

```

1
2
3     typedef double XFLOAT
4     typedef double OTA_FLOAT
5
6
7 {
8     void PrintFrequencyResponse(CNewLogFile &pResultsFile, const CBarkSpectrum
&pOriginalPitchPowerDensitySum, const CBarkSpectrum &pDistortedPitchPowerDensitySum, const
PolqaStatics *statics)
9     {
10         XFLOAT x
11         CNewStdString s
12
13         pResultsFile.WriteString("PERCEPTUAL FREQUENCY RESPONSE in dB\n")
14         pResultsFile.WriteString("Frequency (Bark)   Frequency (Hz)   Relative Level (dB)   Band
Index Number\n")
15         for (int bandIndex = 0  bandIndex < statics->aNumberOfBarkBands  bandIndex++)
16         {
17             if ((pOriginalPitchPowerDensitySum.m_pData[0][bandIndex] < 0.1) &&
(pDistortedPitchPowerDensitySum.m_pData[0][bandIndex] < 0.1))
18             {
19                 x = 98765.43
20             }
21             else
22             {
23                 x = 10 * log((XFLOAT)((pDistortedPitchPowerDensitySum.m_pData[0][bandIndex] + 0.01) /
(pOriginalPitchPowerDensitySum.m_pData[0][bandIndex] + 0.01))) / log(10.0)
24             }
25             s.Format("          %5.1f          %8.0f          %5.1f          %i \n",
statics->aCentreOfBandBark[bandIndex], statics->aCentreOfBandHz[bandIndex], x, bandIndex)
26             pResultsFile.WriteString(s)
27         }
28         pResultsFile.WriteString("\n\n")
29     }
30
31     XFLOAT CalculateFrequencyIndicator_SpeechLQ(const CPOLQAData *POLQAHandle, const CBarkSpectrum&
originalPitchPowerDensity_intact, const CBarkSpectrum& distortedPitchPowerDensity_intact, const
CDoubleArray& OriginalTotalPower, const CDoubleArray& DistortedTotalPower,
32     const int* ActiveRatioOkFlags, const XFLOAT GlobalCompensation1, const XFLOAT
PowerRatioAvgProduct, const CIntArray& SilentFrameFlags, const int numberOfSilentFrames,
XFLOAT PureFreqP1, const PolqaStatics *statics, CNewLogFile& ResultsFile)
33     {
34         CBarkSpectrum  originalPitchPowerDensityPureFrq, distortedPitchPowerDensityPureFrq,
originalLoudnessDensityPureFrq, distortedLoudnessDensityPureFrq
35         CBarkSpectrum  originalPitchPowerDensityAvgPureFrq, distortedPitchPowerDensityAvgPureFrq
36         CBarkSpectrum  originalPitchLoudnessDensityAvgPureFrq,
distortedPitchLoudnessDensityAvgPureFrq
37
38         CBarkSpectrum  originalPitchLoudnessDensityMainAvg, distortedPitchLoudnessDensityMainAvg
39         CIntArray      aBothActive
40         XFLOAT         aOriginalLoudnessPureFrqMean, aDistortedLoudnessPureFrqMean,
aLoudnessPureFrqScaling, loudnessPureFrqVar
41
42         XFLOAT aPureFrqLoudnessMean
43
44         int frameIndex
45         int numberOfSpeechFrames = statics->stopFrameIdx - statics->startFrameIdx
46         originalPitchPowerDensityPureFrq.Initialize("originalPitchPowerDensityPureFrq", POLQAHandle)
47         distortedPitchPowerDensityPureFrq.Initialize("distortedPitchPowerDensityPureFrq",
POLQAHandle)
48         originalLoudnessDensityPureFrq.Initialize("originalLoudnessDensityPureFrq", POLQAHandle)
49         originalPitchPowerDensityAvgPureFrq.Initialize("originalPitchPowerDensityAvgPureFrq",
POLQAHandle)
50         distortedPitchPowerDensityAvgPureFrq.Initialize("distortedPitchPowerDensityAvgPureFrq",
POLQAHandle)
51         originalPitchLoudnessDensityAvgPureFrq.Initialize("originalPitchLoudnessDensityAvgPureFrq",
POLQAHandle)

```

```

52     distortedPitchLoudnessDensityAvgPureFrq.Initialize("distortedPitchLoudnessDensityAvgPureFrq",
POLQAHandle)
53     distortedLoudnessDensityPureFrq.Initialize("distortedLoudnessDensityPureFrq", POLQAHandle)
54     originalPitchLoudnessDensityMainAvg.Initialize("originalPitchLoudnessDensityMainAvg",
POLQAHandle)
55     distortedPitchLoudnessDensityMainAvg.Initialize("distortedPitchLoudnessDensityMainAvg",
POLQAHandle)
56
57     originalPitchPowerDensityPureFrq.FrequencyWarpingOf(POLQAHandle, originalHzPowerSpectrum,
1.0)
58     distortedPitchPowerDensityPureFrq.FrequencyWarpingOf(POLQAHandle, distortedHzPowerSpectrum,
PitchRatio)
59
60     aBothActive.SetSize(statics->nrFrames)
61
62     XFLOAT hulp
63     XFLOAT hulp1 = 0.0
64     XFLOAT hulp2 = 0.0
65
66     for (frameIndex = statics->startFrameIdx  frameIndex <= statics->stopFrameIdx  frameIndex++)
67     {
68         if (ActiveRatioOkFlags[frameIndex])
69         {
70             hulp1 += originalPitchPowerDensityPureFrq.Total(frameIndex, 300.0, 5000.0)
71             hulp2 += distortedPitchPowerDensityPureFrq.Total(frameIndex, 300.0, 5000.0)
72         }
73     }
74     hulp1 /= (numberOfSpeechFrames + 0.01)
75     hulp2 /= (numberOfSpeechFrames + 0.01)
76     hulp1 = 3.0e8 / (hulp1*GlobalCompensation1 + 1.0)
77     hulp2 = 3.0e8 / (hulp2*GlobalCompensation1 + 1.0)
78     for (frameIndex = statics->startFrameIdx  frameIndex <= statics->stopFrameIdx  frameIndex++)
79     {
80         originalPitchPowerDensityPureFrq.MultiplyWith(frameIndex, hulp1)
81         distortedPitchPowerDensityPureFrq.MultiplyWith(frameIndex, hulp2)
82     }
83     distortedLoudnessDensityPureFrq.IntensityWarpingOf(POLQAHandle,
distortedPitchPowerDensityPureFrq)
84
85     for (frameIndex = statics->startFrameIdx  frameIndex <= statics->stopFrameIdx  frameIndex++)
86     {
87         OriginalTotalPower.m_pData[frameIndex] =
originalPitchPowerDensityPureFrq.TotalAudible(POLQAHandle, frameIndex, 1.0e2)
88         DistortedTotalPower.m_pData[frameIndex] =
distortedPitchPowerDensityPureFrq.TotalAudible(POLQAHandle, frameIndex, 1.0e2)
89         hulp = (DistortedTotalPower.m_pData[frameIndex] + 100.0) /
(OriginalTotalPower.m_pData[frameIndex] + 100.0)
90         if (hulp<1.0) hulp = 1 / hulp
91         if ((OriginalTotalPower.m_pData[frameIndex] > 1.0E5) &&
(DistortedTotalPower.m_pData[frameIndex] > 1.0E5) && (hulp < 24.0))
92             aBothActive.m_pData[frameIndex] = TRUE
93         else
94             aBothActive.m_pData[frameIndex] = FALSE
95     }
96
97     for (frameIndex = (statics->startFrameIdx + 2)  frameIndex <= (statics->stopFrameIdx)
frameIndex++)
98     {
99         if (!aBothActive.m_pData[frameIndex])
100         {
101             aBothActive.m_pData[frameIndex - 1] = FALSE
102             aBothActive.m_pData[frameIndex - 2] = FALSE
103         }
104     }
105     for (frameIndex = (statics->stopFrameIdx - 2)  frameIndex >= (statics->startFrameIdx)
frameIndex--)
106     {
107         if (!aBothActive.m_pData[frameIndex])

```

```

108     {
109         aBothActive.m_pData[frameIndex + 1] = FALSE
110         aBothActive.m_pData[frameIndex + 2] = FALSE
111     }
112 }
113
114     originalPitchPowerDensityAvgPureFrq.TimeLpAudibleOf(POLQAHandle,
originalPitchPowerDensityPureFrq, SilentFrameFlags, 1.0)
115     distortedPitchPowerDensityAvgPureFrq.TimeLpAudibleOf(POLQAHandle,
distortedPitchPowerDensityPureFrq, SilentFrameFlags, 1.0)
116     PrintFrequencyResponse(ResultsFile, originalPitchPowerDensityAvgPureFrq,
distortedPitchPowerDensityAvgPureFrq, statics)
117
118     originalLoudnessDensityPureFrq.IntensityWarpingOf(POLQAHandle,
originalPitchPowerDensityPureFrq)
119
120     originalPitchLoudnessDensityMainAvg.TimeLpAudibleOfSilent(originalLoudnessDensityPureFrq,
SilentFrameFlags, 3.0, numberOfSilentFrames)
121     distortedPitchLoudnessDensityMainAvg.TimeLpAudibleOfSilent(distortedLoudnessDensityPureFrq,
SilentFrameFlags, 3.0, numberOfSilentFrames)
122
123     originalLoudnessDensityPureFrq.AudibleNoiseRespCompensationOfPartly(POLQAHandle,
originalPitchLoudnessDensityMainAvg, 0.3)
124     distortedLoudnessDensityPureFrq.AudibleNoiseRespCompensationOfPartly(POLQAHandle,
distortedPitchLoudnessDensityMainAvg, 0.3)
125
126     originalPitchLoudnessDensityAvgPureFrq.TimeLpOf(POLQAHandle, originalLoudnessDensityPureFrq,
aBothActive, 2.0)
127     distortedPitchLoudnessDensityAvgPureFrq.TimeLpOf(POLQAHandle,
distortedLoudnessDensityPureFrq, aBothActive, 2.0)
128
129     {
130         SmartBufferPolqa SB1(POLQAHandle, statics->aNumberOfBarkBands)
131         XFLOAT *temp1 = SB1.Buffer
132         SmartBufferPolqa SB2(POLQAHandle, statics->aNumberOfBarkBands)
133         XFLOAT *temp2 = SB2.Buffer
134
135         matbSqrt2(originalPitchLoudnessDensityAvgPureFrq.m_pData[0], temp1,
statics->aNumberOfBarkBands)
136         matbSqrt2(temp1, temp2, statics->aNumberOfBarkBands)
137         aOriginalLoudnessPureFrqMean = pow(matSum(temp2, statics->aNumberOfBarkBands), 4)
138
139         matbSqrt2(distortedPitchLoudnessDensityAvgPureFrq.m_pData[0], temp1,
statics->aNumberOfBarkBands)
140         matbSqrt2(temp1, temp2, statics->aNumberOfBarkBands)
141         aDistortedLoudnessPureFrqMean = pow(matSum(temp2, statics->aNumberOfBarkBands), 4)
142
143         aLoudnessPureFrqScaling = (aOriginalLoudnessPureFrqMean + 1e-10) /
(aDistortedLoudnessPureFrqMean + 1e-10)
144     }
145
146     loudnessPureFrqVar = 0.0
147     XFLOAT hulpMem = 0.0
148
149     for (int bandIndex = 0 bandIndex < statics->aNumberOfBarkBands bandIndex++)
150     {
151         hulp = ((originalPitchLoudnessDensityAvgPureFrq.m_pData[0][bandIndex] -
aLoudnessPureFrqScaling*distortedPitchLoudnessDensityAvgPureFrq.m_pData[0][bandIndex]))
152         if (hulp < 0) hulp1 = -((-hulp) * sqrt(-hulp))
153         else if (hulp >= 0) hulp1 = hulp * sqrt(hulp)
154         if (hulpMem < 0) hulp2 = -((-hulpMem) * sqrt(-hulpMem))
155         else if (hulpMem >= 0) hulp2 = hulpMem * sqrt(hulpMem)
156         loudnessPureFrqVar += fabs(hulp1 - hulp2) * pow((bandIndex + 1.0), 0.4)
157         hulpMem = hulp
158     }
159
160     aPureFrqLoudnessMean = 0.0
161     for (int bandIndex = 0 bandIndex < statics->aNumberOfBarkBands bandIndex++)

```

```
162     {
163         if ((originalPitchLoudnessDensityAvgPureFrq.m_pData[0][bandIndex])
164             >(aLoudnessPureFrqScaling*distortedPitchLoudnessDensityAvgPureFrq.m_pData[0][bandIndex]))
165         {
166             hulp = (originalPitchLoudnessDensityAvgPureFrq.m_pData[0][bandIndex] -
167                 aLoudnessPureFrqScaling*distortedPitchLoudnessDensityAvgPureFrq.m_pData[0][bandIndex]
168             )
169             aPureFrqLoudnessMean += sqrt(hulp)
170         }
171         else
172         {
173             hulp =
174             (aLoudnessPureFrqScaling*distortedPitchLoudnessDensityAvgPureFrq.m_pData[0][bandIndex]
175             ] - originalPitchLoudnessDensityAvgPureFrq.m_pData[0][bandIndex])
176             aPureFrqLoudnessMean += (PureFreqP1*sqrt(hulp))
177         }
178     }
179     aPureFrqLoudnessMean = aPureFrqLoudnessMean * loudnessPureFrqVar * PowerRatioAvgProduct
180     return aPureFrqLoudnessMean
181 }
```