 UNIVERSIDAD DE OVIEDO DEPARTAMENTO DE MATEMÁTICAS	Asignatura	Métodos Numéricos	Página 1 de 5
	Tema	Análisis de Fourier Series y transformadas	
	Práctica	4.3	
	Autor	César Menéndez Fernández	

1 Series y transformadas

En esta práctica se van a estudiar de forma práctica la obtención y uso de las series de Fourier y las transformadas.

1.1 Series de Fourier

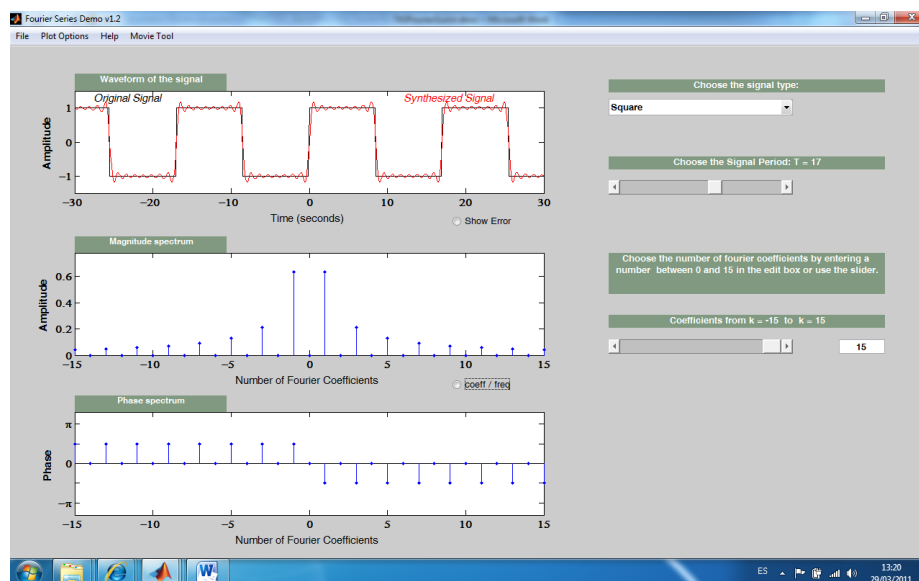
Se obtienen realizando la aproximación por mínimos cuadrados de una función temporal $f(t)$ periódica de periodo T , tomando como base $\{1/2, \sin(\omega t), \cos(\omega t), \dots, \sin(N\omega t), \cos(N\omega t), \dots\}$ donde $\omega = 2\pi/T$. Sus coeficientes vienen dados por

$$\psi(t) = \frac{1}{2}a_0 + \sum_{k=1}^{\infty} (a_k \cos(k\omega t) + b_k \sin(k\omega t)) \rightarrow \begin{cases} a_k = \frac{2}{T} \int_0^T f(t) \cos(k\omega t) dt & a_0 = \frac{2}{T} \int_0^T f(t) dt \\ b_k = \frac{2}{T} \int_0^T f(t) \sin(k\omega t) dt \end{cases}$$

o bien

$$\psi(t) = \frac{1}{2}a_0 + \sum_{k=1}^{\infty} c_k \cos(k\omega t + \theta_k) \leftrightarrow \begin{cases} c_k = \sqrt{a_k^2 + b_k^2} \\ \theta_k = \arctan\left(-b_k/a_k\right) + [\pi \quad a_k < 0] \end{cases}$$

En vez de programar el algoritmo podemos descargar desde [FourierSeriesDemo.zip](http://users.ece.gatech.edu/mcclella/matlabGUIs/) desde <http://users.ece.gatech.edu/mcclella/matlabGUIs/> para comprobar como son los coeficientes de las series de Fourier de algunas señales.



En esta demo se puede cambiar el tipo de señal de entrada, así como el número de términos a mostrar, bien en un diagrama de coeficientes o frecuencias. Es interesante comprobar que los

diagramas no varían cuando se modifica el periodo si trabajamos con los coeficientes, pero al trabajar con frecuencias los valores se acercan cuando se aumenta el periodo.

Descargando el fichero adjunto a la práctica, se tiene la función *apxfourier*, cuya sintaxis es:

$[C, Cs] = \text{apxfourier}(\text{fun}, T, n)$

donde *fun* es la función que define la señal y puede ser tanto simbólica como numérica, *T* es el periodo ($[0, T]$ si $T > 0$ o $[T/2, -T/2]$ si $T < 0$) y *n* es el número de términos que se desea obtener. Como salida se tiene *C* que es un vector con dos filas que almacena los coeficientes ($C(1,1) = a_0$, $C(1,k) = a_{k-1}$, $C(2,k) = b_{k-1}$); en caso de que la función sea simbólica también se puede obtener *Cs*, que muestra los resultados simbólicos para a_0 , a_k y b_k .

Definimos una función impulso simbólica y obtenemos sus coeficientes de Fourier y los representamos

```
%f = evalin(symengine, 'heaviside(x+0.5)-heaviside(x-0.5)');
Syms x real
f = heaviside(x+0.5)-heaviside(x-0.5);
N=20;T=4;
C=apxfourier(f,-T,N)
subplot(1,2,1);stem(sqrt(sum(C.^2)));title('Dominio de la Frecuencia');
xlabel('Termino k-esimo');ylabel('\itAmplitud c_k');axis([0,N,0,1]);
subplot(1,2,2);stem(atan2(C(1,:),C(2,:)));title('Dominio de la Frecuencia');
xlabel('Termino k-esimo');ylabel('\itDesfase \theta');axis([0,N,-3.5,3.5]);
```

La misma función se puede definir de forma numérica, obteniendo el mismo resultado

```
fn = @(x) (x>=-0.5 & x<=0.5);X=linspace(-T/2,T/2);Y=fn(X);plot(X,Y);
C=apxFS(fn,-T,N)
```

También podemos comparar la señal inicial con la aproximación de Fourier mediante

```
Y2=ones(size(X))*norm(C(:,1))/2;w=2*pi/T;
for k=2:N;
    Y2=Y2+C(1,k)*cos((k-1)*w*X)+C(2,k)*sin((k-1)*w*X);
    plot(X,[Y;Y2]);grid on ;title(['Armónico ',int2str(k-1)]);
    legend('Señal','Fourier');
    disp('Pulse una tecla para añadir el siguiente armónico');pause
end
```

Ejercicio recomendado:

1. Repetir el ejemplo anterior aumentando el periodo y tomando más términos. Verificar los resultados tomando funciones numéricas.
2. Idem con la función diente de sierra con la misma duración del pulso y utilizando funciones numéricas¹
3. Idem con la función triangular simétrica con la misma duración del pulso y utilizando funciones numéricas

¹ Con la versión 7.0 no funciona correctamente la definición simbólica, por lo que es necesario utilizar funciones numéricas. En este caso la función de cuadratura adaptativo *quad* y *quadl* dan problemas cuando $f(0)=f(\frac{1}{2}T)=f(T)=0$ o $f(-\frac{1}{2}T)=f(0)=f(\frac{1}{2}T)=0$

1.2 Transformada de Fourier

Cuando la función no es periódica (o haciendo que su periodo tienda a infinito), convertimos los coeficientes de las series de Fourier en funciones continuas. Se pasa del dominio del tiempo al de la frecuencia y viceversa mediante la transformada de Fourier y la transformada inversa:

$$F(i\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \quad f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(i\omega) e^{i\omega t} d\omega$$

En MATLAB, las instrucciones *fourier* e *ifourier* realizan ambos pasos. Una vez que en el apartado anterior hemos representado un tren de ondas cuadradas, vamos a obtener la transformada de Fourier de un pulso cuadrado

```
f = heaviside(x+0.5)-heaviside(x-0.5); ezplot(f,-10,10)
F = fourier(f)
a=100; W=linspace(0,a,201); FW=subs(F,W);
subplot(1,2,1); plot(W,abs(FW)); title('Dominio de la Frecuencia');
xlabel('w'); ylabel('\itAmplitud');
subplot(1,2,2); plot(W,atan2(imag(FW),real(FW))); title('Dominio de la Frecuencia');
xlabel('w'); ylabel('\itDesfase \theta');
```

Si bien el método es válido para cualquier función con un número finito de discontinuidades, la necesidad de definirla como función simbólica complica muchos casos. Así, un tren de ondas cuadradas no es fácil de definir de forma simbólica.

```
f1 = evalin(symengine, 'sign(sin(x))'); ezplot(f1,-10,10)
F1 = fourier(f1)
%
f1=sign(sin(x)) % No lo admite
```

Sin embargo funciona “casi” correctamente con señales simples como un pulso en diente de sierra o un pulso triangular simétrico (genera la transformada pero no permite operar con ella)

```
f1 = (heaviside(x+0.5)-heaviside(x-0.5))*2*x;
F1 = fourier(f1)
f2=(heaviside(x+0.5)-heaviside(x))*(1+2*x)+(heaviside(x)-heaviside(x-0.5))*(1-2*x)
F2 = fourier(f2)
```

1.3 Transformada discreta de Fourier y transformada rápida de Fourier

Cuando se manejan señales reales, es difícil encontrar una función simbólica que admita MATLAB y se ajuste a la señal. Por ello se trabaja muestreando señales continuas para generar valores discretos. A estos se les puede aplicar la transformada rápida de Fourier para extraer sus características.

Ejemplo: Utilizando una muestra de 32 puntos (la FFT necesita que el número de puntos sea potencia de 2) calcular la amplitud y fase de una onda periódica cuadrada de 1” y periodo 4” utilizando la FFT. Dada la simetría de la señal, sólo se representan la mitad de los valores.

```
F=@(x) (x>=-0.5 & x<=0.5); % Definición de la onda
N=32; T=4; t=linspace(-T/2,T/2,N); w0=2*pi/T;
X=F(t);
Y=fft(X);
```

```
subplot(1,2,1);stem(abs(Y(1:N/2)), 'Marker', '.');title('Dominio de la Frecuencia');
xlabel('w');ylabel('\itAmplitud');
subplot(1,2,2);stem(angle(Y(1:N/2)), 'Marker', '.');title('Dominio de la Frecuencia');
xlabel('w');ylabel('\itDesfase \theta');
```

Se observa la similitud con los diagramas vistos en el caso de la serie de Fourier y la transformada de Fourier.

Ejercicio recomendado:

4. Repetir el ejemplo anterior aumentando con periodos de 4", 20" y 40" y tomando 128 términos.

1.4 Aplicaciones

1.4.1 Obtención de frecuencias fundamentales

Consideremos una señal con varias frecuencias fundamentales, que verificamos con la serie de Fourier, sabiendo que el periodo es 10 (y por tanto su frecuencia fundamental $\omega=2\pi/10$, esto es 10Hz) y representando el espectro de amplitudes para los 8 primeros valores (notar que en la gráfica hay un desplazamiento unitario en el eje de abscisas).

```
T=10;w=2*pi/T;N=8;
F=@(t) 3*sin(w*t)+2*cos(4*w*t)-0.5*cos(5*w*t);
C=apxfourier(F,T,N);
subplot(1,2,1);stem(sqrt(sum(C.^2)), 'Marker', '.');title('Dominio de la Frecuencia');
xlabel('Termino k-esimo');ylabel('\itAmplitud c_k');axis([0,N,0,3.5]);
subplot(1,2,2);stem(atan2(C(2,:),C(1,:)), 'Marker', '.');title('Dominio de la Frecuencia');
xlabel('Termino k-esimo');ylabel('\itDesfase \theta');axis([0,N,-3.5,3.5]);
```

Sabiendo que el periodo es 10, vamos a muestrear la señal y calcular su FFT. Comenzamos tomando 16 puntos.

```
N=16;Ts=linspace(0,10,N);
Fs=N/T; % Frecuencia en Hercios
X=F(Ts);
Y=fft(X);
subplot(2,1,1);plot(Ts,X, 'Marker', '.');title('Señal muestreada');
xlabel('t');ylabel('X(t)');
subplot(2,2,3);stem(abs(Y(1:N/2)), 'Marker', '.');title('Dominio de la Frecuencia');
xlabel('w');ylabel('\itAmplitud');
subplot(2,2,4);stem(angle(Y(1:N/2)), 'Marker', '.');title('Dominio de la Frecuencia');
xlabel('w');ylabel('\itDesfase \theta');
```

No se obtiene ningún tipo de información, debido a que la frecuencia del muestreo es demasiado baja.

Ejercicio recomendado:

5. Repetir el ejemplo anterior aumentando la frecuencia de muestreo (tomar 32, 64 y 128 puntos) y analizar los resultados.

Se observa que al aumentar el número de puntos, el espectro de amplitudes cada vez se parece “más” a la serie de Fourier. Si bien los valores de las amplitudes se ven afectados por el número de puntos, los cocientes con respecto a la de mayor valor cada vez se aproximan más a la relación $(1, 2/3, 1/6)$ de la señal inicial.

1.4.2 Filtros

Una vez se tiene la señal en el dominio de la frecuencia, se pueden eliminar componentes de frecuencia determinadas y volver al dominio del tiempo con la transformada inversa *ifft*. Si consideramos el ejemplo anterior con 128 puntos, podemos eliminar las frecuencias más altas y volver al dominio del tiempo. Si queremos eliminar la componente k -ésima, debemos recordar que el vector Y es “simétrico” respecto al punto medio. Para comparar las señales obtenidas al eliminar la 4ª componente, escribimos

```
K=5; Y(k)=0; Y(end-k+1)=0;
XX=ifft(Y)
Plot(Ts, [X;XX])
Legend('original','filtrada')
xlabel('w'); ylabel('\itAmplitud');
subplot(2,2,4); stem(angle(Y(1:N/2)), 'Marker', '.'); title('Dominio de la
Frecuencia');
xlabel('w'); ylabel('\itDesfase \theta');
```