Spreading Function

January 27, 2017

0.1 Example

- This Python example shows the non-linear superposition with parameter $2 * a = alpha = 0.6$**, in the Bark scale**. We construct a matrix which does the actual superposition in the Bark domain, because that is most efficient:

```
In [1]: import numpy as np

        def spreadingfunctionmat(maxfreq,nfilts,alpha):
            #Arguments: maxfreq: half the sampling frequency
            #nfilts: Number of subbands in the Bark domain, for instance 64
            fadB= 14.5+12 # Simultaneous masking for tones at Bark band 12
            fbdb=7.5 # Upper slope of spreading function
            fbbdb=26.0 # Lower slope of spreading function
            maxbark=hz2bark(maxfreq)
            spreadingfunctionBarkdB=np.zeros(2*nfilts)

            #upper slope, fbdB attenuation per Bark, over maxbark Bark (full frequency range), u
            spreadingfunctionBarkdB[0:nfilts]=np.linspace(-maxbark*fbdb,-2.5,nfilts)-fadB

            #lower slope fbbdb attenuation per Bark, over maxbark Bark (full frequency range):
            spreadingfunctionBarkdB[nfilts:2*nfilts]=np.linspace(0,-maxbark*fbbdb,nfilts)-fadB

            #Convert from dB to "voltage" and include alpha exponent
            spreadingfunctionBarkVoltage=10.0**(spreadingfunctionBarkdB/20.0*alpha)

            #Spreading functions for all bark scale bands in a matrix:
            spreadingfuncmatrix=np.zeros((nfilts,nfilts))
            for k in range(nfilts):
                spreadingfuncmatrix[:,k]=spreadingfunctionBarkVoltage[(nfilts-k):(2*nfilts-k)]
            return spreadingfuncmatrix
```

The above produces a prototype of spreading functions for all the bark bands(bark counts based on the resolution)

Below is the psyacmodel python example

```
In [2]: %matplotlib inline
        from psyacmodel import *
```

```
import matplotlib.pyplot as plt

fs=32000 # sampling frequency of audio signal
maxfreq=fs/2
alpha=0.6 #Exponent for non-linear superposition of spreading functions
nfilts=64 #number of subbands in the bark domain

spreadingfuncmatrix=spreadingfunctionmat(maxfreq,nfilts,alpha)

plt.imshow(spreadingfuncmatrix)
plt.title('Matrix spreadingfuncmatrix as Image')
plt.xlabel('Bark Domain Subbands')
plt.ylabel('Bark Domain Subbands')
plt.show()
```