



Tecnológico de Costa Rica
Escuela de Ingeniería en Electrónica
EL5805 Procesamiento Digital de Señales

M. Sc.-Ing. José Miguel Barboza Retana

Tarea 2

Estudiantes:

Gladys Arias Abarca 2013018896
Alexis Gavriel Gómez 2016085662
Cristian Rivera López 2013008650
Andrés Vargas Rivera 2015097412

Semestre 1
2019

Problema 1

Descripción:

El objetivo de este problema era mediante un método recursivo calcular la raíz n-ésima de un número real dado, esto implementado en una función que reciba como entradas dos números, a y b donde a es la raíz n-ésima del número real b.

Solución:

Para calcular la raíz n-ésima de un número se elige calcularla mediante el método de Newton-Raphson, para implementar este método se hace el siguiente análisis.

$$x = y^{(1/a)}$$

$$x^a - y = 0$$

Existe una función $f(x_n)$ talque $f(x_n) = x_n^a - y$ para la cual se puede buscar el cero.

El método Newton-Raphson dice que :

$$x(n+1) = x(n) - f(x_n)/F'(x_n)$$

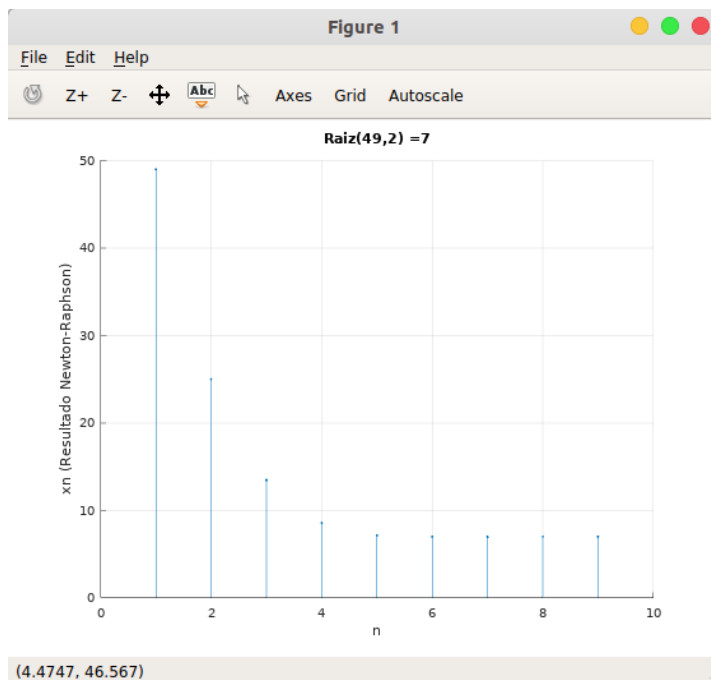
Por lo que la función resultante se itera sobre x para conseguir un valor muy cercano al esperado al final la ecuación del algoritmo implementado queda de la siguiente forma.

$$x(n+1) = x(n) - [x(n)^a - y]/[ax(n)^{(a-1)}]$$

El resultado de implementar lo anterior descrito da como resultado lo siguiente.

Resultados:

Ejemplo para la raíz cuadrada de 49.



Problema 2

Encargado: Andrés Vargas Rivera

Descripción: El problema 2 consiste en la elaboración de una función la cual esta encargada de generar la convolución de dos señales cualesquiera en tiempo discreto especificando tanto sus valores en cada muestra cómo el debido arreglo de las posiciones de la muestra en el tiempo. Se debe crear otra función la cual tenga como propósito generar el resultado de la correlación entre dos señales en tiempo discreto, para ello debe utilizar la función de convolución realizada anteriormente.

Solución: Para poder establecer la solución del problema se debe dividir en varias partes utilizando el concepto de divide y vencerás en pequeños problemas más simples y además potencialmente mejor optimizados. La solución general se divide en:

- Convolucionar dos señales que empiezan en la muestra 0: La razón de este primer problema inicial es que al conocer que ambas señales se encuentra en su primera muestra en el tiempo 0, el algoritmo de convolución puede discernir cuales multiplicaciones son innecesarias y van a resultar en 0 sin necesidad de extender originales, de forma que el algoritmo realiza la cantidad mínima de multiplicaciones posibles la cual en el caso de que ambas funciones inicien en la muestra 0 es: $\sum_{n=1}^m n$, donde m corresponde a el largo del arreglo de x más el largo del arreglo de h, menos 1.
- Desplazar las señales para que inicien en el tiempo 0: En esta función las señales se desplazan utilizando una función de shift que agrega 0 dependiendo del valor inicial de tiempo de la función la cual esta dado en el primer dato del arreglo que describe su arreglo correspondiente de posiciones.
- Invertir la señal h= Luego de ser desplazada la función h debe ser invertida para poder convolucionar con x y de esta forma obtener el resultado de la correlación.
- Desplazar nuevamente para obtener el valor de tiempo inicial de la correlación: El valor inicial de tiempo de la correlación esta dado por la cantidad de muestra de la respuesta al impulso, pero además hay que tomar en cuenta el desplazamiento que se realizó originalmente a la función por lo que tiene el efecto inverso en la respuesta de la correlación por lo que para obtener el punto de tiempo inicial real se utiliza, la cantidad de muestras de h(n) más 1, menos el valor del tiempo inicial de la función de impulso original.
- **Graficar la función:** Una vez encontrados los tiempos exactos para la correlación se realizan los correspondientes gráficos tanto de las funciones iniciales como del resultado de la correlación.

Parametros Comunes:

- X: Función x(n)
- H: Respuesta al impulso

- Nx: Arreglo posiciones de x
- Nh: Arreglo de muestras de h.

Algoritmos:

- plotCorrelation(x,nx,h,nh):Función realiza la correlación de dos señales dando como resultado un arreglo con los valores establecidos para la correlación con posición inicial de muestra absoluto. Así como generar los gráficos correspondientes a las dos señales de entrada y a la función de salida.
- correlation(x,nx,h,nh): Función realiza la correlación de dos señales dando como resultado un arreglo con los valores establecidos para la correlación con posición inicial de tiempo en la muestra 0 de forma relativa a su posición real, como lo haría la función del paquete signal de octave xcorr.
- convolution(x,nx,h,nh):Función realiza la convolución de dos señales dando como resultado un arreglo con los valores establecidos para la convolución con posición inicial de tiempo en la muestra 0 de forma relativa a su posición real, como lo haría la función de octave conv.

Resultados:

Caso 1: Funciones con muestra inicial desplazada

```
#Función que genera la correlación de dos funciones respuesta con muestra
inicial 0 desplazada
>> correlation([5,1,9,4,5],[-2,-1,0,1,2],[1,2,3],[-1,0,1])
ans =

    0    15    13    34    31    32    14     5     0     0

#Función que genera la correlación de dos funciones respuesta con muestra
inicial absoluta y genera el gráfico correspondiente.
>> plotCorrelation([5,1,9,4,5],[-2,-1,0,1,2],[1,2,3],[-1,0,1])
ans =

    15    13    34    31    32    14     5
```

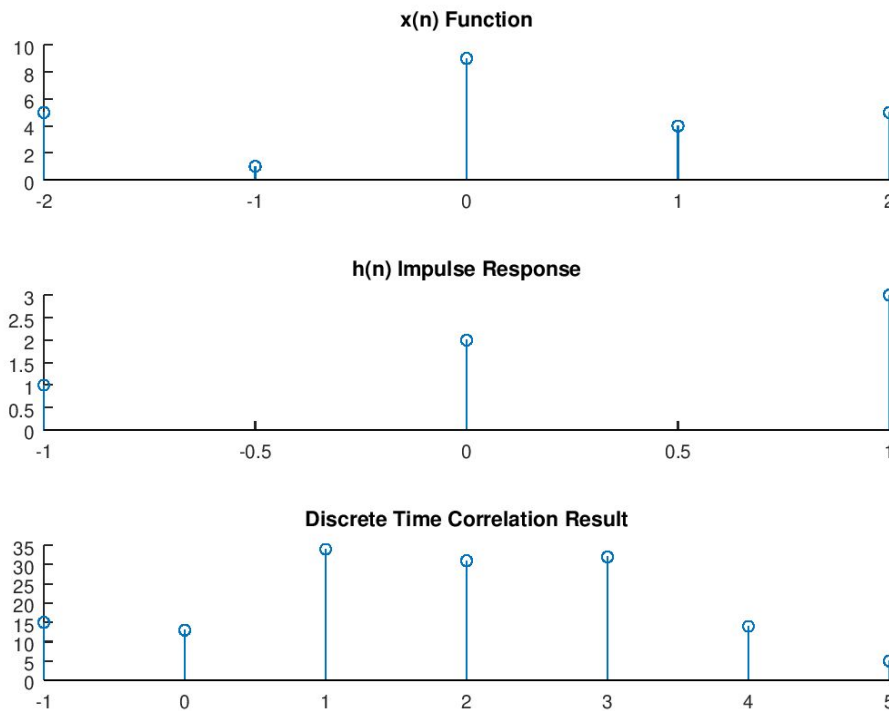


Gráfico 1. Correlación entre dos funciones caso 1

Caso 2: Funciones con muestra inicial en 0

```
#Función que genera la correlación de dos funciones respuesta con muestra
inicial 0 desplazada
>> correlation([5,1,9,4,5,6,3,5],[0,1,2,3,4,5,6,7],[1,2,3,6,3],[0,1,2,3,4])
ans =

    15    33    48    79    73    79    77    65    56    27    13     5
#Función que genera la correlación de dos funciones respuesta con muestra
inicial absoluta y genera el gráfico correspondiente.
>>
plotCorrelation([5,1,9,4,5,6,3,5],[0,1,2,3,4,5,6,7],[1,2,3,6,3],[0,1,2,3,4]
)
ans =

    15    33    48    79    73    79    77    65    56    27    13     5
```

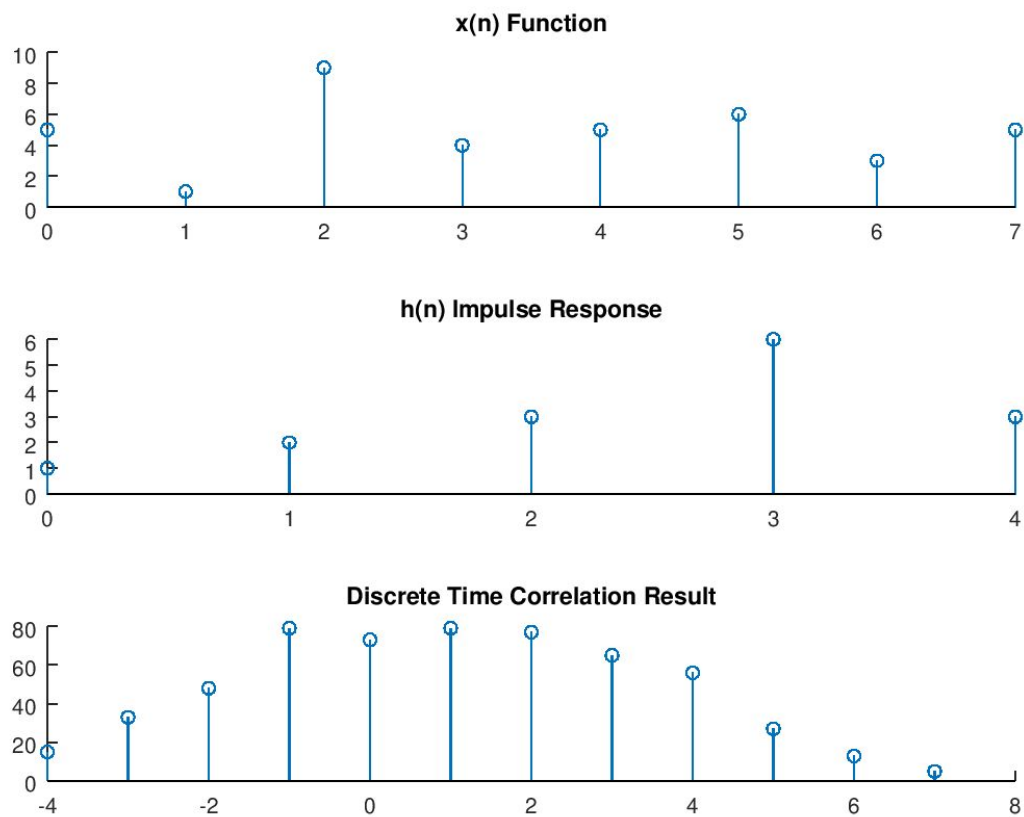


Gráfico 2. Correlación entre dos funciones caso 2

Caso 3: Largos de arreglos con compatibles entre x y nx o h y nh.

```
#Función que genera la correlación de dos funciones respuesta con muestra
inicial 0 desplazada
>> correlation([5,1,9],[0,1,2],[1,2],[0,1,2])
The length of arrays does not match:

#Función que genera la correlación de dos funciones respuesta con muestra
inicial absoluta y genera el gráfico correspondiente.
>> plotCorrelation([5,1,9],[0,1,2],[1,2],[0,1,2])
The length of arrays does not match:
```

Problema 3

Para ejecutar el problema 3:

Abrir la terminal desde la carpeta problema2

Ejecutar:

```
>>>octave-cli
```

```
>>>problema3.m
```

Respuesta teórica inciso (a):

Las señales $x(n)$ y $y(n)$ son parecidas.

Entre sus diferencias están:

$y(n)$ contiene ruido

Está retrasada en el tiempo por un factor D .

Al realizar la autocorrelación con un factor de $K = 0$ se obtiene el valor máximo de la autocorrelación.

Debido a ello, se puede inducir que si se realiza la correlación entre $x(n)$ y $y(n)$ en un factor K tal que $D = K$ (en donde se alinearían la señal del emisor y receptor) se encontraría un pico en el valor de correlación y la posición de este pico indicaría el retraso D .

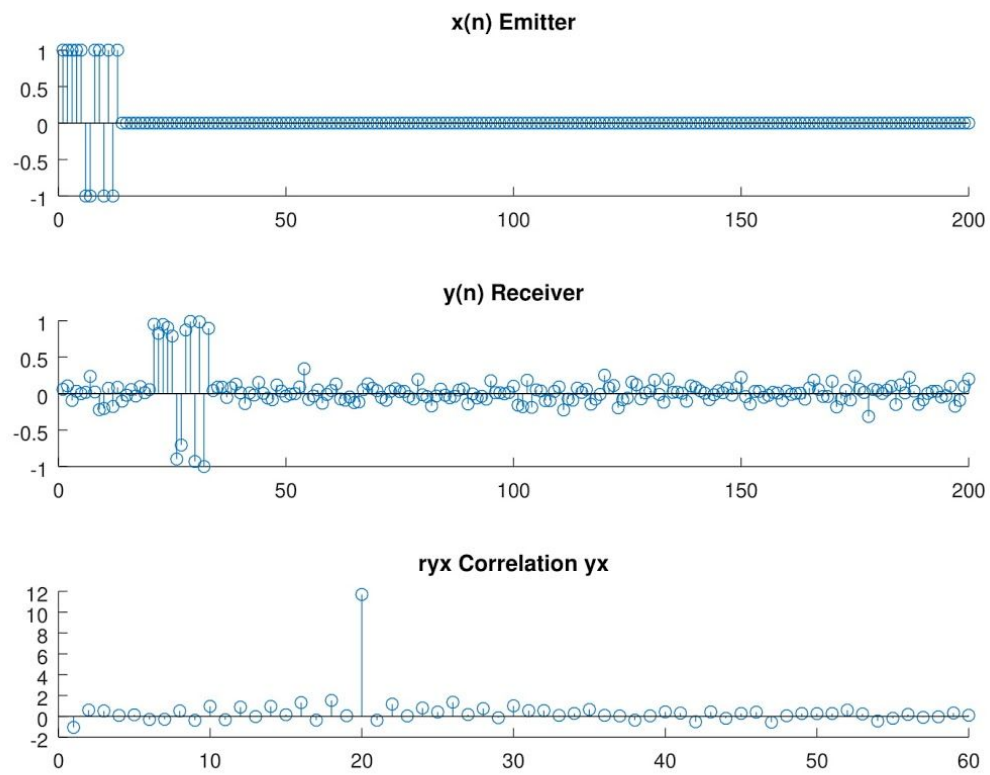
Algoritmo

Se toma la señal $x(n)$, se genera la señal $y(n)$ la cual está retrasada 20 muestras y posee ruido. Se utiliza la correlación para identificar el retraso, el cual se visualiza como un punto máximo en 20.

Para los incisos “e” y “f”, se genera la señal $x(n)$ a partir de un registro de desplazamiento con sumador-mod2. Y se repite el proceso anterior, generar la función $y(n)$ con retraso y ruido, realizar la correlación entre las funciones para encontrar el valor de retraso.

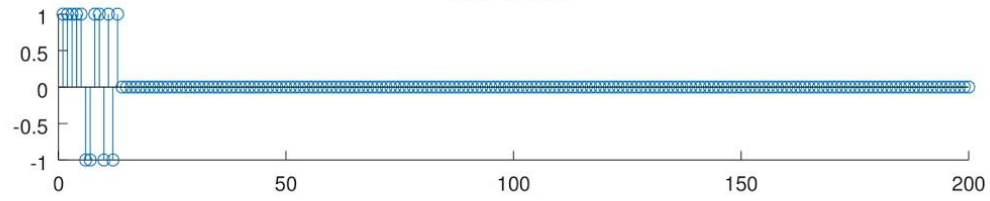
Resultados:

Varianza = 0.01

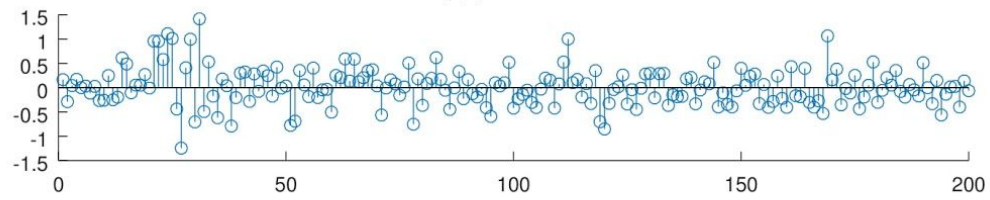


Varianza = 0.1

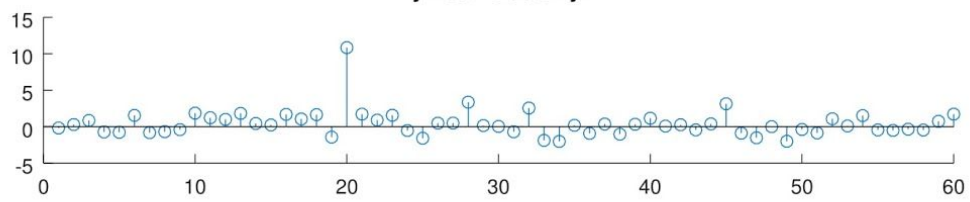
x(n) Emitter



y(n) Receiver

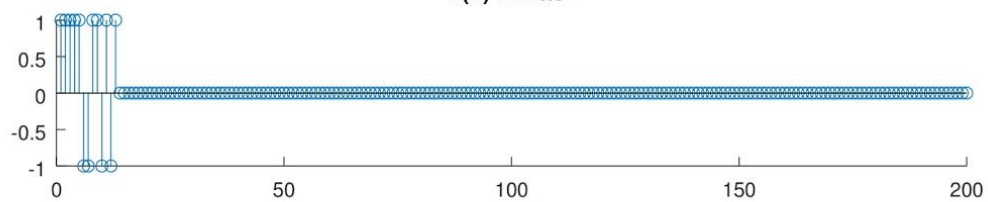


ryx Correlation yx

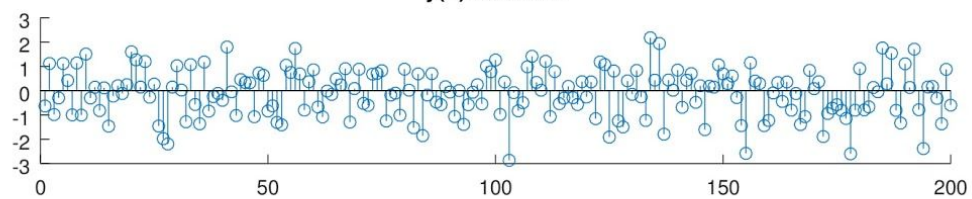


Varianza = 1

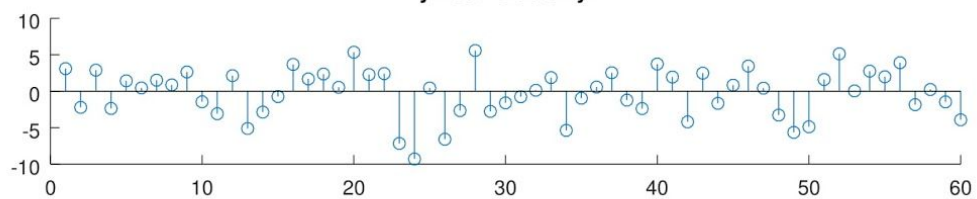
x(n) Emitter



y(n) Receiver

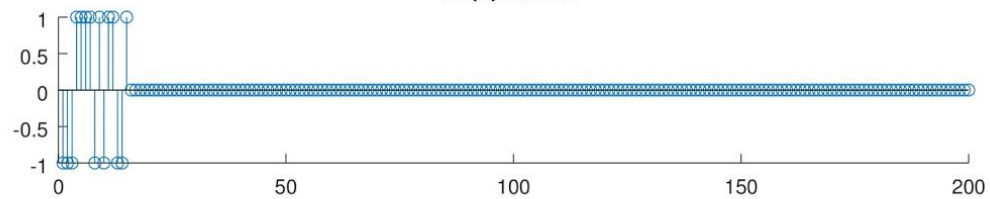


ryx Correlation yx

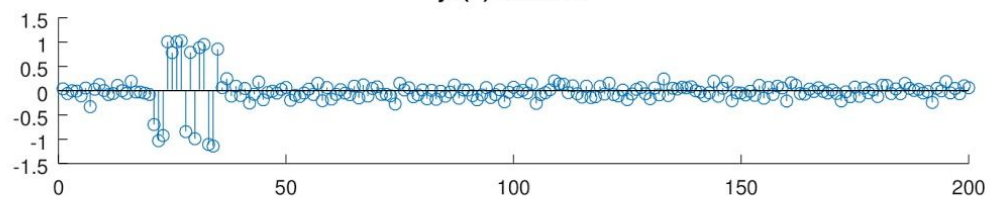


Registro de 4 etapas

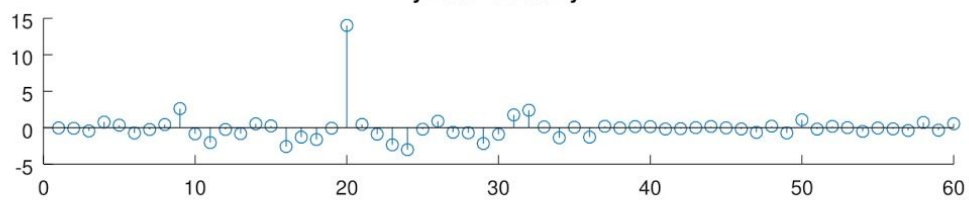
x2(n) Emitter



y2(n) Receiver



ryx Correlation yx



Registro de 7 etapas

