

Universidad Nacional de Rosario
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Escuela de Ingeniería Electrónica



Sistemas y Señales I

Trabajo práctico N°1 : Introducción a Matlab

8

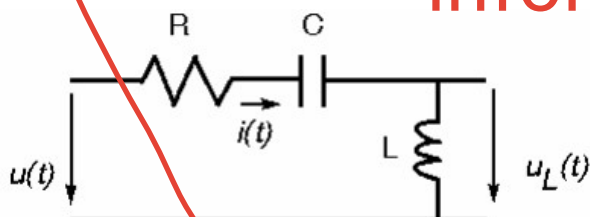
Docentes	
Dr. Juan Carlos Gómez Ing. Franco Del Colle Dr. Ariel Bayá Dr. Gonzalo Sad	

Integrantes del grupo	
BELLINI, Valentin	B-6127/1
GASPOZ, Gastón Leandro	G-5580/8

Fecha de entrega del trabajo	10/05/2021
------------------------------	------------

Problema 1. Circuito RLC.

La guía no es parte del informe.



$$R = 12\Omega$$

$$L = 1\text{mH} = 1 \times 10^{-3}\text{H}$$

$$C = 1\mu\text{F} = 1 \times 10^{-6}\text{F}$$

Figura 1. Circuito RLC serie.

- Escriba la ecuación diferencial ordinaria que describe la dinámica del circuito, tomando como entrada la tensión $u(t)$ y como salida la carga $q(t)$ del capacitor.
- Escriba un sistema de dos ecuaciones diferenciales de primer orden en las variables $x_1(t) = q(t)$ y $x_2(t) = i(t)$, equivalente a la ecuación diferencial obtenida en el apartado a.
- Escriba un script Matlab que implemente la integración numérica de las ecuaciones del apartado b., usando la aproximación de Euler de primer orden de las derivadas $x_1'(t)$ y $x_2'(t)$, y realice la simulación para una entrada pulso rectangular de amplitud 1 V y duración 10 ms en la tensión $u(t)$, a partir de condiciones iniciales nulas en las variables $x_1(t)$ y $x_2(t)$. Realice la simulación con un tiempo total de integración de 20 ms. Utilice el comando `plot` para graficar la tensión en el capacitor $u_c(t)$, la tensión en la inductancia $u_L(t)$ y la corriente por la inductancia $i(t)$, en función del tiempo. Considere los siguientes pasos de integración numérica:

$$\text{c1. } T = \frac{2\pi\sqrt{LC}}{100\sqrt{1 - \frac{R^2C}{4L}}}$$

$$\text{c2. } T = \frac{2\pi\sqrt{LC}}{15\sqrt{1 - \frac{R^2C}{4L}}}$$

Comente sobre los resultados obtenidos.

- Considere ahora entradas senoidales de la forma:

$$u(t) = \sin(2\pi Ft)$$

Como es de su conocimiento (Teoría de Circuitos), cuando el circuito de Fig. 1 se excita con una tensión senoidal, las corrientes y tensiones alcanzan un régimen permanente senoidal, una vez extinguido el transitorio inicial. Con el objetivo de estudiar el comportamiento en frecuencia del circuito (respuesta en frecuencia) realice simulaciones con entradas senoidales de distintas frecuencias y mida la amplificación (atenuación) que introduce el circuito, realizando el cociente entre la amplitud de la salida del mismo (tensión en la inductancia $u_L(t)$) en régimen permanente senoidal y la amplitud de la entrada (tensión $u(t)$). Realice un plot de la amplificación (expresada en decibels (dB)) en función de la frecuencia F en escala logarítmica, es decir:

Activ.

$$20\log_{10}\left(\frac{\text{Amplitud salida}}{\text{Amplitud entrada}}\right) \text{ vs. } \log_{10} F$$

Esta gráfica es la denominada **respuesta en frecuencia de amplitud**. Similarmente se puede medir el desfase entre la senoide de entrada y la señal de salida en régimen permanente senoidal, en función de la frecuencia F , lo que constituye la denominada **respuesta en frecuencia de fase**.

Considere las siguientes frecuencias de la señal de entrada para realizar las simulaciones: 1000 Hz, 2000 Hz, 3000 Hz, 4000 Hz, 4500 Hz, 4700 Hz, 5000 Hz, 5300 Hz, 5500 Hz, 6000 Hz, 8000 Hz y 10000 Hz.

- e. Determine la frecuencia donde la respuesta en frecuencia de amplitud determinada en el apartado d. tiene un máximo. Compárela con la frecuencia $\frac{1}{2\pi\sqrt{LC}}$.

Problema 2: Localización y seguimiento de una fuente sonora

En las últimas décadas, una intensa actividad de investigación se ha llevado a cabo en el área de localización de fuente (*source localization*), principalmente para aplicaciones de radar, sonar, localización en telefonía móvil, sistemas de navegación y posicionamiento global, arreglo de sensores, robótica, interacción humano-computadora, detección y seguimiento de locutor, video conferencias, etc. En particular, la localización de fuente sonora (**SSL: Sound Source Localization**) se ha convertido en una importante área de investigación debido a que la mayoría de los sistemas de reconocimiento de habla, voz, y eventos asumen el conocimiento de la posición de la fuente sonora [3].

Diversas técnicas de SSL han sido propuestas en la literatura. La mayoría de las técnicas se basan en el hecho de que la señal de una fuente sonora se ve sometida a diferentes retardos y atenuaciones cuando alcanza un número de sensores separados espacialmente. Tanto la atenuación como el retardo temporal dependen de la distancia entre la fuente y los sensores.

Una de las técnicas clásicas de SSL es la basada en la Estimación del Tiempo de Retardo (**TDE: Time Delay Estimation**) en un arreglo de micrófonos, que ha probado ser un marco metodológico de alta eficiencia para la localización de fuente sonora. Estas técnicas son también conocidas como **TDOA** (Time Difference of Arrival), o **ITD** (Interaural Time Difference). Otras técnicas para SSL están basadas en mediciones de energía de las señales para localizar la fuente sonora. Una de las más difundidas es la denominada Diferencia de Nivel Interaural (**ILD: Interaural Level Difference**) que mide la diferencia de energía de las señales recibidas por diferentes micrófonos de un arreglo.

Para el caso de SSL en dos dimensiones en ambientes interiores, el uso de técnicas combinadas TDE-ILD permite reducir el número de micrófonos en el arreglo a sólo dos [1],[2]. Típicamente en este caso se asume la hipótesis de campo-lejano (*far-field*) que permite considerar un frente de onda plano de la señal acústica.

En este problema se pretende determinar el ángulo de arribo de la señal de una fuente sonora mediante un arreglo lineal de dos micrófonos en la disposición esquemáticamente representada en la Figura 2. A partir de la señal recibida por los dos micrófonos se deberá computar el ángulo θ de arribo del frente de onda. La distancia L entre los micrófonos se asume conocida.

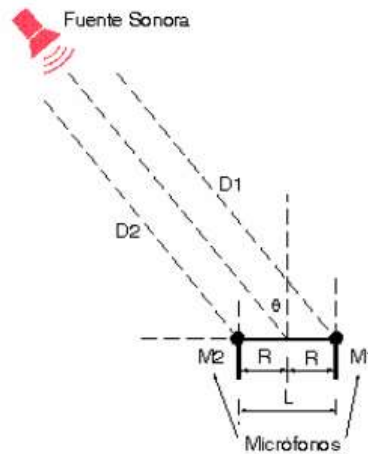


Figura 2: Arreglo de micrófonos y fuente sonora.

- Hallar una expresión para el cálculo del ángulo θ de arribo del frente de onda acústico. Asuma que la velocidad de propagación del sonido en el aire es $c = 343,2 \text{ m/s}$, y que la distancia entre micrófonos es $L = 0,2 \text{ m}$. Note que el ángulo θ depende de la diferencia Δt entre los tiempos de arribo del frente de onda a cada uno de los micrófonos, que a su vez depende de la diferencia $(D1 - D2)$ entre las distancias entre la fuente sonora y cada micrófono.
- Los archivos `datos_1.wav`, `datos_2.wav`, `datos_3.wav`, `datos_4.wav`, `datos_5.wav`, `datos_6.wav`, `datos_7.wav`, `datos_8.wav`, y `datos_9.wav` contienen las señales captadas por ambos micrófonos para 9 posiciones distintas de la fuente sonora. Los archivos pueden cargarse en el espacio de trabajo de Matlab usando el comando `wavread`, con la siguiente sintaxis:

```
[y, Fs] = wavread('datos.wav');
```

donde y es una matriz con dos columnas conteniendo en cada una de ellas la señal de cada micrófono, y donde Fs es la frecuencia de muestreo.

Determine el ángulo asociado a cada uno de los archivos, justificando el procedimiento empleado, e indicando si es hacia la derecha o hacia la izquierda de la posición correspondiente a $\theta = 0^\circ$.

- Determine cuál es el mínimo ángulo que puede detectarse (resolución) y analice si esa resolución es constante para ángulos entre -90° y 90° .

Falta introduccion

Problema 1. Circuito RLC.

En este problema se presenta un circuito RLC serie con una fuente de tensión. Se aplica el método de Euler y se analiza la respuesta de la salida con distintos pasos de integración numérica para una entrada definida como un pulso rectangular. En ítem posteriores se calcula la respuesta en amplitud y frecuencia de fase para luego comparar el valor de frecuencia máxima de amplitud obtenido con el valor máximo teórico conocido en teoría de circuitos.

a) Para el circuito RLC serie, supondremos características de capacitor e inductor lineales para encontrar una relación mediante la tensión de entrada $U(t)$ y la carga del capacitor $q(t)$.

Las relaciones V-A de cada elemento serán las siguientes:

$$U_R(t) = i(t) \cdot R \quad U_L(t) = L \cdot \frac{di(t)}{dt} \quad i_c(t) = C \cdot \frac{du_c(t)}{dt}$$

Planteamos ley de Kirchhoff de tensión en la malla y reemplazamos las tensiones por las relaciones V-A halladas

$$U(t) = U_R(t) + U_L(t) + U_C(t)$$

$$U(t) = i(t) \cdot R + L \frac{di(t)}{dt} + \frac{q(t)}{C}$$

$$U(t) = \frac{dq(t)}{dt} \cdot R + L \frac{d^2q(t)}{dt^2} + \frac{q(t)}{C} \quad (*)$$

b) Obtuvimos una ecuación diferencial ordinaria (EDO) de 2° orden que describe la dinámica del sistema. Buscamos expresarla como un sistema con dos ecuaciones diferenciales ordinarias de 1° orden con un cambio de variables de la forma:

$$\begin{cases} x_1(t) = q(t) \\ x_2(t) = q'(t) = i(t) \end{cases}$$

A partir de estas dos ecuaciones y despejando la derivada segunda de la carga en la ecuación (*), se expresan las siguientes relaciones diferenciales:

$$x_1'(t) = x_2(t)$$

$$x_2'(t) = \frac{U(t) - \frac{x_1(t)}{C} - R x_2(t)}{L}$$

Utilizando el método de Euler encontramos las aproximaciones:

$$\begin{cases} x_1'(t) = x_2(t) \approx \frac{x_1((n+1)T) - x_1(nT)}{T} \\ x_2'(t) = \frac{U(t) - \frac{x_1(nT)}{C} - R x_2(nT)}{L} \approx \frac{x_2((n+1)T) - x_2(nT)}{T} \end{cases}$$

Despejando las variables de interés x_1 y x_2 :

$$\begin{cases} x_1((n+1)T) \approx x_2(nT) + x_1(nT) \\ x_2((n+1)T) \approx \frac{T}{L} \left[U(nT) - \frac{x_1(nT)}{C} - R x_2(nT) \right] + x_2(nT) \end{cases}$$

Cambiando el índice en la discretización:

$$\begin{cases} x_1(nT) \approx x_2((n-1)T) + x_1((n-1)T) \\ x_2(nT) \approx \frac{T}{L} \left[U((n-1)T) - \frac{x_1((n-1)T)}{C} - R x_2((n-1)T) \right] + x_2((n-1)T) \end{cases}$$

Finalmente se obtiene:

$$\begin{cases} x_1(nT) \approx x_2((n-1)T) + x_1((n-1)T) \\ x_2(nT) \approx \frac{T}{L} \left[U((n-1)T) - \frac{x_1((n-1)T)}{C} - R x_2((n-1)T) \right] + x_2((n-1)T) \end{cases}$$

c) En el script de Matlab consideramos las condiciones iniciales de capacitor e inductor nulas, el pulso rectangular de 10ms y amplitud 1V. Se grafica las respuestas para un tiempo total de integración de 20ms y para un tiempo de 2ms con el fin de apreciar mejor el amortiguamiento.

Considerando el paso de integración numérica T1:

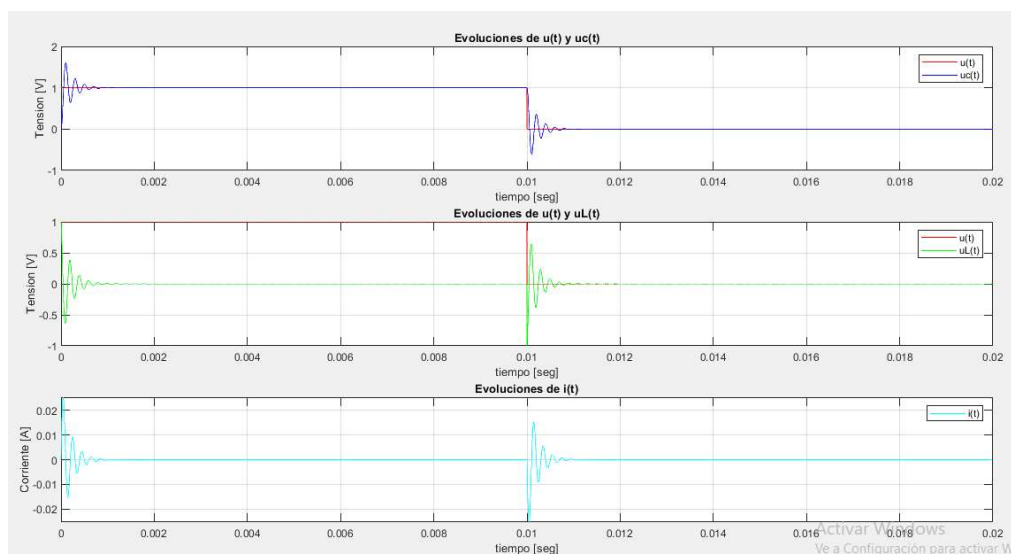


Figura 3. Evoluciones temporales del circuito para un tiempo total de integración de 20ms [T1=1.9869e-06]

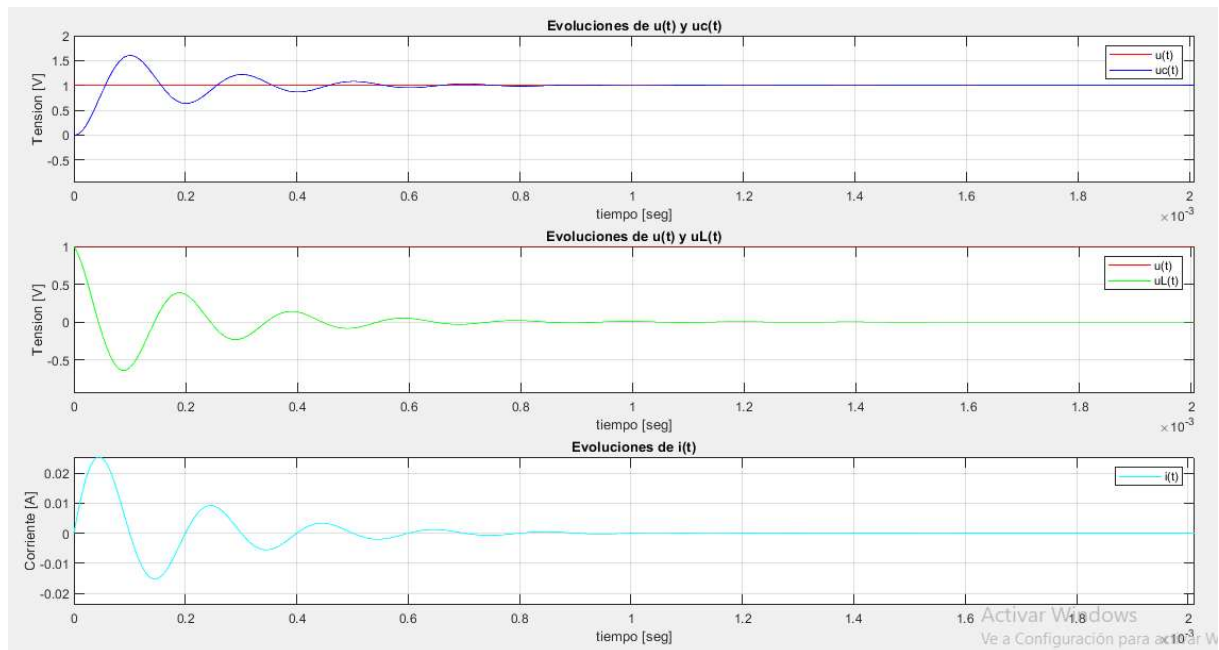


Figura 3. Evoluciones temporales del circuito para un tiempo total de integración de 2ms [$T_1 = 1.9869 \times 10^{-6}$]

Podemos observar cómo, luego de un valor pico, como las respuestas comienza a oscilar estabilizándose mientras transcurre el tiempo. La respuesta que se presenta es de tipo subamortiguada para este paso de integración numérica.

Consideramos el paso de integración numérica T2:

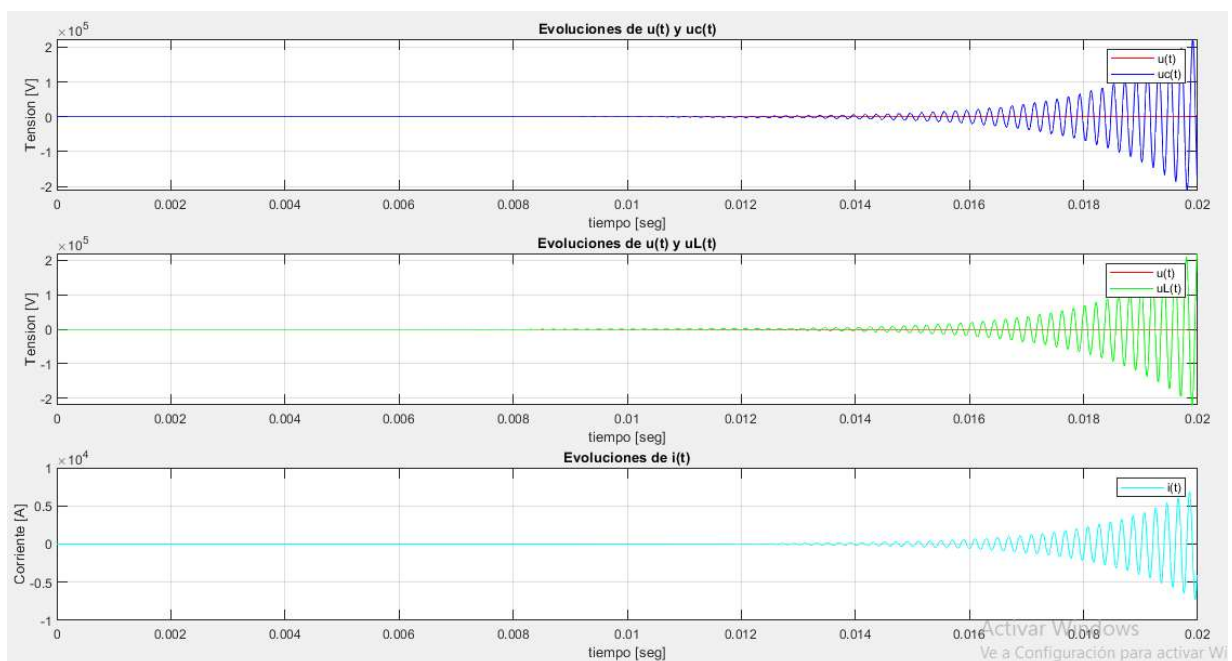


Figura 4. Evoluciones temporales del circuito para un tiempo total de integración de 20ms [$T_2 = 1.3246 \times 10^{-5}$]

Observamos como en este caso el sistema se desestabiliza teniendo valores crecientes en su respuesta. Este tipo de sistema no es realizable físicamente por lo que deducimos que, al tratarse de un paso de integración numérica relativamente grande, el mismo induce error en los cálculos del software (Matlab) para el método de Euler, que son arrastrados dando una respuesta equivocada.

Observación: puede cambiarse el paso de integración variando el parámetro “valor” en el script.

A continuación, se puede ver el script de Matlab realizado para obtener los resultados mostrados.

```
Script: "TP1_Ej1parteI.m"

R = 12; C = 1e-6; L = 1e-3; %Valores de L, R y C dados.
x1(1) = 0; %condicion inicial de carga
x2(1) = 0; % condicion inicial de corriente
t10 = 10e-3; %tiempo pedido
valor = 100; %valor variable para paso de integración
T=(2*pi*sqrt(L*C)/(valor*sqrt(1-R^2*C/4*L))); %paso de integración
Amplitud = 1; %Amplitud del pulso
u = Amplitud * ones(1,round(t10/T)); %Entrada pulso de 10ms
for n = round(t10/T)+1:1:round(2*t10/T)
    u(n) = 0; %definimos 0 el vector desde 10ms a 20ms
end
for n = 1:1:round(2*t10/T)-1 %Método de aproximación de Euler
    x2(n+1) = u(n) * T/L + x2(n)*(1-R*T/L) - x1(n)*T/(C*L);
    x1(n+1) = T*x2(n) + x1(n);
end
tm = [0:1:round(2*t10/T)-1];
t = tm*T; % Escala del tiempo
uc = x1/C; %Tension en el capacitor
uL = u-uc-x2*R; %Tensión en inductor
subplot(3,1,1), plot(t,u,'r',t,uc,'b'), grid on;
%PLOTEOS
xlabel('tiempo [seg]'); ylabel('Tension [V]'); title('Evoluciones de u(t) y uc(t)'); legend('u(t)','uc(t)');
subplot(3,1,2), plot(t,u,'r',t,uL,'g'), grid on;
xlabel('tiempo [seg]'); ylabel('Tension [V]'); title('Evoluciones de u(t) y uL(t)'); legend('u(t)','uL(t)');
subplot(3,1,3), plot(t,x2,'c'), grid on;
xlabel('tiempo [seg]'); ylabel('Corriente [A]'); title('Evoluciones de i(t)'); legend('i(t)');
```

Observación: Podría haberse definido el pulso de 10ms como una diferencia de funciones de Heaviside.

A modo de presentación mostramos las respuestas de $U_c(t)$ y $U_L(t)$ para otros pasos de integración.

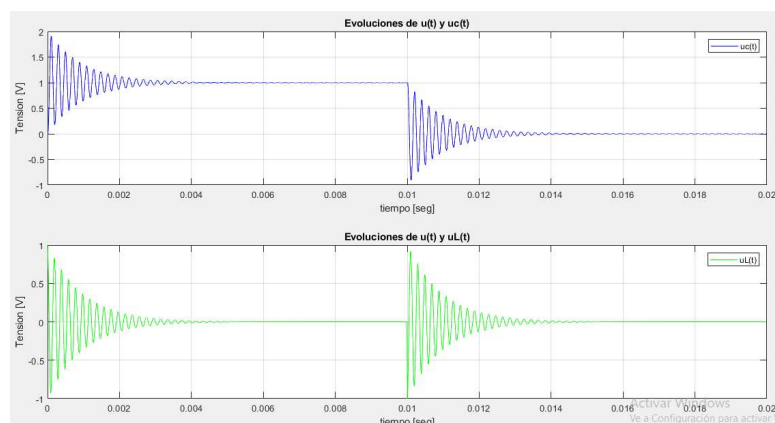


Figura 5a. Respuestas de tensiones para $T = 9.9346e-06$ (“valor” = 20)

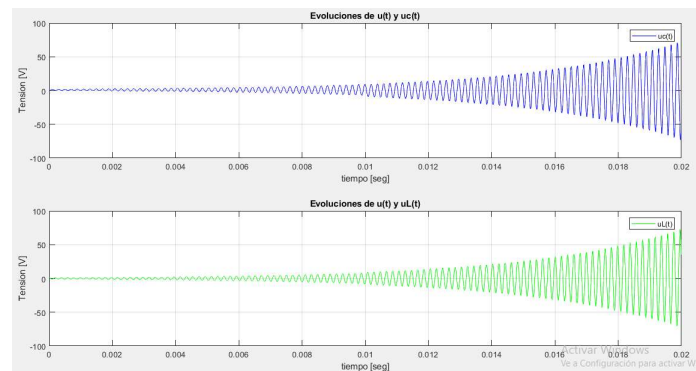


Figura 5b. Respuesta de tensiones para $T = 1.2418e-05$ ("valor" =16)

*Observamos que, para este último, el paso de introducción ya induce un error en la aproximación visible en la gráfica.

d) Una vez acabado el régimen transitorio, veremos cómo es la respuesta del circuito para distintos tipos de frecuencias dados. Trabajaremos con el método fasorial entonces:

$$u(t) = \sin(2\pi Ft) \rightarrow U_{ef} = 1/\sqrt{2}$$

Expresamos las variables del circuito en el sistema fasorial:

$$Z = R + j(\omega L - 1/\omega C)$$

$$I = U_{ef} / Z$$

$$U_L = I \cdot \omega L$$

No es lo
quese pide.
Hay que usar
el script
anterior

Luego la función transferencia del sistema será:

$$H(j\omega) = \frac{U_L}{U_{ef}} \rightarrow \text{Respuesta en frecuencia (RF)} = 20 \log_{10}|H(j\omega)|$$

Utilizamos el siguiente script de Matlab desarrollado para graficar la respuesta de frecuencia de amplitud expresada en decibeles.

Script: "TP1_Ej1parteII.m"

Se usa la simulacion anterior.

```
R = 12;
C = 1e-6;
L = 1e-3;
F = [1000,2000,3000,4000,4500,4700,5000,5300,5500,6000,8000,10000]'; %vector de frecuencias dadas
t = [0:1/(length(F)-1):1]'; %vector de tiempo con tamaño en función del vector de frecuencias.
w = 2*pi*F; %frecuencia angular
U = sin(w.*t); %definimos la entrada U senoidal de frecuencia w
Uef = 1 / sqrt(2);
Z = R + i*(w*L-(1./(w*C))); %Impedancia
I = Uef ./ Z; %Aplicamos ley de ohm
UL = I .* (w*L); %Tensión en el inductor
H = abs(UL./Uef); %Modulo de la función transferencia
RF = 20* log(H); %Respuesta frecuencia de amplitud en decibeles
semilogx(F,RF,'r'), grid on; %Ploteo en escala logaritmica
xlabel('Frecuencia [Rad/s]'); ylabel('Ganancia [dB]'); title('Respuesta en frecuencia de amplitud');
[filas,columna] = find(RF == max(RF),1); %Buscamos cuando el maximo coincide con RF
fteorica = 1./(2*3.14*sqrt(L*C)) %Frecuencia teórica máxima
fexperimental = F(filas) %frecuencia a la cual se da el maximo segun nuestros cálculos
Comparacion = fteorica / fexperimental; %comparacion
```

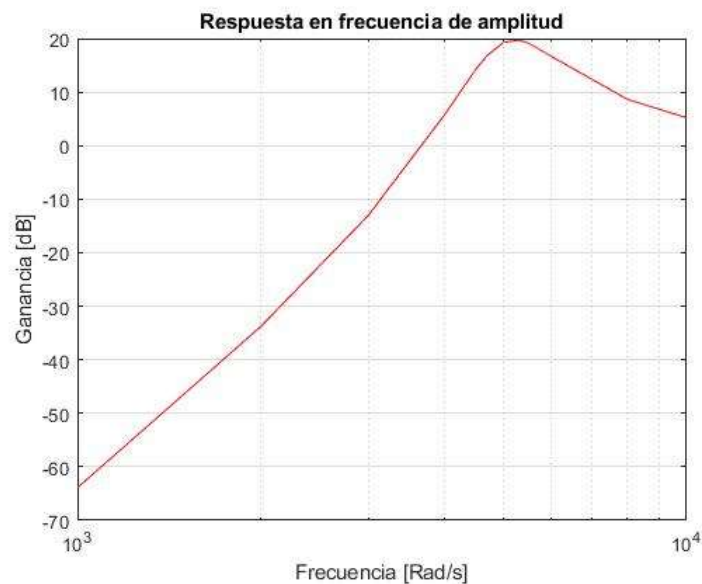


Figura 5a. Respuesta en frecuencia de amplitud

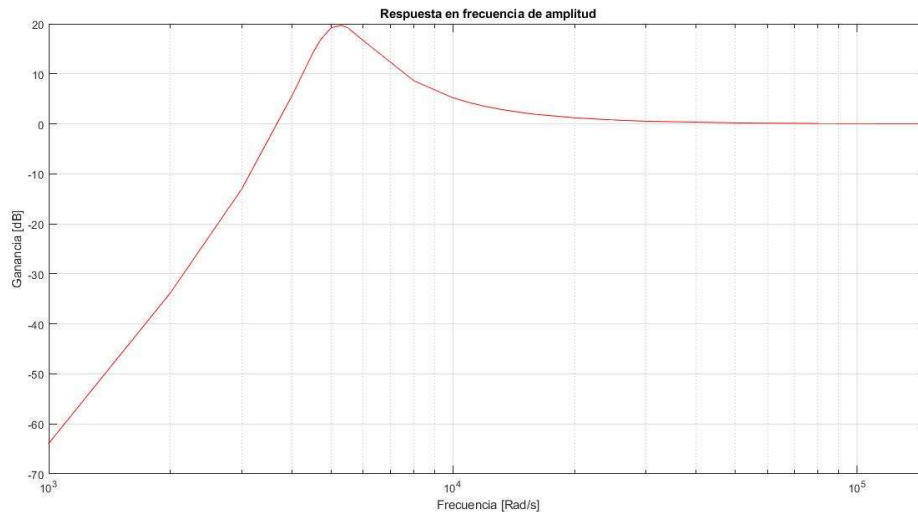


Figura 5b. Respuesta en frecuencia de amplitud con más datos en el vector frecuencias.

Se observa como la ganancia máxima de la salida es 20db. Además, en la figura 5b, se adicionó frecuencias superiores para ver el comportamiento del circuito como filtro.

e) En el script de Matlab que vimos anteriormente, podemos observar cómo se calcula la frecuencia en la cual se produce el máximo de la respuesta. El valor por computo lo llamamos $F_{\text{experimental}}$ mientras que, al teórico, calculado con conocimientos de teoría de circuitos, lo llamamos $F_{\text{teórico}}$. Aquí podemos observar los resultados:

$$F_{\text{Experimental}} = 5300 \text{ Hz}$$

$$F_{\text{Teórica}} = \frac{1}{2\pi\sqrt{LC}} = 5035.5 \text{ Hz}$$

Luego se comparan ambas:

$$\frac{F_{\text{Teórica}}}{F_{\text{Experimental}}} = 0,9501$$

Si tomamos como certero el valor teórico de la frecuencia, calculamos el error absoluto y relativo respecto del experimental.

$$\varepsilon_{\text{abs}} = |F_{\text{Teórica}} - F_{\text{Experimental}}| = 264.5 \text{ Hz}$$

$$\varepsilon_{\text{relativo}} = \frac{264.5 \text{ Hz}}{5035.5 \text{ Hz}} = 0.0525 \rightarrow \text{Error relativo porcentual del 5.2\%}$$

Deducimos que la diferencia entre ambas es debido a que experimentalmente se trabajó con aproximaciones del software y si bien el error podría ser menor, lo tomamos adecuado.

Problema 2. Localización y seguimiento de una fuente sonora

- a. En este problema se presenta un par de micrófonos separados a una distancia L , los cuales pueden pivotear en el punto medio. Se les hace llegar una señal de sonido plana desde diferentes puntos y el sistema tiene que poder calcular su ángulo de incidencia para así poder rotar su eje y hacer un seguimiento de la señal.

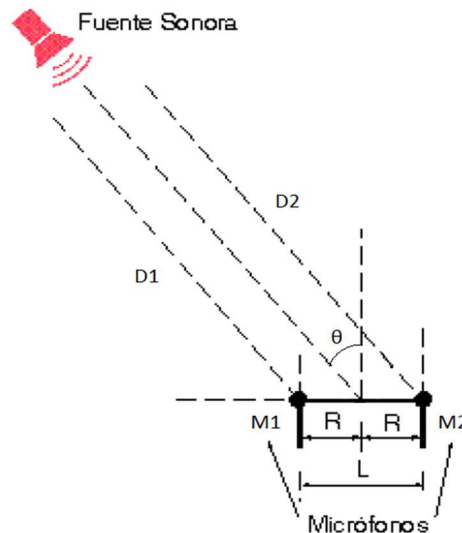


Figura 6. Esquema del modelo.

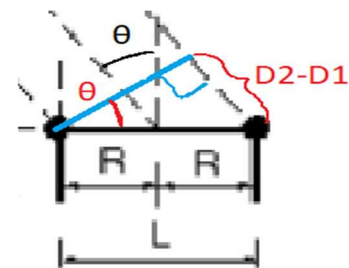


Figura 7. Esquema del modelo ampliado.

Como el frente de onda es plano se puede pensar que las señales $D1$ y $D2$ se mueven en líneas rectas paralelas y que comienzan un plano perpendicular a ambas. Por lo tanto, la diferencia de recorrido de ambas señales (por ende, el tiempo de retardo de la señal 1) se puede visualizar como en la figura 2 y se puede expresar como:

$$\sin(\theta) = \frac{D2 - D1}{L} = \frac{c * (t_2 - t_1)}{L} = \frac{c * (\text{tiempo de retardo})}{L}$$

- b. Ahora la única incógnita que nos resta encontrar es el tiempo de retardo para cada posición de la fuente sonora para así poder calcular su posición. Esto podemos realizarlo haciendo una correlación entre la señal recibida por el micrófono 1 y la recibida por el 2 ya que, al ser la misma señal $x(n)$ pero solo retrasada o adelantada K unidades en el tiempo ($D1=x(n)$ y $D2=x(n \pm K)$), la autocorrelación presentará un máximo en dicho retardo.

$$r_{D1D2}(l) = \sum_{n=-\infty}^{\infty} x(n \pm K + l) * x(n) = r_{xx}(l \pm K)$$

Por lo tanto, si hacemos la correlación entre ambas señales de entrada y buscamos en qué punto es máximo (llamado "muestra" en el script) vamos a poder conseguir el valor de K (llamado delta_t en el script). Hay que tener en cuenta que, como las señales $D1$ y $D2$ son finitas de longitud Y , la correlación nos va a dar un

vector con l en el intervalo $[-Y, Y]$ pero que Matlab lo representa como $[0, 2Y]$, por lo que se es necesario restar Y unidades a nuestra “muestra” para así poder calcular el Δt . Luego se divide la resta por la frecuencia a la cual se hizo el muestreo:

$$\Delta t = \frac{\text{muestra} - Y}{F_s}$$

Con el tiempo de retardo calculado procedemos a calcular el ángulo con la fórmula presentada al principio. Hay que tener presente que si el Δt nos da negativo significa que la señal 1 llega después que la 2, por lo que el ángulo va a ser negativo, entonces la fuente sonora se encuentra a la derecha, mientras que si es positivo se encontraría a la izquierda. Por otro lado, si es 0° , la fuente sonora se encuentra en el centro, a igual distancia de ambos micrófonos.

$$\theta = \arcsen\left(\frac{c * \Delta t}{L}\right)$$



- c. Dicho ángulo theta está relacionado únicamente con la variación de tiempo que midamos, por lo que su resolución a su vez va a depender de la resolución con la cual podamos medirla. Como el Δt se calcula como $\Delta t = \frac{\text{muestra} - Y}{F_s}$ y, como tanto “muestra” e Y son enteros, el valor absoluto del menor valor que podamos calcular sería $1/F_s$. Ésta sería nuestra resolución para el Δt , por lo que la resolución para el ángulo sería:

$$\theta = \arcsen\left(\frac{c}{L * F_s}\right)$$

Podemos observar que dicha resolución se mantiene constante siempre y cuando nuestra frecuencia de muestreo se mantenga constante, por lo cual no va a variar con la variación del ángulo relativo entre la señal y el eje de los micrófonos.

A continuación, se presenta el script de Matlab utilizado.

Script: "TP1_Ej2.m"

```
L = 0.2; %distancia entre microfonos
velocidad_sonido = 342.2; %velocidad del sonido dada
n = [0:88199.9:10];
%Leemos los datos de cada audio y los almacenamos en una matriz
[y1, Fs1] = audioread('datos_1.wav');
[y2, Fs2] = audioread('datos_2.wav');
[y3, Fs3] = audioread('datos_3.wav');
[y4, Fs4] = audioread('datos_4.wav');
[y5, Fs5] = audioread('datos_5.wav');
[y6, Fs6] = audioread('datos_6.wav');
[y7, Fs7] = audioread('datos_7.wav');
[y8, Fs8] = audioread('datos_8.wav');
[y9, Fs9] = audioread('datos_9.wav');
Y = [y1,y2,y3,y4,y5,y6,y7,y8,y9];
Fs = [Fs1, Fs2, Fs3, Fs4, Fs5, Fs6, Fs7, Fs8, Fs9]; %vector frec de muesteo
for i = 1:1:9
    rxy = xcorr(y(:,2*i-1),y(:,2*i)); %correlacion cruzada
    muestra = find(rxy == max(rxy)); %Busca cuando la correlacion cruzada sea máxima.
    delta_t(i) = (muestra-length(y(:,2*i-1))) / Fs(i); %determina el delta_t para cada señal.
    angrad(i) = asin (((velocidad_sonido)*(delta_t(i)))/(L)); %vector de angulos para cada señal.
    ang(i) = (angrad(i) *180) / pi; %expresion de el angulo en grados.
    if (ang(i)<0)
        orientacion = 'izquierda';
    end
    if (ang(i)>0)
        orientacion = 'derecha'; %Condiciones para determinar la orientacion del ángulo.
    end
    if (ang(i)==0)
        orientacion = 'nula';
    end
    fprintf ('paquete %d: angulo en grados sexagesimales: %f, con orientación: %s \n',i,abs(ang(i)),orientacion)
end
%El ángulo mínimo (resolución) corresponde a una muestra y se
% encuentra cuando: [muestra-length(y(:,2*i-1))] = 1 y así
% Ángulo = asin(velocidad_sonido/L*Fs)
angmin = asin(velocidad_sonido/(L*Fs(1))) %angulo minimo en radianes
angminsec = (angmin * 180) / pi
% La resolución es constante debido a que depende de parámetros invariantes
% con el desplazamiento angular.
```

Y los resultados obtenidos para las señales de entrada dadas:

```
paquete 1: angulo en grados sexagesimales: 0.000000, con orientación: nula
paquete 2: angulo en grados sexagesimales: 15.758742, con orientación: derecha
paquete 3: angulo en grados sexagesimales: 30.289966, con orientación: derecha
paquete 4: angulo en grados sexagesimales: 44.296162, con orientación: derecha
paquete 5: angulo en grados sexagesimales: 58.601022, con orientación: derecha
paquete 6: angulo en grados sexagesimales: 13.461334, con orientación: izquierda
paquete 7: angulo en grados sexagesimales: 30.289966, con orientación: izquierda
paquete 8: angulo en grados sexagesimales: 44.296162, con orientación: izquierda
paquete 9: angulo en grados sexagesimales: 58.601022, con orientación: izquierda
```

```
El angulo mínimo en grados sexagesimales es: 2.223530
```

No analiza la resolucion
Faltan conclusiones