

ANÁLISIS DE SEÑAL CODIFICADA

Leandro Caloguerea Farias, Diego Rojas Asenjo

Abstract—Abstract content goes here

Resumen— Se nos presenta como desafío el problema de analizar un archivo con 23000 muestras discretas de una señal interceptada con ruido incluido, la cual contiene un mensaje codificado. Se aplican las metodologías vistas en el curso de análisis de sistemas lineales para el desarrollo de este documento, el cual consiste en procesar el documento de entrada, descomponerlo, aplicar la transformada rápida de Fourier para su mejor manipulación a modo de poder reconstruir la señal original e identificar las frecuencias a lo largo de la señal de 23 segundos. Se nos ha entregado una tabla de equivalencias con las que culminaremos por asociar la señal a un mensaje final.

I - OBJETIVOS DEL TRABAJO PRACTICO

- 1) La principal tarea es poder transcribir una señal discretizada, obteniendo valores de amplitud de señal(espectro), relacionarlos con la frecuencia normalizada que representan las muestras, para finalmente comparar el cifrado y decodificar la señal muestreada.
- 2) Saber identificar cuando una señal posee perturbaciones al momento de normalizar una muestra.
- 3) Comprender el teorema de Nyquist[1], que define la frecuencia de corte mínimo para muestreo.
- 4) Saber identificar qué comprendemos por señal codificada a base de su frecuencia normalizada.

- 5) Razonar y poder comprender el significado de una muestra normalizada, para un problema de codificación por frecuencia.

II - METODOLOGIA DE TRABAJO

El desarrollo del trabajo fue realizado bajo las siguientes arquitecturas y tecnologías:

- Sistema operativo de desarrollo basados en Linux
- Lenguaje de programación Python V2.7
- Librerías y dependencias
 - numpy
 - scipy para el proceso de funciones matemáticas como FFT
 - matplotlib para el proceso y salida de gráficos
- Señal discretizada en archivo de texto como entrada a procesar en nuestro programa

Para el procesamiento matemático nos apoyamos en la transformada rápida de Fourier (FFT en inglés por sus siglas *Fast Fourier Transform*). Procederemos a describir dicho algoritmo:

La transformada rápida de Fourier FFT es un algoritmo que reduce el tiempo de cálculo de n^2 pasos a $n \log 2(n)$. El único requisito es que el número de puntos en la serie tiene que ser una potencia de 2 (2^n puntos), por ejemplo 32, 1024, 4096, etc.

La DFT de una secuencia $x[n]$ es:

Una primera aproximación al cálculo de la DFT requeriría la suma compleja de N multiplicaciones complejas para cada una de las salidas. En total, N^2 multiplicaciones complejas y N^2 sumas complejas para realizar una DFT de N puntos.

• Leandro Caloguerea Farias is with Austral University of Chile

• Diego Rojas Asenjo is with Austral University of Chile.

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk} \quad k = 0, 1, \dots, N-1$$

donde $W_N = e^{-j2\pi/N}$

Figure 1. DFT de una secuencia $x[n]$

Lo que consigue el algoritmo FFT es simplificar enormemente el cálculo de la DFT introduciendo "atajos" matemáticos para reducir drásticamente el número de operaciones.

La optimización del proceso de cálculo de la DTF está basada en las siguientes ideas:

- Simetría y Periodicidad de los términos W_N

$$\begin{aligned} W_N^{n+N} &= W_N^n & W_N^{Nk} &= 1 \\ W_N^{n+N/2} &= -W_N^n & W_N^2 &= W_{N/2} \end{aligned}$$

Figure 2. Simetría y periodicidad de los términos W_N

Se escoge el valor de N de forma que $N = r^m$, Al factor r se le denomina radix y el valor más habitual es 2, de forma que $N = 2^m$ y el algoritmo se denomina FFT radix-2

Dividimos la secuencia de datos de entrada $x[n]$ en dos grupos, uno de índices par y el otro de índices impar. Con estas sub-secuencias se realiza la DFT de $N/2$ puntos.

Finalmente:

$$\begin{aligned} X[k] &= \sum_{n=0}^{N/2-1} x[2n] W_N^{2nk} + \sum_{n=0}^{N/2-1} x[2n+1] W_N^{(2n+1)k} & \text{Sustituimos} \\ &= \sum_{n=0}^{N/2-1} x[2n] W_N^{2nk} + W_N^k \sum_{n=0}^{N/2-1} x[2n+1] W_N^{2nk} & x_1[n] = x[2n] \\ &= \sum_{n=0}^{N/2-1} x_1[n] W_{N/2}^{nk} + W_N^k \sum_{n=0}^{N/2-1} x_2[n] W_{N/2}^{nk} = Y[k] + W_N^k Z[k] & x_2[n] = x[2n+1] \\ & \text{para } k = 0, 1, \dots, N-1 & W_N^{2nk} = W_{N/2}^{nk} \end{aligned}$$

Figure 3. Algoritmo FFT

Una vez determinados la arquitectura y el apoyo matemático-teórico se da paso al ingreso de la señal para su procesamiento, la separamos por número de muestras según el ratio de muestreo asignado (1000Hz) por segundo, la cual generará 23 vectores de 1000 muestras a

analizar c/u. Cada vector posee un trazo de señal la cual será procesada en la FFT, donde nos mostrará un "peak" relacionado con la frecuencia a la que circunda en 1s. Ahora, una vez que obtenemos ese "peak" ligado a una frecuencia por la FFT, debemos capturar ese "peak" resultante de la muestra mediante ventaneos, para luego una vez obtenido el "peak" de la ventana, comparamos según la escala de frecuencias determinada por símbolo, y habremos codificado la señal en 1s en relación al carácter. Para lo cual extrapolamos hacia los 23s. restantes en codificar la señal completa

caracter	codificación
A	sinusoide de 10Hz
B	sinusoide de 20Hz
C	sinusoide de 30Hz
.	.
.	.
.	.
Z	sinusoide de 270Hz
0	sinusoide de 280Hz
1	sinusoide de 290Hz
.	.
.	.
.	.
9	sinusoide de 370Hz

Figure 4. Codificación A-Z 0-9

III - ACTIVIDADES REALIZADAS

Como entendemos cuando una señal es muestreada a 1000Hz, significa que la señal es transmitida a 500Hz, debido a que por la teoría de Nyquist $W_s > 2 * W_m[1]$, la frecuencia mínima de "muestreo" debe ser el doble de la "transmitida", para lograr tener buenos resultados al momento de capturar los "peak" en la señal, evitando desfases o inconsistencias al momento del ventaneo de la señal normalizada. Para saber cuántas normalizaciones obtendremos, subdividimos la señal en muestras de 1000 datos con periodos de 0.001 intervalos de tiempo, para establecer el dominio de las muestras a graficar en la FFT entre [0,1],

asumiendo que la señal fue muestreada a una frecuencia de 1000Hz. En base de los datos iniciales, solo basto con pasar cada trozo de muestras por la FFT, para que nos pueda entregar el espectro normalizado (Rojo) de la señal (Azul) muestreada en 1 segundo (figura [5]), entregandonos los "peaks" en una frecuencia determinada, para un arreglo normalizado de muestras determinada.(Figura [6])

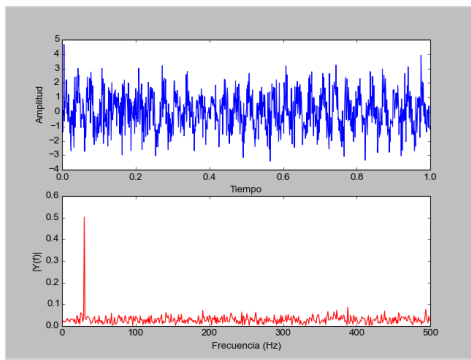


Figure 5. Señal de muestra con 1000 datos en 1 segundo, junto con su espectro normalizado (FTT)

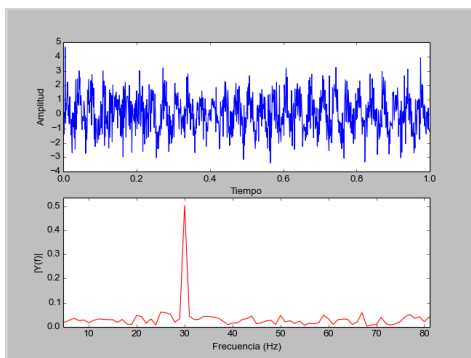


Figure 6. Misma señal pero con una ampliación al "peak" de su frecuencia resultante (30[Hz])

Ahora, que podemos entender según el resultado de la muestra obtenida: "en el segundo 'x', pasa una información 'y' en la frecuencia 'z'Hz" lo que nos indica que la información 'y' pasa a una frecuencia 'z' de ciclos por segundo.

IV - DIAGRAMA DE BLOQUES PARA DECODIFICACION DE MENSAJES

Esta sección debe presentar un diagrama de bloques de cada uno de los pasos seguidos

para la decodificación del mensaje. Con su explicación de cada uno de los bloques y su sustento.

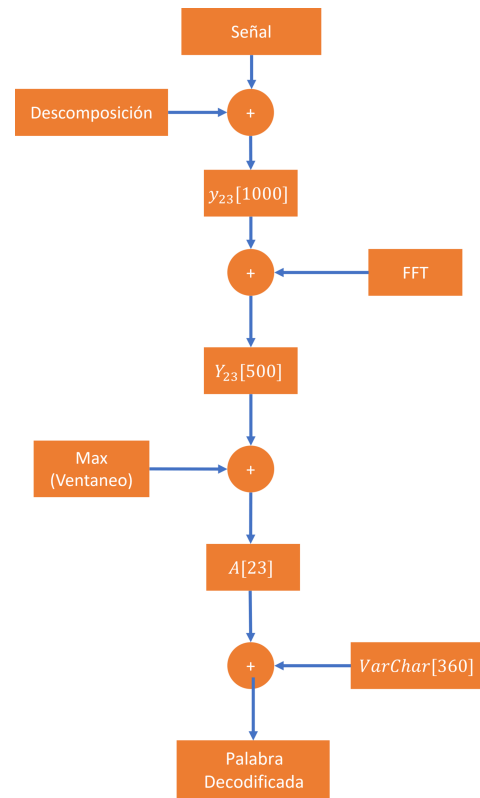


Figure 7. Diagrama de bloques

Señal

La señal de origen es un archivo de texto con 23000 muestras discretas en su interior, las que representan la señal sin codificar junto al ruido que satura su contenido.

Sumador

Segmento del diagrama de bloques en el cual interactúan bajo el operador suma los elementos unidos a él mediante su respectiva flecha.

Descomposición

Se descompone la señal en tramos por según el ratio de muestreo asignado de 1000[Hz]

y(23)[1000]

Arreglo de señal con 1000 muestras por segundo.

FFT

Algoritmo normalizador del arreglo de muestras

Max (Ventaneo)

Para la captura del "peak" de la señal normalizada usamos una ventana Hamming[2], pero como tenemos una señal limpia solo calculamos el maximo valor obtenido en la amplitud de la señal.

A[23]

Vector de "peaks" resultante tras la captura por segundo.

Y(23)[500]

Vector resultante de la FFT de con 500 muestras normalizadas por segundo.

VarChar[360]

Diccionario comparador de los "peak's" sobre A[23] la cual cruza caracteres asociados a frecuencias de hasta 360Hz, en intervalos de 10Hz los cuales comprenden su caracter equivalente de la A-Z 0-9.

Palabra decodificada

Una vez asociado cada "peak" de frecuencia a su codificacion respectiva se obtiene la palabra decodificada que entrega el mensaje final.

V - DECODIFICACION DE LA SENAL

Para el decodificado de la señal, obtuvimos las normalizaciones de los 23s. en un arreglo de la misma longitud para las muestras, la cual nos permitio trabajar de manera controlada los ventaneos correspondientes por cada elemento de muestras. Para estos casos, como la señal no presenta problemas de perturbaciones anexas, indicando que para cada segundo se vieron normalizaciones "limpias" que impidan capturar los "peak" (Figura [8], solo fue necesario encontrar el maximo valor de amplitud del espectro por arreglo de elementos, para determinar que frecuencia es la que buscamos

en determinada set de muestras. Lo cual al guardar cada "peak" de frecuencia en un arreglo, solo basto con comparar con nuestra escala de frecuencia-caracter, que nos permitio decodificar la señal.

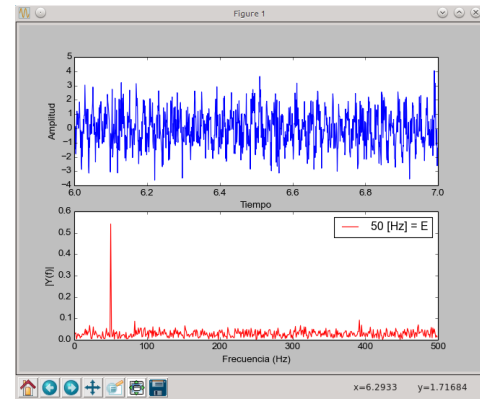


Figure 8. Señal en el segundo 7 con su normalizacion respectiva y decodificada según su frecuencia

VI - RESULTADOS Y DISCUSIONES

Según los resultados ya vistos, y los procedimientos realizados, también pudimos ver que cambiando el ratio de muestreo, ya sea disminuyendo o aumentando, el error que proporcionaba al analizar la señal era evidente, ya que no nos daba mucha información de cómo abordar los "peak", y capturarlos, debido que la irregularidad de la normalizacion definia intervalos muy grandes de muestreo para su precision al espectro que deseamos buscar según su frecuencia. (Figura [9])

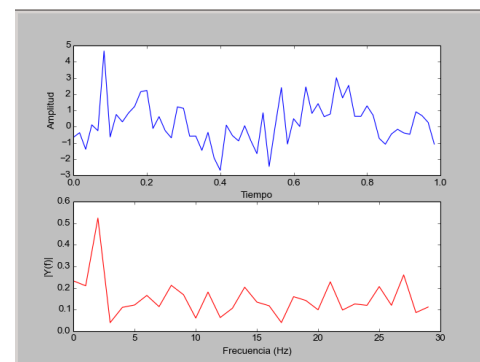


Figure 9. Espectro normalizado a un nivel de 60 muestras por segundo

Lo que nos daba una pérdida de información importante al momento de obtener resultados, lo que es crucial antes es el saber la frecuencia con la que se emite, para saber cuánto es el mínimo a muestrear la señal.

Por lo demás nuestro trabajo enfocado es determinar cuál era la señal que estaba codificada, por lo que al procesarla con los métodos usados obtuvimos nuestra frase:

"COSTANERACENTER3NOV1300"

VII - BIBLIOGRAFIA

[1] Oppenheim Señales y Sistemas.

[2] Documentación asignatura Análisis de sistemas lineales INFO183.