project1_pt2_config1

March 10, 2023

[ ]:
```
import numpy as np
from scipy.integrate import solve_ivp, odeint
from scipy.fftpack import fft
from scipy.signal import find_peaks, hilbert
from tqdm import tqdm
import matplotlib.pyplot as plt
import pywt
# Define the FitzHugh-Nagumo model equations
def fitzhugh_nagumo_coupled(t, xy,alpha , w2, a, b, c, k1, k2):
    x, y , x1, y1= xy
    dxdt = alpha*(y + x - (x**3)/3 + (k1 + c*x1))
    dydt = -(1/alpha) * (w2*x - a + b*y)
    dx1dt = alpha*(y1 + x1 - (x1**3)/3 + (k2 + c*x))
    dy1dt = -(1/alpha) * (w2*x1 - a + b*y1)
    return [dxdt, dydt, dx1dt, dy1dt]

# Define events
# def R_idx(s1,s2):
#     h1 = hilbert(s1)
#     h2 = hilbert(s2)
#     p1 = np.angle(h1)
#     p2 = np.angle(h2)
#     p_diff = p1-p2
#     R = np.abs(np.mean(np.exp(1j*p_diff)))
#     return R

def R_idx(s1, s2):
    #use the Morlet wavele
    wavelet = "cmor1.5-1.0"
    scales = np.arange(1,40) #range of scales
    coi = [2,39] # cone of influence

    #compute the cwt
    cwt_s1, freq_s1 = pywt.cwt(s1, scales, wavelet, sampling_period=40)
    cwt_s2, freq_s2 = pywt.cwt(s2, scales, wavelet, sampling_period=40)

    #find the phases
```

```
    p1 = np.angle(cwt_s1)
    p2 = np.angle(cwt_s2)
    p_diff= p1-p2

    #find the dominating freq of frequency s1
    p_diff = p_diff[np.argmax(np.sum(np.abs(cwt_s1),axis=1)),:]

    R = np.abs(np.mean(np.exp(1j*p_diff)))
    return R

# Define the initial conditions and parameter values
alpha = 3
a = 0.7
b = 0.8
w2 = 1
x0 = 0
y0 = 0
x10 = 0
y10 = 0
xy0=[x0, x10, y0, y10]
c_vals = np.linspace(-0.3,0,num=30)
k1_vals = np.array([-0.3 for i in range(30)])
k2_vals = np.linspace(-0.65,-0.9, num=30)
t_span = [0, 200]
t_eval = np.linspace(0,200,10000)
```

[ ]:
```
Rmap = np.zeros((c_vals.shape[0], k1_vals.shape[0]))
for i, c in tqdm(enumerate(c_vals)):
    for j, k2 in enumerate(k2_vals):
        k1 = k1_vals[len(k1_vals)-1-j]
        sol = solve_ivp(fitzhugh_nagumo_coupled, t_span, xy0,␣
↪args=(alpha,w2,a,b,c,k1,k2),t_eval=t_eval,dense_output=True,rtol=1e-8)

        s1 = sol.y[0]
        s2 = sol.y[2]

        #Apply hamming filter
        window = np.hamming(s1.shape[0])
        s1 = s1 * window
        s2 = s2 * window

        #Calculate R index
        R = R_idx(s1,s2)

        #Save to response map
        Rmap[i, j] = R
```

```
30it [05:10, 10.37s/it]
```

[ ]:
```
Rmap
```

[ ]:
```
array([[0.7937986 , 0.78768135, 0.78488704, 0.77462469, 0.77119832,
        0.75983628, 0.7527187 , 0.7418076 , 0.72790481, 0.7201698 ,
        0.69049887, 0.69125275, 0.66802544, 0.65660092, 0.64236809,
        0.62726654, 0.61045889, 0.59429699, 0.57414259, 0.55762036,
        0.53559861, 0.51534726, 0.49335367, 0.46713563, 0.43865145,
        0.42095421, 0.38429745, 0.36344344, 0.33761425, 0.30366585],
       [0.8141213 , 0.80841733, 0.81540958, 0.8031776 , 0.79035153,
        0.78313217, 0.77721609, 0.76701714, 0.76162701, 0.74817342,
        0.74276433, 0.72564465, 0.72019854, 0.69610338, 0.68269914,
        0.6667346 , 0.65230874, 0.639609  , 0.62118815, 0.60623847,
        0.58716069, 0.56786351, 0.54934038, 0.52660613, 0.50684569,
        0.48329904, 0.45631968, 0.42775491, 0.40795157, 0.37115947],
       [0.80524344, 0.80789848, 0.8162226 , 0.81433232, 0.81622805,
        0.80888378, 0.8150465 , 0.7982914 , 0.78313603, 0.77843735,
        0.76699999, 0.76037419, 0.74840332, 0.73980503, 0.72664707,
        0.71614126, 0.70053703, 0.67136786, 0.66683706, 0.64725212,
        0.63345757, 0.61434193, 0.5994776 , 0.57870342, 0.55949632,
        0.53935094, 0.51611101, 0.49599814, 0.47191844, 0.44533769],
       [0.76030786, 0.77784354, 0.78695183, 0.79888677, 0.80763521,
        0.80982883, 0.81708148, 0.81168018, 0.81377098, 0.80745172,
        0.81017308, 0.78762726, 0.77743437, 0.76997163, 0.75725539,
        0.74892552, 0.73446682, 0.72626473, 0.70792999, 0.69960973,
        0.66505194, 0.66190599, 0.64118831, 0.62660429, 0.60763364,
        0.59165089, 0.57042991, 0.55114909, 0.53121304, 0.50826746],
       [0.69836236, 0.71157412, 0.73013345, 0.74277409, 0.75819479,
        0.77574592, 0.7847227 , 0.79797021, 0.80271034, 0.80871925,
        0.81134995, 0.80379356, 0.80226317, 0.80662009, 0.78924892,
        0.77320762, 0.76816532, 0.75484988, 0.74871919, 0.7331045 ,
        0.72599242, 0.70600591, 0.69796372, 0.66211897, 0.65892907,
        0.63922387, 0.62549817, 0.60791493, 0.59330676, 0.57420452],
       [0.62656225, 0.64735563, 0.66936427, 0.68776902, 0.69710618,
        0.70947683, 0.7232925 , 0.73252488, 0.74806948, 0.75866225,
        0.76974937, 0.78290825, 0.78630936, 0.79559866, 0.790597  ,
        0.79363084, 0.79147665, 0.78951652, 0.76801574, 0.76427127,
        0.75290539, 0.74875919, 0.73515647, 0.73076231, 0.71309485,
        0.70828443, 0.67534346, 0.67363961, 0.65689557, 0.64324269],
       [0.5248312 , 0.55356353, 0.57055568, 0.58596527, 0.60667506,
        0.63074578, 0.65536933, 0.67362946, 0.69307216, 0.71075936,
        0.72132037, 0.73442539, 0.74466338, 0.75098826, 0.7641759 ,
        0.76792358, 0.77848439, 0.77934101, 0.78470539, 0.78541751,
        0.79124276, 0.77135684, 0.76947211, 0.76143626, 0.76103796,
        0.75009823, 0.74727174, 0.73076488, 0.72479452, 0.69217831],
       [0.43153996, 0.45093107, 0.48142374, 0.49677836, 0.53020539,
```

```
 0.55076206, 0.56658925, 0.5880252 , 0.61178083, 0.63402653,
 0.64983738, 0.67069291, 0.68900395, 0.70293844, 0.72413987,
 0.73885961, 0.75130554, 0.76586287, 0.76947721, 0.77941257,
 0.77887092, 0.78506321, 0.78755625, 0.79431669, 0.77615003,
 0.77518532, 0.76670045, 0.76707349, 0.75678842, 0.75306239],
[0.32266747, 0.35566916, 0.38042253, 0.4078459 , 0.43522661,
 0.45941643, 0.48831803, 0.50986496, 0.53887604, 0.5538225 ,
 0.57890018, 0.60250241, 0.62785861, 0.64499085, 0.66693709,
 0.68301286, 0.69886447, 0.71689864, 0.727522  , 0.74616669,
 0.75601358, 0.76871795, 0.78138607, 0.78302139, 0.7968389 ,
 0.80296147, 0.79861912, 0.78703339, 0.78061528, 0.76936196],
[0.2147519 , 0.24459998, 0.27872874, 0.30542441, 0.33548004,
 0.36692054, 0.39286339, 0.42286403, 0.44684819, 0.47959439,
 0.49622823, 0.53127249, 0.55192347, 0.57671538, 0.60046368,
 0.62943453, 0.65150877, 0.67485132, 0.69485013, 0.70962521,
 0.72767634, 0.73657069, 0.75259685, 0.75683686, 0.77043758,
 0.76818677, 0.77874225, 0.78082717, 0.78149455, 0.7730524 ],
[0.12305865, 0.13801203, 0.17202528, 0.2050553 , 0.2340897 ,
 0.26855594, 0.30058608, 0.32858997, 0.36394663, 0.39140972,
 0.41962039, 0.45125392, 0.48015627, 0.51108906, 0.53729574,
 0.56411304, 0.5835347 , 0.60969609, 0.63387004, 0.65588579,
 0.67239698, 0.69320398, 0.70553237, 0.72418665, 0.73330564,
 0.74987653, 0.75466971, 0.76928359, 0.76662463, 0.77969873],
[0.05328387, 0.0515811 , 0.07589843, 0.1107653 , 0.13771963,
 0.16091118, 0.20033129, 0.23252965, 0.26506489, 0.30649594,
 0.33921065, 0.37193838, 0.40767161, 0.43517528, 0.46603917,
 0.49058873, 0.52397285, 0.53576637, 0.56610496, 0.581108  ,
 0.61129206, 0.63102562, 0.65431628, 0.66929678, 0.68989933,
 0.70252857, 0.72076137, 0.73068077, 0.74676709, 0.75251924],
[0.14285127, 0.10131464, 0.07835918, 0.05603994, 0.04109949,
 0.06367215, 0.10584017, 0.14649947, 0.17263526, 0.20719816,
 0.2442431 , 0.27576473, 0.31277937, 0.34618155, 0.375061  ,
 0.4118035 , 0.4388004 , 0.46685182, 0.49338947, 0.52472632,
 0.53775323, 0.56710247, 0.5846999 , 0.61431354, 0.63348912,
 0.6561448 , 0.6709001 , 0.69018697, 0.70383393, 0.72088931],
[0.26384431, 0.22197746, 0.1905783 , 0.15399517, 0.10964203,
 0.07100294, 0.05093635, 0.03100732, 0.06438038, 0.10200708,
 0.14366878, 0.17808781, 0.20360973, 0.24438557, 0.27694608,
 0.30987586, 0.34851875, 0.37778398, 0.41286723, 0.44230336,
 0.4694552 , 0.49792508, 0.52662894, 0.54475098, 0.56838599,
 0.58803962, 0.61776914, 0.63937672, 0.6614558 , 0.67854793],
[0.37273374, 0.33471697, 0.30016673, 0.2634847 , 0.22280727,
 0.18856663, 0.16020452, 0.12419349, 0.08162956, 0.05131279,
 0.02477706, 0.05279176, 0.09541799, 0.13591627, 0.17358184,
 0.20234553, 0.24283018, 0.2777217 , 0.31066427, 0.35185498,
 0.38275272, 0.41450557, 0.44920935, 0.47637818, 0.50608536,
 0.53303301, 0.55617001, 0.57358671, 0.59450615, 0.62541735],
```

```
[0.45464658, 0.42399822, 0.3922324 , 0.35789996, 0.32620323,
 0.29971171, 0.26840433, 0.23229852, 0.19836723, 0.17141054,
 0.13644364, 0.09017426, 0.04688501, 0.02151227, 0.03930733,
 0.09166899, 0.13109414, 0.17053982, 0.20560505, 0.2440205 ,
 0.28136819, 0.31493983, 0.35660507, 0.38931845, 0.42003777,
 0.45789279, 0.48672627, 0.51840759, 0.54570148, 0.5750699 ],
[0.54351262, 0.5125404 , 0.48230351, 0.45595831, 0.42649231,
 0.39496618, 0.36038867, 0.32934798, 0.30264861, 0.27117014,
 0.23474869, 0.19971874, 0.17100162, 0.13390746, 0.08478545,
 0.04312515, 0.02419581, 0.04219391, 0.09519474, 0.13549882,
 0.17513579, 0.21221796, 0.25064511, 0.2901108 , 0.32452612,
 0.36539359, 0.40040796, 0.43162482, 0.47040802, 0.5007871 ],
[0.60452541, 0.58295449, 0.56147791, 0.53861154, 0.51595716,
 0.49499184, 0.47087188, 0.44248329, 0.41088643, 0.37681019,
 0.34509276, 0.31478366, 0.27892954, 0.23750451, 0.19691327,
 0.15939264, 0.11485187, 0.06463906, 0.02831501, 0.03126088,
 0.06043171, 0.11381987, 0.156317  , 0.19506294, 0.22887303,
 0.26851226, 0.30867947, 0.34381252, 0.38253095, 0.4203469 ],
[0.62740556, 0.61495747, 0.60488358, 0.59405498, 0.58090515,
 0.56291603, 0.54287763, 0.5193229 , 0.4932809 , 0.46585208,
 0.43752061, 0.40841311, 0.38039477, 0.35341286, 0.321929  ,
 0.28541959, 0.24243907, 0.19773386, 0.15117331, 0.10274537,
 0.05130797, 0.02170035, 0.0531938 , 0.08621858, 0.14018029,
 0.1878291 , 0.22672013, 0.25918489, 0.29903369, 0.33983576],
[0.66595459, 0.66279691, 0.67603636, 0.57708545, 0.54267395,
 0.52815876, 0.55957469, 0.53397289, 0.52248332, 0.51152646,
 0.4991552 , 0.48380786, 0.46402474, 0.4402873 , 0.41142432,
 0.37957364, 0.34348105, 0.30451815, 0.26247056, 0.22071057,
 0.17914715, 0.14038156, 0.10323084, 0.07606664, 0.07166464,
 0.09189868, 0.11803404, 0.16589965, 0.21539171, 0.25845831],
[0.57050608, 0.57648883, 0.5826223 , 0.58568391, 0.58419455,
 0.57757972, 0.57331577, 0.5741758 , 0.57121009, 0.56212657,
 0.55341567, 0.54913463, 0.58393598, 0.48617429, 0.44938044,
 0.43119848, 0.4155302 , 0.37973569, 0.34648257, 0.32284305,
 0.28086735, 0.24468325, 0.22087947, 0.1975709 , 0.17294928,
 0.15280423, 0.15380579, 0.16832291, 0.18956035, 0.21113799],
[0.41760276, 0.43464545, 0.45062663, 0.46439449, 0.47729363,
 0.48797894, 0.49181244, 0.48992315, 0.48922122, 0.49113369,
 0.49444223, 0.49408294, 0.48793436, 0.48051776, 0.47530585,
 0.47198854, 0.46436335, 0.45965682, 0.4474287 , 0.42264722,
 0.39517755, 0.36431436, 0.32708388, 0.30261054, 0.24869592,
 0.20162847, 0.15821263, 0.15123278, 0.23633833, 0.20666961],
[0.183748  , 0.1859601 , 0.20120518, 0.22678985, 0.25303278,
 0.27927744, 0.30202419, 0.32172681, 0.33707504, 0.35044565,
 0.36708071, 0.38477458, 0.40312534, 0.41671836, 0.42285709,
 0.42790605, 0.43218379, 0.43444916, 0.43426999, 0.42955614,
 0.42476374, 0.42320258, 0.40478694, 0.37719338, 0.3510631 ,
```

```
        0.32588026, 0.30241518, 0.29229802, 0.28187413, 0.26622897],
       [0.21637607, 0.15624271, 0.09585912, 0.03711058, 0.02332166,
        0.07647419, 0.12641384, 0.17399411, 0.2200136 , 0.26465078,
        0.30791547, 0.34946639, 0.38889903, 0.42508488, 0.45716003,
        0.4862365 , 0.5064609 , 0.52525606, 0.54291769, 0.55621605,
        0.56453359, 0.56890786, 0.56962504, 0.56761363, 0.56379035,
        0.55850041, 0.54094528, 0.51716909, 0.50036365, 0.48781591],
       [0.23077217, 0.19217664, 0.17209892, 0.16578292, 0.16726984,
        0.17604781, 0.18962556, 0.2061982 , 0.22434353, 0.24353263,
        0.26438646, 0.2890752 , 0.31641222, 0.34483545, 0.37305957,
        0.40002011, 0.42485308, 0.44696412, 0.46601243, 0.48128928,
        0.4925865 , 0.50090067, 0.5071589 , 0.51230873, 0.51475261,
        0.51484447, 0.51213772, 0.50571069, 0.49481814, 0.47400332],
       [0.18854151, 0.19267649, 0.19156418, 0.19924992, 0.14805414,
        0.15691725, 0.15083628, 0.18497594, 0.17642112, 0.23322968,
        0.10143282, 0.09113561, 0.11154449, 0.17561225, 0.21678102,
        0.22558254, 0.19817991, 0.16439909, 0.14107037, 0.12778857,
        0.12142826, 0.12192213, 0.13220112, 0.15268539, 0.1772914 ,
        0.19290353, 0.18349932, 0.15319771, 0.16971547, 0.12894679],
       [0.33904   , 0.36650402, 0.3784419 , 0.40105257, 0.39975529,
        0.37196442, 0.33648697, 0.29677311, 0.24329887, 0.18710716,
        0.12300154, 0.06108446, 0.03095222, 0.05754579, 0.1044243 ,
        0.14704677, 0.18318865, 0.2059433 , 0.22073762, 0.23177591,
        0.24185579, 0.26127695, 0.28705186, 0.31472458, 0.34199434,
        0.36713577, 0.38971053, 0.40886476, 0.42327831, 0.43230359],
       [0.37628336, 0.36110693, 0.28260287, 0.16529711, 0.16154087,
        0.2540129 , 0.23468227, 0.28410007, 0.31471385, 0.26051743,
        0.33491301, 0.41261642, 0.42117401, 0.33588932, 0.42816594,
        0.33918085, 0.33795835, 0.34743714, 0.33733931, 0.29934183,
        0.28610867, 0.28636226, 0.29348844, 0.31042948, 0.3392655 ,
        0.37974293, 0.43045922, 0.48312709, 0.50325513, 0.27792934],
       [0.68263268, 0.69149647, 0.69878054, 0.69085196, 0.70851008,
        0.70869094, 0.70177164, 0.70863657, 0.71398191, 0.71100725,
        0.71197381, 0.71859882, 0.72220577, 0.71725655, 0.70895735,
        0.70021457, 0.6927762 , 0.68805771, 0.68460364, 0.68191522,
        0.67998656, 0.67858743, 0.6772742 , 0.67639457, 0.67624781,
        0.67717957, 0.67833282, 0.67988069, 0.68167739, 0.684828  ],
       [0.02215449, 0.04167173, 0.03613721, 0.02892835, 0.03752918,
        0.02142867, 0.02414086, 0.02830668, 0.01307367, 0.01244901,
        0.01850426, 0.01014556, 0.00605833, 0.01171805, 0.01219998,
        0.00741111, 0.00922619, 0.01717897, 0.02402316, 0.02842939,
        0.0306225 , 0.0313904 , 0.03154776, 0.03168897, 0.03210196,
        0.03277865, 0.03359494, 0.03444014, 0.0352269 , 0.03578658]])
```

[ ]:
```python
K = []
C = []
for i, c in tqdm(enumerate(c_vals)):
```

```
    C.append(c)
for j, k2 in enumerate(k2_vals):
    K.append((k2+0.3)/(-0.3))

# plt.imshow(Rmap, cmap='viridis', origin='lower', aspect='auto',)
fig, ax = plt.subplots()
im = ax.imshow(Rmap, cmap='viridis', origin='lower', aspect='auto',)

# Set tick labels
ax.set_xticks(np.arange(len(C)))
ax.set_yticks(np.arange(len(K)))
ax.set_xticklabels([f"{c:.3f}" for c in C])
ax.set_yticklabels([f"{k:.3f}" for k in K])

plt.setp(ax.get_xticklabels(), rotation=45, ha="right", rotation_mode="anchor")
cbar = ax.figure.colorbar(im, ax=ax)
cbar.ax.set_ylabel('Data', rotation=-90, va="bottom")

# Add axis labels
ax.set_xlabel('C')
ax.set_ylabel('(k1-k2)/k1')
ax.set_title("Response map for R")
# Show plot
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/Users/timgu/Documents/hacks/NeuralEng/project1_pt2_config1 copy.ipynb Cell 4 i
 ↪<cell line: 3>()
      <a href='vscode-notebook-cell:/Users/timgu/Documents/hacks/NeuralEng/
 ↪project1_pt2_config1%20copy.ipynb#W3sZmlsZQ%3D%3D?line=0'>1</a> K = []
      <a href='vscode-notebook-cell:/Users/timgu/Documents/hacks/NeuralEng/
 ↪project1_pt2_config1%20copy.ipynb#W3sZmlsZQ%3D%3D?line=1'>2</a> C = []
----> <a href='vscode-notebook-cell:/Users/timgu/Documents/hacks/NeuralEng/
 ↪project1_pt2_config1%20copy.ipynb#W3sZmlsZQ%3D%3D?line=2'>3</a> for i, c in
 ↪tqdm(enumerate(c_vals)):
      <a href='vscode-notebook-cell:/Users/timgu/Documents/hacks/NeuralEng/
 ↪project1_pt2_config1%20copy.ipynb#W3sZmlsZQ%3D%3D?line=3'>4</a>     C.append(c)
      <a href='vscode-notebook-cell:/Users/timgu/Documents/hacks/NeuralEng/
 ↪project1_pt2_config1%20copy.ipynb#W3sZmlsZQ%3D%3D?line=4'>5</a> # for j, k2 in
 ↪enumerate(k2_vals):
      <a href='vscode-notebook-cell:/Users/timgu/Documents/hacks/NeuralEng/
 ↪project1_pt2_config1%20copy.ipynb#W3sZmlsZQ%3D%3D?line=5'>6</a> #     K.
 ↪append((k2+0.3)/(-0.3))

NameError: name 'tqdm' is not defined
```

[ ]:
```
zs = [-0.3,-0.4,-0.5,-0.6,-0.7]
k1 = -0.3
k2 = -0.7346
c = -0.0207
# k1 = -0.3
# k2 = -0.68
# c = -0.3
sol = solve_ivp(fitzhugh_nagumo_coupled, t_span, xy0,␣
↪args=(alpha,w2,a,b,c,k1,k2), t_eval=t_eval, dense_output=True,rtol=1e-8)
t = sol.t
s1 = sol.y[0]
s2 = sol.y[2]
```

[ ]:
```
t
```

[ ]:
```
array([0.00000000e+00, 2.00020002e-02, 4.00040004e-02, …,
       1.99959996e+02, 1.99979998e+02, 2.00000000e+02])
```

[ ]:
```
s1
```

[ ]:
```
array([ 0.        , -0.01838055, -0.03754319, …,  1.62281728,
        1.61801692,  1.61323391])
```

[ ]:
```
s2
```

[ ]:
```
array([ 0.        , -0.04527168, -0.09300575, …,  1.54818632,
        1.60376745,  1.65358031])
```

[ ]:
```
wavelet = "cmor1.5-1.0"
scales = np.arange(1,40) #range of scales
coi = [2,39] # cone of influence

#compute the cwt
cwt_s1, freq_s1 = pywt.cwt(s1, scales, wavelet, sampling_period=40)
cwt_s2, freq_s2 = pywt.cwt(s2, scales, wavelet, sampling_period=40)

#find the phases
p1 = np.angle(cwt_s1)
p2 = np.angle(cwt_s2)
p_diff= p1-p2

p_diff
```

[ ]:
```
array([[-2.46762581e-04, -8.01986990e-04, -1.44519122e-03, …,
         3.30547388e+00,  4.68497749e+00, -3.60336614e-02],
       [ 3.99511897e-04,  4.03375635e-04,  3.15134590e-04, …,
         5.00520450e-04,  6.67318264e-04,  8.22769458e-04],
```

```
        [ 3.29245988e-03,  2.17891606e-03,  8.74562926e-03, …,
          6.80934858e-03,  4.92059509e-03,  1.21903789e-02],
        …,
        [-5.87844705e-01, -6.20063380e-01, -6.59093675e-01, …,
          3.54359878e-01,  3.58103897e-01,  3.45344020e-01],
        [-6.92122179e-01, -7.27339105e-01, -7.68972054e-01, …,
          3.64869737e-01,  3.69450799e-01,  3.57080378e-01],
        [-7.80019916e-01, -8.07989609e-01, -8.41526411e-01, …,
          3.79324029e-01,  3.84988894e-01,  3.73009469e-01]])
```

[ ]:
```
np.sum(np.abs(cwt_s1),axis=1)
```

[ ]:
```
array([1.48511040e+01, 2.85933144e-01, 2.02918320e+00, 9.33890420e-01,
       4.00259331e+00, 1.80476843e+00, 6.44618550e+00, 2.88791343e+00,
       9.49795803e+00, 4.47302856e+00, 5.70901489e+00, 6.48278672e+00,
       1.64785827e+01, 1.01609358e+01, 1.34406687e+01, 1.55151644e+01,
       2.05360404e+01, 2.44373871e+01, 3.00614094e+01, 3.69350128e+01,
       5.46824538e+01, 5.43624664e+01, 6.51581724e+01, 7.69247768e+01,
       9.09547755e+01, 1.06121612e+02, 1.22987498e+02, 1.41833251e+02,
       1.62771280e+02, 1.85071502e+02, 2.08616591e+02, 2.33316506e+02,
       2.60826217e+02, 2.89712325e+02, 3.20581346e+02, 3.54236691e+02,
       3.91335609e+02, 4.30363317e+02, 4.71900631e+02])
```

[ ]:
```
np.sum(np.abs(cwt_s2),axis=1)
```

[ ]:
```
array([2.49623847e+01, 3.03887984e-01, 3.05110330e+00, 9.94405567e-01,
       5.88757725e+00, 1.93907680e+00, 9.40846455e+00, 3.14879037e+00,
       1.38570928e+01, 5.12895626e+00, 6.81840372e+00, 7.86104091e+00,
       2.39678528e+01, 1.35343150e+01, 1.86588537e+01, 2.25160932e+01,
       3.07584294e+01, 3.79346701e+01, 4.79956951e+01, 6.03489931e+01,
       8.88866467e+01, 9.24667623e+01, 1.12816241e+02, 1.35300259e+02,
       1.62368161e+02, 1.91765151e+02, 2.24389554e+02, 2.60631958e+02,
       3.00864859e+02, 3.44349607e+02, 3.90846737e+02, 4.40161644e+02,
       4.95778641e+02, 5.53939309e+02, 6.15734931e+02, 6.80762134e+02,
       7.50574647e+02, 8.22507507e+02, 8.98732900e+02])
```

[ ]:
```
#find the dominating freq of frequency s1
p_diff = p_diff[np.argmax(np.sum(np.abs(cwt_s1),axis=1)),:]

R = np.abs(np.mean(np.exp(1j*p_diff)))
R
```

[ ]:
```
0.3398132988561196
```

[ ]:
```
# plt.plot(t[0:100], (np.abs(np.fft.fft(v))**2)[0:100])
plt.plot(t, s1)
plt.plot(t, s2)
```

```
plt.xlabel('t')
plt.ylabel('v')
plt.show()
```