GRANT AMES

43 P.

IN-17533

Final Report

to

NASA - AMES RESEARCH CENTER

Contract Award No. NASA NAG2-304

for

A SURVEY OF THE STATE OF THE ART AND FOCUSED RESEARCH IN RANGE SYSTEMS - TASK I

(NASA-CR-176900) A SURVEY OF THE STATE-OF-THE-ART AND FOCUSED RESEARCH IN RANGE SYSTEMS, TASK 1 Final Report (California Univ.) 43 p CSCL 17B

N86-29118

Unclas

G3/32 43448

Principal Investigator

Jim K. Omura, Professor

Electrical Engineering Department
University of California at Los Angeles
Los Angeles, California USA

July 1986

ABSTRACT

This final report presents our latest research activity in voice compression. We have designed a non-real time simulation system that is implemented around an IBM-PC where the IBM-PC is used as a speech work station for data acquisition and analysis of voice samples. A real-time implementation is also proposed. This real-time Voice Compression Board (VCB) is architected around the Texas Instruments TMS-3220.

The voice compression algorithm investigated here was described in an earlier report titled, "Low Cost Voice Compression for Mobile Digital Radios," by the author. We will assume the reader is familiar with the voice compression algorithm discussed in this report. The VCB compresses speech waveforms at data rates ranging from 4.8 K bps to 16 K bps. This board interfaces to the IBM-PC 8-bit bus, and plugs into a single expansion slot on the mother board.

SUMMARY OF RESEARCH

1. Speech Work Station

To provide an inexpensive speech work station, we selected the IBM-PC. With the large pool of software/hardware available for this personal computer, a completely self-contained speech work station was integrated.

Data acquisition hardware necessary for the speech work station must be composed of the following modules:

- .12-bit A/D module with minimum 20 microsecond conversion time
- .12-bit A/D module with minimum 20 microsecond settling time

The following features are required:

- .Programmable input gain: ranging 2,4,8
- .Single Differential input channel
- .Programmable sampling rate
- .Fully compatible with IBM Personal Computer

We decided to use the Data Translation (Marlborough, Massachusetts) DT-2801A, a single board analog and digital I/O system for the IBM-PC. This board contains all of the above modules and has 16 single-ended or 8 differential A/D input channels.

Software drivers for this board are available from numerous vendors. We decided to use the Interactive Laboratory System (ILS) software package for the IBM-PC. This software package is developed by the Signal Technology Corp. (Goleta, California). ILS performs numerous functions in a variety of areas including: signal display, spectral analysis, display and editing functions.

An analog interface circuitry was designed to interface a microphone to a single differential input channel for the A/D operation, and a speaker to a single D/A channel.

Voice compression experiments are performed by first sampling the input speech waveform. The speech samples are stored on the hard disk drive. A program was written to translate the ILS sampled data files to standard ascii character files. This file is then transferred via the serial RS-232 port on the IBM-PC to the Vax 11/750. Another program converts this ascii file to a binary packed format. The voice compression program reads this input file, and creates an output file which contains the synthesized compressed speech samples. This file is translated from the binary packed format to ascii format, and then transferred to the IBM-PC. This ascii file is translated into an ILS sample data file. The compressed speech may now be played on the speaker by performing D/A operation on this file.

In Fig. (1) we have shown this development environment.

We shall now discuss the experimental results that we have obtained by using this development station.

2. Description of Experiments

In the course of our experiments, we evaluated several different variations of the voice compression algorithm, namely,

1. Filter Coefficient Smoothing
2. Automatic Gain Control
3. Fixed Point Implementation
4. Parameter Quantization

Filter Coefficient Smoothing: In this version, the filter coefficients during each sampling period are smoothed by linear interpolation. This is done for extension filtering of the tree search algorithm, and the synthesis filter.

Automatic Gain Control: In this version, a gain factor is used to scale consecutive samples of the same value, i.e.,

$$g_i = \begin{cases} ca_{i-1}; & \text{if } ai = a_{i-1} \\ 1 \;; & \text{otherwise} \end{cases}$$

This gain is used in both extension filtering and synthesis filter.

Fixed Point Implementation: To implement the voice compression algorithm on a digital signal processor, it is necessary to perform all the arithmetic in the field of real integers modulo 2^{16}.

Parameter Quantization: To transmit the filter coefficients over the communication channel, it is necessary to quantize the filter coefficients (Total of 8) using a parameter quantization scheme as described in the previous report.

In the enclosed appendix we have included a set of experiments that we have performed. In Experiments 101 through 107, the speech waveforms for the duration of each utterance is displayed. Furthermore, the spectrum and the waveform for different segments of each utterance is also depicted.

3. Summary of Our Experiments

Filter smoothing does smooth the compressed voice signal. However, it degrades the intelligibility of high frequency, low amplitude (ex. nasal consonants) signals.

The compressed voice is extremely sensitive to the constant gain coefficients. A gain value of 1.2 resulted in the best performance. The vowels were slightly distorted using this technique.

The results of the Integer implementation of the voice compression algorithm were quite exciting. We found no loss in performance when 16-bit resolution with proper scaling was used.

Parameter quantization resulted in negligible loss of performance.

Based on our observations and the results of our experiments, one may conclude that from the waveform tracking viewpoint of the tree search algorithm, this algorithm performs quite well. The reconstructed speech waveform closely tracks the original speech waveform. There is, however, some tracking error for high frequency components (3 Khz and above). From the spectrum of the reconstructed speech, it is clear that the spectrum of the synthesized speech and original speech resemble each other. It is, however, difficult to extract the signal- to-noise ratio directly from the spectrum.[1]

From the intelligibility viewpoint, this voice compression algorithm performs quite well. And given our integer implementation of this algorithm, we have shown the viability of building a VCP using a digital signal processor with 16-bits of resolution.

[1] It would be more appropriate to compute spectrogram for these waveforms, rather than the three-dimensional spectrum. Unfortunately, we did not have this facility.

4. Real-Time Voice Compression Board

The primary candidates for the real-time processor were Fujitsu MB-8764 and Texas Instruments TMS-32020. MB-8764 was rejected, based on a comparison of the instruction set and the available support for this processor.

In Fig. (2) we show the architecture for a real-time voice compression processor. A standard IBM-PC wire wrap board, which plugs into the IBM-PC bus is used to build the prototype for the VCB.

At initialization time all the programs are down-loaded via the IBM-PC bus into the global two-port memory. Each processor then copies the proper segment of each task into its local RAM.

Based on the computational complexity of the voice compression algorithm, it has been determined that to implement the voice compression algorithm in real-time, it requires at least two TMS-32020 to execute the algorithm. For this reason the algorithm is broken down in two segments. The analysis filter processor performs the withening operation and outputs the residue sequence. And the filter coefficients are computed. The second processor performs all the tree filtering algorithm and the reconstruction filter.

All task synchronizations are performed via mailboxes resident in the global memory. The second processor lags behind the first processor by a full frame cycle (10 msec). The operation of the two processors is completely overlapped.

All of the results from the analysis filter are copied from the local RAM to the global RAM. The second processor reads this block of data from the global RAM and copies it into its local RAM. The output of the second processor is binary-packed data, and quantized parameters. These results are accessible from the PC-Bus. The host processor (Intel 8088) may access these results from the PC-Bus and transmit them through the serial RS-232C port, which can be transmitted via a modem to the circuit- switched telephone lines.

The serial bus on the first TMS-32020 is used for A/D operation, and the serial bus on the second processor is used for the D/A operation. This off-loads the processor memory bandwidth.

The real-time implementation of the voice compression algorithm was outlined. Some experimental results of the variation of the voice compression algorithm were stated. These results showed the tracking performance of the voice compression algorithm.

This report confirms the feasibility of building a VCB using two TMS-32020. This board may be used for low-cost digital mobile voice terminal which can interface to the telephone line or other transmission mediums.

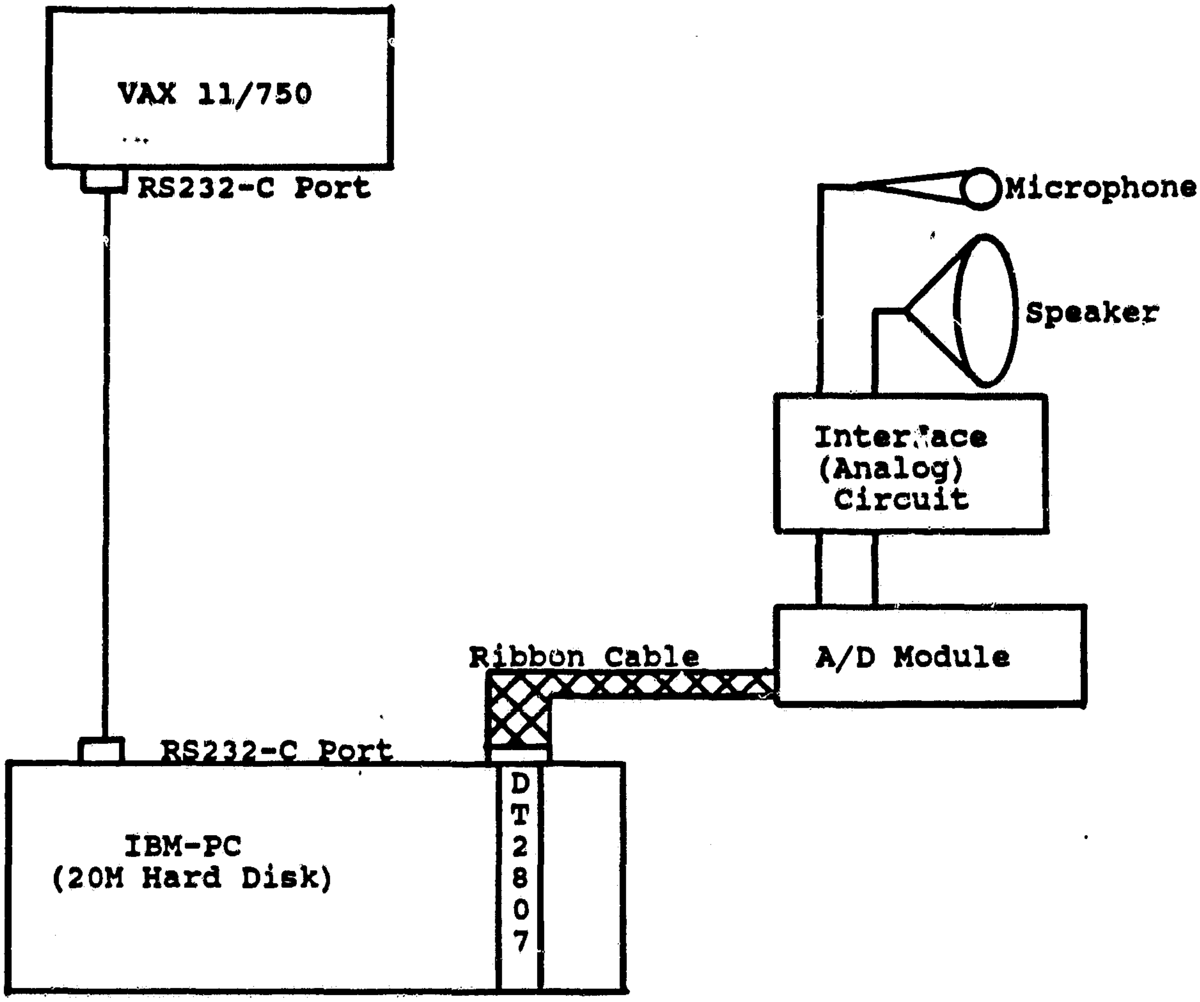
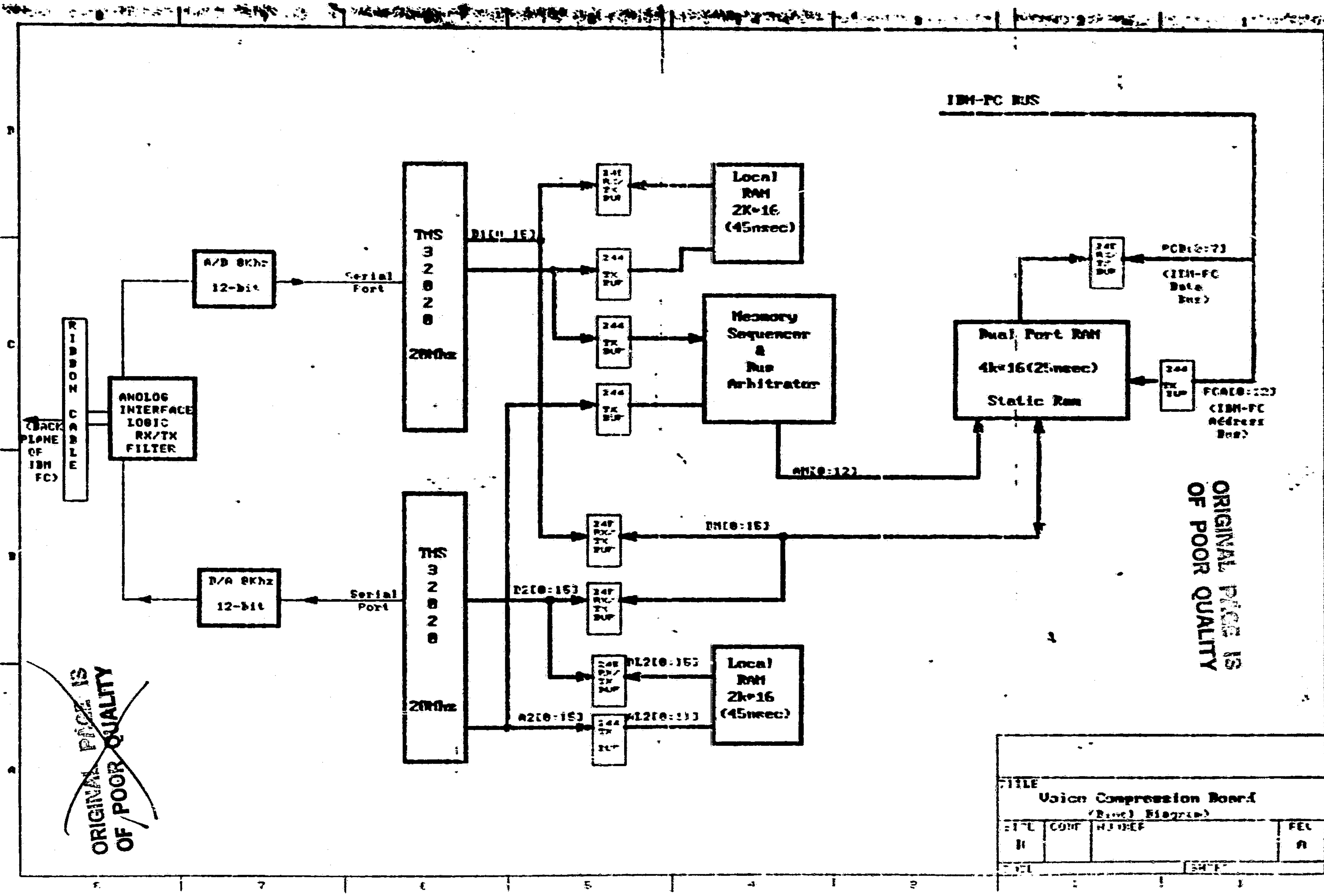


Figure 1. Speech Development Workstation



ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

EXPERIMENT # : 100

Speaker: Ramin Sadr

Time: 12: P.M

Date: 9-16-85

Description: The minimal tree algorithm was modified *1 for the Vax, and it was used to compress 11 *2 sec of speech. The end frame on the P.C. (64 pts/Frame) is 1500.

Source File: ns-20g (Ex552)

Output File: ns-2.cmp. (EX551)

Object File: Stree Sept 15, 1985

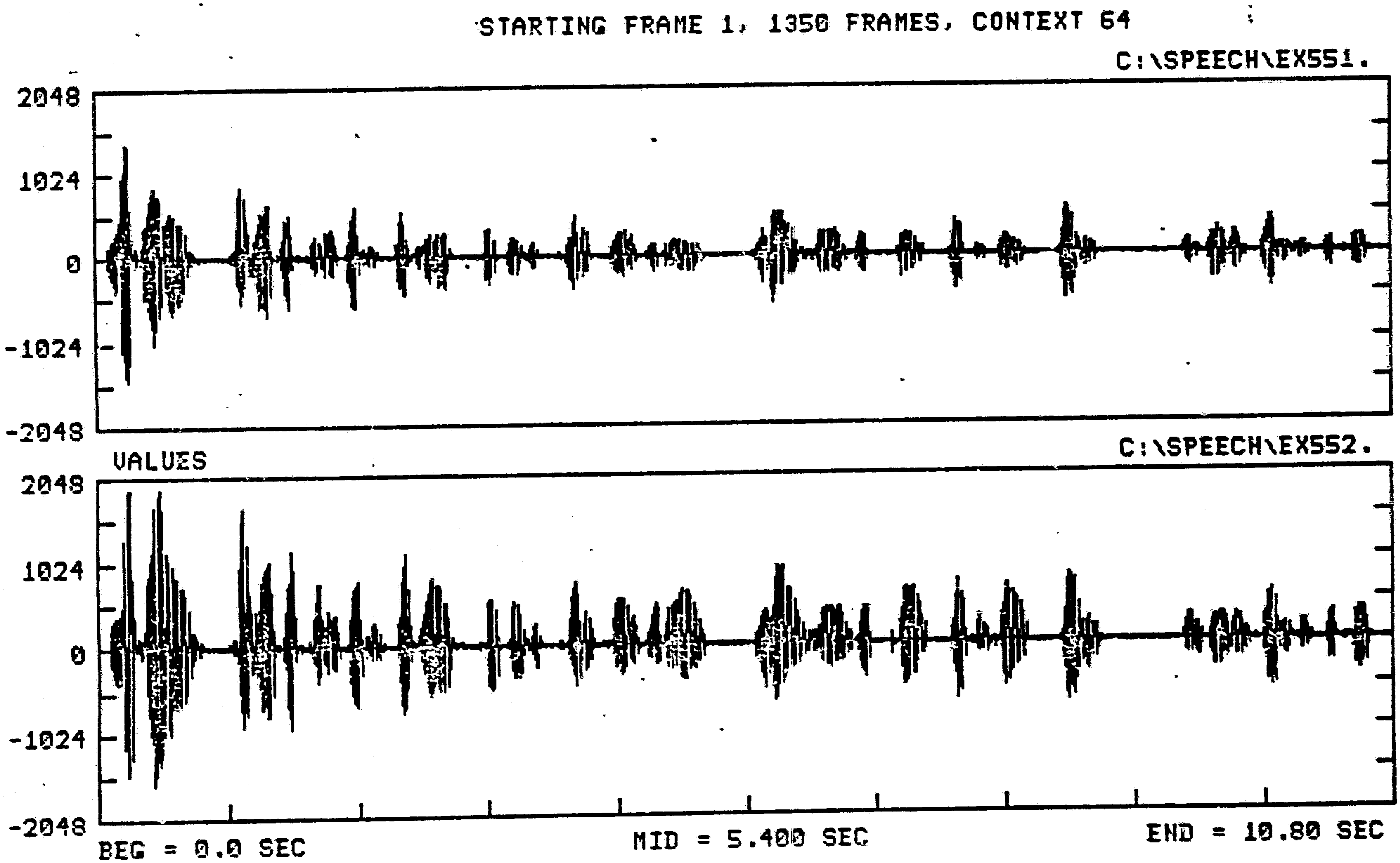
Conclusion & Observation:

The synthesized Waveform amplitude did NOT track the absolute magnitude. However, the overall relative magnitude seemed to be correct, especially during the voiced segments.

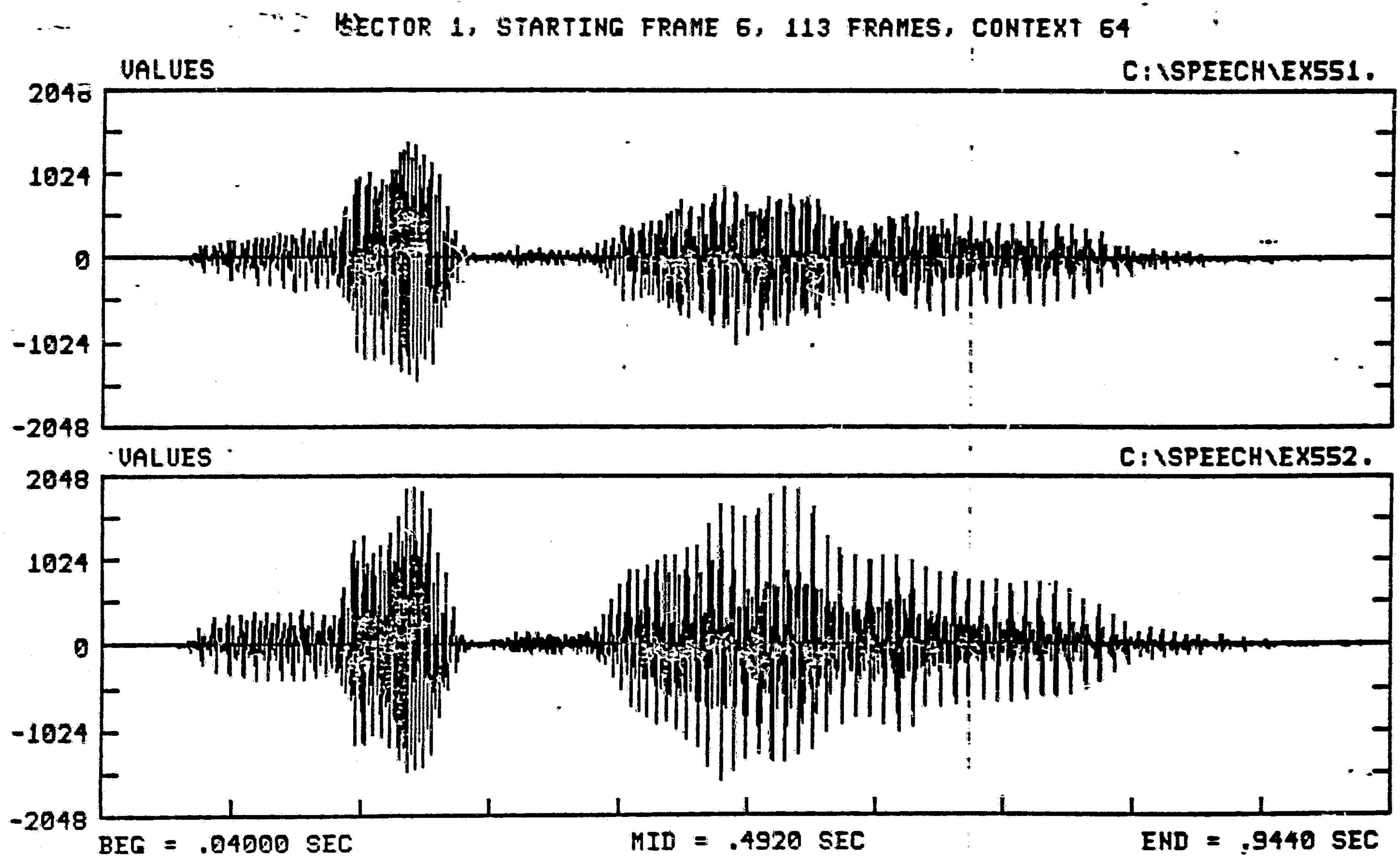
The spectrum (Log magnitude, SOZ LM,, 204) seems to match pretty well. But it is difficult to distinguish the formant frequencies.

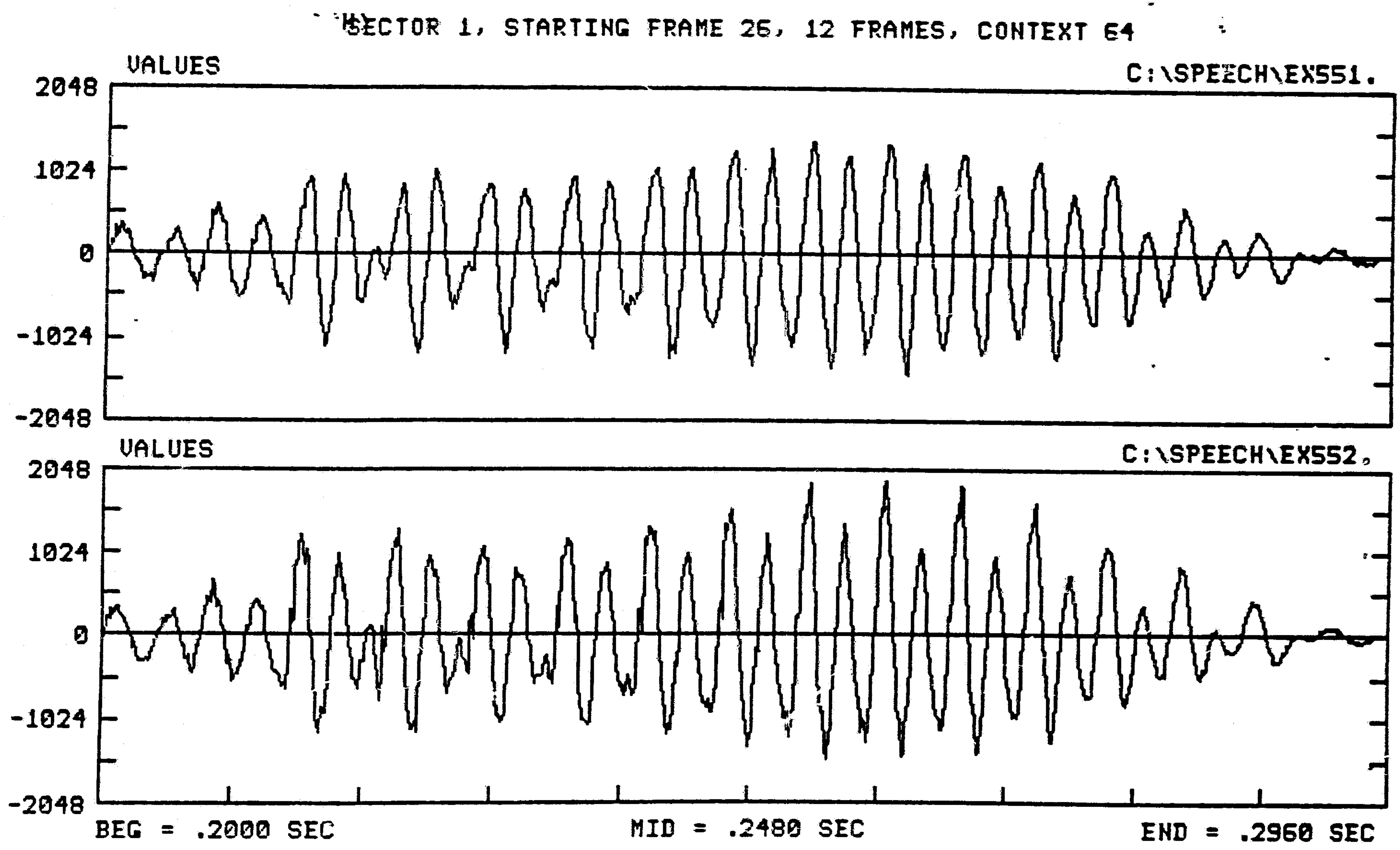
*1 modified I/O for Ascii printable 8d for easy transfer Vax to P.C

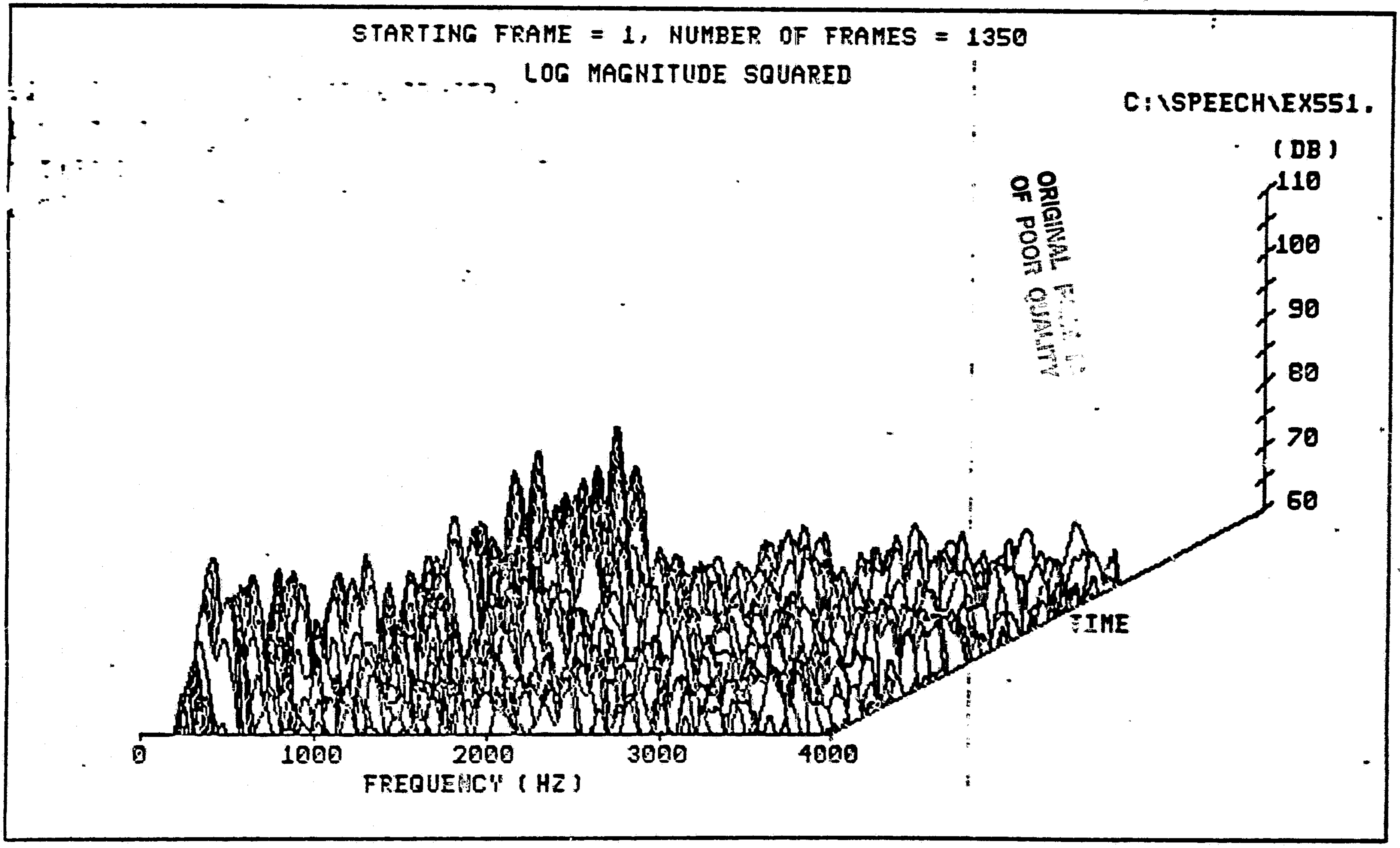
*2 Originally - it is 16 seconds, but only 12 second is read in at a time.


STARTING FRAME 1, 1350 FRAMES, CONTEXT 64
C:\SPEECH\EX551.
2048
1024
0
-1024
-2048
VALUES
C:\SPEECH\EX552.
2048
1024
0
-1024
-2048
BEG = 0.0 SEC
MID = 5.400 SEC
END = 10.80 SEC

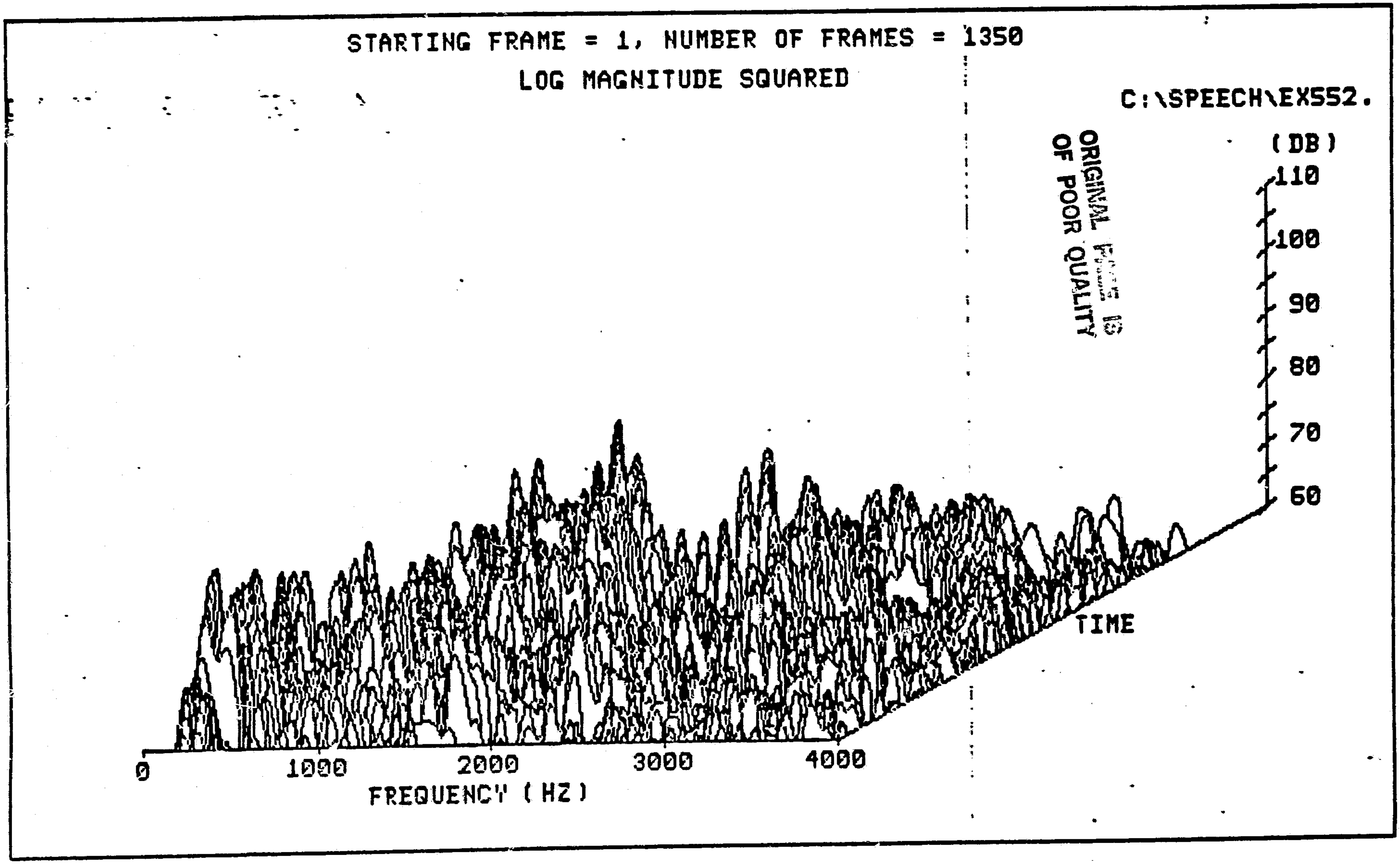
Expcri: 100


SECTOR 1, STARTING FRAME 6, 113 FRAMES, CONTEXT 64
VALUES
C:\SPEECH\EX551.
2048
1024
0
-1024
-2048
VALUES
C:\SPEECH\EX552.
2048
1024
0
-1024
-2048
BEG = .04000 SEC
MID = .4920 SEC
END = .9440 SEC


SECTOR 1, STARTING FRAME 26, 12 FRAMES, CONTEXT 64
VALUES
C:\SPEECH\EX551.
2048
1024
0
-1024
-2048
VALUES
C:\SPEECH\EX552.
2048
1024
0
-1024
-2048
BEG = .2000 SEC
MID = .2480 SEC
END = .2960 SEC


STARTING FRAME = 1, NUMBER OF FRAMES = 1350
LOG MAGNITUDE SQUARED
C:\SPEECH\EX551.
(DB)
110
100
90
80
70
60
TIME
0
1000
2000
3000
4000
FREQUENCY (HZ)
ORIGINAL
OF POOR QUALITY

Exp # 100


STARTING FRAME = 1, NUMBER OF FRAMES = 1350
LOG MAGNITUDE SQUARED
C:\SPEECH\EX552.
(DB)
110
100
90
80
70
60
TIME
0
1000
2000
3000
4000
FREQUENCY (HZ)

ORIGINAL PAGE IS
OF POOR QUALITY

EXPERIMENT # : 101

Speaker: Ramin Sadr

Time: 2:05pm

Date: 9-18-85

Description: The smoothed Tree algorithm:
convex-addition - computation of filter coefficients, RMS value, and normalization factor.

was modified for the vax

Source File: rS-2.org

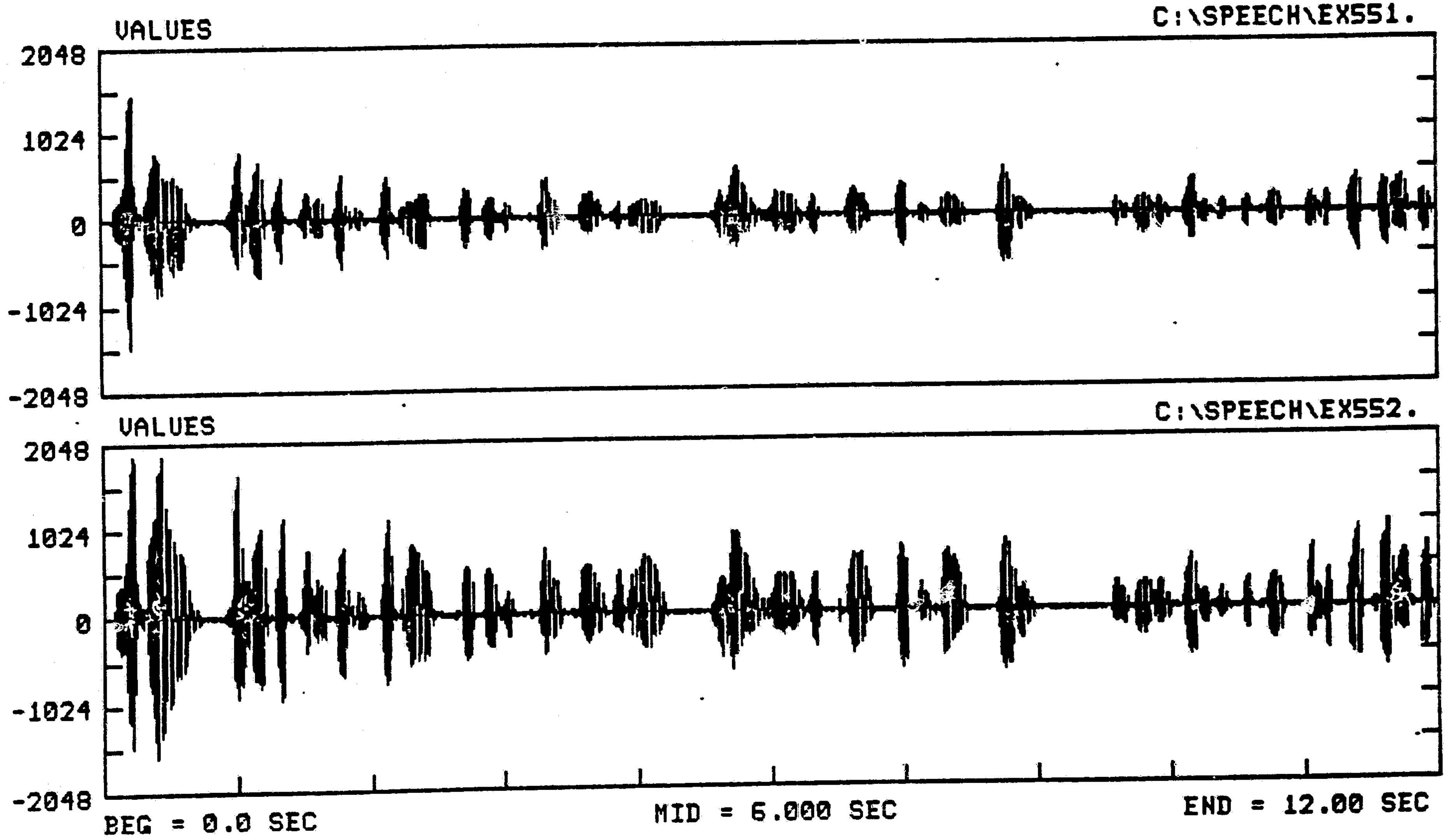
Output File: rS-2-3. cmp

Object File: Smgtree

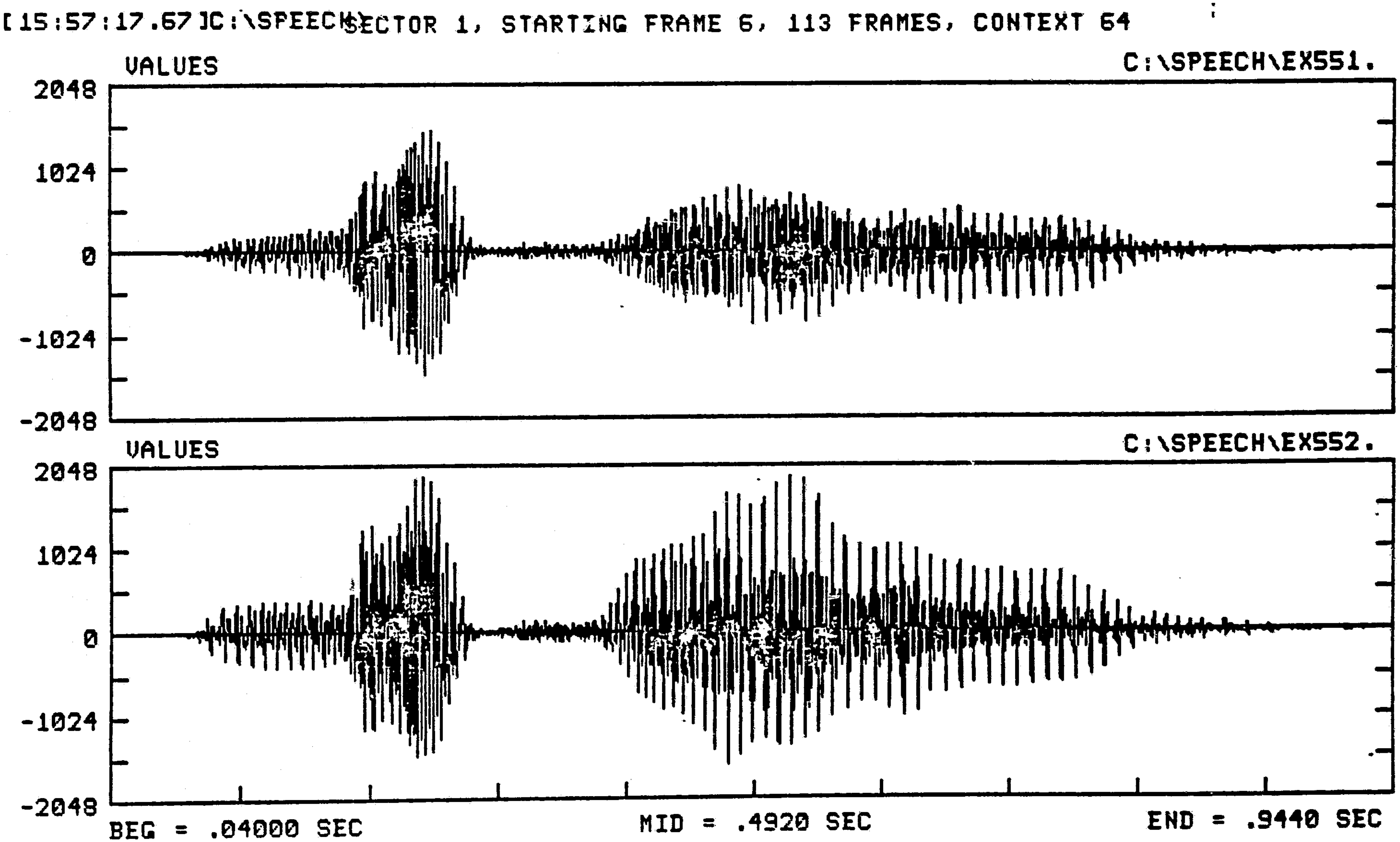
Conclusion & Observation:

No Significant improvement was observed. It was expected to see a smoother waveform which was not the case. Even, with the additional cost of Processing!

[15:48:35.49]C:\SPEECH\SECTOR 1, STARTING FRAME 1, 1500 FRAMES, CONTEXT 64

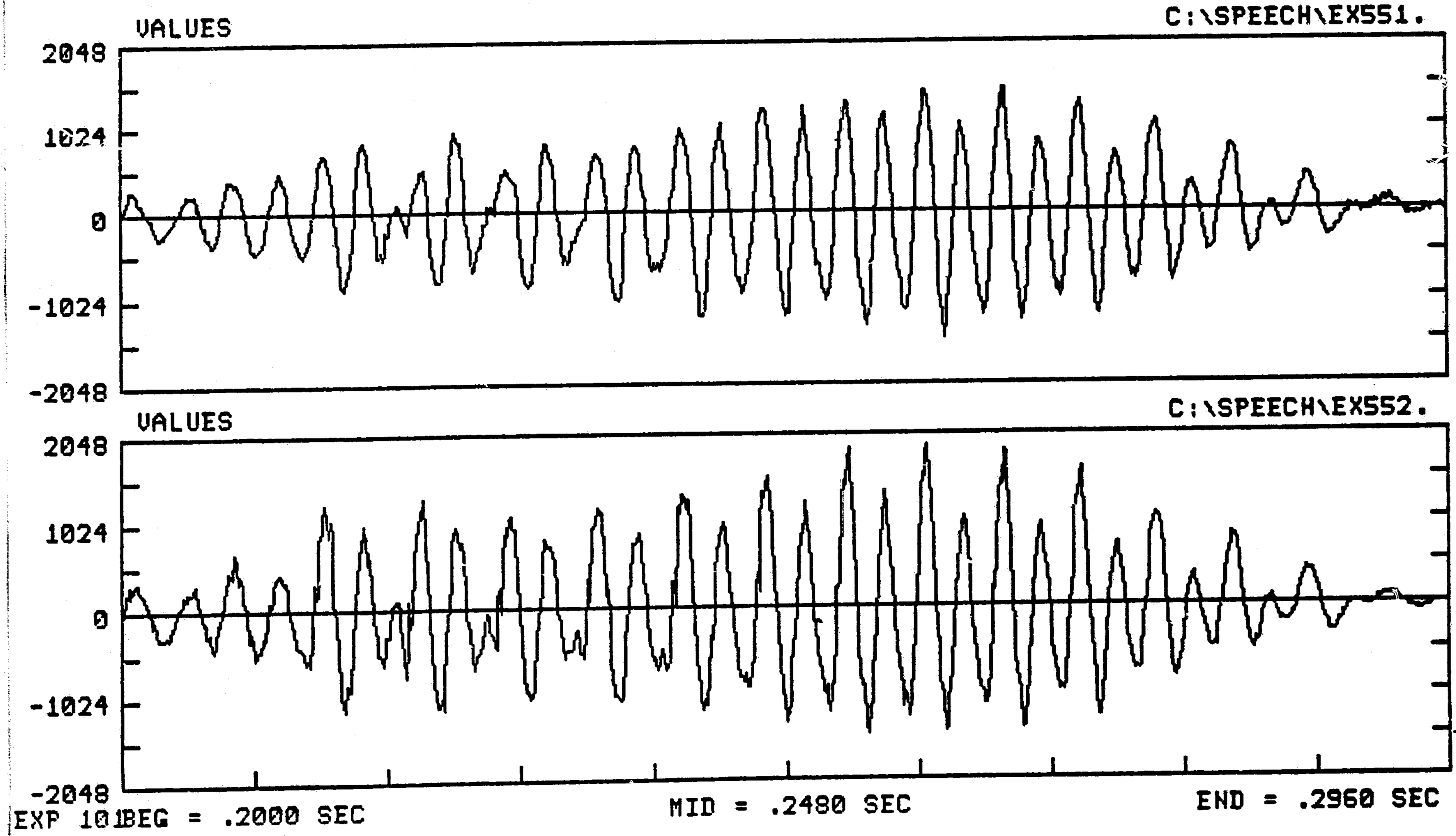


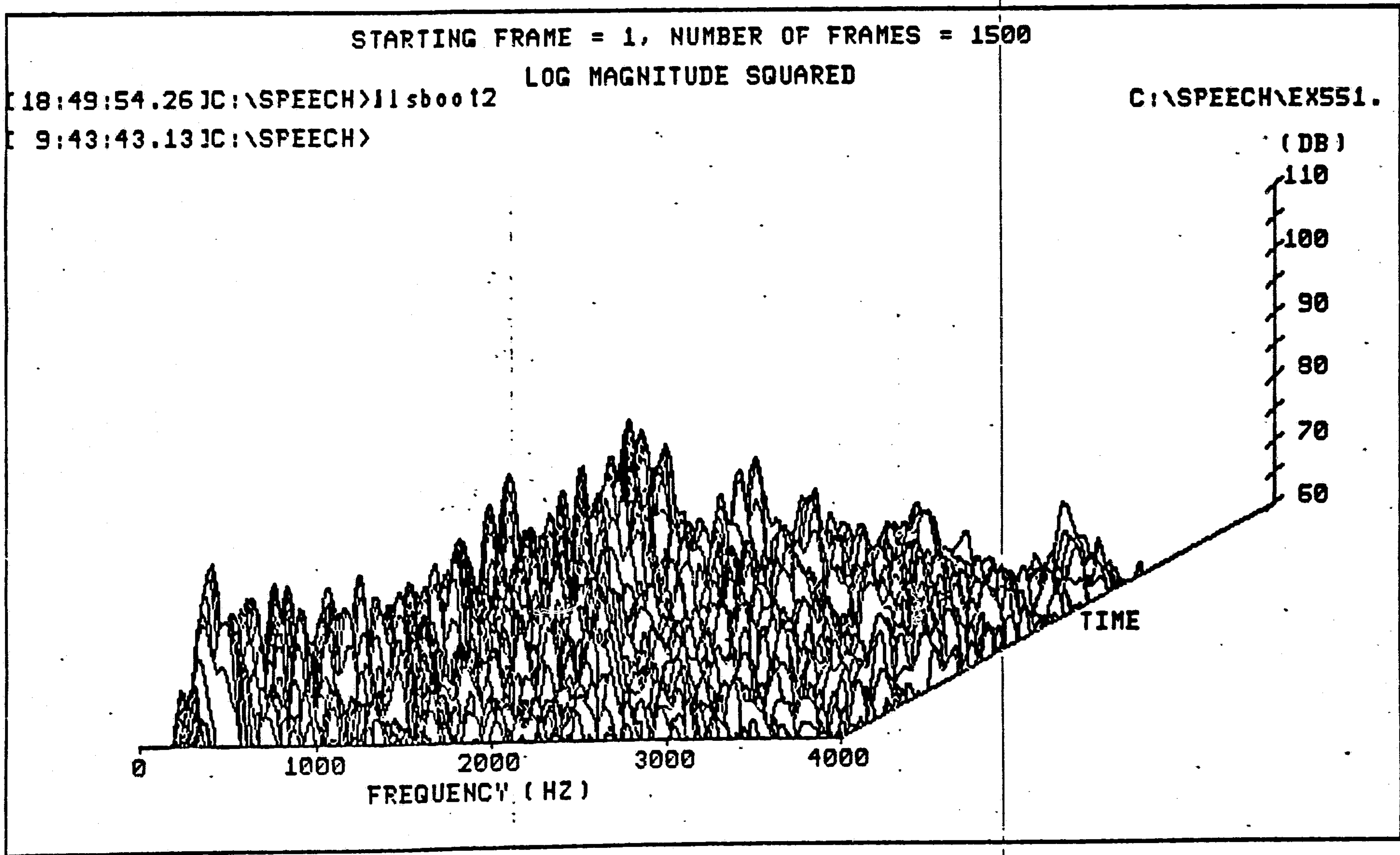
#101


[15:57:17.67]C:\SPEECH\SECTOR 1, STARTING FRAME 6, 113 FRAMES, CONTEXT 64
VALUES
C:\SPEECH\EX551.
2048
1024
0
-1024
-2048
VALUES
C:\SPEECH\EX552.
2048
1024
0
-1024
-2048
BEG = .04000 SEC
MID = .4920 SEC
END = .9440 SEC

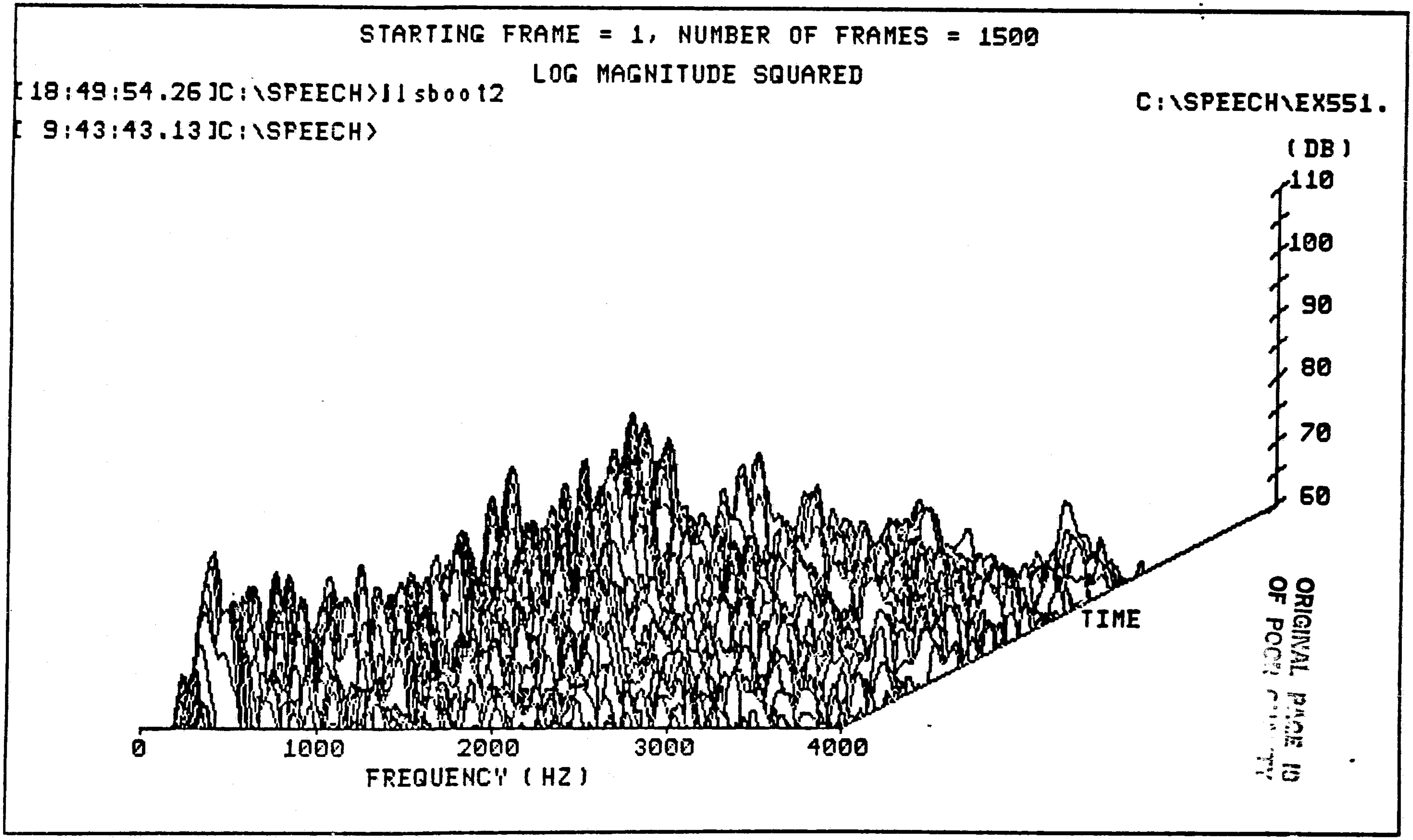
#101

C>
C>trm m,;Exp 101 SECTOR 1, STARTING FRAME 26, 12 FRAMES, CONTEXT 64




STARTING FRAME = 1, NUMBER OF FRAMES = 1500
LOG MAGNITUDE SQUARED
[18:49:54.26]C:\SPEECH>llsboot2
[9:43:43.13]C:\SPEECH>
C:\SPEECH\EX551.
(DB)
110
100
90
80
70
60
TIME
0
1000
2000
3000
4000
FREQUENCY (HZ)

Exp 101



ORIGINAL PAGE IS OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

EXPERIMENT # : 102

Speaker: Paul Pearson

Time: 5pm

Date: 8-18-85

Description: 16 Second Speech was shipped over to Vax. The tree search Algorithm was used to ~~this~~ compress the speech.

Source File: PP_1.org

Output File: PP_1.cmp , PP_1_6_2.cmp (D/A gain set to 2)

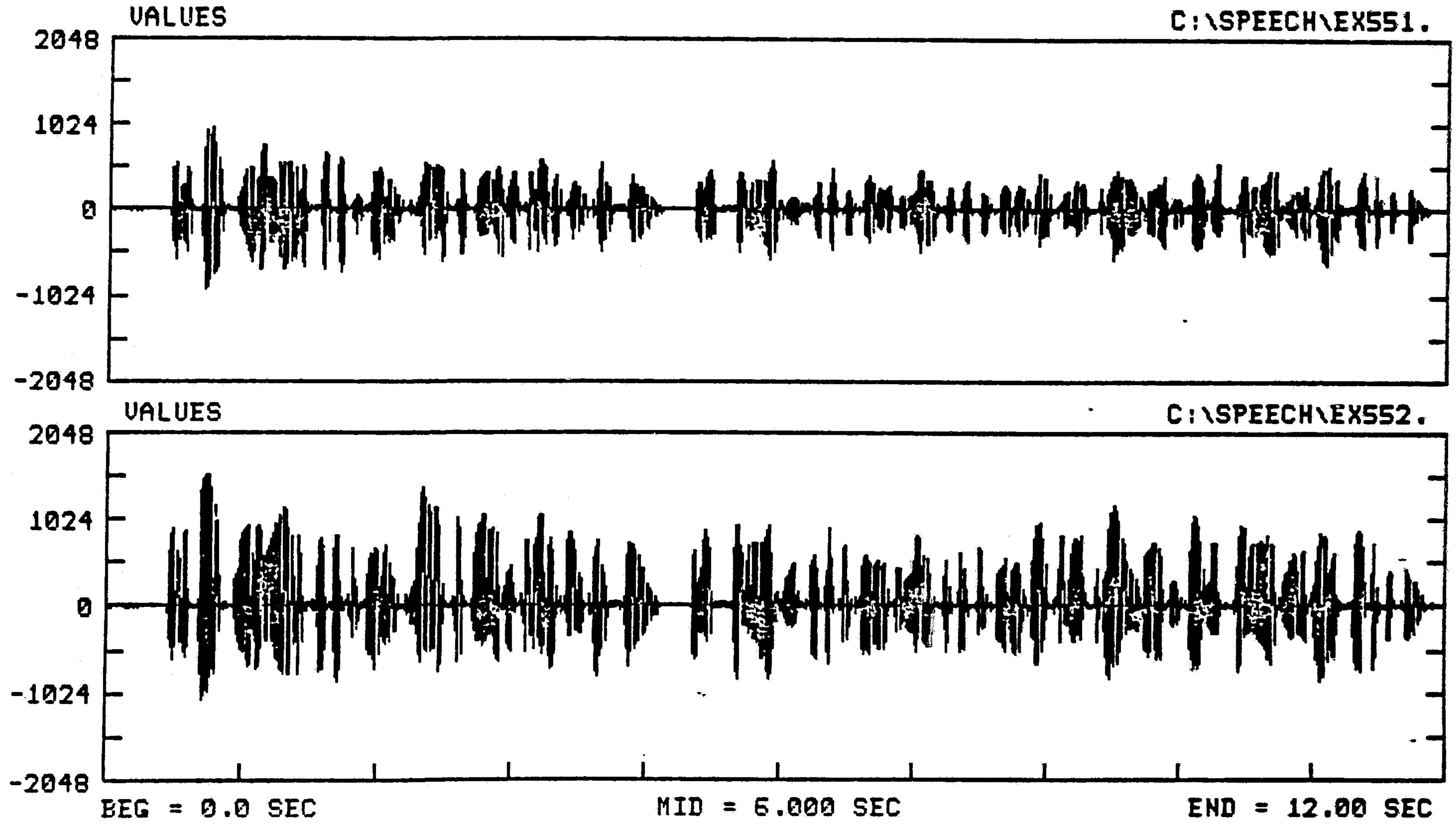
Object File: 9tree

Conclusion & Observation:

The speech sounded fairly well. The electronic accent was clearly there the magnitude of the speech was too low. I tried the D/A with gain 2. and showed the wave form for this (the utterance for this is "In the fast moving World – 61 → 228). In the following diagrams this is referred to as Experiment 2b.

Frame   Frame

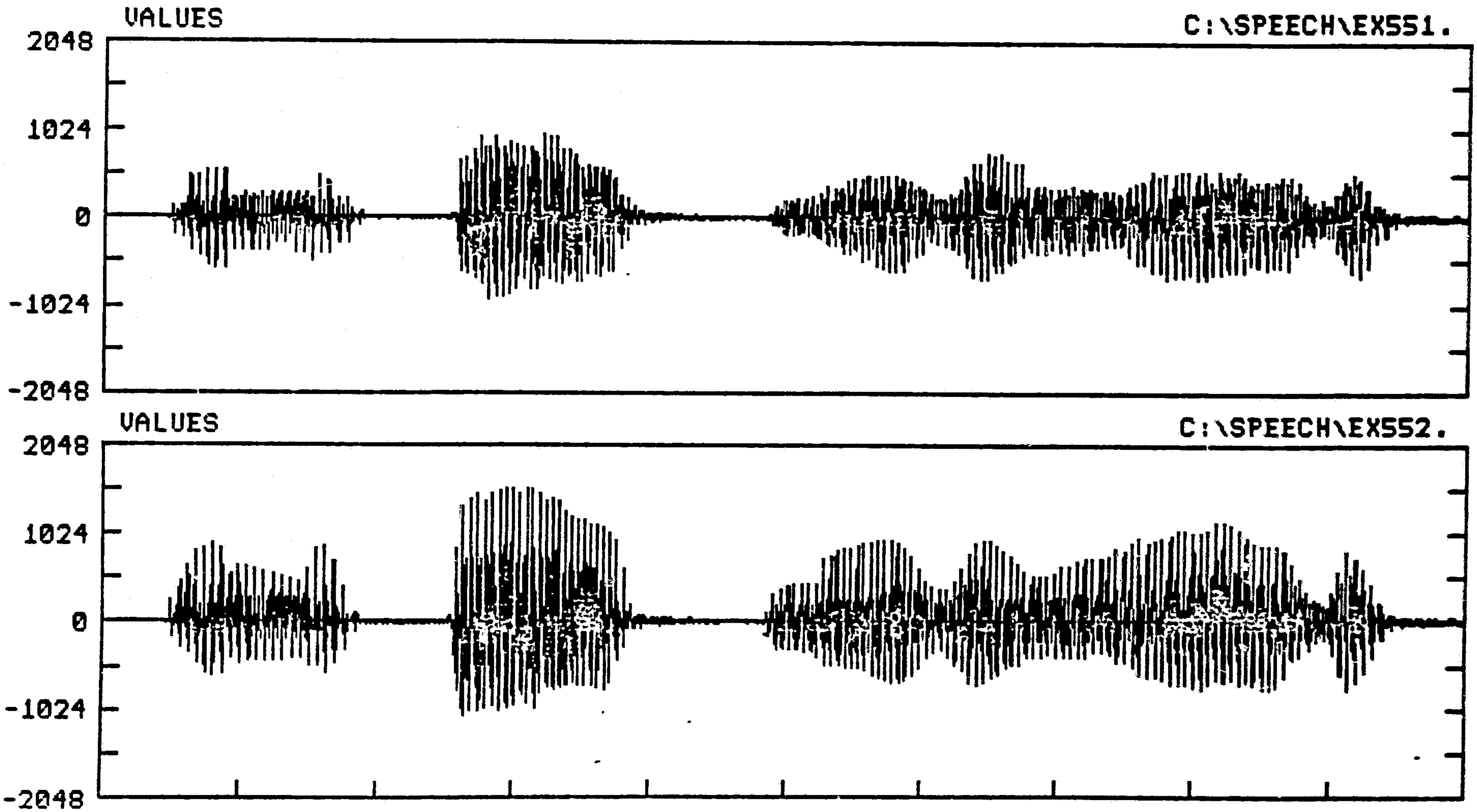
[11:31:35.05]C:\SPEECH SECTOR 1, STARTING FRAME 1, 1500 FRAMES, CONTEXT 64



Eyrer: 102

C>

SECTOR 1, STARTING FRAME 61, 167 FRAMES, CONTEXT 64

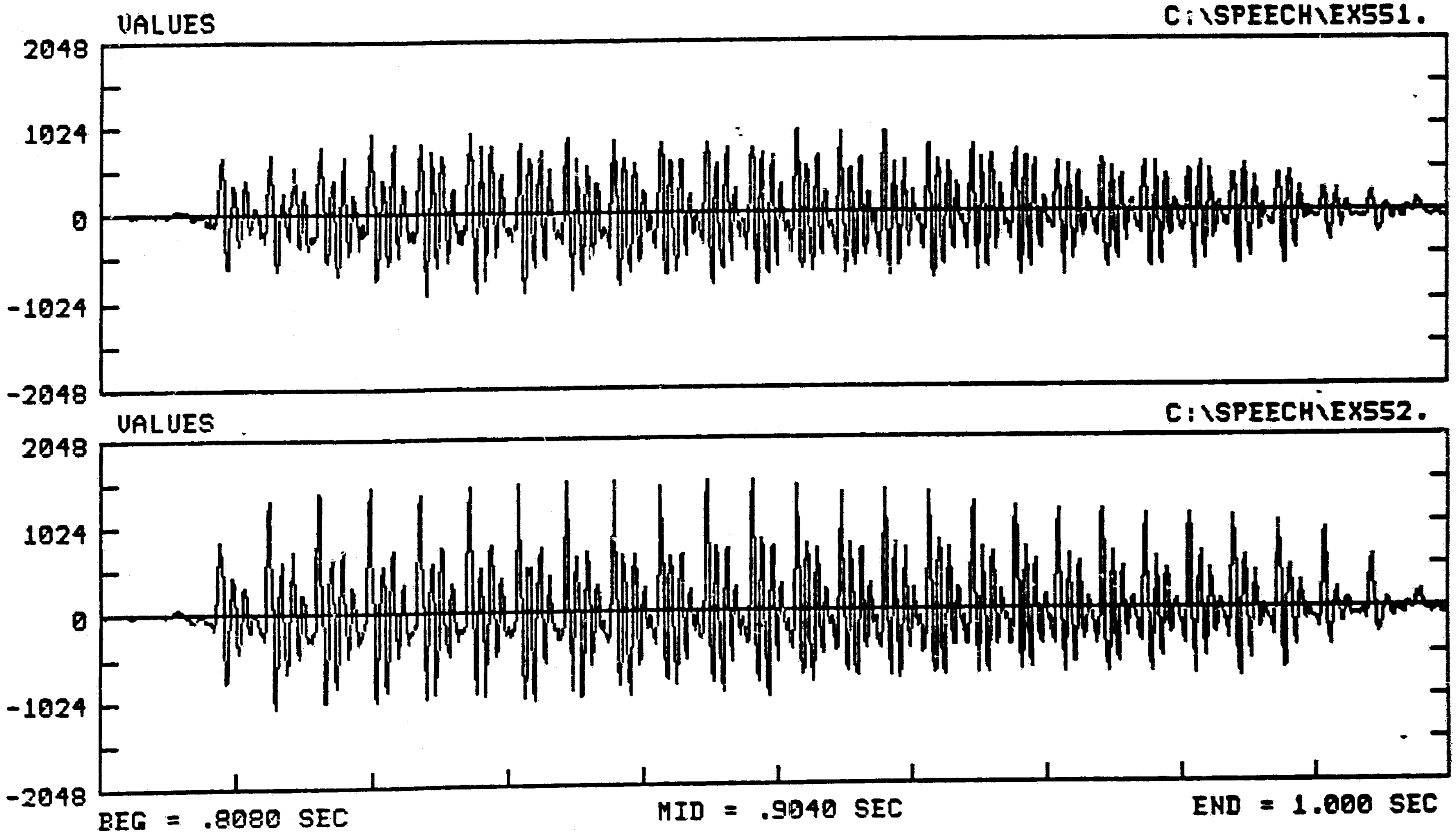


BEG = .4800 SEC MID = 1.148 SEC END = 1.816 SEC

102

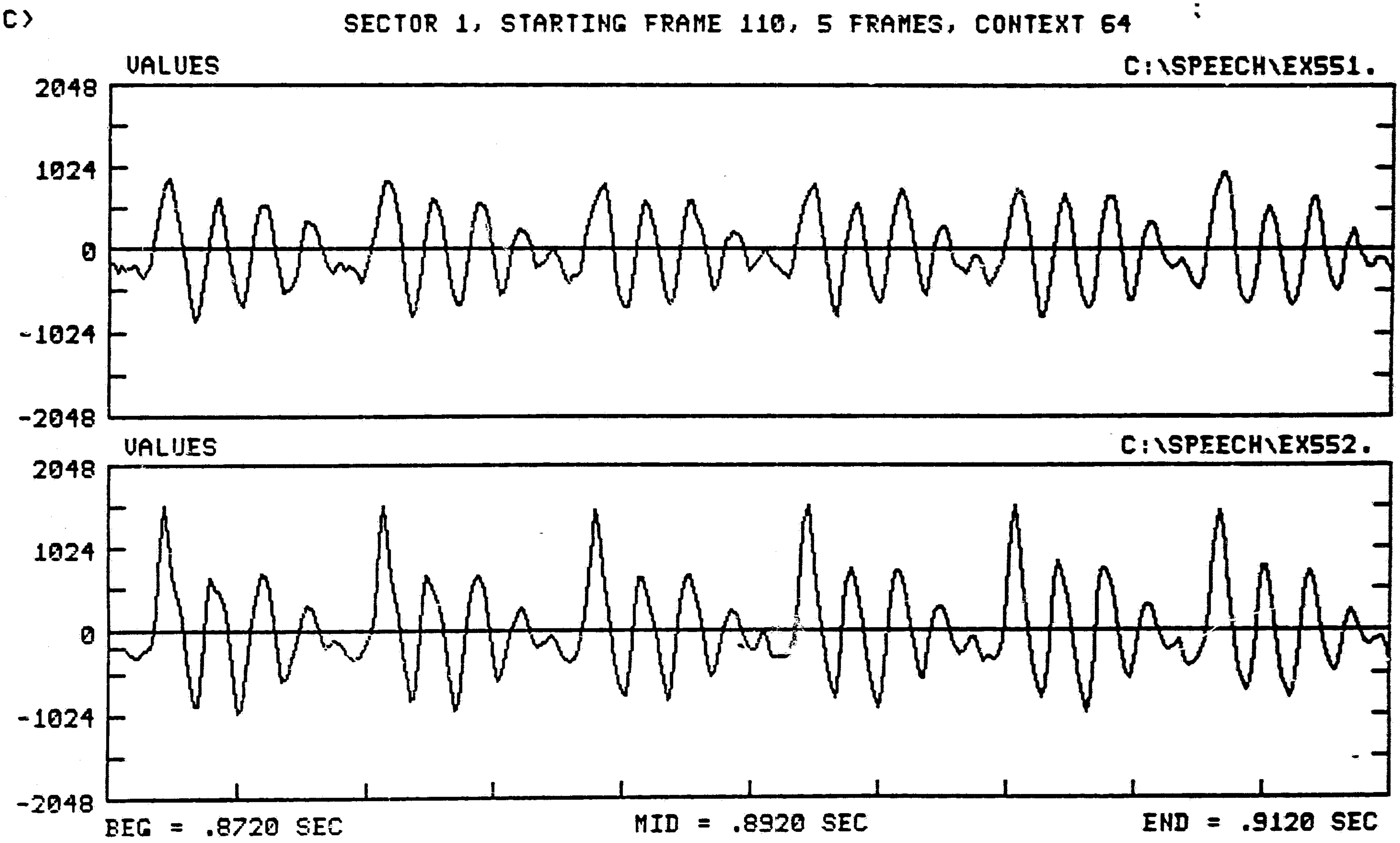
C>

SECTOR 1, STARTING FRAME 102, 24 FRAMES, CONTEXT 64



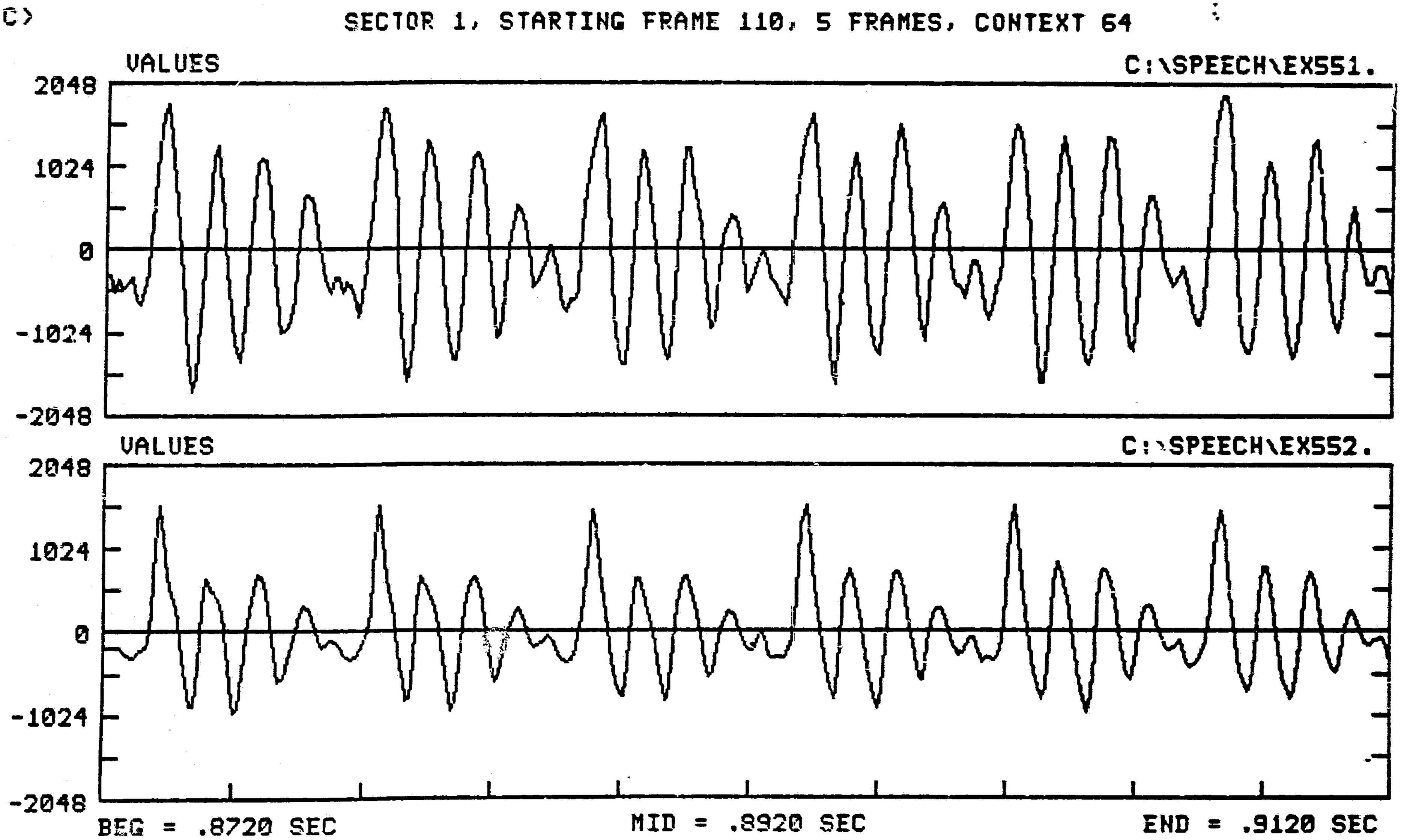
#102

C>


SECTOR 1, STARTING FRAME 110, 5 FRAMES, CONTEXT 64
VALUES
C:\SPEECH\EX551.
2048
1024
0
-1024
-2048
VALUES
C:\SPEECH\EX552.
2048
1024
0
-1024
-2048
BEG = .8720 SEC
MID = .8920 SEC
END = .9120 SEC

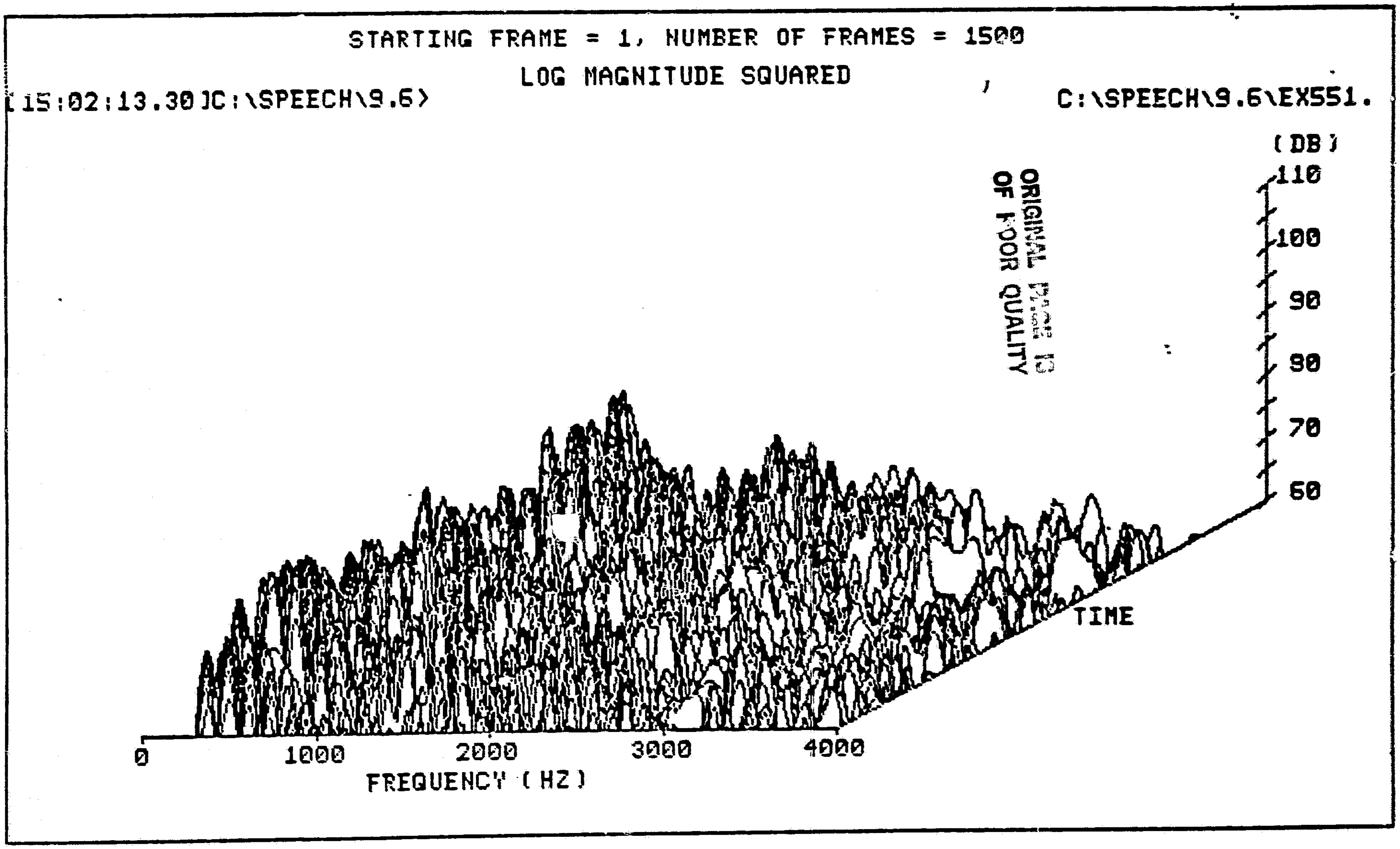
#102

C>

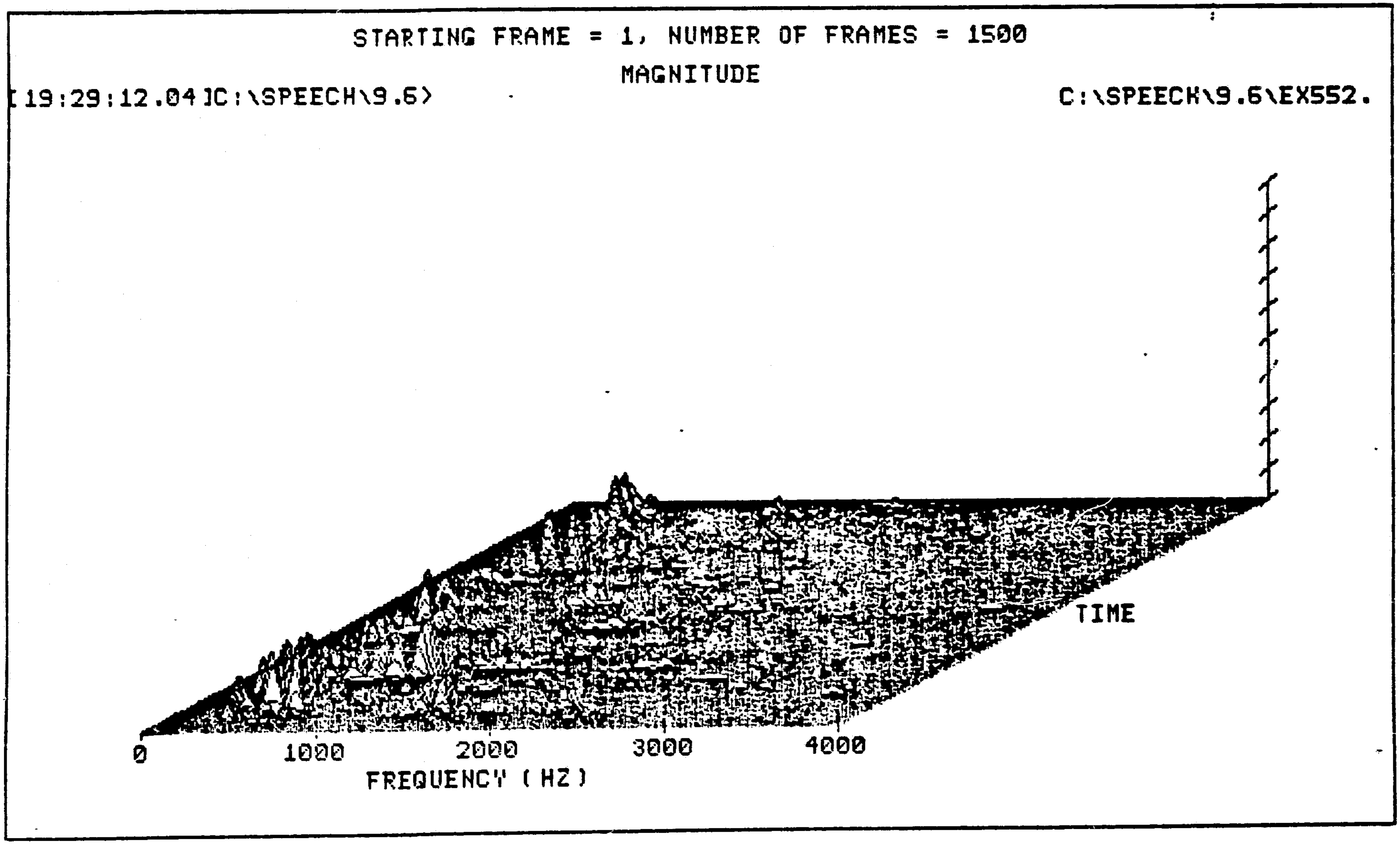


D/A

Exp #102 gain = 2


STARTING FRAME = 1, NUMBER OF FRAMES = 1500
LOG MAGNITUDE SQUARED
[15:02:13.30]C:\SPEECH\9.6>
C:\SPEECH\9.6\EX551.
(DB)
110
100
90
80
70
60
TIME
0
1000
2000
3000
4000
FREQUENCY (HZ)

+ /02


STARTING FRAME = 1, NUMBER OF FRAMES = 1500
MAGNITUDE
[19:29:12.04]C:\SPEECH\9.6>
C:\SPEECH\9.6\EX552.
TIME
0
1000
2000
3000
4000
FREQUENCY (HZ)

#162

EXPERIMENT # : 103

Speaker: Paul Pearson

Time: 10:42 AM

Date: 9-20-85

Description: Smoothed Tree algorithm was used to compress Paul's Utterence. The original utterence is 16 seconds long.

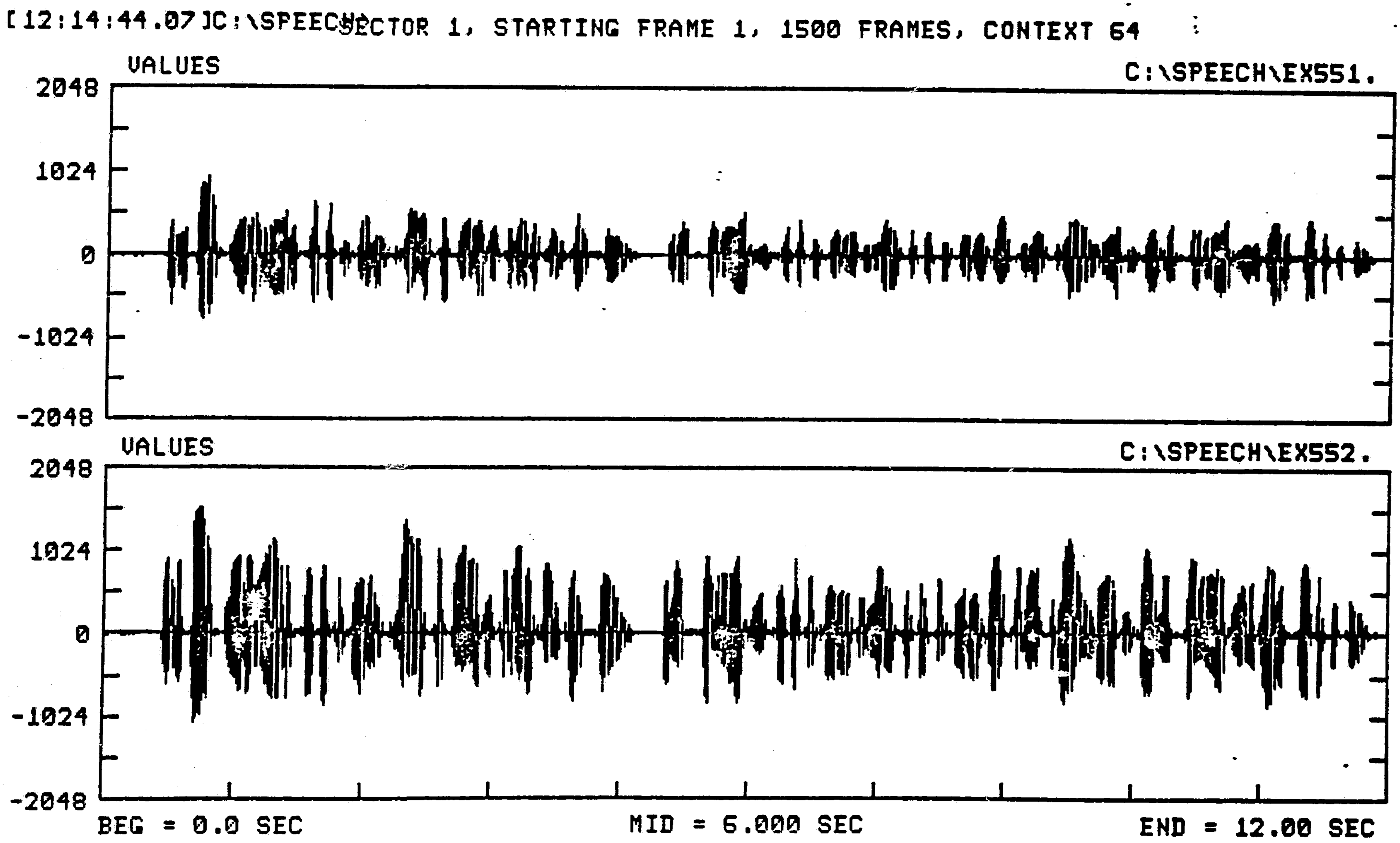
Source File: PP_1.Org

Output File: PP_1_S.cmp

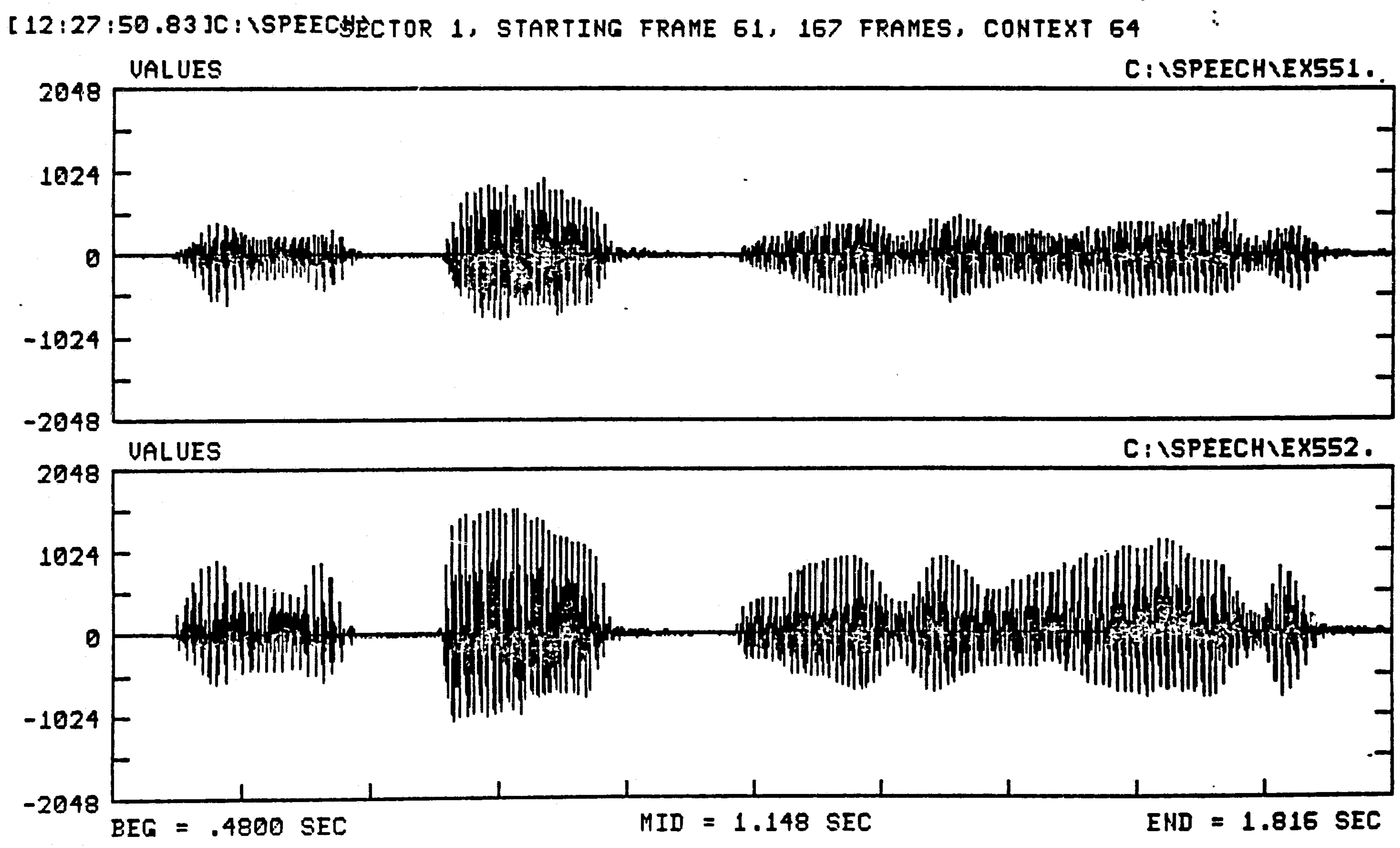
Object File: SmTree

Conclusion & Observation:

The resulting wave form is certainly smoother than the bare Tree algorithm.


[12:14:44.07]C:\SPEECH\SECTOR 1, STARTING FRAME 1, 1500 FRAMES, CONTEXT 64
VALUES
C:\SPEECH\EX551.
2048
1024
0
-1024
-2048
VALUES
C:\SPEECH\EX552.
2048
1024
0
-1024
-2048
BEG = 0.0 SEC
MID = 6.000 SEC
END = 12.00 SEC

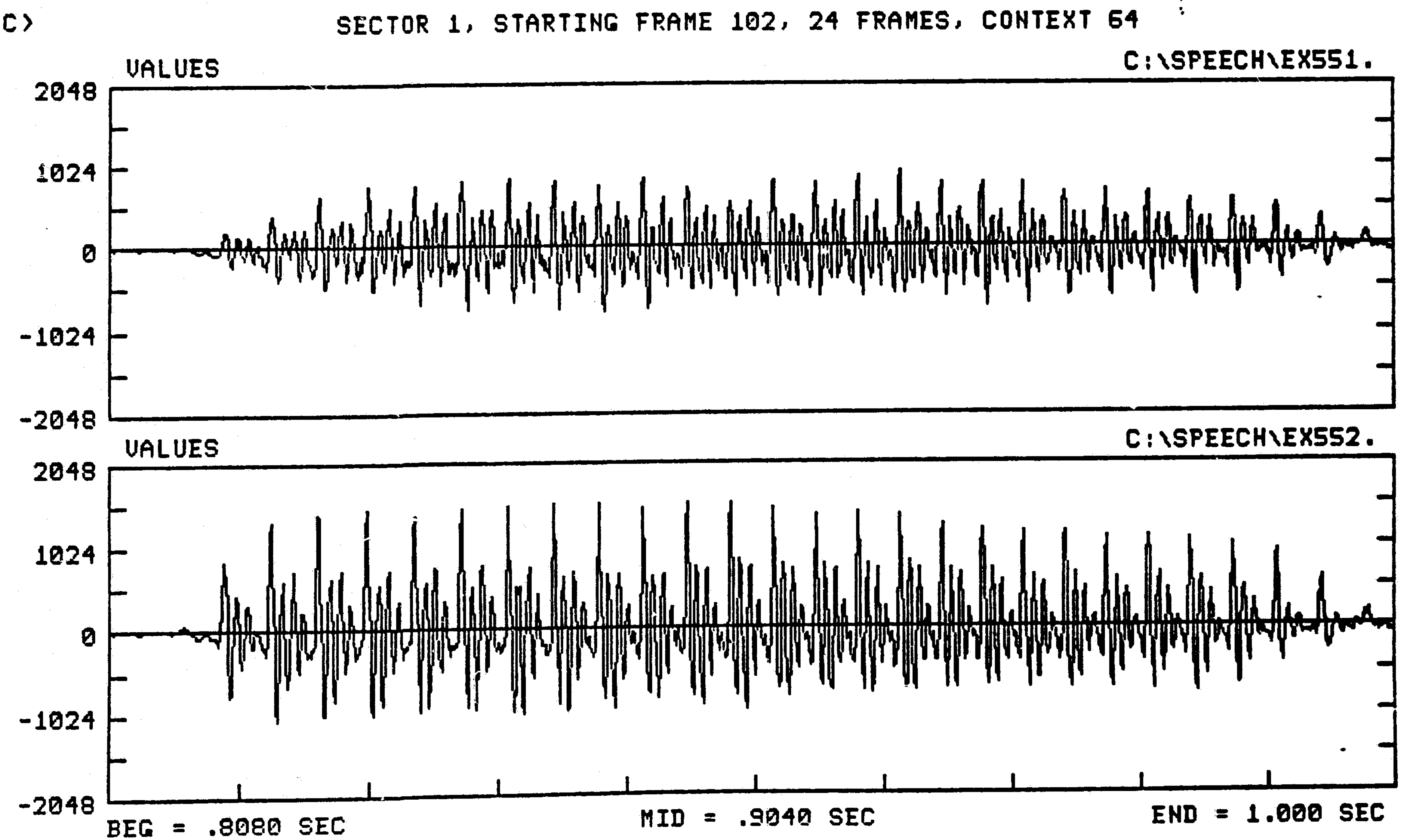
Exp 103



[12:27:50.83]C:\SPEECH\SECTOR 1, STARTING FRAME 61, 167 FRAMES, CONTEXT 64
VALUES
C:\SPEECH\EX551.
2048
1024
0
-1024
-2048
VALUES
C:\SPEECH\EX552.
2048
1024
0
-1024
-2048
BEG = .4800 SEC
MID = 1.148 SEC
END = 1.816 SEC

Exper 103

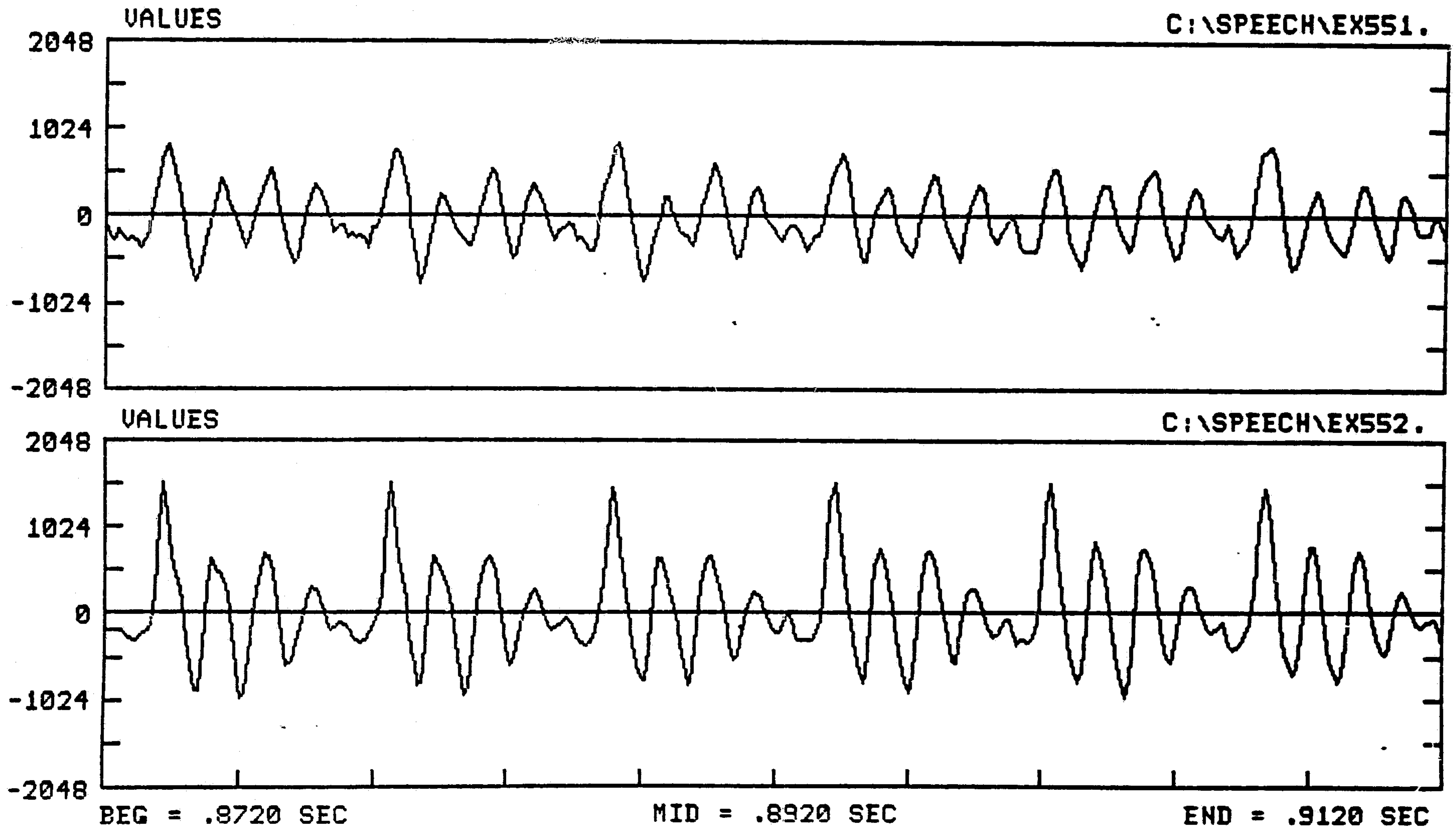
C>



Exp 103 'fast'

C>

SECTOR 1, STARTING FRAME 110, 5 FRAMES, CONTEXT 64



Exp

EXPERIMENT # : 104

Speaker:

Time: 10 am

Date: 9-23-85

Description: The smoothed tree algorithm, truncation lenght was increased to 32

Source File:

Output File:

Object File:

Conclusion & Observation:

EXPERIMENT # : 105

Speaker: Sandy

Time: 10:AM

Date: 10-10-85

Description: Compressed at 9.6K for 12.5 sec speech tried the L=32. A New Interface for M-preamp and A/D-D/A was wire-wrapped and tested. This board uses the Harris 5512 Switched capacitor filter for Receive & Transmitt filter.

Source File: Ex#105.Org , Stree.c

Output File: Ex#105.cmp , Ex#105_S.cmp (smoothed), Ex105S4.cmp (Smoothed with gain=4)

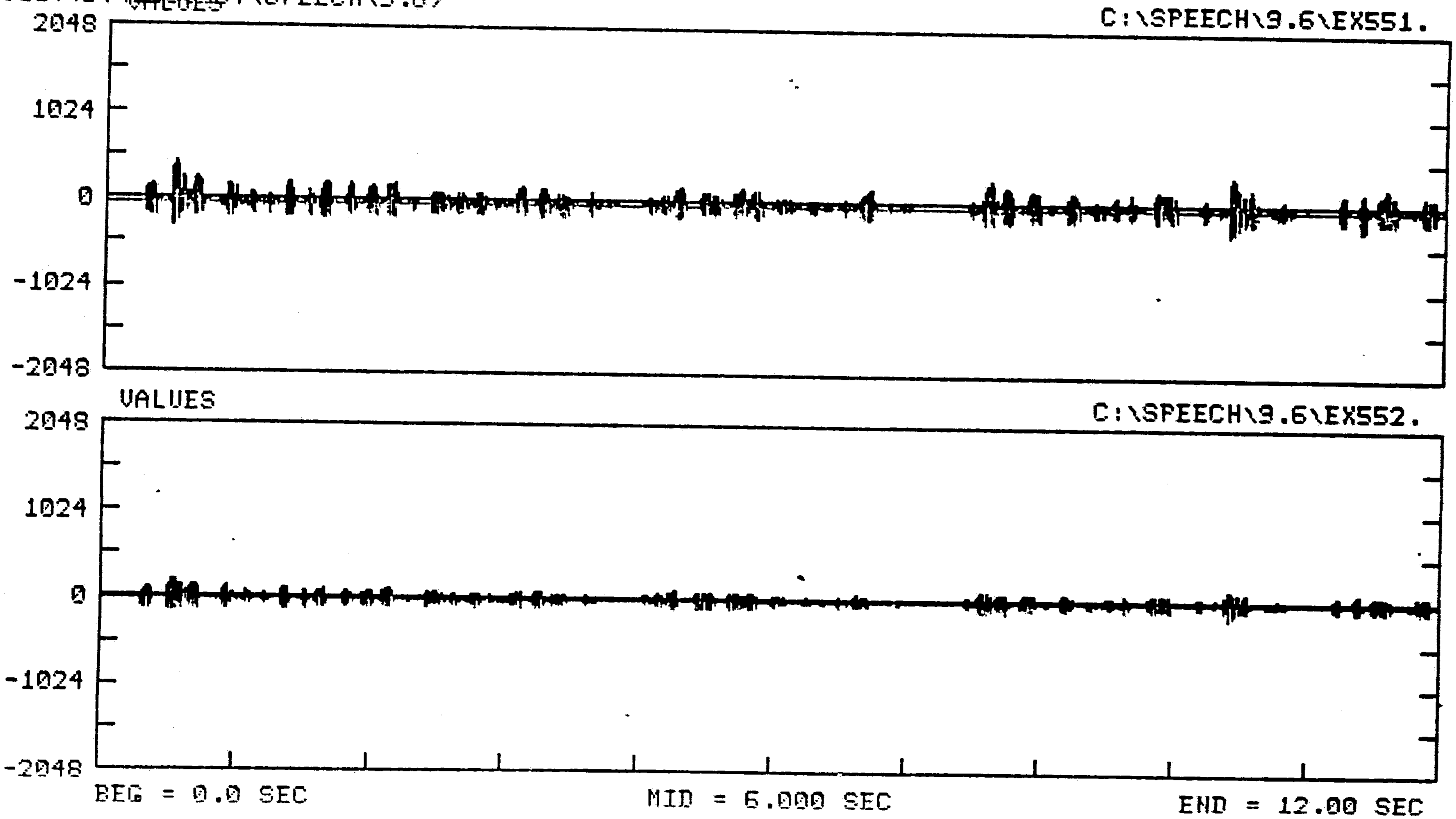
Object File: Stree , SmsTree

Conclusion & Observation:

The original sampled speech had a very low voltage level (1/4 Maximum Dynamic Rang) and had a DC-bias of about (.3-.5) Volts. A gain of 4 was selected for (from vax.) process to scale the synthesized speech. This waveform is shown (called Experiment #105-4).

ORIGINAL PAGE IS
OF POOR QUALITY

[11:49:20.44]C:\SPEECH\9.6>... SECTOR 1, STARTING FRAME 1, 1500 FRAMES, CONTEXT 64
[11:49:42.02]C:\SPEECH\9.6>



EXPERIMENT # : 107

Speaker: Rani, Paul

Time: 4:30 pm

Date: 10-15-85

Description: AGC=1.2 and M=16, L=32, Smoothed Version
Frame=80

Source File: PS-2.org, PP_1.org

Output File: PS801AGS.cmp, PP801AGS.cmp.

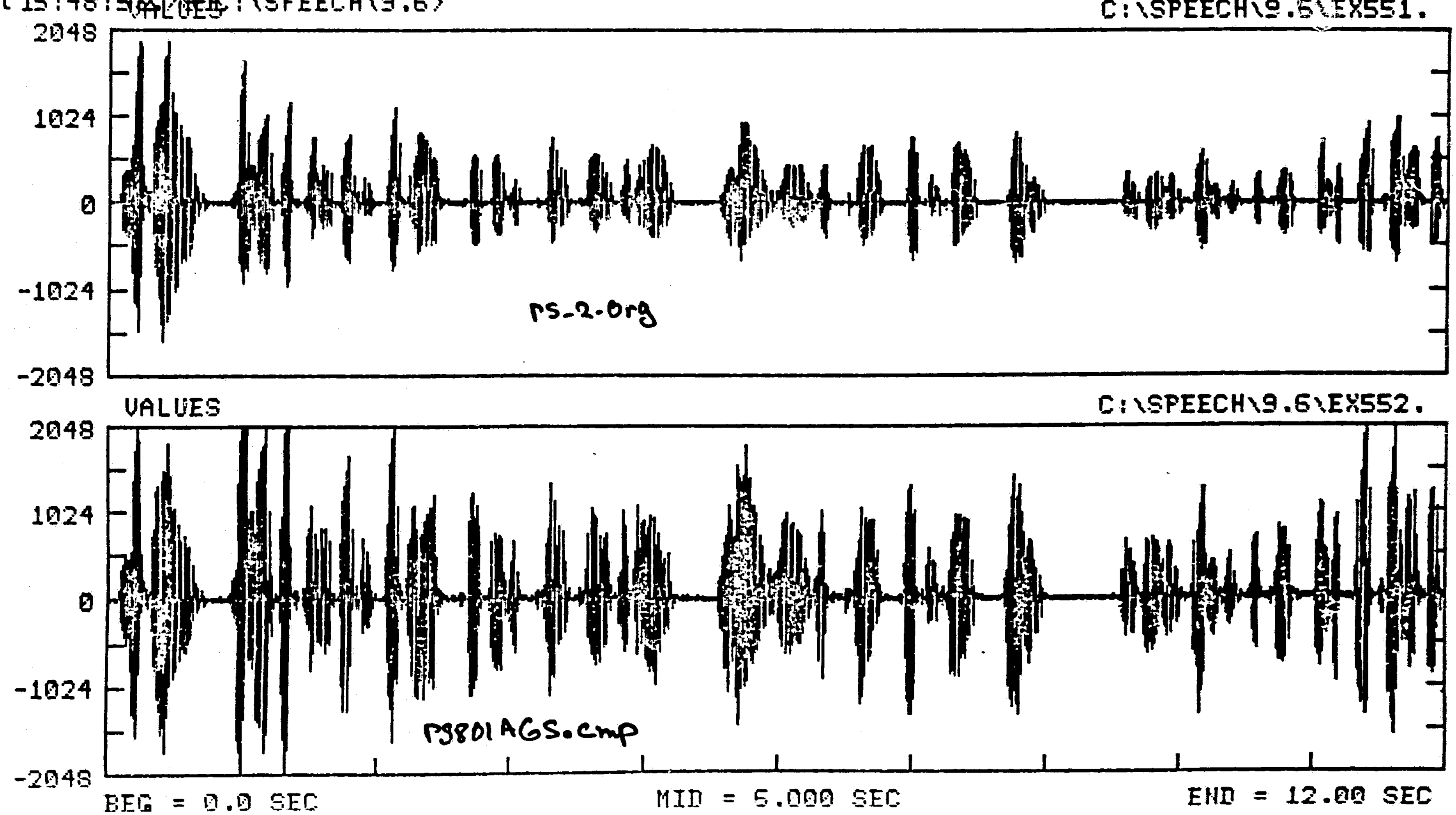
Object File: Sm80tm

Conclusion & Observation:

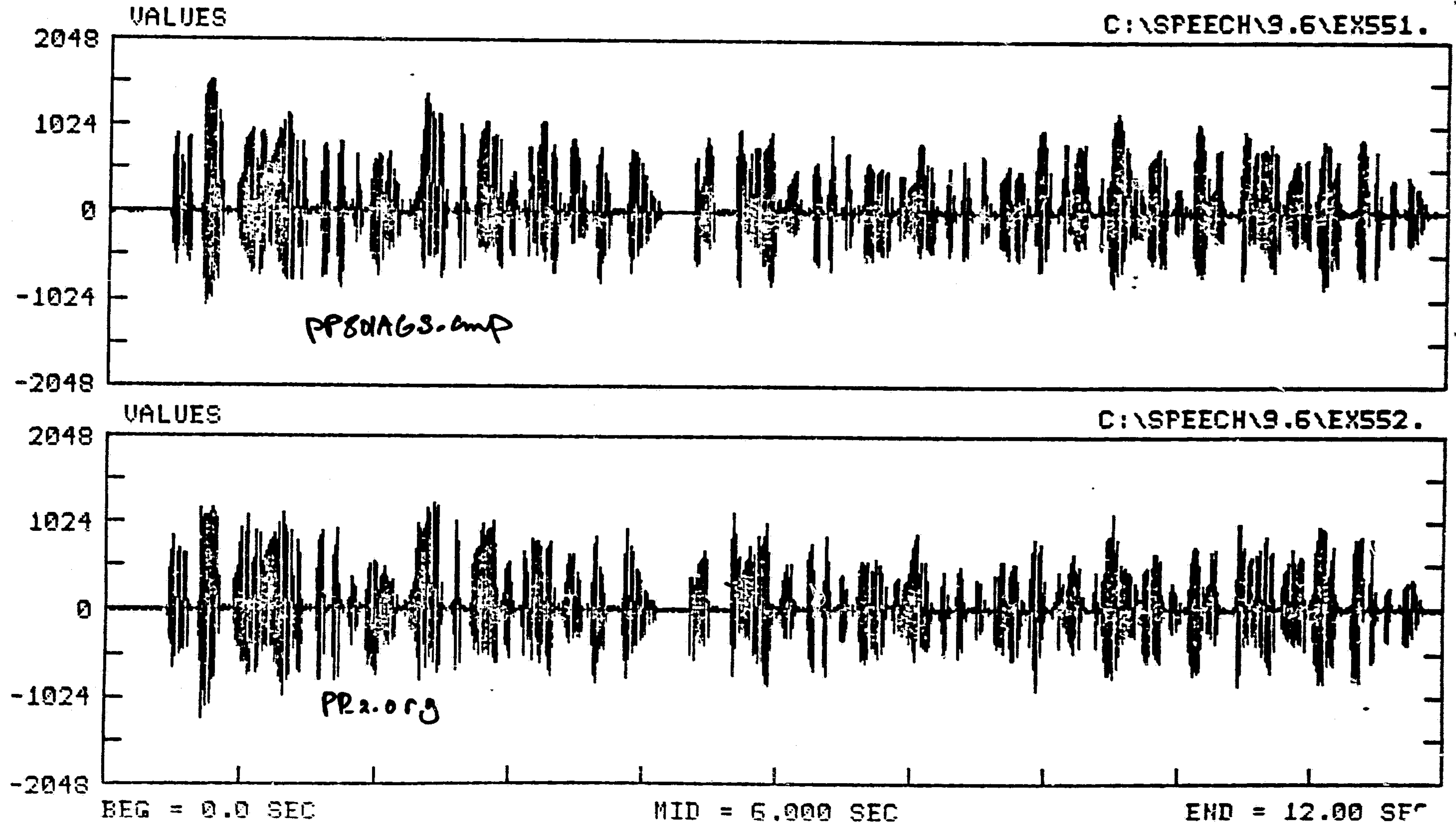
There is lots of clipping, I think the AGC is the cause of this effect. There is an obvious oversho in Voiced segments.

[15:48:37.20]C:\SPEECH\9.6\... STARTING FRAME 1, 1500 FRAMES, CONTEXT 64

[15:48:52.70]C:\SPEECH\9.6>



[16:46:50.18]C:\SPEECH\9.6\EX551. VECTOR 1, STARTING FRAME 1, 1500 FRAMES, CONTEXT 64



EXPERIMENT # : 106

Speaker: Paul Pearson, Rami Saadi

Time: 11:30

Date: 10-18-85

Description: NO AGC, M=16, L=32, Smoothed Version
Frame=80

Source File: rs-2.org, PP.2.org

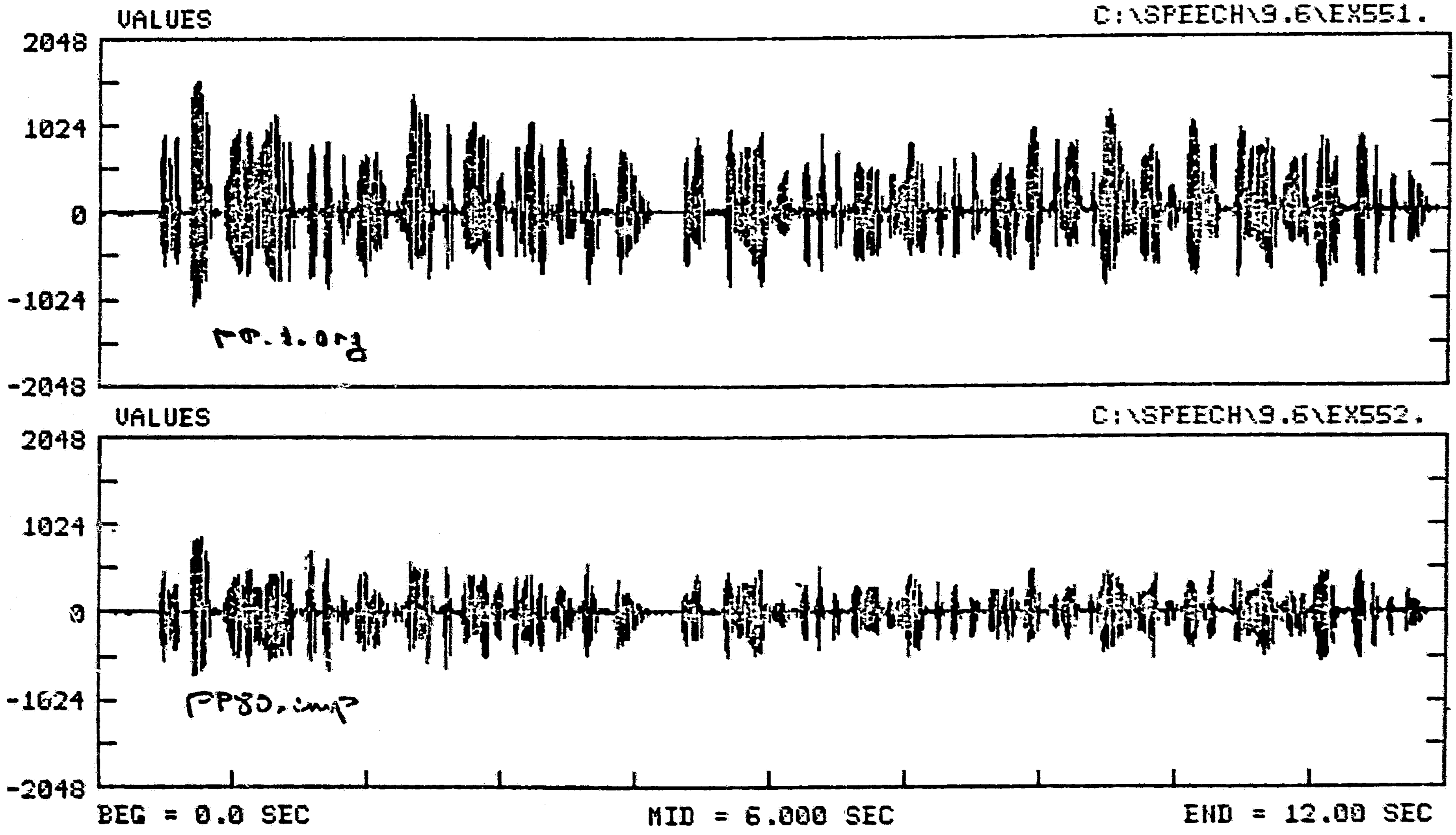
Output File: rS80.cmp, PP80.cmp

Object File: Smooth.

Conclusion & Observation:

This experiment is basically the same as the previous experiment. Except there is no Automatic gain control. The resulting speech sounds good to fair.

[11:35:08.33]C:\SPEECH\9.6 VECTOR 1, STARTING FRAME 1, 1500 FRAMES, CONTEXT 64



EXPERIMENT # : 107

Speaker: Rani

Time: 4pm

Date: 11-6-85

Description: Integer Implementation of Filter - Inverse

$$\rightarrow \boxed{H^{-1}} - \boxed{H} -$$

Source File: ns-2.org

Output File: rs107.cmp

Object File: int.1

Conclusion & Observation:

Sounded great. good step toward the true integer version

[16:30:57.56]C:\SPEECH STARTING FRAME 1, 1500 FRAMES, CONTEXT 64

[16:34:27.15]C:\SPEECH\9.6>