

Estado	Finalizado
Comenzado	martes, 6 de mayo de 2025, 17:36
Completado	martes, 6 de mayo de 2025, 18:03
Duración	26 minutos 59 segundos
Calificación	67 de 100



Pregunta 1

Correcta

Se puntúa 33 sobre 33

Complete el texto con los siguientes conceptos según corresponda:

$N(t)$ es el con $N_0/2$ si para un conjunto finito de funciones $g_1(\alpha), \dots, g_k(\alpha)$,

$$Z_i = \int N(\alpha) g_i(\alpha) d\alpha \quad i = 1, 2, \dots, k$$

es un conjunto de variables aleatorias de media cero y covarianza

$$\text{cov}(Z_i, Z_j) = E[Z_i Z_j^*] = \frac{N_0}{2} \int g_i(t) g_j^*(t) dt = \frac{N_0}{2}$$

Es decir que, si $g_1(t), \dots, g_k(t)$ es un conjunto de funciones , $Z = (Z_1, \dots, Z_k)^T$ es un vector aleatorio con de media cero y componentes de varianza $\sigma^2 = \frac{N_0}{2}$.

Respuesta correcta

La respuesta correcta es:

Complete el texto con los siguientes conceptos según corresponda:

$N(t)$ es el [ruido gaussiano] con [densidad espectral de potencia] $N_0/2$ si para un conjunto finito de funciones [realmente valuadas] $g_1(\alpha), \dots, g_k(\alpha)$,

$$Z_i = \int N(\alpha) g_i(\alpha) d\alpha \quad i = 1, 2, \dots, k$$

es un conjunto de variables aleatorias [conjuntamente gaussianas] de media cero y covarianza

$$\text{cov}(Z_i, Z_j) = E[Z_i Z_j^*] = \frac{N_0}{2} \int g_i(t) g_j^*(t) dt = \frac{N_0}{2}$$

Es decir que, si $g_1(t), \dots, g_k(t)$ es un conjunto de funciones [ortonormales realmente valuadas], $Z = (Z_1, \dots, Z_k)^T$ es un vector aleatorio con [distribución gaussiana] de media cero y componentes [independientes e igualmente distribuidas] de varianza $\sigma^2 = \frac{N_0}{2}$.



Pregunta 2

Correcta

Se puntúa 34 sobre 34

Indique las opciones correctas

Seleccione una o más de una:

- ☐ a. La salida de un filtro apareado se expresa como $y(t) = \int r(\alpha)b^*(t - \alpha)d\alpha$, siendo $r(t)$ la entrada al receptor y $b(t)$ una función $\psi_j(t)$ o $w_j(t)$.
- ☒ b. Para obtener una implementación de $\int r(t)b^*(t)dt$ la salida del filtro apareado debe ser muestreada.  Verdadero.
- ☒ c. La respuesta al impulso del filtro apareado es $h(t) = b^*(T - t)$, siendo T un parámetro seleccionado de manera que $h(t)$ sea causal.  Verdadero.
- ☐ d. La única forma de implementar una operación de tipo $\int r(t)b^*(t)dt$ es a través de la utilización de un filtro apareado.

Respuesta correcta

Las respuestas correctas son: Para obtener una implementación de $\int r(t)b^*(t)dt$ la salida del filtro apareado debe ser muestreada., La respuesta al impulso del filtro apareado es $h(t) = b^*(T - t)$, siendo T un parámetro seleccionado de manera que $h(t)$ sea causal.

Pregunta 3

Incorrecta

Se puntúa 0 sobre 33

Los siguientes parámetros permiten calcular el bit error rate para una modulación antipodal (por ejemplo BPSK)

P_T =Potencia de señal transmitida [*watts*]

λ =Longitud de onda de la portadora [*m*]

G_T =Ganancia de la antena transmisora

G_R =Ganancia de la antena receptora

d = distancia [*Km*]

T_N = Temperatura de ruido del receptor [*kelvin*]

R_b = bit rate [*kbps*]

Escriba una función en python3 `BER(pt,lda,gt,gr,d,tn,rb)` que devuelva el bit error rate (BER)

Expresa el resultado con con 4 decimales (para su corrección automática)

Ayuda:

incluya la librería numpy para acceder a la función `sqrt()`

incluya la librería scipy para acceder a la función `Q`

"from scipy.stats import norm"

Editor online: <https://repl.it/languages/python3>

Por ejemplo:

Prueba	Resultado
<code>print(BER(16.8 ,0.13,575.44 ,1.38e6 ,1.6e8,13.5 ,117.6))</code>	0.012

Respuesta: (sistema de penalización: 10, 20, ... %)

Reiniciar respuesta

```
1 # incluir librerías necesarias
2 import math
3
4 kb = 1.381e-23 # Constante de Boltzmann
5
6 def truncate(x, decimals=4):
7     factor = 10 ** decimals
8     return math.floor(x * factor) / factor
9
10 def BER(pt, lda, gt, gr, d, tn, rb):
11     d_m = d * 1000
12     rb_bps = rb * 1e3
13     pr = pt * (lda / (4 * math.pi * d_m))**2 * gt * gr
14     eb_no = pr / (kb * tn * rb_bps)
15     q_arg = math.sqrt(2 * eb_no)
16     ber = 0.5 * math.erfc(q_arg / math.sqrt(2))
17     return truncate(ber, 4)
```

Debug: source code from all test runs

Run 1

```
# incluir librerías necesarias
import math

kb = 1.381e-23 # Constante de Boltzmann

def truncate(x, decimals=4):
    factor = 10 ** decimals
    return math.floor(x * factor) / factor

def BER(pt, lda, gt, gr, d, tn, rb):
    d_m = d * 1000
    rb_bps = rb * 1e3
    pr = pt * (lda / (4 * math.pi * d_m))**2 * gt * gr
    eb_no = pr / (kb * tn * rb_bps)
    q_arg = math.sqrt(2 * eb_no)
    ber = 0.5 * math.erfc(q_arg / math.sqrt(2))
    return truncate(ber, 4)

__student_answer__ = """# incluir librerías necesarias
import math

kb = 1.381e-23 # Constante de Boltzmann

def truncate(x, decimals=4):
    factor = 10 ** decimals
    return math.floor(x * factor) / factor

def BER(pt, lda, gt, gr, d, tn, rb):
    d_m = d * 1000
    rb_bps = rb * 1e3
    pr = pt * (lda / (4 * math.pi * d_m))**2 * gt * gr
    eb_no = pr / (kb * tn * rb_bps)
    q_arg = math.sqrt(2 * eb_no)
    ber = 0.5 * math.erfc(q_arg / math.sqrt(2))
    return truncate(ber, 4)"""

SEPARATOR = "<ab@17943918#@>#"

print(BER(16.8 ,0.13,575.44 ,1.38e6 ,1.6e8,13.5 ,117.6))
print(SEPARATOR)
print(BER(16.8 ,0.26,575.44 ,1.38e6 ,1.6e11,13.5 ,117.6))
print(SEPARATOR)
print(BER(10.8 ,0.13,300 ,1.38e5 ,1.6e8,10.5 ,117.6))
print(SEPARATOR)
print(BER(1 ,0.15,1 ,100 ,800,300 ,1000))
```

	Prueba	Esperado	Conseguido	
✓	print(BER(16.8 ,0.13,575.44 ,1.38e6 ,1.6e8,13.5 ,117.6))	0.012	0.012	✓
✓	print(BER(16.8 ,0.26,575.44 ,1.38e6 ,1.6e11,13.5 ,117.6))	0.4982	0.4982	✓
✗	print(BER(10.8 ,0.13,300 ,1.38e5 ,1.6e8,10.5 ,117.6))	0.3197	0.3198	✗
✓	print(BER(1 ,0.15,1 ,100 ,800,300 ,1000))	0.0005	0.0005	✓

Su código debe superar todas las pruebas para conseguir puntuación. Vuelva a intentarlo.

Mostrar diferencias

► Show/hide question author's solution (Python3)

Incorrecta

Puntos para este envío: 0/33.

