Universidad de Sonora

División de Ciencias Exactas y Naturales

Física computacional
Carlos Lizarraga Celaya

Reconstruyendo la señal

Campos Quiñonez Jorge Andres

26 de Abril del 2017

Introducción

En la actividad anterior, buscamos los componentes que poseen las mareas en base a los datos obtenidos de la NOAA y el CICESE para los sitios de sondeo: "Puerto Morelosz "Bar Harbor". Ahora, en base a estos datos realizaremos una reconstrucción de la señal, o sea, obtendremos la función de onda para cada sitio y la compararemos con los datos originales descargados de estas dos plataformas. En base a esa actividad anterior, obtuvimos las amplitudes de las frecuencias naturales de la marea en cada sitio, ahora utilizaremos esas componentes.

Desarrollo

Para comenzar a reconstruir la señal, partiremos del código empleado en la actividad anterior, pero en esta ocasión, utilizaremos un método distinto para obtener las componentes de cada señal de datos. Para ello, primero numeramos a los datos en base a cada hora con el siguiente código y a la vez, lo agregamos a la tabla:

```
z = np.arange(0.0, 1440.0, 1.0)
df['Hora'] =  pd.Series(z, index =None)
df.head()
```

Para mostrar ejemplo, obtuvimos lo siguiente:

	Date Time	Water Level	Hora
0	2016-01-01 00:00:00	1.126	0.0
1	2016-01-01 01:00:00	0.600	1.0
2	2016-01-01 02:00:00	0.364	2.0
3	2016-01-01 03:00:00	0.443	3.0
4	2016-01-01 04:00:00	0.897	4.0

Ahora, con el siguiente código obtuvimos los valores para cuando la amplitud adquiere un valor considerable para la altura y de ahí, determinar el periodo en el que este ocurre y así obtener la componente de la marea que le corresponde.

```
print(np.where(a[:,]>0.045))
b= a[a[:,]>0.045]
b
```

Lo que obtuvimos fue:

```
(array([   0,   60,  114,  115,  116,  118,  120, 1320, 1322, 1324, 1325,
       1326, 1380]),)

array([ 1.80362569,  0.06320611,  0.14515994,  0.08528659,  0.7979768 ,
        0.04644003,  0.12942897,  0.12942897,  0.04644003,  0.7979768 ,
        0.08528659,  0.14515994,  0.06320611])
```

Utilizando estos valores, obtuvimos los valores del periodo y amplitud que le corresponden:

```
print( 'Primer Armónico notorio')
print('Amplitud=',np.absolute(yf[0,]/N))
print('frecuencia=', xf[int(720 +0),])
print('periodo', 1/xf[int(720+0),])

print()
print( 'Segundo Armónico notorio')
print('Amplitud=',np.absolute(yf[60,]/N))
print('frecuencia=', xf[int(720 +60),])
print('periodo', 1/xf[int(720+60),])

print()
print('Tercer Armónico notorio')
print('Amplitud=',np.absolute(yf[114,]/N))
print('frecuencia=', xf[int(720 +114),])
print('periodo', 1/xf[int(720 +114),])

print()
print('Cuarto armónico notorio')
print('Amplitud=',np.absolute(yf[116,]/N))
print('frecuencia=', xf[int(720 +116),])
print('periodo', 1/xf[int(720 +116),])

print()
print('Quinto armónico notorio')
print('Amplitud=',np.absolute(yf[118,]/N))
print('frecuencia=', xf[int(720 +118),])
```

```
print('periodo', 1/xf[int(720 +118),])

print()
print('Sexto Armónico notorio')
print('Amplitud=',np.absolute(yf[120,]/N))
print('frecuencia=', xf[int(720 +120),])
print('periodo', 1/xf[int(720 +120),])
```

Y esto es lo que obtuvimos:

```
Primer Armónico notorio
Amplitud= 1.80362569444
frecuencia= 0.0
periodo inf

Segundo Armónico notorio
Amplitud= 0.0632061063292
frecuencia= 0.0416666666667
periodo 24.0

Tercer Armónico notorio
Amplitud= 0.145159940457
frecuencia= 0.0791666666667
periodo 12.6315789474

Cuarto armónico notorio
Amplitud= 0.797976796367
frecuencia= 0.0805555555556
periodo 12.4137931034

Quinto armónico notorio
Amplitud= 0.0464400282825
frecuencia= 0.0819444444444
periodo 12.2033898305

Sexto Armónico notorio
Amplitud= 0.129428968618
frecuencia= 0.0833333333333
periodo 12.0
```

Este proceso se realizó para ambos sitios de sondeo, una vez obtenidos los componentes de cada señal en cada marea, los gráficamos en sus respectivas gráficas de señal.

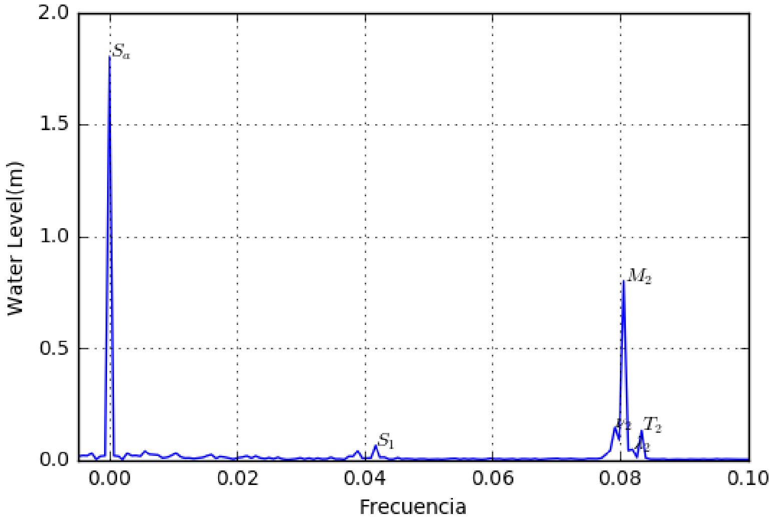


Figura 1: Tabla de frecuenta y altura de la marea para el sitio de sondeo ubicado en Bar Harbor

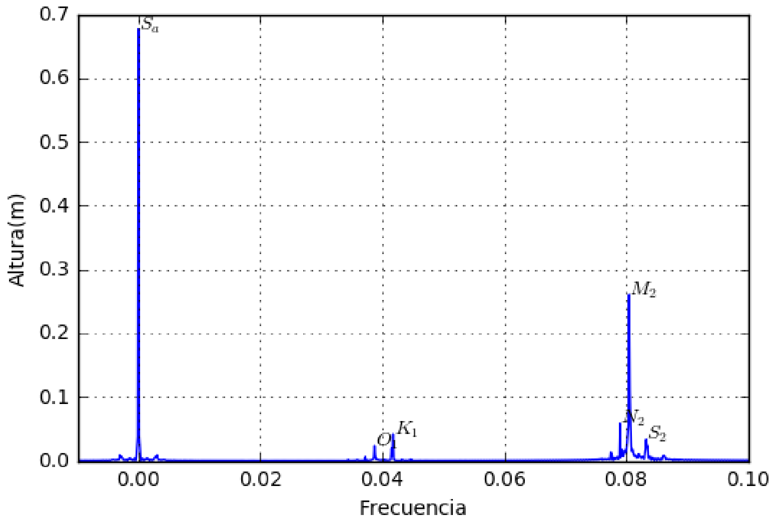


Figura 2: Tabla de frecuenta y altura de la marea para el sitio de sondeo ubicado en Puerto Morelos

Ahora es cuando comienza la parte divertida, vamos a determinar los valores para la amplitud, la frecuencia y el periodo para cada señal y de esta manera poder reconstruirla. Para obtener los componentes de la amplitud, frecuencia y periodo se utilizaron los siguientes códigos:

```
Para Bar Harbor

#Amplitud
Sa = np.absolute(yf[0,]/N)
S1 = np.absolute(yf[60,]/N)
\nu2 = np.absolute(yf[114,]/N)
M2 = np.absolute(yf[116,]/N)
\lambda2 = np.absolute(yf[118,]/N)
T2 = np.absolute(yf[120,]/N)

#Frecuencia
f_Sa = xf[int(720 +0)]
f_S1 = xf[int(720 +60),]
f_\nu2 = xf[int(720 +114),]
f_M2 = xf[int(720 +116),]
f_\lambda2 = xf[int(720 +118)]
f_T2 = xf[int(720 +120),]

#Fase
QSa = np.angle(yf[int(0),])
QS1 = np.angle(yf[int(60),])
Q\nu2 = np.angle(yf[int(114),])
QM2 = np.angle(yf[int(116),])
Q\lambda2 = np.angle(yf[int(118),])
QT2 = np.angle(yf[int(120),])
```

```
Para Puerto Morelos
#Amplitud
Sa = np.absolute(yf[0,]/N)
O1 = np.absolute(yf[340,]/N)
K1 = np.absolute(yf[367,]/N)
N2 = np.absolute(yf[694,]/N)
M2 = np.absolute(yf[707,]/N)
S2 = np.absolute(yf[732,]/N)

#Frecuencia
f_Sa = xf[int(4392 +0)]
f_O1 = xf[int(4392 +340),]
f_K1 = xf[int(4392 +367),]
f_N2 = xf[int(4392 +694),]
f_M2 = xf[int(4392 +707)]
f_S2 = xf[int(4392 +732),]

#Fase
QSa = np.angle(yf[int(0),])
QO1 = np.angle(yf[int(340),])
QK1 = np.angle(yf[int(367),])
QN2 = np.angle(yf[int(694),])
QM2 = np.angle(yf[int(707),])
QS2 = np.angle(yf[int(732),])
```

Como uno de los últimos pasos, utilizamos el siguiente código para reconstruir la función que generará la onda que representará a la señal reconstruida. Utilizando los valores para amplitud, frecuencia y fase que justo acabamos de obtener. Este código es el ejemplo para Puerto Morelos:

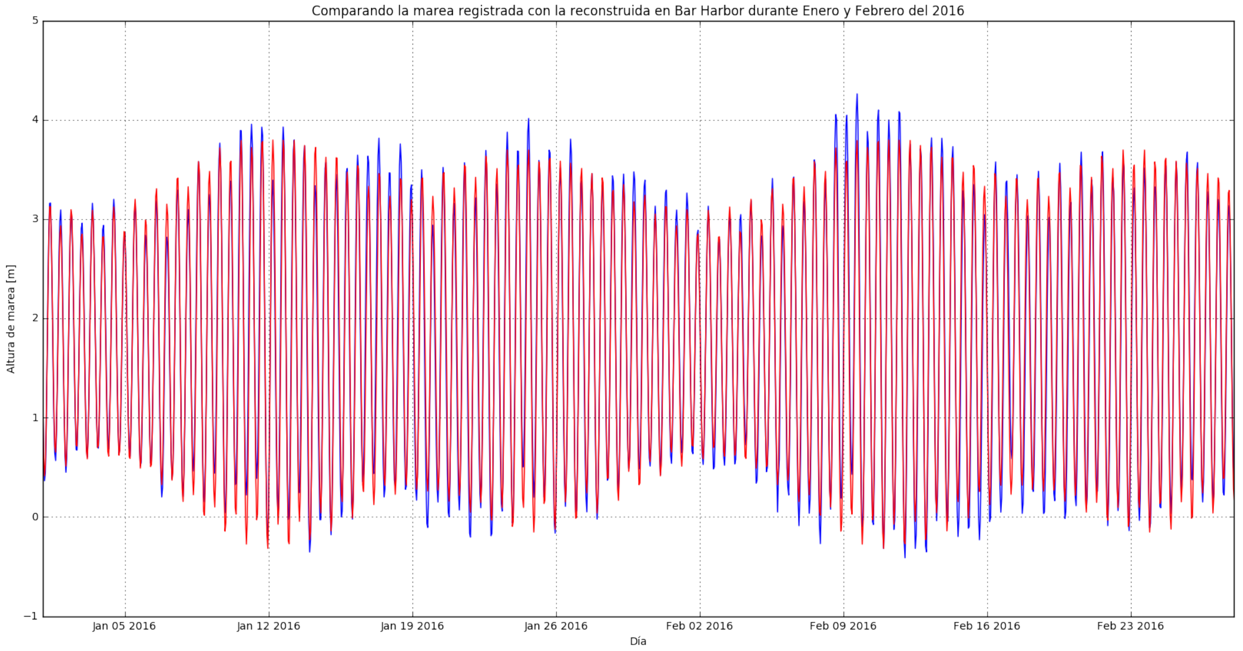
```
w= 2.0*np.pi
a = 0
def f(t):
    return Sa + 2*(O1*np.cos(w*f_O1*t+QO1) + K1*np.cos(w *f_K1
 *t+QK1) + N2*np.cos(w*f_N2*t+QN2) + M2*np.cos(w*f_M2*t+QM2) +
 S2*np.cos(w*f_S2*t+QS2))
```

Ahora, gráficamos la comparación entre la gráfica original y la reconstruida:

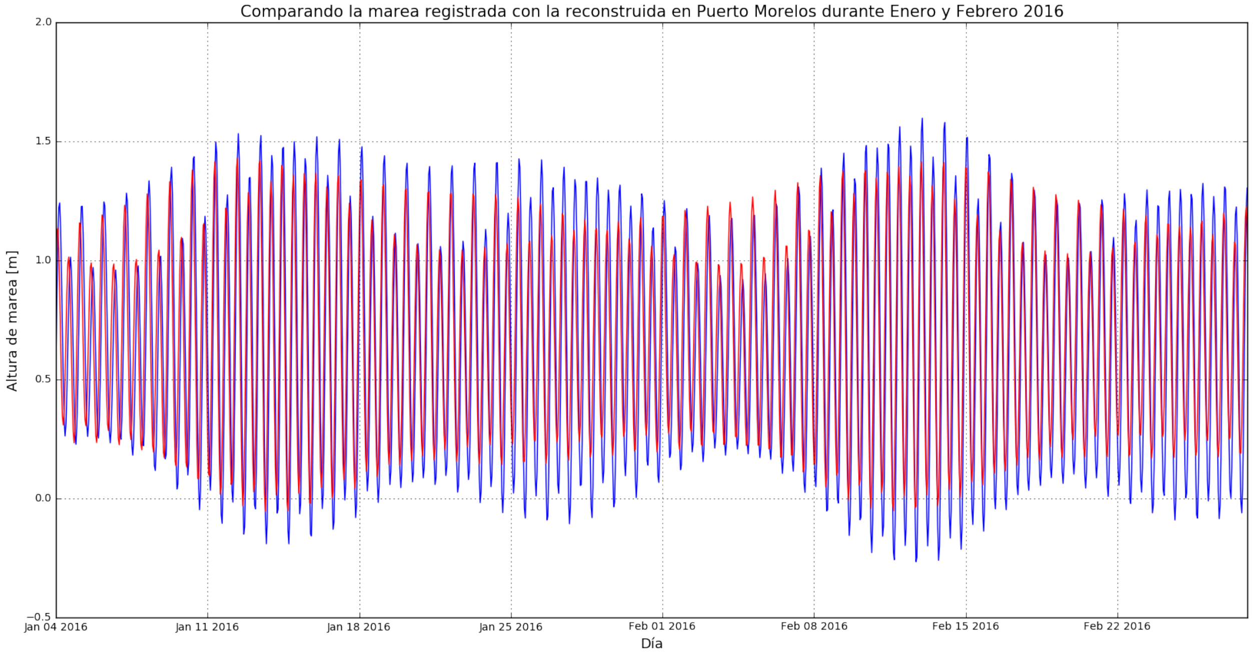
- Para Bar Harbor

```
plt.plot(df['date'], df['Altura(m)'], 'b-', label ="Altura")

plt.plot(df['date'], f(df['Hora']), 'r-', label='Altura recosntruida')

plt.xlim(pd.Timestamp('2016-01-04 00:00:00'),
pd.Timestamp('2016-02-28 00:23:00'))

plt.ylabel('Altura de marea [m]', fontsize = 13)

plt.xlabel('Día', fontsize = 13)

plt.title('Comparando la marea registrada con la reconstruida en
Puerto Morelos durante Enero y Febrero 2016', fontsize= 15)

plt.grid(True)

fig = plt.gcf()
plt.show()
```

La gráfica resultante de este código es:



- Para Puerto Morelos



Como podemos observar, la que tiene un parecido mucho mayor a la original, se dio en el caso del sitio de sondeo en Bar Harbor, aunque para Puerto Morelos también hay una gran aproximación.

Para finalizar, obtendremos el valor de error que existe entre los valores de la gráfica original a la señal reconstruida:

- Para Bar Harbor:

  ```
  Error = sum(abs(y-f(df['Hora']))**2) / sum(abs(y)**2)
  Error = 0.010390260387641036
  ```

  Esto quiere decir que el error es únicamente del 1 %. La aproximación es muy aproximada.

- Para Puerto Morelos:

  ```
  Error = sum(abs(y-f(df['Hora']))**2) / sum(abs(y)**2)
  Error = 0.10848309065942305
  ```

  Esto nos dice que obtuvimos un error del 10 %, que aun así, no se trata de un valor muy grande.