

UNIVERSIDAD NACIONAL DE TUCUMÁN

Facultad de Ciencias Exactas y Tecnología



Señales Eléctricas

TEMA: Modulación y demodulación en banda estrecha para una señal de audio

ALUMNO: Uberti, Ulises Leandro

AÑO: 2025

INTRODUCCIÓN:

Se tiene como objetivo la programación de un sistema de modulación y demodulación en banda estrecha para transmitir una señal de audio, entre las especificaciones se requiere que la señal de audio sea un mensaje de voz de corta duración, la modulación sea AM o digital simple, la portadora no supere los 10[kHz], y que el archivo de la señal modulada pueda escucharse como un archivo de audio, este archivo debe ser el punto de partida del receptor.

Se optó por desarrollar el sistema con una señal de audio en un archivo del tipo .wav, utilizar modulación OOK, trabajar con una portadora de 5[kHz] y demodular el archivo que contiene a la señal modulada.

HERRAMIENTAS:

Editor de código

- Visual Studio Code

Lenguaje

- Python

Archivos del tipo

- .wav
- .py
- .ipynb*

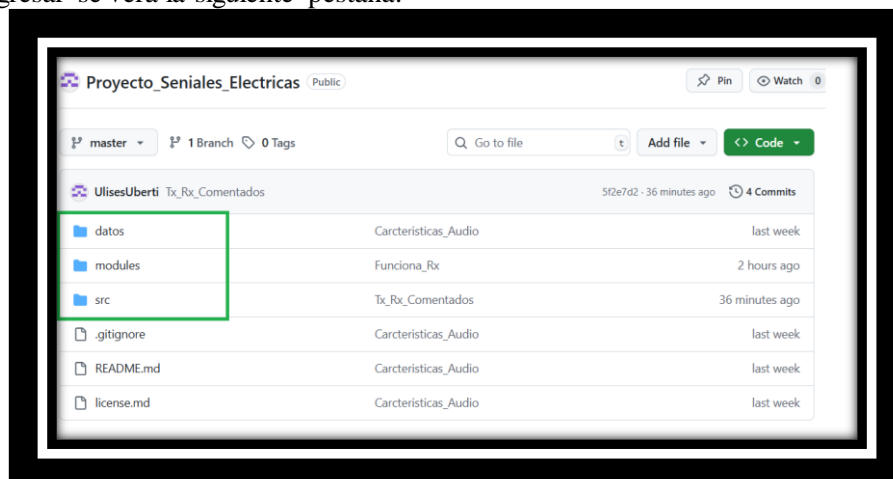
*Los archivos fundamentales se trabajan en Jupyter Notebook por lo que se requiere un entorno que los permita.

ACCESO A CARPETA DE TRABAJO:

- 1- Ingresar al repositorio en 'github' a través del siguiente link:

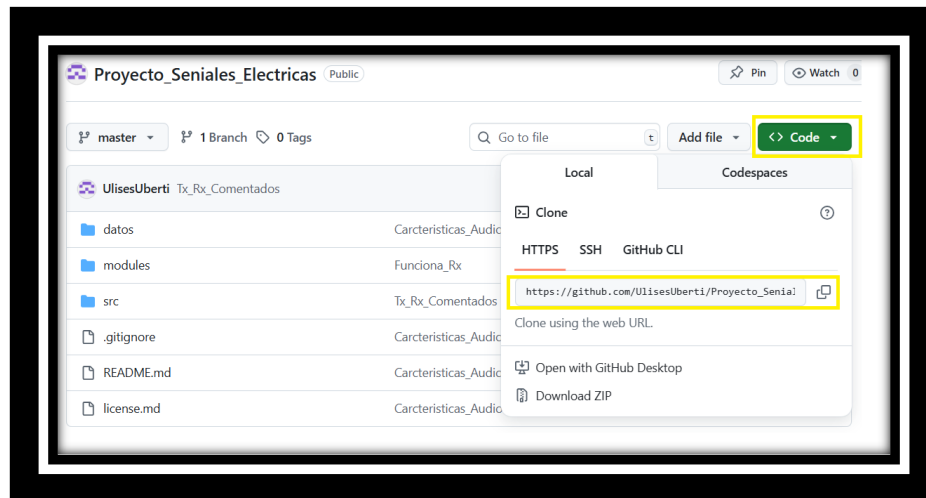
https://github.com/UlisesUberti/Proyecto_Seniales_Electricas.git

- 2- Al ingresar se verá la siguiente pestaña:



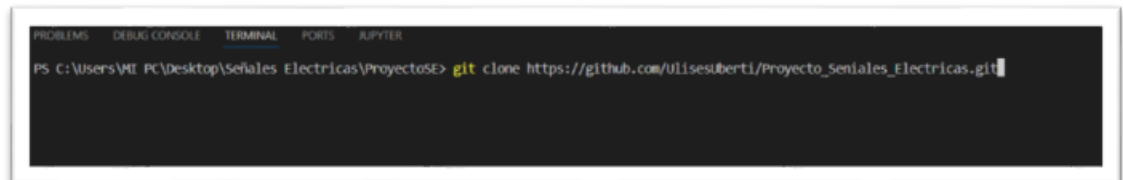
El cuadro verde indica las carpetas de interés, ‘datos’ contiene el archivo de audio original trabajado, ‘modules’ contiene los archivos .py que permiten modularizar el código al utilizar funciones reutilizables, y ‘src’ contiene los archivos principales donde se encuentra el transmisor y receptor.

3- Clonar repositorio:



Presionar el botón verde “Code” y copiar la dirección del repositorio que se indica en el cuadro amarillo inferior.

Luego se ingresa a VSC, se abre la terminal y se escribe el siguiente comando:



git clone https://github.com/UlisesUberti/Proyecto_Seniales_Electricas.git

El repositorio se descargará en la carpeta en que se encuentre ubicado al momento de ingresar el comando.

TRANSMISOR:

La primera parte del sistema consta del ‘Transmisor’, en esta primera parte se realizan las operaciones de tratamiento de la señal de audio como muestreo, codificación, modulación y conversión en archivo de audio a la señal de salida.

Archivo .wav:

El archivo de audio que se trabajo es un tipo .wav, este tipo de archivo mantiene un formato ‘RIFF’, su estructura está separada por bloques de datos o “chunks”. Este archivo almacena el audio luego de muestrear y cuantizar la señal analógica, podría decirse que en formato PCM pero sin el paso de codificarla tal que se tenga una señal digital serial.

Uno de los bloques de datos contiene las características de la señal de audio:

- Frecuencia de muestreo
- Cantidad de bits por muestra
- Cantidad de muestras
- Cantidad de canales

Respecto a la cantidad de canales se trabajó con audio de un solo canal (mono), pudiendo ampliarse a estéreo con algunas modificaciones.

Particularmente la cantidad de bits por muestra fue de '16' y la frecuencia de muestreo de 24[kHz]

Estos datos se encuentran disponibles en el inicio del código del transmisor. Este formato del archivo permite conocer todos los parámetros de la señal con la que se tratara y facilita a la hora de trabajarla.

Señal de Audio

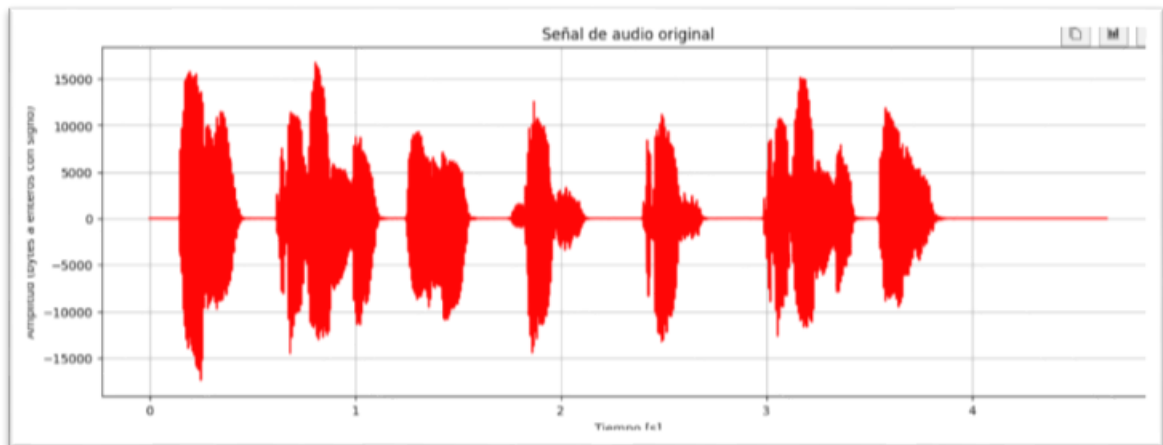
Con los parámetros de la señal podemos graficarla en función del tiempo y tener una idea de su forma y amplitud, además de observar cuando concluye exactamente para no transmitir muestras que no aporten nada, por ejemplo, un audio con duración de 5 segundos donde la voz solo está presente hasta el segundo 4.

Estos parámetros también permiten realizar una FFT y obtener el espectro de frecuencias de la señal de audio. El archivo .wav muestrea por defecto la señal en un valor establecido, obtener la FFT de la señal nos permitirá determinar si es posible un re-muestreo de la señal.

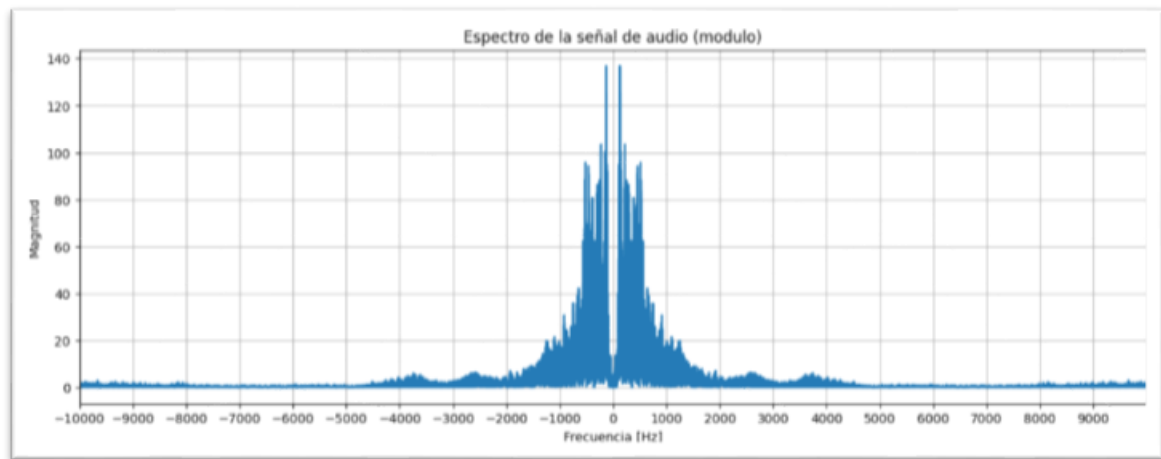
Esto se realizó mediante el espectro de densidad de energía, una vez obtenido el espectro se busca la componente máxima de la señal en base al 95% de la energía de la señal.

Y se obtuvo una componente que permitía un re-muestreo de la señal menor que el que se realizó en el archivo .wav.

En el programa se realizó un re-muestreo a 3000[Hz], una frecuencia de muestreo 8 veces menor que la original.



Señal de audio en función del tiempo



FFT de la señal de audio

Señal Moduladora:

Para la señal moduladora requerimos una señal digital, y lo que tenemos es un arreglo de bytes correspondientes a cada muestra de la señal de audio.

Entonces uno de los pasos es convertir esos bytes en bits, generando un archivo de bits, donde sabemos que cada 16 bits (caso particular del archivo con el que se trabajó) tenemos una muestra, una vez que tenemos el arreglo de bits, generamos pulsos correspondientes a cada bit, pero el ancho de cada pulso depende de la tasa de bits

Tasa de Bits:

La tasa de bits teórica que deberíamos utilizar se define por la cantidad de bits y la frecuencia de muestreo utilizada, pero como este no es el caso de un sistema en el que se necesita escuchar el audio en tiempo real, podemos permitirnos extender la duración de la señal, lo que se logra utilizando una tasa de bits menor a la teórica, esto se explica mejor con una de las propiedades de la transformada de Fourier:

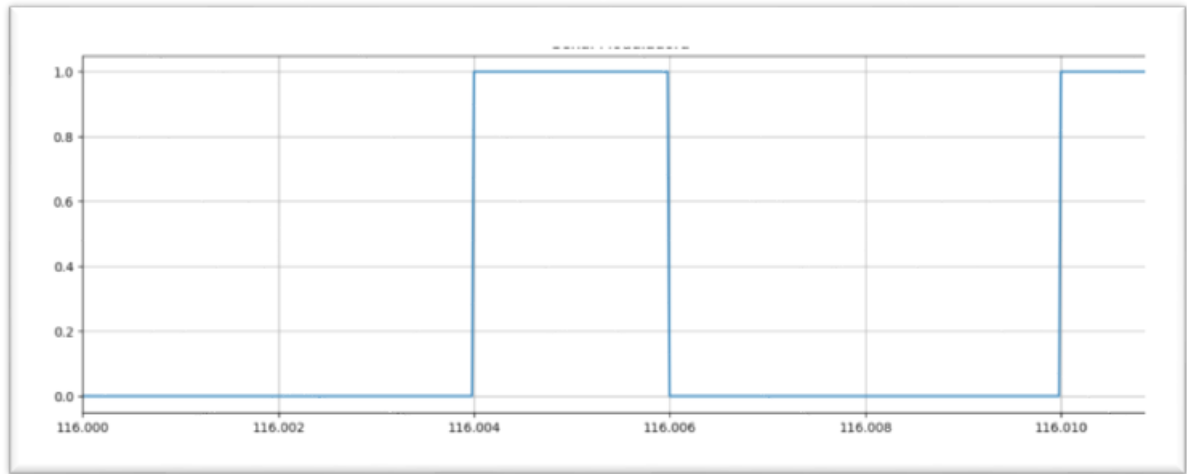
Propiedad Escalar, sea una función $f(t)$ con su transformada $F(f)$, y sea una constante real 'a', entonces sea $f(at)$ su transformada es:

$$\frac{1}{|a|} F\left(\frac{f}{a}\right)$$

La señal en el tiempo se 'expande', mientras que el espectro en la frecuencia se 'comprime', en ambos casos la forma no cambia.

De esta manera podemos adoptar una tasa de bits 'Rb' y como consecuencia obtener un AB de la señal moduladora menor y una duración en el tiempo mayor.

En el programa se tomó un factor de '48' ($a=48$), debido a una tasa de bits, Rb, de 1000[bps], esto sale de que Rb original era de 48[kbps].

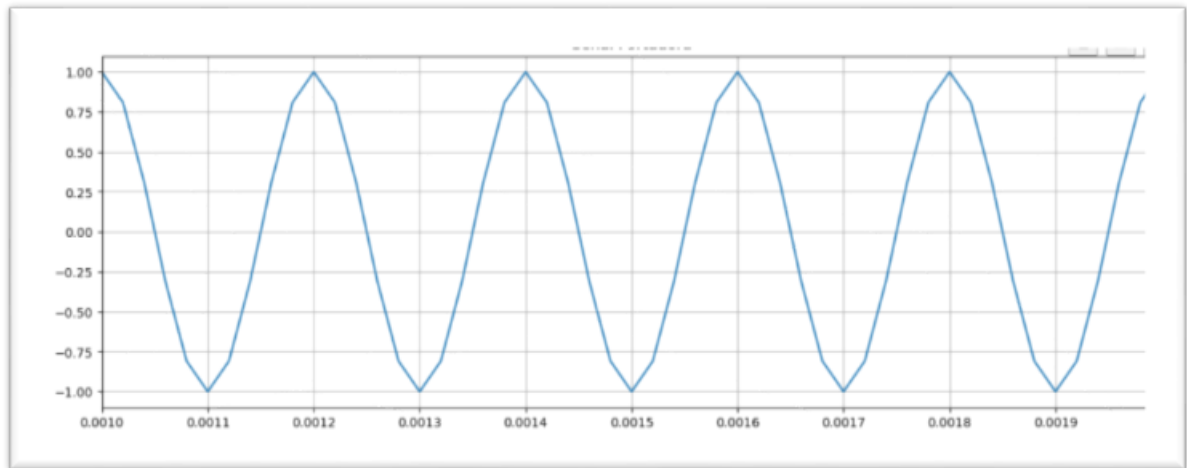


Señal moduladora del programa

Señal Portadora:

Como requisito se tiene que la señal de portadora debe tener una frecuencia entre 5[kHz] y 10[kHz], la portadora consta de una señal sinusoidal que será modulada por la señal digital. Se eligió una frecuencia de 5[kHz] para la portadora.

Basta con generar la señal a la frecuencia deseada y con la duración misma de la señal digital y muestrearla a la misma frecuencia, para que la modulación se realice sin problema.

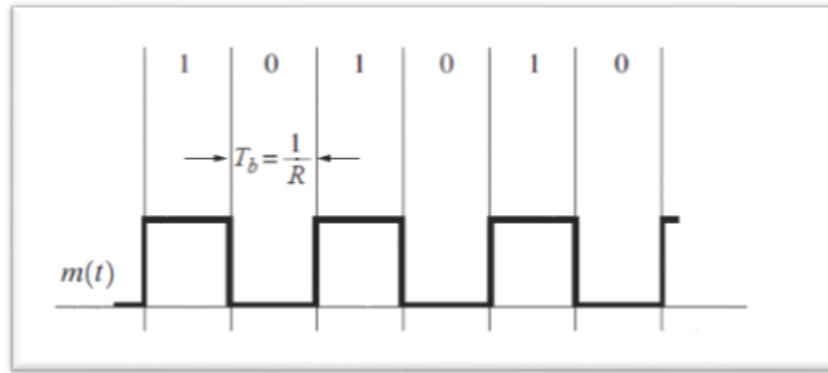


Señal Portadora del programa

Modulación OOK:

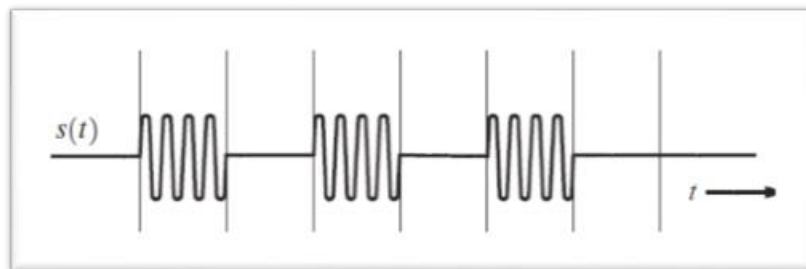
Modulación ASK o modulación por corrimiento de amplitud, consiste en la conmutación de una señal sinusoidal (portadora), produciendo un “encendido” y “apagado” mediante la señal digital da datos (moduladora). Es una técnica sencilla de implementar

Supongamos una señal de datos binarios del tipo unipolar:



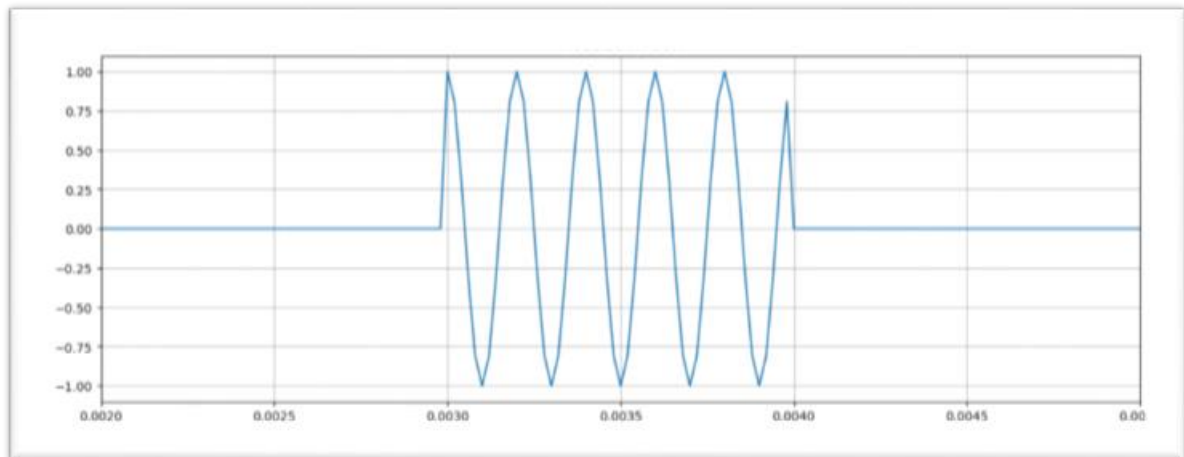
Señal Moduladora

La señal OOK será resultado del producto en el tiempo de la señal moduladora y la portadora.



Señal OOK

Luego convertimos la señal modulada en un archivo de audio, este archivo de audio tendrá una duración mayor a la del audio original, debido a la expansión de la señal moduladora en el tiempo, por lo que el receptor deberá tener en cuenta esto a la hora de demodular.



Ejemplo de la señal OOK del programa

RECEPTOR

El receptor recibe el archivo de audio de la señal modulada, y comenzamos con un bloque de comprobación de la señal.

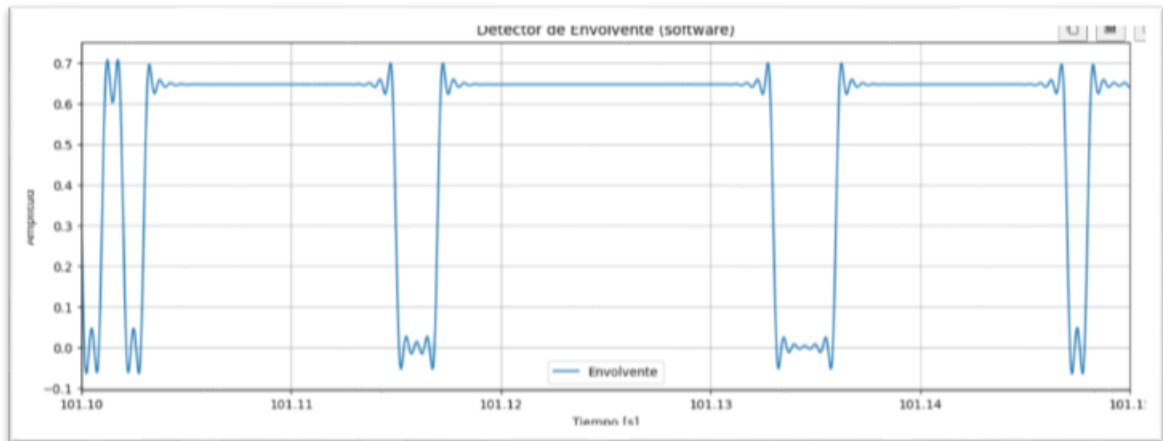
Podemos verificar que se trata de la señal OOK graficándola nuevamente, verificando su duración y la cantidad de elementos que contienen los arreglos correspondientes.

Demodulador:

En la práctica para demodular una señal OOK basta con utilizar un detector de envolvente y un comparador para reconstruir los pulsos de la mejor manera.

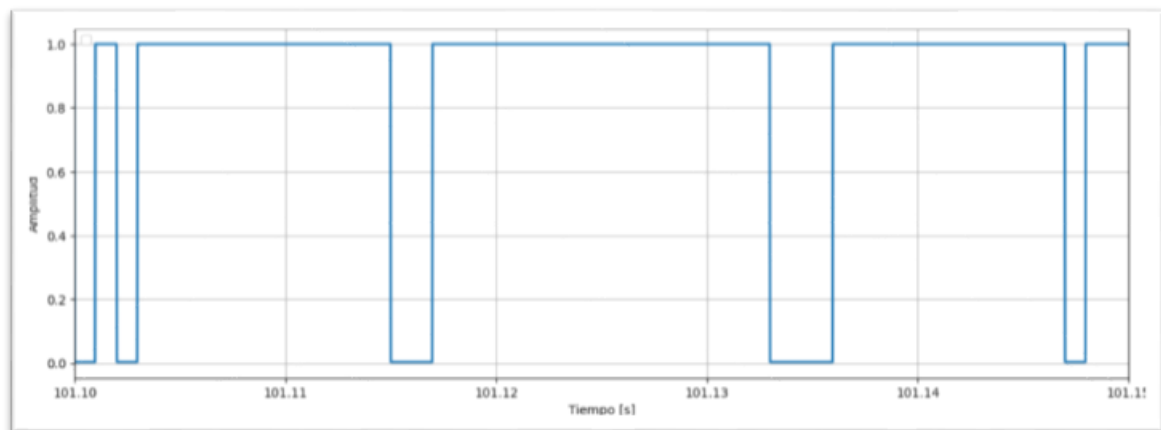
Entonces lo que se hace es rectificar la señal, y luego filtrarla con un filtro pasabajos, a una frecuencia de corte mayor que la frecuencia considerada máxima de la señal moduladora pero a una frecuencia menor de la señal de portadora.

De esta manera recuperamos la envolvente de la señal, que se aproxima a pulsos que representan la moduladora.



Envolvente obtenida luego del filtro

Después de filtrar utilizamos un comparador tomando un valor de referencia a partir del cual se considera como '1' o '0', de esta manera reconstruimos la señal digital original



Señal digital moduladora luego del comparador

De esta manera recuperamos la señal digital y solo resta trabajarla para que sea de nuevo un arreglo de 16 bits por muestra.

Conociendo la frecuencia de muestreo de la señal modulada, podemos construir un arreglo que contenga los bits correspondientes a cada pulso. Luego debemos agrupar estos bits de a 16, una vez que tenemos grupos de 16 bits, solo resta convertirlos a enteros con signos, como en el caso inicial de la señal de audio.

A partir de acá se genera un archivo .wav de la señal demodulada, también puede graficarse en función del tiempo o su FFT para comparar con la original. Para esto debe saberse de antemano que se modificó la frecuencia de muestreo de la señal de audio original.

BIBLIOGRAFIA:

- Sistemas de comunicaciones digitales y analógicos – Couch / Cuevas / Romero
- Introducción a la teoría y sistemas de comunicación – Lathi