TECH[illegible]

#TECHRP REV[illegible]

Title Reference Signal Noise from D-1000 Mfg. Pro[illegible]	**Filename** 970210jg.doc / **Revision** 01
Project or Program Name Diasensor 1000	**Program Root Number** 51
Program Task Description Reference Standard	**Program Task Number** 99
Name Jeremy Grata	**Department** Engineering / **Date** 2/10/97
Technical Area Physics\Electro-Optics	
Subject and Key Technical Words Noise, Optimum Integration Time	
Associated Reports N/A	

Abstract

The noise content of the reference signal for M0 was calculated in a.u. and plotted as a function of signal integration time and as a function of sampling time. This was performed to determine the optimum integration time at which high frequency random noise and low frequency 1/f noise are simultaneously minimized. Also, the amplitude spectrum was calculated for various signal integration times to determine the corner frequency (frequency at which the random and 1/f noise have the same amplitude) as a check for the above results.

INVESTIGATION OF REFERENCE SIGNAL NOISE FROM M6

Jeremy A. Grata
Test Engineer

Technical Report

Biocontrol Technology, Inc.
Engineering Department

February 10, 1997

Manager:

Augustyn Waczynski

Introduction:

To minimize the noise in any signal recorded by the D-1000, it is first necessary to characterize the background noise that the D-1000 detects. In this report, M6 is used to record reference spectra (signal from white light generated by a tungsten filament back-scattered off of an aluminum oxide sample), for various integration (averaging) times. The noise (deviation in a.u.'s) is then calculated for various sampling times greater than the integration time. This allows the noise to be plotted vs. integration time or vs. sampling time or in 3-space with noise, integration time and sampling time mutually orthogonal.

What is expected is that for a fixed sampling time, as integration time increases, the noise will decrease in magnitude due to the averaging of random (high frequency) noise. Also, after some optimum integration time, the magnitude of the noise will increase with integration time due to the 1/f (low frequency) noise content. For a fixed integration time, however, it is expected that the noise will increase with sampling time and, if the signal does not contain radically varying 1/f content, approach some maximum. For integration times other than the optimum, the overall magnitude of the noise vs. sampling time curve will be greater than that of the optimum integration time.

The optimum integration time can be found by 1; plotting noise vs. integration time for a given sampling time and finding the minimum 2; plotting noise vs. sampling time for various integration times and identifying the integration time whose curve is overall lowest in magnitude and 3; plot the amplitude spectrum vs. frequency on log log scales and determine the inverse of the corner frequency. All three of these methods are used with the third being a check for the results of the first two.

Data Acquisition:

To minimize a large amplitude low frequency (1/f) noise constituent in the data, the air conditioning unit was turned off in the M6 lab.

All data was acquired using the pccore.exe program. M6 has a data acquisition rate of 455 spectra/0.3 seconds. For a given integration time in seconds, pccore is given the corresponding number of spectra to average by the following equation:

$$\#\ of\ spectra = Integrationtime * \left(\frac{455}{0.3\ seconds}\right)$$

For each integration time, 600 spectra were recorded and stored in a binary file which was demuxed for use with either matlab or excel. There was no time delay between spectral recordings, but it should be noted that there was a finite amount of time required for averaging the number of spectra to be stored as one spectrum in the final data set. For example, an integration time of 0.3 seconds requires 455 spectra to be averaged and stored as one spectrum in the final data set. This results in approximately 1.12 seconds (~0.82 seconds for averaging the 455 spectra) between the spectra in the final data set. The time between spectra in the final data set is the desired integration time + the time required for the averaging. The time required for averaging is of course dependent upon the number of spectra being averaged. References to data sets in this report will be done so by the integration time and date (if necessary) unless otherwise specified.

Besides the above data (a given integration time, 600 spectra with no time delay between spectra), a data set (referred to in this report as the "long data set") comprised of 0.3 second integration time, 5401 spectra (approximately 1hr 40min) with no time delay between spectra, was used in plotting the amplitude spectrum.

Data sets were recorded on various days, at various times of the day.

Data Analysis Procedure:

Calculation of Noise in a.u.

All noise calculations were performed using matlab. Plots were made with excel and matlab.

Noise vs. Integration Time:

The final data set for each integration time was organized in the following manner: 600 rows, each corresponding to one integrated spectrum; the first column was the time machine time index; columns 2-65 were the 64 channels (each a different wavelength) of spectrometer data (voltage readings); columns 66-75 were the data for the 10 sensors (voltage readings). Each data file was loaded into matlab in the form of a 600x75 matrix for analysis.

To calculate the noise in a.u. the spectral data for each row was first normalized with respect to wavelength by dividing each element in columns 2-65 by the average of columns 2-65 for a given row.

Next, a sampling time was chosen for converting the spectral data into a.u. Since $\log_{10}(\frac{SpectrumB}{SpectrumA})$ coverts the spectral data into a.u., the sampling time here corresponds to the time between spectrumB and spectrumA or the difference in the machine time index for spectrumB and spectrumA.

If the sampling time was an integer multiple of the time between successive spectra, then a new matrix was created for which each row of the new matrix was equal to $\log_{10}(\frac{SpectrumB}{SpectrumA})$ where spectrumA and spectrumB were columns 2-65 of the data normalized w.r.t. wavelength and spectrumB was recorded (sampling time) seconds after spectrumA. If the sampling time was not an integer multiple of the time between successive spectra, then two new matrices were created in the above manner. One of the new matrices had a time spacing between spectrumB and spectrumA such that it was the integer multiple of the time between successive spectra just greater than the sampling time. The other's time spacing was the integer multiple of the time between successive spectra just less than the sampling time.

The noise was then calculated as the average of the standard deviations of columns 1-10, 20-44 and 50-60 of the $\log_{10}(\frac{SpectrumB}{SpectrumA})$ matrix. For the case where two $\log_{10}(\frac{SpectrumB}{SpectrumA})$ matrices were calculated, the noise values of each were used to interpolate for the noise corresponding to the chosen sampling time.

For fixed sampling times, plots of noise vs. integration time were generated using excel (see Results section).

Noise vs. Sampling Time:

The noise data calculated above was then plotted as noise vs. sampling for fixed integration times.

"Long Data" Analysis:

Due to some unexpected results (deviation in trend) in the noise vs. sampling time plots for integration times longer than the optimum integration time, the long data set was recorded and analyzed as described below. To determine if the anomalous results were being caused by very low frequency 1/f noise, the long data set was filtered with a high pass filter, integrated by averaging and then analyzed using the techniques described above. It was expected that the anomalous results would not be found in the filtered/integrated plots if they were truly being caused by frequencies lower than the those that the high pass filter would allow.

The filtered long data set was created by a moving average of a number successive rows of the long data set (ex. row 2 of the filtered data set was the average of rows 1-3 of the long data set). Then the filtered data set was subtracted from the long data set to leave only the high frequency portion of the data (small 1/f content). This filtered long data contains only frequencies higher than $1/\Delta t$ where Δt is the time between successive rows of the long data (~1.12 seconds) times the number of rows used in the moving average. So as more rows were used in the moving average to filter the long data, more 1/f noise was added into the filtered long data. Note that the filtered long data has the same number of rows and the same time spacing between rows as the long data set.

The filtered long data was then integrated. The integrated data was calculated by replacing an integer number of rows by their average. The integration time was determined by multiplying the number of rows being averaged by the time spacing between rows (~1.12 seconds). The noise for this data was then calculated using the techniques in the above section and was plotted in the same manner (see Results section). Also, since the noise data could be plotted in three space (noise, integration time, sampling time) and the three space plot changes with longer filtering, a matlab movie was created to show these changes w.r.t. filtering. The movie is in a file titled "longmove.mpg" and can be viewed with any mpeg player. For better quality, run the m-file "noismovi.m" which loads the actual matlab movie matrix and plays the movie in matlab. To do this, you need to unzip the file "longmove.zip" in the same directroy you plan to run the m-file from.

Longmove.mpg Noismovi.m Longmove.zip

<u>Corner Frequency:</u>

For a given integration time, the rows of a given column of that data set represents the same quantity. Variations in the values of a column are then regarded as noise. By performing a fft on a column and calculating the amplitude spectrum ($\approx\sqrt{\textit{Power Spectrum}}$), it is possible to determine the corner frequency (frequency where 1/f noise and random noise have the same amplitude). The corner frequency then, is the frequency on a log log plot of the amplitude spectrum where a linear fit to the 1/f portion of the plot and a linear fit to the random noise portion of the plot intersect. Also, 1/(corner frequency) represents the integration time at the minimum of the noise vs. integration time plot or the integration time curve with the lowest magnitude on a noise vs. sampling time plot. Since all 64 columns of the spectral data were subject to the same environmental conditions, the average of all 64 amplitude spectra was used as the amplitude spectrum for a given integration time data set.

It should be noted that the only amplitude spectra capable of revealing the corner frequency are those with time spacings between successive spectra less than 1/(corner frequency). Log log plots of amplitude spectra for various integration times are show in the results section.

Results:

From the noise vs. integration time plots and the noise vs. sampling time plots, it is clear that the optimum sampling time is ~8.1 seconds. The change in trend of the curves representing integration times greater than 8 seconds was the cause for the long data set to be recorded and analyzed. Note that the anomalous trends are not present in the filtered/integrated long data noise plots. The corner frequency expected (1/8.1 seconds) in the log log plot of the amplitude spectra is ~0.12 Hz. Note that this is in good agreement with the actual plots especially for the long data set which has the highest frequency measurable (because of the short time between spectra) and a high resolution of frequency due to the long time required to record all 5401 spectra.

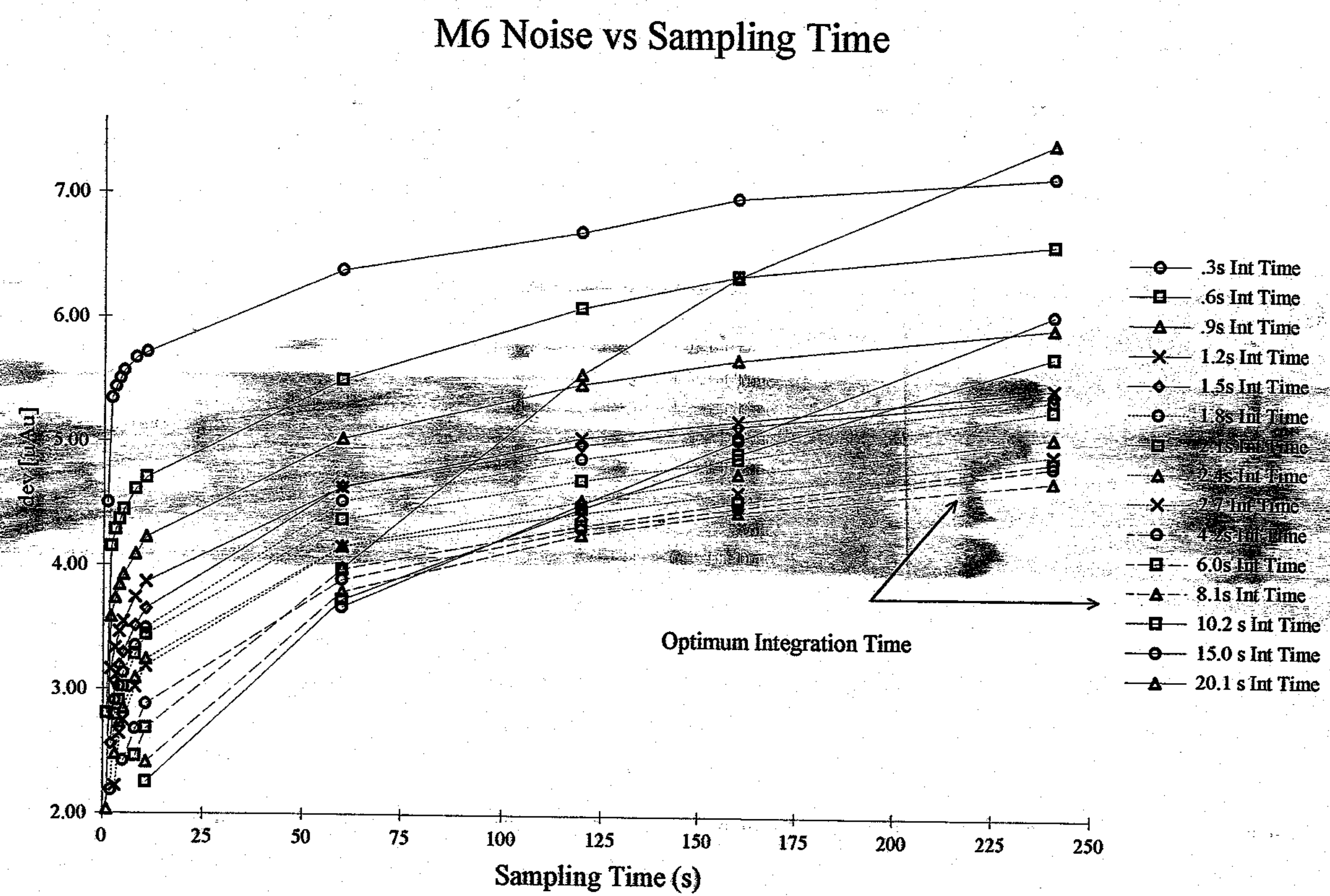

FIGURE 1: NOISE VS. SAMPLING TIME FOR VARIOUS INTEGRATION TIMES

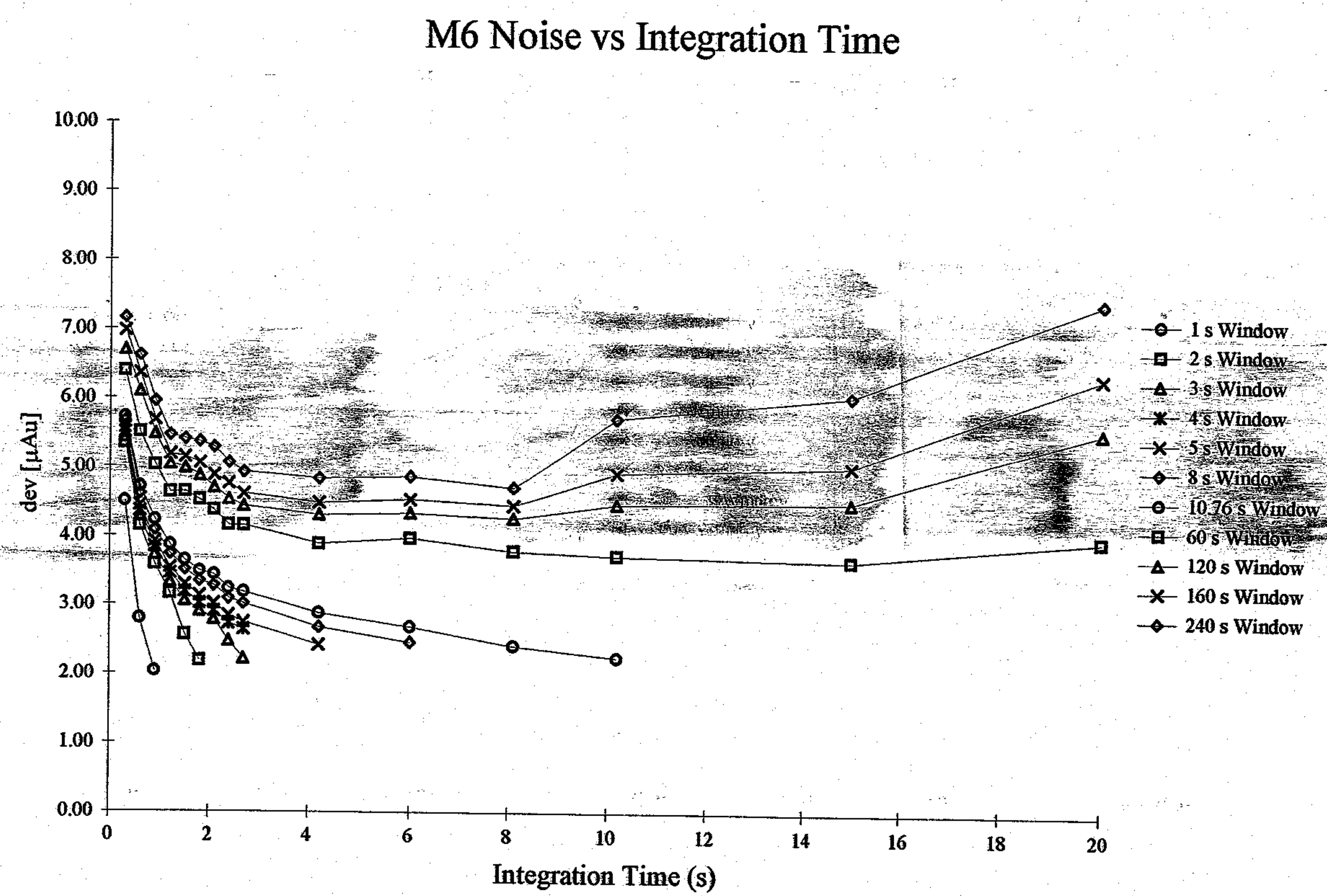

FIGURE 2: NOISE VS. INTEGRATION TIME FOR VARIOUS SAMPLING TIMES

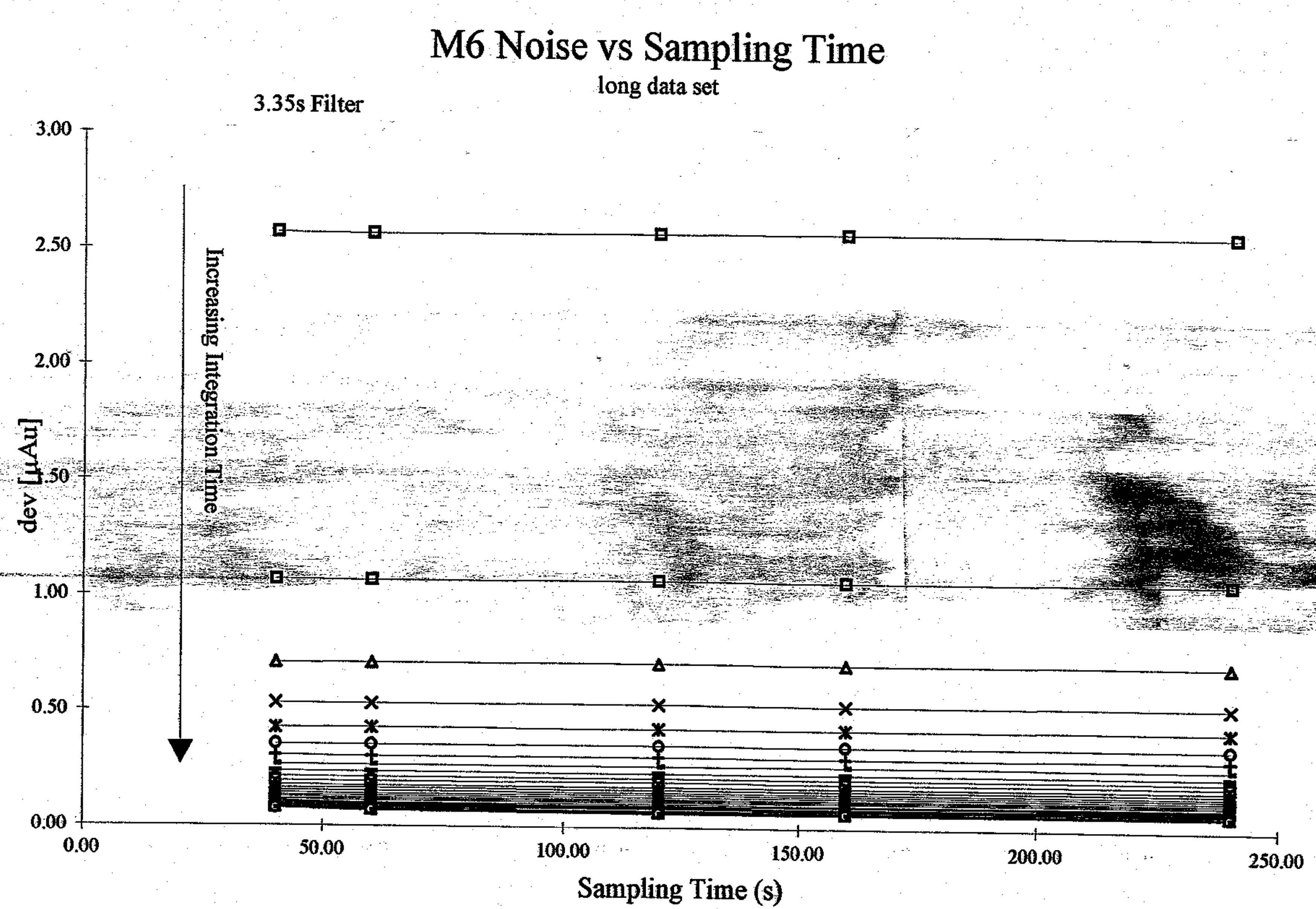

FIGURE 3: NOISE VS. SAMPLING TIME FOR 3.35S FILTER (LONG DATA SET)

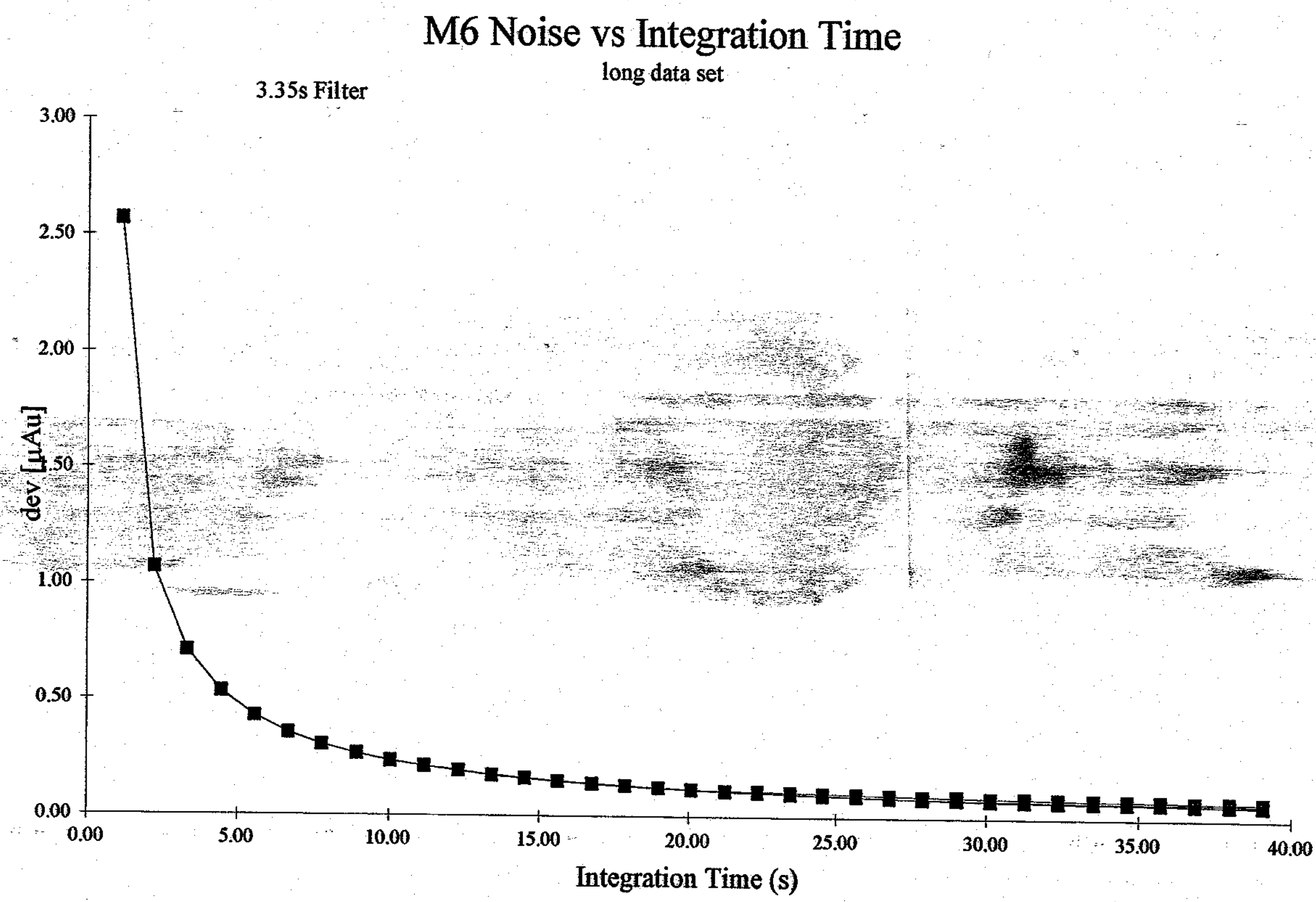

FIGURE 4: NOISE VS. INTEGRATION TIME FOR 3.35S FILTER (LONG DATA SET)

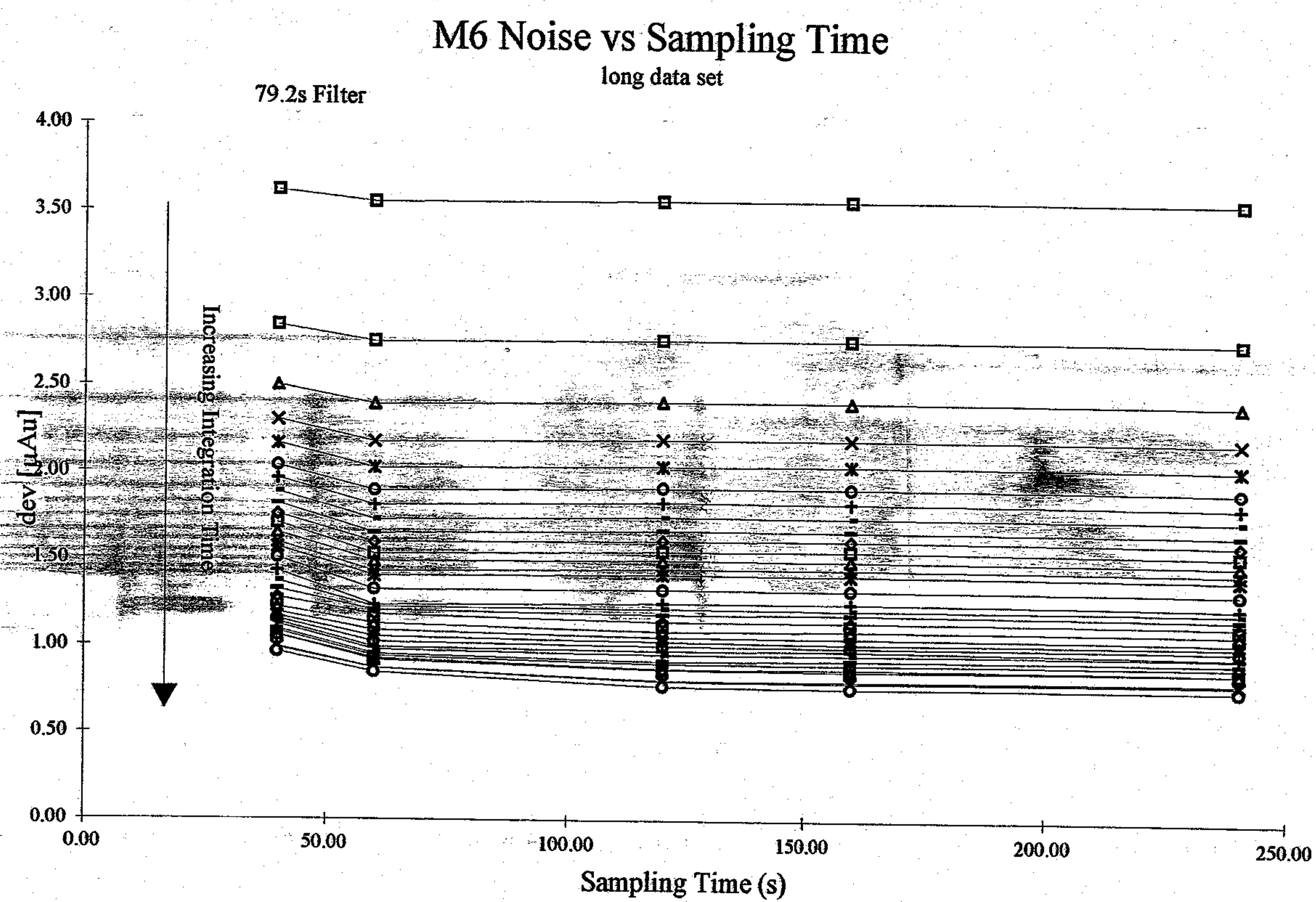

FIGURE 5: NOISE VS. SAMPLING TIME FOR 79.2S FILTER (LONG DATA SET)

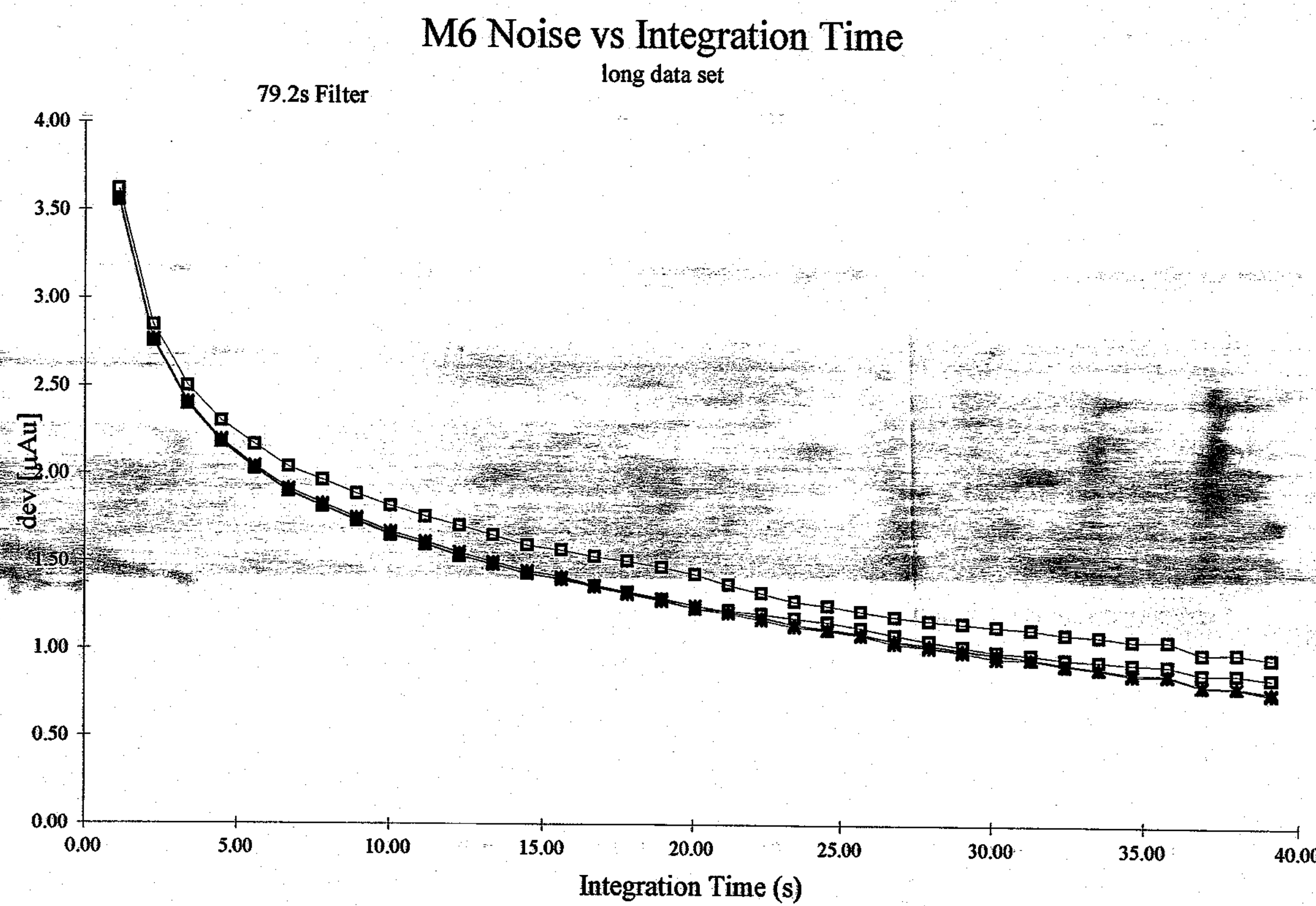

FIGURE 6: NOISE VS. INTEGRATION TIME FOR 79.2S FILTER (LONG DATA SET)

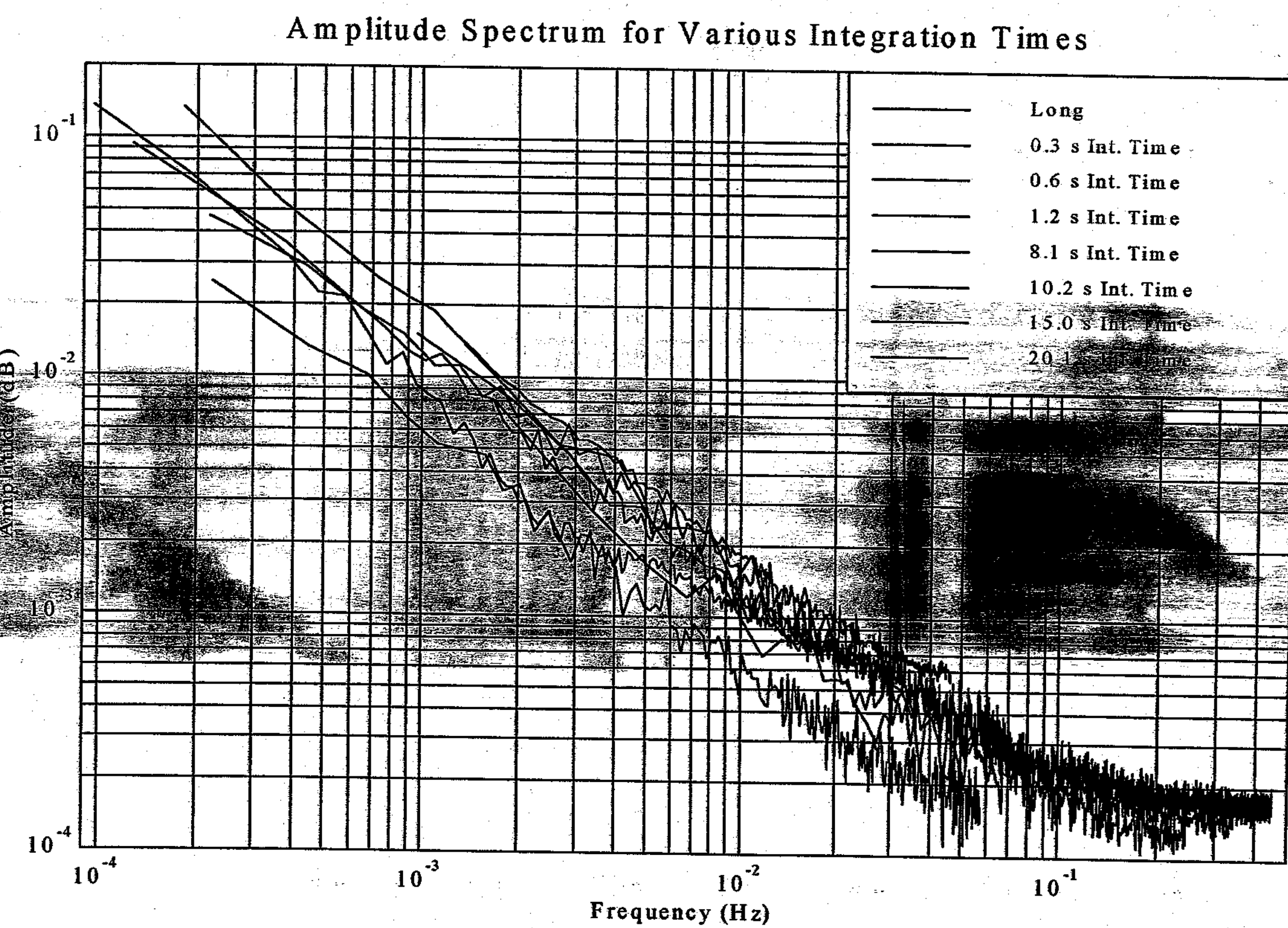

FIGURE 7: AMPLITUDE SPECTRUM FOR VARIOUS INTEGRATION TIMES

Conclusions:

Clearly, the optimum integration time required to minimize noise as much as possible is approximately 8 seconds. This is supported by the results of the noise vs. sampling time plots (and noise vs. integration time plots) as well as the results of the amplitude spectrum plots which show a corner frequency between 0.1 and 0.2 Hz (10 second and 5 second integration times respectively). However, an inspection of the noise vs. integration time plots show that there is very little reduction in noise for integration times longer than one or two seconds (< 1 μa.u.).

The study of the "long" data set seems to confirm the suspicion that the anomalous results found in the noise vs. sampling time plots were caused by some very low frequency noise source. This is supported by the absence of the anomalous trend deviations in the noise vs. sampling time plots of the high pass filtered data. The most probable source of this very low frequency noise is temperature variation. Temperature changes can be shown to cause proportional changes in spectral data. For long data acquisition times, temperature can vary quite non-uniformly and with considerable magnitude.