Cairo University
Faculty of Engineering

Department of Computer Engineering

ELC 325B – Spring 2023

Digital Communications

Assignment #2

Submitted to

Dr. mai

Dr. hala

Eng. Mohamed Khaled

Submitted by

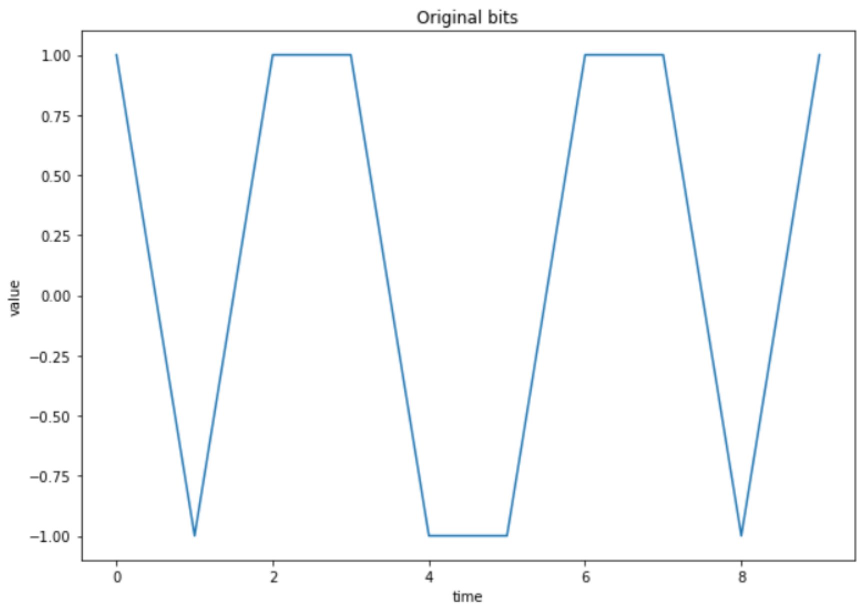
Name	Sec	BN
Norhan Reda Abdelwahed Ahmed	2	32
Hoda Gamal Hamouda Ismail	2	34

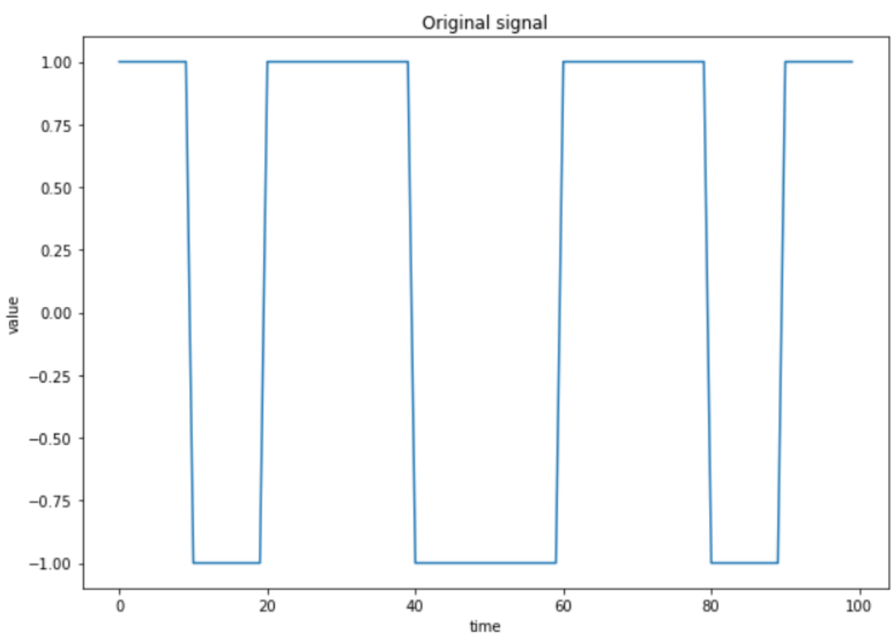
Part 2:

Figures:

Test for no of bits = 10, each bit from no of samples = 10

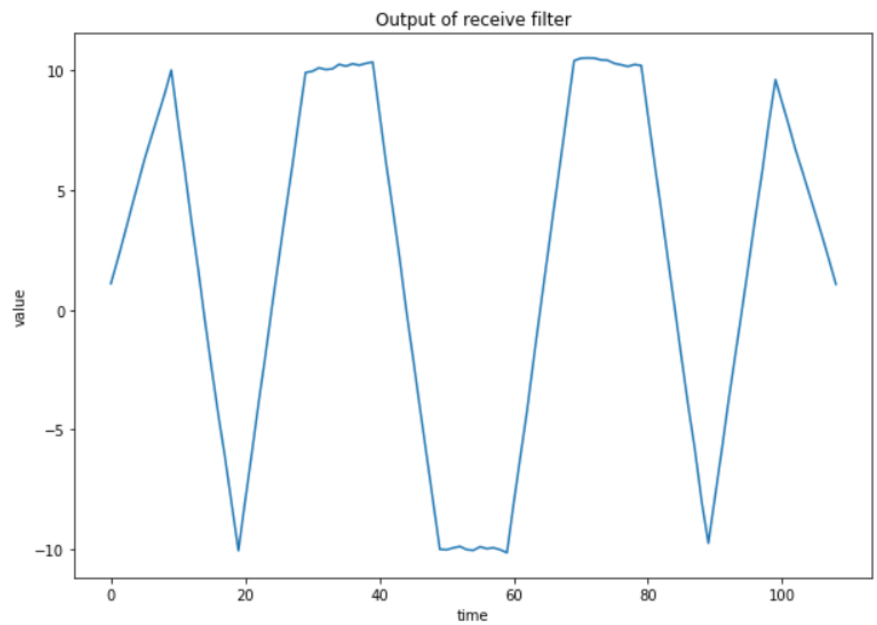
Original signal:

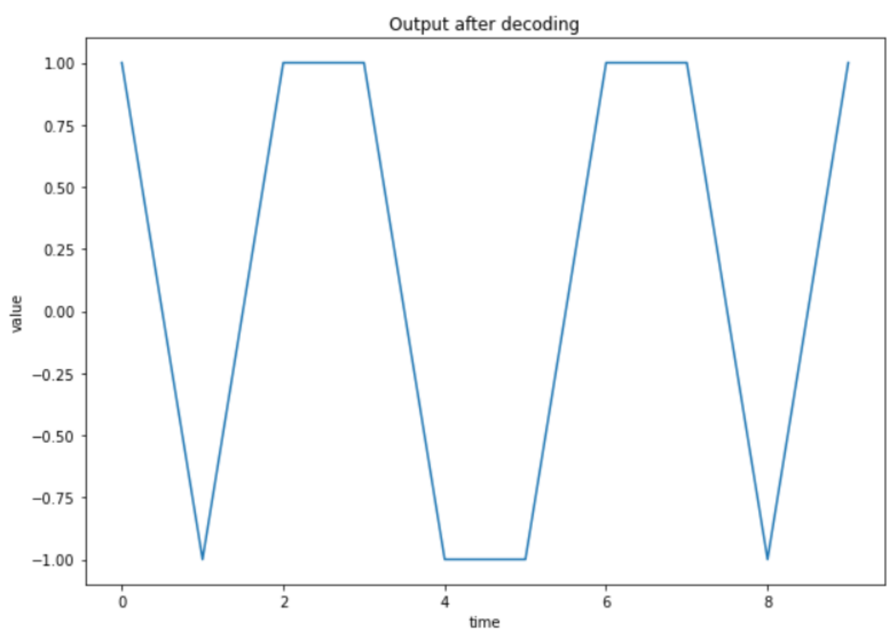




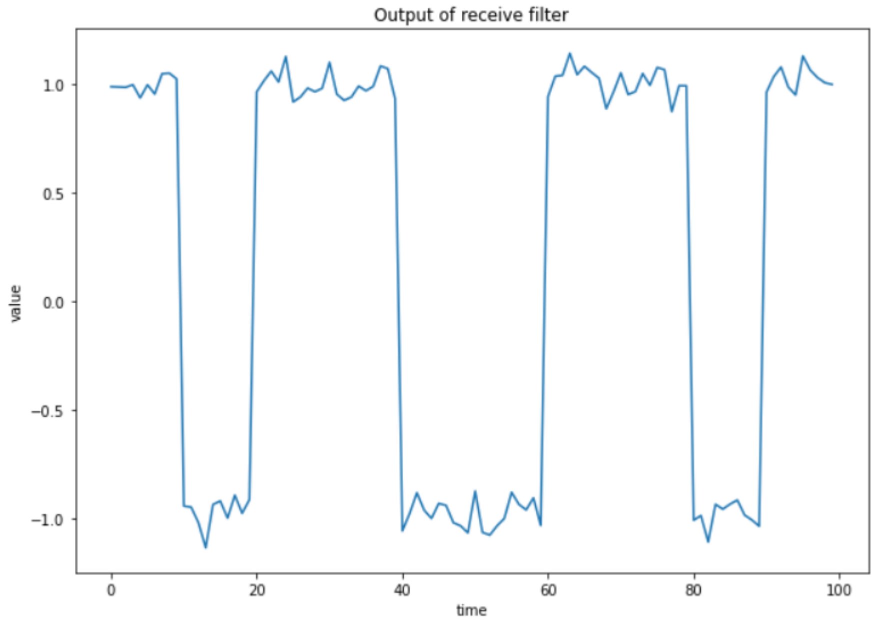
Output of the receive filter:

Case 1:

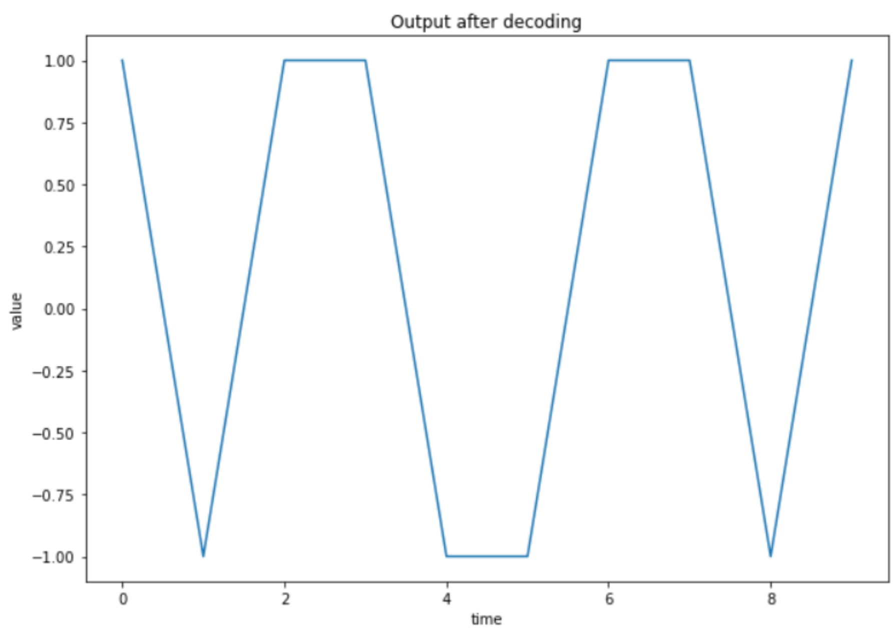




Case 2:

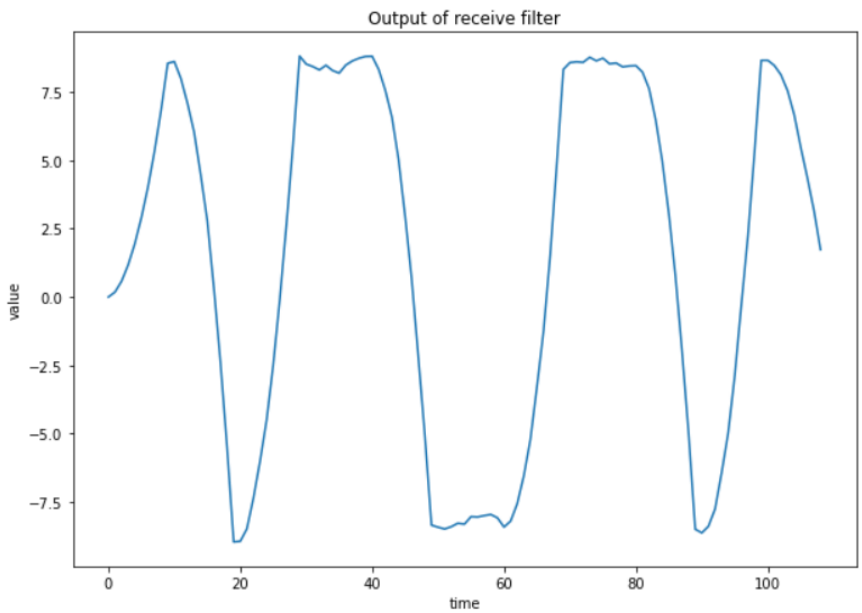


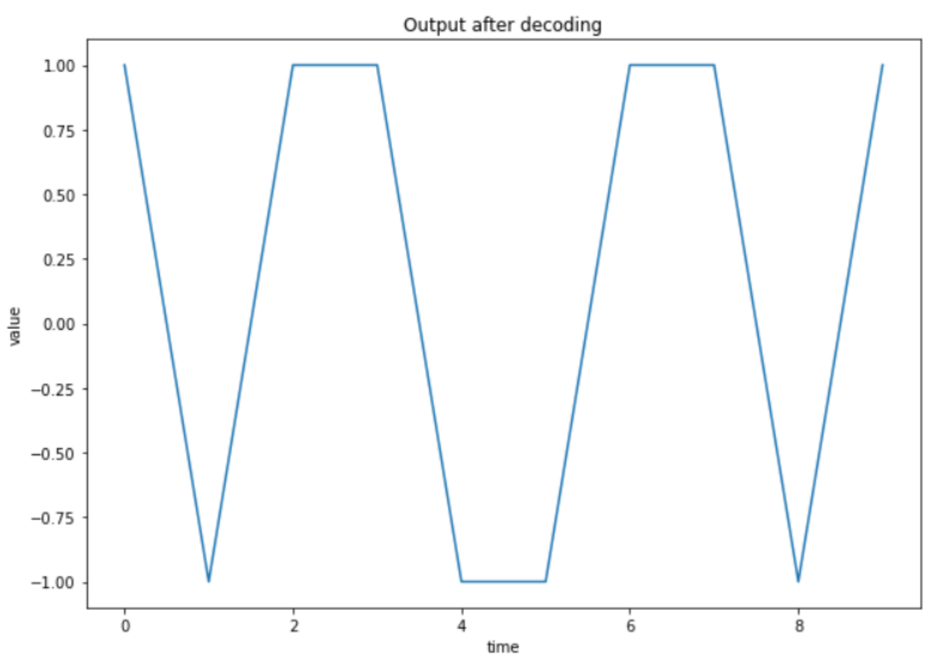
Output of receive filter
value
time
1.0
0.5
0.0
−0.5
−1.0
0
20
40
60
80
100



Output after decoding
value
time
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
2
4
6
8

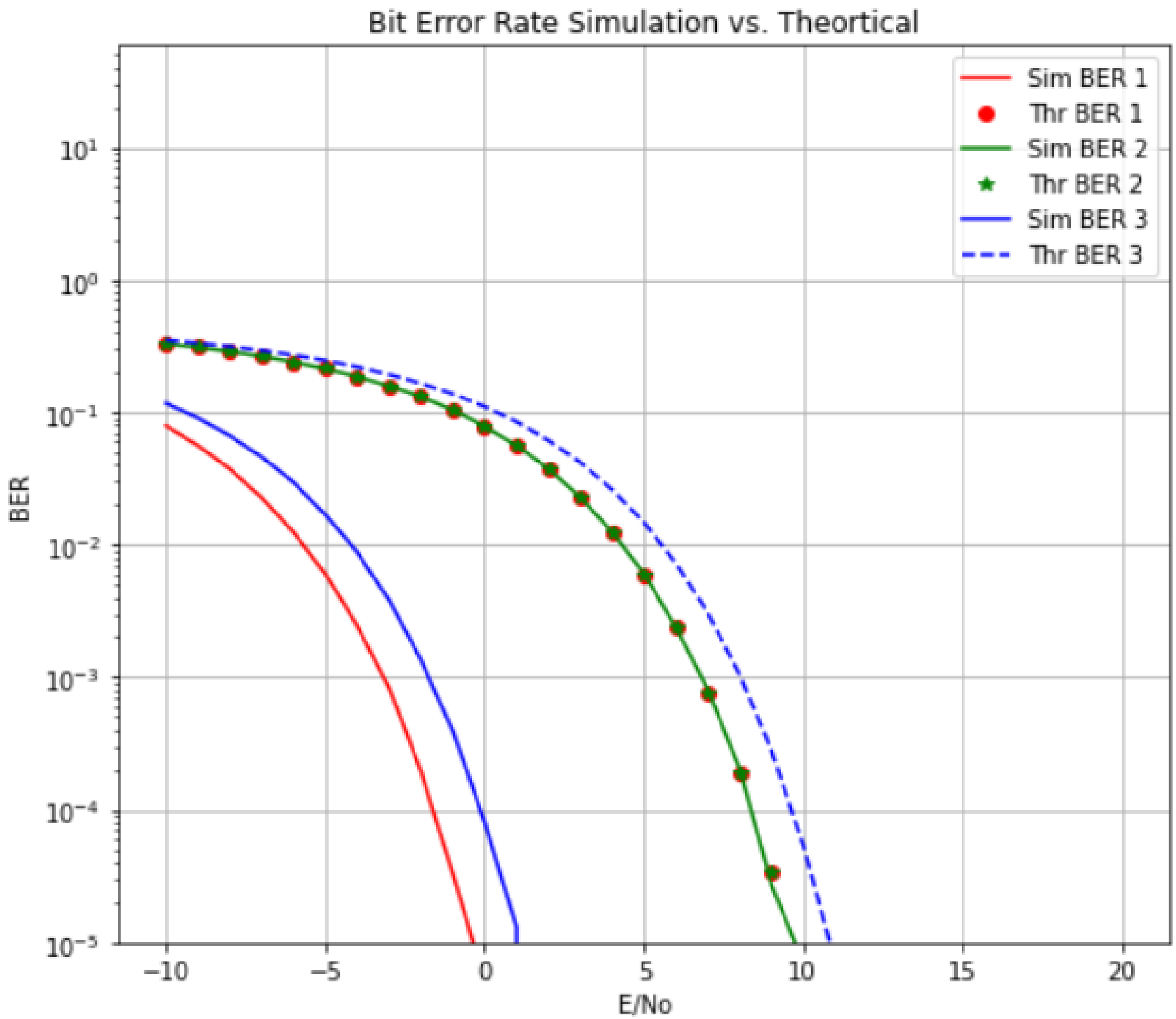
Case 3:


Output of receive filter
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
value
0
20
40
60
80
100
time


Output after decoding
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
value
0
2
4
6
8
time

Simulation & Theoretical BER vs. E/No:

Test for no of bits = 1000000, each bit from no of samples = 10



Note:

Thr BER 1 = Thr BER 2

avg sim BER 1 = 0.007015290322580645

avg sim BER 2 = 0.07850425806451612

avg sim BER 3 = 0.01229625806451613

min avg sim BER = 0.007015290322580645 -> case 1 of matched filter

avg sim BER 1 < avg sim BER 3 < avg sim BER 2

5. Is the BER an increasing or a decreasing function of E/No? Why?
increasing E/No -> decreasing BER

-> The sequence:
increase E/No
decrease No
decrease sigma of gaussian noise -> make gaussian noise distribution narrower
increase the transmitted energy to noise energy (increase SNR)
then noise didn't affect transmitted signal a lot
and make it easier to detect transmitted bits

-> The relationship between E/No ratio and BER is not linear, but logarithmic
BER decreases exponentially as the E/No ratio increases

6. Which case has the lowest BER? Why?
Case 1: The receive filter $h(t)$ is a matched filter with unit energy
it gives the lowest average simulation BER

because of that it uses a matched filter
which is designed to match the characteristics of the transmitted signal
and this makes the filter provides the best detection strategy
by correlating the received signal with a replica of the transmitted signal
so as a result the BER will be reduced and the SNR will be maximized

Code:

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

# generate random noise with gaussian distribution
def generate_gaussian_noise(mu, sigma, bits_no,bit_samples_no):
   return np.random.normal(mu, sigma, bits_no * bit_samples_no)
#############################################################

# generate random bits with values {-1,1} with probability = 0.5
def generate_random_bits(bits_no):
   return np.random.choice(a = [-1,1], size=bits_no, p=[1./2, 1./2])
#############################################################

def add_gaussian_noise(bits_no, bit_samples_no, signal_bits, noise):

   size = bits_no * bit_samples_no
   signal_samples=np.zeros(size)
   noisy_signal_samples=np.zeros(size)

    # repeat the bit with no of samples per bit
   for i in range(bits_no):
      # 0:9 10:19 20:39
      signal_samples[i * bit_samples_no : (i + 1) * bit_samples_no] = signal_bits[i]

   # add signal to the noise
   noisy_signal_samples = signal_samples + noise

   return signal_samples,noisy_signal_samples
#############################################################

def convolution(noisy_signal_samples,receive_filter,bits_no,bit_samples_no):

   size = bits_no * bit_samples_no
   result_samples = np.zeros(size)
   result_bits = np.zeros(bits_no)
   decoded_result_bits = np.zeros(bits_no)
```

```
    # convolution between signal and receive filter to get filter output
    if(receive_filter is None):
        result_samples = noisy_signal_samples
    else:
        result_samples = np.convolve(noisy_signal_samples, receive_filter)

    # sampling at T
    for i in range(bits_no):
        # 0:9->9 10:19->19 20:29->29
        result_bits[i]=result_samples[i * bit_samples_no + bit_samples_no - 1]

    # decoding to {-1,1}
    decoded_result_bits = np.sign(result_bits)

    return result_samples,result_bits,decoded_result_bits
##########################################################

def calc_simulation_error(signal_bits,decoded_result_bits,bits_no):
    error = np.sum(signal_bits != decoded_result_bits) / bits_no
    return error
##########################################################

def calc_theoretical_error(z):
     return math.erfc(z)
##########################################################

def test(receive_filter, thr_error_coeff, is_plot):
    sim_error, thr_error = [], []

    for E_No_db in range(-10, 21):
        E_No = 10 ** (E_No_db / 10) #get E/No from its db value
        E = 1
        sigma = np.sqrt(E/(2*E_No))

        noise = generate_gaussian_noise(0, sigma, bits_no, bit_samples_no)
        signal_samples, noisy_signal_samples = add_gaussian_noise(bits_no, bit_samples_no, signal_bits,
noise)
        result_samples,result_bits,decoded_result_bits = convolution(noisy_signal_samples, receive_filter,
bits_no, bit_samples_no)
```

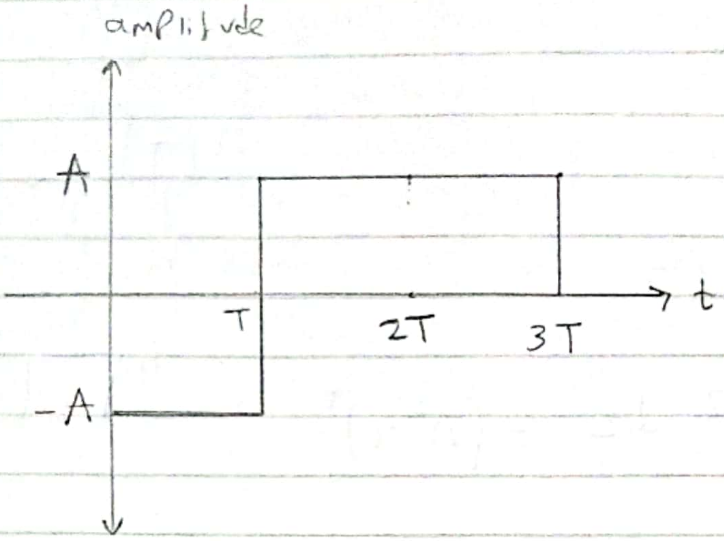
```
        x = calc_simulation_error(signal_bits,decoded_result_bits,bits_no)
        y = 0.5 * calc_theoretical_error(thr_error_coeff * (E_No ** 0.5))
        sim_error.append(x)
        thr_error.append(y)

    if(is_plot):
        plt.figure(figsize=(10,7))
        plt.plot(range(0,len(signal_bits)), signal_bits)
        plt.title("Original bits")
        plt.xlabel('time')
        plt.ylabel('value')

        plt.figure(figsize=(10,7))
        plt.plot(range(0,len(signal_samples)), signal_samples)
        plt.title("Original signal")
        plt.xlabel('time')
        plt.ylabel('value')

        plt.figure(figsize=(10,7))
        plt.plot(range(0,len(result_samples)), result_samples)
        plt.title("Output of receive filter")
        plt.xlabel('time')
        plt.ylabel('value')

        plt.figure(figsize=(10,7))
        plt.plot(range(0,len(decoded_result_bits)), decoded_result_bits)
        plt.title("Output after decoding")
        plt.xlabel('time')
        plt.ylabel('value')

    return sim_error, thr_error
#########################################################

bits_no,bit_samples_no = 10 , 10
size = bits_no * bit_samples_no
signal_bits = generate_random_bits(bits_no)

receive_filter1 = np.ones(bit_samples_no)
receive_filter2 = None
receive_filter3 = np.linspace(0, 1, bit_samples_no) * np.sqrt(3)
```

```
sim_error1, thr_error1, sim_error2, thr_error2, sim_error3, thr_error3 = [], [], [], [], [], []

sim_error1, thr_error1 = test(receive_filter1, 1, True)
sim_error2, thr_error2 = test(receive_filter2, 1, True)
sim_error3, thr_error3 = test(receive_filter3, (3**0.5/2), True)
##########################################################

# test to simulate error
bits_no,bit_samples_no = 1000000 , 10
signal_bits = generate_random_bits(bits_no)

sim_error1, thr_error1 = test(receive_filter1, 1, False)
sim_error2, thr_error2 = test(receive_filter2, 1, False)
sim_error3, thr_error3 = test(receive_filter3, (3**0.5/2), False)

# plot
plt.figure(figsize=(8,7))
x_axis_range = range(-10, 21)

plt.semilogy(x_axis_range,sim_error1,'r-')
plt.semilogy(x_axis_range,thr_error1,'ro')
plt.semilogy(x_axis_range,sim_error2,'g-')
plt.semilogy(x_axis_range,thr_error2,'g*')
plt.semilogy(x_axis_range,sim_error3,'b-')
plt.semilogy(x_axis_range,thr_error3,'b--')

plt.title('Bit Error Rate Simulation vs. Theortical')
plt.legend(['Sim BER 1', 'Thr BER 1', 'Sim BER 2', 'Thr BER 2', 'Sim BER 3', 'Thr BER 3'])
plt.xlabel('E/No')
plt.ylabel('BER')
plt.yscale('log')
plt.ylim(10**(-4))
plt.grid()
plt.show()
```

Subject موضوع الدرس Date التاريخ

Part I

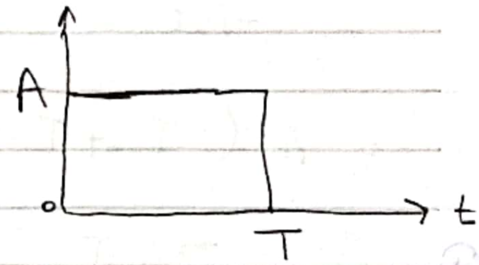
(a)

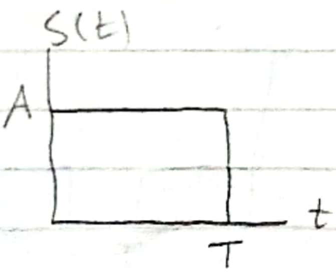


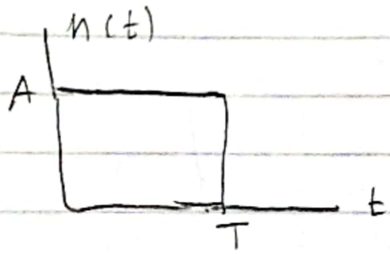
(b) $g(t) = A\,\mathrm{rect}\left(\frac{t-\frac{T}{2}}{T}\right)$

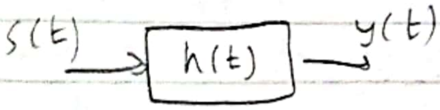
$h(t) = g(T-t) = A\,\mathrm{rect}\left(\frac{T-t-\frac{T}{2}}{T}\right) = h(t)$

$h(t) = A\,\mathrm{rect}\left(\frac{\frac{T}{2}-t}{T}\right)$



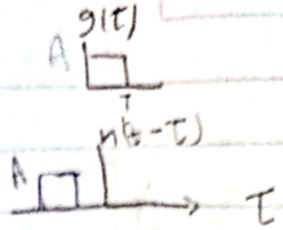






$y(t) = s(t) * h(t)$

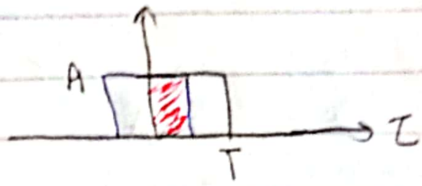
$$y(t) = \int_{-\infty}^{\infty} s(\tau)\, h(t-\tau)\, d\tau$$



* for $t < 0$

$y(t) = 0$

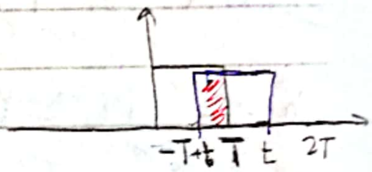
* for $0 < t < T$



$$y(t) = \int_0^t A^2 \, d\tau = \left(A^2 \tau\right)_0^t$$

$$= A^2 t$$

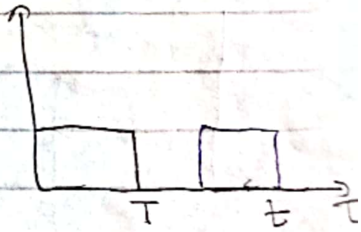
* for $T < t < 2T$



$$y(t) = \int_{-T+t}^{T} A^2 \, d\tau = \left[A^2 \tau\right]_{-T+t}^{T}$$

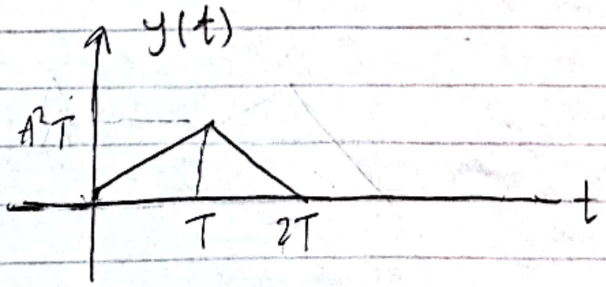
$$= A^2(T + T - t) = A^2(2T - t)$$

* for $t > 2T$



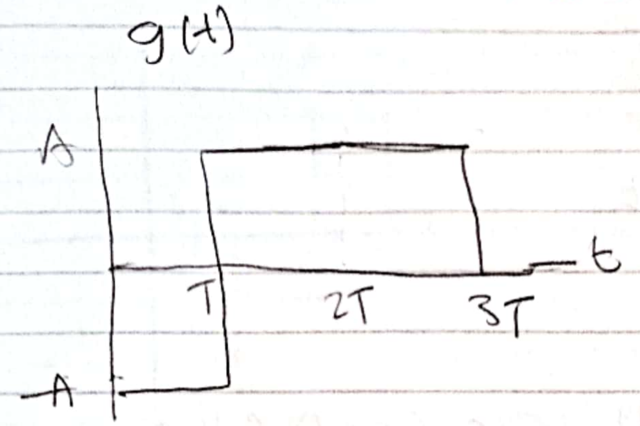
$y(t) = 0$

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

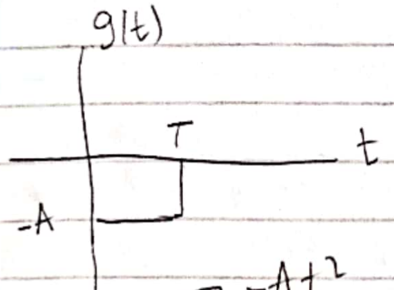


3

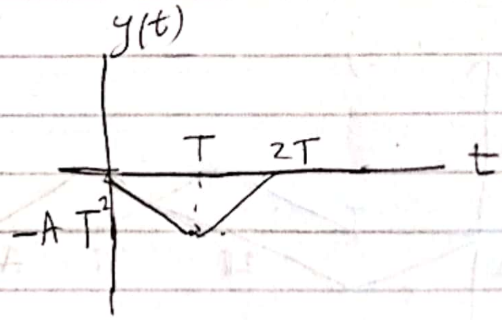
Subject | موضوع الدرس | Date | التاريخ



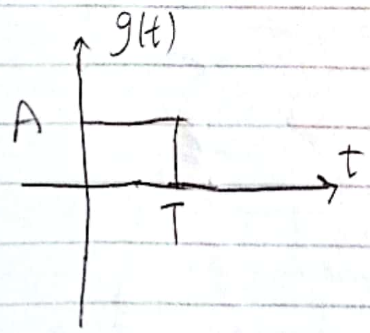
from the last result

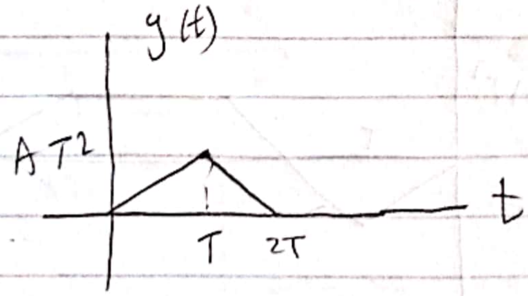


$\Longrightarrow$



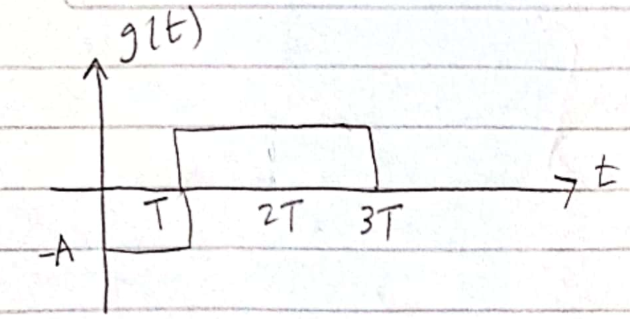
$$y(t) = \begin{cases} -At^2 & 0 < t < T \\ -A^2(2T-t) & T < t < 2T \\ 0 & \text{otherwise} \end{cases}$$



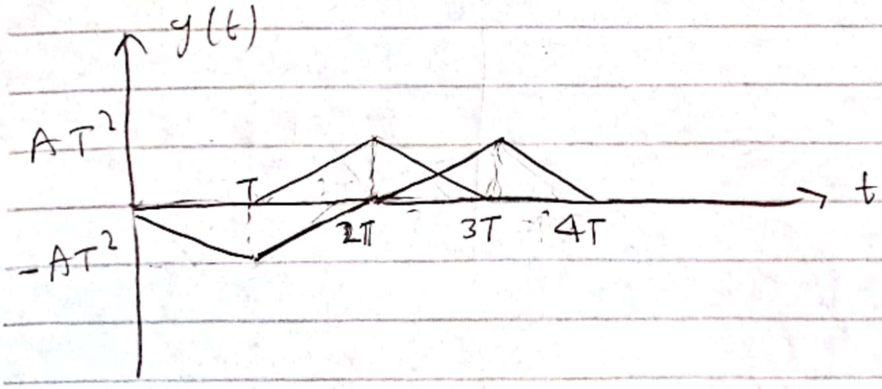


$$y(t) = \begin{cases} At^2 & 0 < t < T \\ A^2(2T-t) & T < t < 2T \\ 0 & \text{otherwise} \end{cases}$$

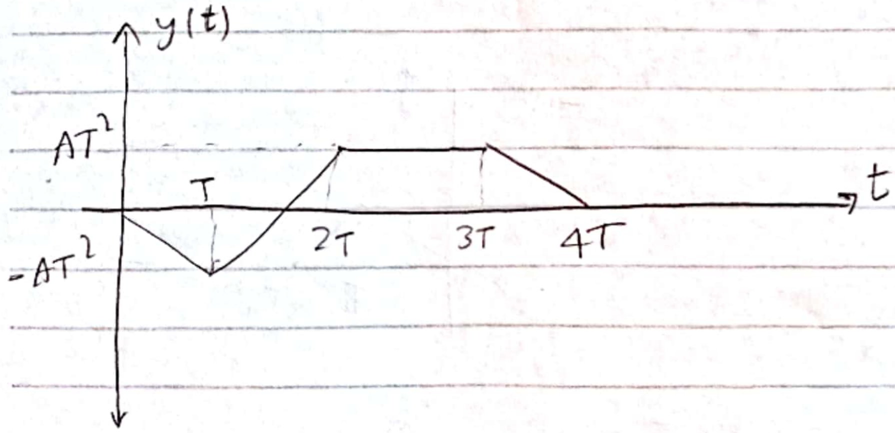
Subject موضوع الدرس Date التاريخ



for this signal by applying super position to get y(t)



by applying summitions

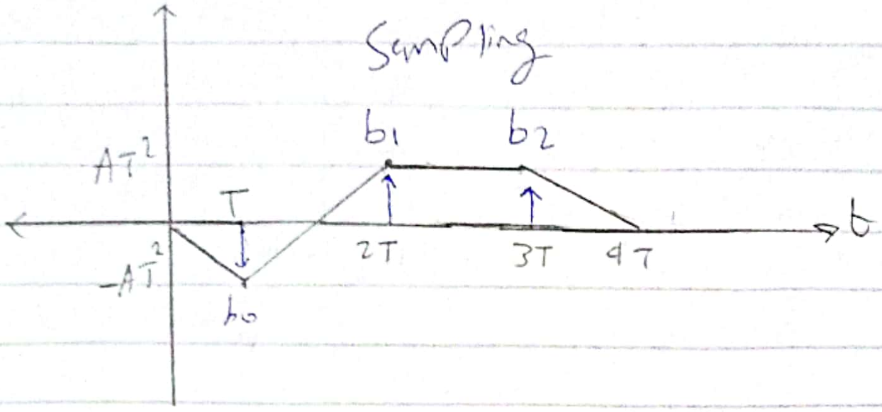


this is the output of the matched filter

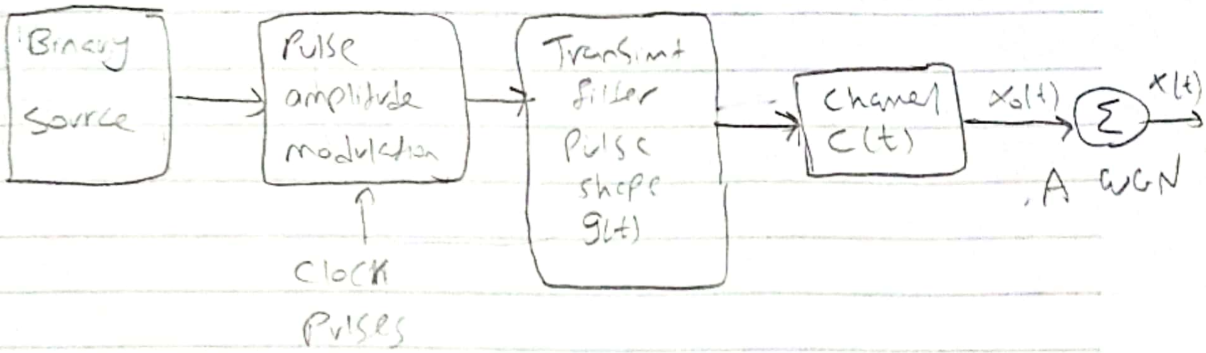
Subject موضوع الدرس

Date التاريخ

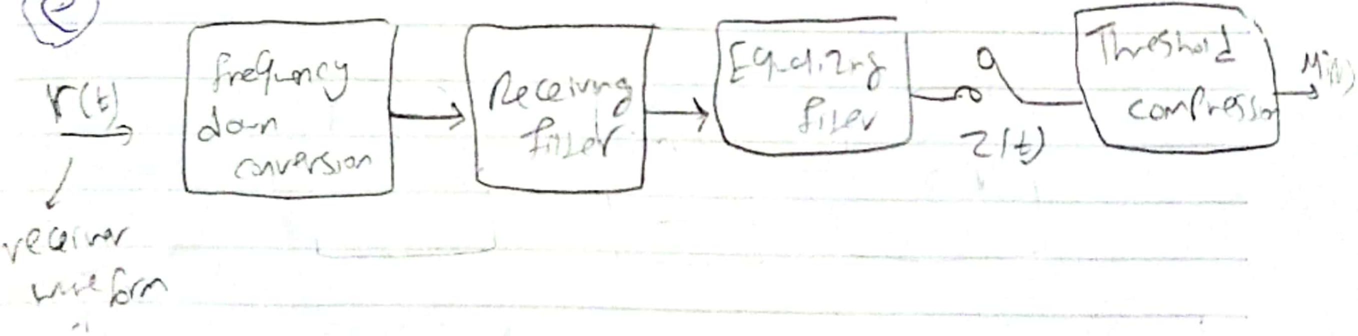
c



d



e

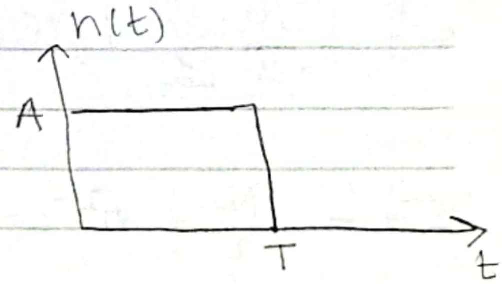


Part II $\quad g(t) = \begin{cases} -A \\ A, & 0 < t < T \end{cases}$

ⓐ the receiver filter $h(t)$ is a matched filter with unit energy

$$g(t) = A \,\text{rect}\left(\frac{t - \frac{T}{2}}{T}\right) \qquad r(t) \rightarrow \boxed{h(t)} \rightarrow y(t)$$

$$h(t) = g(T-t) = A\,\text{rect}\left(\frac{T - t - \frac{T}{2}}{T}\right) = A\,\text{rect}\left(\frac{\frac{T}{2} - t}{T}\right)$$



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

$y(t) = r(t) * h(t)$

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

$\quad = g_o(t) + n(t)$

from part Ⓘ ⓑ

$$g_o(t) = \begin{cases} 0 & \text{otherwise} \\ A^2 t & 0 < t \leq T \\ A^2(2T - t) & T < t \leq 2T \end{cases}$$

$$\therefore y(T) = \begin{cases} -A^2 T + n(T) & \text{"0"} \\ A^2 T + n(T) & \text{"1"} \end{cases}$$

$N_y = E(y(T)) = E(g_o(T)) + E(n(T))$

$N_y = E(\pm A^2 T + n(T)) = \pm A^2 T + E(n(T))$

$E(n(T)) = E\left(\int w(\tau)\, h(T - \tau)\, d\tau\right)$

$\quad = E\left(\int w(\tau)\, g(\tau)\, d\tau\right) =$

$E\left(\int_0^T \pm A\, w(\tau)\, d\tau\right) = \int_0^T \pm A\, E(w(\tau))\, d\tau = 0$

$\therefore N_y = \pm A^2 T$

$$\therefore N_y = \begin{cases} -A^2 T & (0) \\ A^2 T & (1) \end{cases}$$

Subject موضوع الدرس Date التاريخ

$$\sigma_y^2 = Var(y(T)) = E(g_0(T) + n(T))$$

$$\sigma_y^2 = var(n(T)) = E(n^2(T)) - E(\underbrace{n(T)}_{0})^2$$

$$\sigma_y^2 = E(n^2(T)) = \int_{-\infty}^{\infty} S_n(f)\, df$$

$$\sigma_y^2 = \frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2\, df = \frac{N_0}{2} \int_0^T |h(t)|^2\, dt$$

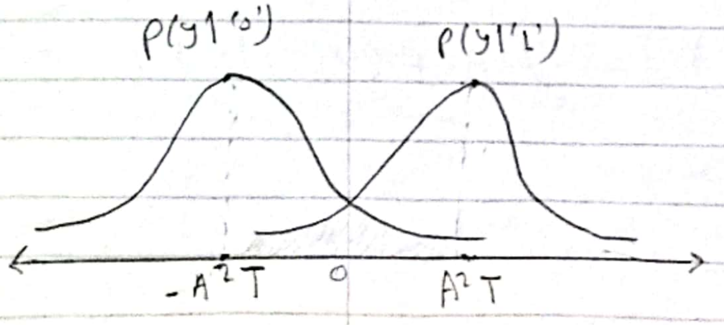
$$\sigma_y^2 = \frac{N_0}{2} * A^2 T = \frac{N_0 A^2 T}{2}$$

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

$$p(y\,|\,'0') = \frac{1}{\sqrt{2\pi * \frac{N_0 A^2 T}{2}}} e^{-\frac{(y+A^2T)^2}{2 \times \frac{N_0 A^2 T}{2}}}$$

$$p(y\,|\,'0') = \frac{1}{\sqrt{\pi N_0 A^2 T}} e^{-\frac{(y+A^2T)^2}{N_0 A^2 T}}$$

$$p(y\,|\,'1') = \frac{1}{\sqrt{\pi N_0 A^2 T}} e^{-\frac{(y-A^2T)^2}{N_0 A^2 T}}$$



$$P(e|'0') = \int_0^\infty \frac{1}{\sqrt{\pi N_0 A^2 T}} e^{\frac{-(y+A^2T)^2}{N_0 A^2 T}} dy$$

let $z = \frac{y + A^2T}{\sqrt{N_0 A^2 T}}$

$$dz = \frac{dy}{\sqrt{N_0 A^2 T}}$$

$$P(e|'0') = \int_{\frac{A^2T}{\sqrt{N_0A^2T}}}^{\infty} \frac{1}{\sqrt{\pi}\cancel{\sqrt{N_0A^2T}}} \times e^{-z^2} \times \cancel{\sqrt{N_0A^2T}}\, dz$$

$$P(e|'0') = \int_{\frac{A^2T}{\sqrt{N_0A^2T}}}^{\infty} \frac{1}{\sqrt{\pi}} e^{-z^2} dz = \frac{1}{2} \operatorname{erfc}\left(\frac{A^2T}{\sqrt{N_0 A^2 T}}\right)$$

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

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

$$\therefore P(e) = 2\,P(e|'0') \times 0.5 = P(e|'0')$$

$$\therefore P(e) = \frac{1}{2} \operatorname{errfc}\left(\frac{A^2T}{\sqrt{N_0 A^2 T}}\right)$$

$A = 1$, $T = 1$

$$\therefore P(e) = \frac{1}{2} \operatorname{errfc}\left(\frac{1}{\sqrt{N_0}}\right)$$

(b) the receive filter $h(t)$ is not existent $(h(t) = \delta(t))$

$y(t) = r(t) * h(t) = r(t) * \delta(t) = y(t)$

convolution with δ gives the same signal

$y(T) = g(T) + w(T)$

$y(T) = \pm A + w(T)$

$E(y(T)) = E(\pm A + w(T)) = \pm A + E(w(T))$ $\to 0$

$$N_y = \begin{cases} -A & ('0') \\ A & ('1') \end{cases}$$

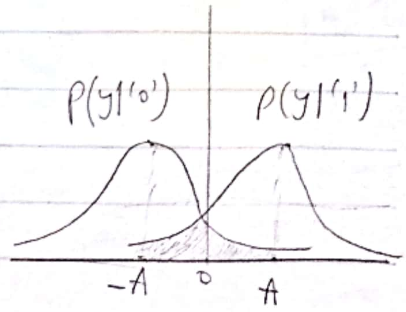
$\sigma_y^2 = Var(y(T)) = Var(\pm A + w(T)) = Var(w(T))$

$\sigma_y^2 = var(w(T)) = \frac{N_0}{2}$

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

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

Due to symmetry $\lambda = 0$, $P('0') = P('1')$



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

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

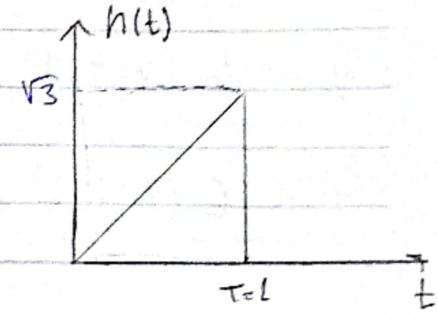
let $z = \frac{y+A}{\sqrt{N_0}}$, $dz = \frac{dy}{\sqrt{N_0}}$

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

$P(e) = p(e|'0') \times P('0') + P(e|'1')\, P('1') = 2P(e|'0') \times 0.5 =$

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

(c) The receive filter $h(t)$ has the following impulse response



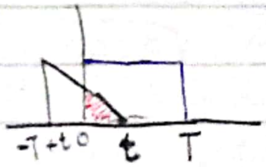
$$y(t) = r(t) * h(t) = g_o(t) + n(t)$$

$$g_o(t) = g(t) * h(t)$$

$$g_o(t) = \int_{-\infty}^{\infty} g(\tau)\, h(t-\tau)\, d\tau$$

* for $t < 0$

$$g_o(t) = 0$$



* for $0 < t \leq T$

$$g_o(t) = \int_0^t \sqrt{3} A \tau\, d\tau = \sqrt{3} A \left[\frac{\tau^2}{2}\right]_0^t = \frac{\sqrt{3}}{2} A t^2$$

$$= \sqrt{3} A t$$

$$y(T) = \begin{cases} -\frac{\sqrt{3}}{2} A T^2 + n(T) & \text{'0'} \\ \frac{\sqrt{3}}{2} A T^2 + n(T) & \text{'1'} \end{cases}$$

$$\mu_y = E(y(T)) = E(g_o(T)) + E(n(T))$$

$$\mu_y = E\left(\pm \frac{\sqrt{3}}{2} A T^2 + n(T)\right) = \pm \frac{\sqrt{3}}{2} A T^2 + E(n(T))$$

$$E(n(T)) = E\left(\int w(\tau) g(\tau)\, d\tau\right)$$

$$= E\left(\int_0^T \pm A\, w(\tau)\, d\tau\right) = \int_0^T \pm A\, E(w(\tau))\, d\tau = 0$$

$$\therefore \mu_y = \begin{cases} -\frac{\sqrt{3}}{2} A T^2 & \text{'0'} \\ \frac{\sqrt{3}}{2} A T^2 & \text{'1'} \end{cases}$$

$$\sigma_y^2 = Var(y(T)) = E(g_0(T) + n(T)) = E(\pm\frac{\sqrt{3}}{2}AT^2 + n(T))$$

$$\sigma_y^2 = Var(n(T)) = E(n^2(T)) + E(n(T))^2$$

$$\sigma_y^2 = E(n^2(T)) = \int_{-\infty}^{\infty} S_n(f)\,df$$

$$\sigma_y^2 = \frac{N_0}{2}\int_{-\infty}^{\infty} |H(f)|^2 df = \frac{N_0}{2}\int_0^T |h(t)|^2 dt$$

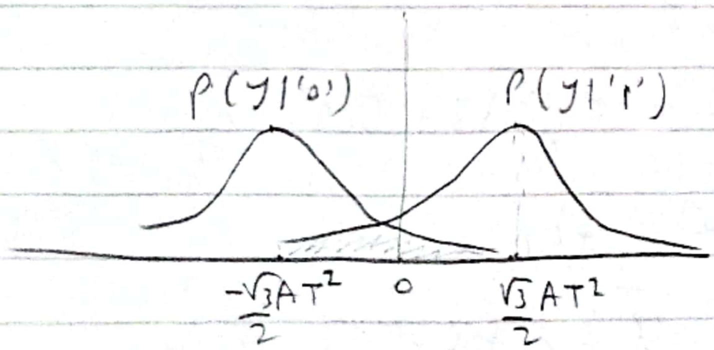
$$\sigma_y^2 = \frac{N_0}{2}\int_0^T 3t^2\,dt = \frac{N_0}{2}\left[3\frac{t^3}{3}\right]_0^T = \frac{N_0}{2}T^3$$

$$P(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-N)^2}{2\sigma^2}}$$

$$P(y|'0') = \frac{1}{\sqrt{2\pi \times \frac{N_0T^3}{2}}} e^{-\frac{(y+\frac{\sqrt{3}}{2}AT^2)}{N_0T^3}}$$

$$P(y|'0') = \frac{1}{\sqrt{\pi N_0 T^3}} e^{\frac{-(y+\frac{\sqrt{3}}{2}AT^2)}{N_0T^3}}$$

$$P(y|'1') = \frac{1}{\sqrt{\pi N_0 T^3}} e^{\frac{-(y-\frac{\sqrt{3}}{2}AT^2)}{N_0T^3}}$$



$$p(e|'0') = \int_0^\infty \frac{1}{\sqrt{\pi N_o T^3}} e^{-\frac{(y+\frac{\sqrt{3}}{2}AT^2)}{N_o T^3}}$$

$$\text{let } z = \frac{y+\frac{\sqrt{3}}{2}AT^2}{\sqrt{N_o T^3}}$$

$$dz = \frac{dy}{\sqrt{N_o T^3}}$$

$$p(e|'0') = \int_{\frac{\frac{\sqrt{3}}{2}AT^2}{\sqrt{N_o T^3}}}^{\infty} \frac{1}{\sqrt{\pi}\cdot\cancel{\sqrt{N_o T^3}}} e^{-z} \cancel{\sqrt{N_o T^3}}\, dz$$

$$= \frac{1}{2} \operatorname{erfc}\left(\frac{\frac{\sqrt{3}}{2}AT^2}{\sqrt{N_o T^3}}\right)$$

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

$$p(e) = p(e|'0') = \frac{1}{2} \operatorname{erfc}\left(\frac{\frac{\sqrt{3}}{2}AT^2}{\sqrt{N_o T^3}}\right)$$

$$A = 1,\ T = 1$$

$$p(e) = p(e|'0') = \frac{1}{2} \operatorname{erfc}\left(\frac{\frac{\sqrt{3}}{2}}{\sqrt{N_o}}\right)$$