

Principios de Comunicaciones

Proyecto N°3-1

Simulación de un sistema FM

Integrantes: Jean Cherubini
Profesor: César Azurdia
Auxiliares: Sandy Bolufe
Alejandro Cuevas
Ayudante: Nicolas Ortega
Fecha de realización: 6 de julio de 2018
Fecha de entrega: 6 de Julio de 2018
Santiago, Chile

Índice de Contenidos

1. Introducción	1
2. Desarrollo del trabajo	1
2.1. Caracterización del mensaje que se desea transmitir	1
2.1.1. Señal Moduladora	1
3. Modulación de la Señal	3
3.1. Señal Portadora	3
3.2. Modulación con desviación $\Delta f = 75$ [Hz]	4
3.3. Modulación con desviación $\Delta f = 100$ [Hz]	5
4. Adición de Ruido AWGN a la Señal Modulada	7
5. Demodulación de la señal	11
5.1. Modulación con $\Delta f = 3000$ [Hz]	14
6. Audios de Comparación	15
7. Conclusiones	16

Lista de Figuras

1	Señal Cuadrada de 500 [Hz]	2
2	Señal Senoidal de 10 [kHz]	3
3	Señal modulada con $\Delta 75$	4
4	Señal modulada con $\Delta 100$	6
5	Comparación entre ondas moduladas y señal original	6
6	Adición de ruido de 15 dB a la señal modulada con $\Delta 75$ [Hz]	8
7	Adición de ruido de 15 dB a la señal modulada con $\Delta 100$ [Hz]	9
8	Adición de ruido de 30 dB a la señal modulada con $\Delta 75$ [Hz]	10
9	Adición de ruido de 30 dB a la señal modulada con $\Delta 100$ [Hz]	11
10	Demodulación de señal con $\Delta f = 75$ [Hz] y SNR de 15 dB	12
11	Demodulación de señal con $\Delta f = 100$ [Hz] y SNR de 15 dB	12
12	Demodulación de señal con $\Delta f = 75$ [Hz] y SNR de 30 dB	13
13	Demodulación de señal con $\Delta f = 100$ [Hz] y SNR de 30 dB	13
14	Modulación de señal con $\Delta f = 3000$ [Hz]	14
15	Demodulación de señal con $\Delta f = 3000$ [Hz] y SNR de 30 dB	15

Lista de Códigos

1	Código utilizado para generar señal cuadrada de 500 Hz y graficarla en dominios de tiempo y frecuencia	2
---	--	---

2	Código utilizado para generar señal senoidal de 10 kHz y graficarla en dominios de tiempo y frecuencia	3
3	Código utilizado para modular señal con $f = \Delta 75$	4
4	Código utilizado para modular señal con $f = \Delta 100$	6
5	Cálculo y adición de ruido	7
6	Código utilizado para demodular señales con ruido	13
7	Código utilizado para crear archivos de audio.	15

1. Introducción

En el presente informe se presentará la implementación y los resultados obtenidos a partir del proyecto N°3 del curso Principios de Comunicaciones, donde se pide simular un sistema FM

El trabajo fue desarrollado utilizando *Python 3.6*¹. El proyecto tiene como objetivo crear señales moduladora y portadora para así hacer modulaciones, adiciones de ruido y demodulaciones.

2. Desarrollo del trabajo

2.1. Caracterización del mensaje que se desea transmitir

2.1.1. Señal Moduladora

Se pidió generar una onda cuadrada con $V_{pp} = 1,5 [V]$ y frecuencia $f = 500 [Hz]$. Para esto, se utilizó la librería *scipy.signals.square*, que recibe los atributos de amplitud y frecuencia. Se tiene además que el muestreo de esta señal se hace con una frecuencia $f_s = 10^5 [Hz]$ (es decir, $T_s = 0,00001 [s]$) durante $0,2 [s]$. Se graficó la señal tanto en su dominio del tiempo como en su dominio de frecuencia, como se puede ver en la Figura 1. Se aprecia que el periodo de la señal corresponde a $T_{500} = 0,002 [s]$. Además, si se observa el espectro de frecuencia, es posible ver peaks en la frecuencia fundamental y además en los múltiplos impares de ella. Esto se debe a que la serie de fourier de la señal cuadrada corresponde a series de $Sinc(\pi f)$ centrados en los multiplos impares de la frecuencia fundamental $500 [Hz]$, como se muestra en la Ecuación 1. Para realizar esta sección se utilizó el código mostrado en Código 1.

$$\sum_1^{\infty} b_n \sin(2n\pi 500) ; a_0 = 0 \qquad b_n = 1,5 \cdot \text{sinc}\left(\frac{n\pi}{2}\right) a_n = 0 \qquad (1)$$

¹ <https://www.python.org/>

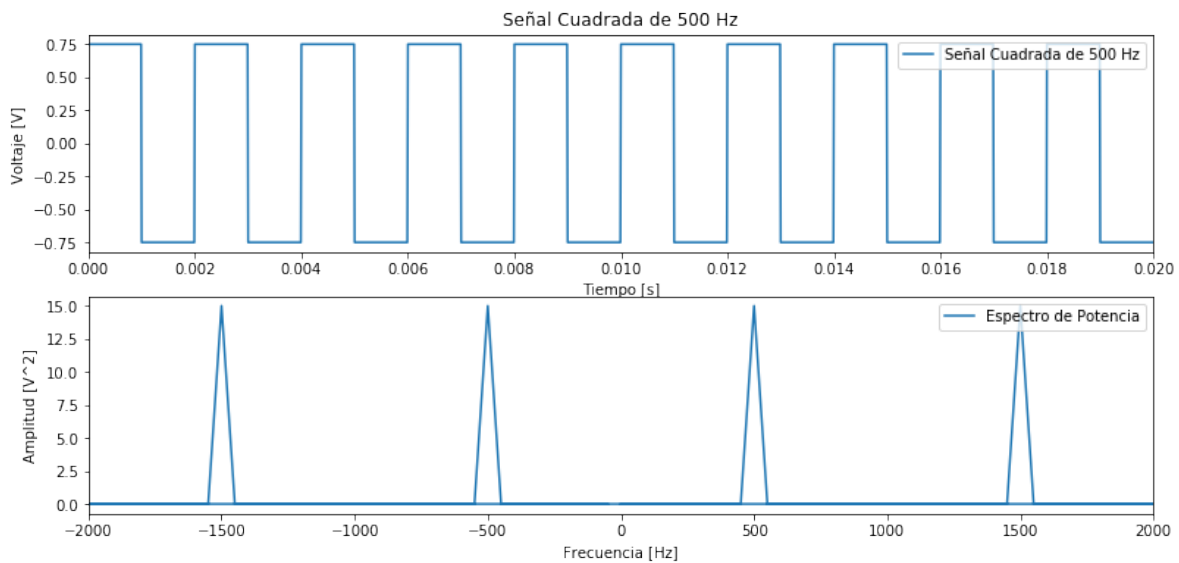


Figura 1: Señal Cuadrada de 500 [Hz]

Código 1: Código utilizado para generar señal cuadrada de 500 Hz y graficarla en dominios de tiempo y frecuencia

```

1 timestep=1/(10**5) #Periodo de Muestreo
2 t = np.arange(0, 0.2, timestep) #Tiempo de Muestreo
3 signal_500=0.75*signal.square(2*np.pi*500*t) #Seal cuadrada de 500 Hz
4
5 #Fft seal cuadrada de 500 Hz
6 fft500=np.fft.fft(signal_500)
7 fftFreq500=np.fft.fftfreq(signal_500.shape[-1],timestep)
8
9 #Grafico de seal
10 figure1, (ax1,ax2)=plt.subplots(2,1,figsize=(13, 6))
11 ax1.plot(t,signal_500, label='Seal Cuadrada de 500 Hz')
12 ax1.set_title('Seal Cuadrada de 500 Hz')
13 ax1.set_xlabel('Tiempo [s]')
14 ax1.set_ylabel('Voltaje [V]')
15 ax1.set_xlim([0,10/500])
16 ax1.set_xticks(np.arange(0,11/500,step=1/500))
17 ax1.legend(loc='upper right')
18
19 #Grafico espectro de seal
20 ax2.plot(fftFreq500,fft500,label='Espectro de Potencia')
21 ax2.set_xlabel('Frecuencia [Hz]')
22 ax2.set_ylabel('Amplitud [V^2]')
23 ax2.legend(loc='upper right')
24 ax2.set_xlim(-2000,2000)

```

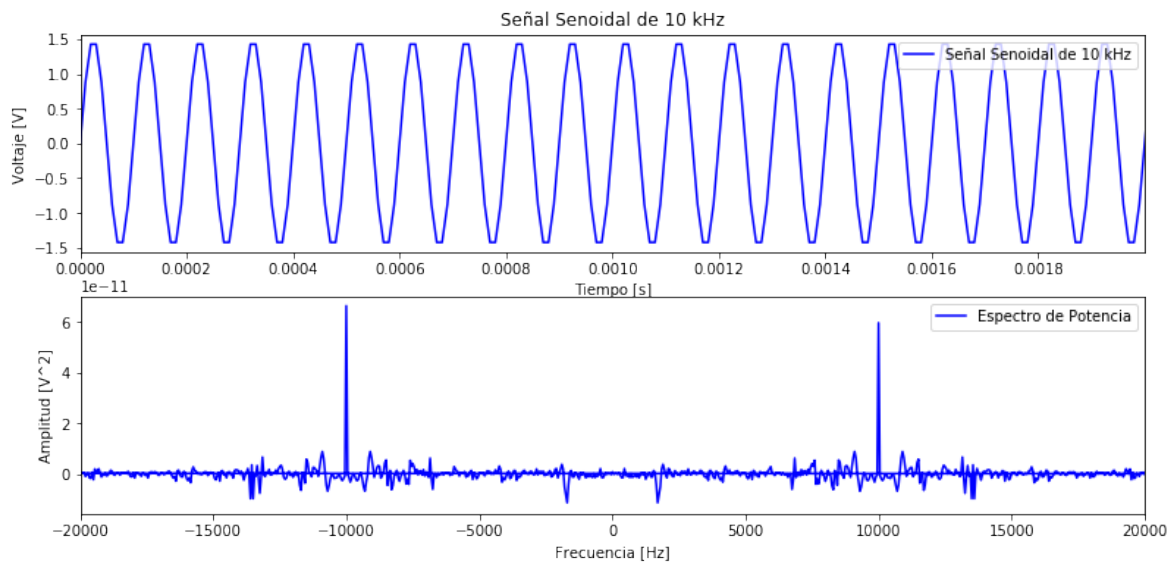


Figura 2: Señal Senoidal de 10 [kHz]

3. Modulación de la Señal

3.1. Señal Portadora

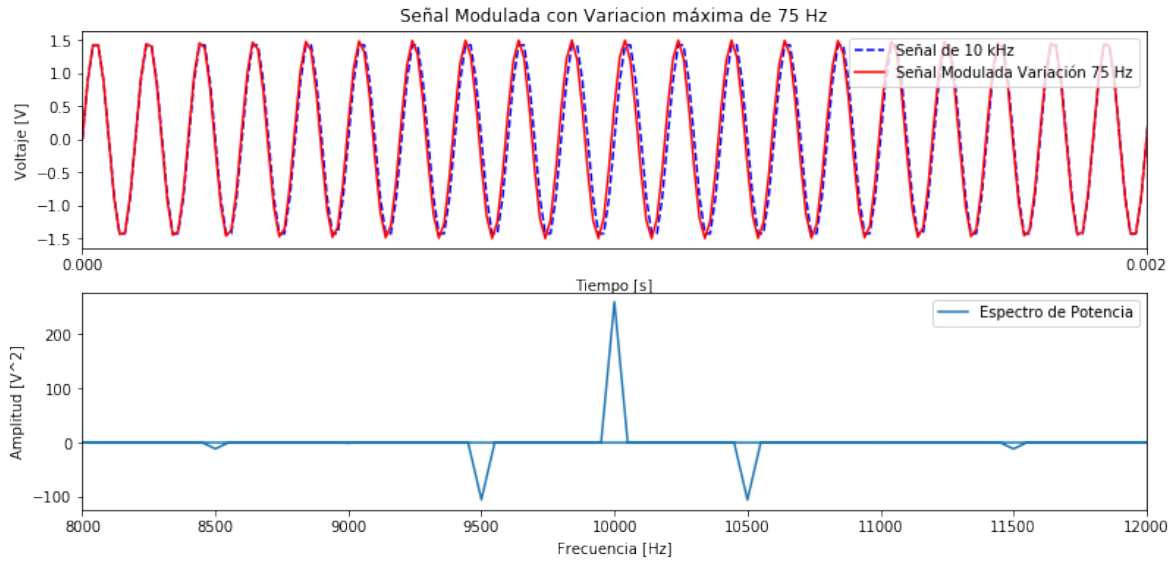
Se pidió generar una señal senoidal de 10 [kHz] con amplitud $V_{pp} = 3$ [V]. Para esto se utilizó la función predeterminada de python $\sin()$, como se muestra en el Código 2. Para esto se utilizó el mismo tiempo de muestreo T_s que en la parte anterior, pues este corresponde a una frecuencia de muestreo de más del doble de la frecuencia 10 [kHz], por lo tanto es suficiente para muestrear de buena forma la señal. Se graficó parcialmente la señal en la Figura 2 donde se aprecia que el periodo corresponde a 0,0001 [s]. Se aprecia además que en el dominio de la frecuencia los peaks se encuentran en ± 10 [kHz]. Esto se debe a que la señal es sinusoidal.

Código 2: Código utilizado para generar señal senoidal de 10 kHz y graficarla en dominios de tiempo y frecuencia

```

1 signal_10kHz=1.5*np.sin(2*np.pi*(10**4)*t) #Generacion de Seal sinusoidal de 500 kHz
2
3 #Fft seal cuadrada de 10kHz Hz
4 fft10kHz=np.fft.fft(signal_10kHz)
5 fftFreq10kHz=np.fft.fftfreq(signal_10kHz.shape[-1],timestep)
6
7 #Grafico de seal
8 figure1, (ax1,ax2)=plt.subplots(2,1,figsize=(13, 6))
9 ax1.plot(t,signal_10kHz,'b', label='Seal Cuadrada de 10 kHz')
10 ax1.set_title('Seal Cuadrada de 10 kHz')
11 ax1.set_xlabel('Tiempo [s]')
12 ax1.set_ylabel('Voltaje [V]')
13 ax1.set_xlim([0,1/500])
14 ax1.set_xticks(np.arange(0,1/500,2/(10**4)))
15 ax1.legend(loc='upper right')

```

Figura 3: Señal modulada con $\Delta 75$

```

16
17 #Grafico espectro de seal
18 ax2.plot(fftFreq10kHz,fft10kHz,'b',label='Espectro de Potencia')
19 ax2.set_xlabel('Frecuencia [Hz]')
20 ax2.set_ylabel('Amplitud [V^2]')
21 ax2.legend(loc='upper right')
22 ax2.set_xlim(-20000,20000)

```

3.2. Modulación con desviación $\Delta f = 75 [Hz]$

Para esta sección, fue necesario utilizar una librería llamada *numpy.trapz* pues esta permite calcular la integral de una función hasta cierto punto. Para hacer la modulación como se pide se utilizó la Ecuación 2. Y el código mostrado en Código 3. La señal modulada obtenida se puede observar en la Figura 3 tanto en el tiempo como en frecuencia (solo se graficó en el periodo de la señal de $500 Hz$ en el tiempo para el primer gráfico y un rango de frecuencia en donde existían picks para mejor apreciación de la señal).

$$\begin{aligned}
 \Phi_{FM_{\Delta 75}} &= E_c \cdot \sin \left(2\pi f_c + k_f \int_{-\infty}^t m(t) \right) & ; m(t) : \text{Señal cuadrada de } 500Hz \\
 &= 1,5 \cdot \sin \left(2\pi 10^4 + 2\pi 75 \int_{-\infty}^t m(t) \right)
 \end{aligned} \quad (2)$$

Código 3: Código utilizado para modular señal con $f = \Delta 75$

```

1 #Espectro frecuencia
2 fftMod75=np.fft.fft(modulatedSignal75)

```

```

3 fftFreq75=np.fft.fftfreq(modulatedSignal75.shape[-1],timestep)
4
5 figure1, (ax1,ax2) =plt.subplots(2,1,figsize=(13, 6))
6
7 #Grafico en el tiempo
8 ax1.plot(t,signal_10kHz,'--b',label='Seal de 10 kHz')
9 ax1.plot(t,modulatedSignal75,'r', label='Seal Modulada Variación 75 Hz')
10 ax1.set_title('Seal Modulada con Variacion máxima de 75 Hz')
11 ax1.set_xlabel('Tiempo [s]')
12 ax1.set_ylabel('Voltaje [V]')
13 ax1.set_xlim([0,1/500])
14 ax1.set_xticks(np.arange(0,2/500,step=1/500))
15 ax1.legend(loc='upper right')
16
17 #Grafico en Frecuencia
18 ax2.plot(fftFreq75,fftMod75,label='Espectro de Potencia')
19 ax2.set_xlabel('Frecuencia [Hz]')
20 ax2.set_ylabel('Amplitud [V^2]')
21 ax2.set_xlim([8000,12000])
22 ax2.legend(loc='upper right')

```

3.3. Modulación con desviación $\Delta f = 100 \text{ [Hz]}$

De forma muy similar a la Sección 3.2, se utilizó la Ecuación 3 para realizar la modulación. El Código 4 fue el utilizado para realizar esta modulación. Lo obtenido puede apreciarse en la Figura 5 tanto para el tiempo como para la frecuencia. En la Figura 4 puede observarse una sección de las ondas, mostrando la comparación entre las modulaciones y la señal original.

$$\begin{aligned}
 \Phi_{FM_{\Delta 100}} &= E_c \cdot \sin \left(2\pi f_c + k_f \int_{-\infty}^t m(t) \right) & ; m(t) : \text{Señal cuadrada de } 500\text{Hz} \\
 &= 1,5 \cdot \sin \left(2\pi 10^4 + 2\pi 100 \int_{-\infty}^t m(t) \right)
 \end{aligned} \tag{3}$$

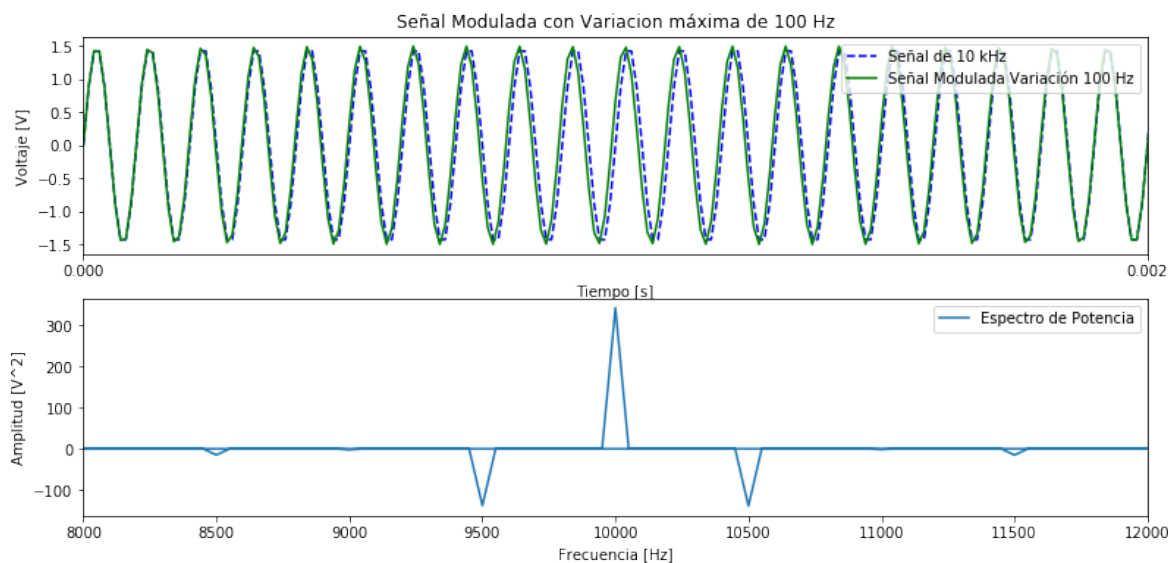
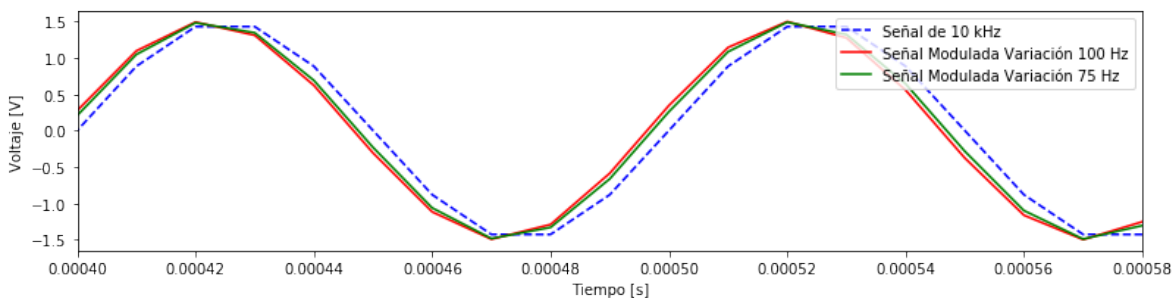
Figura 4: Señal modulada con $\Delta 100$ 

Figura 5: Comparación entre ondas moduladas y señal original

Código 4: Código utilizado para modular señal con $f = \Delta 100$

```

1 #Espectro frecuencia
2 fftMod100=np.fft.fft(modulatedSignal100)
3 fftFreq100=np.fft.fftfreq(modulatedSignal100.shape[-1],timestep)
4
5 figure1, (ax1,ax2) =plt.subplots(2,1,figsize=(13, 6))
6
7 #Grafico en el tiempo
8 ax1.plot(t,signal_10kHz,'--b',label='Seal de 10 kHz')
9 ax1.plot(t,modulatedSignal100,'r', label='Seal Modulada Variación 100 Hz')
10 ax1.set_title('Seal Modulada con Variacion máxima de 100 Hz')
11 ax1.set_xlabel('Tiempo [s]')
12 ax1.set_ylabel('Voltaje [V]')
13 ax1.set_xlim([0,1/500])
14 ax1.set_xticks(np.arange(0,2/500,step=1/500))
15 ax1.legend(loc='upper right')
16

```

```

17 #Grafico en Frecuencia
18 ax2.plot(fftFreq100,fftMod100,label='Espectro de Potencia')
19 ax2.set_xlabel('Frecuencia [Hz]')
20 ax2.set_ylabel('Amplitud [V^2]')
21 ax2.set_xlim([8000,12000])
22 ax2.legend(loc='upper right')

```

4. Adición de Ruido AWGN a la Señal Modulada

Para esta sección, se utilizó la Ecuación 4 para encontrar la potencia del ruido en cada caso para obtener 15 y 30 [dB] para las modulaciones de $\Delta 75$ y $\Delta 100$ Hz]. Después se utilizó que dado que el canal tiene ganancia unitaria y la media del ruido blanco gaussiano es cero, es posible crear una señal con una distribución normal con una varianza igual a la potencia del ruido que se necesita, como aparece en la Ecuación 5. Así, la señal de ruido blanco será basada en una distribución $\mathcal{N}(0, \sigma^2)$. Luego de generar esta señal de ruido, esta fue sumada a cada señal obteniendo las figuras 6,7,8,9. Lo anterior se realizó con el código mostrado en Código 5, pero con ligeras modificaciones para cada caso.

$$SNR_{dB} = 10 \log_{10} \left(\frac{P_{Señal}}{P_{Ruido}} \right) \quad (4)$$

$$P_{Ruido} = \sigma^2 \quad (5)$$

Código 5: Calculo y adición de ruido

```

1 #Calculo potencia del Ruido
2 potRuidoLog75_30=((np.trapz(abs((modulatedSignal75[0:indexPeriod])**2),dx=timestep))/(t[
    indexPeriod]))/(10**(30/10))
3 #Desviacion estandar para la distrib normal
4 std75_30=np.sqrt(potRuidoLog75_30)
5 #Creacion de Vector de Ruido Blanco
6 ruido75_30=np.random.normal(0,std75_30,modulatedSignal75.shape[-1])
7 #Suma seal+Ruido
8 senalRuidosa75_30=modulatedSignal75+ruido75_30

```

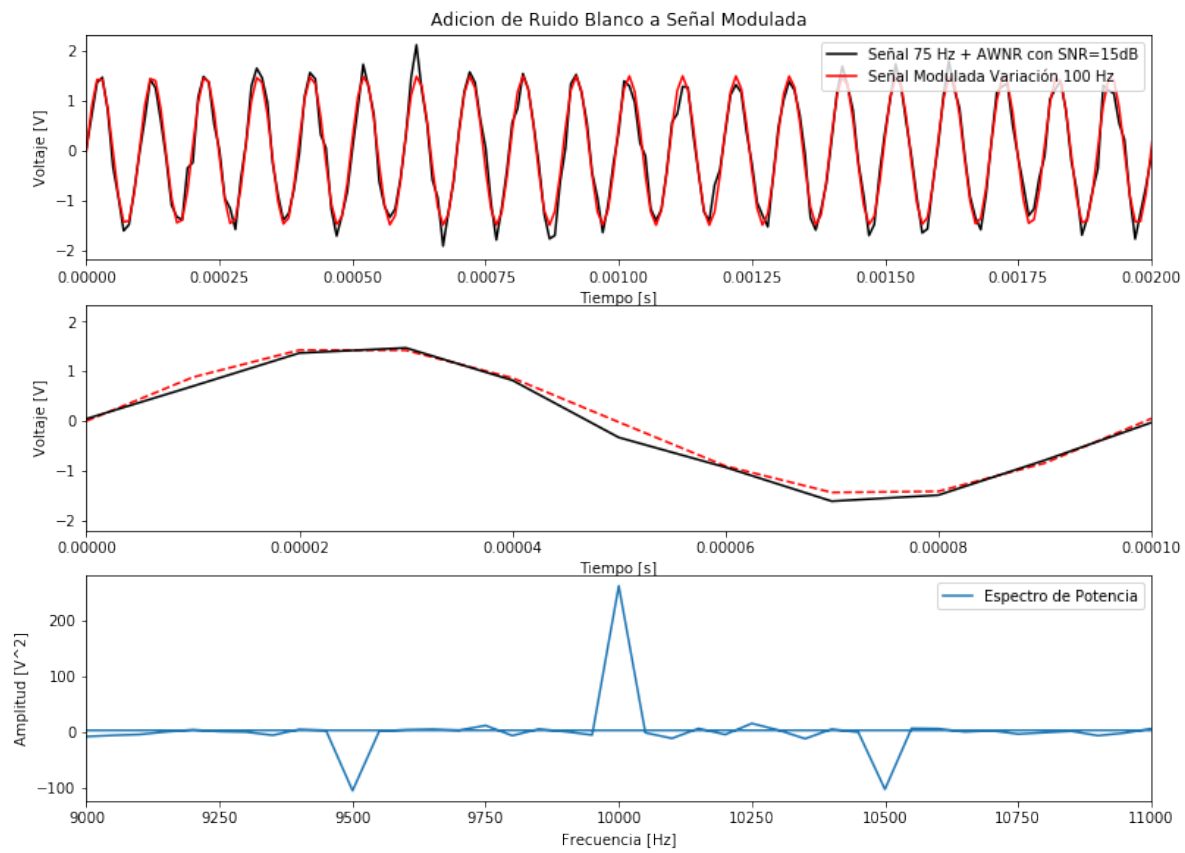


Figura 6: Adición de ruido de 15 dB a la señal modulada con $\Delta 75$ [Hz]

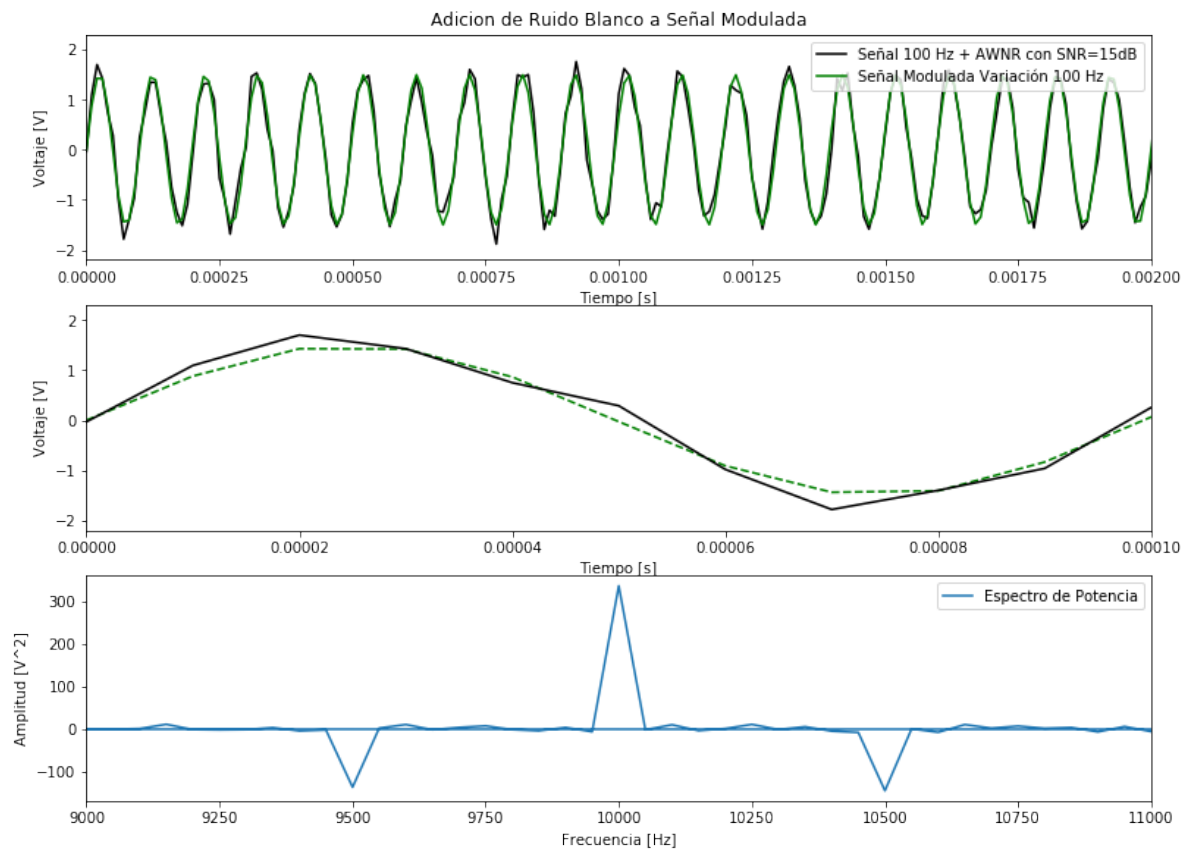


Figura 7: Adición de ruido de 15 dB a la señal modulada con $\Delta 100$ [Hz]

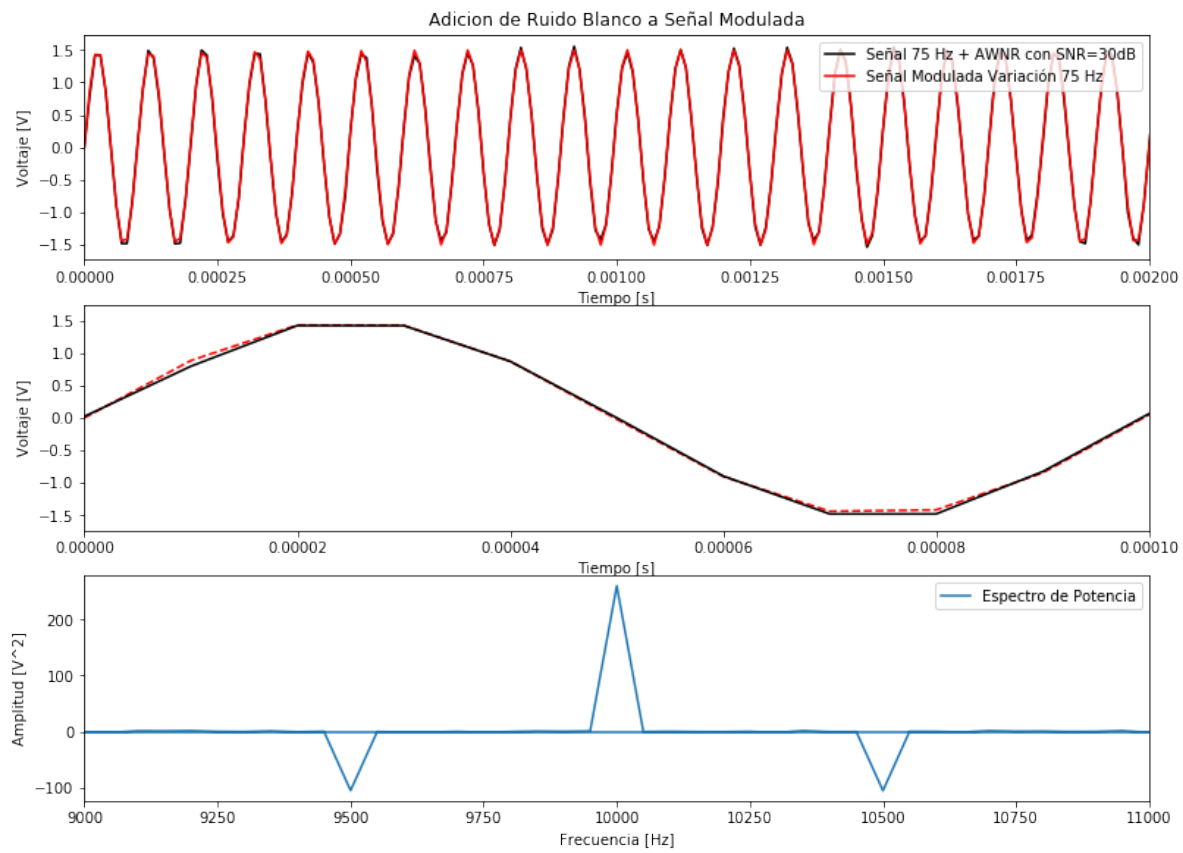


Figura 8: Adición de ruido de 30 dB a la señal modulada con $\Delta 75$ [Hz]

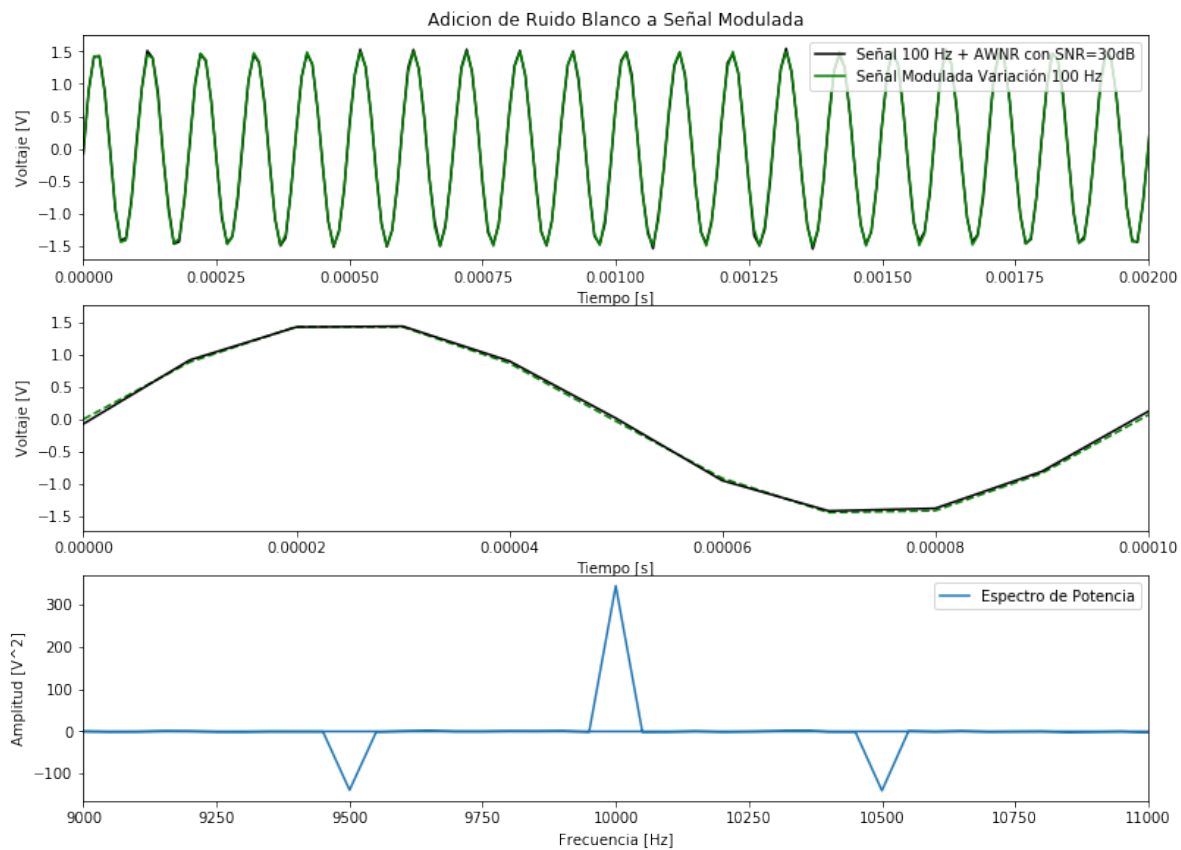


Figura 9: Adición de ruido de 30 dB a la señal modulada con $\Delta 100$ [Hz]

5. Demodulación de la señal

Para demodular la señal, primero se aplicó una derivación con respecto al tiempo para así poder ver cómo es la variación de frecuencia. Luego de hacer esto, se utilizó la librería *scipy.signal.hilbert* para encontrar la señal envolvente, que entregaría la señal moduladora, es decir, la información requerida.

Desafortunadamente, debido al nivel de ruido, no fue posible recuperar la señal de buena manera para ninguno de los 4 casos anteriores, como se puede ver en las figuras 10, 11, 12 y 13. Que distan claramente de la señal original mostrada en 1.

Esto se hizo utilizando variaciones de la sección de Código mostrada en 6.

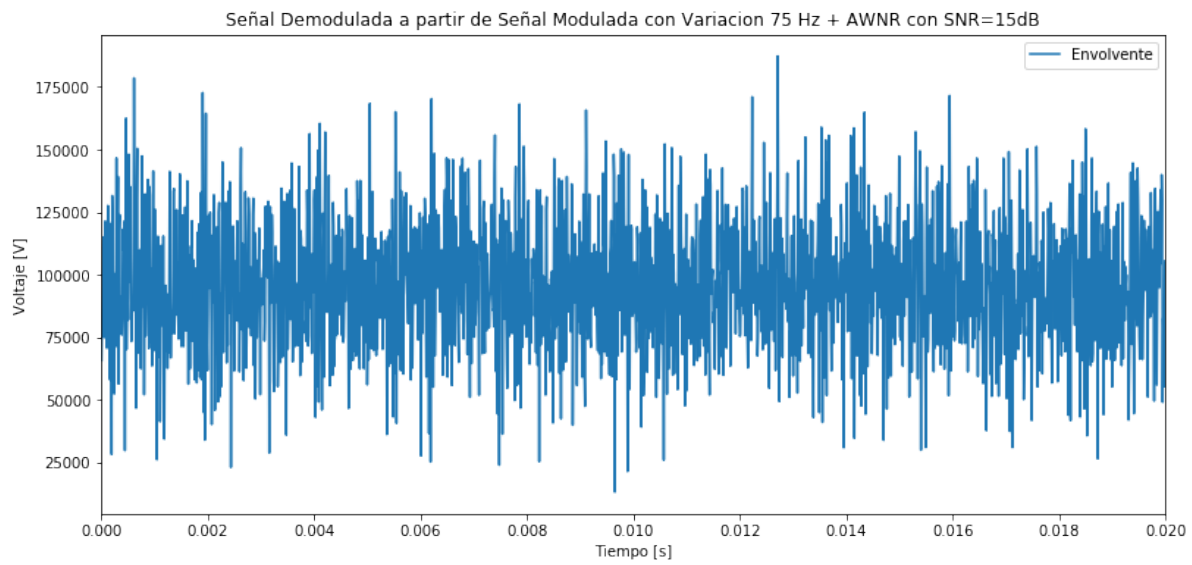


Figura 10: Demodulación de señal con $\Delta f = 75 [Hz]$ y SNR de 15 dB

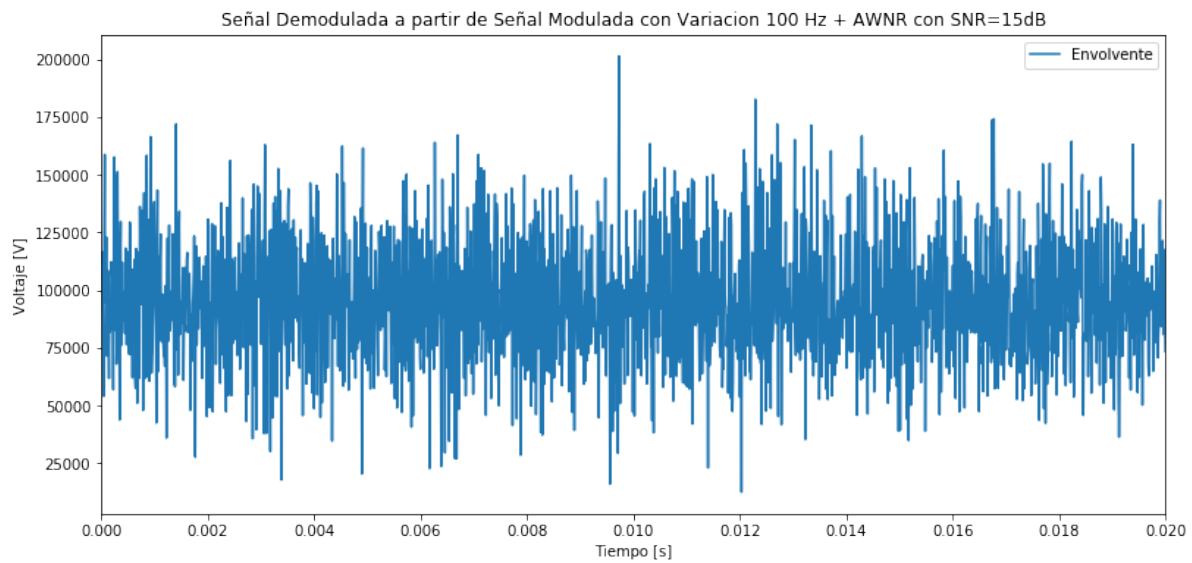


Figura 11: Demodulación de señal con $\Delta f = 100 [Hz]$ y SNR de 15 dB

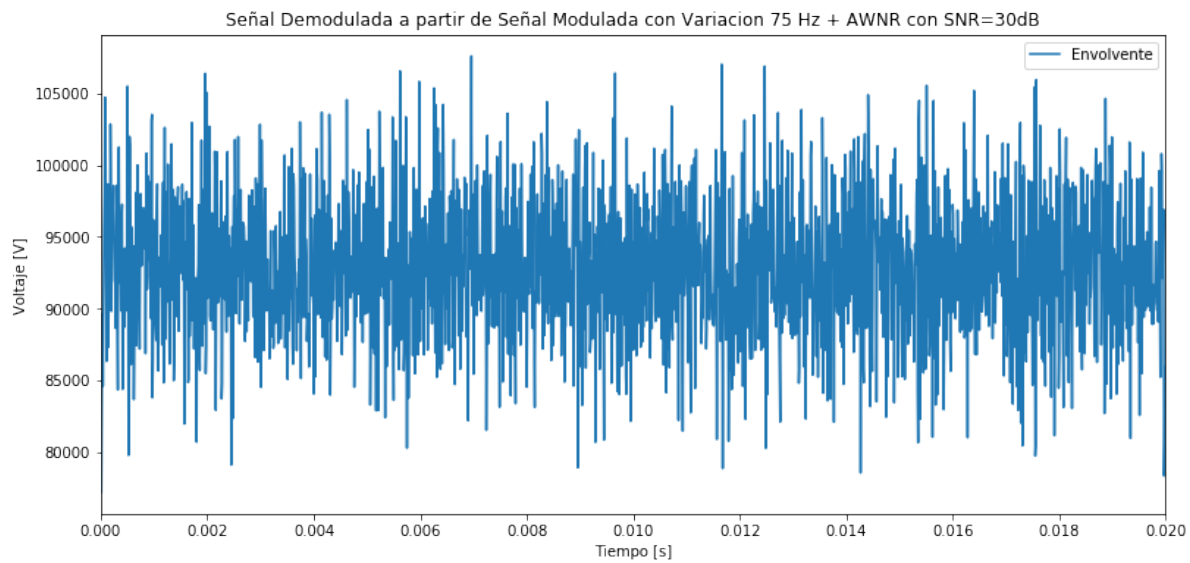


Figura 12: Demodulación de señal con $\Delta f = 75 [Hz]$ y SNR de 30 dB

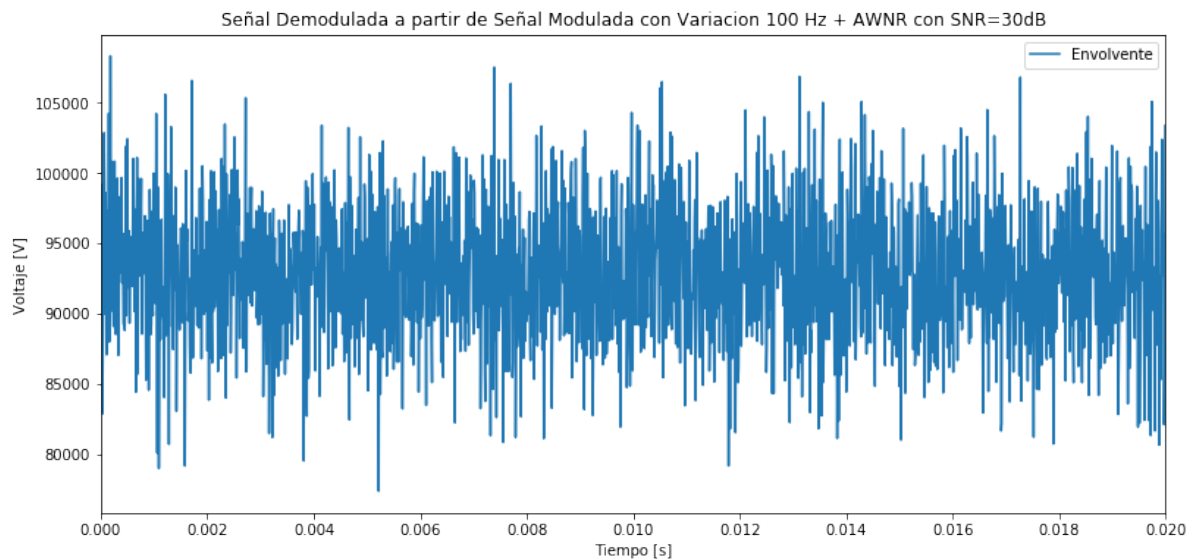


Figura 13: Demodulación de señal con $\Delta f = 100 [Hz]$ y SNR de 30 dB

Código 6: Código utilizado para demodular señales con ruido

```

1 deriv75_15=np.diff(senalRuidosa75_15)/timestep
2 hilb75_15 = sp.signal.hilbert(deriv75_15)
3 envolv75_15 = np.abs(hilb75_15)
4
5 fig = plt.figure(figsize=(13,6))
6 plt.plot(t[1:], envolv75_15, label='Envolvente')
7 plt.title('Seal Demodulada a partir de Seal Modulada con Variacion 75 Hz + AWRN con SNR
            =15dB')
```



```

8 plt.xlim([0,2/500])
9 plt.xlabel("Tiempo [s]")
10 plt.ylabel('Voltaje [V]')
11 plt.legend()

```

5.1. Modulación con $\Delta f = 3000 \text{ [Hz]}$

Es por esto que se propuso una solución que funcionó de mejor manera. Utilizando la misma técnica de demodulación pero en este caso cambiando parámetros de modulación.

Se propuso modular utilizando una mayor variación en frecuencia, obteniendo lo mostrado en la Figura 14, y siguiendo el proceso de adición de ruido ($\text{SNR}=30 \text{ [dB]}$) y demodularon, se obtuvo lo mostrado en la Figura 15. Esta señal es claramente de 500 [Hz] aunque más ruidosa. Su amplitud además es distinta pero esto puede ser fácilmente normalizado aplicando una ganancia.

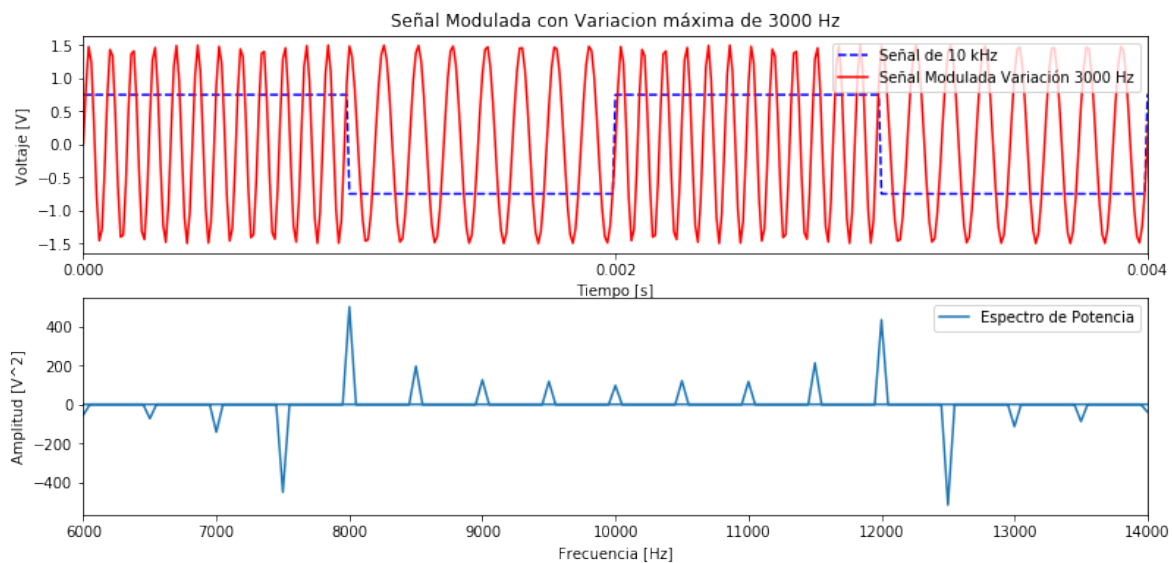


Figura 14: Modulación de señal con $\Delta f = 3000 \text{ [Hz]}$

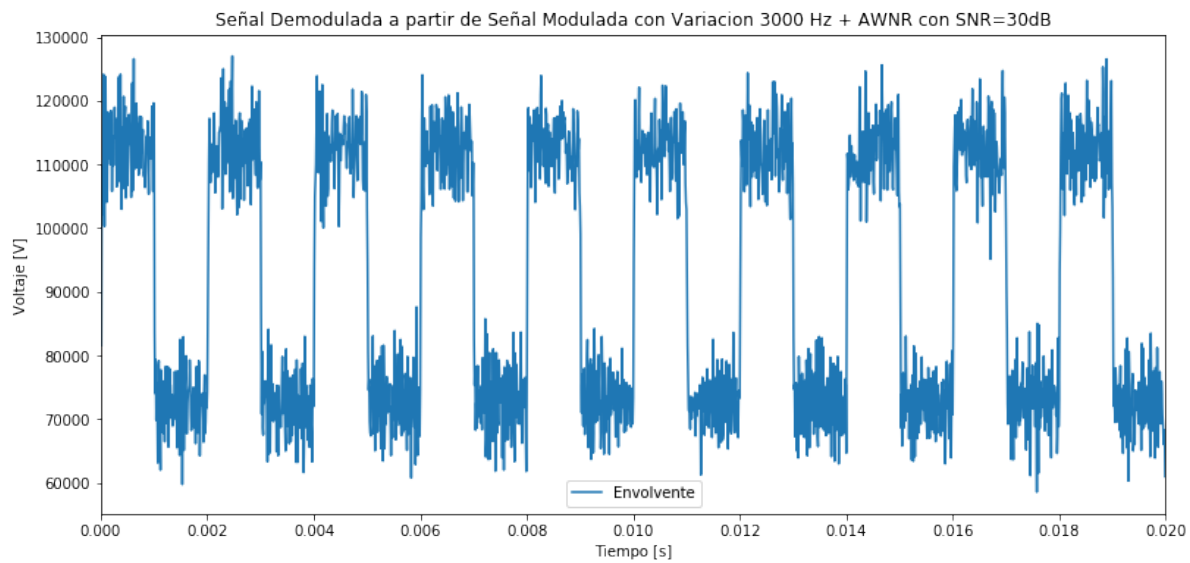


Figura 15: Demodulación de señal con $\Delta f = 3000 [Hz]$ y SNR de 30 dB

6. Audios de Comparación

Finalmente, para realizar una comparación auditiva entre las demodulaciones, se crearon 3 archivos de audio. En primer lugar, “**originalSignal.wav**” corresponde a la señal original que se transmitió, correspondiente a una señal cuadrada de 500 [Hz].

En segundo lugar, se eligió la mejor demodulación obtenida de las que fueron pedidas, que corresponde a la señal modulada con $\Delta f = 100 [Hz]$ y $SNR = 30 [dB]$ se adjunta en el archivo “**demodSignal100_30.wav**”.

Por último, se adjunta la señal demodulada de la modulación propuesta que corresponde a $\Delta f = 3000$ y $SNR = 30 [dB]$ en el archivo “**demodSignal3000_30.wav**”. Esto se realizó utilizando la sección de código 7.

En estos audios, es posible apreciar que la señal demodulada con $\Delta 100 [Hz]$ es prácticamente sólo ruido, y que la señal con modulación $\Delta 3000 [Hz]$ es reconocible y similar a la señal original, solo que un poco mas ruidosa y con menor amplitud, debido que al al hacer un escalado de la señal, se tomaron valores del ruido.

Código 7: Código utilizado para crear archivos de audio.

```
1 from scipy.io.wavfile import write
2
3 originalSignalNormaliz=np.int16(signal_500/np.max(np.abs(signal_500)) * 32767)
4 demodSignal100_30Normaliz=np.int16(envolv100_30/np.max(np.abs(envolv100_30)) * 32767)
5 demodSignal3000_30Normaliz=np.int16(envolv3000_30/np.max(np.abs(envolv3000_30)) * 32767)
6 for i in range(10):
7     originalSignalNormaliz=np.concatenate((originalSignalNormaliz,originalSignalNormaliz)
8     )
9     demodSignal100_30Normaliz=np.concatenate((demodSignal100_30Normaliz,
10     demodSignal100_30Normaliz))
```

```
9     demodSignal3000_30Normaliz=np.concatenate((demodSignal3000_30Normaliz,  
10     demodSignal3000_30Normaliz))  
11 write('originalSignal.wav', 44100, originalSignalNormaliz)  
12  
13 write('demodSignal100_30.wav', 44100, demodSignal100_30Normaliz)  
14  
15 write('demodSignal3000_30.wav', 44100, demodSignal3000_30Normaliz)
```

7. Conclusiones

Fue posible concluir que la modulación en frecuencia es mucho mas compleja que la modulación en amplitud, pues es mas difícil introducir la variación de frecuencia en una señal portadora en comparación con realizar una simple multiplicación entre moduladora y portadora, como se hace en modulación AM.

Se comprobó además que al menos para este tipo de modulación, el ruido juega un papel muy importante y por lo tanto, sería deseable que el ruido fuera casi inexistente, osea, que la relación entre la potencia de la señal modulada y la del ruido fuera muy grande.

Finalmente, se pudo comprobar que una mayor variación en la frecuencia permite una mejor demodulación bajo una misma proporción de potencias (SNR).