mixed_BPMs

August 14, 2015

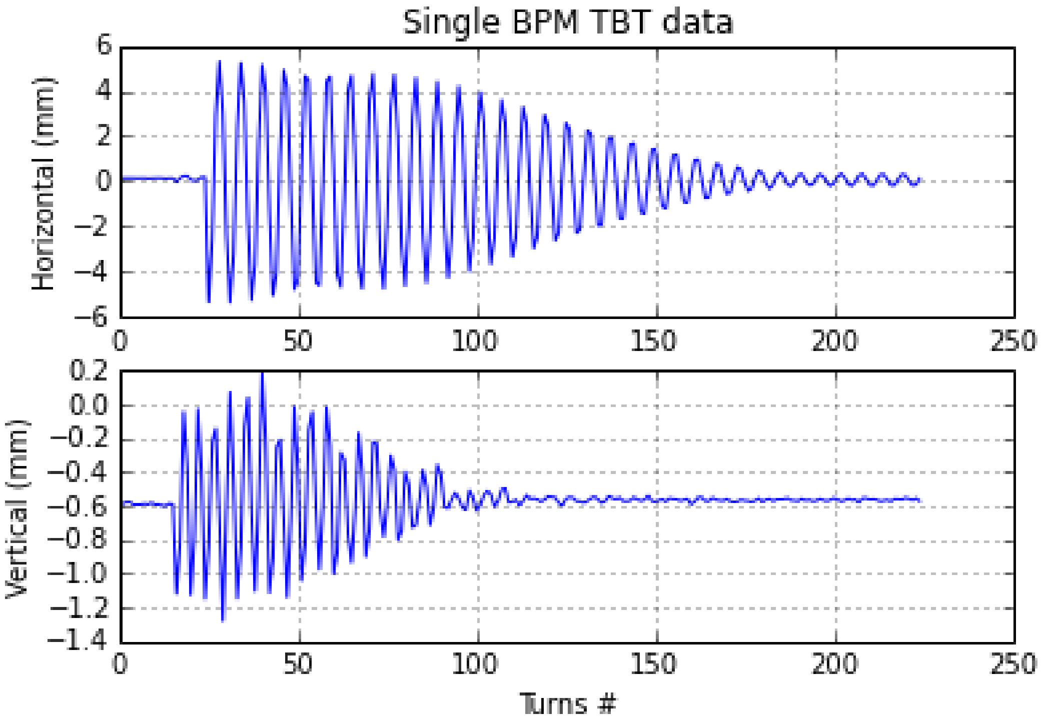
0.1 Mixed BPM tune measurements

```
In [3]: %matplotlib inline

        import numpy as np
        import matplotlib.pyplot as plt
        import mpld3
        from pylab import *
        from tuneDiagram import tuneDiagram

        mpld3.enable_notebook()
```

Ignore this. Just importing and polishing up data here

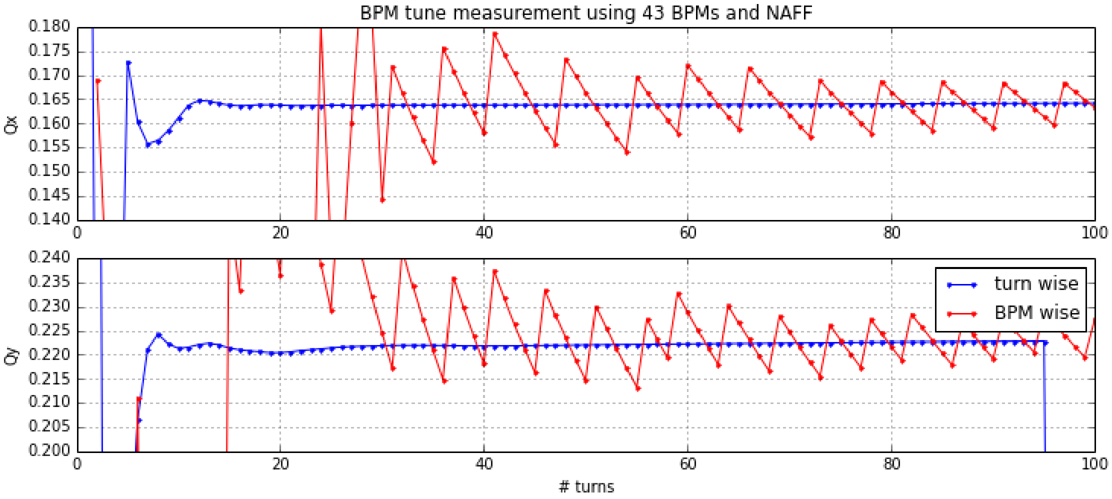
```
In [34]: mix_2 = np.genfromtxt('NAFF_mixed_2BPM.txt',dtype=np.str)
         stack_2 = np.genfromtxt('NAFF_stacked_2BPM.txt',dtype=np.str)
         mix_3 = np.genfromtxt('NAFF_mixed_3BPM.txt',dtype=np.str)
         stack_3 = np.genfromtxt('NAFF_stacked_3BPM.txt',dtype=np.str)
         mix_4 = np.genfromtxt('NAFF_mixed_4BPM.txt',dtype=np.str)
         stack_4 = np.genfromtxt('NAFF_stacked_4BPM.txt',dtype=np.str)
         mix_6 = np.genfromtxt('NAFF_mixed_6BPM.txt',dtype=np.str)
         stack_6 = np.genfromtxt('NAFF_stacked_6BPM.txt',dtype=np.str)
         mix_12 = np.genfromtxt('NAFF_mixed_12BPM.txt',dtype=np.str)
         stack_12 = np.genfromtxt('NAFF_stacked_12BPM.txt',dtype=np.str)
         mix_43 = np.genfromtxt('NAFF_mixed.txt',dtype=np.str)
         stack_43 = np.genfromtxt('NAFF_stacked.txt',dtype=np.str)
         TBT_Pinger = np.genfromtxt('TBT_BPM21.txt')

         data = np.array([mix_2, stack_2, mix_3, stack_3, mix_4,
                          stack_4, mix_6, stack_6, mix_12, stack_12, mix_43, stack_43])

         # Remove NaN in data and change dtype to float
         for k in range(0,len(data)):
             for i,clm in enumerate(data[k].T):
                 for j,tmp in enumerate(clm):
                     if tmp =='NaN':
                         data[k][j,i] = np.nan
                     else:
                         data[k][j,i] = float(tmp)
```

```
In [5]: figure(5)
        plt.subplot(2,1,1)
        plt.plot(data5[:,0])
        plt.grid(True)
        plt.title('Single BPM TBT data')
        plt.ylabel('Horizontal (mm)')
        plt.subplot(2,1,2)
        plt.plot(data5[:,1])
        plt.grid(True)
        plt.ylabel('Vertical (mm)')
        plt.xlabel('Turns #')
        plt.show()
```



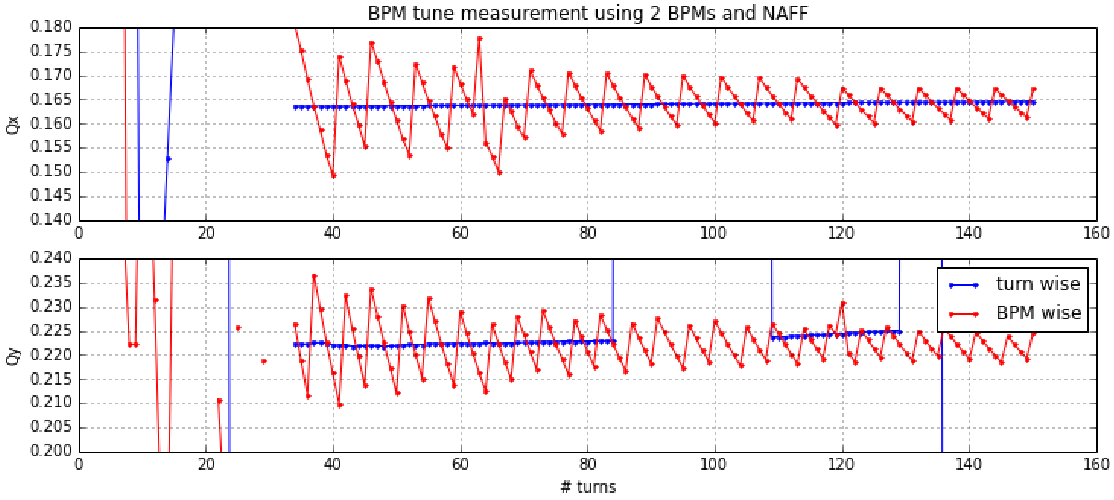
0.2 Turns vs Tune measurement using all 43 BPMs

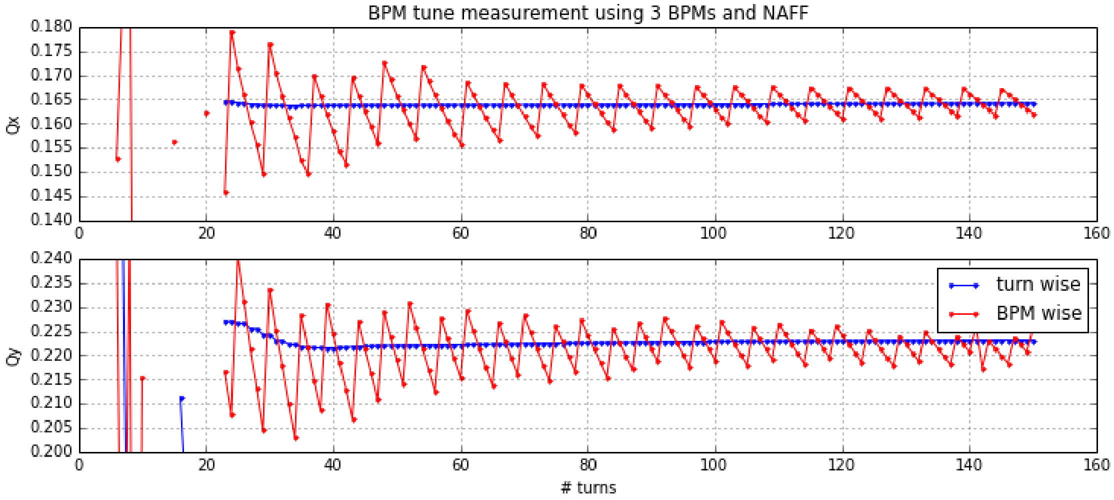
```
In [50]: figure(1,figsize=(12,5))
         plt.subplot(2,1,1)
         plt.plot(data[10][:,2],data[10][:,0],marker='.',color='b')
         plt.plot(data[11][:,2],data[11][:,0],marker='.',linestyle='-',color='r')
         plt.grid(True)
         plt.ylabel('Qx')
         plt.title('BPM tune measurement using 43 BPMs and NAFF')
         plt.gca().set_ylim([0.14, 0.18])

         plt.subplot(2,1,2)
         plt.plot(data[10][:,2],data[10][:,1],marker='.',color='b')
         plt.plot(data[11][:,2],data[11][:,1],marker='.',linestyle='-',color='r')
         plt.grid(True)
         plt.xlabel('# turns')
         plt.ylabel('Qy')
         plt.gca().set_ylim([0.20, 0.24])
         plt.legend(('turn wise','BPM wise'))
         plt.show()
```

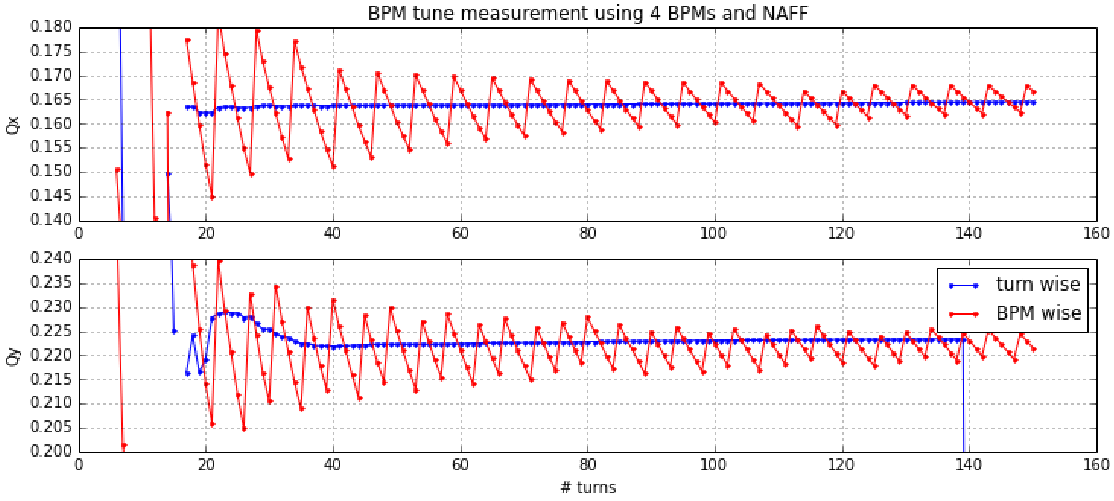


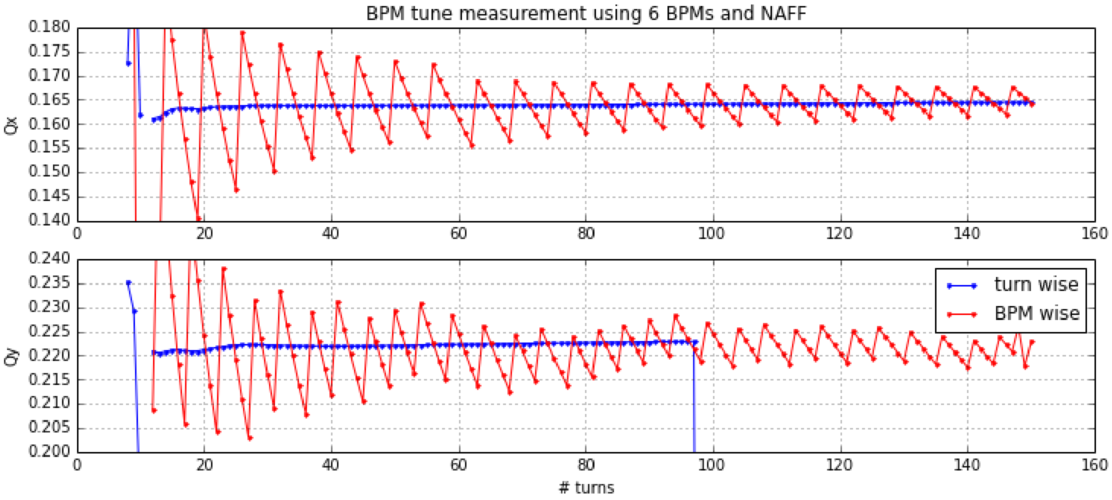
0.3 Tune measurements with 2,3,4,6,12 symmetric BPMs mixed/stacked together

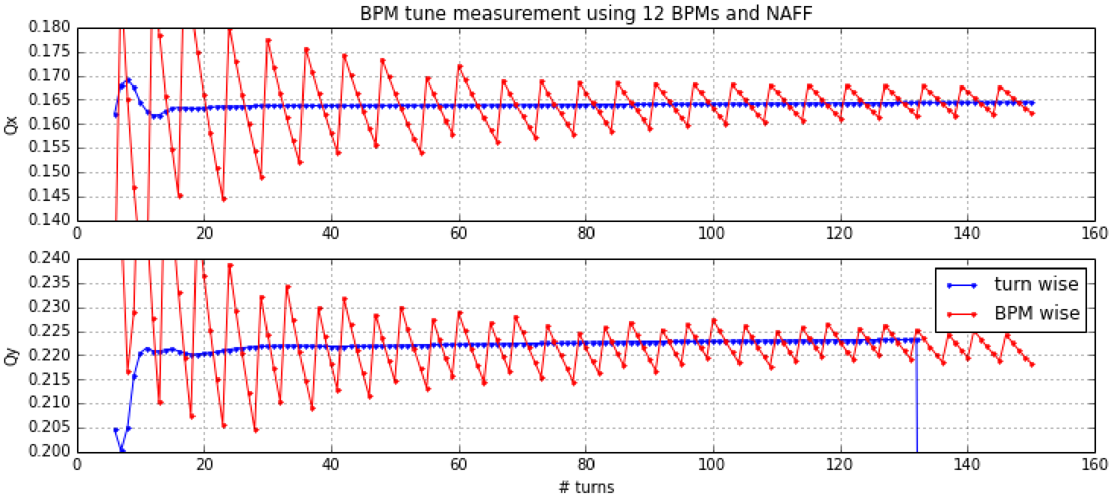
```
In [70]: title_num = [2, 3, 4, 6, 12]
         i = 0
         while i < (len(data)-3):
             figure(i+2,figsize=(12,5))
             plt.subplot(2,1,1)
             plt.plot(data[i][:,2],data[i][:,0],marker='.',color='b')
             plt.plot(data[i+1][:,2],data[i+1][:,0],marker='.',linestyle='-',color='r')
             plt.grid(True)
             plt.ylabel('Qx')
             title = 'BPM tune measurement using ' + str(title_num[i/2]) + ' BPMs and NAFF'
             plt.title(title)
             plt.gca().set_ylim([0.14, 0.18])

             plt.subplot(2,1,2)
             plt.plot(data[i][:,2],data[i][:,1],marker='.',color='b')
             plt.plot(data[i+1][:,2],data[i+1][:,1],marker='.',linestyle='-',color='r')
             plt.grid(True)
             plt.xlabel('# turns')
             plt.ylabel('Qy')
             plt.gca().set_ylim([0.20, 0.24])
             plt.legend(('turn wise','BPM wise'))
             plt.show()
             i+=2
```


BPM tune measurement using 2 BPMs and NAFF
Qx
0.180
0.175
0.170
0.165
0.160
0.155
0.150
0.145
0.140
Qy
0.240
0.235
0.230
0.225
0.220
0.215
0.210
0.205
0.200
0
20
40
60
80
100
120
140
160
# turns
turn wise
BPM wise


BPM tune measurement using 3 BPMs and NAFF
Qx
0.180
0.175
0.170
0.165
0.160
0.155
0.150
0.145
0.140
Qy
0.240
0.235
0.230
0.225
0.220
0.215
0.210
0.205
0.200
0
20
40
60
80
100
120
140
160
# turns
turn wise
BPM wise

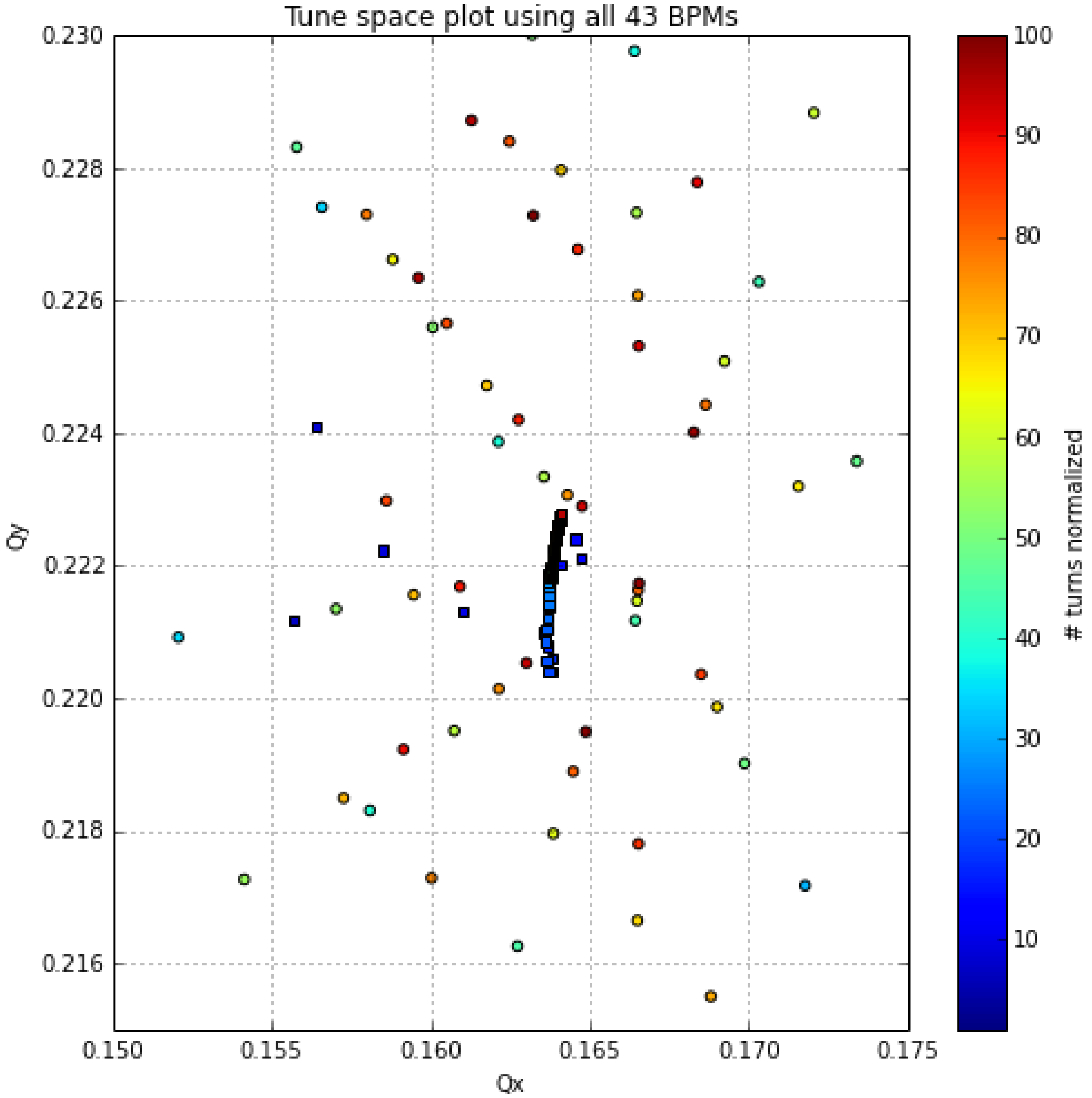

BPM tune measurement using 4 BPMs and NAFF
Qx
0.180
0.175
0.170
0.165
0.160
0.155
0.150
0.145
0.140
Qy
0.240
0.235
0.230
0.225
0.220
0.215
0.210
0.205
0.200
0
20
40
60
80
100
120
140
160
# turns
turn wise
BPM wise





```
In [90]: figure(10,figsize=(8,8))
         test = np.arange(1,101)
         plt.scatter(data[10][:,0],data[10][:,1],marker='s',c=test)
         plt.scatter(data[11][:,0],data[11][:,1],marker='o',c=test)
         cb = plt.colorbar()
         cb.set_label('# turns normalized')
         plt.axis((0.15, 0.175, 0.215, 0.23))
         #plt.legend(('turn wise','BPM wise'))
         plt.grid(True)
         plt.title('Tune space plot using all 43 BPMs')
         plt.xlabel('Qx')
         plt.ylabel('Qy')
```

```
Out[90]: <matplotlib.text.Text at 0x7efc82ff0e10>
```


Tune space plot using all 43 BPMs
Qy
0.230
0.228
0.226
0.224
0.222
0.220
0.218
0.216
Qx
0.150
0.155
0.160
0.165
0.170
0.175
# turns normalized
100
90
80
70
60
50
40
30
20
10

```
In [ ]:
```