Laboratorio 1 Procesamiento Digital de Señales

Alejandro Ayala Gil y David Contreras Franco

I. Objetivos

1. Generar ruido gaussiano a una imagen.
2. Eliminar ruido de una señal a través del promedio de varias señales con ruido.
3. Percibir el comportamiento de la relación señal a ruido, cuando se aumentan el número de muestras.
4. Extraer de una base de datos, mediciones de radiación solar.
5. Implementar y evaluar la función de correlación de señales discretas.
6. Realizar el proceso de correlación y autocorrelación de señales discretas.

II. Materiales

1. Matlab
2. Base de datos de la Red Meteorológica Automatizada (RMA) de la industria de la caña de azúcar de Colombia.

III. Introducción

En este documente se busca manipular imágenes con el fin de probar que se puede hacer corrección de ruido en ellas, a través de la definición del promedio de señales, analizando como cambia la respuesta al aumentar el número de muestras. Además se busca realizar un código de correlación para vectores discretos, el cual permita determinar los valores cero de cada vector y su rango de salida. También se desea saber en los casos de prueba del algoritmo, de manera gráfica, cual seria el periodo de cambio climático de la variable de radiación solar en el Valle del Cauca.

IV. Procedimiento

1. Para este primer punto, se requiere como primer paso tomar una foto de los dos integrantes del grupo, para ello, se hizo uso de un celular xiaomi mi a2 con cámara frontal de 20MP. Después de la toma imagen a estudiar, se realiza un proceso de extracción en un computador y posteriormente ser analizada en el lenguaje de programación de matlab. A través de la rutina *imread*, que brinda matlab se logra cargar esta señal en formato RGB (3 matrices de 958x1280 con tipo de datos uint8). Posteriormente, se hace uso de la rutina de matlab *imshow* para hacer una visualización de la imagen. Como se puede ver en la siguiente figura 1.

Figura 1. Imagen tomada de los integrantes del grupo de laboratorio.

Como se buscar realizar un proceso de promedio de señal, es necesario del uso de la división en las componentes de nuestra señal. Sin embargo, al tener una señal con tipos de datos uint 8, la precisión se ve reducida a gran escala. Para esto es necesario hacer uso de la rutina de matlab *im2double* logrando asi convertir los datos de uint8 a double.

Posteriormente, se crea un arreglo de imágenes con ceros, con el fin de que este almacene cada uno de los promedios de las imágenes contaminadas. Y otro vector el cual se encargara de almacenar la relación señal a ruido de estas imágenes promediadas con respecto a la imagen original dependiendo del número de muestras. Se parte entonces de un ciclo, que terminara al completar el tamaño de muestras que haya en el vector de muestras y consistirá en genera una imagen contaminada con ruido gausseano utilizando la rutina de matlab *imnoise*. A su vez empezara otro ciclo, el cual llegara a su fin cuando llegue a un valor antes del número de muestras, esto con el fin de generar una señal de ruido y sumarla con el acumulado de las demás sumas de las señales con ruido. Después, se divide toda la suma, en el numero de muestras. Logrando asi satisfacer la siguiente ecuación de promedio:

$$\mu = \frac{1}{N}\sum_{i=0}^{N-1} Xi \tag{1}$$

Para después, utilizando la rutina de matlab *psnr* calcular la relación señal a ruido de la imagen original con la promediada y asi ir almacenando en los vectores de imágenes promediadas y relación señal a ruido los datos que sean generados.

Este proceso es repetido el número de veces que tenga el vector de muestras, en nuestro caso de pruebas del laboratorio se repite 5 veces con muestras de 5, 10, 20, 50 y 100 imágenes contaminadas. Al haber generado el vector de cada una de las relaciones señal a ruido, se procede a realizar una gráfica discreta del número de muestras contra la relación señal a ruido. De lo cual se obtuvo el siguiente resultado en la figura 2.

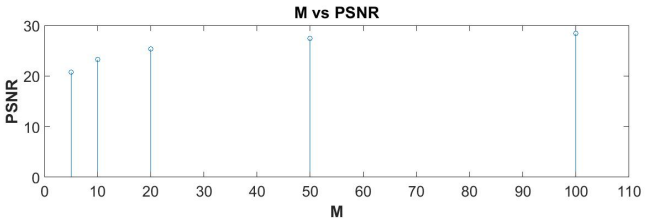

Figura 2. Gráfica de la relación señal a ruido dependiendo del numero de muestras.

Ahora con el vector de imágenes promediadas se procede a hacer una visualización de cada una de ellas, utilizando partiendo de poco número de muestras contaminadas a muchas muestras contaminadas. Por lo que para 5 imágenes contaminadas obtenemos el siguiente resultado de su promedio en la figura 3.

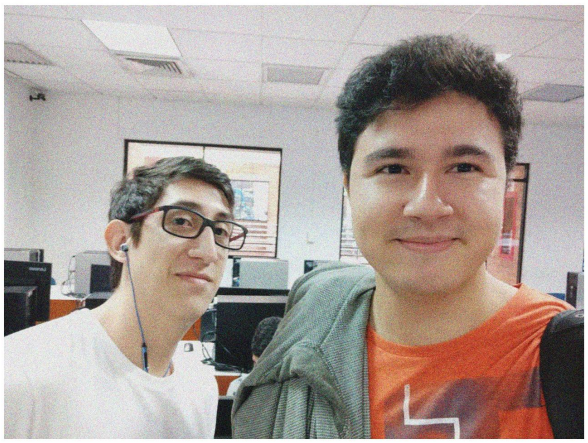

Figura 3. Promedio de 5 imágenes contaminadas con ruido gausseano.

Para 10 muestras contaminadas con ruido gausseano promediadas, se logra ver que el ruido aún persiste en la imagen resultante, como se puede ver a continuación en la figura 4.

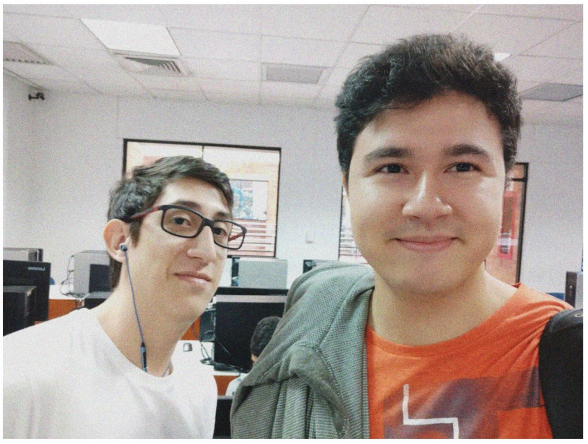

Figura 4. Promedio de 10 imágenes contaminadas con ruido gausseano.

Después se realiza el promedio de 20 muestras de imágenes contaminadas con ruido gausseano, se puede notar que el ruido disminuye en comparación a solo usar 10 muestras, como se puede ver en la siguiente figura 5.

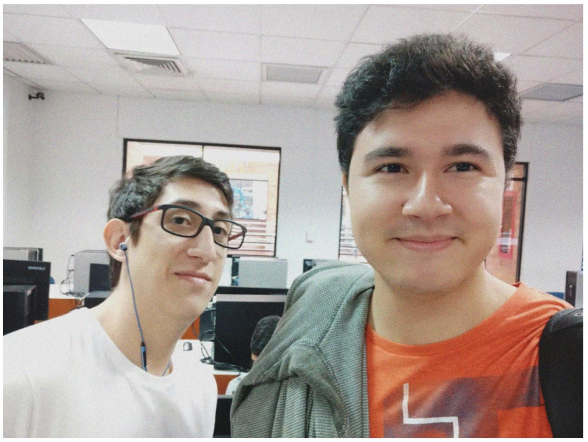

Figura 5. Promedio de 20 imágenes contaminadas con ruido gausseano.

Posterior a eso se promedian 50 imágenes contaminadas con ruido gausseano, se nota que este resultado se va acercando a lo que es nuestra imagen original, como se puede observar en la siguiente figura 6.

Figura 6. Promedio de 50 imágenes contaminadas con ruido gausseano.

Por ultimo se realiza el promedio de 100 imágenes con ruido gausseano, donde se obtiene un resultado similar a la imagen original, pero no varia mucho en comparación a si se toman 50 muestras. Como se puede ver en la figura 7.

Figura 7. Promedio de 100 imágenes contaminadas con ruido gausseano.

2. En lo que respecta al segundo punto del laboratorio, se requiere implementar un algoritmo el cual permita hacer la correlación de dos señales discretas. Este algoritmo debe recibir como parámetros dos vectores x y y, sus índices donde se determinará el valor 0 de cada uno de ellos y el rango en que estará definido la respuesta. Partiendo de lo anterior, el índice determinará cuántos valores tendremos en la parte positiva y cuántos en la parte negativa de cada vector. Sin embargo para lograr hacer una correlación es necesario que ambos vectores tengan el mismo tamaño. Por lo que se busca primero rellenar con ceros tanto izquierda, como derecha del vector cada vez que sea necesario.

   El proceso anterior en el software se realiza generando 4 vectores, los cuales se encargan de extraer la parte del vector desde la primera posición hasta el índice que se denominaría como la parte inferior y lo restante del vector seria la parte superior de este, para cada uno de los vectores x y y. Luego se procede a comparar los tamaños de las partes inferiores y las partes superiores, para asi rellenar con ceros un nuevo vector que logre x y y que haga que ambos sean del mismo tamaño. Para rellenarlo con ceros se hace uso de la rutina de matlab *zeros*, la cual agrega número de ceros dependiendo el parámetro que tenga.

   Después se define el tamaño del vector resultante como dos veces el tamaño de uno de los vectores creados y con esto se define el vector a retornar llenándolo de ceros, usando la rutina *zeros*. Posterior a esto se desplaza el vector x, agregándole ceros a la izquierda y a la derecha del tamaño del mismo vector. Seguido a esto se itera lo que seria el desplazamiento de este vector tomándolo desde su primer elemento de la izquierda y desplazándolo uno a uno hacia la derecha, haciéndole en cada iteración la multiplicación con el vector y de cada una de sus componentes, y a su vez realizando la sumatoria de todo este vector, agregando el resultado en la posición del vector normal en la que este el iterador. Buscando la forma de satisfacer la siguiente ecuación mediante software:

$$rnorm(l) = \sum_{n=-\infty}^{\infty} x(n+l) * y(n) \quad l = \pm 1, \pm 2... \quad (2)$$

   Luego para definir nuestra respuesta, hacemos uso del valor de rang, el cual permite seleccionar el rango en el que se desarrollara la respuesta. En el código toma el resultado de rnorm desde la diferencia entre el tamaño de un vector y el rang, hasta la suma entre el mismo tamaño del vector con el rang. De todo esto se busca retornar tanto este vecotr rnorm seleccionado a partir del rang, y un vector que se caractiza desde los valores negativos de rang hasta los valores positivos, para asi representar el lag dentro de la correlación que ofrece matlab con la rutina *xcorr*.

   Con el código de correlación ya definido, se parte a hacer las distintas pruebas solicitadas en la guía. Para el primer caso se procede a hacer la correlación entre dos vectores, los cuales tienen su punto de índice en 8 y 7 respectivamente. Al hacer la ejecución de este código en el mayor rango de respuesta se obtiene lo que se observa en la figura 8 Además, en la gráfica podemos ver el valor que toma el índice de la correlación, para esta prueba el valor es de 0.1013.

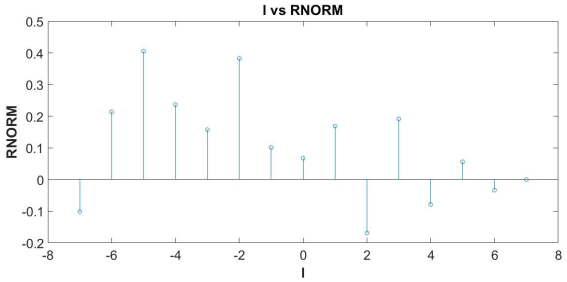

Figura 8. Grafica de l vs RNORM para la primera prueba del punto

En lo que respecta la prueba de que el código funciona, se decide comparar el resultado brindado por la correlación de la rutina *xcorr* de matlab, con los resultados de nuestra correlación. Para poder realizar este proceso, se debe completar con ceros el vector que tendrá la función xcorr, de tal manera de que tengan indicies iguales al vector de nuestra correlación, dichos resultados se observan en la figura 9 donde se observa la similitud entre ambos resultados.

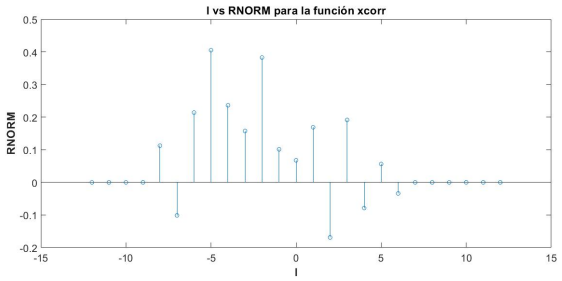

Figura 9. Grafica de l vs RNORM para la primera prueba del punto con uso de la función xcorr

Ahora para el segundo caso de pruebas se extraen los datos de radiación solar, ya normalizados por la siguiente ecuación:

$$Zi = \frac{Xi - \mu}{\sigma} \tag{3}$$

Permitiendo asi tener una correlación más visible, al tener un comportamiento normal de la variable. Los datos son extraídos utilizando la rutina de matlab *csvread*, teniendo en cuenta las filas y columnas a extraer, de un archivo llamado *cana.cvs* brindado por el profesor. Al realizar el proceso de la auto correlación de estos datos ya normalizados, haciendo uso de un índice 1 para ambos vectores y un rang de 200, se obtiene los resultados de la figura 10.

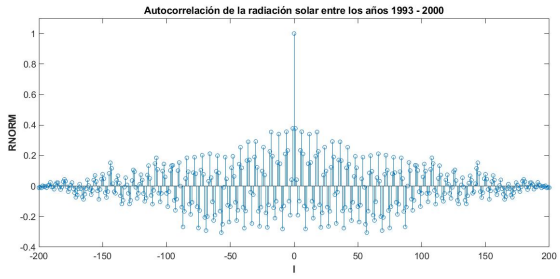

Figura 10. Autocorrelación de la radiación solar entre los años 1993 - 2000.

Ahora para comparar si nuestro procedimiento fue satisfactorio, hacemos uso de la rutina de matlab *xcorr*, con el fin de comparar si los resultados concuerdan. Definiendo la autocorrelación de la variable de radiación y usando la forma normalizada de la respuesta, se puede implementar una gráfica tal como se ve en la figura 11. El cual nos muestra la similitud en todos los datos de ambos procesos.

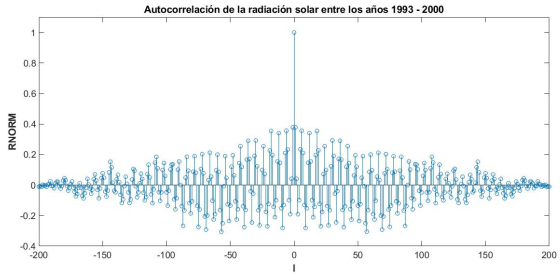

Figura 11. Autocorrelación de la radiación solar entre los años 1993 - 2000 haciendo uso de la función xcorr.

Dados todos estos datos brindados por la autocorrelación, se vuelve a realizar el mismo proceso pero con una salida más limitada, para si lograr determinar la relación del periodo de tiempo con el clima que se presenta en la región. Esto se puede observar en la figura 12.

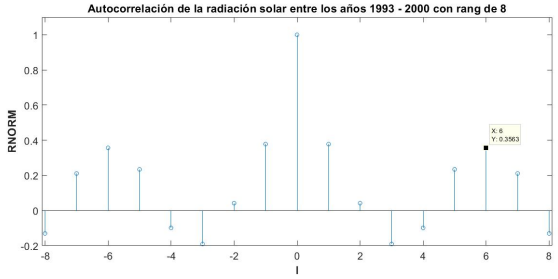

Figura 12. Autocorrelación de la radiación solar entre los años 1993 - 2000, con rang de 8

Teniendo en cuenta lo visto en clase de como determinar el periodo, en señales periódicas a partir de la correlación, se puede observar en la figura 12 que tiene un máximo global en 0 y la diferencia entre sus máximos locales se dan cada 6 periodos, lo que determinaría entonces que cada 6 meses la variable de radiación alcanza un punto máximo.

V. Conclusiones

1. Es buena práctica tener en cuenta el tipo de dato en el que se extraen los archivos, debido a que dependiendo del tipo de dato puede cambiar la precisión de las operaciones a realizar.
2. A medida que se van aumentando el número de muestras, se puede lograr una mejor corrección de errores generados por ruido en una imagen.
3. El cambio de la relación señal a ruido, no se ve tan significativo cuando se toman demasiadas muestras.
4. El código de correlación que se implemento en el laboratorio, permite hacer uso de índices en los vectores de entrada, esto con el fin de determinar el punto 0 de cada uno de los vectores.
5. Al tener un parámetro rang se logra que en muestras demasiado grandes, se puedan observar regiones pequeñas centralizadas. Como se usó para determinar el periodo de radiación.
6. En el laboratorio se puedo hacer uso de la aplicación de conceptos vistos en clase, tales como el promedio de señales para la corrección de ruido y la correlación. Esto con el de llevarlos a escenarios que se pueden dar en la vida practica, tales como la corrección de imagen de una cámara o para predecir en que épocas se obtienen niveles de radiación mas elevados.

V-A. Temas que cada integrante trabajo

El estudiante Alejandro Ayala se encargo de hacer el código de la correlación en cual, buscó la manera de cumplir con todo lo requerido del laboratorio. Verificó su funcionamiento mediante el uso de la rutina *xcorr* de matlab y determinó la relación del periodo con el clima en el Valle del Cauca.

Mientras que el estudiante David Contreras se encargo de hacer la toma de la imagen a trabajar, la extracción de la misma en matlab, la corrección del tipo de dato y el código encargado de hacer tanto el promedio de cada una de las imágenes, como el código que para poder graficar la relación señal a ruido dependiendo el número de muestras.

VI. Observaciones

1. Se puede observar que a medida que vamos aumentando el número de imágenes se corrige más ruido. Sin embargo se puede ver, que al usar una cantidad de muestras demasiado grandes, no se obtienen cambios supremamente significativos en la salida.
2. En la correlación diseñada en el laboratorio, se da la opción no solo de obtener el resultado normalizado, sino además poder determinar tanto los índices de cada vector, como el rango de salida de la correlación. Además de esto se puedo determinar con la ayuda de este algoritmo el periodo en el que la radiación solar alcanza sus puntos máximos en el Valle del Cauca.

VII. Literatura consultada

- ”(Not recommended) Read comma-separated value (CSV) file - MATLAB csvread”, Mathworks.com, 2020. [Online]. Available: https://www.mathworks.com/help/matlab/ref/csvread.html. [Accessed: 17- Feb- 2020].
- Read image from graphics file - MATLAB imread”, Mathworks.com, 2020. [Online]. Available: https://www.mathworks.com/help/matlab/ref/imread.html. [Accessed: 17- Feb- 2020].
- Çonvert image to double precision - MATLAB, im2double”, Mathworks.com, 2020. [Online]. Available: https://www.mathworks.com/help/matlab/ref/im2double.html [Accessed: 17- Feb- 2020].
- Çross-correlation - MATLAB xcorr”, Mathworks.com, 2020. [Online]. Available: https://www.mathworks.com/help/matlab/ref/xcorr.html [Accessed: 17- Feb- 2020].
- Çreate array of all zeros - MATLAB zeros”, Mathworks.com, 2020. [Online]. Available: https://www.mathworks.com/help/matlab/ref/zeros.html [Accessed: 17- Feb- 2020].
- ”Sum of array elements - MATLAB sum”, Mathworks.com, 2020. [Online]. Available: https://www.mathworks.com/help/matlab/ref/sum.html [Accessed: 17- Feb- 2020].
- .ªdd noise to image - MATLAB imnoise”, Mathworks.com, 2020. [Online]. Available: https://www.mathworks.com/help/images/ref/imnoise.html [Accessed: 17- Feb- 2020].
- Plot discrete sequence data - MATLAB stem”, Mathworks.com, 2020. [Online]. Available: https://www.mathworks.com/help/matlab/ref/stem.html [Accessed: 17- Feb- 2020].
- Apuntes de clase de procesamiento digital de señales. Ingeniería Electrónica y ciencias de la computación. Pontificia Universidad Javeriana Cali. Febrero 2020.