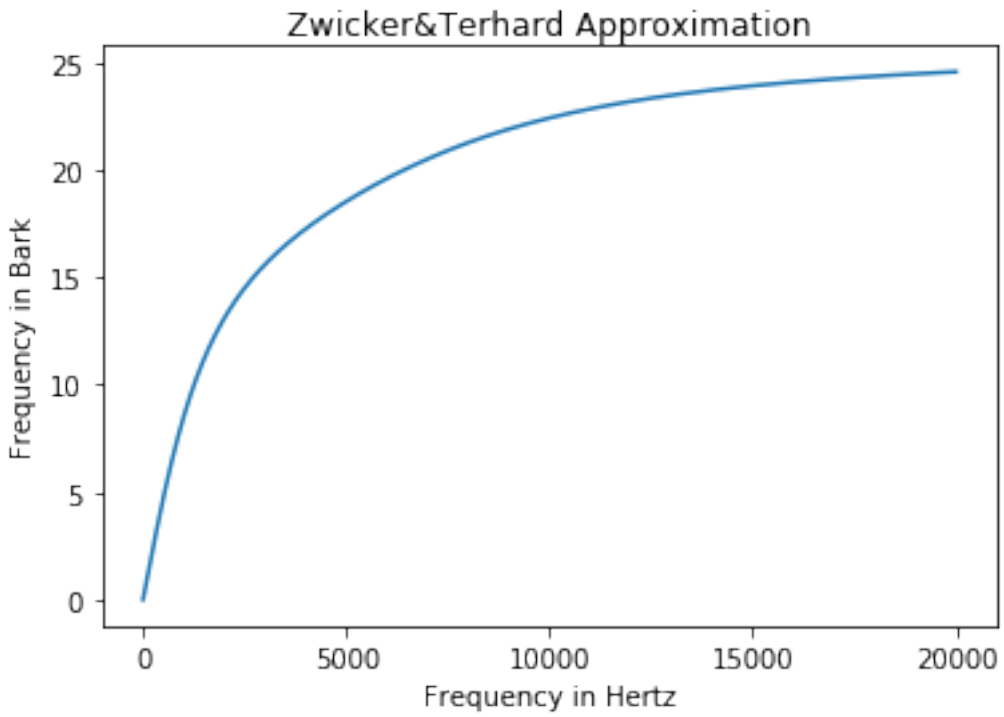
Zwicker&Terhard Bark scale approximation

January 27, 2017

0.1 We can test the Zwicker & Terhard approximation in ipython:

In [1]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np

#Frequency array between 0 and 20000 Hz in 1000 steps:
f = np.linspace(0, 20000, 1000)

#Computation of Zwickers Bark approximation formula:
z = 13 * np.arctan(0.00076 * f) + 3.5 * np.arctan((f / 7500.0) ** 2)

#plot Bark over Hertz:
plt.plot(f, z)
plt.xlabel('Frequency in Hertz')
plt.ylabel('Frequency in Bark')
plt.title('Zwicker&Terhard Approximation')
```

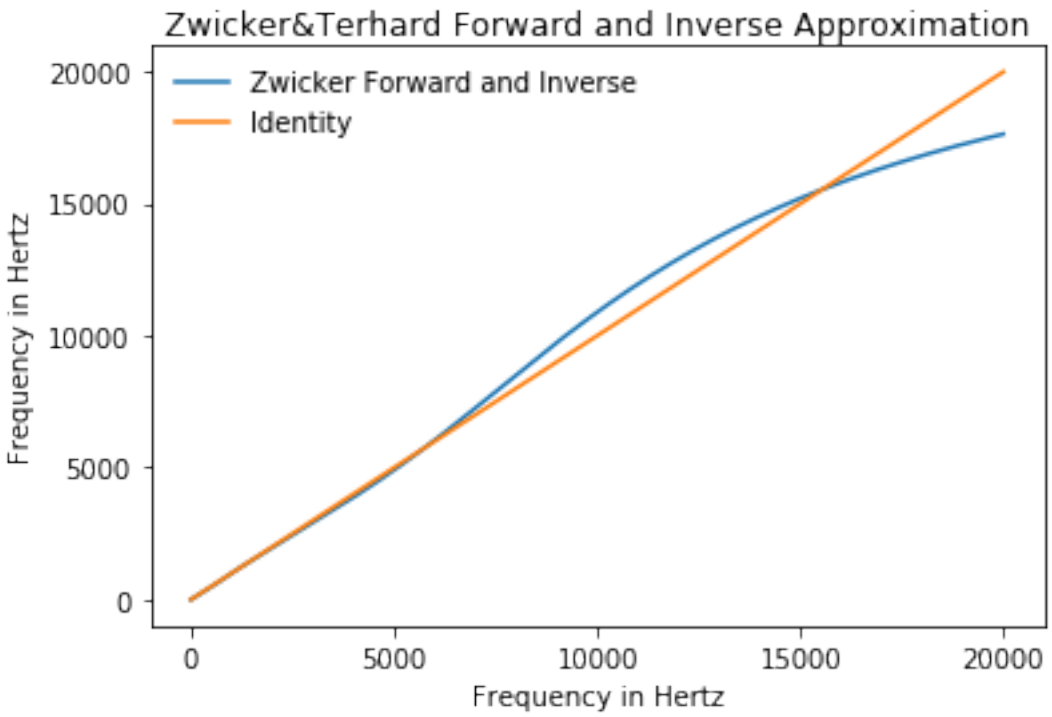
Out[1]:
```
<matplotlib.text.Text at 0x7db5430>
```



0.2 Bark Scale Approximations, Zwicker&Terhard, Inverse

We can test the Zwicker & Terhard inverse approximation in ipython

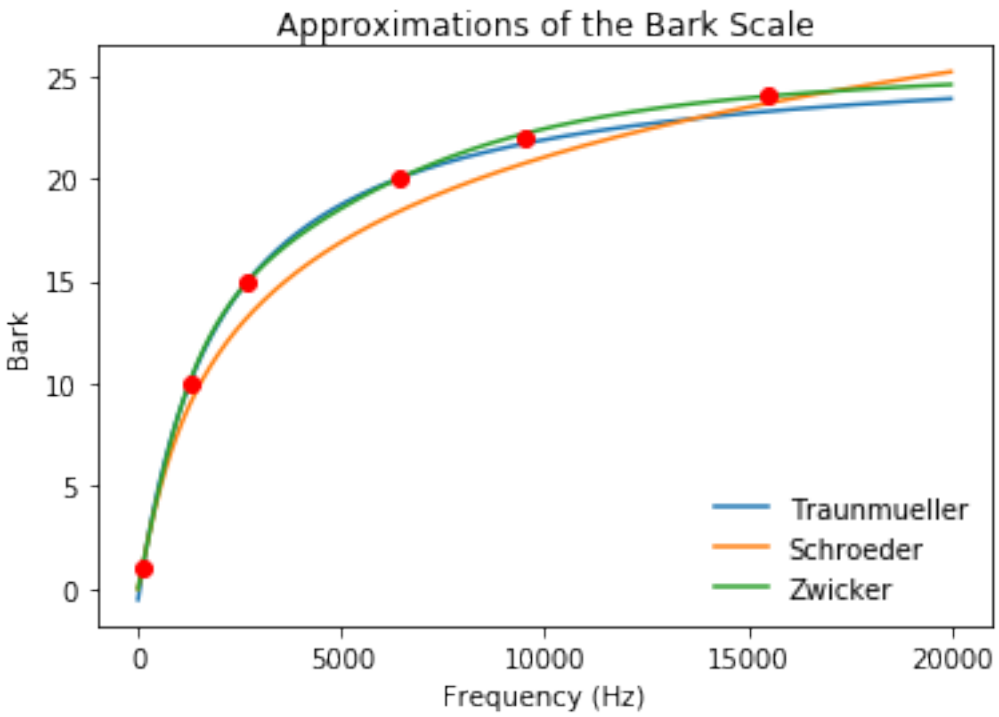
```
In [2]: f = np.linspace(0,20000,1000)

        #Computation of Zwickers Bark approximation formula:
        z = 13 * np.arctan(0.00076*f) + 3.5 * np.arctan((f/7500.0)**2)

        #computation of the approximate inverse, frec: reconstructed freq.:
        frec = (((np.exp(0.219 * z) / 352.0) + 0.1) * z - 0.032 * np.exp(-0.15 * (z - 5) ** 2))

        #plot reconstructed freq. Over original freq:
        plt. plot(f, frec)

        #comparison: identity:
        plt.plot(f, f)
        plt.xlabel('Frequency in Hertz')
        plt.ylabel('Frequency in Hertz')
        plt.title('Zwicker&Terhard Forward and Inverse Approximation')
        plt.legend(('Zwicker Forward and Inverse','Identity'))

Out[2]: <matplotlib.legend.Legend at 0x7e46830>
```



0.3 Bark Scale Approximations, Comparisons

- Use ipython for the comparison:

```
In [3]: f = np.arange(0, 20000, 10)
        z = 26.81 * f / (1960.0 + f) - 0.53 #Traunmueller
        plt.plot(f, z)
        z = 6 * np.arcsinh(f / 600.0) #Schroeder
        plt.plot(f, z)
        z=13 * np.arctan(0.00076 * f) + 3.5 * np.arctan(( f / 7500.0) ** 2) #Zwicker
        plt.plot(f, z)
        plt.legend(('Traunmueller','Schroeder','Zwicker'))

        #plot single comparison points:
        plt.plot([100,1270,2700,6400,9500,15500],[1,10,15,20,22,24],'ro')
        plt.xlabel('Frequency (Hz)')
        plt.ylabel('Bark')
        plt.title('Approximations of the Bark Scale')

Out[3]: <matplotlib.text.Text at 0x80c20f0>
```


Approximations of the Bark Scale
Bark
Frequency (Hz)
0
5
10
15
20
25
0
5000
10000
15000
20000
Traunmueller
Schroeder
Zwicker