

```

1
2
3     typedef double XFLOAT
4     typedef double OTA_FLOAT
5
6     typedef double OTA_FLOAT
7     typedef MAT_DCplx OTA_CPLX
8
9
10  {
11
12  typedef struct
13  {
14      float FrameWeightWeight
15      bool   UseRelDistance
16      float ViterbiDistanceWeightFactor
17  } VITERBI_PARA
18
19  typedef struct
20  {
21      long Samplerate
22      int  mSRDetectFineAlignCorrlen
23      int  mDelayFineAlignCorrlen
24      int  WindowSize[8]
25      int  CoarseAlignCorrlen[8]
26      float pViterbiDistanceWeightFactor[8]
27  } SPEECH_WINDOW_PARA
28
29  typedef struct
30  {
31      SPEECH_WINDOW_PARA Win[3]
32      float LowEnergyThresholdFactor
33      float LowCorrelThreshold
34
35      float FineAlignLowEnergyThresh
36      float FineAlignLowEnergyCorrel
37      float FineAlignShortDropOfCorrelR
38      float FineAlignShortDropOfCorrelRLastBest
39      float ViterbiDistanceWeightFactorDist
40      float ViterbiDistanceWeightFactor
41  } SPEECH_TA_PARA
42
43  typedef struct
44  {
45      SPEECH_WINDOW_PARA Win[3]
46      float LowEnergyThresholdFactor
47      float LowCorrelThreshold
48
49      float FineAlignLowEnergyThresh
50      float FineAlignLowEnergyCorrel
51      float FineAlignShortDropOfCorrelR
52      float FineAlignShortDropOfCorrelRLastBest
53      float ViterbiDistanceWeightFactorDist
54      float ViterbiDistanceWeightFactor
55  } AUDIO_TA_PARA
56
57  typedef struct
58  {
59      float mCorrForSkippingInitialDelaySearch
60      int  CoarseAlignSegmentLengthInMs
61  } GENERAL_TA_PARA
62
63  typedef struct
64  {
65      void Init(long Samplerate)
66      {
67          if (Samplerate==16000)    MaxWin=4
68          else if (Samplerate==8000) MaxWin=4

```

```

69         else                                     MaxWin=4
70
71         LowPeakEliminationThreshold= 0.2000000029802322
72
73         if (Samplerate==16000)      PercentageRequired = 0.05F
74         else if (Samplerate==8000)  PercentageRequired = 0.1F
75         else                        PercentageRequired = 0.02F
76
77         MaxDistance = 14
78
79         MinReliability = 7
80
81         PercentageRequired = 0.7
82         OTA_FLOAT MaxGradient = 1.1
83         OTA_FLOAT MaxTimescaling = 0.1
84
85         if (Samplerate==48000)      MaxStepPerFrame = MaxGradient * 1024.0
86         else if (Samplerate==8000)  MaxStepPerFrame = MaxGradient * 128.0
87         MaxBins = ((int)(MaxStepPerFrame*2.0*0.9))
88         MaxStepPerFrame *= 4
89
90     }
91
92     float LowEnergyThresholdFactor
93     float LowCorrelThreshold
94
95     int     MaxStepPerFrame
96     int     MaxBins
97     int     MaxWin
98     int     MinHistogramData
99
100    float   MinReliability
101
102    double  LowPeakEliminationThreshold
103    float   MinFrequencyOfOccurrence
104    float   LargeStepLimit
105
106    float   MaxDistanceToLast
107    float   MaxDistance
108    float   MaxLargeStep
109
110    float   ReliabilityThreshold
111    float   PercentageRequired
112
113    float   AllowedDistancePara2
114    float   AllowedDistancePara3
115 } SR_ESTIMATION_PARA
116
117 class CParameters
118 {
119     public:
120         CParameters()
121         {
122             int i
123             mTAPara.mCorrForSkippingInitialDelaySearch = 0.6F
124             mTAPara.CoarseAlignSegmentLengthInMs = 600
125
126             SPEECH_WINDOW_PARA    SpeechWinPara[] =
127             {
128                 {8000, 32, 32,
129                  {128, 256, 128, 64, 32, 0, 0},
130                  {-1, -1, -1, 86, 34, 0, 0},
131                  {-1, -1, -1, 15, 12, 0, 0}},
132                 {16000, 64, 64,
133                  {256, 512, 256, 128, 64, 0},
134                  {-1, -1, -1, 63, 33, 0},
135                  {-1, -1, -1, 13, 10, 0}},
136                 {48000, 256, 256,

```

```

137         {512, 1024, 512, 512, 128, 0},
138         {-1, -1, -1, 115, 61, 0},
139         {-1, -1, -1, 17, 16, 0}}
140     }
141
142     for (i=0 i<3 i++)
143     {
144         mSpeechTAPara.Win[i].Samplerate = SpeechWinPara[i].Samplerate
145         mSpeechTAPara.Win[i].mDelayFineAlignCorrlen =
SpeechWinPara[i].mDelayFineAlignCorrlen
146         mSpeechTAPara.Win[i].mSRDetectFineAlignCorrlen =
SpeechWinPara[i].mSRDetectFineAlignCorrlen
147         for (int k=0 k<8 k++)
148         {
149             mSpeechTAPara.Win[i].CoarseAlignCorrlen[k] =
SpeechWinPara[i].CoarseAlignCorrlen[k]
150             mSpeechTAPara.Win[i].WindowSize[k] = SpeechWinPara[i].WindowSize[k]
151
152             mSpeechTAPara.Win[i].pViterbiDistanceWeightFactor[k] =
SpeechWinPara[i].pViterbiDistanceWeightFactor[k]
153         }
154         mSpeechTAPara.LowEnergyThresholdFactor = 15.0F
155         mSpeechTAPara.LowCorrelThreshold = 0.4F
156         mSpeechTAPara.FineAlignLowEnergyThresh = 2.0
157         mSpeechTAPara.FineAlignLowEnergyCorrel = 0.6F
158         mSpeechTAPara.FineAlignShortDropOfCorrelR = -1
159         mSpeechTAPara.FineAlignShortDropOfCorrelRLastBest = 0.65F
160
161         mSpeechTAPara.ViterbiDistanceWeightFactorDist = 5
162
163         SPEECH_WINDOW_PARA AudioWinPara[] =
164         {
165             {8000, 32, 32,
166              {64, 128, 64, 64, 16, 0, 0},
167              {-1, -1, -1, 128, 32, 0, 0},
168              {-1, -1, -1, 6, 6, 0, 0}},
169             {16000, 64, 64,
170              {128, 256, 128, 128, 32, 0},
171              {-1, -1, -1, 64, 32, 0},
172              {-1, -1, -1, 12, 12, 0}},
173             {48000, 256, 2048,
174              {512, 1024, 512, 512, 256, 128, 0},
175              {-1, -1, -1, 512, 1024, 2048, 0},
176              {-1, -1, -1, 16, 16, 32, 0}}
177         }
178
179         for (i=0 i<3 i++)
180         {
181             mAudioTAPara.Win[i].Samplerate = AudioWinPara[i].Samplerate
182             mAudioTAPara.Win[i].mDelayFineAlignCorrlen =
AudioWinPara[i].mDelayFineAlignCorrlen
183             mAudioTAPara.Win[i].mSRDetectFineAlignCorrlen =
AudioWinPara[i].mSRDetectFineAlignCorrlen
184             for (int k=0 k<8 k++)
185             {
186                 mAudioTAPara.Win[i].CoarseAlignCorrlen[k] =
AudioWinPara[i].CoarseAlignCorrlen[k]
187                 mAudioTAPara.Win[i].WindowSize[k] = AudioWinPara[i].WindowSize[k]
188                 mAudioTAPara.Win[i].pViterbiDistanceWeightFactor[k] =
AudioWinPara[i].pViterbiDistanceWeightFactor[k]
189             }
190         }
191         mAudioTAPara.LowEnergyThresholdFactor = 1
192         mAudioTAPara.LowCorrelThreshold = 0.85F
193         mAudioTAPara.FineAlignLowEnergyThresh = 32.0
194         mAudioTAPara.FineAlignLowEnergyCorrel = 0.8F
195         mAudioTAPara.FineAlignShortDropOfCorrelR = -1

```

```

196     mAudioTAPara.FineAlignShortDropOfCorrelLastBest = 0.8F
197     mAudioTAPara.ViterbiDistanceWeightFactorDist = 6
198
199     mSREPara.LowEnergyThresholdFactor = 15.0F
200     mSREPara.LowCorrelThreshold = 0.4F
201
202     mSREPara.MaxStepPerFrame = 160
203     mSREPara.MaxBins = ((int)(mSREPara.MaxStepPerFrame*2.0*0.9))
204
205     mSREPara.MaxWin=4
206     mSREPara.LowPeakEliminationThreshold=0.200000029802322F
207     mSREPara.PercentageRequired = 0.04F
208
209     mSREPara.LargeStepLimit = 0.08F
210     mSREPara.MaxDistanceToLast = 7
211     mSREPara.MaxLargeStep = 5
212     mSREPara.MaxDistance = 14
213
214     mSREPara.MinReliability = 7
215     mSREPara.MinFrequencyOfOccurrence = 3
216
217     mSREPara.AllowedDistancePara2 = 0.85F
218     mSREPara.AllowedDistancePara3 = 1.5F
219
220     mSREPara.ReliabilityThreshold = 0.3F
221     mSREPara.MinHistogramData = 8
222
223     mViterbi.UseRelDistance = false
224     mViterbi.FrameWeightWeight = 1.0F
225 }
226
227 void Init(long Samplerate)
228 {
229     mSREPara.Init(Samplerate)
230 }
231
232 VITERBI_PARA      mViterbi
233 GENERAL_TA_PARA   mTAPara
234 SPEECH_TA_PARA    mSpeechTAPara
235 AUDIO_TA_PARA     mAudioTAPara
236 SR_ESTIMATION_PARA mSREPara
237 }
238 }
239
240
241 {
242
243 class CProcessData
244 {
245     public:
246     CProcessData()
247     {
248         int i
249
250         mCurrentIteration = -1
251         mStartPlotIteration=10
252         mLastPlotIteration =10
253         mEnablePlotting=false
254         mpLogFile = 0
255
256         mWindowSize = 2048
257         mSRDetectFineAlignCorrlen = 1024
258         mDelayFineAlignCorrlen = 1024
259         mOverlap = 1024
260         mSamplerate = 48000
261         mNumSignals = 0
262         mpMathlibHandle = 0
263         mMinLowVarDelay = -99999999

```

```

264         mMaxHighVarDelay = 9999999
265
266         mMinStaticDelayInMs = -2500
267         mMaxStaticDelayInMs = 2500
268
269         mMaxToleratedRelativeSamplerateDifference = 1.0
270
271         for (i=0 i<8 i++)
272             mpViterbiDistanceWeightFactor[i] = 0.0001F
273     }
274
275     int mMinStaticDelayInMs
276     int mMaxStaticDelayInMs
277
278     int mMinLowVarDelayInSamples
279     int mMaxHighVarDelayInSamples
280
281     int mStartPlotIteration
282     int mLastPlotIteration
283     bool mEnablePlotting
284     long mSamplerate
285
286     FILE* mpLogFile
287
288     int mCurrentIteration
289
290     int mpWindowSize[8]
291
292     int mpOverlap[8]
293
294     int mpCoarseAlignCorrlen[8]
295
296     float mpViterbiDistanceWeightFactor[8]
297
298     int mDelayFineAlignCorrlen
299     int mSRDetectFineAlignCorrlen
300     float mMaxToleratedRelativeSamplerateDifference
301     int mWindowSize
302
303     int mOverlap
304
305     int mCoarseAlignCorrlen
306
307     int mNumSignals
308     void* mpMathlibHandle
309
310     int mMinLowVarDelay
311     int mMaxHighVarDelay
312     int mStepSize
313
314     bool Init(int Iteration, float MoreDownsampling)
315     {
316         assert(MoreDownsampling)
317
318         mCurrentIteration = Iteration
319         mP.Init(mSamplerate)
320
321         mWindowSize = (int)((float)mpWindowSize[Iteration]*MoreDownsampling)
322         mOverlap = (int)((float)mpOverlap[Iteration]*MoreDownsampling)
323         mCoarseAlignCorrlen = mpCoarseAlignCorrlen[Iteration]
324         mStepSize = mWindowSize - mOverlap
325         mMinLowVarDelay = mMinLowVarDelayInSamples / mStepSize
326         mMaxHighVarDelay = mMaxHighVarDelayInSamples / mStepSize
327
328         float D = mpViterbiDistanceWeightFactor[Iteration]
329         D = D * mSamplerate / mStepSize / 1000
330         float F = ((float)log(1+0.5)) / (D*D)
331         mP.mViterbi.ViterbiDistanceWeightFactor = F

```

```

332         D = mP.mSpeechTAPara.ViterbiDistanceWeightFactorDist
333         D = D * mSamplerate / 1000
334         F = ((float) log(1+0.5) / (D*D))
335         mP.mSpeechTAPara.ViterbiDistanceWeightFactor = F
336
337         return true
338     }
339 }
340
341 CParameters    mP
342 }
343
344 class SECTION
345 {
346     public:
347         int Start
348         int End
349         int Len() {return End-Start }
350         void CopyFrom(const SECTION &src)
351         {
352             this->Start = src.Start
353             this->End    = src.End
354         }
355     }
356
357 typedef struct OTA_RESULT
358 {
359     void CopyFrom(const OTA_RESULT* src)
360     {
361         mNumFrames      = src->mNumFrames
362         mStepsize        = src->mStepsize
363         mResolutionInSamples = src->mResolutionInSamples
364         if (src->mpDelay != NULL && mNumFrames > 0)
365         {
366             matFree(mpDelay)
367             mpDelay = (long*)matMalloc(mNumFrames * sizeof(long))
368             for (int i = 0 i < mNumFrames i++)
369                 mpDelay[i] = src->mpDelay[i]
370         }
371         else
372         {
373             matFree(mpDelay)
374             mpDelay = NULL
375         }
376
377         if (src->mpReliability != NULL && mNumFrames > 0)
378         {
379             matFree(mpReliability)
380             mpReliability = (OTA_FLOAT*)matMalloc(mNumFrames * sizeof(OTA_FLOAT))
381             for (int i = 0 i < mNumFrames i++)
382                 mpReliability[i] = src->mpReliability[i]
383         }
384         else
385         {
386             matFree(mpReliability)
387             mpReliability = NULL
388         }
389         mAvgReliability    = src->mAvgReliability
390         mRelSamplerateDev   = src->mRelSamplerateDev
391
392         mNumUtterances = src->mNumUtterances
393         if (src->mpStartSampleUtterance != NULL && mNumUtterances > 0)
394         {
395             matFree(mpStartSampleUtterance)
396             mpStartSampleUtterance = (int*)matMalloc(mNumUtterances * sizeof(int))
397             for (int i = 0 i < mNumUtterances i++)
398                 mpStartSampleUtterance[i] = src->mpStartSampleUtterance[i]
399         }

```

```

400     else
401     {
402         matFree(mpStartSampleUtterance)
403         mpStartSampleUtterance = NULL
404     }
405     if (src->mpStopSampleUtterance != NULL && mNumUtterances > 0)
406     {
407         matFree(mpStopSampleUtterance)
408         mpStopSampleUtterance = (int*)matMalloc(mNumUtterances * sizeof(int))
409         for (int i = 0 i < mNumUtterances i++)
410             mpStopSampleUtterance[i] = src->mpStopSampleUtterance[i]
411     }
412     else
413     {
414         matFree(mpStopSampleUtterance)
415         mpStopSampleUtterance = NULL
416     }
417     if (src->mpDelayUtterance != NULL && mNumUtterances > 0)
418     {
419         matFree(mpDelayUtterance)
420         mpDelayUtterance = (int*)matMalloc(mNumUtterances * sizeof(int))
421         for (int i = 0 i < mNumUtterances i++)
422             mpDelayUtterance[i] = src->mpDelayUtterance[i]
423     }
424     else
425     {
426         matFree(mpDelayUtterance)
427         mpDelayUtterance = NULL
428     }
429
430     mNumSections = src->mNumSections
431     if (src->mpRefSections != NULL && mNumSections > 0)
432     {
433         delete[] mpRefSections
434         mpRefSections = new SECTION[mNumSections]
435         for (int i = 0 i < mNumSections i++)
436             mpRefSections[i].CopyFrom(src->mpRefSections[i])
437     }
438     else
439     {
440         delete[] mpRefSections
441         mpRefSections = NULL
442     }
443     if (src->mpDegSections != NULL && mNumSections > 0)
444     {
445         delete[] mpDegSections
446         mpDegSections = new SECTION[mNumSections]
447         for (int i = 0 i < mNumSections i++)
448             mpDegSections[i].CopyFrom(src->mpDegSections[i])
449     }
450     else
451     {
452         delete[] mpDegSections
453         mpDegSections = NULL
454     }
455
456     mSNRRefdB = src->mSNRRefdB
457     mSNRDegdB = src->mSNRDegdB
458     mNoiseLevelRef = src->mNoiseLevelRef
459     mNoiseLevelDeg = src->mNoiseLevelDeg
460     mSignalLevelRef = src->mSignalLevelRef
461     mSignalLevelDeg = src->mSignalLevelDeg
462     mNoiseThresholdRef = src->mNoiseThresholdRef
463     mNoiseThresholdDeg = src->mNoiseThresholdDeg
464
465     if (src->mpActiveFrameFlags != NULL && mNumFrames > 0)
466     {
467         matFree(mpActiveFrameFlags)

```

```

468     mpActiveFrameFlags = (int*)matMalloc(mNumFrames * sizeof(int))
469     for (int i = 0 i < mNumFrames i++)
470         mpActiveFrameFlags[i] = src->mpActiveFrameFlags[i]
471 }
472 else
473 {
474     matFree(mpActiveFrameFlags)
475     mpActiveFrameFlags = NULL
476 }
477
478 if (src->mpIgnoreFlags != NULL && mNumFrames > 0)
479 {
480
481     matFree(mpIgnoreFlags)
482     mpIgnoreFlags = (int*)matMalloc(mNumFrames * sizeof(int))
483     mNumIngoreFlags = src->mNumIngoreFlags
484     for (int i = 0 i < mNumFrames i++)
485         mpIgnoreFlags[i] = src->mpIgnoreFlags[i]
486 }
487 else
488 {
489     matFree(mpIgnoreFlags)
490     mpIgnoreFlags = NULL
491 }
492
493 for (int i = 0 i < 5 i++)
494     mTimeDiffs[i] = src->mTimeDiffs[i]
495
496 mAslFrames = src->mAslFrames
497 mAslFramelength = src->mAslFramelength
498 if (src->mpAslActiveFrameFlags != NULL && mAslFrames > 0)
499 {
500     matFree(mpAslActiveFrameFlags)
501     mpAslActiveFrameFlags = (int*)matMalloc(mAslFrames * sizeof(int))
502     for (int i = 0 i < mAslFrames i++)
503         mpAslActiveFrameFlags[i] = src->mpAslActiveFrameFlags[i]
504 }
505 else
506 {
507     matFree(mpAslActiveFrameFlags)
508     mpAslActiveFrameFlags = NULL
509 }
510
511 mAslFramesDeg = src->mAslFramesDeg
512 if (src->mpAslActiveFrameFlagsDeg != NULL && mAslFramesDeg > 0)
513 {
514     matFree(mpAslActiveFrameFlagsDeg)
515     mpAslActiveFrameFlagsDeg = (int*)matMalloc(mAslFramesDeg * sizeof(int))
516     for (int i = 0 i < mAslFramesDeg i++)
517         mpAslActiveFrameFlagsDeg[i] = src->mpAslActiveFrameFlagsDeg[i]
518 }
519 else
520 {
521     matFree(mpAslActiveFrameFlagsDeg)
522     mpAslActiveFrameFlagsDeg = NULL
523 }
524
525 FirstRefSample = src->FirstRefSample
526 FirstDegSample = src->FirstDegSample
527 }
528
529 OTA_RESULT()
530 {
531     mNumFrames = 0
532     mpDelay = NULL
533
534     mpReliability = NULL
535

```



```

536     mNumUtterances = 0
537     mpStartSampleUtterance = NULL
538     mpStopSampleUtterance = NULL
539     mpDelayUtterance      = NULL
540
541     mNumSections = 0
542     mpRefSections = NULL
543     mpDegSections = NULL
544
545     mpActiveFrameFlags = NULL
546     mpIgnoreFlags = NULL
547     mNumIngoreFlags = 0
548
549     mAslFramelength = 0
550     mAslFrames = 0
551     mpAslActiveFrameFlags = NULL
552     mAslFramesDeg = 0
553     mpAslActiveFrameFlagsDeg = NULL
554
555     FirstRefSample = FirstDegSample = 0
556 }
557
558 ~OTA_RESULT()
559 {
560     matFree(mpDelay)
561     mpDelay = NULL
562
563     matFree(mpReliability)
564     mpReliability = NULL
565
566     matFree(mpStartSampleUtterance)
567     mpStartSampleUtterance = NULL
568
569     matFree(mpStopSampleUtterance)
570     mpStopSampleUtterance = NULL
571
572     matFree(mpDelayUtterance)
573     mpDelayUtterance      = NULL
574
575     delete[] mpRefSections
576     mpRefSections = NULL
577     delete[] mpDegSections
578     mpDegSections = NULL
579
580     matFree(mpActiveFrameFlags)
581     mpActiveFrameFlags = NULL
582
583     matFree(mpIgnoreFlags)
584     mpIgnoreFlags = NULL
585
586     matFree(mpAslActiveFrameFlags)
587     mpAslActiveFrameFlags = NULL
588     matFree(mpAslActiveFrameFlagsDeg)
589     mpAslActiveFrameFlagsDeg = NULL
590 }
591
592 long mNumFrames
593 int mStepsize
594 int mResolutionInSamples
595 int mPitchFrameSize
596 long *mpDelay
597 OTA_FLOAT *mpReliability
598 OTA_FLOAT mAvgReliability
599 OTA_FLOAT mRelSamplerateDev
600
601 int mNumUtterances
602 int* mpStartSampleUtterance
603 int* mpStopSampleUtterance

```

```

604     int* mpDelayUtterance
605     int FirstRefSample
606     int FirstDegSample
607
608     int          mNumSections
609     SECTION      *mpRefSections
610     SECTION      *mpDegSections
611
612     double mSNRRefdB, mSNRDegdB
613     double mNoiseLevelRef, mNoiseLevelDeg
614     double mSignalLevelRef, mSignalLevelDeg
615     double mNoiseThresholdRef, mNoiseThresholdDeg
616
617     int *mpActiveFrameFlags
618
619     int *mpIgnoreFlags
620     int mNumIgnoreFlags
621     int mAslFrames
622     int mAslFrameLength
623     int *mpAslActiveFrameFlags
624     int mAslFramesDeg
625     int *mpAslActiveFrameFlagsDeg
626
627     double mTimeDiffs[5]
628
629 }OTA_RESULT
630
631 struct FilteringParameters
632 {
633     int pListeningCondition
634     double cutOffFrequencyLow
635     double cutOffFrequencyHigh
636     double disturbedEnergyQuotient
637 }
638
639 class ITempAlignment
640 {
641     public:
642
643     virtual bool Init(CProcessData* pProcessData)=0
644     virtual void Free()=0
645     virtual void Destroy()=0
646
647     virtual bool SetSignal(int Index, unsigned long SampleRate, unsigned long NumSamples, int
NumChannels, OTA_FLOAT** pSignal)=0
648
649     virtual void GetFilterCharacteristics(FilteringParameters *FilterParams)=0
650
651     virtual bool FilterSignal(int Index, FilteringParameters *FilterParams)=0
652
653     virtual bool Run(unsigned long Control, OTA_RESULT* pResult, int TArunIndex)=0
654
655     virtual void GetNoiseSwitching(OTA_FLOAT* pBGNSwitchingLevel, OTA_FLOAT*
pNoiseLevelSpeechDeg, OTA_FLOAT* pNoiseLevelSilenceDeg)=0
656
657     virtual OTA_FLOAT GetPitchFreq(int Signal, int Channel)=0
658
659     virtual OTA_FLOAT GetPitchVector(int Signal, int Channel, OTA_FLOAT* pVector, int NumFrames,
int SamplesPerFrame)=0
660     virtual int GetPitchFrameSize()=0
661 }
662
663 enum AlignmentType
664 {
665     TA_FOR_SPEECH=0,
666
667 }
668

```

```

669 ITempAlignment* CreateAlignment(AlignmentType Type)
670 }
671 }
672
673 {
674 {
675
676 struct FILTER_COEFFICIENTS
677 {
678     long SampleRate
679     int NumTaps
680     OTA_FLOAT LowCutOff
681     OTA_FLOAT HighCutOff
682     OTA_FLOAT HPCoef[128]
683     OTA_FLOAT LPCoef[128]
684 }
685
686 FILTER_COEFFICIENTS FilterCoefficients[2][3] =
687 {
688     {
689         {
690             8000, 128, (OTA_FLOAT)320, (OTA_FLOAT)3400,
691             {(OTA_FLOAT)-0, (OTA_FLOAT)-3.11583033328733e-006, (OTA_FLOAT)-1.22605465123787e-005,
(OTA_FLOAT)-2.53447644123974e-005, (OTA_FLOAT)-3.80571666477877e-005,
(OTA_FLOAT)-4.43746809393845e-005, (OTA_FLOAT)-3.74083070970163e-005,
(OTA_FLOAT)-1.05012053091289e-005, (OTA_FLOAT)4.15411849101502e-005,
(OTA_FLOAT)0.000121235067397927, (OTA_FLOAT)0.000227373834396267,
(OTA_FLOAT)0.000354307928141495, (OTA_FLOAT)0.000491679894182871,
(OTA_FLOAT)0.000624707859347635, (OTA_FLOAT)0.00073505992972446,
(OTA_FLOAT)0.000802302338862371, (OTA_FLOAT)0.000805840870148991,
(OTA_FLOAT)0.00072721409741341, (OTA_FLOAT)0.000552544461865933,
(OTA_FLOAT)0.000274914860306623, (OTA_FLOAT)-0.00010358096031938,
(OTA_FLOAT)-0.000570362981001483, (OTA_FLOAT)-0.00110168648881982,
(OTA_FLOAT)-0.00166293264913993, (OTA_FLOAT)-0.00220996942735605,
(OTA_FLOAT)-0.00269159766809306, (OTA_FLOAT)-0.00305300110112502,
(OTA_FLOAT)-0.00324002119316419, (OTA_FLOAT)-0.0032039864519629,
(OTA_FLOAT)-0.00290674935333309, (OTA_FLOAT)-0.00232553012197935,
(OTA_FLOAT)-0.00145714136818128, (OTA_FLOAT)-0.00032117527538677,
(OTA_FLOAT)0.00103822260437251, (OTA_FLOAT)0.0025522928547536,
(OTA_FLOAT)0.00412950899978116, (OTA_FLOAT)0.00565939628109387,
(OTA_FLOAT)0.0070182430446411, (OTA_FLOAT)0.00807647703382874,
(OTA_FLOAT)0.00870733759617603, (OTA_FLOAT)0.00879635204278891,
(OTA_FLOAT)0.00825103087676689, (OTA_FLOAT)0.00701014147489744,
(OTA_FLOAT)0.00505190993213903, (OTA_FLOAT)0.00240054023728007,
(OTA_FLOAT)-0.000869470199786394, (OTA_FLOAT)-0.00463160122369293,
(OTA_FLOAT)-0.00870836644831797, (OTA_FLOAT)-0.0128751328822397,
(OTA_FLOAT)-0.0168664228081104, (OTA_FLOAT)-0.0203841213279029,
(OTA_FLOAT)-0.0231066645508917, (OTA_FLOAT)-0.0246978384501282,
(OTA_FLOAT)-0.0248131711568158, (OTA_FLOAT)-0.0231008068557507,
(OTA_FLOAT)-0.0191916293308248, (OTA_FLOAT)-0.0126688180175239,
(OTA_FLOAT)-0.00299608465142941, (OTA_FLOAT)0.0106451005788457,
(OTA_FLOAT)0.0297460629169149, (OTA_FLOAT)0.0575374913705589,
(OTA_FLOAT)0.102613789546255, (OTA_FLOAT)0.197033159228519,
(OTA_FLOAT)0.631503179844918, (OTA_FLOAT)-0.631503179844918,
(OTA_FLOAT)-0.197033159228519, (OTA_FLOAT)-0.102613789546255,
(OTA_FLOAT)-0.0575374913705589, (OTA_FLOAT)-0.0297460629169149,
(OTA_FLOAT)-0.0106451005788457, (OTA_FLOAT)0.00299608465142941,
(OTA_FLOAT)0.0126688180175239, (OTA_FLOAT)0.0191916293308248,
(OTA_FLOAT)0.0231008068557507, (OTA_FLOAT)0.0248131711568158,
(OTA_FLOAT)0.0246978384501282, (OTA_FLOAT)0.0231066645508917,
(OTA_FLOAT)0.0203841213279029, (OTA_FLOAT)0.0168664228081104,
(OTA_FLOAT)0.0128751328822397, (OTA_FLOAT)0.00870836644831797,
(OTA_FLOAT)0.00463160122369293, (OTA_FLOAT)0.000869470199786394,
(OTA_FLOAT)-0.00240054023728007, (OTA_FLOAT)-0.00505190993213903,
(OTA_FLOAT)-0.00701014147489744, (OTA_FLOAT)-0.00825103087676689,
(OTA_FLOAT)-0.00879635204278891, (OTA_FLOAT)-0.00870733759617603,
(OTA_FLOAT)-0.00807647703382874, (OTA_FLOAT)-0.0070182430446411,
(OTA_FLOAT)-0.00565939628109387, (OTA_FLOAT)-0.00412950899978116,

```

(OTA\_FLOAT)-0.0025522928547536, (OTA\_FLOAT)-0.00103822260437251,  
 (OTA\_FLOAT)0.00032117527538677, (OTA\_FLOAT)0.00145714136818128,  
 (OTA\_FLOAT)0.00232553012197935, (OTA\_FLOAT)0.00290674935333309,  
 (OTA\_FLOAT)0.0032039864519629, (OTA\_FLOAT)0.00324002119316419,  
 (OTA\_FLOAT)0.00305300110112502, (OTA\_FLOAT)0.00269159766809306,  
 (OTA\_FLOAT)0.00220996942735605, (OTA\_FLOAT)0.00166293264913993,  
 (OTA\_FLOAT)0.00110168648881982, (OTA\_FLOAT)0.000570362981001483,  
 (OTA\_FLOAT)0.00010358096031938, (OTA\_FLOAT)-0.000274914860306623,  
 (OTA\_FLOAT)-0.000552544461865933, (OTA\_FLOAT)-0.00072721409741341,  
 (OTA\_FLOAT)-0.000805840870148991, (OTA\_FLOAT)-0.000802302338862371,  
 (OTA\_FLOAT)-0.00073505992972446, (OTA\_FLOAT)-0.000624707859347635,  
 (OTA\_FLOAT)-0.000491679894182871, (OTA\_FLOAT)-0.000354307928141495,  
 (OTA\_FLOAT)-0.000227373834396267, (OTA\_FLOAT)-0.000121235067397927,  
 (OTA\_FLOAT)-4.15411849101502e-005, (OTA\_FLOAT)1.05012053091289e-005,  
 (OTA\_FLOAT)3.74083070970163e-005, (OTA\_FLOAT)4.43746809393845e-005,  
 (OTA\_FLOAT)3.80571666477877e-005, (OTA\_FLOAT)2.53447644123974e-005,  
 (OTA\_FLOAT)1.22605465123787e-005, (OTA\_FLOAT)3.11583033328733e-006, (OTA\_FLOAT)0},  
 692 { (OTA\_FLOAT)-0, (OTA\_FLOAT)-1.19237625835112e-006, (OTA\_FLOAT)9.62538896808589e-006,  
 (OTA\_FLOAT)-2.81231206959131e-005, (OTA\_FLOAT)5.07643685368071e-005,  
 (OTA\_FLOAT)-6.29732619511813e-005, (OTA\_FLOAT)4.6326043415256e-005,  
 (OTA\_FLOAT)1.31216433707137e-005, (OTA\_FLOAT)-0.000115834401485274,  
 (OTA\_FLOAT)0.000242778021807158, (OTA\_FLOAT)-0.000355607643552515,  
 (OTA\_FLOAT)0.000404611693827812, (OTA\_FLOAT)-0.000343437856894063,  
 (OTA\_FLOAT)0.000146994199208401, (OTA\_FLOAT)0.000172960119075268,  
 (OTA\_FLOAT)-0.00056040728754611, (OTA\_FLOAT)0.000920252169170894,  
 (OTA\_FLOAT)-0.00113734674979644, (OTA\_FLOAT)0.00110649215851059,  
 (OTA\_FLOAT)-0.000766578959456769, (OTA\_FLOAT)0.00012942823050283,  
 (OTA\_FLOAT)0.00070633135447172, (OTA\_FLOAT)-0.00156343190260447,  
 (OTA\_FLOAT)0.00221818210047269, (OTA\_FLOAT)-0.00245223178754076,  
 (OTA\_FLOAT)0.00211309295836296, (OTA\_FLOAT)-0.00116833255996348,  
 (OTA\_FLOAT)-0.000262454542244551, (OTA\_FLOAT)0.0019103788839396,  
 (OTA\_FLOAT)-0.0033998862860742, (OTA\_FLOAT)0.00432669764846672,  
 (OTA\_FLOAT)-0.00435646800075709, (OTA\_FLOAT)0.0033219439720518,  
 (OTA\_FLOAT)-0.00129344900401471, (OTA\_FLOAT)-0.00139936523907736,  
 (OTA\_FLOAT)0.00420740469503434, (OTA\_FLOAT)-0.00646290340537264,  
 (OTA\_FLOAT)0.00752504066464281, (OTA\_FLOAT)-0.00694105878493074,  
 (OTA\_FLOAT)0.00458572976395982, (OTA\_FLOAT)-0.000742279045265008,  
 (OTA\_FLOAT)-0.00390292393148497, (OTA\_FLOAT)0.00836264675823261,  
 (OTA\_FLOAT)-0.0115372535873582, (OTA\_FLOAT)0.0124570177369915,  
 (OTA\_FLOAT)-0.0105294575758111, (OTA\_FLOAT)0.00573572492373141,  
 (OTA\_FLOAT)0.00127513282412512, (OTA\_FLOAT)-0.00922843042613262,  
 (OTA\_FLOAT)0.0164109040361915, (OTA\_FLOAT)-0.0209804156042358,  
 (OTA\_FLOAT)0.0213477674919676, (OTA\_FLOAT)-0.01656022776385,  
 (OTA\_FLOAT)0.00661014918052088, (OTA\_FLOAT)0.00739781425834486,  
 (OTA\_FLOAT)-0.0232611774392236, (OTA\_FLOAT)0.0378764212628487,  
 (OTA\_FLOAT)-0.0475684509725642, (OTA\_FLOAT)0.0484383267975965,  
 (OTA\_FLOAT)-0.0365031142357209, (OTA\_FLOAT)0.00708215904333225,  
 (OTA\_FLOAT)0.0485386397047851, (OTA\_FLOAT)-0.161141030330073,  
 (OTA\_FLOAT)0.618934983985472, (OTA\_FLOAT)0.618934983985472,  
 (OTA\_FLOAT)-0.161141030330073, (OTA\_FLOAT)0.0485386397047851,  
 (OTA\_FLOAT)0.00708215904333225, (OTA\_FLOAT)-0.0365031142357209,  
 (OTA\_FLOAT)0.0484383267975965, (OTA\_FLOAT)-0.0475684509725642,  
 (OTA\_FLOAT)0.0378764212628487, (OTA\_FLOAT)-0.0232611774392236,  
 (OTA\_FLOAT)0.00739781425834486, (OTA\_FLOAT)0.00661014918052088,  
 (OTA\_FLOAT)-0.01656022776385, (OTA\_FLOAT)0.0213477674919676,  
 (OTA\_FLOAT)-0.0209804156042358, (OTA\_FLOAT)0.0164109040361915,  
 (OTA\_FLOAT)-0.00922843042613262, (OTA\_FLOAT)0.00127513282412512,  
 (OTA\_FLOAT)0.00573572492373141, (OTA\_FLOAT)-0.0105294575758111,  
 (OTA\_FLOAT)0.0124570177369915, (OTA\_FLOAT)-0.0115372535873582,  
 (OTA\_FLOAT)0.00836264675823261, (OTA\_FLOAT)-0.00390292393148497,  
 (OTA\_FLOAT)-0.000742279045265008, (OTA\_FLOAT)0.00458572976395982,  
 (OTA\_FLOAT)-0.00694105878493074, (OTA\_FLOAT)0.00752504066464281,  
 (OTA\_FLOAT)-0.00646290340537264, (OTA\_FLOAT)0.00420740469503434,  
 (OTA\_FLOAT)-0.00139936523907736, (OTA\_FLOAT)-0.00129344900401471,  
 (OTA\_FLOAT)0.0033219439720518, (OTA\_FLOAT)-0.00435646800075709,  
 (OTA\_FLOAT)0.00432669764846672, (OTA\_FLOAT)-0.0033998862860742,  
 (OTA\_FLOAT)0.0019103788839396, (OTA\_FLOAT)-0.000262454542244551,  
 (OTA\_FLOAT)-0.00116833255996348, (OTA\_FLOAT)0.00211309295836296,

```

(OTA_FLOAT)-0.00245223178754076, (OTA_FLOAT)0.00221818210047269,
(OTA_FLOAT)-0.00156343190260447, (OTA_FLOAT)0.00070633135447172,
(OTA_FLOAT)0.00012942823050283, (OTA_FLOAT)-0.000766578959456769,
(OTA_FLOAT)0.00110649215851059, (OTA_FLOAT)-0.00113734674979644,
(OTA_FLOAT)0.000920252169170894, (OTA_FLOAT)-0.00056040728754611,
(OTA_FLOAT)0.000172960119075268, (OTA_FLOAT)0.000146994199208401,
(OTA_FLOAT)-0.000343437856894063, (OTA_FLOAT)0.000404611693827812,
(OTA_FLOAT)-0.000355607643552515, (OTA_FLOAT)0.000242778021807158,
(OTA_FLOAT)-0.000115834401485274, (OTA_FLOAT)1.31216433707137e-005,
(OTA_FLOAT)4.6326043415256e-005, (OTA_FLOAT)-6.29732619511813e-005,
(OTA_FLOAT)5.07643685368071e-005, (OTA_FLOAT)-2.81231206959131e-005,
(OTA_FLOAT)9.62538896808589e-006, (OTA_FLOAT)-1.19237625835112e-006, (OTA_FLOAT)-0}
693   },
694   {
695       16000, 128, (OTA_FLOAT)320, (OTA_FLOAT)3400,
696       {(OTA_FLOAT)-0, (OTA_FLOAT)6.65908919899535e-020, (OTA_FLOAT)1.586496896698e-006,
(OTA_FLOAT)7.19267321420086e-006, (OTA_FLOAT)1.921863178562e-005,
(OTA_FLOAT)3.98966029957852e-005, (OTA_FLOAT)7.11548395539486e-005,
(OTA_FLOAT)0.000114484971269058, (OTA_FLOAT)0.000170817662138448,
(OTA_FLOAT)0.000240411337391996, (OTA_FLOAT)0.000322758527731017,
(OTA_FLOAT)0.000416514014824822, (OTA_FLOAT)0.000519448463315367,
(OTA_FLOAT)0.000628430601322213, (OTA_FLOAT)0.00073944027873603,
(OTA_FLOAT)0.000847613908901011, (OTA_FLOAT)0.000947322906084213,
(OTA_FLOAT)0.00103228479235357, (OTA_FLOAT)0.00109570568893002,
(OTA_FLOAT)0.00113045195572206, (OTA_FLOAT)0.00112924782604515,
(OTA_FLOAT)0.00108489502867423, (OTA_FLOAT)0.000990509622599959,
(OTA_FLOAT)0.000839770615712969, (OTA_FLOAT)0.000627174419367237,
(OTA_FLOAT)0.000348288825728668, (OTA_FLOAT)-3.40646020854252e-017,
(OTA_FLOAT)-0.000419254032681219, (OTA_FLOAT)-0.000909269747262774,
(OTA_FLOAT)-0.00146789029335279, (OTA_FLOAT)-0.00209084888200293,
(OTA_FLOAT)-0.00277164801902062, (OTA_FLOAT)-0.00350147826773971,
(OTA_FLOAT)-0.00426917909158791, (OTA_FLOAT)-0.00506124301983794,
(OTA_FLOAT)-0.00586186290326828, (OTA_FLOAT)-0.0066530203794415,
(OTA_FLOAT)-0.00741461183962234, (OTA_FLOAT)-0.00812460615468203,
(OTA_FLOAT)-0.00875922612402317, (OTA_FLOAT)-0.00929314296855955,
(OTA_FLOAT)-0.00969967004394336, (OTA_FLOAT)-0.00995093805596276,
(OTA_FLOAT)-0.0100180290178168, (OTA_FLOAT)-0.00987103935740738,
(OTA_FLOAT)-0.00947903292161399, (OTA_FLOAT)-0.00880983042336143,
(OTA_FLOAT)-0.00782956027120375, (OTA_FLOAT)-0.00650186180029422,
(OTA_FLOAT)-0.00478657704515611, (OTA_FLOAT)-0.00263767566934936,
(OTA_FLOAT)9.36354757781549e-017, (OTA_FLOAT)0.00319586439452034,
(OTA_FLOAT)0.00704181063335111, (OTA_FLOAT)0.0116657633769799,
(OTA_FLOAT)0.0172549030222906, (OTA_FLOAT)0.024097527616987,
(OTA_FLOAT)0.0326635519582126, (OTA_FLOAT)0.043772732964646,
(OTA_FLOAT)0.0589869822444135, (OTA_FLOAT)0.0816748156330617,
(OTA_FLOAT)0.120629763168131, (OTA_FLOAT)0.208160986435128,
(OTA_FLOAT)0.635266416307386, (OTA_FLOAT)-0.635266416307386,
(OTA_FLOAT)-0.208160986435128, (OTA_FLOAT)-0.120629763168131,
(OTA_FLOAT)-0.0816748156330617, (OTA_FLOAT)-0.0589869822444135,
(OTA_FLOAT)-0.043772732964646, (OTA_FLOAT)-0.0326635519582126,
(OTA_FLOAT)-0.024097527616987, (OTA_FLOAT)-0.0172549030222906,
(OTA_FLOAT)-0.0116657633769799, (OTA_FLOAT)-0.00704181063335111,
(OTA_FLOAT)-0.00319586439452034, (OTA_FLOAT)-9.36354757781549e-017,
(OTA_FLOAT)0.00263767566934936, (OTA_FLOAT)0.00478657704515611,
(OTA_FLOAT)0.00650186180029422, (OTA_FLOAT)0.00782956027120375,
(OTA_FLOAT)0.00880983042336143, (OTA_FLOAT)0.00947903292161399,
(OTA_FLOAT)0.00987103935740738, (OTA_FLOAT)0.0100180290178168,
(OTA_FLOAT)0.00995093805596276, (OTA_FLOAT)0.00969967004394336,
(OTA_FLOAT)0.00929314296855955, (OTA_FLOAT)0.00875922612402317,
(OTA_FLOAT)0.00812460615468203, (OTA_FLOAT)0.00741461183962234,
(OTA_FLOAT)0.0066530203794415, (OTA_FLOAT)0.00586186290326828,
(OTA_FLOAT)0.00506124301983794, (OTA_FLOAT)0.00426917909158791,
(OTA_FLOAT)0.00350147826773971, (OTA_FLOAT)0.00277164801902062,
(OTA_FLOAT)0.00209084888200293, (OTA_FLOAT)0.00146789029335279,
(OTA_FLOAT)0.000909269747262774, (OTA_FLOAT)0.000419254032681219,
(OTA_FLOAT)3.40646020854252e-017, (OTA_FLOAT)-0.000348288825728668,
(OTA_FLOAT)-0.000627174419367237, (OTA_FLOAT)-0.000839770615712969,
(OTA_FLOAT)-0.000990509622599959, (OTA_FLOAT)-0.00108489502867423,

```



697

(OTA\_FLOAT)-0.00112924782604515, (OTA\_FLOAT)-0.00113045195572206,  
 (OTA\_FLOAT)-0.00109570568893002, (OTA\_FLOAT)-0.00103228479235357,  
 (OTA\_FLOAT)-0.000947322906084213, (OTA\_FLOAT)-0.000847613908901011,  
 (OTA\_FLOAT)-0.00073944027873603, (OTA\_FLOAT)-0.000628430601322213,  
 (OTA\_FLOAT)-0.000519448463315367, (OTA\_FLOAT)-0.000416514014824822,  
 (OTA\_FLOAT)-0.000322758527731017, (OTA\_FLOAT)-0.000240411337391996,  
 (OTA\_FLOAT)-0.000170817662138448, (OTA\_FLOAT)-0.000114484971269058,  
 (OTA\_FLOAT)-7.11548395539486e-005, (OTA\_FLOAT)-3.98966029957852e-005,  
 (OTA\_FLOAT)-1.921863178562e-005, (OTA\_FLOAT)-7.19267321420086e-006,  
 (OTA\_FLOAT)-1.586496896698e-006, (OTA\_FLOAT)-6.65908919899535e-020, (OTA\_FLOAT)0},  
 {(OTA\_FLOAT)0, (OTA\_FLOAT)3.05597618803318e-006, (OTA\_FLOAT)5.2995175799649e-006,  
 (OTA\_FLOAT)-2.27132521707932e-005, (OTA\_FLOAT)-4.09991450214529e-005,  
 (OTA\_FLOAT)3.46716283242721e-005, (OTA\_FLOAT)0.000118730379417801,  
 (OTA\_FLOAT)6.56591952999708e-006, (OTA\_FLOAT)-0.000213370985935312,  
 (OTA\_FLOAT)-0.000139129201199306, (OTA\_FLOAT)0.000261939592782832,  
 (OTA\_FLOAT)0.000364142888735702, (OTA\_FLOAT)-0.000183032882309492,  
 (OTA\_FLOAT)-0.000625311618065512, (OTA\_FLOAT)-8.70841584754242e-005,  
 (OTA\_FLOAT)0.000809567645527397, (OTA\_FLOAT)0.000553391952773557,  
 (OTA\_FLOAT)-0.000774423318157872, (OTA\_FLOAT)-0.00113226574623215,  
 (OTA\_FLOAT)0.000398243527869886, (OTA\_FLOAT)0.00164836402599976,  
 (OTA\_FLOAT)0.000360086564471873, (OTA\_FLOAT)-0.00186719701423751,  
 (OTA\_FLOAT)-0.00141229235170514, (OTA\_FLOAT)0.00156130923489738,  
 (OTA\_FLOAT)0.00252365380039181, (OTA\_FLOAT)-0.000595614020267493,  
 (OTA\_FLOAT)-0.00334255227175313, (OTA\_FLOAT)-0.000992456180701031,  
 (OTA\_FLOAT)0.00347908003974297, (OTA\_FLOAT)0.00294606332694152,  
 (OTA\_FLOAT)-0.00261975403579494, (OTA\_FLOAT)-0.00479889969276422,  
 (OTA\_FLOAT)0.000651242139793398, (OTA\_FLOAT)0.00595288349216803,  
 (OTA\_FLOAT)0.00224230786710318, (OTA\_FLOAT)-0.00581649109887999,  
 (OTA\_FLOAT)-0.0055429238462915, (OTA\_FLOAT)0.00397772400086495,  
 (OTA\_FLOAT)0.00844707330829848, (OTA\_FLOAT)-0.000371427902917342,  
 (OTA\_FLOAT)-0.0100029185542627, (OTA\_FLOAT)-0.00460428078877974,  
 (OTA\_FLOAT)0.00931790440049332, (OTA\_FLOAT)0.01006073928337,  
 (OTA\_FLOAT)-0.00579727694283533, (OTA\_FLOAT)-0.0147002581318333,  
 (OTA\_FLOAT)-0.000638061809540575, (OTA\_FLOAT)0.0169990889874772,  
 (OTA\_FLOAT)0.00940462383093061, (OTA\_FLOAT)-0.0154541152852821,  
 (OTA\_FLOAT)-0.019212587874169, (OTA\_FLOAT)0.00882566134884216,  
 (OTA\_FLOAT)0.0281194979256692, (OTA\_FLOAT)0.00372474552336024,  
 (OTA\_FLOAT)-0.0336032329881464, (OTA\_FLOAT)-0.0227769164028228,  
 (OTA\_FLOAT)0.0323895220594741, (OTA\_FLOAT)0.0495666036273973,  
 (OTA\_FLOAT)-0.0189636420516584, (OTA\_FLOAT)-0.0901965216404877,  
 (OTA\_FLOAT)-0.024744918804442, (OTA\_FLOAT)0.192449732030053,  
 (OTA\_FLOAT)0.39406915415067, (OTA\_FLOAT)0.39406915415067, (OTA\_FLOAT)0.192449732030053,  
 (OTA\_FLOAT)-0.024744918804442, (OTA\_FLOAT)-0.0901965216404877,  
 (OTA\_FLOAT)-0.0189636420516584, (OTA\_FLOAT)0.0495666036273973,  
 (OTA\_FLOAT)0.0323895220594741, (OTA\_FLOAT)-0.0227769164028228,  
 (OTA\_FLOAT)-0.0336032329881464, (OTA\_FLOAT)0.00372474552336024,  
 (OTA\_FLOAT)0.0281194979256692, (OTA\_FLOAT)0.00882566134884216,  
 (OTA\_FLOAT)-0.019212587874169, (OTA\_FLOAT)-0.0154541152852821,  
 (OTA\_FLOAT)0.00940462383093061, (OTA\_FLOAT)0.0169990889874772,  
 (OTA\_FLOAT)-0.000638061809540575, (OTA\_FLOAT)-0.0147002581318333,  
 (OTA\_FLOAT)-0.00579727694283533, (OTA\_FLOAT)0.01006073928337,  
 (OTA\_FLOAT)0.00931790440049332, (OTA\_FLOAT)-0.00460428078877974,  
 (OTA\_FLOAT)-0.0100029185542627, (OTA\_FLOAT)-0.000371427902917342,  
 (OTA\_FLOAT)0.00844707330829848, (OTA\_FLOAT)0.00397772400086495,  
 (OTA\_FLOAT)-0.0055429238462915, (OTA\_FLOAT)-0.00581649109887999,  
 (OTA\_FLOAT)0.00224230786710318, (OTA\_FLOAT)0.00595288349216803,  
 (OTA\_FLOAT)0.000651242139793398, (OTA\_FLOAT)-0.00479889969276422,  
 (OTA\_FLOAT)-0.00261975403579494, (OTA\_FLOAT)0.00294606332694152,  
 (OTA\_FLOAT)0.00347908003974297, (OTA\_FLOAT)-0.000992456180701031,  
 (OTA\_FLOAT)-0.00334255227175313, (OTA\_FLOAT)-0.000595614020267493,  
 (OTA\_FLOAT)0.00252365380039181, (OTA\_FLOAT)0.00156130923489738,  
 (OTA\_FLOAT)-0.00141229235170514, (OTA\_FLOAT)-0.00186719701423751,  
 (OTA\_FLOAT)0.000360086564471873, (OTA\_FLOAT)0.00164836402599976,  
 (OTA\_FLOAT)0.000398243527869886, (OTA\_FLOAT)-0.00113226574623215,  
 (OTA\_FLOAT)-0.000774423318157872, (OTA\_FLOAT)0.000553391952773557,  
 (OTA\_FLOAT)0.000809567645527397, (OTA\_FLOAT)-8.70841584754242e-005,  
 (OTA\_FLOAT)-0.000625311618065512, (OTA\_FLOAT)-0.000183032882309492,  
 (OTA\_FLOAT)0.000364142888735702, (OTA\_FLOAT)0.000261939592782832,

```

(OTA_FLOAT)-0.000139129201199306, (OTA_FLOAT)-0.000213370985935312,
(OTA_FLOAT)6.56591952999708e-006, (OTA_FLOAT)0.000118730379417801,
(OTA_FLOAT)3.46716283242721e-005, (OTA_FLOAT)-4.09991450214529e-005,
(OTA_FLOAT)-2.27132521707932e-005, (OTA_FLOAT)5.2995175799649e-006,
(OTA_FLOAT)3.05597618803318e-006, (OTA_FLOAT)0}
698     },
699     {
700         48000, 128, (OTA_FLOAT)320, (OTA_FLOAT)3400,
701         {(OTA_FLOAT)-0, (OTA_FLOAT)-2.6983883994397e-006, (OTA_FLOAT)-1.0687696107602e-005,
(OTA_FLOAT)-2.37494875840618e-005, (OTA_FLOAT)-4.15843325453698e-005,
(OTA_FLOAT)-6.38103242876749e-005, (OTA_FLOAT)-8.99620156103482e-005,
(OTA_FLOAT)-0.000119489770824989, (OTA_FLOAT)-0.000151759527541704,
(OTA_FLOAT)-0.000186052956842886, (OTA_FLOAT)-0.000221568005064868,
(OTA_FLOAT)-0.000257419794683033, (OTA_FLOAT)-0.000292641855697718,
(OTA_FLOAT)-0.000326187652395574, (OTA_FLOAT)-0.000356932363344371,
(OTA_FLOAT)-0.000383674864879293, (OTA_FLOAT)-0.000405139860037364,
(OTA_FLOAT)-0.000419980085743287, (OTA_FLOAT)-0.00042677852084975,
(OTA_FLOAT)-0.000424050506139799, (OTA_FLOAT)-0.00041024567428938,
(OTA_FLOAT)-0.000383749572657814, (OTA_FLOAT)-0.00034288484410067,
(OTA_FLOAT)-0.000285911810113899, (OTA_FLOAT)-0.000211028275657584,
(OTA_FLOAT)-0.000116368344857943, (OTA_FLOAT)2.11844521182629e-017,
(OTA_FLOAT)0.000140078849087584, (OTA_FLOAT)0.000305946194467046,
(OTA_FLOAT)0.000499764058146784, (OTA_FLOAT)0.000723789427771377,
(OTA_FLOAT)0.000980388622627241, (OTA_FLOAT)0.00127205585862345,
(OTA_FLOAT)0.00160143696990024, (OTA_FLOAT)0.00197135949141317,
(OTA_FLOAT)0.00238487063159877, (OTA_FLOAT)0.0028452850955481,
(OTA_FLOAT)0.00335624529732888, (OTA_FLOAT)0.00392179728292591,
(OTA_FLOAT)0.00454648675635105, (OTA_FLOAT)0.00523548108358237,
(OTA_FLOAT)0.00599472522492496, (OTA_FLOAT)0.00683114249244957,
(OTA_FLOAT)0.0077528952699248, (OTA_FLOAT)0.00876972703215903,
(OTA_FLOAT)0.00989341621778745, (OTA_FLOAT)0.0111383864687109,
(OTA_FLOAT)0.0125225393246692, (OTA_FLOAT)0.0140684095664809,
(OTA_FLOAT)0.015804798677585, (OTA_FLOAT)0.0177691340229538,
(OTA_FLOAT)0.0200109598098972, (OTA_FLOAT)0.0225972483802055,
(OTA_FLOAT)0.0256207450012417, (OTA_FLOAT)0.0292135809519727,
(OTA_FLOAT)0.0335704930563346, (OTA_FLOAT)0.0389906152045799,
(OTA_FLOAT)0.0459578419490588, (OTA_FLOAT)0.0553088553412474,
(OTA_FLOAT)0.0686251794236621, (OTA_FLOAT)0.0892973169642373,
(OTA_FLOAT)0.126142797829284, (OTA_FLOAT)0.21149639206314, (OTA_FLOAT)0.636382790541622,
(OTA_FLOAT)-0.636382790541622, (OTA_FLOAT)-0.21149639206314,
(OTA_FLOAT)-0.126142797829284, (OTA_FLOAT)-0.0892973169642373,
(OTA_FLOAT)-0.0686251794236621, (OTA_FLOAT)-0.0553088553412474,
(OTA_FLOAT)-0.0459578419490588, (OTA_FLOAT)-0.0389906152045799,
(OTA_FLOAT)-0.0335704930563346, (OTA_FLOAT)-0.0292135809519727,
(OTA_FLOAT)-0.0256207450012417, (OTA_FLOAT)-0.0225972483802055,
(OTA_FLOAT)-0.0200109598098972, (OTA_FLOAT)-0.0177691340229538,
(OTA_FLOAT)-0.015804798677585, (OTA_FLOAT)-0.0140684095664809,
(OTA_FLOAT)-0.0125225393246692, (OTA_FLOAT)-0.0111383864687109,
(OTA_FLOAT)-0.00989341621778745, (OTA_FLOAT)-0.00876972703215903,
(OTA_FLOAT)-0.0077528952699248, (OTA_FLOAT)-0.00683114249244957,
(OTA_FLOAT)-0.00599472522492496, (OTA_FLOAT)-0.00523548108358237,
(OTA_FLOAT)-0.00454648675635105, (OTA_FLOAT)-0.00392179728292591,
(OTA_FLOAT)-0.00335624529732888, (OTA_FLOAT)-0.0028452850955481,
(OTA_FLOAT)-0.00238487063159877, (OTA_FLOAT)-0.00197135949141317,
(OTA_FLOAT)-0.00160143696990024, (OTA_FLOAT)-0.00127205585862345,
(OTA_FLOAT)-0.000980388622627241, (OTA_FLOAT)-0.000723789427771377,
(OTA_FLOAT)-0.000499764058146784, (OTA_FLOAT)-0.000305946194467046,
(OTA_FLOAT)-0.000140078849087584, (OTA_FLOAT)-2.11844521182629e-017,
(OTA_FLOAT)0.000116368344857943, (OTA_FLOAT)0.000211028275657584,
(OTA_FLOAT)0.000285911810113899, (OTA_FLOAT)0.00034288484410067,
(OTA_FLOAT)0.000383749572657814, (OTA_FLOAT)0.00041024567428938,
(OTA_FLOAT)0.000424050506139799, (OTA_FLOAT)0.00042677852084975,
(OTA_FLOAT)0.000419980085743287, (OTA_FLOAT)0.000405139860037364,
(OTA_FLOAT)0.000383674864879293, (OTA_FLOAT)0.000356932363344371,
(OTA_FLOAT)0.000326187652395574, (OTA_FLOAT)0.000292641855697718,
(OTA_FLOAT)0.000257419794683033, (OTA_FLOAT)0.000221568005064868,
(OTA_FLOAT)0.000186052956842886, (OTA_FLOAT)0.000151759527541704,
(OTA_FLOAT)0.000119489770824989, (OTA_FLOAT)8.99620156103482e-005,

```

```

(OTA_FLOAT)6.38103242876749e-005, (OTA_FLOAT)4.15843325453698e-005,
(OTA_FLOAT)2.37494875840618e-005, (OTA_FLOAT)1.0687696107602e-005,
(OTA_FLOAT)2.6983883994397e-006, (OTA_FLOAT)0},
702      {(OTA_FLOAT)0, (OTA_FLOAT)1.3782876517259e-006, (OTA_FLOAT)9.94209950627019e-006,
(OTA_FLOAT)2.82130108165914e-005, (OTA_FLOAT)5.09266270309217e-005,
(OTA_FLOAT)6.50453128313997e-005, (OTA_FLOAT)5.35490480839141e-005,
(OTA_FLOAT)2.18943235427543e-006, (OTA_FLOAT)-9.28269381369556e-005,
(OTA_FLOAT)-0.000218947987820953, (OTA_FLOAT)-0.000345779759036334,
(OTA_FLOAT)-0.000429593197019712, (OTA_FLOAT)-0.000423773924198532,
(OTA_FLOAT)-0.000293225563975695, (OTA_FLOAT)-2.90917167844116e-005,
(OTA_FLOAT)0.000340670956072303, (OTA_FLOAT)0.000748992022294435,
(OTA_FLOAT)0.00109818284419612, (OTA_FLOAT)0.00127915819569711,
(OTA_FLOAT)0.00119829470440758, (OTA_FLOAT)0.000806154773079058,
(OTA_FLOAT)0.000120733481440792, (OTA_FLOAT)-0.000761990155256807,
(OTA_FLOAT)-0.00167537832780554, (OTA_FLOAT)-0.0024088114948586,
(OTA_FLOAT)-0.0027505256326668, (OTA_FLOAT)-0.00253882971528567,
(OTA_FLOAT)-0.00171057092770846, (OTA_FLOAT)-0.000334598165599848,
(OTA_FLOAT)0.00138032698709215, (OTA_FLOAT)0.00310923753079091,
(OTA_FLOAT)0.00446578169503636, (OTA_FLOAT)0.005080277629007,
(OTA_FLOAT)0.00468656933576519, (OTA_FLOAT)0.00319914433855478,
(OTA_FLOAT)0.000761563836479265, (OTA_FLOAT)-0.00224890902674718,
(OTA_FLOAT)-0.00526787065507411, (OTA_FLOAT)-0.00763856406572342,
(OTA_FLOAT)-0.00874093718006057, (OTA_FLOAT)-0.00813172836727459,
(OTA_FLOAT)-0.00566695503180555, (OTA_FLOAT)-0.00157829499740953,
(OTA_FLOAT)0.00351898069327334, (OTA_FLOAT)0.00869731599309229,
(OTA_FLOAT)0.012863155768335, (OTA_FLOAT)0.0149581598980236,
(OTA_FLOAT)0.0141819498946929, (OTA_FLOAT)0.0101950225976027,
(OTA_FLOAT)0.00325995060380757, (OTA_FLOAT)-0.00571333747433954,
(OTA_FLOAT)-0.0152373956253714, (OTA_FLOAT)-0.0234314461268484,
(OTA_FLOAT)-0.0282977186728134, (OTA_FLOAT)-0.0280488942568708,
(OTA_FLOAT)-0.0214330762627211, (OTA_FLOAT)-0.00799925772490239,
(OTA_FLOAT)0.011746970648862, (OTA_FLOAT)0.0363314591160109,
(OTA_FLOAT)0.0634540739203883, (OTA_FLOAT)0.0902704359742152,
(OTA_FLOAT)0.113772980878523, (OTA_FLOAT)0.131213207479863,
(OTA_FLOAT)0.140498333359239, (OTA_FLOAT)0.140498333359239,
(OTA_FLOAT)0.131213207479863, (OTA_FLOAT)0.113772980878523,
(OTA_FLOAT)0.0902704359742152, (OTA_FLOAT)0.0634540739203883,
(OTA_FLOAT)0.0363314591160109, (OTA_FLOAT)0.011746970648862,
(OTA_FLOAT)-0.00799925772490239, (OTA_FLOAT)-0.0214330762627211,
(OTA_FLOAT)-0.0280488942568708, (OTA_FLOAT)-0.0282977186728134,
(OTA_FLOAT)-0.0234314461268484, (OTA_FLOAT)-0.0152373956253714,
(OTA_FLOAT)-0.00571333747433954, (OTA_FLOAT)0.00325995060380757,
(OTA_FLOAT)0.0101950225976027, (OTA_FLOAT)0.0141819498946929,
(OTA_FLOAT)0.0149581598980236, (OTA_FLOAT)0.012863155768335,
(OTA_FLOAT)0.00869731599309229, (OTA_FLOAT)0.00351898069327334,
(OTA_FLOAT)-0.00157829499740953, (OTA_FLOAT)-0.00566695503180555,
(OTA_FLOAT)-0.00813172836727459, (OTA_FLOAT)-0.00874093718006057,
(OTA_FLOAT)-0.00763856406572342, (OTA_FLOAT)-0.00526787065507411,
(OTA_FLOAT)-0.00224890902674718, (OTA_FLOAT)0.000761563836479265,
(OTA_FLOAT)0.00319914433855478, (OTA_FLOAT)0.00468656933576519,
(OTA_FLOAT)0.005080277629007, (OTA_FLOAT)0.00446578169503636,
(OTA_FLOAT)0.00310923753079091, (OTA_FLOAT)0.00138032698709215,
(OTA_FLOAT)-0.000334598165599848, (OTA_FLOAT)-0.00171057092770846,
(OTA_FLOAT)-0.00253882971528567, (OTA_FLOAT)-0.0027505256326668,
(OTA_FLOAT)-0.0024088114948586, (OTA_FLOAT)-0.00167537832780554,
(OTA_FLOAT)-0.000761990155256807, (OTA_FLOAT)0.000120733481440792,
(OTA_FLOAT)0.000806154773079058, (OTA_FLOAT)0.00119829470440758,
(OTA_FLOAT)0.00127915819569711, (OTA_FLOAT)0.00109818284419612,
(OTA_FLOAT)0.000748992022294435, (OTA_FLOAT)0.000340670956072303,
(OTA_FLOAT)-2.90917167844116e-005, (OTA_FLOAT)-0.000293225563975695,
(OTA_FLOAT)-0.000423773924198532, (OTA_FLOAT)-0.000429593197019712,
(OTA_FLOAT)-0.000345779759036334, (OTA_FLOAT)-0.000218947987820953,
(OTA_FLOAT)-9.28269381369556e-005, (OTA_FLOAT)2.18943235427543e-006,
(OTA_FLOAT)5.35490480839141e-005, (OTA_FLOAT)6.50453128313997e-005,
(OTA_FLOAT)5.09266270309217e-005, (OTA_FLOAT)2.82130108165914e-005,
(OTA_FLOAT)9.94209950627019e-006, (OTA_FLOAT)1.3782876517259e-006, (OTA_FLOAT)0}
703      }
704    },

```



```

705 {
706   {
707     8000, 128, (OTA_FLOAT)290, (OTA_FLOAT)3300,
708     {(OTA_FLOAT)-0, (OTA_FLOAT)-3.05404847942334e-007, (OTA_FLOAT)1.63580143569299e-006,
(OTA_FLOAT)1.01169676297253e-005, (OTA_FLOAT)2.88338989258826e-005,
(OTA_FLOAT)6.01467120480472e-005, (OTA_FLOAT)0.000104439046151219,
(OTA_FLOAT)0.000159654323352956, (OTA_FLOAT)0.000221076560193795,
(OTA_FLOAT)0.000281404110534686, (OTA_FLOAT)0.000331139978987798,
(OTA_FLOAT)0.000359293813749445, (OTA_FLOAT)0.000354360598109242,
(OTA_FLOAT)0.000305511876050426, (OTA_FLOAT)0.000203909583899209,
(OTA_FLOAT)4.40325844533288e-005, (OTA_FLOAT)-0.000175106159877803,
(OTA_FLOAT)-0.000448976668680166, (OTA_FLOAT)-0.00076690081542634,
(OTA_FLOAT)-0.0011187287165383, (OTA_FLOAT)-0.00146090434583459,
(OTA_FLOAT)-0.00178593034468523, (OTA_FLOAT)-0.00205527333424225,
(OTA_FLOAT)-0.00223561563006736, (OTA_FLOAT)-0.00229438736001068,
(OTA_FLOAT)-0.00220243536531247, (OTA_FLOAT)-0.00193680383218103,
(OTA_FLOAT)-0.00148343237802751, (OTA_FLOAT)-0.000839564329571236,
(OTA_FLOAT)-1.56587688771088e-005, (OTA_FLOAT)0.000963384557914574,
(OTA_FLOAT)0.00205784831619151, (OTA_FLOAT)0.00321378606447147,
(OTA_FLOAT)0.00436463894245902, (OTA_FLOAT)0.00543388061543495,
(OTA_FLOAT)0.00633862343731497, (OTA_FLOAT)0.00699405166376024,
(OTA_FLOAT)0.00731848490112164, (OTA_FLOAT)0.00723882129153027,
(OTA_FLOAT)0.00669606933800941, (OTA_FLOAT)0.00565065327658828,
(OTA_FLOAT)0.00408717216691274, (OTA_FLOAT)0.00201830904693764,
(OTA_FLOAT)-0.000512375860880372, (OTA_FLOAT)-0.0034289752884003,
(OTA_FLOAT)-0.00662321757080101, (OTA_FLOAT)-0.00995593496334061,
(OTA_FLOAT)-0.0132599386414919, (OTA_FLOAT)-0.0163440934182299,
(OTA_FLOAT)-0.0189982383373247, (OTA_FLOAT)-0.0209984234747386,
(OTA_FLOAT)-0.0221117043229826, (OTA_FLOAT)-0.0220994000041538,
(OTA_FLOAT)-0.020717165815416, (OTA_FLOAT)-0.0177092018308156,
(OTA_FLOAT)-0.0127918338001011, (OTA_FLOAT)-0.0056171176544944,
(OTA_FLOAT)0.00430385050779483, (OTA_FLOAT)0.017767243736987,
(OTA_FLOAT)0.036268931682656, (OTA_FLOAT)0.0630711067547088,
(OTA_FLOAT)0.106824852972185, (OTA_FLOAT)0.199666718356604,
(OTA_FLOAT)0.632399333150009, (OTA_FLOAT)-0.632399333150009,
(OTA_FLOAT)-0.199666718356604, (OTA_FLOAT)-0.106824852972185,
(OTA_FLOAT)-0.0630711067547088, (OTA_FLOAT)-0.036268931682656,
(OTA_FLOAT)-0.017767243736987, (OTA_FLOAT)-0.00430385050779483,
(OTA_FLOAT)0.0056171176544944, (OTA_FLOAT)0.0127918338001011,
(OTA_FLOAT)0.0177092018308156, (OTA_FLOAT)0.020717165815416,
(OTA_FLOAT)0.0220994000041538, (OTA_FLOAT)0.0221117043229826,
(OTA_FLOAT)0.0209984234747386, (OTA_FLOAT)0.0189982383373247,
(OTA_FLOAT)0.0163440934182299, (OTA_FLOAT)0.0132599386414919,
(OTA_FLOAT)0.00995593496334061, (OTA_FLOAT)0.00662321757080101,
(OTA_FLOAT)0.0034289752884003, (OTA_FLOAT)0.000512375860880372,
(OTA_FLOAT)-0.00201830904693764, (OTA_FLOAT)-0.00408717216691274,
(OTA_FLOAT)-0.00565065327658828, (OTA_FLOAT)-0.00669606933800941,
(OTA_FLOAT)-0.00723882129153027, (OTA_FLOAT)-0.00731848490112164,
(OTA_FLOAT)-0.00699405166376024, (OTA_FLOAT)-0.00633862343731497,
(OTA_FLOAT)-0.00543388061543495, (OTA_FLOAT)-0.00436463894245902,
(OTA_FLOAT)-0.00321378606447147, (OTA_FLOAT)-0.00205784831619151,
(OTA_FLOAT)-0.000963384557914574, (OTA_FLOAT)1.56587688771088e-005,
(OTA_FLOAT)0.000839564329571236, (OTA_FLOAT)0.00148343237802751,
(OTA_FLOAT)0.00193680383218103, (OTA_FLOAT)0.00220243536531247,
(OTA_FLOAT)0.00229438736001068, (OTA_FLOAT)0.00223561563006736,
(OTA_FLOAT)0.00205527333424225, (OTA_FLOAT)0.00178593034468523,
(OTA_FLOAT)0.00146090434583459, (OTA_FLOAT)0.0011187287165383,
(OTA_FLOAT)0.00076690081542634, (OTA_FLOAT)0.000448976668680166,
(OTA_FLOAT)0.000175106159877803, (OTA_FLOAT)-4.40325844533288e-005,
(OTA_FLOAT)-0.000203909583899209, (OTA_FLOAT)-0.000305511876050426,
(OTA_FLOAT)-0.000354360598109242, (OTA_FLOAT)-0.000359293813749445,
(OTA_FLOAT)-0.000331139978987798, (OTA_FLOAT)-0.000281404110534686,
(OTA_FLOAT)-0.000221076560193795, (OTA_FLOAT)-0.000159654323352956,
(OTA_FLOAT)-0.000104439046151219, (OTA_FLOAT)-6.01467120480472e-005,
(OTA_FLOAT)-2.88338989258826e-005, (OTA_FLOAT)-1.01169676297253e-005,
(OTA_FLOAT)-1.63580143569299e-006, (OTA_FLOAT)3.05404847942334e-007, (OTA_FLOAT)0},
709   {(OTA_FLOAT)0, (OTA_FLOAT)-3.05596017461066e-006, (OTA_FLOAT)9.29522069431249e-006,
(OTA_FLOAT)-7.85066874065769e-006, (OTA_FLOAT)-1.41710532597134e-005,

```

```

(OTA_FLOAT)6.08131650178262e-005, (OTA_FLOAT)-0.000118729757266445,
(OTA_FLOAT)0.000156904876327746, (OTA_FLOAT)-0.000137248974452344,
(OTA_FLOAT)3.34672280713139e-005, (OTA_FLOAT)0.000149338996306247,
(OTA_FLOAT)0.000364140980614182, (OTA_FLOAT)0.000528407394236643,
(OTA_FLOAT)-0.000549387131621376, (OTA_FLOAT)0.000362020493729429,
(OTA_FLOAT)3.3877215493707e-005, (OTA_FLOAT)-0.0005533890529803,
(OTA_FLOAT)0.00103606775539705, (OTA_FLOAT)-0.00128872968629326,
(OTA_FLOAT)0.00115217074830989, (OTA_FLOAT)-0.000570964108726107,
(OTA_FLOAT)-0.000360084677605642, (OTA_FLOAT)0.00139564786736829,
(OTA_FLOAT)-0.00219556506440491, (OTA_FLOAT)0.00242722833323754,
(OTA_FLOAT)-0.0018863205208857, (OTA_FLOAT)0.000595610899229105,
(OTA_FLOAT)0.00115780091570151, (OTA_FLOAT)-0.00287130587281438,
(OTA_FLOAT)0.00395984223944525, (OTA_FLOAT)-0.00394141181810288,
(OTA_FLOAT)0.00261974030819181, (OTA_FLOAT)-0.000200815039882865,
(OTA_FLOAT)-0.00270730067457599, (OTA_FLOAT)0.0052300924725435,
(OTA_FLOAT)-0.0064734382269568, (OTA_FLOAT)0.00581646062026242,
(OTA_FLOAT)-0.00316017397375063, (OTA_FLOAT)-0.000956832894849882,
(OTA_FLOAT)0.00543350420210937, (OTA_FLOAT)-0.00887596153222004,
(OTA_FLOAT)0.0100028661386183, (OTA_FLOAT)-0.0080757928291605,
(OTA_FLOAT)-0.00322066519824294, (OTA_FLOAT)0.00347742061796963,
(OTA_FLOAT)-0.0101682781114691, (OTA_FLOAT)0.0147001811019646,
(OTA_FLOAT)-0.0152476753420488, (OTA_FLOAT)0.0109345116437842,
(OTA_FLOAT)-0.00226226189729776, (OTA_FLOAT)-0.00881081794846686,
(OTA_FLOAT)0.0192124871995341, (OTA_FLOAT)-0.0254792726690014,
(OTA_FLOAT)0.0247052667209489, (OTA_FLOAT)-0.0154842653014079,
(OTA_FLOAT)-0.00140616287164379, (OTA_FLOAT)0.0227767970509813,
(OTA_FLOAT)-0.0433325529225627, (OTA_FLOAT)0.056416043456163,
(OTA_FLOAT)-0.0548643031820446, (OTA_FLOAT)0.0312424778606904,
(OTA_FLOAT)0.0247447891401989, (OTA_FLOAT)-0.143847733278979,
(OTA_FLOAT)0.612624196943492, (OTA_FLOAT)0.612624196943492,
(OTA_FLOAT)-0.143847733278979, (OTA_FLOAT)0.0247447891401989,
(OTA_FLOAT)0.0312424778606904, (OTA_FLOAT)-0.0548643031820446,
(OTA_FLOAT)0.056416043456163, (OTA_FLOAT)-0.0433325529225627,
(OTA_FLOAT)0.0227767970509813, (OTA_FLOAT)-0.00140616287164379,
(OTA_FLOAT)-0.0154842653014079, (OTA_FLOAT)0.0247052667209489,
(OTA_FLOAT)-0.0254792726690014, (OTA_FLOAT)0.0192124871995341,
(OTA_FLOAT)-0.00881081794846686, (OTA_FLOAT)-0.00226226189729776,
(OTA_FLOAT)0.0109345116437842, (OTA_FLOAT)-0.0152476753420488,
(OTA_FLOAT)0.0147001811019646, (OTA_FLOAT)-0.0101682781114691,
(OTA_FLOAT)0.00347742061796963, (OTA_FLOAT)0.00322066519824294,
(OTA_FLOAT)-0.0080757928291605, (OTA_FLOAT)0.0100028661386183,
(OTA_FLOAT)-0.00887596153222004, (OTA_FLOAT)0.00543350420210937,
(OTA_FLOAT)-0.000956832894849882, (OTA_FLOAT)-0.00316017397375063,
(OTA_FLOAT)0.00581646062026242, (OTA_FLOAT)-0.0064734382269568,
(OTA_FLOAT)0.0052300924725435, (OTA_FLOAT)-0.00270730067457599,
(OTA_FLOAT)-0.000200815039882865, (OTA_FLOAT)0.00261974030819181,
(OTA_FLOAT)-0.00394141181810288, (OTA_FLOAT)0.00395984223944525,
(OTA_FLOAT)-0.00287130587281438, (OTA_FLOAT)0.00115780091570151,
(OTA_FLOAT)0.000595610899229105, (OTA_FLOAT)-0.0018863205208857,
(OTA_FLOAT)0.00242722833323754, (OTA_FLOAT)-0.00219556506440491,
(OTA_FLOAT)0.00139564786736829, (OTA_FLOAT)-0.000360084677605642,
(OTA_FLOAT)-0.000570964108726107, (OTA_FLOAT)0.00115217074830989,
(OTA_FLOAT)-0.00128872968629326, (OTA_FLOAT)0.00103606775539705,
(OTA_FLOAT)-0.0005533890529803, (OTA_FLOAT)3.3877215493707e-005,
(OTA_FLOAT)0.000362020493729429, (OTA_FLOAT)-0.000549387131621376,
(OTA_FLOAT)0.000528407394236643, (OTA_FLOAT)-0.000364140980614182,
(OTA_FLOAT)0.000149338996306247, (OTA_FLOAT)3.34672280713139e-005,
(OTA_FLOAT)-0.000137248974452344, (OTA_FLOAT)0.000156904876327746,
(OTA_FLOAT)-0.000118729757266445, (OTA_FLOAT)6.08131650178262e-005,
(OTA_FLOAT)-1.41710532597134e-005, (OTA_FLOAT)-7.85066874065769e-006,
(OTA_FLOAT)9.29522069431249e-006, (OTA_FLOAT)-3.05596017461066e-006, (OTA_FLOAT)0}
},
{
16000, 128, (OTA_FLOAT)290, (OTA_FLOAT)3300,
{(OTA_FLOAT)0, (OTA_FLOAT)2.09246405460004e-006, (OTA_FLOAT)9.51149646890098e-006,
(OTA_FLOAT)2.37602876236498e-005, (OTA_FLOAT)4.59939400981994e-005,
(OTA_FLOAT)7.69397630765245e-005, (OTA_FLOAT)0.000116827805618319,
(OTA_FLOAT)0.000165334132873436, (OTA_FLOAT)0.000221539091570805,

```

710  
711  
712  
713

(OTA\_FLOAT)0.000283902489393468, (OTA\_FLOAT)0.000350257233927784,  
 (OTA\_FLOAT)0.000417822549060194, (OTA\_FLOAT)0.00048323741966789,  
 (OTA\_FLOAT)0.00054261442004609, (OTA\_FLOAT)0.000591613569483497,  
 (OTA\_FLOAT)0.000625535342138959, (OTA\_FLOAT)0.000639431450652153,  
 (OTA\_FLOAT)0.000628231536580361, (OTA\_FLOAT)0.000586883448410472,  
 (OTA\_FLOAT)0.000510504381656901, (OTA\_FLOAT)0.000394539806738295,  
 (OTA\_FLOAT)0.00023492682919912, (OTA\_FLOAT)2.82584224042059e-005,  
 (OTA\_FLOAT)-0.000228055147377841, (OTA\_FLOAT)-0.000535631413637879,  
 (OTA\_FLOAT)-0.000894970965816245, (OTA\_FLOAT)-0.00130532635350048,  
 (OTA\_FLOAT)-0.001764587863988, (OTA\_FLOAT)-0.00226918904225252,  
 (OTA\_FLOAT)-0.00281403437073519, (OTA\_FLOAT)-0.00339245098224269,  
 (OTA\_FLOAT)-0.00399616564570655, (OTA\_FLOAT)-0.00461530754436875,  
 (OTA\_FLOAT)-0.00523843656042112, (OTA\_FLOAT)-0.00585259588770726,  
 (OTA\_FLOAT)-0.00644338680846541, (OTA\_FLOAT)-0.00699506237731813,  
 (OTA\_FLOAT)-0.00749063553026784, (OTA\_FLOAT)-0.00791199573528792,  
 (OTA\_FLOAT)-0.00824002665479666, (OTA\_FLOAT)-0.00845471528952305,  
 (OTA\_FLOAT)-0.00853524054756141, (OTA\_FLOAT)-0.00846002586639676,  
 (OTA\_FLOAT)-0.00820673599408838, (OTA\_FLOAT)-0.00775219165197553,  
 (OTA\_FLOAT)-0.00707216650167503, (OTA\_FLOAT)-0.00614101690245842,  
 (OTA\_FLOAT)-0.00493107349810215, (OTA\_FLOAT)-0.00341168979981045,  
 (OTA\_FLOAT)-0.00154778798870695, (OTA\_FLOAT)0.000702349540691461,  
 (OTA\_FLOAT)0.00339045232657068, (OTA\_FLOAT)0.00658348900153348,  
 (OTA\_FLOAT)0.0103719647606968, (OTA\_FLOAT)0.0148835253513044,  
 (OTA\_FLOAT)0.0203062064154604, (OTA\_FLOAT)0.0269302936456114,  
 (OTA\_FLOAT)0.035228792993986, (OTA\_FLOAT)0.0460256005064514,  
 (OTA\_FLOAT)0.0608877338790662, (OTA\_FLOAT)0.083189677350527,  
 (OTA\_FLOAT)0.12173166583442, (OTA\_FLOAT)0.208830158059189, (OTA\_FLOAT)0.635490811965674,  
 (OTA\_FLOAT)-0.635490811965674, (OTA\_FLOAT)-0.208830158059189,  
 (OTA\_FLOAT)-0.12173166583442, (OTA\_FLOAT)-0.083189677350527,  
 (OTA\_FLOAT)-0.0608877338790662, (OTA\_FLOAT)-0.0460256005064514,  
 (OTA\_FLOAT)-0.035228792993986, (OTA\_FLOAT)-0.0269302936456114,  
 (OTA\_FLOAT)-0.0203062064154604, (OTA\_FLOAT)-0.0148835253513044,  
 (OTA\_FLOAT)-0.0103719647606968, (OTA\_FLOAT)-0.00658348900153348,  
 (OTA\_FLOAT)-0.00339045232657068, (OTA\_FLOAT)-0.000702349540691461,  
 (OTA\_FLOAT)0.00154778798870695, (OTA\_FLOAT)0.00341168979981045,  
 (OTA\_FLOAT)0.00493107349810215, (OTA\_FLOAT)0.00614101690245842,  
 (OTA\_FLOAT)0.00707216650167503, (OTA\_FLOAT)0.00775219165197553,  
 (OTA\_FLOAT)0.00820673599408838, (OTA\_FLOAT)0.00846002586639676,  
 (OTA\_FLOAT)0.00853524054756141, (OTA\_FLOAT)0.00845471528952305,  
 (OTA\_FLOAT)0.00824002665479666, (OTA\_FLOAT)0.00791199573528792,  
 (OTA\_FLOAT)0.00749063553026784, (OTA\_FLOAT)0.00699506237731813,  
 (OTA\_FLOAT)0.00644338680846541, (OTA\_FLOAT)0.00585259588770726,  
 (OTA\_FLOAT)0.00523843656042112, (OTA\_FLOAT)0.00461530754436875,  
 (OTA\_FLOAT)0.00399616564570655, (OTA\_FLOAT)0.00339245098224269,  
 (OTA\_FLOAT)0.00281403437073519, (OTA\_FLOAT)0.00226918904225252,  
 (OTA\_FLOAT)0.001764587863988, (OTA\_FLOAT)0.00130532635350048,  
 (OTA\_FLOAT)0.000894970965816245, (OTA\_FLOAT)0.000535631413637879,  
 (OTA\_FLOAT)0.000228055147377841, (OTA\_FLOAT)-2.82584224042059e-005,  
 (OTA\_FLOAT)-0.00023492682919912, (OTA\_FLOAT)-0.000394539806738295,  
 (OTA\_FLOAT)-0.000510504381656901, (OTA\_FLOAT)-0.000586883448410472,  
 (OTA\_FLOAT)-0.000628231536580361, (OTA\_FLOAT)-0.000639431450652153,  
 (OTA\_FLOAT)-0.000625535342138959, (OTA\_FLOAT)-0.000591613569483497,  
 (OTA\_FLOAT)-0.00054261442004609, (OTA\_FLOAT)-0.00048323741966789,  
 (OTA\_FLOAT)-0.000417822549060194, (OTA\_FLOAT)-0.000350257233927784,  
 (OTA\_FLOAT)-0.000283902489393468, (OTA\_FLOAT)-0.000221539091570805,  
 (OTA\_FLOAT)-0.000165334132873436, (OTA\_FLOAT)-0.000116827805618319,  
 (OTA\_FLOAT)-7.69397630765245e-005, (OTA\_FLOAT)-4.59939400981994e-005,  
 (OTA\_FLOAT)-2.37602876236498e-005, (OTA\_FLOAT)-9.51149646890098e-006,  
 (OTA\_FLOAT)-2.09246405460004e-006, (OTA\_FLOAT)-0},  
 714 { (OTA\_FLOAT)0, (OTA\_FLOAT)-1.97665270936294e-006, (OTA\_FLOAT)-1.15972625137046e-005,  
 (OTA\_FLOAT)3.96268745132946e-006, (OTA\_FLOAT)5.17142809009005e-005,  
 (OTA\_FLOAT)3.31879996410512e-005, (OTA\_FLOAT)-9.35769857636302e-005,  
 (OTA\_FLOAT)-0.000137204856274504, (OTA\_FLOAT)7.26330853569951e-005,  
 (OTA\_FLOAT)0.000284241723375608, (OTA\_FLOAT)7.6445301124661e-005,  
 (OTA\_FLOAT)-0.000386234473519562, (OTA\_FLOAT)-0.00036651340667661,  
 (OTA\_FLOAT)0.000318397334527441, (OTA\_FLOAT)0.000716892951196256,  
 (OTA\_FLOAT)1.69417971671096e-005, (OTA\_FLOAT)-0.000953179076439295,  
 (OTA\_FLOAT)-0.000615080675530749, (OTA\_FLOAT)0.000862013061925577,

```

(OTA_FLOAT)0.00132005174598715, (OTA_FLOAT)-0.000289997017682688,
(OTA_FLOAT)-0.00183683718469768, (OTA_FLOAT)-0.000749365979243514,
(OTA_FLOAT)0.00181884977467718, (OTA_FLOAT)0.00201076423494457,
(OTA_FLOAT)-0.00101282311774396, (OTA_FLOAT)-0.00303828603480154,
(OTA_FLOAT)-0.000588055899648117, (OTA_FLOAT)0.00328967936066047,
(OTA_FLOAT)0.00264868247381217, (OTA_FLOAT)-0.00233989160553942,
(OTA_FLOAT)-0.00451234377338135, (OTA_FLOAT)0.000100426425968604,
(OTA_FLOAT)0.0053611461339616, (OTA_FLOAT)0.0030311003056721,
(OTA_FLOAT)-0.00449009972863852, (OTA_FLOAT)-0.0061693622113754,
(OTA_FLOAT)0.00161766488998147, (OTA_FLOAT)0.00812651201453167,
(OTA_FLOAT)0.00287544716508195, (OTA_FLOAT)-0.00776154989461552,
(OTA_FLOAT)-0.0078837696951438, (OTA_FLOAT)0.00440725966880395,
(OTA_FLOAT)0.0117531408355639, (OTA_FLOAT)0.00175525569872253,
(OTA_FLOAT)-0.0126865401531798, (OTA_FLOAT)-0.00950835454098354,
(OTA_FLOAT)0.00929277173025337, (OTA_FLOAT)0.0166872026589508,
(OTA_FLOAT)-0.00113309107693372, (OTA_FLOAT)-0.0205562341796008,
(OTA_FLOAT)-0.010892355964647, (OTA_FLOAT)0.0183809136596394,
(OTA_FLOAT)0.0244287503524718, (OTA_FLOAT)-0.00800134875073106,
(OTA_FLOAT)-0.0358097831747896, (OTA_FLOAT)-0.011900791323909,
(OTA_FLOAT)0.0401866793364094, (OTA_FLOAT)0.042465624664692,
(OTA_FLOAT)-0.0304885583598849, (OTA_FLOAT)-0.0888594946134594,
(OTA_FLOAT)-0.0124322033158126, (OTA_FLOAT)0.197337076112644,
(OTA_FLOAT)0.384175071519774, (OTA_FLOAT)0.384175071519774,
(OTA_FLOAT)0.197337076112644, (OTA_FLOAT)-0.0124322033158126,
(OTA_FLOAT)-0.0888594946134594, (OTA_FLOAT)-0.0304885583598849,
(OTA_FLOAT)0.042465624664692, (OTA_FLOAT)0.0401866793364094,
(OTA_FLOAT)-0.011900791323909, (OTA_FLOAT)-0.0358097831747896,
(OTA_FLOAT)-0.00800134875073106, (OTA_FLOAT)0.0244287503524718,
(OTA_FLOAT)0.0183809136596394, (OTA_FLOAT)-0.010892355964647,
(OTA_FLOAT)-0.0205562341796008, (OTA_FLOAT)-0.00113309107693372,
(OTA_FLOAT)0.0166872026589508, (OTA_FLOAT)0.00929277173025337,
(OTA_FLOAT)-0.00950835454098354, (OTA_FLOAT)-0.0126865401531798,
(OTA_FLOAT)0.00175525569872253, (OTA_FLOAT)0.0117531408355639,
(OTA_FLOAT)0.00440725966880395, (OTA_FLOAT)-0.0078837696951438,
(OTA_FLOAT)-0.00776154989461552, (OTA_FLOAT)0.00287544716508195,
(OTA_FLOAT)0.00812651201453167, (OTA_FLOAT)0.00161766488998147,
(OTA_FLOAT)-0.0061693622113754, (OTA_FLOAT)-0.00449009972863852,
(OTA_FLOAT)0.0030311003056721, (OTA_FLOAT)0.0053611461339616,
(OTA_FLOAT)0.000100426425968604, (OTA_FLOAT)-0.00451234377338135,
(OTA_FLOAT)-0.00233989160553942, (OTA_FLOAT)0.00264868247381217,
(OTA_FLOAT)0.00328967936066047, (OTA_FLOAT)-0.000588055899648117,
(OTA_FLOAT)-0.00303828603480154, (OTA_FLOAT)-0.00101282311774396,
(OTA_FLOAT)0.00201076423494457, (OTA_FLOAT)0.00181884977467718,
(OTA_FLOAT)-0.000749365979243514, (OTA_FLOAT)-0.00183683718469768,
(OTA_FLOAT)-0.000289997017682688, (OTA_FLOAT)0.00132005174598715,
(OTA_FLOAT)0.000862013061925577, (OTA_FLOAT)-0.000615080675530749,
(OTA_FLOAT)-0.000953179076439295, (OTA_FLOAT)1.69417971671096e-005,
(OTA_FLOAT)0.000716892951196256, (OTA_FLOAT)0.000318397334527441,
(OTA_FLOAT)-0.00036651340667661, (OTA_FLOAT)-0.000386234473519562,
(OTA_FLOAT)7.6445301124661e-005, (OTA_FLOAT)0.000284241723375608,
(OTA_FLOAT)7.26330853569951e-005, (OTA_FLOAT)-0.000137204856274504,
(OTA_FLOAT)-9.35769857636302e-005, (OTA_FLOAT)3.31879996410512e-005,
(OTA_FLOAT)5.17142809009005e-005, (OTA_FLOAT)3.96268745132946e-006,
(OTA_FLOAT)-1.15972625137046e-005, (OTA_FLOAT)-1.97665270936294e-006, (OTA_FLOAT)0}
},
{
    48000, 128, (OTA_FLOAT)290, (OTA_FLOAT)3300,
    {(OTA_FLOAT)0, (OTA_FLOAT)-2.23897856247086e-006, (OTA_FLOAT)-8.75532523944886e-006,
    (OTA_FLOAT)-1.91975250057304e-005, (OTA_FLOAT)-3.31460930014185e-005,
    (OTA_FLOAT)-5.01132511427717e-005, (OTA_FLOAT)-6.95429023792986e-005,
    (OTA_FLOAT)-9.08108924526822e-005, (OTA_FLOAT)-0.00011322545259763,
    (OTA_FLOAT)-0.000136028453940511, (OTA_FLOAT)-0.000158395505575195,
    (OTA_FLOAT)-0.000179438112846391, (OTA_FLOAT)-0.000198204621153492,
    (OTA_FLOAT)-0.000213681854398354, (OTA_FLOAT)-0.000224796756912184,
    (OTA_FLOAT)-0.000230418082664723, (OTA_FLOAT)-0.000229358075860651,
    (OTA_FLOAT)-0.000220374079122337, (OTA_FLOAT)-0.000202169996479676,
    (OTA_FLOAT)-0.000173397528076385, (OTA_FLOAT)-0.000132657081532078,
    (OTA_FLOAT)-7.8498250851616e-005, (OTA_FLOAT)-9.41973712042425e-006,

```



(OTA\_FLOAT)7.61314347148772e-005, (OTA\_FLOAT)0.000179761572772911,  
 (OTA\_FLOAT)0.000303133048033594, (OTA\_FLOAT)0.000447968332263366,  
 (OTA\_FLOAT)0.000616055624052191, (OTA\_FLOAT)0.000809256398770985,  
 (OTA\_FLOAT)0.00102951528846988, (OTA\_FLOAT)0.00127887278668171,  
 (OTA\_FLOAT)0.00155948138660689, (OTA\_FLOAT)0.00187362590709111,  
 (OTA\_FLOAT)0.00222374894987425, (OTA\_FLOAT)0.00261248267846822,  
 (OTA\_FLOAT)0.00304268843407926, (OTA\_FLOAT)0.00351750613571656,  
 (OTA\_FLOAT)0.00404041599035974, (OTA\_FLOAT)0.00461531582250181,  
 (OTA\_FLOAT)0.00524661840418207, (OTA\_FLOAT)0.00593937464953628,  
 (OTA\_FLOAT)0.00669943061471832, (OTA\_FLOAT)0.00753362919107919,  
 (OTA\_FLOAT)0.00845007162129215, (OTA\_FLOAT)0.00945846016871387,  
 (OTA\_FLOAT)0.0105705524885501, (OTA\_FLOAT)0.0118007722097639,  
 (OTA\_FLOAT)0.0131670418131792, (OTA\_FLOAT)0.0146919379975394,  
 (OTA\_FLOAT)0.0164043250026625, (OTA\_FLOAT)0.0183417134895455,  
 (OTA\_FLOAT)0.0205537510439994, (OTA\_FLOAT)0.0231075328565812,  
 (OTA\_FLOAT)0.0260959457509884, (OTA\_FLOAT)0.029651280362664,  
 (OTA\_FLOAT)0.0339684496269281, (OTA\_FLOAT)0.0393467791253324,  
 (OTA\_FLOAT)0.046270369385838, (OTA\_FLOAT)0.0555761213139239,  
 (OTA\_FLOAT)0.0688457891799863, (OTA\_FLOAT)0.0894701157650805,  
 (OTA\_FLOAT)0.126266879085292, (OTA\_FLOAT)0.211571103850569,  
 (OTA\_FLOAT)0.636407741809339, (OTA\_FLOAT)-0.636407741809339,  
 (OTA\_FLOAT)-0.211571103850569, (OTA\_FLOAT)-0.126266879085292,  
 (OTA\_FLOAT)-0.0894701157650805, (OTA\_FLOAT)-0.0688457891799863,  
 (OTA\_FLOAT)-0.0555761213139239, (OTA\_FLOAT)-0.046270369385838,  
 (OTA\_FLOAT)-0.0393467791253324, (OTA\_FLOAT)-0.0339684496269281,  
 (OTA\_FLOAT)-0.029651280362664, (OTA\_FLOAT)-0.0260959457509884,  
 (OTA\_FLOAT)-0.0231075328565812, (OTA\_FLOAT)-0.0205537510439994,  
 (OTA\_FLOAT)-0.0183417134895455, (OTA\_FLOAT)-0.0164043250026625,  
 (OTA\_FLOAT)-0.0146919379975394, (OTA\_FLOAT)-0.0131670418131792,  
 (OTA\_FLOAT)-0.0118007722097639, (OTA\_FLOAT)-0.0105705524885501,  
 (OTA\_FLOAT)-0.00945846016871387, (OTA\_FLOAT)-0.00845007162129215,  
 (OTA\_FLOAT)-0.00753362919107919, (OTA\_FLOAT)-0.00669943061471832,  
 (OTA\_FLOAT)-0.00593937464953628, (OTA\_FLOAT)-0.00524661840418207,  
 (OTA\_FLOAT)-0.00461531582250181, (OTA\_FLOAT)-0.00404041599035974,  
 (OTA\_FLOAT)-0.00351750613571656, (OTA\_FLOAT)-0.00304268843407926,  
 (OTA\_FLOAT)-0.00261248267846822, (OTA\_FLOAT)-0.00222374894987425,  
 (OTA\_FLOAT)-0.00187362590709111, (OTA\_FLOAT)-0.00155948138660689,  
 (OTA\_FLOAT)-0.00127887278668171, (OTA\_FLOAT)-0.00102951528846988,  
 (OTA\_FLOAT)-0.000809256398770985, (OTA\_FLOAT)-0.000616055624052191,  
 (OTA\_FLOAT)-0.000447968332263366, (OTA\_FLOAT)-0.000303133048033594,  
 (OTA\_FLOAT)-0.000179761572772911, (OTA\_FLOAT)-7.61314347148772e-005,  
 (OTA\_FLOAT)9.41973712042425e-006, (OTA\_FLOAT)7.8498250851616e-005,  
 (OTA\_FLOAT)0.000132657081532078, (OTA\_FLOAT)0.000173397528076385,  
 (OTA\_FLOAT)0.000202169996479676, (OTA\_FLOAT)0.000220374079122337,  
 (OTA\_FLOAT)0.000229358075860651, (OTA\_FLOAT)0.000230418082664723,  
 (OTA\_FLOAT)0.000224796756912184, (OTA\_FLOAT)0.000213681854398354,  
 (OTA\_FLOAT)0.000198204621153492, (OTA\_FLOAT)0.000179438112846391,  
 (OTA\_FLOAT)0.000158395505575195, (OTA\_FLOAT)0.000136028453940511,  
 (OTA\_FLOAT)0.000113225545259763, (OTA\_FLOAT)9.08108924526822e-005,  
 (OTA\_FLOAT)6.95429023792986e-005, (OTA\_FLOAT)5.01132511427717e-005,  
 (OTA\_FLOAT)3.31460930014185e-005, (OTA\_FLOAT)1.91975250057304e-005,  
 (OTA\_FLOAT)8.75532523944886e-006, (OTA\_FLOAT)2.23897856247086e-006, (OTA\_FLOAT)0},  
 719 {(OTA\_FLOAT)0, (OTA\_FLOAT)2.98192764209086e-006, (OTA\_FLOAT)1.25399624386053e-005,  
 (OTA\_FLOAT)2.43609768652177e-005, (OTA\_FLOAT)2.81492129453465e-005,  
 (OTA\_FLOAT)1.13477286951114e-005, (OTA\_FLOAT)-3.51437614143393e-005,  
 (OTA\_FLOAT)-0.000111100667744684, (OTA\_FLOAT)-0.000203130724094745,  
 (OTA\_FLOAT)-0.000284707193216374, (OTA\_FLOAT)-0.000320975251334624,  
 (OTA\_FLOAT)-0.00027785629583166, (OTA\_FLOAT)-0.000133533190222227,  
 (OTA\_FLOAT)0.000110703972252248, (OTA\_FLOAT)0.000423676435095292,  
 (OTA\_FLOAT)0.000744517733859796, (OTA\_FLOAT)0.000991365466785937,  
 (OTA\_FLOAT)0.00107799063534734, (OTA\_FLOAT)0.000935552587300108,  
 (OTA\_FLOAT)0.000534777043634481, (OTA\_FLOAT)-9.71234856373313e-005,  
 (OTA\_FLOAT)-0.000870149704612278, (OTA\_FLOAT)-0.00163947594557516,  
 (OTA\_FLOAT)-0.002228408171878, (OTA\_FLOAT)-0.0024635368545848,  
 (OTA\_FLOAT)-0.0022158720633941, (OTA\_FLOAT)-0.00143930214269119,  
 (OTA\_FLOAT)-0.000196946986489111, (OTA\_FLOAT)0.00133270912170472,  
 (OTA\_FLOAT)0.00287464523539361, (OTA\_FLOAT)0.00410089495385752,  
 (OTA\_FLOAT)0.00469311790593184, (OTA\_FLOAT)0.00441330127756002,

```

(OTA_FLOAT)0.0031683827805415, (OTA_FLOAT)0.00105388710188467,
(OTA_FLOAT)-0.00163589470469237, (OTA_FLOAT)-0.00443822664527055,
(OTA_FLOAT)-0.00679218195229901, (OTA_FLOAT)-0.00813981986465468,
(OTA_FLOAT)-0.00804167497316753, (OTA_FLOAT)-0.00628486045931389,
(OTA_FLOAT)-0.00296082758971948, (OTA_FLOAT)0.00150694189319646,
(OTA_FLOAT)0.00639749133882223, (OTA_FLOAT)0.0107905919894783,
(OTA_FLOAT)0.0137177835552774, (OTA_FLOAT)0.0143440600881766,
(OTA_FLOAT)0.0121497332319482, (OTA_FLOAT)0.00707870131311368,
(OTA_FLOAT)-0.000377926446366041, (OTA_FLOAT)-0.00918517548699536,
(OTA_FLOAT)-0.0178632742374449, (OTA_FLOAT)-0.0246749296111302,
(OTA_FLOAT)-0.0278771206168382, (OTA_FLOAT)-0.0260005877842487,
(OTA_FLOAT)-0.0181126586736862, (OTA_FLOAT)-0.00401877970603624,
(OTA_FLOAT)0.0156344404381715, (OTA_FLOAT)0.0393776602018564,
(OTA_FLOAT)0.065062886813746, (OTA_FLOAT)0.0901170709843561,
(OTA_FLOAT)0.111870694754096, (OTA_FLOAT)0.127913602749937,
(OTA_FLOAT)0.136424639778673, (OTA_FLOAT)0.136424639778673,
(OTA_FLOAT)0.127913602749937, (OTA_FLOAT)0.111870694754096,
(OTA_FLOAT)0.0901170709843561, (OTA_FLOAT)0.065062886813746,
(OTA_FLOAT)0.0393776602018564, (OTA_FLOAT)0.0156344404381715,
(OTA_FLOAT)-0.00401877970603624, (OTA_FLOAT)-0.0181126586736862,
(OTA_FLOAT)-0.0260005877842487, (OTA_FLOAT)-0.0278771206168382,
(OTA_FLOAT)-0.0246749296111302, (OTA_FLOAT)-0.0178632742374449,
(OTA_FLOAT)-0.00918517548699536, (OTA_FLOAT)-0.000377926446366041,
(OTA_FLOAT)0.00707870131311368, (OTA_FLOAT)0.0121497332319482,
(OTA_FLOAT)0.0143440600881766, (OTA_FLOAT)0.0137177835552774,
(OTA_FLOAT)0.0107905919894783, (OTA_FLOAT)0.00639749133882223,
(OTA_FLOAT)0.00150694189319646, (OTA_FLOAT)-0.00296082758971948,
(OTA_FLOAT)-0.00628486045931389, (OTA_FLOAT)-0.00804167497316753,
(OTA_FLOAT)-0.00813981986465468, (OTA_FLOAT)-0.00679218195229901,
(OTA_FLOAT)-0.00443822664527055, (OTA_FLOAT)-0.00163589470469237,
(OTA_FLOAT)0.00105388710188467, (OTA_FLOAT)0.0031683827805415,
(OTA_FLOAT)0.00441330127756002, (OTA_FLOAT)0.00469311790593184,
(OTA_FLOAT)0.00410089495385752, (OTA_FLOAT)0.00287464523539361,
(OTA_FLOAT)0.00133270912170472, (OTA_FLOAT)-0.000196946986489111,
(OTA_FLOAT)-0.00143930214269119, (OTA_FLOAT)-0.0022158720633941,
(OTA_FLOAT)-0.0024635368545848, (OTA_FLOAT)-0.002228408171878,
(OTA_FLOAT)-0.00163947594557516, (OTA_FLOAT)-0.000870149704612278,
(OTA_FLOAT)-9.71234856373313e-005, (OTA_FLOAT)0.000534777043634481,
(OTA_FLOAT)0.000935552587300108, (OTA_FLOAT)0.00107799063534734,
(OTA_FLOAT)0.000991365466785937, (OTA_FLOAT)0.000744517733859796,
(OTA_FLOAT)0.000423676435095292, (OTA_FLOAT)0.000110703972252248,
(OTA_FLOAT)-0.000133533190222227, (OTA_FLOAT)-0.00027785629583166,
(OTA_FLOAT)-0.000320975251334624, (OTA_FLOAT)-0.000284707193216374,
(OTA_FLOAT)-0.000203130724094745, (OTA_FLOAT)-0.000111100667744684,
(OTA_FLOAT)-3.51437614143393e-005, (OTA_FLOAT)1.13477286951114e-005,
(OTA_FLOAT)2.81492129453465e-005, (OTA_FLOAT)2.43609768652177e-005,
(OTA_FLOAT)1.25399624386053e-005, (OTA_FLOAT)2.98192764209086e-006, (OTA_FLOAT)0}
}
}
}
void CAudioSignal::CreateAsCopy(const CAudioSignal* const rhs)
{
    mSampleRate = rhs->mSampleRate
    mNumChannels = rhs->mNumChannels
    mSignalLength = rhs->mSignalLength
    unsigned long BufferSize = (mSignalLength+rhs->mStartSample) * sizeof(OTA_FLOAT)
    mStartSample = rhs->mStartSample

    for (int c=0 c<mNumChannels c++)
    {
        mpData[c] = (OTA_FLOAT*)matMalloc(BufferSize)
        matbCopy(rhs->mpData[c]-rhs->mStartSample, mpData[c], mSignalLength+rhs->mStartSample)
        mpData[c] += mStartSample
        mpData[c+2] = (OTA_FLOAT*)matMalloc(BufferSize)
        matbCopy(rhs->mpData[c+2]-rhs->mStartSample, mpData[c+2], mSignalLength+rhs->mStartSample)
        mpData[c+2] += mStartSample
    }
}

```

```

741 }
742
743 CAudioSignal::CAudioSignal(CAudioSignal const& rhs)
744 {
745     CreateAsCopy(&rhs)
746 }
747
748 CAudioSignal::CAudioSignal(CAudioSignal* rhs)
749 {
750     CreateAsCopy(rhs)
751 }
752
753 CAudioSignal& CAudioSignal::operator=( const CAudioSignal& rhs)
754 {
755     CreateAsCopy(&rhs)
756     return *this
757 }
758
759 OTA_FLOAT CAudioSignal::GetAverageEnergy(int Channel, unsigned long NumSamples)
760 {
761     if (NumSamples)
762         return matDotProd(mpData[Channel], mpData[Channel], NumSamples) / (OTA_FLOAT)NumSamples
763     else
764         return 0
765 }
766
767 bool CAudioSignal::Set(unsigned long SampleRate, unsigned long NumSamples, int NumChannels,
768     OTA_FLOAT** pSignal, MAT_HANDLE newMh)
769 {
770     int c
771     bool rc=true
772
773     mh = newMh
774
775     Free()
776
777     NumSamples -= 0
778
779     for (c=0 c<NumChannels c++)
780     {
781         mpData[c] = (OTA_FLOAT*)matMalloc((NumSamples+128+128) * sizeof(OTA_FLOAT))
782         mpData[c+2] = (OTA_FLOAT*)matMalloc((NumSamples+128+128) * sizeof(OTA_FLOAT))
783         if (mpData[c])
784         {
785             int s
786             for (s=0 s<NumSamples s++)
787                 mpData[c][s] = pSignal[c][s+0]
788             for (s=NumSamples; s<NumSamples+128+128 s++)
789                 mpData[c][s] = 0
790             matbCopy(mpData[c], mpData[c+2], NumSamples+128+128)
791         }
792         else
793             rc = false
794     }
795
796     mSampleRate = SampleRate
797     mSignalLength = NumSamples+128+128
798     mNumChannels = NumChannels
799
800     if (!rc) Free()
801
802     return rc
803 }
804
805 unsigned long CAudioSignal::SetStartByTrigger(int Channel, OTA_FLOAT Level, int
806     NumConsecutiveSamples, int Offset)
807 {
808     unsigned long TriggerPoint = 0

```

```

807     if (NumConsecutiveSamples<=0) return TriggerPoint
808     if (Level<=0.0) return TriggerPoint
809     if (Channel>=mNumChannels) return TriggerPoint
810     if (!mpData[Channel]) return TriggerPoint
811
812     bool Triggered = false
813     OTA_FLOAT* pBuffer = (OTA_FLOAT*)matMalloc(NumConsecutiveSamples*sizeof(OTA_FLOAT))
814     int NextPos = 0
815     for (int i=0 i<mSignalLength && !Triggered i++)
816     {
817         OTA_FLOAT Data = mpData[Channel][i]
818         pBuffer[NextPos++] = fabs(Data)
819         if (NextPos==NumConsecutiveSamples) NextPos = 0
820         if (i>=NumConsecutiveSamples-1)
821         {
822             OTA_FLOAT Avg=0
823             for (int j=0 j<NumConsecutiveSamples j++)
824                 Avg += pBuffer[j]
825             Avg /= (OTA_FLOAT)NumConsecutiveSamples
826             if (Avg>Level)
827             {
828                 Triggered = true
829                 TriggerPoint = max(0, i-Offset-NumConsecutiveSamples)
830                 SetOffset(TriggerPoint)
831             }
832         }
833     }
834
835     matFree(pBuffer)
836     return TriggerPoint
837 }
838
839 bool CAudioSignal::ApplyFilter(FilteringParameters *filterParams)
840 {
841     int c
842     bool rc = true
843     OTA_FLOAT AverageEnergyBeforeFilter[2]
844
845     int pListeningCondition = 0
846
847     if(filterParams)
848         pListeningCondition = filterParams->pListeningCondition
849
850     for (c=0 c<mNumChannels c++)
851         AverageEnergyBeforeFilter[c] = GetAverageEnergy(c, mSignalLength)
852
853     OTA_FLOAT* pHPCoefficients=0
854     OTA_FLOAT* pLPCoefficients=0
855     int NumTaps=0
856     int FilterDefIndex
857     int ModeIndex = pListeningCondition == 3 ? 0 : 1
858     for (FilterDefIndex=0 FilterDefIndex<3 FilterDefIndex++)
859         if (FilterCoefficients[ModeIndex][FilterDefIndex].SampleRate == mSampleRate)
860             break
861
862     OTA_FLOAT LowCutOff
863     OTA_FLOAT HighCutOff
864     switch(ModeIndex)
865     {
866     {
867         case 0: LowCutOff = 320      HighCutOff = 3400  break
868         case 1: LowCutOff = 290      HighCutOff = 3300  break
869     }
870
871     if (FilterDefIndex<3 && LowCutOff ==FilterCoefficients[ModeIndex][FilterDefIndex].LowCutOff &&
872         HighCutOff==FilterCoefficients[ModeIndex][FilterDefIndex].HighCutOff)
873     {

```



```

874     pHPCoefficients = FilterCoefficients[ModeIndex][FilterDefIndex].HPCoef
875     pLPCoefficients = FilterCoefficients[ModeIndex][FilterDefIndex].LPCoef
876     NumTaps = FilterCoefficients[ModeIndex][FilterDefIndex].NumTaps
877 }
878
879 HighpassFilter((LowCutOff / 128000.0) * 128000/mSampleRate, pHPCoefficients, NumTaps)
880 mSignalLength = mSignalLength-128
881
882 LowpassFilter ((HighCutOff/ 128000.0) * 128000/mSampleRate, pLPCoefficients, NumTaps)
883 mSignalLength = mSignalLength-128
884
885 for (c=0 c<mNumChannels c++)
886 {
887     OTA_FLOAT AverageEnergyAfterFilter = GetAverageEnergy(c, mSignalLength)
888     OTA_FLOAT Ratio = AverageEnergyBeforeFilter[c] / (AverageEnergyAfterFilter+0.1)
889
890     matbMpy1(sqrt(Ratio), mpData[c], mSignalLength)
891
892     for (int i = 0 i < mSignalLength i++)
893         mpData[c][i] = ((int)(mpData[c][i]*1e3)) / (OTA_FLOAT)1e3
894 }
895
896 return rc
897 }
898
899 bool CAudioSignal::LowpassFilter(OTA_FLOAT NormalizedCutOffFreq, OTA_FLOAT* pCoefficients, int
NumTaps)
900 {
901
902     OTA_FLOAT* pTaps
903
904     if (!pCoefficients)
905     {
906         NumTaps = 128
907         pTaps = (OTA_FLOAT*)matxMalloc(NumTaps)
908
909         matGenLowPassCoefficients(NormalizedCutOffFreq, pTaps, NumTaps, MAT_WinBlackman)
910     }
911     else pTaps = pCoefficients
912
913     OTA_FLOAT* pSig = matxMalloc(mSignalLength)
914     for (int c=0 c<mNumChannels c++)
915     {
916         int i
917         matRunFIRFilter(mh, mpData[c], pSig, mSignalLength, pTaps, NumTaps, MAT_FIRNoDelayComp)
918         matbCopy(pSig+NumTaps, mpData[c], mSignalLength-NumTaps)
919         for (i=mSignalLength-NumTaps i<mSignalLength i++)
920             mpData[c][i] = 0
921     }
922     matFree(pSig)
923     if (!pCoefficients)
924         matFree(pTaps)
925
926     return true
927 }
928
929 bool CAudioSignal::HighpassFilter(OTA_FLOAT NormalizedCutOffFreq, OTA_FLOAT* pCoefficients, int
NumTaps)
930 {
931
932     OTA_FLOAT* pTaps
933
934     if (!pCoefficients)
935     {
936         NumTaps = 128
937         pTaps = (OTA_FLOAT*)matxMalloc(NumTaps)
938         matGenHighPassCoefficients(NormalizedCutOffFreq, pTaps, NumTaps, MAT_WinBlackman)
939     }

```

```

940     else pTaps = pCoefficients
941
942     OTA_FLOAT* pSig = matxMalloc(mSignalLength)
943     for (int c=0 c<mNumChannels c++)
944     {
945         int DelyLineIndex=0
946         matRunFIRFilter(mh, mpData[c], pSig, mSignalLength, pTaps, NumTaps, MAT_FIRNoDelayComp)
947         matbCopy(pSig+NumTaps, mpData[c], mSignalLength-NumTaps)
948         for (int i=mSignalLength-NumTaps i<mSignalLength i++)
949             mpData[c][i] = 0
950     }
951     matFree(pSig)
952     if (!pCoefficients)
953         matFree(pTaps)
954
955     return true
956 }
957
958 OTA_FLOAT CAudioSignal::GetEnergy(int Channel)
959 {
960     if (Channel<0 || Channel>mNumChannels) return -1.0
961     return matDotProd(mpData[Channel], mpData[Channel], mSignalLength) / (OTA_FLOAT)mSignalLength
962 }
963
964 OTA_FLOAT CAudioSignal::Amplify(int Channel, OTA_FLOAT Factor)
965 {
966     if (Channel<0 || Channel>mNumChannels) return -1.0
967     matbMpy1(Factor, mpData[Channel], mSignalLength)
968     return 0.0
969 }
970
971 bool CMusicSignal::Set(unsigned long SampleRate, unsigned long NumSamples, int NumChannels, double**
pSignal)
972 {
973     int c
974     bool rc=true
975
976     Free()
977
978     NumSamples -= 0
979
980     for (c=0 c<NumChannels c++)
981     {
982         mpData[c] = (OTA_FLOAT*)matMalloc((NumSamples+2*128) * sizeof(OTA_FLOAT))
983         mpData[c+2] = (OTA_FLOAT*)matMalloc((NumSamples+2*128) * sizeof(OTA_FLOAT))
984         if (mpData[c])
985         {
986             int s
987             for (s=0 s<NumSamples s++)
988                 mpData[c][s] = pSignal[c][s+0]
989             for (s<NumSamples+2*128 s++)
990                 mpData[c][s] = 0
991             matbCopy(mpData[c], mpData[c+2], NumSamples+2*128)
992         }
993         else
994             rc = false
995     }
996
997     mSampleRate = SampleRate
998     mSignalLength = NumSamples+2*128
999     mNumChannels = NumChannels
1000
1001     if (!rc) Free()
1002
1003     return rc
1004 }
1005
1006 }

```

1007