

Detección de Señales

Ejercicio N° 1

Objetivos	<ul style="list-style-type: none">• Buenas prácticas en programación de Tipos de Datos Abstractos (TDAs)• Modularización de sistemas• Correcto uso de recursos (memoria dinámica y archivos)
Instancias de Entrega	Entrega 1: clase 4 (31/03/2015). Entrega 2: clase 6 (14/04/2015).
Temas de Repaso	<ul style="list-style-type: none">• Uso de structs y typedef• Uso de macros y archivos de cabecera• Funciones para el manejo de Strings
Criterios de Evaluación	<ul style="list-style-type: none">• Criterios de ejercicios anteriores• Cumplimiento de la totalidad del enunciado del ejercicio• Entrega de un diagrama explicativo junto con el informe• Ausencia de variables globales• Correcta encapsulación en TDAs y separación en archivos• Uso de interfaces para acceder a datos contenidos en TDAs• Empleo de memoria dinámica de forma ordenada y moderada• Acceso a información de archivos de forma ordenada y moderada

Índice

[Introducción](#)

[Teoría de Detección de Señales](#)

[Modelo de Comunicaciones](#)

[Detección de una Señal en Ruido](#)

[Ejemplo 1](#)

[Ejemplo 2](#)

[Descripción](#)

[Entrada](#)

[Salida](#)

[Formato de Línea de Comandos](#)

[Códigos de Retorno](#)

[Entrada y Salida Estándar](#)

[Ejemplos de Ejecución](#)

[Ejemplo 1](#)

[Ejemplo 2](#)

[Restricciones](#)

[Referencias](#)

Introducción

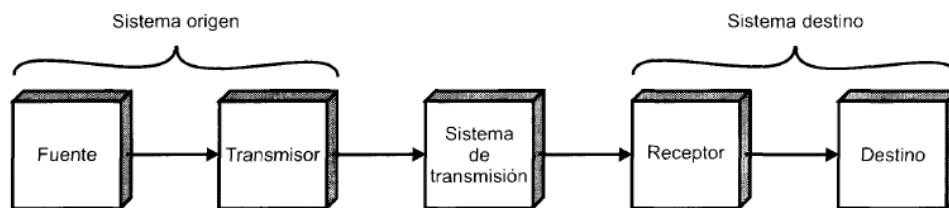
Una importante empresa nos encarga un programa que permita detectar las señales recibidas por un receptor a fin de ser implementado en uno de los últimos productos de la compañía.

El programa deberá detectar señales de diversas longitudes, permitir utilizar distintas codificaciones (S_0 y S_1), y diversos parámetros ($P(1)$, $P(0)$, σ^2).

Teoría de Detección de Señales

Modelo de Comunicaciones

En un modelo de comunicaciones identificamos los siguientes elementos:

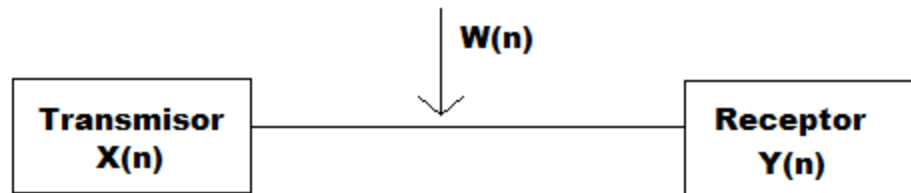


- **Fuente:** Es quien genera los datos a transmitir.
- **Transmisor:** Por diversas razones (sincronismo entre el transmisor y receptor, mejor ancho de banda, valor medio nulo), los datos a transmitir no son enviados como fueron generados, sino que el transmisor los codifica utilizando alguna codificación conveniente.
- **Sistema de transmisión:** Es el medio por el que se transmite la señal codificada. Por ej: un cable UTP, un cable coaxial, fibra óptica o el aire. Estos medios agregan ruido a la señal transmitida por varios motivos (ruido térmico, intermodulación, ruido impulsivo, etc).
- **Receptor:** Es el encargado de recibir la señal del medio de transmisión, distinguir entre la señal y el ruido y decodificar la misma de modo que pueda ser usada por el destino.
- **Destino:** Es quien utiliza los datos recibidos.

Detección de una Señal en Ruido

En teoría de señales, se define a la señal como una función que define cierta información sobre el comportamiento o atributos de un fenómeno. Esta definición teórica contempla posibles cambios introducidos a la señal, mediante la adición de otra señal también definida por una función. En la práctica,

las señales emitidas por transmisores no son recibidas por los receptores sino con alteraciones provocadas por el medio de transmisión. Esas modificaciones sobre la información original se conocen como ruido y es de sumo interés en la teoría de la Detección de Señales [1] que estudia mecanismos que permitan su cancelación.



La señal recibida por el receptor $y(n)$ será:

$$y(n) = x(n) + w(n) \text{ para } 0 \leq n < N$$

Donde $x(n)$ es la señal transmitida y será $x(n) = S_0(n)$ si se transmitió un “0” o $x(n) = S_1(n)$ si se transmitió un “1”. S_0 y S_1 indican las codificaciones utilizadas por el transmisor para enviar un 0 o un 1. Y $w(n)$ es el ruido adicionado por el sistema de transmisión. Para los efectos de este TP consideramos que el mismo se puede modelar como ruido blanco gaussiano de media nula.

Dado que el receptor no tendrá exactamente la señal $x(n)$ enviada por el transmisor, sino que tendrá la señal $y(n)$ que es la suma de la señal transmitida más el ruido inducido por el medio de transmisión, deberá generar un algoritmo que permita decidir, a partir de $y(n)$, si la señal transmitida $x(n)$ era S_1 o S_0 .

Para esto haciendo un análisis estadístico se realiza un “Test de hipótesis”[2], donde las hipótesis son H_0 : Se transmitió un 0 y H_1 : Se transmitió un 1.

Este Test si bien no nos asegura que no nos equivoquemos en la detección (o sea que creamos que se nos envió un 0 cuando en realidad fue un 1 y viceversa), sí se encarga de maximizar la probabilidad de que estemos tomando la decisión correcta.

Asumiendo que el ruido es blanco gaussiano se llega a la siguiente regla de decisión:

Sean, utilizando la notación vectorial tal que $y(n)$ para $0 \leq n < N$ sea

$$y = (y(0), y(1), \dots, y(N-1))$$

$$L(y) = \overline{y} (\overline{S_1 - S_0})^T$$

y

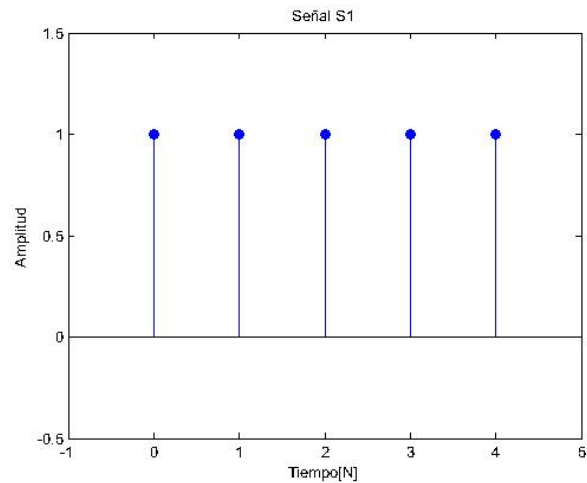
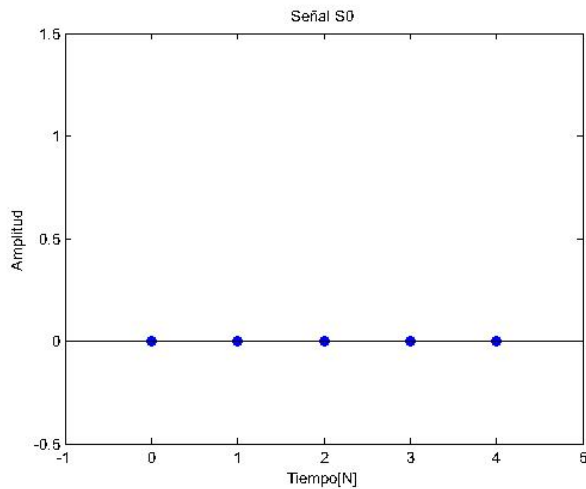
$$\gamma = \sigma^2 \ln\left(\frac{P(0)}{P(1)}\right) + \frac{1}{2}(S_1 S_1^T - S_0 S_0^T)$$

Donde σ^2 es la varianza del ruido, $P(1)$ y $P(0)$ son las probabilidades de que sean transmitidos un 1 y un 0 respectivamente.

Luego de ciertas comprobaciones matemáticas, se llega a la conclusión de que si $L(y)$ es mayor a γ , el dato transmitido fue un 1 y en caso contrario, fue un 0.

Ejemplo 1

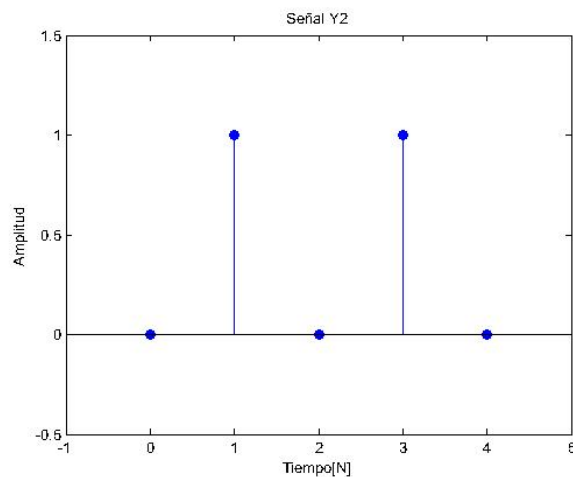
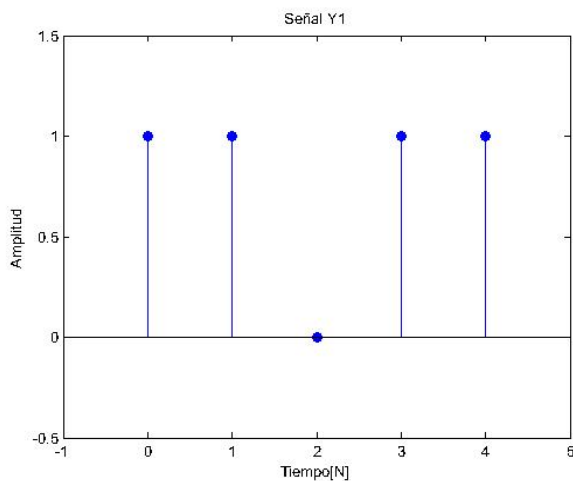
$N=5$, $S_0=(0,0,0,0,0)$, $S_1=(1,1,1,1,1)$, $P_0=0.5$, $P_1=0.5$, $\sigma^2 = 2$



El γ calculado será $\gamma = 2.5$

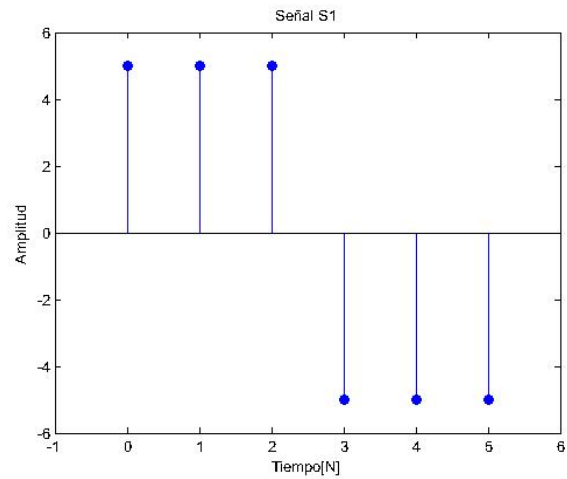
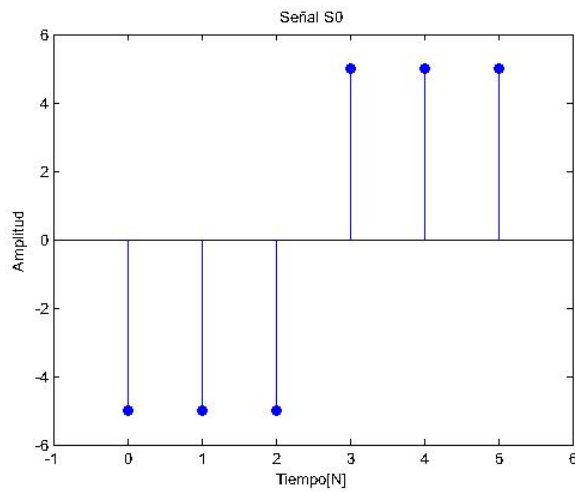
Si recibimos la señal $Y_1=(1,1,0,1,1)$, entonces el $L(Y_1)=4$ es mayor a γ y por lo tanto concluimos que la señal detectada es 1.

Por el contrario, si recibimos la señal $Y_2=(0,1,0,1,0)$, entonces el $L(Y_2)=2$ es menor a γ y por lo tanto concluimos que la señal detectada es 0.



Ejemplo 2

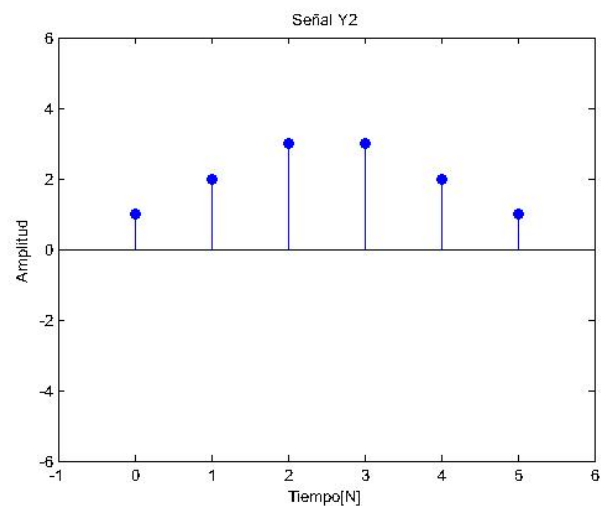
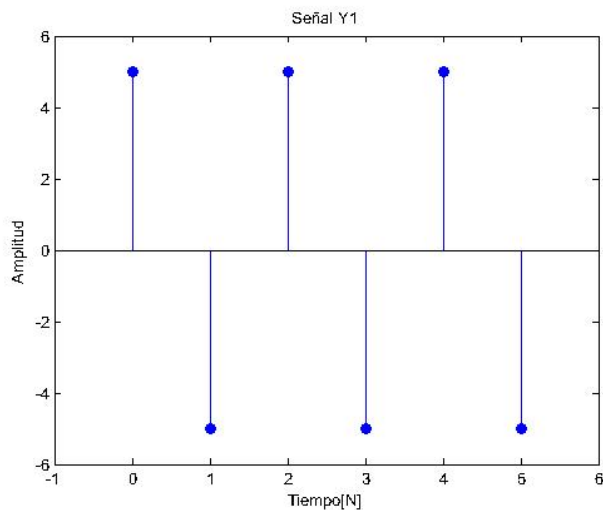
$N=6$, $S_0=(-1,-1,-1,1,1,1)$, $S_1=(1,1,1,-1,-1,-1)$, $P_0=0.4$, $P_1=0.6$, $\sigma^2 = 1$



El γ calculado será $\gamma = 0.4055$

Si recibimos la señal $Y1=(5,-5,5,-5,5,-5)$, entonces el $L(Y1)=100$ es mayor a γ y por lo tanto concluimos que la señal detectada es 1.

Por el contrario, si recibimos la señal $Y2=(1,2,3,3,2,1)$, entonces el $L(Y2)=0$ es menor a γ y por lo tanto concluimos que la señal detectada es 0.



Descripción

El presente trabajo consta de implementar el algoritmo de detección de señales expuesto anteriormente. A partir de señales recibidas por el detector y los parámetros indicados se deberá indicar cuál es la señal detectada.

Entrada

Los parámetros y las codificaciones vendrán en un archivo de entrada especificado o, en caso de omitirse, se hará por entrada estándar, con el siguiente formato:

```

<N>, <P(0)>, <P(1)>, < $\sigma^2$ >
<S0>
<S1>
<Y1>
<Y2>
...
<Yn>

```

Donde <N> indica la longitud de las señales S0 y S1, <P(0)> es la probabilidad de que se haya transmitido un 0, <P(1)> es la probabilidad de que se haya transmitido un 1 y < σ^2 > es la varianza del ruido.

<S0> y <S1> indican la codificación utilizada para transmitir 0s y 1s respectivamente. Serán una serie de números enteros entre -128 y 127 separados por comas (",").

<Yi> indica la señal recibida. La longitud de la misma puede asumirse como múltiplo exacto de la longitud de las señales S0 y S1. Al igual que en el caso anterior, se trata de una serie de números enteros entre -128 y 127 separados por comas (",").

Se podrá asumir que el archivo de entrada está bien formado y que la longitud máxima de cada línea es de 255 caracteres.

A continuación se presenta un ejemplo del formato de un archivo de entrada:

```

5,0.5,0.5,1
0,0,0,0,0
1,1,1,1,1
1,1,0,1,1,0,1,0,0,1
2,0,-1,0,0,2,1,1,0,1,1,1,1,0,0
0,0,1,0,0

```

Salida

La salida del sistema se realizará en un archivo único indicado por el usuario o, en caso de omitirse, se hará por salida estándar.

El formato de la misma será:

```

<SeñalDetectada1>
<SeñalDetectada2>
...
<SeñalDetectadaN>

```

Dónde <SeñalDetectada_i> será una serie de 0s y 1s separados por comas (","), correspondientes a la señal detectada a partir de la entrada Y_i.

Formato de Línea de Comandos

El formato de línea de comandos para ejecutar el programa será

```
./tp [-i entrada] [-o salida]
```

Parámetro -i: especifica el archivo de entrada. En caso de omitirse se deberá usar la entrada estándar.

Parámetro -o: especifica el archivo de salida. En caso de omitirse se deberá usar la salida estándar.

Códigos de Retorno

El programa deberá retornar:

- 0: Ejecución exitosa.
- 1: Parámetros erróneos. Se especificaron parámetros distintos a los detallados anteriormente.
- 2: El archivo de entrada o salida especificados no pueden ser accedidos.

Entrada y Salida Estándar

La entrada y salida estándar sólo son utilizadas si no se especifican los archivos correspondientes. El formato de los mismos es el ya especificado anteriormente.

Ejemplos de Ejecución

Para aclarar la secuencia de ejecución y los formatos de archivos, se expondrán los siguientes ejemplos:

Ejemplo 1

Sea el archivo de entrada entrada.txt:

```
5,0.5,0.5,1
0,0,0,0,0
1,1,1,1,1
1,1,0,1,1,0,1,0,0,1
2,0,-1,0,0,2,1,1,0,1,1,1,1,0,0
0,0,1,0,0
```

Entonces los parámetros son: $N=5$, $P(0)=0.5$, $P(1)=0.5$, $\sigma^2 = 1$, $S_0=(0,0,0,0,0)$, $S_1=(1,1,1,1,1)$

El resto de las líneas contienen las señales recibidas por el detector.

Entonces al ejecutar:

```
./tp -i entrada.txt -o salida.txt
```

Se deberá generar el archivo “salida.txt” con el siguiente contenido:

```
1,0
0,1,1
0
```

El código de retorno deberá ser 0.

Ejemplo 2

Sea el archivo de entrada entrada.txt:

```
6,0.6,0.4,3
2,2,2,-2,-2,-2
-2,-2,-2,2,2,2
2,0,-1,0,0,2,3,1,2,-2,4,-5,1,1,1,0,0,-3
3,1,2,-2,4,-5,-2,0,2,-2,0,2,3,4,5,-1,-2,-3
```

Entonces los parámetros son: $N=6$, $P(0)=0.6$, $P(1)=0.4$, $\sigma^2 = 3$, $S_0=(2,2,2,-2,-2,-2)$, $S_1=(-2,-2,-2,2,2,2)$

El resto de las líneas contienen las señales recibidas por el detector.

Entonces al ejecutar:

```
./tp -i entrada.txt
```

Se deberá imprimir por salida estándar el siguiente contenido:

```
1,0,0
0,0,0
```

El código de retorno deberá ser 0.

Restricciones

La siguiente es una lista de restricciones técnicas exigidas por el cliente:

1. El sistema debe desarrollarse en ISO C (C99).
2. Está prohibido el uso de variables globales.
3. El archivo de entrada deberá ser leído una única vez.
4. No se permitirá cargar todas las señales recibidas en memoria al mismo tiempo, deberán ser procesadas una a una.
5. No se podrá utilizar el stack para almacenar las señales recibidas, deberá utilizarse el heap.

Referencias

[1] Teoría de detección de señales: http://en.wikipedia.org/wiki/Detection_theory

[2] Test de Hipótesis: http://es.wikipedia.org/wiki/Contraste_de_hipótesis