# Saddle_h_Jan7_2021

January 9, 2021

[ ]:

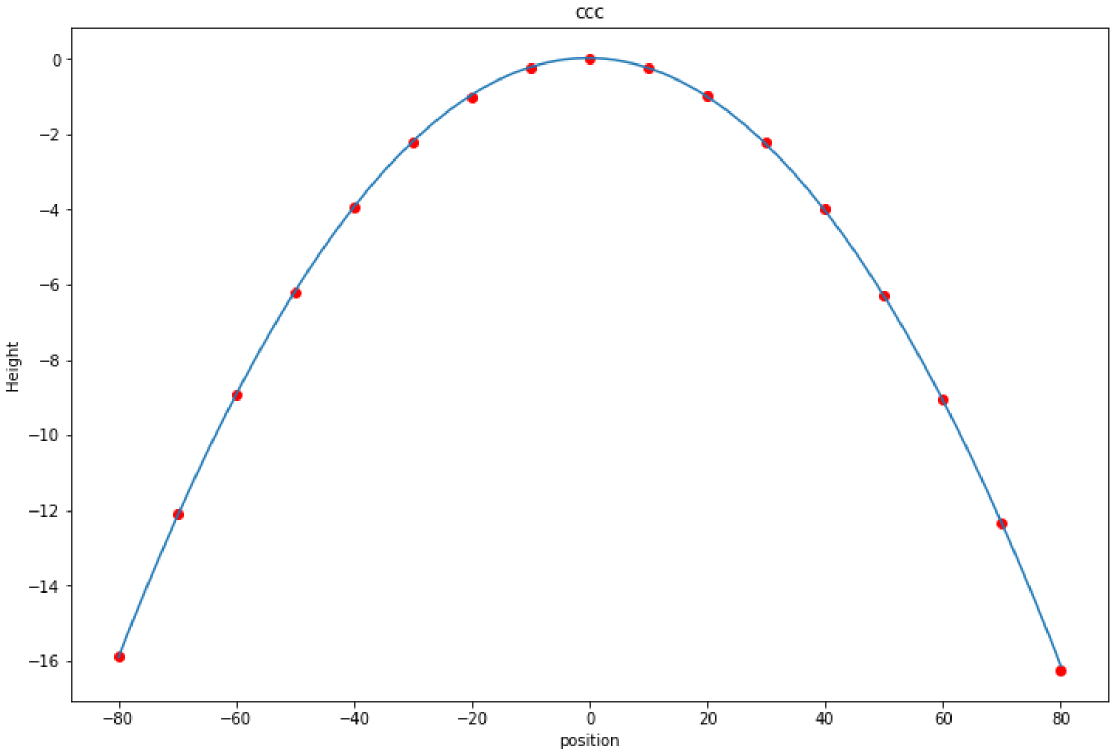
```
```

[35]:

```
  import numpy as np
  import matplotlib.pyplot as plt

  #Four saddles, measured Jan 7, 2021 NB 12/20 p6.

 #Saddle 1 (W1)
  pos = np.array([-80, -70, -60, -50, -40, -30, -20, -10, 0, 10, 20, 30, 40,␣
↪50, 60, 70, 80])
  z_1p = np.array([713, 554, 414, 284, 184, 103, 44, 9.5, 0.5, 6.1, 47.4, 108.
↪1, 189.5, 290.5, 416, 596, 766])
  z_1m = np.array([385, 534, 653, 755, 847, 914, 963, 993, 1000, 990, 962,␣
↪915, 845, 758, 649, 524, 379])
  z_1m = z_1m - 1000

  #Convert units
  z_1p = (z_1p/1000)*25.4     #to millimeters.
  z_1m = (z_1m/1000)*25.4     #to millimeters.

 #Saddle 2 (W2)
  z_2p = np.array([844, 671, 522, 390, 288, 205, 150, 114, 100, 112, 146,␣
↪199, 280, 381, 504, 656, 835 ])
  z_2m = np.array([302, 461, 601, 723, 824, 903, 956, 990, 1000, 990, 957,␣
↪904, 829, 729, 609, 470, 310])
  z_2m = z_2m - 1000

  #Convert units
  z_2p = (z_2p/1000)*25.4     #to millimeters.
  z_2m = (z_2m/1000)*25.4     #to millimeters.

 #Saddle 3 (P1)
  z_3p = np.array([674, 510, 370, 251, 157, 84, 34, 6, 0, 10, 48, 98, 176,␣
↪274, 395, 541, 705])
  z_3m = np.array([281, 452, 595, 719, 822, 900, 954, 987, 1000, 986, 950,␣
↪888, 805, 698, 569, 420, 249])
  z_3m = z_3m - 1000
```

```
    #Convert units
    z_3p = (z_3p/1000)*25.4     #to millimeters.
    z_3m = (z_3m/1000)*25.4     #to millimeters.

  #Saddle 4 (P2)
    z_4p = np.array([680, 521, 379, 258, 168, 94, 41, 10, 0, 11, 44, 93, 171, 268, 389, 527, 687])
    z_4m = np.array([385, 531, 654, 760, 847, 914, 961, 990, 1000, 991, 962, 914, 846, 756, 650, 522, 370])
    z_4m = z_4m - 1000

    #Convert units
    z_4p = (z_4p/1000)*25.4     #to millimeters.
    z_4m = (z_4m/1000)*25.4     #to millimeters.





    #Choose particular data set

    z=z_4m

    #make corrections due to finite radius of probe.
    #See NB 12/20 p8.
    #Correction is a function of position alone.
    R = 25.4/8     #radius is 1/8"
    poly_2 = -0.002506  #value from fit of square term
    poly_1 = -.00144 #value from fit of linear term
    slope = poly_1+2*poly_2*pos
    correction = 0 * slope    #test for zero correction.
    correction = R*(np.sqrt(1 + slope**2) - 1)
    z=z-correction      #All corrections decrease the number.


    #fit to polynomial
    poly_param = np.polyfit(pos, z, 2)
    p = np.poly1d(poly_param)
    x = np.linspace(-80,80,100)
    z_fit= p(x)


    #fig = plt.figure()
    fig = plt.figure(figsize=(12, 8))
    plt.title('ccc')
    plt.xlabel('position')
```

```
plt.ylabel('Height')

plt.plot(pos, z, 'ro')
plt.plot(x,z_fit)

plt.show()
poly_param
```



```
[35]: array([-0.00250565, -0.0014412 ,  0.02299574])
```