

ANÁLISIS DE LAS MODULACIONES ANALÓGICAS

Estudio de su impacto en los radiofaros VOR

Eduard Algar
CA1

Índice:

Introducción	1
Funciones	1-2
Aplicación de las funciones implementadas	2-14
Que no hacer para obtener valores erróneos.....	14-15
Análisis de la señal generada por una estación VOR	15-18
Conclusiones	18
Bibliografía	19

Introducción:

El objetivo de este proyecto es analizar la evolución de una señal aplicando diferentes técnicas de modulación usando MATLAB. Así como contrastar los resultados provenientes de la experimentación del código con los esperados de la teoría. Para finalizar veremos cómo se comporta una señal transmitida por un VOR usando las herramientas previamente trabajadas.

Funciones:

Para trabajar de manera ordenada y eficiente, emplearemos funciones de creación propia. Esto permite tener un código más limpio y poder reaprovechar las funciones para otros ejemplos:

Modulación FM:

Aquí creamos la modulación FM que nos devuelve la función ya modulada, pasando los parámetros de la entrada, trabajo con la fd para hacer el caso lo más genérico posible.

Para la integral, se aproxima mediante el método numérico de primer orden de suma de rectángulos.

```
function [y] = modulacion_FM(t, Tm, moduladora, A_fm, f0, fd)
integral = cumsum(moduladora)*Tm;
y=A_fm*cos(2*pi*f0*t+2*pi*fd*integral);
end
```

Modulación AM:

Como en el caso anterior modelizamos la expresión continua de la modulación aplicándola en un conjunto discreto de tiempos

```
function [y] = modulacion_AM(x, t, A, m, f_port)
y=A*(1+m*x).*cos(2*pi*t*f_port);
end
```

Demodulación FM:

Para calcular la señal demodulada se hace la derivada (aproximación de primer orden) y luego calculamos su envolvente

```
function [up, lo] = demoduladorFM(y, Tm)
derivada = (y(1:end-1)-y(2:end))/Tm;
[up, lo]=envelope(derivada);
end
```

Demodulación AM:

Para la AM es tan sencillo como calcular directamente su envolvente usando `envelope()`

Esto nos dará la señal inicial más una continua pudiéndose esta filtrar o restar para así recuperar íntegramente la señal inicial.

Plot Fourier:

Para graficar la transformada de Fourier usamos la `dft`, el problema que nos encontramos, es que esta solo trabaja en la parte real, haciendo una simetría de las dos partes. Es necesario manipularla mediante `fftshift()` y centrar el tiempo la posición central de nuestro vector tiempo en 0 de tal forma que los dos extremos sean equidistantes.

Realmente lo que estamos obteniendo es la gráfica de la densidad espectral de potencia, ya que la FFT nos da valores complejos, que no se pueden graficar en MATLAB, así lo que hacemos es aplicar el valor absoluto y elevarlo al cuadrado lo que nos lleva a la definición de la densidad espectral de potencia.

```
function PlotFourrier(Tm,y,escala)
n= length(y);
Y = fftshift(fft(y));
fs=1/Tm;
fshift = (-n/2:n/2-1)*(fs/n); % zero-centered frequency range
powershift = abs(Y).^2; % zero-centered power
figure
plot(fshift,powershift)
end
```

Aplicación de las funciones implementadas:

Una vez que tenemos todas las funciones necesarias para realizar el proyecto comenzaremos por hacer una prueba, ploteando cada resultado.

Empezaremos por definir las variables y realizar las llamadas a funciones pertinentes:

seleccionamos un periodo de muestreo de 10^{-7} s ya que con experimentación se ha comprobado que con periodos más pequeños aumenta dramáticamente el tiempo de ejecución. La señal será muestreada durante un segundo:

```
Tm = 1e-7;
t= 0:Tm:1;
f_ref =9960;f_m=30;f_c=10;
v=0;m=1;A=1; %Parametros de fase del tono puro y modulacion de la señal am/fm

x= cos(2*pi*f_m*t);
y= modulacion_FM(t,Tm,x,1,f_ref,16*f_m);
x2= demoduladorFM(y,Tm);
y2 = modulacion_AM(x,t,1,0.3,108e6);
```

seguidamente mostramos los resultados obtenidos para determinar el correcto funcionamiento del código:

```
%plot coseno
figure
plot(t,x)
xlabel('t [s]')
ylabel('A[V]')
title('coseno')
```

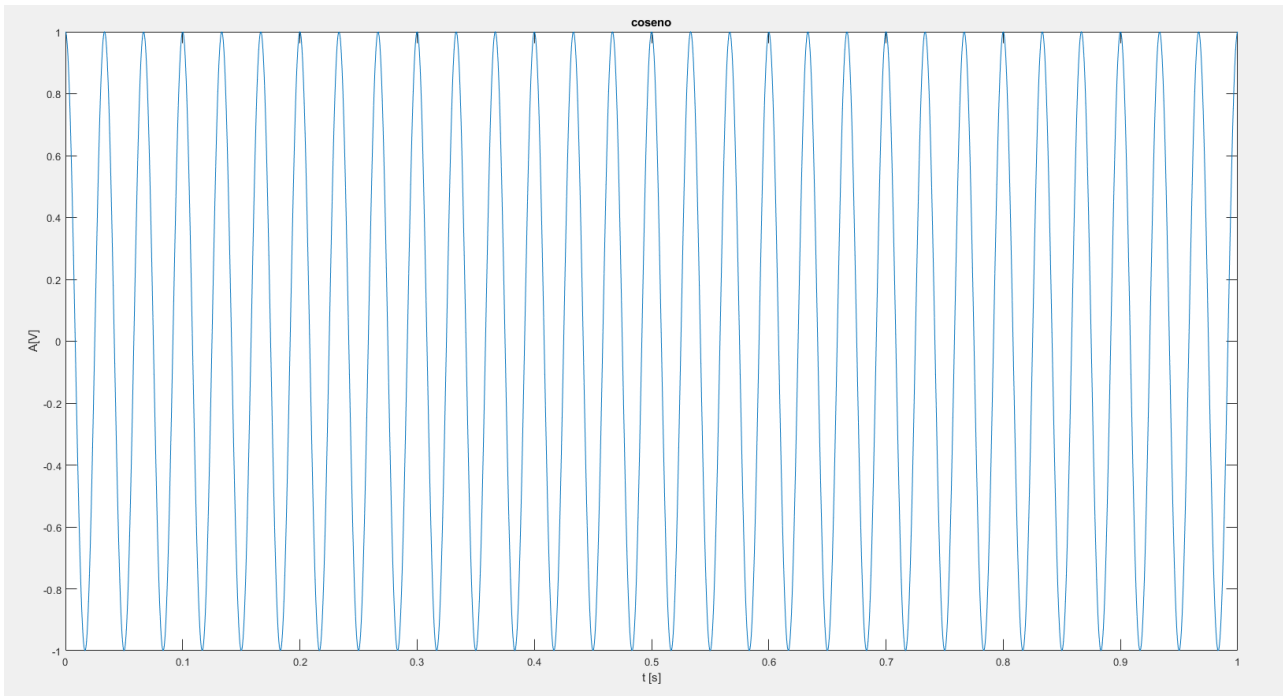


figura 1 gráfica del tono a 30Hz

```
PlotFourrier (Tm, x, 'Coseno')
```

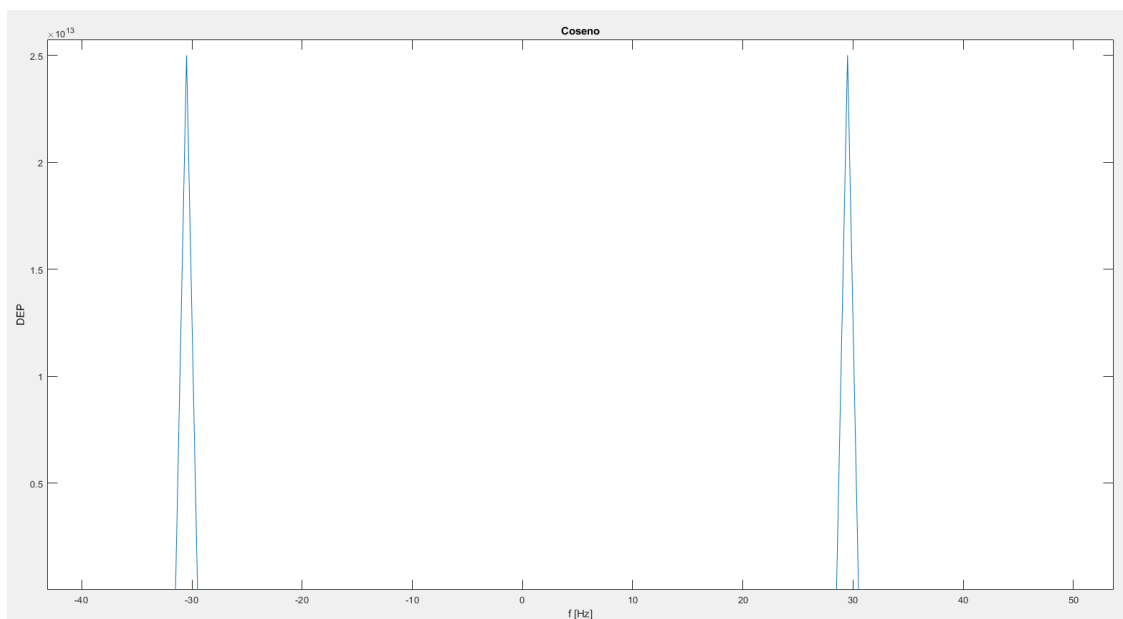


figura 2 Densidad espectral de potencia del tono

La señal obtenida, es igual que la esperada, seguimos el hilo comprobando que la FFT y la fftshift funcionan correctamente, para este caso deberíamos obtener 2 deltas en 30 y -30 Hz. Como podemos observar la función funciona correctamente, ahora procederemos a comprobar las modulaciones analógicas:

```
%plot FM + demodulación
figure
plot(t.^-1,y)
hold on
tff = t(1:end-1);%elimino la ultima posicion ya que la derivada me saca
el ultimo valor.
plot(tff.^-1,x2)
xlabel('f [Hz]')
ylabel('A[V]')
hold off
title('señal modulada y demodulada en fm')
legend('Modulación FM','Demodulación')
```

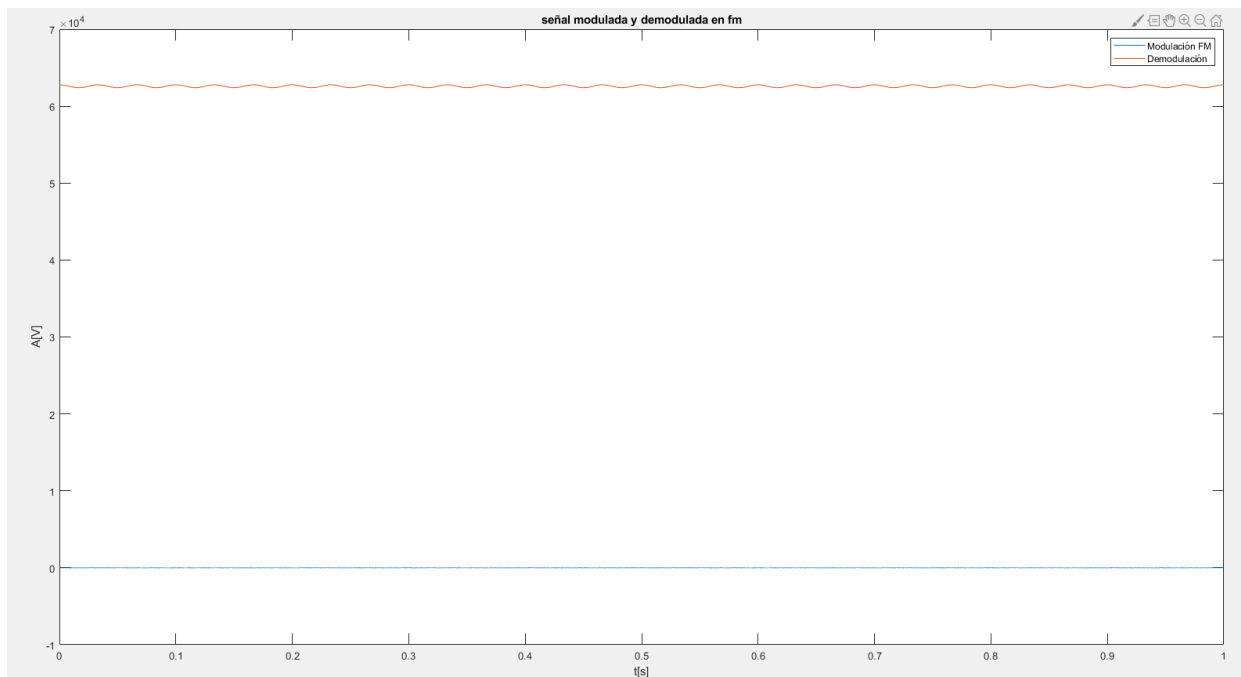


figura 3 Comparación de la señal modulada y demodulada

A primera vista el resultado anterior, puede parecer incorrecto pero hay que fijarse en los ordenes de magnitud, si modificamos el código anterior:

```
%plot FM + demodulación
figure
plot(t,y)
hold on
tff = t(1:end-1);%elimino la ultima posicion ya que la derivada me saca
el ultimo valor.
plot(tff,x2)
hold off
xlabel('t[s]')
ylabel('A[V]')
ylim([-1.5,1.5])
title('señal modulada y demodulada en fm')
```

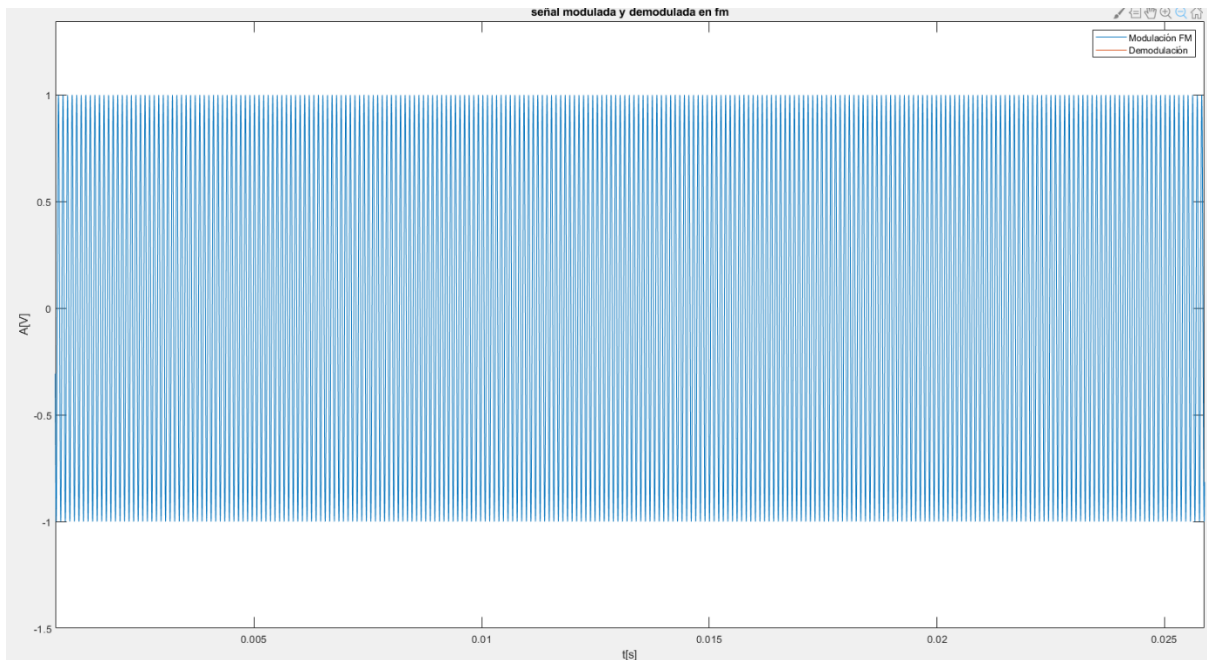


figura 4 zoom de la figura3 con el eje y centrado entre -1.5 y 1.5

Vemos que realmente es un valor correcto y tiene las propiedades esperadas de la señal modulada.

Seguiremos analizando las transformadas de Fourier de la modulación FM:

```
PlotFourrier(Tm, y, 'Modulación FM')
```

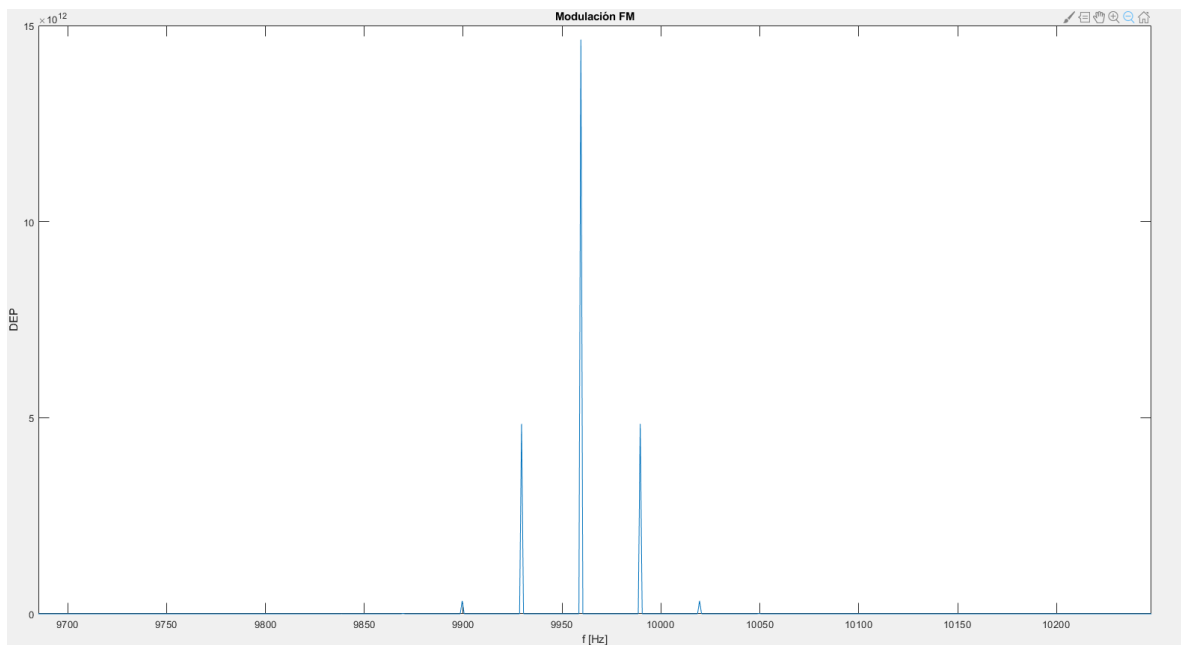


figura 5 transformada de Fourier de la señal FM (solo se muestra la parte frecuencial positiva, a -9960Hz tenemos la imagen de esta)

Finalmente, nos queda comprobar las funciones referentes a la AM:

```
% Comparar la modulación con la señal original AM
figure
plot(t,y2)
hold on
plot(t,x)
hold off
title('Señal AM vs original')
xlabel('t[s]')
ylabel('Amplitud')
legend('AM', 'original')
```

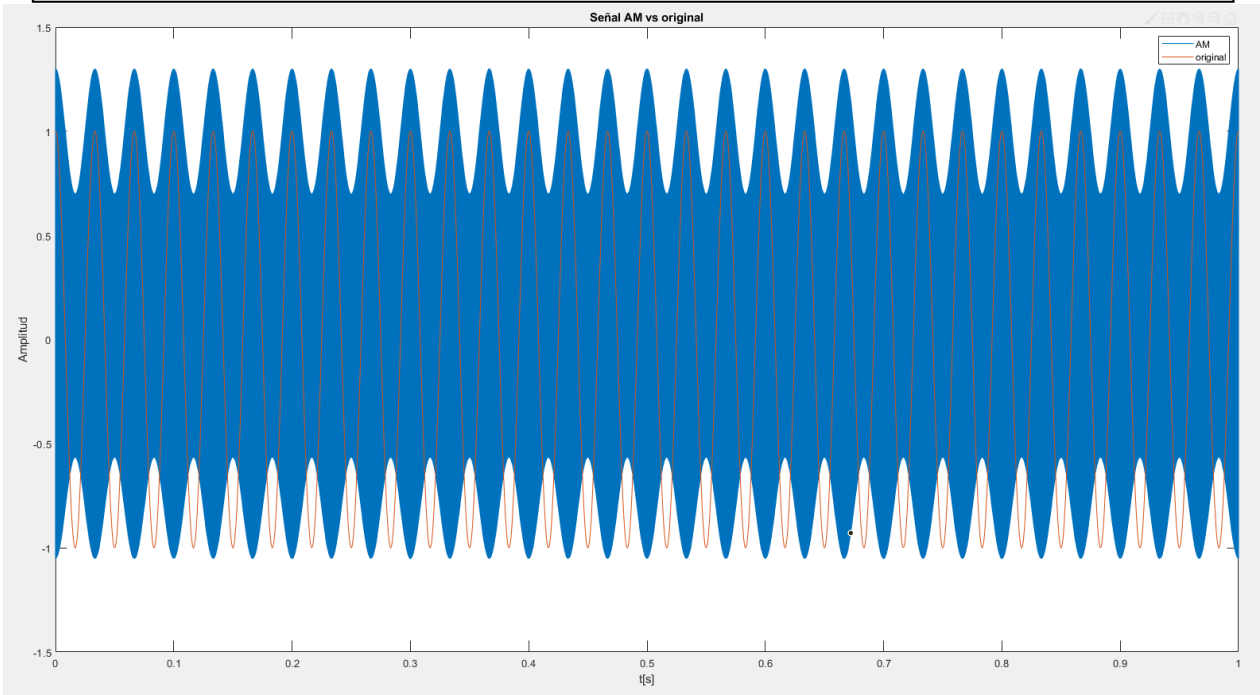


figura 6 señal antes y después de ser modulada, como era de esperar, al ser una modulación en amplitud, la frecuencia se mantiene con un valor constante y la amplitud varia.

```
% TF AM
PlotFourrier(Tm,y2, 'Modulación AM')
```

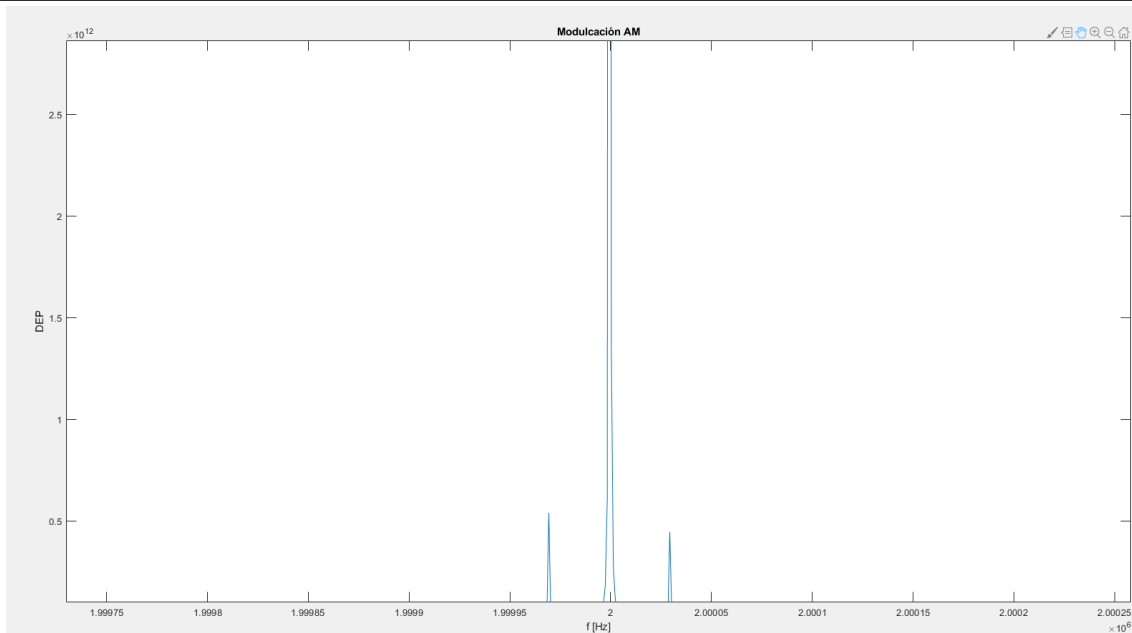
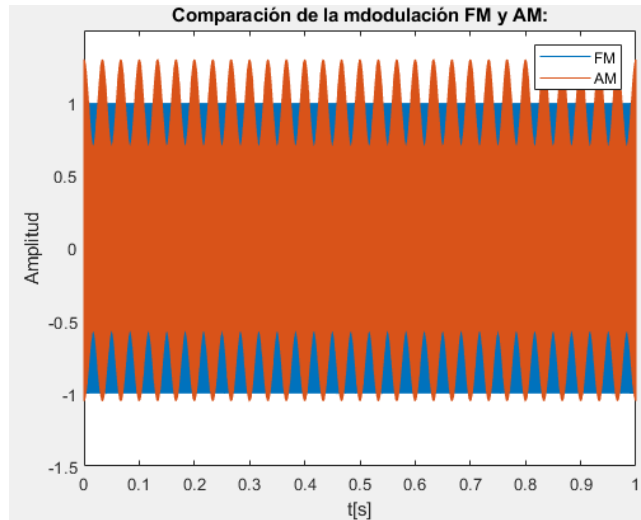


figura 7 Transformada de Fourier de una señal AM (falta su imagen), se puede observar la delta de la portadora, así como la señal cosenoidal, también se pueden apreciar los efectos de trabajar con una señal finita y un enventanado.

Finalmente, podemos comprobar lo que ya se ha dicho antes sobre la diferencia de la modulación en amplitud y frecuencia, de las que hablaremos más en detalle en el futuro:

```
%comparación de la mdodulación
FM y AM:
figure
plot(t,y)
hold on
plot(t,y2)
hold off
title('Comparación de la
mdodulación FM y AM:')
xlabel('t[s]')
ylabel('Amplitud')
legend('FM','AM')
```



Así pues, vemos que todos los elementos funcionan correctamente, lo que nos permite solucionar problemas más complejos de forma más fácil y eficiente, ya que no tenemos que volver a escribir código cada vez al mismo tiempo que podemos descomponerlos en trozos más sencillos.

A continuación, discutiremos al detalle cada una de las modulaciones:

Modulaciones de Amplitud:

Como ya hemos introducido anteriormente, se caracterizan por que la señal de información (moduladora) altera la amplitud de la portadora:

El esquema básico de esta modulación se compone de un mezclador y un sumador donde la portadora y la moduladora son multiplicadas y luego la portadora es añadida mediante el sumador:

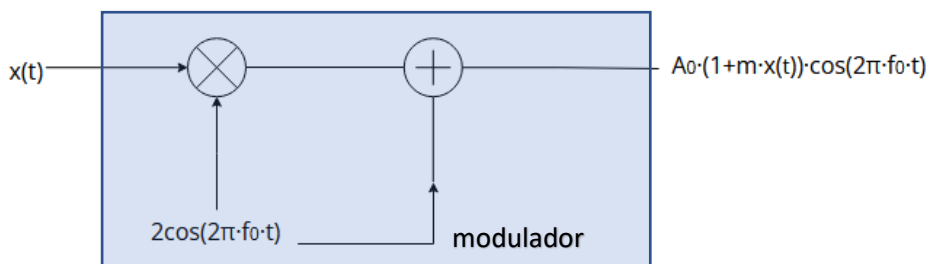


figura 8 diagrama de bloques de una modulación AM

Por la propiedad de la transformada de Fourier [modulación] (recordemos que el coseno son dos deltas) la señal se ve desplazada arriba y debajo de la f_0 , en f_0 tendremos una delta correspondiente a la portadora.

Al operar y sacar factor común nos queda la expresión a la salida del modulador que es lo que se observa en la figura 6. Existen variaciones de esta modulación las cuales no trataremos en profundidad ya que se salen del objetivo de este proyecto, tendríamos los siguientes casos:

La doble banda lateral DBL, donde se elimina la portadora para ahorrar energía y la banda lateral única BLU, que solo transmite la mitad de la DBL ahorrando así ancho de banda del canal.

Otro aspecto importante a tener en cuenta de la modulación AM es el fenómeno de sobre modulación, que se produce cuando la amplitud de la señal es inferior a 0:

$$A_m(1 + m \cdot x(t)) > 0$$

condición para evitar la sobremodulación

Para comprobar dichos efectos, trabajaremos con el tono a 30Hz (figura 1), al tener amplitud unitaria, la $x(t)$ mínima tiene el valor de -1V, luego para cumplir con la condición anterior $m \in (0,1)$. Para este caso usaremos una portadora ubicada en la banda de comunicaciones aeronáuticas con valor de 122.8MHz:

```
% Comparar la modulación con la señal original AM
figure
plot(t,y2)
hold on
plot(t,x)
hold off
title('Señal AM vs original')
xlabel('t[s]')
ylabel('Amplitud')
legend('AM', 'original')
%Comparar señal vs demodulación
figure
plot(t,x3)
hold on
plot(t,x)
hold off
xlabel('t[s]')
ylabel('Amplitud')
legend('señal demodulada', 'señal original')
```

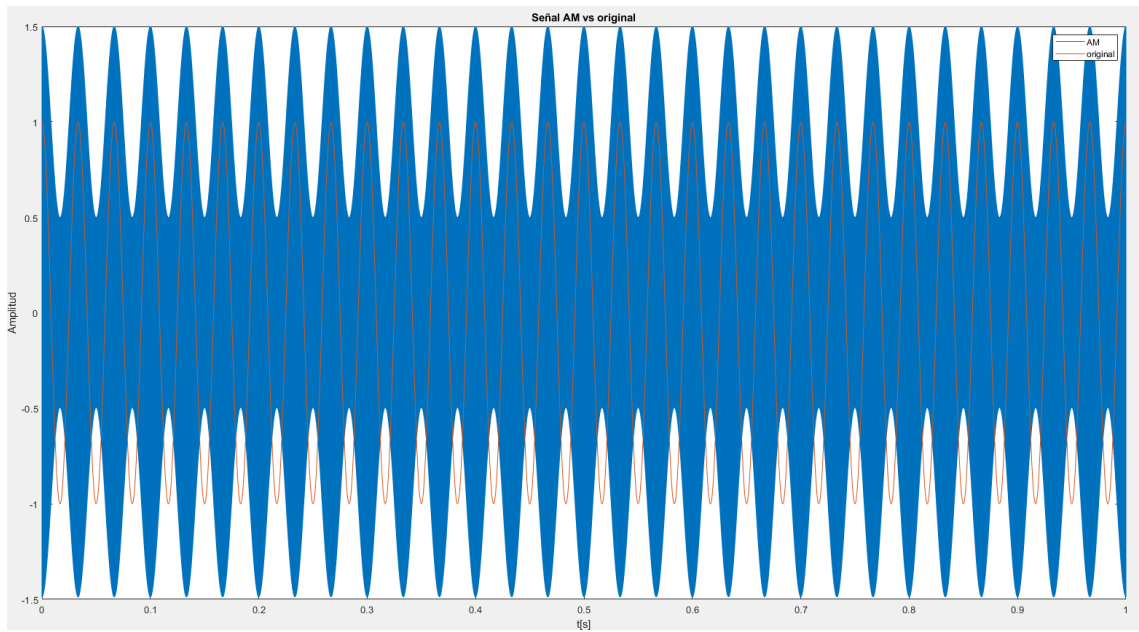


figura 9 comparación entre la señal a la entrada y salida del modulador, esta figura es muy parecida a figura6, se muerta para posteriormente evaluar el efecto de la sobre modulación.

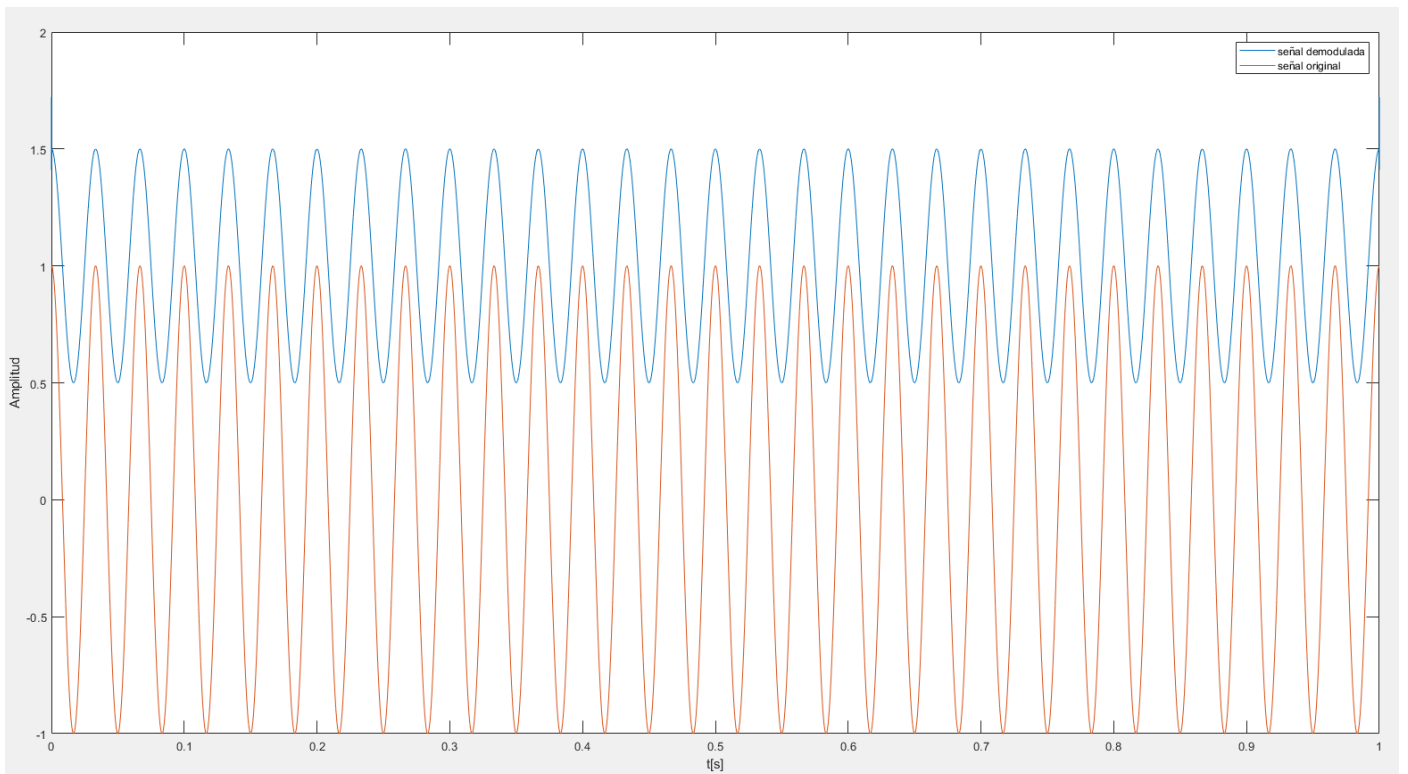


figura 10 como se puede observar la señal demodulada es la misma que la original, más una continua y con la misma amplitud

Ahora realizaremos el mismo procedimiento, pero cambiando el índice de modulación por 2:

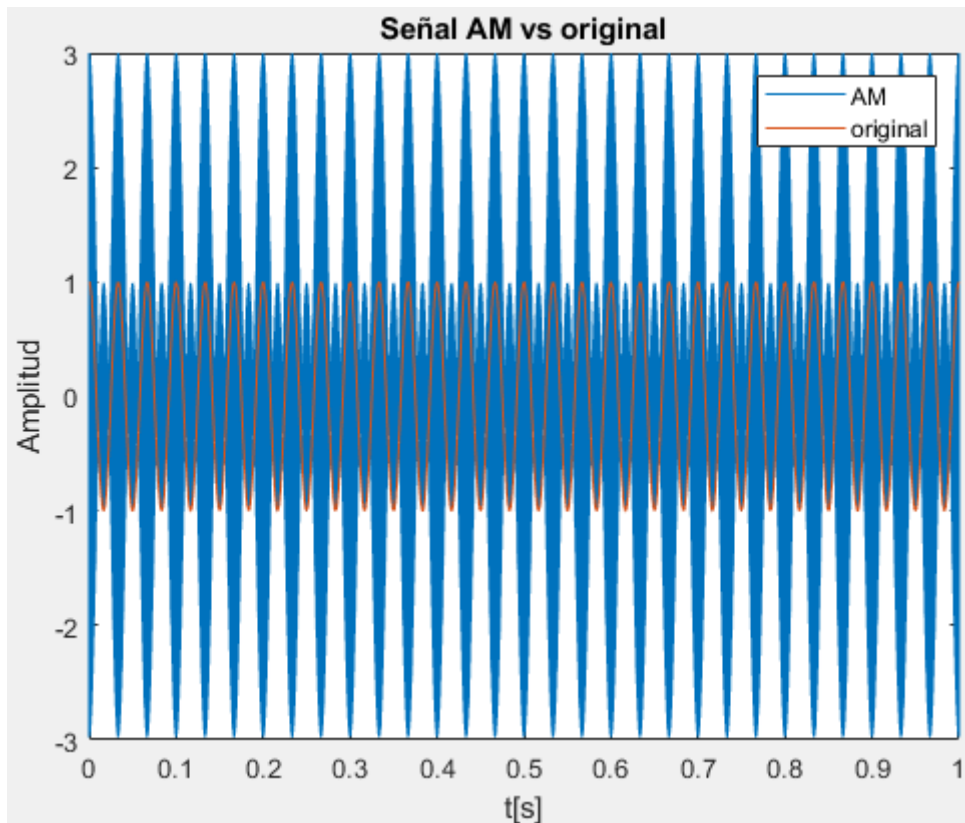


figura 12 señal original VS señal modulada, se puede apreciar claramente el fenómeno de sobre modulación, parte de la amplitud inicialmente positiva pasa a ser negativa y viceversa.

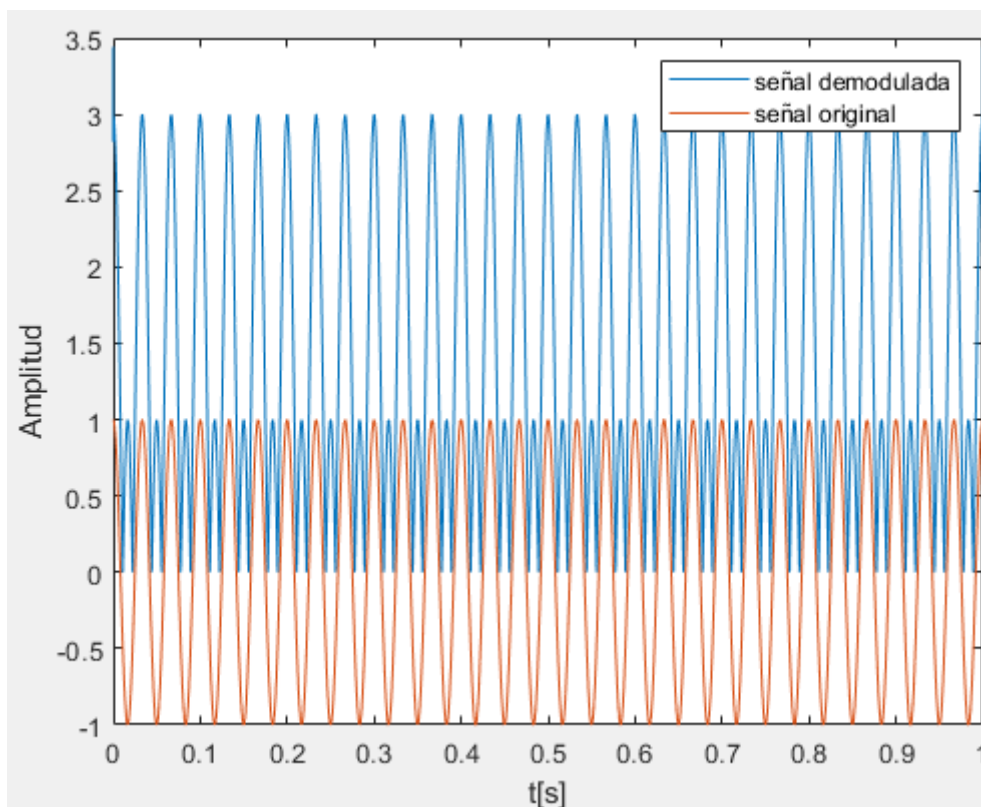


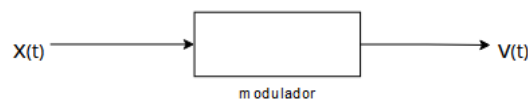
figura 11 podemos observar como la sobre modulación provoca que el demodulador no sea capaz de devolver la señal original, arrojando así una señal errónea.

Modulación FM:

Una vez tratada las modulaciones en amplitud, pasaremos a ver la modulación en frecuencia:

Esta modulación se beneficia de una resistencia mayor frente a las interferencias ya que se beneficia del efecto de captación siempre y cuando la SNR a la entrada sea superior a 10dB. También tiene una mejor relación-síñal a ruido en comparación a la AM.

Por otro lado, el espectro de la FM es teóricamente infinito, nos tenemos que apoyar en la regla de Carson para poder eventanar la síñal, otro inconveniente es la complejidad de los circuitos que precisa, aunque con los avances tecnológicos ya no supone un problema.



$$v(t) = A_0 \cdot \cos(2\pi \cdot f_0 \cdot t + 2\pi \cdot f_d \cdot \int_0^t x(\tau) d\tau)$$

Si nos detenemos a observar la expresión anterior podemos concluir que existe una frecuencia instantánea equivalente a:

$$f_i = f_0 + f_d \cdot x(t)$$

f_d tiene unidades de Hz/V e indica la variación de la frecuencia instantánea de una síñal cuando más alta sea más modulada se encontrará la síñal.

Otro parámetro clave que podemos obtener es el índice de modulación/relación de desviación, que indica cuanto se ve afectada la frecuencia portadora por la síñal de información:

$$D = \frac{\max(x(t)) \cdot f_d}{W} = \frac{\Delta f}{W}$$

De aquí podemos obtener el ancho de banda de la síñal modulada siguiendo la regla de Carson:

$$B_T = \begin{cases} 2 \cdot (\Delta f + W) & D > 0.4 \\ 2 \cdot W & D \leq 0.4 \end{cases}$$

Otra forma para determinar el ancho de banda requerido es usando la gráfica de Schwartz, sabiendo el índice de modulación:

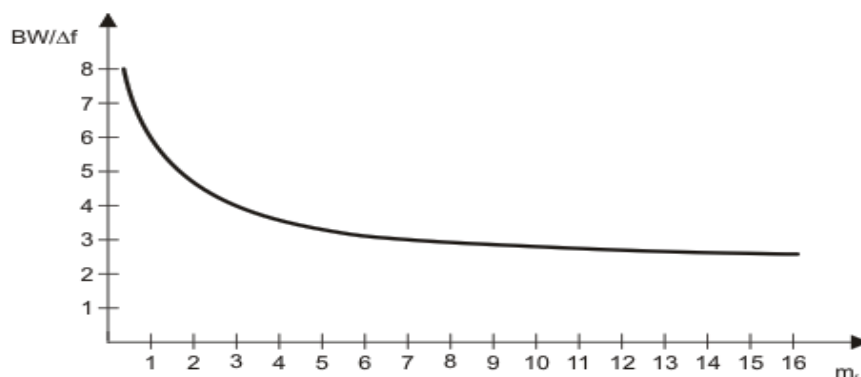


figura 13 función de Schwartz que relaciona el índice de modulación con el ancho de banda

De primeras se puede ver que aumentar la fd, implica tener un ancho de banda más amplio, a continuación, estudiaremos los efectos de variar la fd en MATLAB:

```
%Comparar las fd en la FM:
fd=10;
figure
while fd<= 1e4
    %ta= reshape(t,int32(size(t)/2),[]);
    y= modulacion_FM(t,Tm,x,1,f_ref,fd);
    ta=t(1:int32(length(t)/8));
    y=y(1:int32(length(y)/8));
    plot(ta,y)
    hold on
    fd=fd*10;
end
hold off
xlim([0,1e-4])
legend('10Hz', '100Hz', '1KHz', '10KHz', '100KHz')
title('FM')
xlabel('t[s]')
ylabel('Amplitud')
%evaluar los efectos de la fd en la demodulación
figure
fd=10;
while fd<= 1e4
    %ta= reshape(t,int32(size(t)/2),[]);
    y= modulacion_FM(t,Tm,x,1,f_ref,fd);
    x2= demoduladorFM(y,Tm);
    plot(t(1:end-1),x2)
    hold on
    fd=fd*10;
end
hold off
title('Demodulador FM')
xlabel('t[s]')
ylabel('Amplitud')
legend('10Hz', '100Hz', '1KHz', '10KHz', '100KHz')
```

```
figure%zoom de la figura anterior
fd=10;
while fd<= 1e4
    %ta= reshape(t,int32(size(t)/2),[]);
    y= modulacion_FM(t,Tm,x,1,f_ref,fd);
    x2= demoduladorFM(y,Tm);
    plot(t(1:end-1),x2)
    hold on
    fd=fd*10;
end
hold off
title('Demodulador FM')
xlabel('t[s]')
ylabel('Amplitud')
legend('10Hz', '100Hz', '1KHz', '10KHz')
xlim([0,0.5])
```

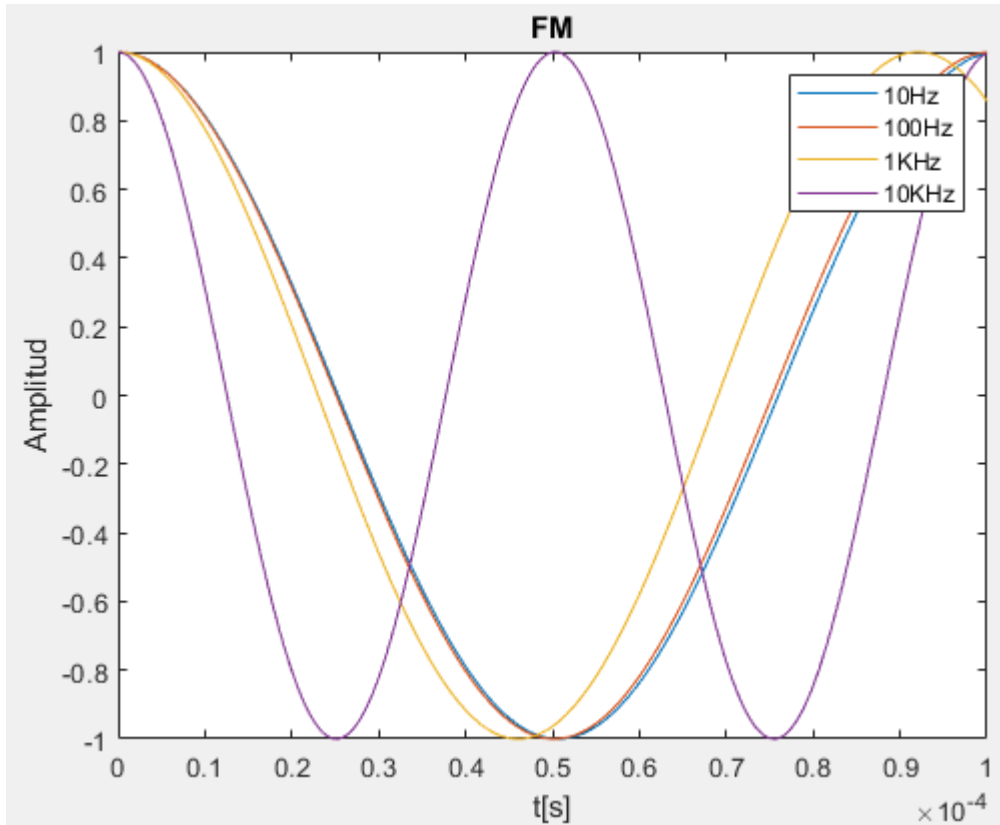


figura 14 comparación de la evolución de la modulación FM para una misma señal con distintas f_d

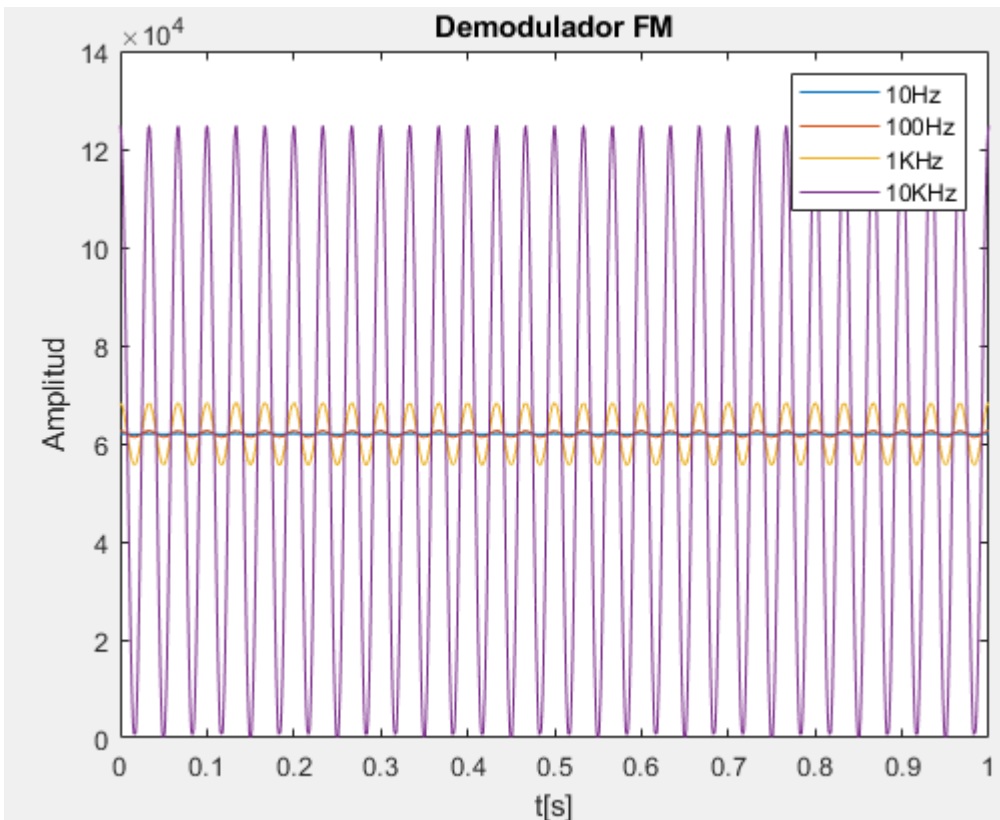


figura 15 Comparación de la demodulación FM para una misma señal con distintas f_d

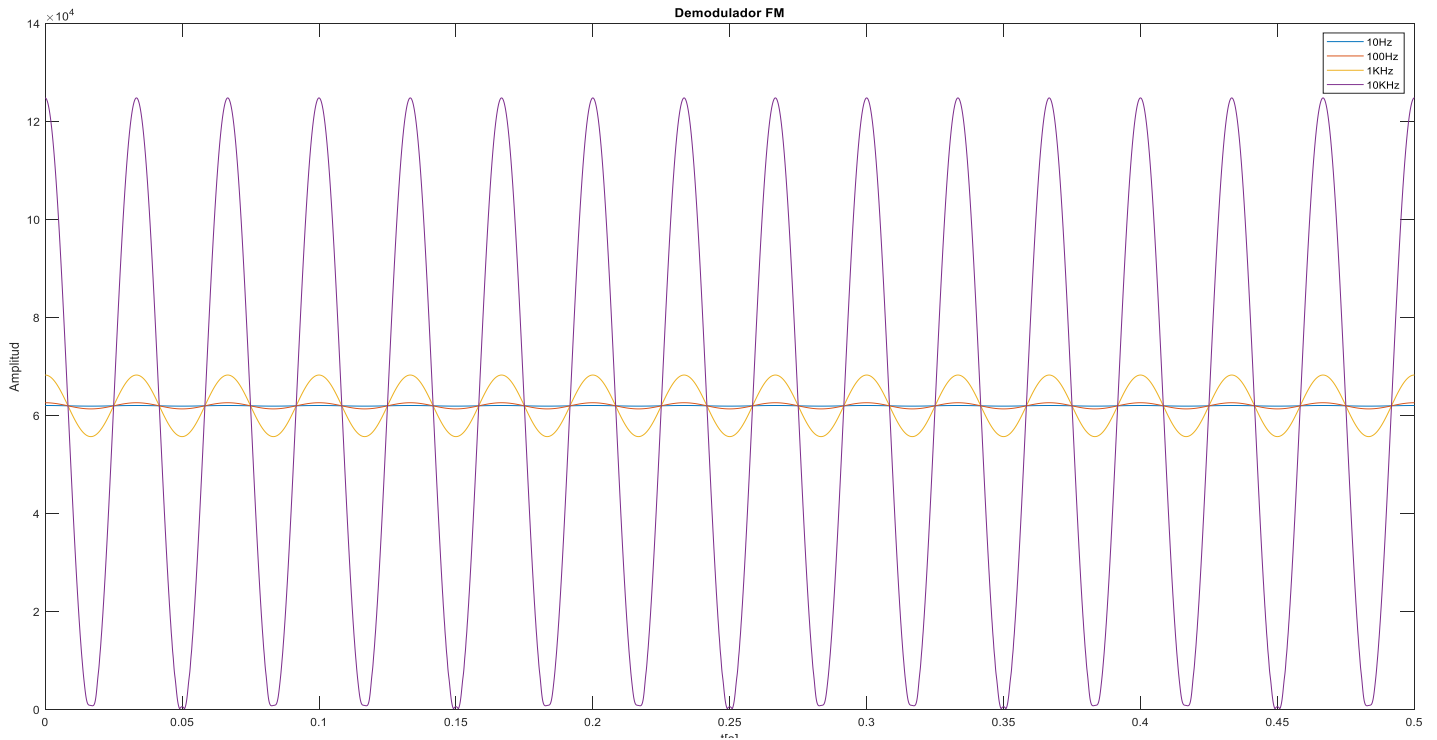


figura 16 zoom de la figura 15

Se puede observar que, a parte de los efectos mencionados anteriormente, la señal una vez demodulada tiene una amplitud mayor a medida que f_d es mayor.

Otro efecto interesante es como la señal modulada varía en periodo, es decir: a medida que f_d aumenta, esta sufre un avance en el periodo haciéndolo más pequeño.

Que no hacer para obtener valores erróneos:

Un efecto que me resultó interesante mientras realizaba la práctica, es cuando aun cumpliendo con el teorema de Nyquist, si trabajamos a frecuencias cercanas al tiempo de muestreo, obtenemos resultados erróneos que carecen de sentido, en particular si en el ejemplo anterior intentamos modular y luego demodular a frecuencias más altas:

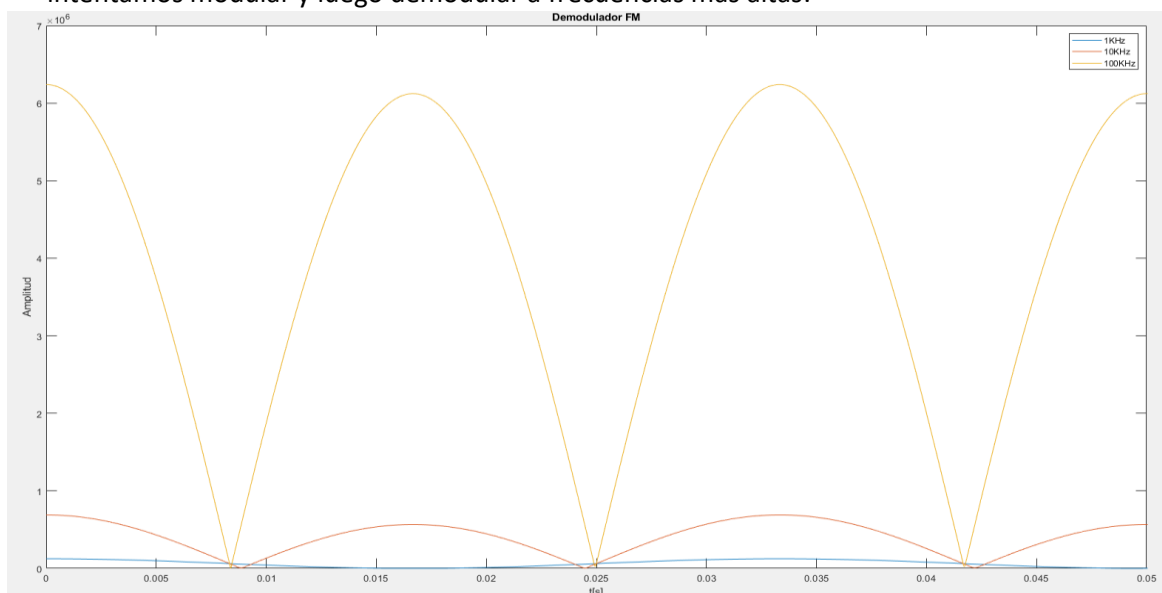


figura 17 errores producidos en la demodulación debidos a trabajar por encima del umbral de frecuencias permitidas.

El motivo reside en que se carecen de puntos cuantitativos que permitan “dibujar la señal, uniendo puntos”, experimentalmente se ha llegado a la conclusión de que para obtener un resultado aceptable hay que trabajar con frecuencias de dos órdenes de magnitud por debajo del tiempo de muestreo.

Alternativamente se podría evitar el problema, disminuyendo el tiempo de muestreo, pero para un $T_m = 1e-9$, en un segundo de señal se obtienen variables con un espacio en memoria RAM de 64GB, lo que hace bastante inviable esta alternativa debido a los recursos empleados, así como el tiempo de ejecución.

También se podría intentar solucionar el problema usando un espacio temporal menor.

Análisis de la señal generada por una estación VOR:

Finalmente, analizaremos cómo evoluciona la señal de un radio faro VOR aplicando todo lo visto hasta ahora, usando la información de la colección de problemas:

La señal VOR es transmitida mediante modulación en amplitud donde dentro de esta se encuentra una señal modulada en FM más un tono:

$$x(t, \vartheta) = FM + \text{tono} = \cos(2\pi f_{ref}t + 16 \sin 2\pi f_b t) + \cos(2\pi f_b t + \vartheta)$$

$$s(t, \vartheta) = A(1 + mx(t, \vartheta)) \cos(2\pi f_c t)$$

Empezamos definiendo nuestras variables del problema, así como el tiempo de trabajo el cual reducimos para poder trabajar con frecuencias más altas.

```
Tm = 1e-11;  
t= 0:Tm:0.05;  
f_ref =9860;  
f_b=30;  
f_c=108e6;  
f_d=(16*f_ref)/1;  
radial=0;
```

Creamos nuestra señal compuesta de un tono a 30 Hz modulada en FM más el tono:

```
x= cos(2*pi*f_b*t);  
y= modulacion_FM(t, Tm, x, 1, f_ref, f_d);  
y=y+cos(2*pi*f_b*t);  
y2=modulacion_AM(y, t, A, m, f_c);
```

Entre medio y al final vendrían los plots, que no entraremos en detalle pues es el mismo código que el empleado anteriormente. Como nota final, se ha intentado reducir la ventana de muestreo a 0.035 segundos al mismo tiempo que aumentar la frecuencia de muestreo para evitar los errores que se han comentado en el apartado anterior, aunque no ha sido posible del todo mitigar dichos efectos como se verán en los resultados:

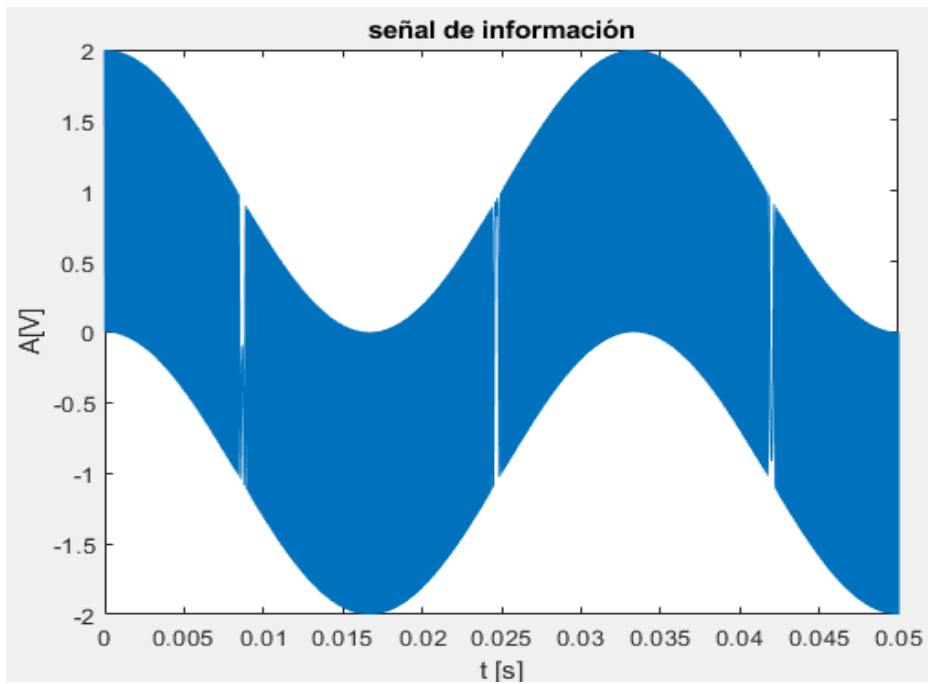


figura 18 señal de información transmitida por un VOR se puede apreciar que, al estar solo un orden de magnitud por debajo de la frecuencia de muestreo, se empiezan a producir errores

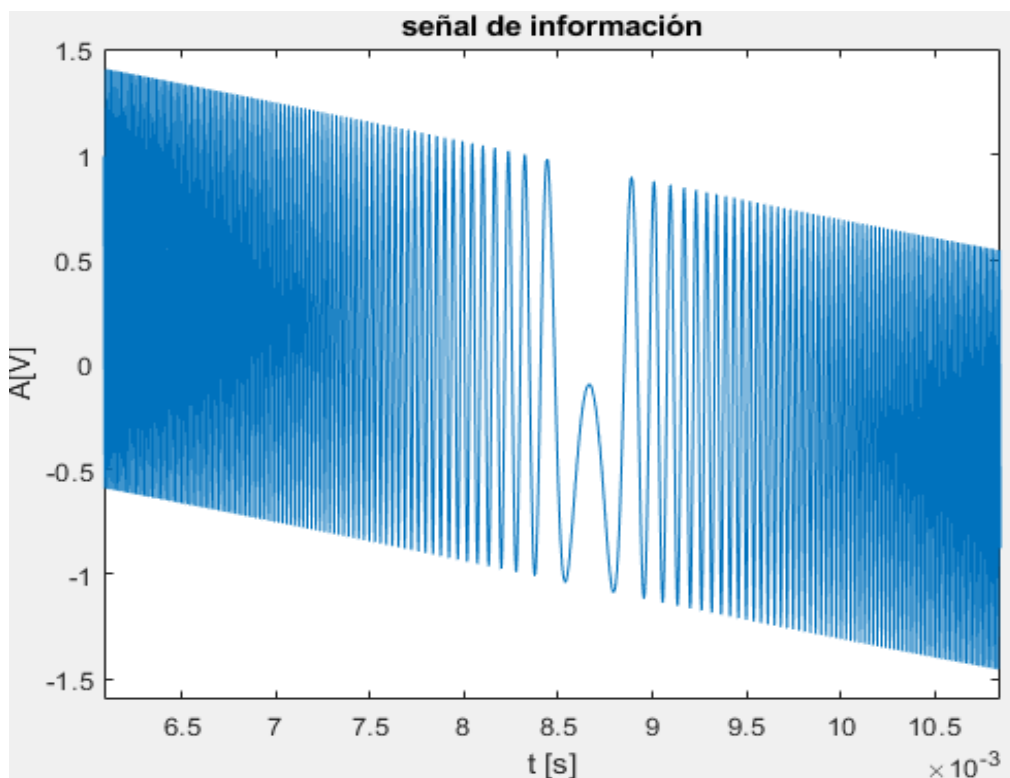


figura 19 ampliación de la figura 18 en la zona "conflictiva"

Como se ha comentado antes, no ha sido posible mitigar del todo el problema pues muestrear una oscilación completa ocupa alrededor de 46 GB de RAM y para hacerlo correctamente, nos vamos a rangos de alrededor de 80 GB de RAM (que se encuentran fuera de las capacidades de la computadora con la que se ha trabajado).

Otro aspecto interesante es cómo la señal de información evoluciona temporalmente pues al ser una modulación frecuencial la amplitud inicialmente se mantendría constante, sin embargo, al estar esta sumada a un tono que contiene información sobre un determinado radial VOR, esta hace que la señal que se debería mantener constante en amplitud oscile de forma cosenoidal con un desfase equivalente al radial seleccionado.

A continuación, observaremos la forma de la transformada de Fourier:

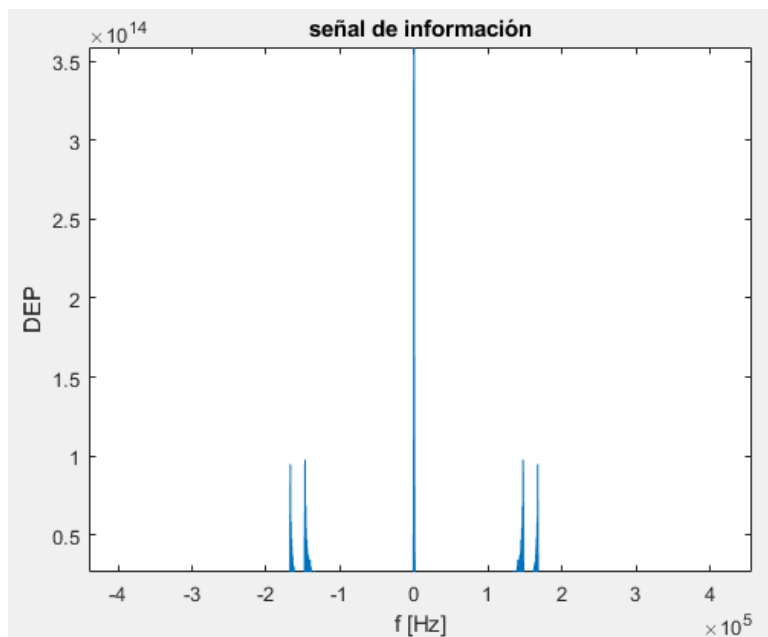


figura 20 transformada de Fourier de la señal de información, se pueden apreciar las deltas a 30 y -30Hz, así como las derivadas

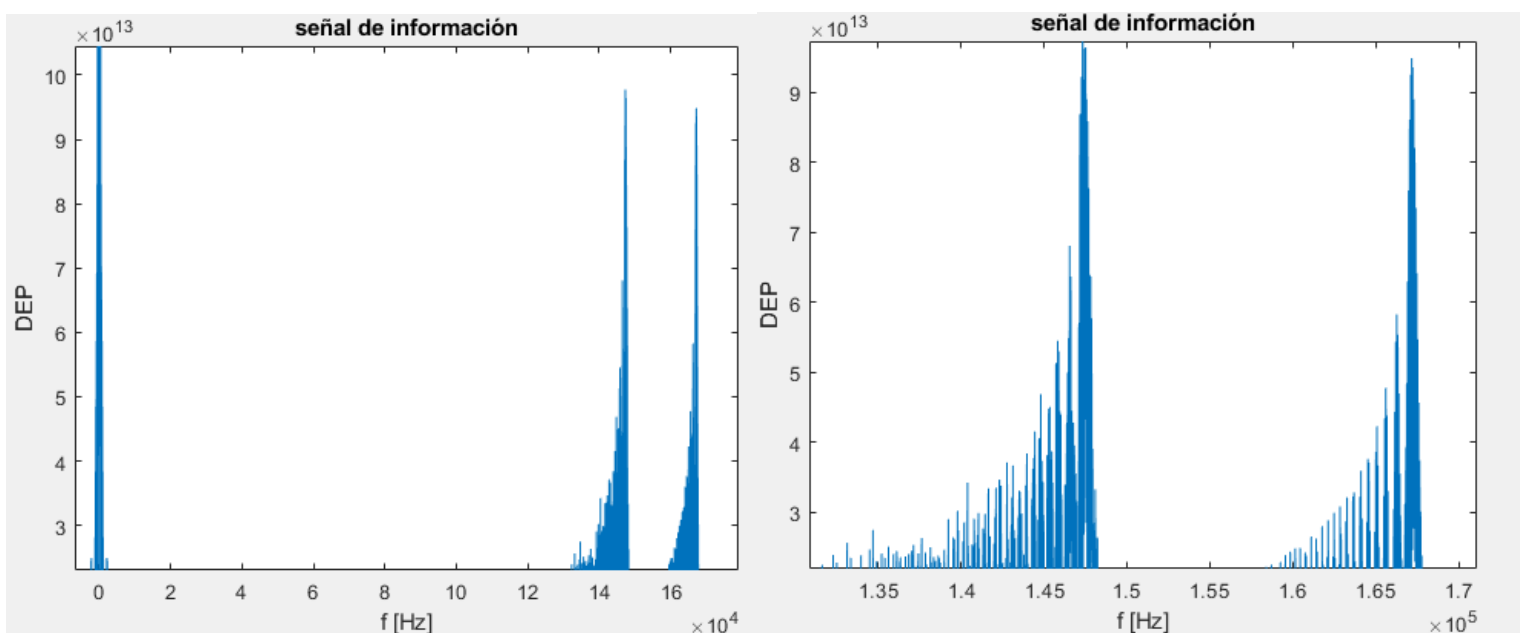


figura 21 ampliación de la figura 20 donde se puede apreciar con más detalle la densidad espectral de potencia proveniente de la FM

Para terminar, observaremos cómo sería la señal que recibe un avión por parte de la estación VOR:

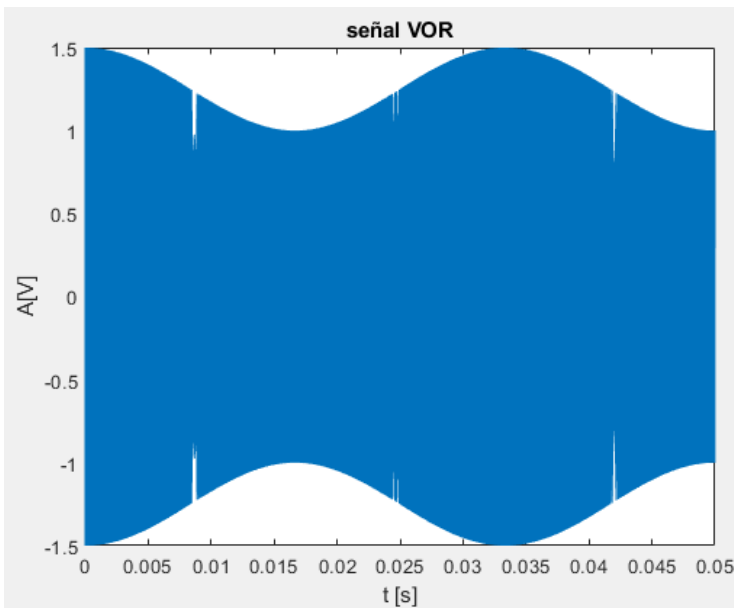


figura 22 señal transmitida por la estación VOR

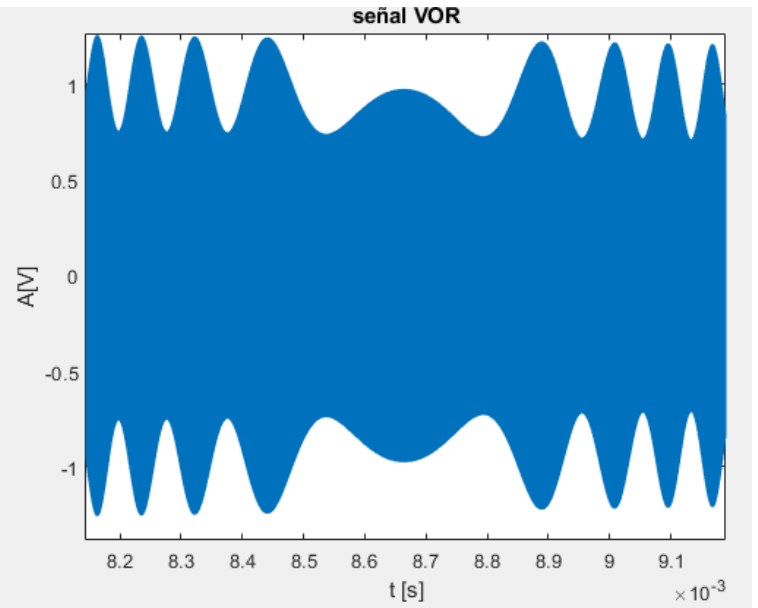


figura 23 ampliación de la figura 22 en la zona conflictiva

Al comportarse la señal de información de forma cosenoidal, era de esperar que la señal transmita tenga un gran parecido a la señal AM modulada con anterioridad a 30Hz.

Conclusiones:

Este proyecto me ha permitido entender mejor como funcionan las modulaciones analógicas. Aprender el funcionamiento de un VOR desde el punto de vista técnico y por último y no menos importante:

Entender las dificultades y retos que hay al procesar señales a altas frecuencias, analizando los requerimientos del tiempo de muestreo y resolución para poder trabajar con señales en la banda de los 100-150 MHz. Viendo así los motivos de los costes elevados de equipos como analizadores de espectros que trabajan a frecuencias elevadas, pues con un computador pensado para trabajar con CFD, métodos numéricos y desarrollo de software, he experimentado limitaciones y ha sido un reto intentar simular estas señales tan elevadas en la última parte de este proyecto.

Bibliografía:

http://openaccess.uoc.edu/webapps/o2/bitstream/10609/69406/5/Sistemas%20de%20comunicaci%C3%B3n%20I_M%C3%B3dulo%202_Comunicaciones%20anal%C3%B3gicas%3B%20modulaciones%20AM%20y%20FM.pdf

<https://www.profesores.frc.utn.edu.ar/electronica/electronicaaplicadaiii/aplicada/cap05ssb.pdf>

<https://online.visual-paradigm.com/drive/#diagramlist:proj=0&dashboard>