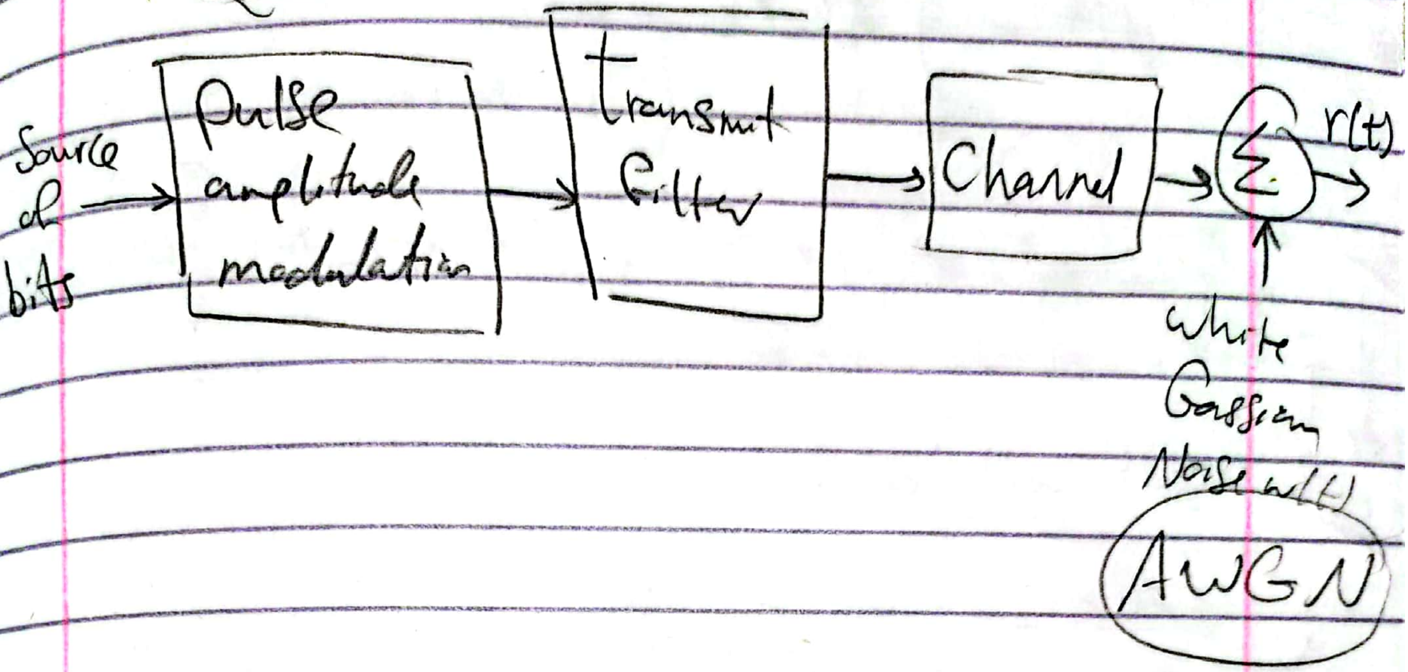
$y(T) = 2A^2T - 3A^2T = -A^2T$

$y(2T) = 2A^2T - 3A^2T$

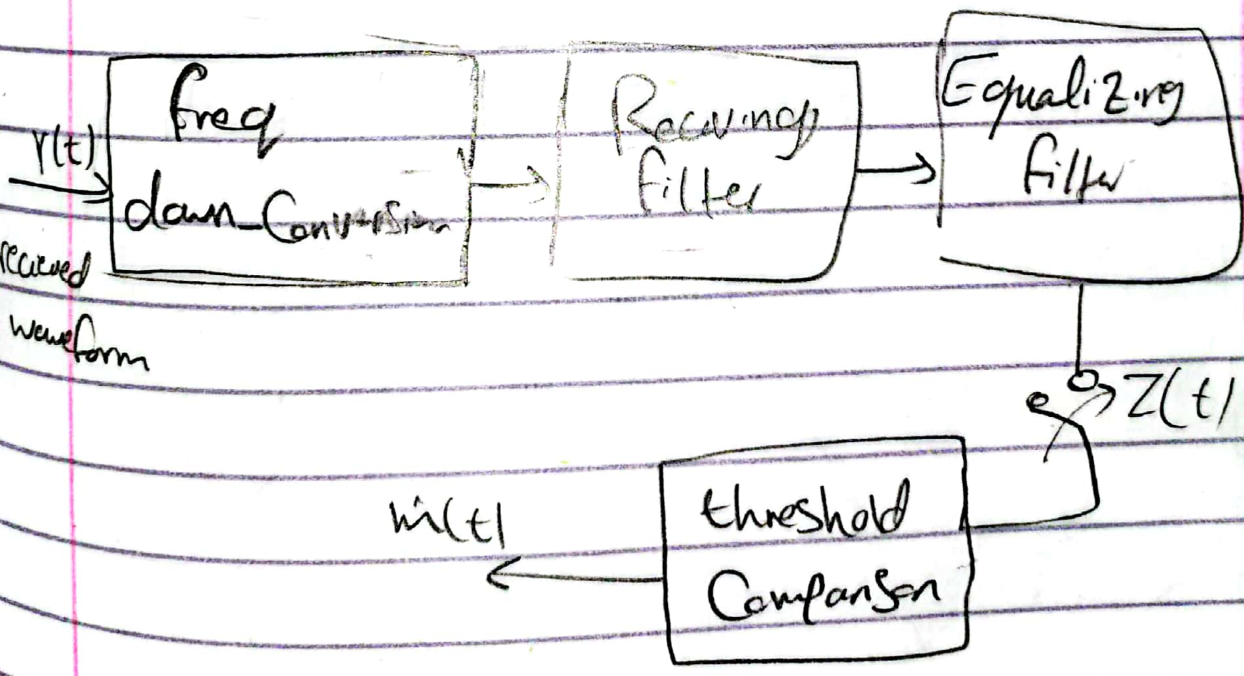
$y(3T) = A$

$y(4T) = -A^2(4T) + 4A^2T = 0$

~~receive~~ transimeter

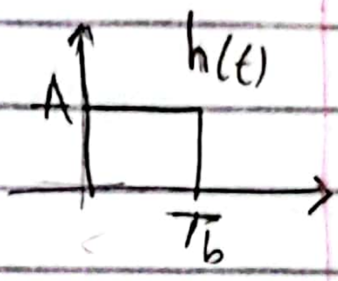


~~trans.~~ receiver



حل

① matched filter with unit energy



$y(t) = r(t) * h(t) = g(t) * h(t) + w(t) * h(t)$

as $r(t) = g(t) + w(t)$

given $g(t) = \begin{cases} A & \text{for '1'} \\ -A & \text{for '0'} \end{cases}$

$g_o(T_b) = g(t) * h(t) \rightarrow \int_0^{T_b} A^2 \, d\tau = A^2 \tau \Big|_0^{T_b} = A^2 T_b$ for '1'

$\rightarrow \int_0^{T_b} -A \cdot A \, d\tau = -A^2 \tau \Big|_0^{T_b} = -A^2 T_b$ for '0'

$y(T_b) = \begin{cases} A^2 T_b + n(T_b) \\ -A^2 T_b + n(T_b) \end{cases}$

$n(T_b) = w(t) * h(t)$

$\mu_y = E\{y(T_b)\} = E\{g_o(T_b)\} + E\{n(T_b)\}$

$E\{g_o(T_b)\} = \begin{cases} A^2 T_b & \text{for '1'} \\ -A^2 T_b & \text{for '0'} \end{cases}$

$$E\{n(T_b)\} = E\{\int w(\tau)\, h(T_b-\tau)\, d\tau\}$$
$$= \int_0^1 A_0 E(w(t))\, d\tau = \underline{0}$$

$$\boxed{\mu_y = \begin{cases} A^2T_b & \text{for } '1' \\ -A^2T_b & \text{for } '0' \end{cases}}$$

$$\sigma_y^2 = var\{y(T_b)\} = E\{(\cancel{g_o(T_b)} + n(T_b) - \cancel{\mu_y})^2\}$$
$$= E\{(n(T_b))^2\} = \frac{N_0}{2}\int_{-\infty}^{\infty}|H(f)|^2\, df$$
$$= \frac{N_0}{2}\int_0^1 |h(t)|^2\, dt = \frac{N_0}{2}(A^2)(T_b) = \frac{N_0}{2}A^2T_b$$

$$\boxed{\sigma_y^2 = \frac{N_0}{2}A^2T_b}$$

$$P(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(y-\mu_y)^2}{2\sigma^2}\right)$$

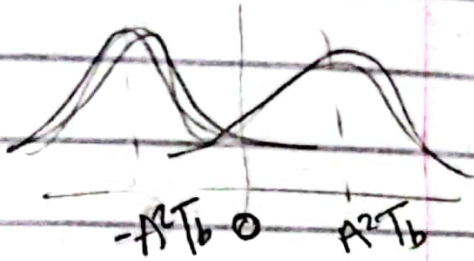
$$P(y|'0') = \frac{1}{\sqrt{2\pi \frac{N_0A^2T_b}{2}}} \exp\left(\frac{-(y-A^2T_b)^2}{2\frac{N_0A^2T_b}{2}}\right)$$

$$P(y|'1') = \frac{1}{\sqrt{2\pi \frac{N_0}{2}A^2T_b}} \exp\left(\frac{-(y-A^2T_b)^2}{2\frac{N_0A^2T_b}{2}}\right)$$

assume $p('0') = p('1') = 0.5$

$\therefore p(e) = p(e|'0')$

$= p(e|'1')$



$$p(e) = \int_0^\infty p(y|'0')\,dy$$

$$= \int_0^\infty \frac{1}{\sqrt{\pi N_0 A^2 T_b}} \exp\left(-\frac{(y+A^2T_b)^2}{N_0 A^2 T_b}\right) dy$$

let $z = \dfrac{y+A^2T_b}{\sqrt{N_0A^2T_b}}$

$$z\sqrt{N_0A^2T_b} = y + A^2T_b$$

$$z\sqrt{N_0A^2T_b} - A^2T_b = y$$

$$dz\sqrt{N_0A^2T_b} = dy$$

@ $y = 0 \rightarrow z = \dfrac{A^2T_b}{\sqrt{N_0A^2T_b}}$

$y = \infty \rightarrow z = \infty$

$$p(e) = \int_0^\infty \frac{1}{\sqrt{\pi N_0 A^2 T_b}} \exp\left(-\frac{(y+A^2T_b)^2}{N_0 A^2 T_b}\right)$$

$$= \int_{\frac{A^2T_b}{\sqrt{N_0A^2T_b}}}^\infty \frac{1}{\sqrt{\pi}\sqrt{N_0A^2T_b}} \exp(-z^2)\sqrt{N_0A^2T_b}\,dz$$

$$= \frac{1}{2}\,\mathrm{erfc}\left(\frac{A^2T}{\sqrt{N_0A^2T_b}}\right) \rightarrow \boxed{= \frac{1}{2}\,\mathrm{erfc}\left(\sqrt{\frac{A^2T_b}{N_0}}\right)}$$

m.S

2

$$g(t) = \begin{cases} A & \text{for '1'} \\ -A & \text{for '0'} \end{cases}, \quad h(t) = \delta(t)$$

$$y(t) = r(t) * h(t) = r(t) * \delta(t) = r(t)$$
$$= g(t) + w(t)$$

$$y(T_b) = \begin{cases} A + w(t) & \text{for '1'} \\ -A + w(t) & \text{for '0'} \end{cases}$$

m.S

$$\mu_y = \begin{cases} A & \text{for '1'} \\ -A & \text{for '0'} \end{cases} \quad \text{as } E(w(t)) = 0$$

$$\sigma_y^2 = Var(y(t)) = Var(w(t)) = \frac{N_0}{2}$$

$$P(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(y-\mu_y)^2}{2\sigma^2}\right)$$

$$p(y\,/\,'0') = \frac{1}{\sqrt{2\pi \frac{N_0}{2}}} \exp\left(\frac{-(y+A)^2}{2\frac{N_0}{2}}\right)$$

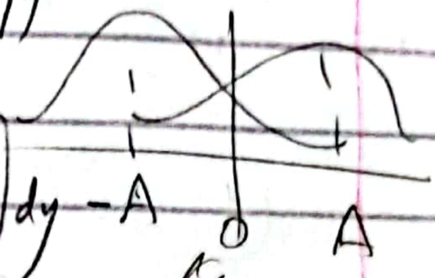
$$= \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{(y+A)^2}{N_0}\right)$$

$$p(y\,/\,'1') = \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{(y-A)^2}{N_0}\right)$$

assume $P('0') = P('1') = 0.5$

$\therefore P(e) = P(e\,|\,'0') = P(e\,|\,'1')$

$$= \int_0^\infty \frac{1}{\sqrt{\pi N_0}} \exp\left(\frac{-(y+A)^2}{N_0}\right) dy$$



let $z = \dfrac{y+A}{\sqrt{N_0}}$

$z\sqrt{N_0} - A = y$

$dz\sqrt{N_0} = dy$

@ $y = 0 \rightarrow z = \dfrac{A}{\sqrt{N_0}}$

@ $y = \infty \rightarrow z = \infty$

$$= \int_{\frac{A}{\sqrt{N_0}}}^\infty \frac{1}{\sqrt{\pi}\cancel{\sqrt{N_0}}} \exp(-z^2)\, \cancel{\sqrt{N_0}}\, dz$$

$$= \frac{1}{2} \operatorname{erfc}\left(\frac{A}{\sqrt{N_0}}\right) = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{A^2}{N_0}}\right)$$

here

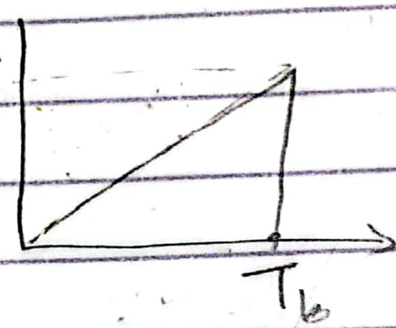
$$= \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{1}{N_0}}\right)$$

(3)

$$g(t) = \begin{cases} A & \text{for '1'} \\ -A & \text{for '0'} \end{cases}$$

$\frac{\sqrt{3}}{T_b}$

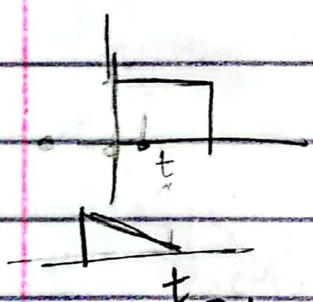
$h(t) \rightarrow \sqrt{3}$



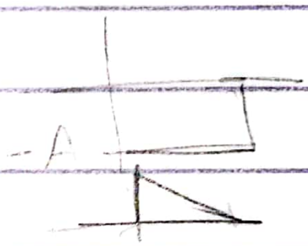
$y = \frac{\sqrt{3}}{T_b} t$

$$y(t) = r(t) * h(t) = \underbrace{g(t) * h(t)}_{g_o(t)} + \underbrace{w(t) * h(t)}_{n(t)}$$

for '1'



for '0'



$$g_o(t) = \int_0^t \frac{\sqrt{3}}{T_b} \tau A \, d\tau = \frac{\sqrt{3}}{2T_b} A \tau^2 \Big|_0^t \qquad g_o(t) = \int_0^{T_b} -\sqrt{3} A \, d\tau$$

$$= \frac{\sqrt{3}A}{2T_b} t^2 \qquad = -\sqrt{3} A T_b$$

$$y(T_b) = \begin{cases} \frac{\sqrt{3}AT_b^2}{2T_b} + n(T_b) \ \{\text{for '1'}\} & 0 < t < T_b \\ \frac{-\sqrt{3}AT_b^2}{2T_b} + n(T_b) \ \{\text{for '0'}\} & 0 < t < T_b \end{cases}$$

$$E(y(T_b)) = \mu_y = \begin{cases} \frac{\sqrt{3}}{2} A T_b \\ -\frac{\sqrt{3}}{2} A T_b \end{cases} \quad \text{as } \underline{E(n(T_b)) = 0}$$

$$\sigma_y = E\left((n(T_b))^2\right) =$$

$$= \frac{N_0}{2} \int_0^{T_b} |h(t)|^2 \, dt$$

$$= \frac{N_0}{2} \int_0^{T_b} \left(\frac{\sqrt{3}}{T_b} t\right)^2 dt$$

$$= \frac{N_0}{2} \int_0^{T_b} \frac{3}{T_b^2} t^2 \, dt$$

$$= \frac{3 N_0}{2 T_b^2} \frac{t^3}{3} \Big|_0^{T_b} = \frac{N_0}{2 \cancel{T_b^2}} \cancel{T_b^3}\, T_b$$

$$= \frac{N_0}{2} T_b$$

$$p(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(y-\mu_y)^2}{2\sigma^2}\right)$$

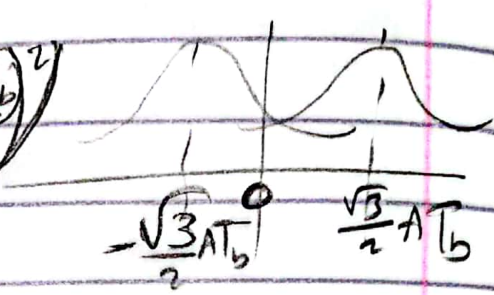
$$p(y|'0') = \frac{1}{\sqrt{2\pi \frac{N_0}{2} T_b}} \exp\left(\frac{-(y+\frac{\sqrt{3}}{2} A T_b)^2}{2 \frac{N_0}{2} T_b}\right)$$

$$p(y|'1') = \frac{1}{\sqrt{2\pi \frac{N_0}{2} T_b}} \exp\left(\frac{-(y-\frac{\sqrt{3}}{2} A T_b)^2}{2 \frac{N_0}{2} T_b}\right)$$

assume $p('0') = p('1') = 0.5$

$\therefore p(e) = p(e|'0') = p(e|'1')$

$$= \int_0^\infty \frac{1}{\sqrt{\pi N_0 T_b}} \exp\left(\frac{-\left(y+\frac{\sqrt{3}}{2}AT_b\right)^2}{N_0T_b}\right)$$



$-\sqrt{3}$ let $z = \dfrac{y+\frac{\sqrt{3}}{2}AT_b}{\sqrt{N_0T_b}}$

$$z\sqrt{N_0T_b} - \frac{\sqrt{3}}{2}AT_b = y$$

$$dz\sqrt{N_0T_b} = dy$$

@ $y=0 \rightarrow z = \dfrac{\frac{\sqrt{3}}{2}AT_b}{\sqrt{N_0T_b}}$

@ $y=\infty \rightarrow z=\infty$

$$p(e) = \int_{\frac{\sqrt{3}\,AT_b}{2\sqrt{N_0T_b}}}^{\infty} \frac{1}{\sqrt{\pi}\,\cancel{\sqrt{N_0T_b}}} \exp(-z^2)\,\cancel{\sqrt{N_0T_b}}\,dz$$

$$= \frac{1}{2}\,\mathrm{erfc}\left(\frac{\sqrt{3}\,AT_b}{2\sqrt{N_0T_b}}\right) = P_e$$

$$\boxed{= \frac{1}{2}\,\mathrm{erfc}\left(\sqrt{\frac{3A^2T_b}{2^2N_0}}\right)}$$

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