REAL TIME FOURIER TRANSFORMATION OF MAGNETOTELLURIC DATA

By

D. E. Wight, F. X. Bostick, Jr., and H. W. Smith

December 15, 1977

ELECTRICAL GEOPHYSICS RESEARCH LABORATORY

prepared by

Contract E(40-1) 5244

Energy Research and Development Administration

and

Contract (14-08-0001-16504)

United States Geological Survey

ELECTRICAL ENGINEERING RESEARCH LABORATORY

THE UNIVERSITY OF TEXAS AT AUSTIN

ABSTRACT

Power spectra estimates for magnetotelluric data are computed using a continuous Fourier transformation technique which is implemented in real time on a Hewlett-Packard 9825 programmable calculator. This technique, known as cascade decimation, applies 32 point sine and cosine transforms to sequences of data produced by successively applying a low pass digital filter and decimation by two operator to the original data. The resulting spectra estimates are naturally spaced on a log frequency scale and represent the average of independent samples in time.

Descriptions of the algorithms for cascade decimation are presented along with their implementation on the HP-9825. Also, a comparison is presented between cascade decimation and FFT for the computation of power spectra estimates for magnetotelluric data.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. Magnetotellurics

Magnetotellurics (MT) is a geophysical prospecting technique which estimates subsurface electrical properties of the earth from measurements of the natural fluctuations in its electric and magnetic field as measured at the surface. The first practical approach to MT was put forth by Cagniard (1953). His model, while adequate for a layered earth, may yield highly distorted results in regions having more complex structures. The tensor impedance model (Cantwell, 1960; Bostick and Smith, 1962; Swift, 1967) better describes the two dimensional geologies often encountered in practical applications of MT. The tensor impedance model relates the tensor relationships between the earth's electric (E) and magnetic (H) fields as

$$[E] = [Z]\,[H] \tag{1-1}$$

where the impedance tensor is given by

$$[Z] = \begin{bmatrix} Z_{xx} & Z_{xy} \\ Z_{yx} & Z_{yy} \end{bmatrix} \tag{1-2}$$

In two dimensional situations, the measured electric and magnetic fields are projected onto a set of rotated axes and the principle impedance values Zxy' and Zyx' are calculated with their axes parallel and perpendicular to the strike of the two dimensional feature. If the geology may be represented by a two dimensional model, Zxx' and Zyy' become zero and the tensor decouples giving

$$E_x = Z_{xy}H_y \qquad (1\text{-}3a)$$

$$E_y = Z_{yx}H_x \qquad (1\text{-}3b)$$

From these impedances, the apparent resistivities

$$\rho_{Axy} = \frac{|Z_{xy}|^2}{\omega\mu} \qquad (1\text{-}4a)$$

$$\rho_{Axy} = \frac{|Z_{yx}|^2}{\omega\mu} \qquad (1\text{-}4b)$$

may be calculated. The principle value for which the electric field is parallel to the strike of the two dimensional structure may be determined by use of data from the vertical magnetic field (Word, 1970).

Apparent resistivities are computed for a wide range of frequencies using the Fourier transform components of the electric and magnetic field time series measured in the field. Many procedures have been developed for inverting apparent resistivity versus frequency data to yield resistivity versus depth soundings (Wu, 1968; Becher and Sharpe, 1969; Patrick and Bostick, 1970; Laird and Bostick, 1970; Baily, 1970; Johnson and

Smylie, 1970). A simplified one dimensional algorithm has recently been developed by professor F. X. Bostick of the University of Texas which yields a continuous one dimensional inversion using only a small fraction of the computational effort required by earlier methods.

B. Real Time MT Processing

The development of new analysis techniques as well as the improved data acquisition equipment and procedures now make it possible to measure and analyze MT data in real time over the frequency range from .001 to 3000 Hz. Field work by the Electrical Geophysics Laboratory of the University of Texas during the summer of 1977 demonstrated such an ability. The broadband MT (BBMT) system developed at the University of Texas has demonstrated the ability to measure and process on site MT data over the total frequency range shown above for up to two sites per day.

The availability of powerful, yet relatively inexpensive digital processors has made such a system both possible and practical. However, the extreme measurement accuracies required for MT analysis and the extremely broad band of frequencies over which analysis must be performed, has made it necessary to develop new data acquisition equipment and analysis techniques to realize such a system.

Data in the audio frequency MT (AMT) band (10 to 3000 Hz) is Fourier transformed using a tunable bandpass filter realized as a phase sensitive dual hetrodyne receiver. The receiver is tuned to selected frequencies, approximately equally spaced on a log scale in the AMT band, one frequency at a time. The output of the receiver is digitized under the control of and is immediately processed by a Hewlett-Packard 9810 programmable calculator.

Unfortunately frequencies lower that about 1 Hz cannot be analyzed in the same manner as the AMT data. The AMT analysis is done one frequency at a time. This is acceptable in the audio band where data unfolds rapidly. However, up to four hours would be required to analyze the lowest frequencies. If real time analysis is to be done at frequencies in the range of .001 to 1 Hz, all frequencies must be analyzed at once. Data in this frequency range has previously been analyzed by recording the time series data in the field onto digital or analog tapes and analyzing the data at a later time using the fast Fourier transform (FFT) algorithm (Tukey and Cooley, 1965) implemented on a large digital computer.

One approach which has been considered for real time processing of data at frequencies below 1 Hz is use of the techniques now employed for processing AMT data, but duplicating the equipment as necessary to run all frequencies simultaneously. This approach, however, is quickly discarded when one investigates the cost and size of the components necessary to build filters of the quality demanded by MT equipment for such low frequencies.

Another approach considered is adaption of the FFT technique for on site implementation. The availability and relatively low cost of powerful minicomputers make it possible to record data as is presently done, then process it on site using the same techniques which are now implemented on large computers. It will be shown, however, that another technique, cascade decimation, not only yields data which is more easily adaptable to the MT problem, but also can be implemented on a significantly smaller processor that required for FFT processing.

The concept of decimation as a means for computing spectra was developed by Blackman and Tukey (1959). It was first applied to MT data as a technique for computing dynamic or running power spectra (Sims, 1965). The first application of cascade decimation to real time MT processing was during the summer of 1975 in the Snake River Plain Survey (Stanley, et al, 1977) using a Hewlett-Packard 9810 programmable calculator. The small number of usable reqisters made actual real time processing impossible. However, data could be taken under control of the calculator and stored on the digital cassette recorder which served as external memory for the calculator. After the run was complete, the cassette tape was replayed and the data analyzed using cascade decimation by the 9810. Unfortunately, several hours were required for the analysis after the data had been taken. Of significance, however, is the fact that all the results were available while the survey was in progress. Data was also recorded on analog tape for later digitization and analysis at the large central computational facilities of The University of Texas. The tensor impedance estimates obtained by cascade decimation were of high quality and compared favorably to those computed in the conventional way, using large FFT programs on the large computer.

A practical implementation of cascade decimation has been developed by the Electrical Geophysics Laboratory at the University of Texas. A Hewlett-Packard 9825 programmable calculator was acquired for use as the data acquisition controller and data processor for the new real time MT system. A digitizer which operates under the control of the HP-9825 was designed and built for the new system (Franklin, 1977). During the summer of 1977, two MT surveys were completed using this new system.

The remainder of this report describes the cascade decimation process, both as implemented in the University of Texas real time MT system and in a general sense. A comparison is presented between the performance of FFT and cascade decimation for analysis of MT data. Also presented is a discussion of current efforts in developing a more general and more powerful implementation.

II. REAL TIME MAGNETOTELLURICS USING CASCADE DECIMATION

A. Definition of Cascade Decimation

Cascade decimation is a Fourier transformation technique which yields spectral information on a log scale in the frequency domain. It is a continuous process which eliminates the need to store a number of data points prior to computing the spectrum. Actual transformation is accomplished by multiplying data points by weighted sine and cosine coefficients and accumulating the resulting products. This process is applied to the original data. Then, the data is decimated by two, discarding every other point. Decimation by two halves the Nyquist frequency for sampled data. Therefore a digital alias filter must be applied to the data prior to decimation to remove frequency components higher than the new Nyquist frequency. The new series of data, its points occuring only half as often as the original data, is transformed in exactly the same manner as the original data. The frequencies of the components evaluated at this level are half that of those obtained from the original data. These decimation and digital alias filter stages may be cascaded so that the frequency domain data produced at each level is at half the frequency of data produced by the stage prior to it. Figure 1 illustrates how the cascaded alias filters (Fi) and decimation operators (Di) are used to

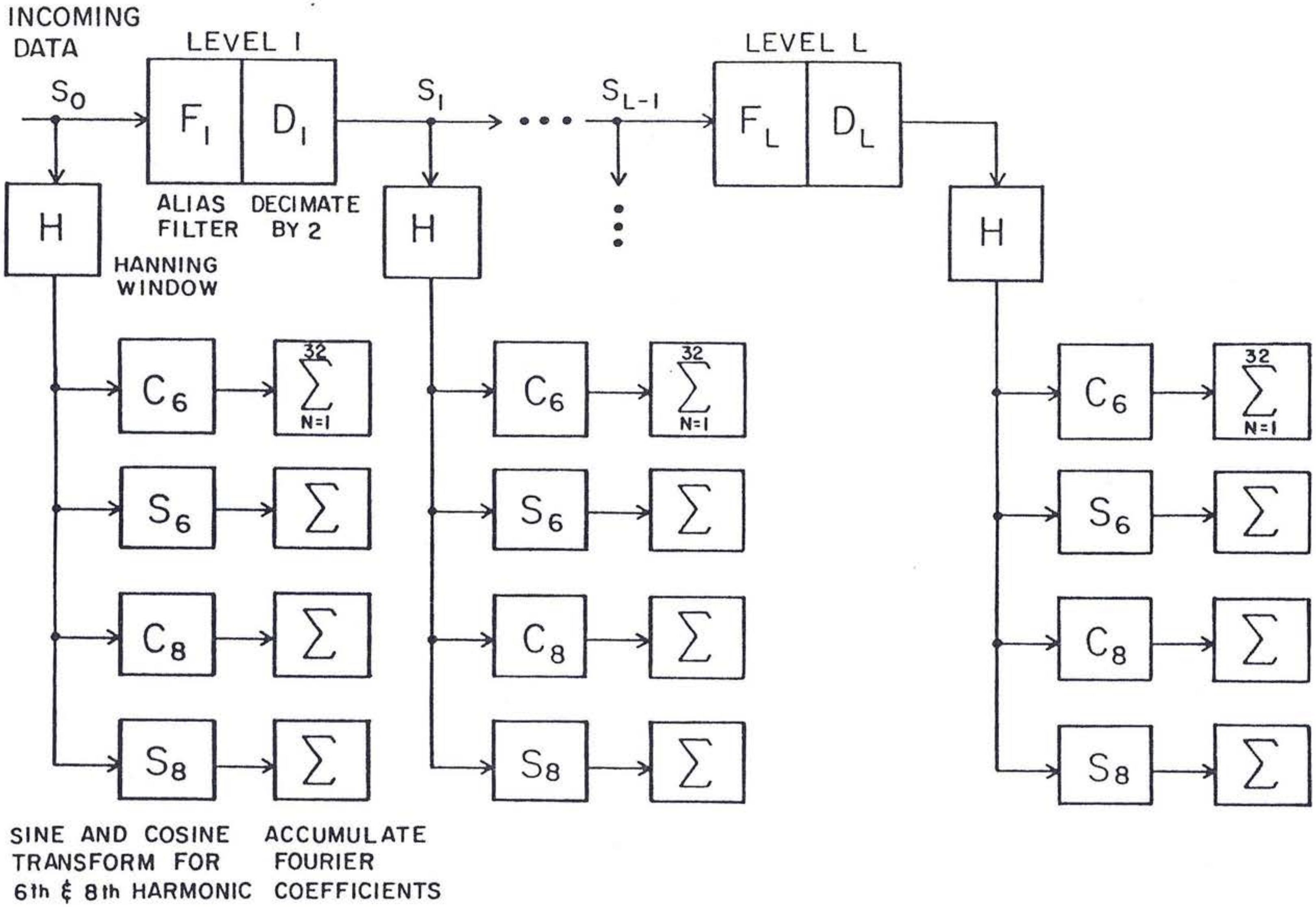


Figure 1 - Block Diagram of Cascade Decimation

generate data sequences (Si) to which a Hanning window (H) and the appropriate sine and cosine transforms for the sixth and eighth harmonics are applied. Each transform requires 32 points from the data sequence Si. The Fourier coefficients resulting from this 32 point transform are used along with coefficients calculated in the same manner for the other channels to calculate an independent set of auto and cross power spectra estimates. By selecting appropriate harmonics to be computed at each level, cascade decimation yields data points approximately equally spaced on a log frequency scale.

For application of cascade decimation to MT, two harmonics, the sixth and the eighth, are computed at each level. This yields approximately seven equally spaced points per decade, a number roughly comparable to the number of frequencies investigated in the audio band.

B. Hardware For Real Time Processing

Prior to digitization, the signals from the electric and magnetic field sensors are processed by equipment previously designed and built at the University of Texas. Reference is made to Figure 2. The electric field sensors are pairs of buffered electrolyte electrodes emplanted in the earth. Two orthogonal pairs give the components of the electric field Ex and Ey. Each pair of electrodes is fed into a differential amplifier. Line

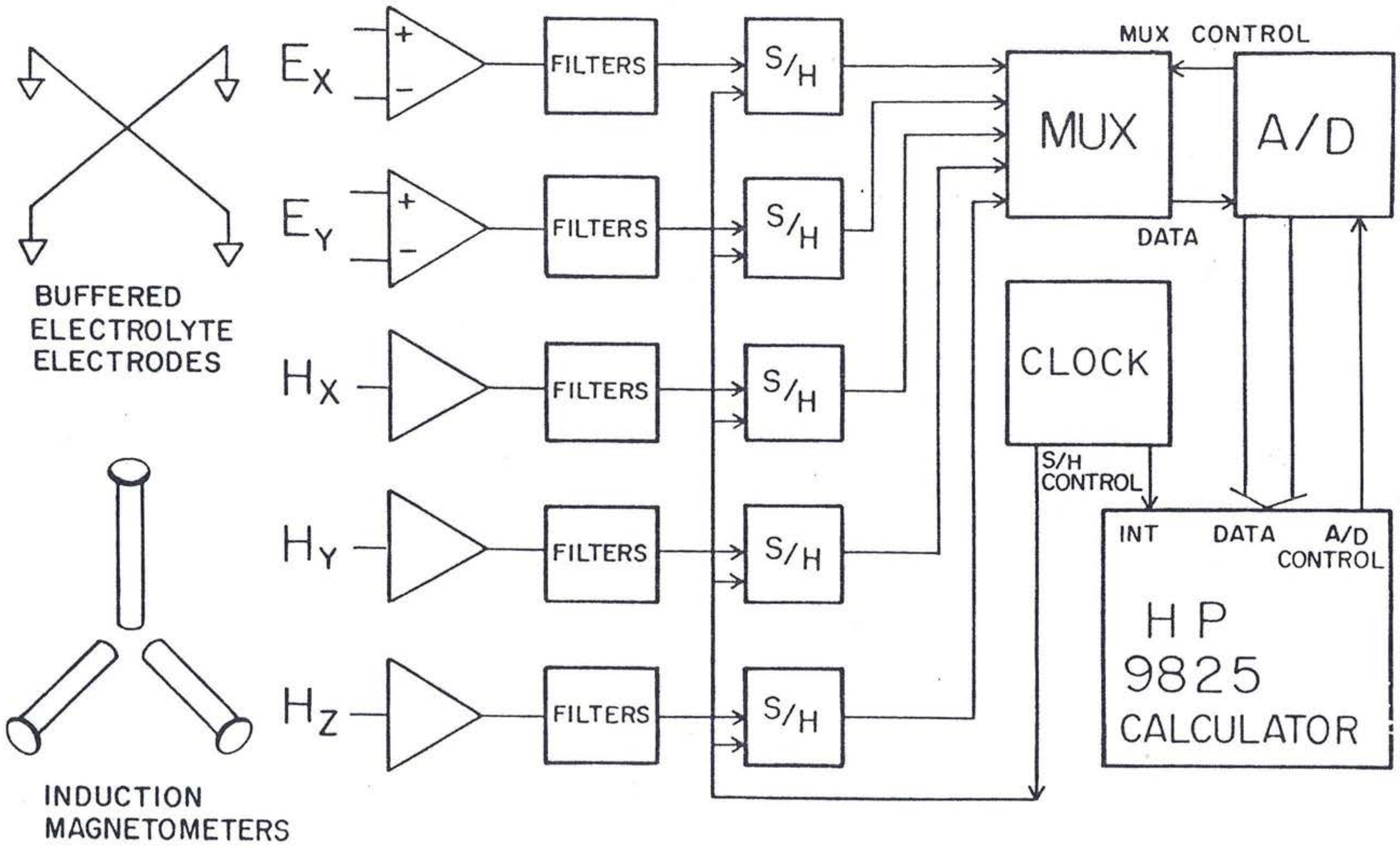


Figure 2 - Hardware for Real Time Processing

drivers amplify the resulting signal and send it through an instrumentation cable to the mobile processing station. The three components of the magnetic field Hx, Hy, and Hz are sensed by induction magnetometers. The magnetic field sensors are connected by shielded cable directly to the mobile processing station.

All five component signals are applied to sets of highly accurate, operator switchable filters. Variable gain buffer amplifiers following the filters enable the operator to utilize the full dynamic range of the analog to digital (A/D) converters. The outputs of the buffer amplifiers are fed directly into a bank of sample and hold (S/H) amplifiers. A digital clock based on a crystal controlled oscillator is used for controlling the S/H control signals. The digitizing interval is set by the operator. During data runs, the clock circuitry causes the S/H amplifiers on all five channels to simultaneously "hold" the signal voltage present at that moment on each channel. Then an interrupt signal is generated to the system controller. The controller, which also handles the task of real time analysis of the incoming data, is a Hewlett-Packard 9825 programmable calculator. The 9825 services the interrupt request by directing the digitizer to in turn digitize the voltage on each of the S/H amplifiers and send the resulting 12 bit binary numbers to the calculator for processing. While the calculator is processing that set of data points, the S/H amplifiers revert to the sample mode, tracking the signal voltages until another sample interval passes and the control circuitry initiates another digitization.

The period between digitizations must, of course, remain constant and must be at least as long as the worst case processing time for a data point. In order to make most efficient use of the calculator, the algorithms for cascade decimation are written in such a way as to distribute the calculations as evenly as possible between all data points.

C. Spectral Estimates

Data over the entire frequency range for low frequency MT may be analyzed in one run by cascade decimation. The computational requirements are such that it is not necessary to divide the alalysis into several frequency bands as is done for FFT analysis. Computation of the MT tensor impedances requires at least two independent estimates for each auto or cross power spectral term. One key difference between FFT and cascade decimation is the nature of these independent estimates.

The fast Fourier transform algorithm is applied to a data record of length N, where N is some power of 2. The algorithm yields the Fourier coefficients for the real and imaginary components of the first N/2 harmonics of the data record. For adequate alias protection, a twofold sampling redundancy is usually invoked, resulting in use of only the first N/4 harmonics. Fourier coefficients are computed in this manner for each of the 5 component signal. For each harmonic frequency, the 5 auto power spectral estimates and 10 complex cross power spectral estimates are computed.

Groups of power spectra at adjacent harmonics are averaged to yield power spectra estimates having a constant percentage bandwidth (constant Q). These estimates are equally spaced on a log frequency scale as is desired for MT analysis. The number of independent estimates averaged is, of course, much greater for the higher frequency constant Q estimates than for the lower ones. At the low frequency end, the two independent sample requirement necessitates discarding several of the lowest harmonics because the percentage bandwidth resulting from each average is much larger than that used for the remaining averages. The choice of a value for percentage bandwidth represents a tradeoff between resolution and scatter rejection in the spectral estimates. An example comparing estimates from FFT and from cascade decimation will be presented in a later section and should serve to clarify the details of implementing the concepts just presented.

The cascade decimation algorithm operates on a continuous string of data rather than on records of fixed length. It yields spectral estimates which represent approximately equal percentage bandwidth and therefore are equally spaced on a log frequency scale. Because constant bandwidth averaging is not necessary, independent samples for each power spectra estimate are obtained by averaging in the time domain rather than in the frequency domain. Practical considerations favor independent estimates in the time domain. Short periods of anomolously high energy sometimes result in saturations of system electronics which, in turn, result in erroneous estimates of power spectra. When using an FFT approach a number of saturations can be tolerated without serious degradation of estimate quality. However, if too many of them occur, the entire FFT record must be discarded and rerun. Cascade decimation, on the other hand, notes the number of saturated points which contribute to each independent record. If the number is too large, the record is discarded. For example, at Level 0 each record of 32 input data points is considered separately. If the record has any saturations, that one record is discarded. Records at lower frequencies are evaluated similarly. However, because as many as several thousand data points may contribute to these records, some saturations are allowed since their effect is negligible. The number of independent records accumulated is monitored by the operator. Even if many saturations are occuring, only estimates from periods of good data are accumulated. The operator simply allows the continous process to run until the desired number of independent samples is reached.

There is almost no practical limit to how long cascade decimation can continuously run. The longer it runs, the more independent samples are used for estimation of the power spectra, and hence, the more accurate the spectral estimates. There is, of course, always an abundance of data at the higher frequencies and the length of the run is determined by the minimum acceptable number of independent samples at the lowest frequencies. It will be explained later how overlapping Hanning windows are employed to double the number of samples accumulated at the lowest frequencies in a given amount of time at a cost of adjacent records being largely, but no longer totally independent.

D. Resistivity Soundings

When the operator who is monitoring the number of independent samples contributing to the power spectra estimates feels that sufficient data has been collected, the run is terminated. The raw power spectra estimates are normalized for the number of independent samples and the response of the cascade decimation process. The resulting data; all possible auto and cross power spectra for the 5 channels, for each frequency investigated, represents the total amount of information available from the data run. This data is recorded on magnetic cartridges and is also used for on site calculation of preliminary MT resistivity soundings. The procedure for computing resistivity soundings is briefly outlined below.

Tensor impedance estimates and data quality indicators are computed for reqularly spaced angles of rotation. For each angle, the measured electric and magnetic fields are projected onto a set of rotated axes and two of the tensor impedance estimates are computed.

The electric and magnetic field components are related by the impedance relationships

$$E_x = Z_{xx}H_x + Z_{xy}H_y \tag{2-1a}$$

$$E_y = Z_{yx}H_x + Z_{yy}H_y \tag{2-1b}$$

From these expressions, it is possible to derive the tensor impedance estimate

$$Z_{xy} = \frac{(H_x H_x^*)(E_x H_y^*) - (E_x H_x^*)(H_x H_y^*)}{(H_x H_x^*)(H_y H_y^*) - (H_y H_x^*)(H_x H_y^*)} \qquad (2\text{-}2)$$

and similar expressions for Zxx, Zyx, and Zyy. In like manner, the admittance relationships

$$H_x = Y_{xx} E_x + Y_{xy} E_y \qquad (2\text{-}3a)$$

$$H_y = Z_{yx} H_x + Z_{yy} H_y \qquad (2\text{-}3b)$$

may be used to derive the tensor admittance estimate

$$Y_{xy} = \frac{(E_x E_x^*)(H_x E_y^*) - (H_x E_x^*)(E_x E_y^*)}{(E_x E_x^*)(E_y E_y^*) - (E_y E_x^*)(E_x E_y^*)} \qquad (2\text{-}4)$$

and similar expressions for the other admittance terms. An impedance matrix is calculated from the relation

$$[Z] = [Y]^{-1} \qquad (2\text{-}5)$$

and from this comes the second set of estimates used for Zxy and Zyx. It can be shown that the latter estimates are biased upward by noise on the E signal and the former are biased downward by noise on the H signal (Sims, 1971). The angle which maximizes

$$|Z_{xy}|^2 + |Z_{xy}|^2 \qquad (2\text{-}6)$$

locates the direction parallel to the strike of a two dimensional feature. Maximum and minimum apparent resistivity curves are computed for angles parallel and perpendicular to the strike by equations (1-4a) and (1-4b) where the impedance estimates used are given by (2-2). This estimate is usually picked because the E system noise has been found to be smaller than the noise in the H system most of the time. The percentage spread between the two sets of impedance estimates described above has proven to be a very good indicator of data quality.

Determination of maximum and minimum apparent resistivity curves is actually based upon factors in addition to those enumerated above. A complete discussion of selecting impedance estimates and determining maximum and minimum apparent resistivity curves would be extensive and is not presented here. Resistivity soundings are obtained by applying a recently developed phase smoothing technique (Boehl and Bostick, 1977) and the continuous inversion algorithm (Bostick, et al, 1977) to both apparent resistivity curves. Interpretation of the resulting inversions to obtain a clear picture of the geological structure is an extensive subject and is beyond the scope of this thesis.

Although the picking of impedance estimates and determination of apparent resistivity curves in the field is done hastily and is subject to careful scrutiny at a later time, resistivity soundings produced in the field have proven very valuable in selecting future sites to obtain the most information about geological structure from a limited number of MT sites.

III. OVERVIEW OF CASCADE DECIMATION

The real time computation of MT spectra is naturally divided into three parts. First, the process of decimation by two and applying a digital alias filter is continuously being applied to both the sequence of input data points and to other sequences of points generated within the machine by the cascade decimation process. The decimation and alias filtering process are accomplished simultaneously using an algorithm described later. Associated with this process is a control algorithm which determines what processing, if any, is to be done at each of the decimation levels for a given data point. The algorithm limits the number of decimation and filter operations performed between any two incoming data points.

The decimation and filtering operation produces a string of data points (for each of the five channels) at each level. At Level 0 the string is simply the original set of data points and thus a point is produced at Level 0 on every digitization cycle. At Level 1, the string is produced by applying a digital low pass filter to the original data string then discarding every other point (decimation by two) of the filtered data string. Thus, a point is produced at Level 1 on every other digitization cycle. At Level 2, the string is produced by applying a digital low pass filter to the string produced by Level 1 and decimating by two, resulting in a point produced at level two on every 4th data point at Level 0. This process is repeated for each level of decimation. Presently, nine levels of decimation are being implemented, but the process will work for any reasonable number.

The second part consists of applying a Hanning window to each data sequence and calculating the Fourier coefficients by transforming records of 32 consecutive points within the sequence. At each level, sine and cosine transforms for the sixth and eighth harmonics are computed. This results in a set of Fourier coefficients at frequencies approximately equally spaced on a log scale as shown in Figure 3. Harmonics computed using FFT must be averaged over constant percentage bandwidths in order to achieve such a distribution.

Four arrays of 32 numbers each are stored in the calculator memory. These four arrays represent 6 and 8 cycles of a cosine and 6 and 8 cycles of a sine, each with a Hanning window applied. Each data point in the 32 point record being transformed is multiplied by the appropriate factor from each of these four arrays and the products are summed over the 32 points into four accumulator registers. When the 32nd point in the transform record has been processed, the four accumulator registers contain the Fourier coefficients of the real and imaginary components

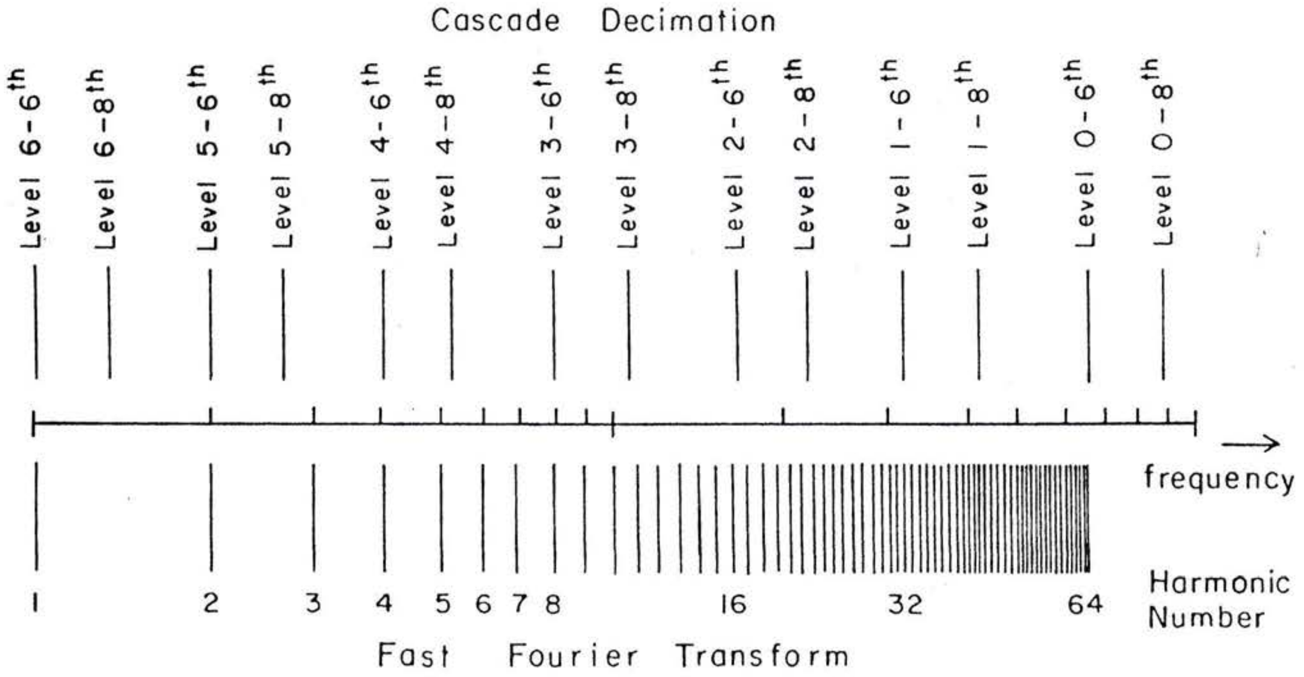


Figure 3 - Frequency Distribution of Fourier Coefficients

at the frequencies coresponding to the sixth and eighth harmonics at that level.

At the higher decimation levels where a large number of digitization cycles are required to produce each data point at that level, records are interlaced. Interlacing causes each data point to contribute to two frequency records, its contribution to each determined by the Hanning window functions. This produces twice as many tranformed values, although adjacent interlaced records are not totally independent. Fourier transformation is performed concurrently with the decimation process and is largely controlled by the same algorithm.

The third part of the real time process is the computation of the auto and cross power spectra estimates for the five channels. Each time a Fourier transform record is complete at any level the auto and cross power spectra components, 25 for each frequency are calculated (ExEx*, ExEy* real, ExEy* imag, ExHx* real, ExHx* imag, ... HyHz* real, HyHz* imag, HzHz*). These terms are summed into registers which accumulate the auto and cross terms. When the run is complete, these accumulators are normalized for the number of records accumulated and the system response of the cascade decimation process. The values are then recorded on digital cassette and are used for calculation of the tensor impedance estimates as previously explained.

complete and control returns to the main program.

The main program simply scans the "record complete flags" in a particular order. Each time it encounters a flag which is set, it computes the auto and cross power spectra for that level then clears the flag. The sampling period is set equal to the longest interrupt service time. Cross and auto spectra are computed only during the interval between completion of interrupt service and the end of this period. Every effort has been made to equalize the computation time required to service interrupts. The result of this interrupt driven system is that the processor is almost constantly being used. Higher priority levels are always processed promptly. Levels where an excess of data will be produced are processed only when no other processing can be done.

IV. DECIMATION AND ALIAS FILTERING

A. Objective

The sampling theorem states that if a signal has no frequencies greater than the Nyquist frequency as given by

$$f_n = \frac{1}{2T} \qquad (4-1)$$

where T is the sampling interval, then the continuous function h(t) can be uniquely determined from a knowledge of its sampled values by

$$h(t) = T \sum_{n=-\infty}^{\infty} h(nT) \; \frac{\sin 2\pi \, fn \, (t-nT)}{\pi \, (t - nT)} \qquad (4-2)$$

Frequency components greater than the Nyquist frequency result in a distortion of the Fourier transform known as aliasing. Prior to digitization a signal must be band limited such that no significant energy is present at frequencies greater than Fn. In practice, the finite roll-off of alias filters suggests using twofold sampling redundancy for adequate alias protection. Such a practice will be employed here. Thus, the maximum sample interval is given by

$$T = \frac{1}{4f_o} \qquad (4-3)$$

where f_o represents the highest frequency of interest.

Decimation by two (discarding every second data point) of a data sequence to produce a new data sequence with an effective sample interval twice that of the original sequence requires that the bandwidth of the original data be reduced prior to decimation. The upper frequency limit for twofold alias protection becomes

$$f_1 = \frac{f_o}{2} \tag{4-4}$$

Thus, prior to decimation, it is necessary to apply an alias filter to the data. The desired frequency response for such a filter is shown in Figure 4. Such alias filtering is realized by a finite impulse response digital filter of length five. The filter length represents a compromise between computational effort, memory requirements, and start-up transients on one hand, and filter effectiveness on the other. Present implementation of cascade decimation is such that significant computational advantages result from the use of a filter having an odd number of coefficients. A filter of length five provides adequate alias protection not possible with three, while the added protection gained by going to a length of seven was judged insufficient to justify the added computational requirements. The filter coefficients were determined by use of the Remez Exchange Algorithm

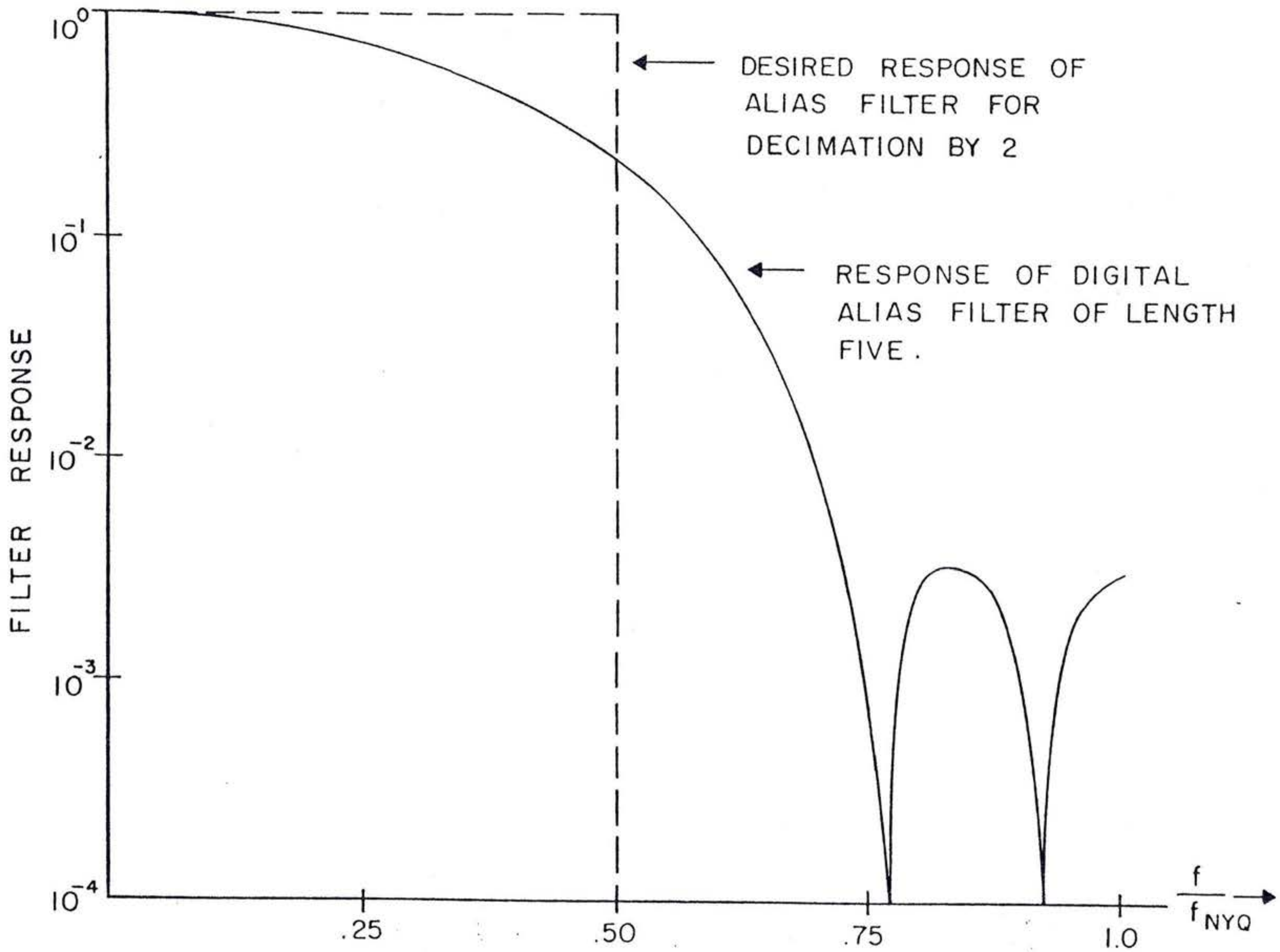


Figure 4 - Alias Filter Frequency Response

(Walker, 1976). The filter coefficients used are:

H0 = 1.0
H1 = 3.41421356
H2 = 4.87100924
H3 = 3.41421356
H4 = 1.0

In the following discussion, however, these coefficients will be referred to as 1,3,4,3,1 for simplicity. The frequency response of the filter is given by

$$H(e^{j\omega}) = \sum_{n=o}^{N-1} h_n e^{-j\omega nT} \tag{4-5}$$

where N is the filter length and T is the sample period. The frequency response of the above filter is shown in Figure 4. Performance of this filter has proven to be completely adequate, relulting in aliasing of less than 0.3 percent. Convolution of the filter with the data sequence is accomplished simultaneously with the decimation process as represented in Figure 5. Each horizontal line of points represents a sequence of data points at a given level. Level 0 is simply the sequence of time series data points from the digitizer. Each point at Level 1 is a weighted sum of five consecutive points at Level 0 as shown, the weighting factors being the filter coefficients described above. The "1,3,4,3,1" notation in Figure 5 clarifies the procedure.

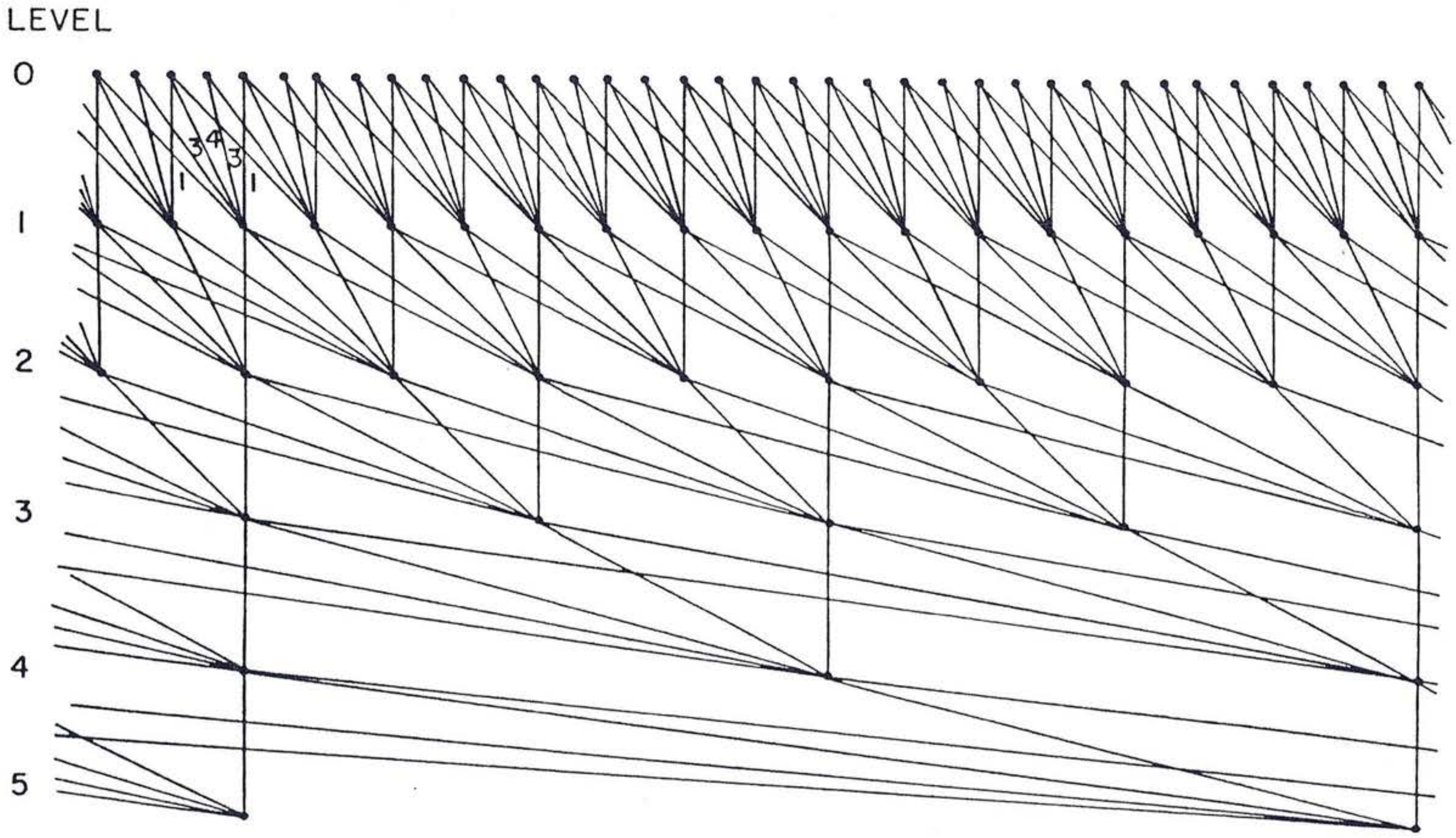


Figure 5 - Schematic Representation of Decimation

B. Implementation

Figure 5, depicting the cascade decimation process, also may be used to explain the implementation of cascade decimation using only three registers per level. Examining how each point at Level 0 is used, one finds that only two cases exist. It may contribute to three points at Level 1 with weighting factors of 1,4, and 1 respectively, or it may contribute to only two points with weighting factors of 3 and 3. Further, it may be noted that these two cases occur on alternate points at any level. Henceforth, these two cases will be referred to as "even" processing corresponding to the case where factors of 1,4, and 1 are used and "odd" processing corresponding the case where factors 3 and 3 are used.

It will now be shown how decimation and filtering is accomplished in three registers- A,B, and C. Define three procedures- "even processing", "odd processing", and "shift" which operate on registers A,B, and C and incoming data point N . Parentheses denote "contents of".

Even Processing of Data Point N

Data Point N x 1 + (A) → A

Data Point N x 4 + (B) → B

Data Point N x 1 + (C) → C

Odd Processing of Data Point N

Data Point N x 0 + (A) → A

Data Point N x 3 + (B) → B

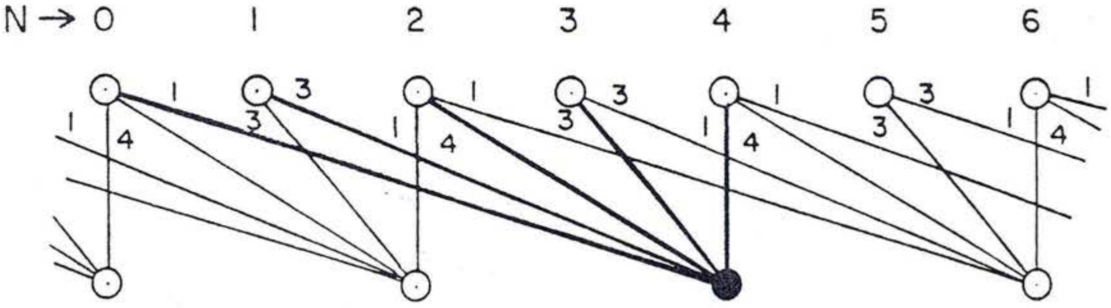
Data Point N x 3 + (C) → C

Shift

(B) → A
(C) → B
0 → C

These three procedures will provide a new way of approaching the filtering and decimation problem, an approach which is functionally equivalent to the process depicted in Figure 5. Both approaches are depicted in Figure 6. An expanded view of the cascade decimation scheme is presented with the inclusion of weighting factors. Also presented is a chart which summarizes how even processing, odd processing, and shifting are used to implement this scheme. Numbers in the table indicate the factors by which data point N is multiplied before adding it to the contents of the A,B, and C registers. For example, the first data point is simply added to the contents of registers A and C and is multiplied by 4 and added to the contents of register B. The next data point is multiplied by 3 and added to both B and C, but register A is not effected. Prior to processing of the third point, the contents of B will be transferred to A, those of C to B, and register C will be cleared.

The process is best understood by tracing the computation of a decimated point. Heavily shaded lines indicate that the point of interest is computed as 1 times data point 0, plus 3 times data point 1, plus 4 times data point 2, plus 3 times data point 3, plus 1 times data



N	a	b	c	
0	1	4	①	EVEN
1	0	3	③	ODD
	←	←	←	SHIFT
2	1	④	1	EVEN
3	0	③	3	ODD
	←	←	←	SHIFT
4	①	4	1	EVEN
3	⓪	3	3	ODD
	←	←	←	SHIFT
6	1	4	1	EVEN
7	3	3	3	ODD

Figure 6 - Implementation Using Three Registers

point 4. Circled entries in the table indicate development of this data point. Tracing this development verifies that it is computed exactly as indicated above.

Several points regarding this process should be noted. Each even cycle, a new decimated point is completed. This point is in the A register. Because odd processing does not effect the A register, the point is available for two successive cycles at that level. In fact, the A register at Level L always contains the most recent point in the data sequence S_L. It will be recalled from Figure 1 that the data sequence which feeds the decimation process at Level L is S_{L-1}. Hence, for any level of decimation L, the input data will be found in register A_{L-1}. Further, the same data will be available for two consecutive cycles, an even cycle and an odd cycle. This fact will be used to greatly even out the processing time for each point by computing the sixth harmonic coefficients on even cycles and the eighth harmonic coefficients on odd cycles.

One other key feature arises from the fact that the filter coefficients are symmmetrical (linear phase) and are scaled such that two coefficients are unity. Examining the procedures for even and odd processing, one finds that only one multiplication, either by 3 or by 4 is required for each decimation cycle. Since multiplications often consume a disproportionate amount of computation time compared to simple replacement and additions, this feature can significantly reduce the worst case computational effort required for processing an incoming data point.

Presently, the decimation process is implemented on a HP-9825 programmable calculator which utilizes a 64-bit floating point representation for all variables. It may be shown that such a representation makes it possible to normalize for the decimator response after the run is completed without loss of significant accuracy. The response factor for the Kth harmonic (K=6 or 8) at Level L due to cascade decimation is

$$A_{KL} = \prod_{i=1}^{L-1} \sum_{n=o}^{M-1} h_n \left| e^{-j \frac{KnT}{2^{j}}} \right| \qquad (4-6)$$

where T represents the sample period of the input data, M represents the filter length, and Hn are the filter coefficients. For other processors using different internal representations for the variables, it may be necessary to do some scaling during the decimation process to maintain the desired accuracy.

As with any digital filtering process, start-up transients result in erroneous values for the initial data values. For levels where a large amount of data is available, the transient effect is completely eliminated by discarding all effected data. At levels corresponding to very low frequencies, some error due to start-up transients is allowed since discarding data at these levels can mean a significant increase in data acquisition time and because the error introduced is very small. The first three points of the 32 used for the first Fourier transform record at each level may be effected by the initial start-up. However, because a Hanning window is applied to the record prior to transformation, the contribution of the first three points to the calculated Fourier coefficients is quite small.

C. Limiting The Number of Decimations Per Point

Each sample period an input data point is processed by Level 1. Consecutive points are processed alternately using even and odd processing. On each even cycle, a new data point is completed at Level 1 which immediately processed by Level 2. This level also uses even and odd processing for alternate points. Each even processing cycle at the second level produces a new point which is immediately processed by Level 3. This pattern, when extended to a system of L levels, results in a new point in data string Si (i=0,1,2...L) being produced each 2^i input data points. Figure 7 illustrates the levels at which processing occurs for successive input data points. Note that although half the time only one level is processed, every 2^L points, L decimations occur. Since

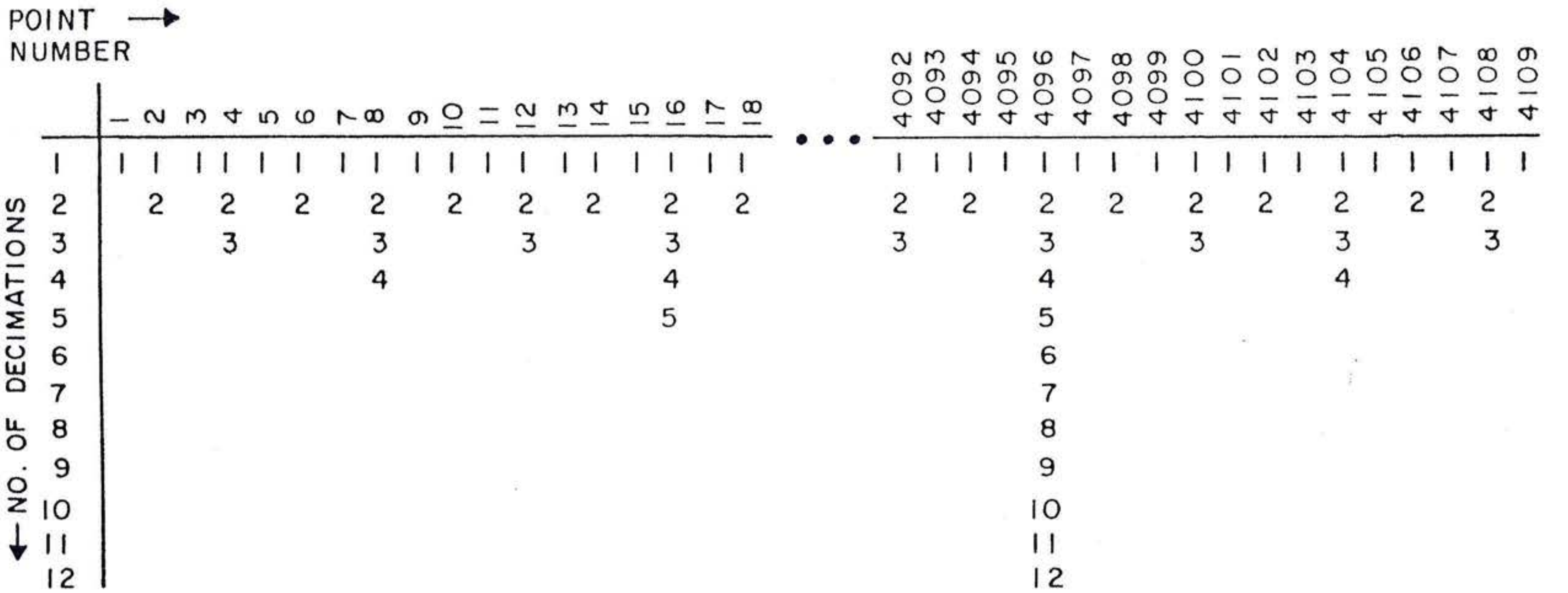


Figure 7 - Levels at Which Decimation Occurs

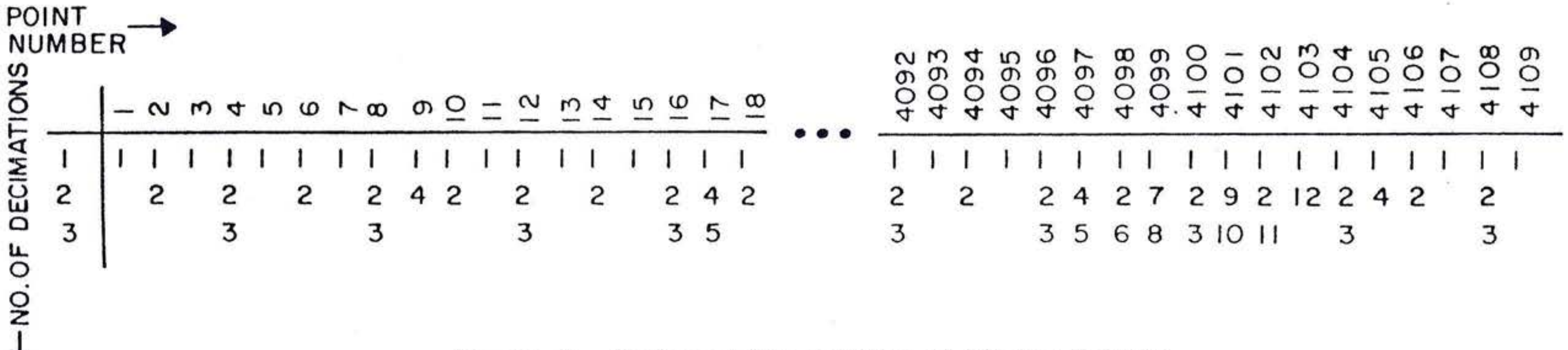


Figure 8 - Deferred Processing of Higher Levels

the sample period must be selected to allow completion of the worst case computation, this situation obviously results in very poor time utilization. By deferring some of the decimations as shown in Figure 8, the maximum number of decimations which must be performed can be limited to three. It will be noted that a pattern emerges which repeats every four input data points. The pattern may be characterized as

1-B-B , 1-2-B , 1-B-B , 1-2-3

where 1,2, and 3 refer to decimation at levels 1,2, and 3 respectively and B refers to a blank segment of time which is available for processing Levels 4 and higher as required. Further note that: 1) Level 1 is processed odd on the first and third cycles, and even on the second and fourth; 2) Level 2 is processed odd on the second cycle and even on the fourth; 3) Level 3 is processed on the fourth cycle only, alternating even and odd for successive repetitions of the pattern. Utilization of these patterns and a set of flags which control decimation at Levels 4 and higher are the keys to control of cascade decimation as will be explained in a later section.

V. FOURIER TRANSFORMATION

A. Objective

Fourier coefficients are computed for two frequencies at each level by applying a Hanning window to 32 consecutive points at any level then applying sine and cosine transforms to get the Fourier coefficients for the real and imaginary components for the frequencies corresponding to the sixth and eighth harmoncs of that record. Thus, for a set of 32 consecutive points at level L: S_o, S_1, ... S_n , the Fourier coefficients for the frequencies

$$f_6 = \frac{6}{32T \times 2^L} \qquad (5\text{-}1a)$$

$$f_8 = \frac{8}{32T \times 2^L} \qquad (5\text{-}1b)$$

where T is the sample period, are computed as

$$C_{6real} = \sum_{n=o}^{31} S_n \frac{1}{2} \left[1 - \cos\frac{n\pi}{16}\right] \left[\cos\frac{6n\pi}{16}\right] \qquad (5\text{-}2a)$$

$$C_{6imag} = \sum_{n=o}^{31} S_n \frac{1}{2} \left[1 - \cos\frac{n\pi}{16}\right] \left[-\sin\frac{6n\pi}{16}\right] \qquad (5\text{-}2b)$$

$$C_{8real} = \sum_{n=o}^{31} S_n \frac{1}{2} \left[1 - \cos\frac{n\pi}{16}\right] \left[\cos\frac{8n\pi}{16}\right] \qquad (5\text{-}2c)$$

$$C_{8imag} = \sum_{n=o}^{31} S_n \frac{1}{2} \left[1 - \cos\frac{n\pi}{16}\right] \left[-\sin\frac{8n\pi}{16}\right] \qquad (5\text{-}2d)$$

The response of this operation may be evaluated by computing the Fourier coefficients for an input sequence

$$S_n = A\cos\frac{Kn\pi}{16} - B\sin\frac{Kn\pi}{16} \qquad (5\text{-}3)$$

where k represents one of the harmonics being evaluated, as follows:

$$C_{Kreal} = \sum_{n=o}^{31} \frac{1}{2}\left[A\cos\frac{Kn\pi}{16} - B\sin\frac{Kn\pi}{16}\right]\left[1 - \cos\frac{n\pi}{16}\right]\left[\cos\frac{Kn\pi}{16}\right] \qquad (5\text{-}4a)$$

$$= 8A$$

$$C_{Kimag} = \sum_{n=o}^{31} \frac{1}{2}\left[A\cos\frac{Kn\pi}{16} - B\sin\frac{Kn\pi}{16}\right]\left[1 - \cos\frac{n\pi}{16}\right]\left[-\sin\frac{Kn\pi}{16}\right] \qquad (5\text{-}4b)$$

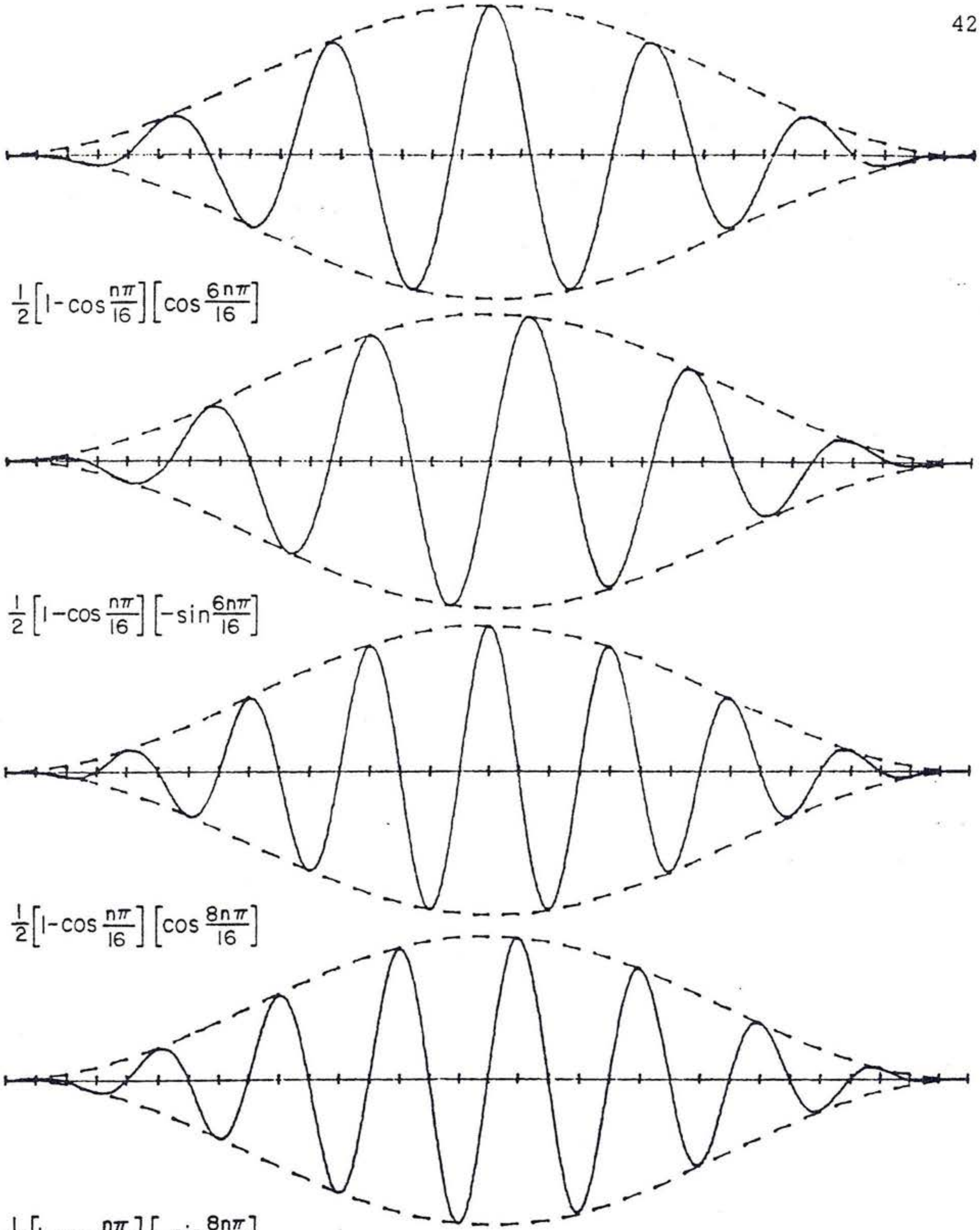
$$= 8B$$

Because these coefficients are computed from data to which a window of finite length has been applied, the coefficients actually represent the contribution of bands of frequency centered about the sixth and eighth harmonics. The percentage bandwidths for the sixth and eighth harmonics computed after application of a Hanning window may be shown to be 33 and 25 percent respectively, for any level (Blackman and Tukey, 1959). It should be noted that each set of Fourier coefficients is independent and therefore the power spectra estimates computed from those sets are independent. While 32 consecutive points are required to compute a set of Fourier coefficients for any level, the 32 point records may be adjacent or may be separated an arbitrary number of points- points which do not contribute to any spectral estimates.

B. Implementation

Four arrays of length 32 are used to store the factors by which the data values are multiplied as shown in equations (5-2a) through (5-2d). The functions represented by these coefficients are shown in Figure 9. An index register for each level counts the number of data points contributing to each set of coefficients and serves as an index for the stored sine and cosine arrays.

A new data point at any level will always be produced on an "even" cycle (see Figure 6). At that time, the value of that point is multiplied by the appropriate coefficients from the "cosine6" and "sine6" coefficient arrays and the result summed to the Fourier coefficient accumulators. Similar processing of the point using the "cosine8" and "sine8" arrays will be deferred until the next "odd" cycle since it is known that the data (in the a register) will not be affected by "odd" processing. The ability to distribute the multiplications for calculating the Fourier coefficients evenly over all points at a given level contributes a great deal to the efficient use of the processor by cascade decimation and signficantly reduces the worst case computing time for incoming data points,



$\frac{1}{2}\left[1-\cos\frac{n\pi}{16}\right]\left[\cos\frac{6n\pi}{16}\right]$

$\frac{1}{2}\left[1-\cos\frac{n\pi}{16}\right]\left[-\sin\frac{6n\pi}{16}\right]$

$\frac{1}{2}\left[1-\cos\frac{n\pi}{16}\right]\left[\cos\frac{8n\pi}{16}\right]$

$\frac{1}{2}\left[1-\cos\frac{n\pi}{16}\right]\left[-\sin\frac{8n\pi}{16}\right]$

Figure 9 - Functions Stored in Sine and Cosine Arrays

thus enabling a faster sample rate.

After both the sixth and eighth harmonic coefficients have been computed for the 32nd point of a data record at any level, a flag for that level is set signifying that a set of Fourier coefficients is complete and available for computation of power spectra estimates Until the spectra estimates are computed and the flag cleared, any data points which occur at that level are not used for computing Fourier coefficients. It will be recalled that decimation and computation of Fourier coefficients is implemented as an interrupt service routine. Computation of power spectra estimates by the main program takes place only in the time interval between completion of interrupt service for a data point and the occurence of the interrupt for the next point. Hence, the time required to service and clear a flag may vary, depending on the immediate demands being made on the processor.

In practice, flags are usually serviced and cleared before the occurrence of the next point at all but the lowest levels. Because of the abundance of data at these levels, service to these flags is the processor's lowest priority task and is done only when no higher levels require service. This fact, plus the relatively short time interval between occurrence of new points at these levels, result in the non-utilization of some points at these levels.

Figure 10 shows how a sequence of points at any level is used to compute independent sets of Fourier coefficients. Data records are indicated by the Hanning window functions applied to them.

C. Interlacing

At the upper levels of decimation, the time interval between adjacent points may become quite large. For a data run of reasonable length, the result is often a relatively small number of independent spectra estimates being used in computing the tensor impedances, resulting in statistical scatter in the impedance estimates.The number of spectra estimates obtained from a given sequence of data may be doubled at the expense of the estimates being largely, but no longer totally independent by interlacing records.

Note in Figure 10 that the contribution of data points near either end of record is very small. In order to glean more information from these points, one creates new records such that these points make a large contribution to the new records, but points which dominated the original records make only a small contribution. Figure 11 uses the Hanning window functions to illustrate how adjacent records may be interlaced in this manner.

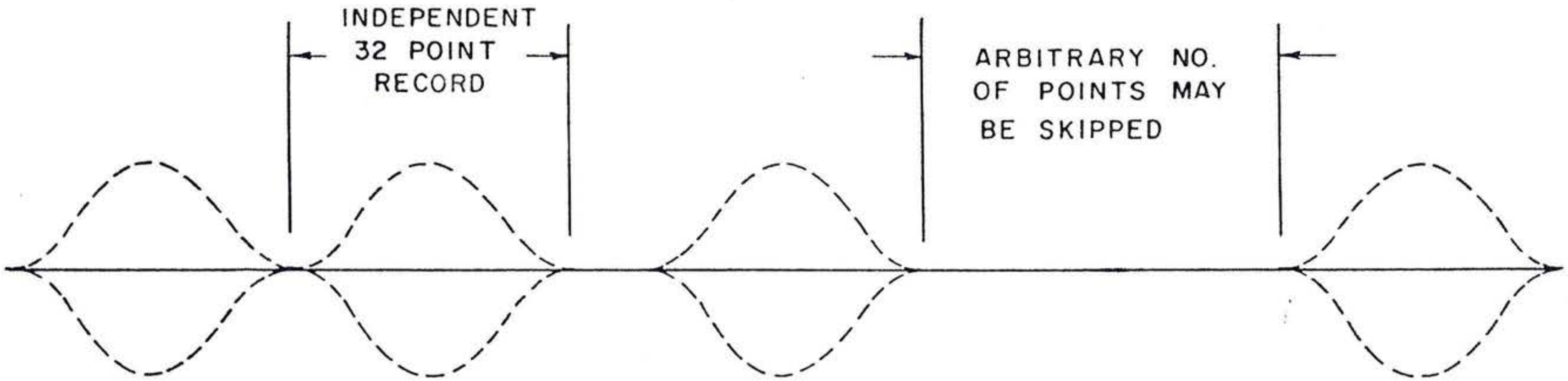


Figure 10 - Fourier Transform Records

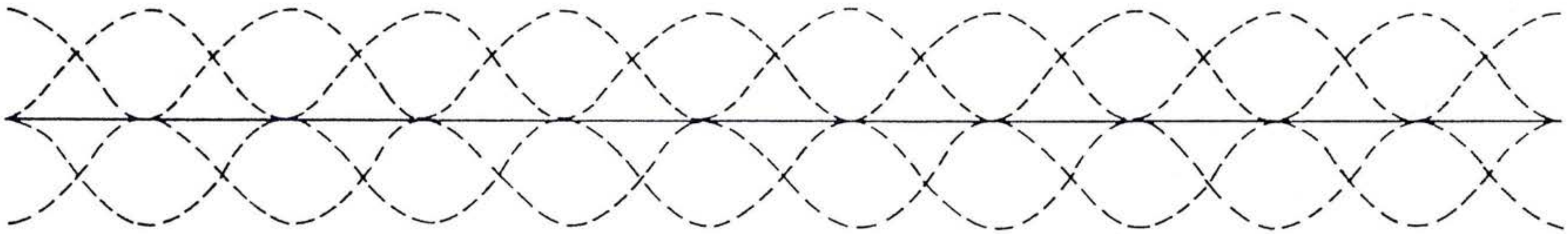


Figure 11 - Transform Records at Interlaced Levels

At levels which are interlaced, each data point contributes to two accumulators. It is multiplied by the appropriate factors from the first half of the stored coefficient arrays and the products accumulated in Fourier coefficient accumulator set one. It is also multiplied by the appropriate factors from the second half and the products summed to accumulator set two. After 16 points have been processed in this manner, the contents of set two are stored away for use in computing power spectra estimates, the contents of set one are transferred to set two, and set one is cleared. An index register for each level counts the number of points accumulated and serves as an index for the sine and cosine coefficient arrays. Each 16 points a new set of Fourier coefficients is computed and stored away, a flag is set signifying that a set of Fourier coefficients is available for computation of spectra estimates, and the index register is reset to repeat the process. Although interlaced levels may not skip data points as was done at other levels, this is no problem for three reasons: 1) Interlacing is utilized only at levels where the time between adjacent points is quite large; 2) Flags at interlaced levels receive high priority service by the processor's main program; and 3) Because the completed coefficients are stored in registers separate from where they were calculated, the processor has up to 16 cycles to service and clear the flag.

VI. COMPUTATION OF SPECTRA ESTIMATES

A. Priorities for Computing Spectra

The processes of decimation and computation of Fourier coefficients are implemented as part of the interrupt service which occurs each time a new set of data is digitized and input to the calculator. The period of time between completion of this interrupt service and the initiation of the succeeding interrupt is used to compute auto and cross power spectra estimates for any level or levels which have complete sets of Fourier coefficients waiting to be processed.

Because coefficients may be complete and waiting at several levels, a set of priorities must be established for the order of service ("service" meaning computation of power spectra estimates). Interlace levels must be serviced before the occurrence of 16 points at that level or the interlace routine will destroy good data and be unable to recover. Therefore, highest priority is given to servicing interlaced levels, beginning with the lowest level number since the period between points at that level is shortest. If no interlaced levels require service, other levels are serviced, beginning with the highest non-interlaced level. Since service to non-interlaced levels may be delayed indefinitely without disrupting program operation, levels where the least data occurs are given highest priority.

It must be remembered that while a given level is being serviced, numerous interrupts may occur and in the course of processing these, new sets of Fourier coefficients may have been completed. Hence, after computing a set of spectra estimates at any level, the processor rechecks all levels in order of priority as given above. If none require processing, an operator setable flag which indicates "end of run" is checked. This prioritized checking of all levels repeats indefinitely.

Each set of Fourier coefficients at level L is contributed to by 32 x 2^L data points. Any data points having a value of plus or minus the full scale value of the digitizer are referred to as a "digitizer saturation". Signal levels in the analog circuitry ahead of the digitizer are such that digitizer saturation occurs at lower levels than saturation of the circuitry and its resulting system nonlinearities. A counter is associated with each level which keeps track of the number of saturated data points which contribute to each set of Fourier coefficients completed. Before a set of spectra estimates are computed, the number of saturated points contributing to the completed Fourier coefficients is compared to the number of saturations which will be tolerated at level as preset by the operator. Too many saturations cause that set of Fourier coefficients to be discarded. In practice, no saturations are allowed at the lowest levels. At higher levels however, the large number of data points contributing to each records and the length of time required to make up for discarded data lead to a number of allowed saturations, representing a compromise between data quality and practical considerations.

B. Auto and Cross Power Spectra Estimates

Discussion of the decimation process and calculation of Fourier coefficients has generally dealt with only one channel. The reader is reminded that the system being dealt with has five channels (see Figure 2). Computation of impedance estimates and data coherencies requires knowledge of the auto and cross power spectra for the five channels. From the Fourier coefficients for the real and imaginary parts of each of the five channels, the five auto and ten cross power spectra estimates may be calculated in a straight-forward manner.

$$(E_x E_x^*)_r = (E_{xr} E_{xr} + E_{xi} E_{xi}) \tag{6-1a}$$

$$(E_x E_y^*)_r = (E_{yr} E_{xr} + E_{xi} E_{yi}) \tag{6-1b}$$

$$(E_x E_y^*)_i = (E_{xi} E_{yr} - E_{xr} E_{yi}) \tag{6-1c}$$

$$(E_x H_x^*)_r = (H_{xr}E_{xr} + E_{xi}H_{xi}) \tag{6-1d}$$

$$(E_x H_x^*)_i = (E_{xi}H_{xr} - E_{xr}H_{xi}) \tag{6-1e}$$

$$\vdots$$

$$(H_y H_z^*)_r = (H_{zr}H_{yr} + H_{yi}H_{zi}) \tag{6-1f}$$

$$(H_y H_z^*)_i = (H_{yi}H_{zr} - H_{yr}H_{zi}) \tag{6-1g}$$

$$(H_z H_z^*)_r = (H_{zr}H_{zr} + H_{zi}H_{zi}) \tag{6-1h}$$

When any level is processed, estimates for each of the spectral terms are computed from the appropriate Fourier coefficients as shown in Figure 12 for both the sixth and eighth harmonics. The values are computed and summed into separate registers which accumulate each of the power spectra components for each frequency. A register for each level counts the number of independent spectra estimates accumulated at that level and after the run is complete is used to normalize the power spectra estimates.

Although only two independent spectra estimates are theoretically used to compute the magnetotelluric tensor impedances, additional estimates are required to reduce the statistical scatter to an acceptable level. It may be shown that the scatter between the true spectra and the mean of the spectra estimates decreases as a larger number of independent estimates are incorporated in the

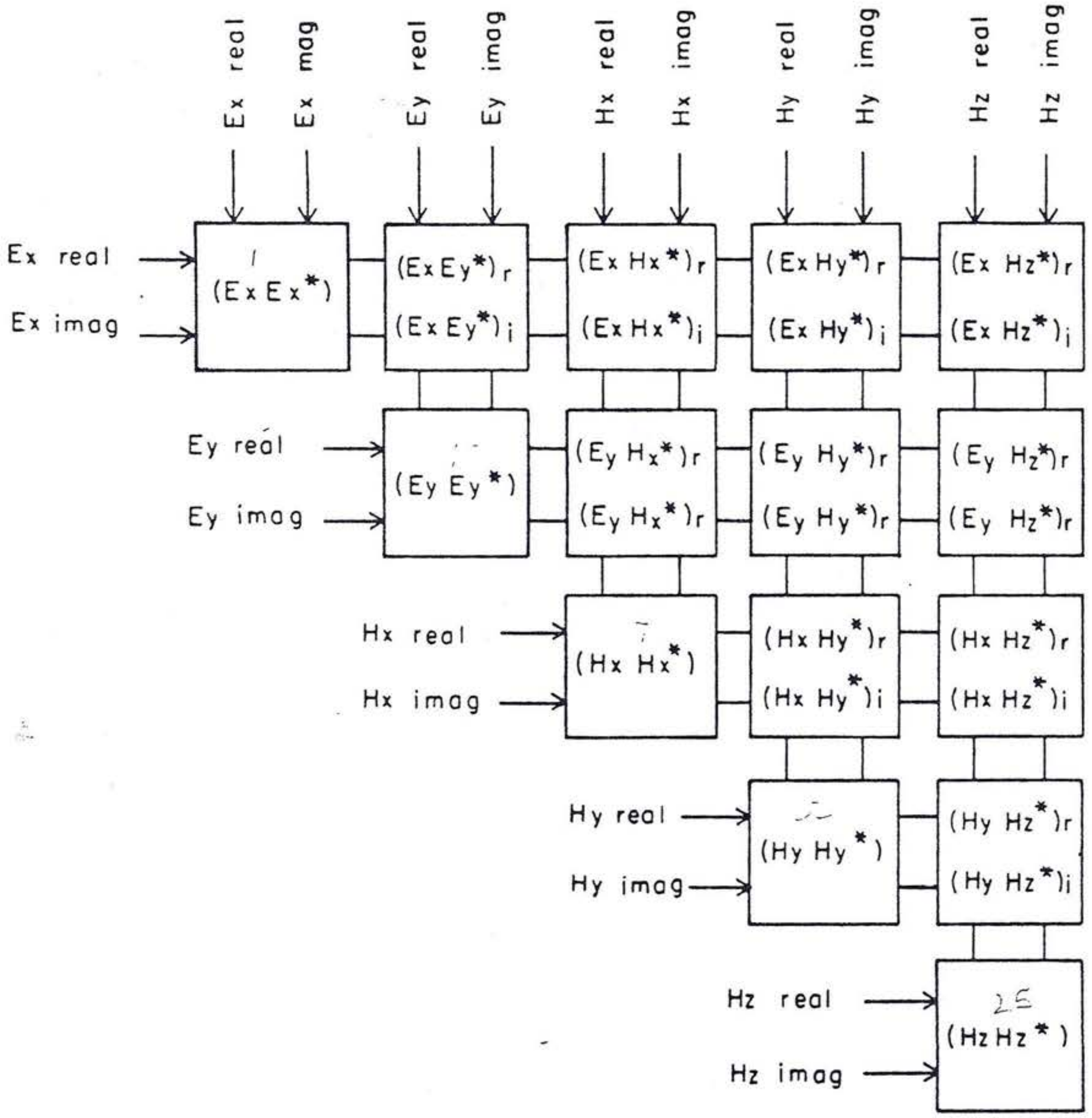


Figure 12 - Computation of Auto and Cross Power Spectra

mean. Hence, the larger the number of independent estimates, the more accurate the mean spectra estimates. The minimum number of estimates accepted during the analysis of two MT surveys during the summer of 1977 was usually seven estimates from interlaced records.

VII. THE IMPLEMENTATION OF CASCADE DECIMATION

A. The Hewlett-Packard 9825 Programmable Calculator

The 9825 programmable calculator was introduced by Hewlett-Packard in 1975. Its powerful,scientifically oriented language, its proficiency for input and output (I/O) operations, and its relatively low cost led to its selection for use in the University of Texas' real time MT system. Hewlett-Packard's high level language, "HPL" is an interpreter based language best described as a cross between BASIC and FORTRAN. Portions of the program were originally written and tested in FORTRAN. These were quickly and easily transformed into HPL for field use on the HP-9825. The use of a high level language such as HPL and the powerful editing features of the 9825 were invaluable during the highly evolutionary period of program development.

The University of Texas system uses a HP-9825 calculator equipped with 24K 8-bit bytes of semiconductor memory. A high speed mini-cartridge tape drive, alpha-numeric display, and thermal printer are part of the calculator. Also included in the system are read-only-memory (ROM) modules for advanced programming, general and extended I/O, string variable processing, and plotter control. A 16 bit parallel interface for data input from the digitizer, and an X-Y digital plotter for plotting results complete the system.

B. Main Program- Spectra Computation

The primary task of the main program for decimation is to scan for flags which signal that a set of Fourier coefficients is ready for computation of auto and cross power spectra.The order of the scanning establishes the priorities for computing spectra as explained in section VI-A. The program then calculates estimates for the auto and cross power spectra for the appropriate level as discussed in section VI-B. It also checks a flag which is set by the operator to terminate the run.

Prior to the start of a data run, the main program prompts the operator for and accepts site data such as date, site name, run number, and electrode line lengths and azimuths. From the sample period entered by the operator it computes an array of frequencies corresponding to the spectra produced. Also entered by the operator are the settings of the analog filters ahead of the digitizer and the allowable number of saturations. for each level.

After the run has begun, the number of independent spectra estimates at each level is displayed on the calculator. When the minimum acceptable number of estimates are completed at the highest level, the operator terminates the run by pressing one of the special function keys on the HP-9825 which has previously been programmed to set the appropriate flag to terminate the run.

Following termination of the data run, the spectra are normalized for the number of estimates at each level and the response of the decimation and Fourier transformation processes (equations 4-6 and 5-4). The operator then has an opportunity to enter any comments regarding data quality, equipment malfunctions, etc. These comments, as well as site information, filter settings, sample period, allowable saturations, number of estimates at each level, and all of the spectra and their respective frequencies are recorded on digital tape.

The spectra may be printed out for inspection if desired. If on site determination of tentative resitivity profiles is desired, another program executed on the HP-9825 removes the response due to the sensors and analog system, then uses the corrected spectra to compute the apparent resistivity curves which may then be inverted to yield resistivity versus depth profiles. The procedure for this is briefly outlined in section II-D.

A flow chart depicting the general operation of the main program and one showing how auto and cross spectra estimates are computed for any level are included in Appendix 1. The actual coding which implements the program may be found in Appendix 2.

C. Interrupt Service - Decimation

It should be clear by now that a majority of the computation associated with the process of cascade decimation takes place during the processiing of interrupts. Interrupt requests are serviced by first inputing data points for each of the five channels from the digitizer. Each point is checked for a digitizer saturation and saturation counters at all levels are incremented if the data on any channel is full scale.

The sequence of input data points S_0 must have both the sixth and eighth harmonic coefficients computed each time since there is no "even" or "odd" processing associated with level 0. Levels 1, 2, and 3 are processed according to the pattern noted in section IV-C, where cycle numbers 0, 1, 2, or 3 refer to the four possible cases. Appendix 1 includes a flow chart which shows how this pattern is utilized to process the lower levels. If less than the maximum number of three allowable decimations are performed at the lower levels, level 4 and higher may be processed according to the three states corresponding to: 1) needs odd processing, 2) needs even processing, and 3) new point ready, are used to control the processing of these levels.

Even processing, odd processing, computation of the sixth and eighth harmonics at interlaced and non-interlaced levels are all implemented as subroutines whose only calling parameter is L, the level to which they are to be applied. The sequence for calling these subroutines is completely determined by the two algorithms just presented - one for levels three and below which utilizes a repeating pattern, and another for levels greater than three which utilizes a set of three-state flags.

Flowcharts for each of these subroutines are included in Appendix 1. Regarding the terminology used for computation of Fourier coefficients, the following conventions are used: The stored sine and cosine registers of length 32 are referred to as C_6, S_6, C_8, and S_8 and individual elements within the arrays are addressed by index I . The real and imaginary Fourier coefficients for the sixth and eighth harmonic non-interlaced level L are computed in registers referred to as RE_{6L}, IM_{6L}, RE_{8L}, and IM_{8L}. At interlaced levels, these same designations refer to the registers where completed Fourier coefficients are stored. At these levels, the coefficients are computed in two sets of registers as explained in section V-C. The registers corresponding to the first set are referred to as RE^1_{6L}, IM^1_{6L}, RE^1_{8L}, and IM^1_{8L}, and the to second set as RE^2_{6L}, IM^2_{6L}, RE^2_{8L}, and IM^2_{8L}.

The actual program which implements the decimation and filtering process is included in Appendix 2.

VIII. CONCLUSIONS

A. Performance of Cascade Decimation

The performance of cascade decimation when given idealized data can easily be evaluated by applying artificial data - data made up of a sum of cosine waves of known magnitudes and frequencies. Computing a data sequence of Sn according to

$$S_n = \sum_{i=o}^{L} \cos \frac{6n\pi}{32 \times 2^i} + \cos \frac{8n\pi}{32 \times 2^i} \qquad (8\text{-}1)$$

for all five channels and analyzing the sequence using cascade decimation produces the results shown in Table 1.

However, field data is far from ideal. It is made up of a continuous distribution of frequency components rather than discrete frequencies. Unless properly pre-whitened, the limited dynamic range of the digitizer may cause spectral estimates in frequency bands of low energy to be of low quality. Data quality, however, may be estimated by careful analysis of the spread in the various tensor impedance estimates, the coherencies between data channels and the repeatability in time of impedance estimates, taking into account indicators of the two or three dimensionality of the local geology.

TABLE 1

Results of Cascade Decimation Applied to Artificial Data

Sample Period T = 0.5 Seconds

$$\text{Data Sequence } Sn = \sum_{i=0}^{9} \cos \frac{6n\pi}{32 \times 2^i} + \cos \frac{8n\pi}{32 \times 2^i}$$

* Denotes Interlaced Levels

Freq.	Lev. # - Harm. #	No. Sample Points	Theoretical Auto Spectra ExEx*	Calculated Auto Spectra ExEx*
.50000	0-8	2203	1.000000	1.000010
.37500	0-6	2203	1.000000	1.000074
.25000	1-8	1101	1.000000	1.000016
.18750	1-6	1101	1.000000	1.000099
.12500	2-8	550	1.000000	1.000014
.09375	2-6	550	1.000000	1.000092
.06250	3-8	274	1.000000	1.002539
.04688	3-6	274	1.000000	1.000097
.03125	4-8	136	1.000000	1.003177
.02344	4-6	136	1.000000	1.000092
.01563	5*-8	136	1.000000	1.003174
.01172	5*-6	136	1.000000	0.9998655
.00781	6*-8	67	1.000000	1.003168
.00586	6*-6	67	1.000000	0.9998837
.00391	7*-8	33	1.000000	1.002926
.00293	7*-6	33	1.000000	0.9994385
.00195	8*-8	16	1.000000	1.002945
.00146	8*-6	16	1.000000	0.9995602
.00098	9*-8	7	1.000000	1.002874
.00073	9*-6	7	1.000000	0.9994499

Cascade Decimation was tested in the field during the summer of 1977. Thirty-five MT sites were completed as part of a survey of geothermal areas in southwestern Utah. Measurements at fifteen additional sites in the Snake River plain area of eastern Idaho were made as part of the Electrical Geophysics Laboratory's program for developing new instrumentation and analysis techniques for geothermal exploration. A wide variety of conditions were encountered including some areas of extremely three dimensional geologies. As evaluated by personnel of the Electrical Geophysic Research Lab, the data quality achieved during the surveys mentioned surpasses that of any prior survey. While this improvement in data quality may be largely credited to improvements in data acquisition hardware, careful work by an experienced crew, and a better understanding of the intricacies of analyzing and interpreting MT data, it is important to note that the quality of the spectral estimates obtained using cascade decimation equalled that of those obtained on previous surveys using techniques based on the FFT algorithm.

The availability of preliminary results during the survey proved to be a valuable aid in determining the location of future sites to yield the most information about the features of interest. This represents a great improvement over the procedure for prior surveys when time series data was recorded in the field and analyzed after completion of the field work using the large central computer at the University of Texas.

Data from 5 Hz to 3 kHz was analyzed using a phase sensitive dual hetrodyne receiver which sequentially analyzes one frequency at a time. The audio MT (AMT) system operates under control of a Hewlett-Packard 9810 programmable calculator,a predecessor of the HP-9825.

Data from .001465 Hz to .5 Hz was analyzed in real time using cascade decimation implemented on the HP-9825. Nine levels of decimation were used with the upper five levels interlaced. Table 1 shows the resulting frequency distribution. Data from Level 9 was generally discarded because an insufficient number of samples could be acquired in the allotted time to obtain the statistical confidence desired for the spectral estimates. It is interesting to note that by computing the spectra in the field, the data volume was reduced to such a degree that all of the data acquired from a 50 MT site survey could be recorded on one DC100A mini-datacartridge.

Because 0.5 second is the maximum sample rate at which real time cascade decimation will operate on the HP-9825, data in the range from .46875 to 5.0 Hz was analyzed by digitizing and recording blocks of data lasting slightly more than a minute, then analyzing them using cascade decimation at a rate set by the execution time of the cascade decimation algorithm. Four levels of data were used, with all of them interlaced.

Thus, it was possible to obtain spectra estimates while still on site for the entire frequency range from .001 Hz to 3 kHz. Tensor impedance estimates, apparent resistivities, and even tentative resisitivity versus depth soundings could be produced while still on site if desired. It is believed that the University of Texas real time, broad-band MT system is the most advanced system presently available for real time, on site processing of magnetotelluric data. Further, the method of cascade decimation has proven to be not only feasible for the processing of MT data, but in many ways superior to FFT based methods now in widespread use.

B. Comparison of Cascade Decimation and FFT

The computation of power spectra for MT data has in recent years been totally dominated by methods which make use of the fast Fourier transform (FFT) algorithm for computation of the frequency components for the various data channels. Data is usually recorded on either digital or analog tape and later analyzed at a large computational facility. Practical computational constraints generally limit the size of FFT records to 4096 points. A Hanning window is usually applied to each data record to lessen the effects of the finite record length. For each record, Fourier coefficients are computed for the fundamental and all of its harmonics up to the Nyquist frequency. After coefficients have been computed for each component channel, independent auto and cross power spectra estimates are computed for each of the harmonics. Groups of spectra estimates at adjacent harmonics are then averaged to produce spectra estimates representing aproximately equal percentage bandwidths. The lowest harmonics are discarded due to the unacceptably large percentage bandwidths which result from averaging two of more adjacent spectra.

Assume that a minimum of four independent samples are desired for statistical confidence in the spectral estimates and that the desired percentage bandwidth is about 30 percent. These criteria are typical and may be satisfied by use of two FFT records of 4096 points each which are taken independently in time with the spectra estimates computed as outlined below. In order to deal with actual frequencies, consider the data to be sampled at 40 Hz. Table 2 shows how groups of adjacent harmonics are averaged to produce constant Q power spectra estimates.

If no Hanning window was applied to the data record prior to application of FFT, the bandwidths of the Fourier coefficients (as represented by the primary lobe of its (sin x)/x shape) would be as shown in Figure 13. Application of a Hanning window, however, doubles the

Table 2 - FFT Harmonics Averaged to Give Constant Percentage Bondwidth Spectra

4096 point transform T=.025 second

Frequency	Harmonics Averaged	Percentage Bondwidth	No. Indep. Samples
.11719	11,12,13	33	2
.14648	15,16,17	25	2
.20508	19-23	28	3
⋮	⋮	⋮	⋮
7.6953	604-894	25	101
10.000	896-1152	25	129

Table 3 - Six Levels of Cascade Decimation

102.4 seconds of data T=.025 second

Frequency	Level No. - Harm No.	Percentage Bondwidth	No. Indep. Samples
.11719	6-6	33	2
.15625	6-8	25	2
.23438	5-6	33	4
.31250	5-8	25	4
⋮	⋮	⋮	⋮
7.5000	0-6	33	128
10.000	0-8	25	128

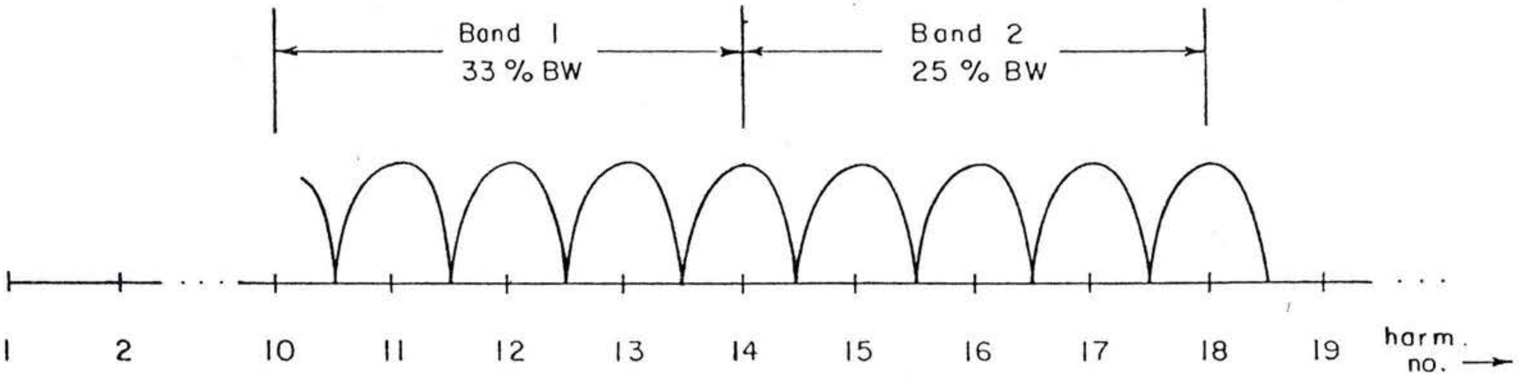


Figure 13 - Constant Q FFT Averages

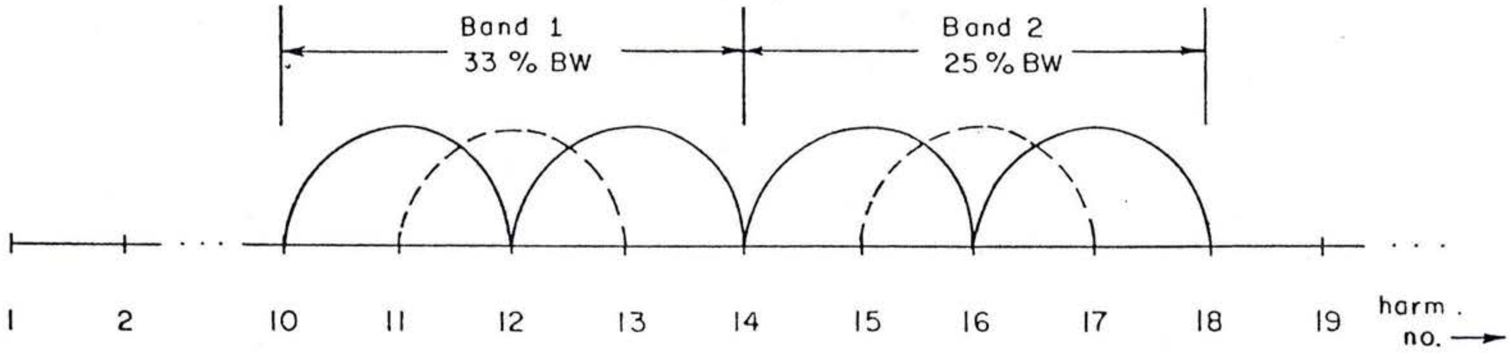


Figure 14 - Constant Q FFT Averages with Hanning Window

bandwidth of each harmonic as shown in Figure 14. For the first group averaged, the resulting percentage bandwidth is 33 percent and for the second is 25. A larger number of harmonics is averaged for each subsequent band so that the percentage bandwidth remains approximately constant. As indicated in Table 2, as many as 257 harmonics are averaged for the highest bands. For the first two bands, three harmonics may be averaged. However, the result represents only two independent samples in the frequency domain. Thus, two FFT records which are taken independently in time give four independent samples at the lowest frequency. At higher frequencies an increasing number of independent samples are used up to a maximum of 258 from two FFT records.

Spectra estimates over the same frequency range may be obtained by using six levels of cascade decimation as shown in Table 3. When applied to the same number of data points, FFT and cascade decimation give estimates at the lowest frequencies which represent the same percentage bandwidths and the same number of independent samples. At all other frequencies, the bandwidth and number of samples are comparable. Start-up transients result in the first sample at all levels being discarded, but interlacing the lower levels doubles the number of samples incorporated in each estimate. For this example, the result would be seven samples being used for the lowest frequencies.

Use of FFT as outlined above requires that 4096 data points for each channel be recorded for each of the two records. The amount of computer memory required for implementing FFT varies for different computer programs. The requirement for most falls in the range of 2N to 4N for N data points although less storage may be used if one is willing to expend more computational effort. Thus, a computer having at least 8 to 16,000 memory locations is required for implementation of the FFT algorithm for 4096 points. The number of multiplications may be computed as $N\log_2 N$, or 49,152 for each FFT record.

Cascade decimation, on the other hand, is a continuous process which does not require the mass storage of raw data prior to processing. The computer memory requirement for each level of cascade decimation is as follows: decimation registers- 3; accumulators for Fourier coefficients- 4; index for sine and cosine arrays- 1; decimation and spectra flags- 2; saturation allowed and saturation counter registers- 2; and if the level is interlaced, accumulators for the 1st and 2nd half of each interlaced record- 8. Additionally, 128 registers are required for storage of the sine and cosine arrays. The total storage requirement for six levels of cascade decimation will be less than 250 memory locations. Additional memory will be required for the storage of the power spectra, but this requirement will be the same for both methods. To cover the same frequency range as an N point FFT record, the number of levels of cascade decimation required is given as

$$L \geq \log_2 \left(\frac{N}{64} \right) \tag{8-2}$$

The memory requirement for computation of Fourier coefficients by FFT and by cascade decimation may be shown to be

FFT: 2N to 4N (8-3)

Cascade Decimation:

$$128 + 12 \times \text{integer} \left[\log_2 \left(\frac{N}{64} \right) + 1 - \varepsilon \right] + 8I$$

where I is the number of levels which are interlaced and ε represents a vanishingly small number.

The processing of each incoming data point at the first level requires one multiplication for either even or odd processing, four multiplications for computation of both the sixth and eighth Fourier coefficients at Level 0, and two multiplications for computation of the sixth or eighth at Level 1. On the average, the total number of multiplications for all levels below the first will be less than three if none of these levels are interlaced and less than five if all are interlaced. Therefore, for the worst case- all levels below the first interlaced, the average number of multiplications per point will be less than 12. However, if all but the first three levels are interlaced, the average number of multiplications per point drops to less than 10.5. For the example cited above, FFT would require a total of about 98,300 multiplications. Cascade decimation implemented with all but the lowest three levels would require about 86,000. Additional multiplications would be required for computation of the power spectra estimates, but the number would be the same for both methods. In general, the number of multiplications required for computation of Fourier coefficients by FFT and by cascade decimation are given as

FFT: $N \log_2 N$

(8-4)

Cascade Decimation:

$$4 + \sum_{n=1}^{L} \frac{3}{n} + \sum_{L-I+1}^{L} \frac{2}{n}$$

where L is the number of decimation levels required to cover the same frequency range as an N point FFT record (equation 8-2) and I is the number of levels which are interlaced. If no levels are interlaced, only the first two terms are considered.

The advantages of cascade decimation become most apparent when a broader range of frequencies is to be investigated. When FFT is utilized as described above, the frequency ratio which can be investigated is about 85:1. A broader frequency range may be covered by recording additional data records using a different sample period. If in the example above additional data were desired for frequencies extending down to .001 Hz, two additional FFT records could be recorded and processed exactly as before except that the sample period for this data would be 2.93 seconds. The lowest frequency evaluated from these records would be .001 Hz and the highest .0853 Hz.

Extending the lower frequency bound for cascade decimation simply involves adding the more levels of decimation. Once cascade decimation is operational at a given sample rate, the additional computation resulting from adding more levels is negligible since points at higher levels occur very seldom relative to the occurence of points at the first few levels. A small time savings in total data acquisition time may be realized by the simultaneous processing of the entire frequency range rather than the sequential evaluation of two or more frequency bands. The time required for evaluation of the lowest frequencies, however, is so dominant that relatively little time is saved.

C. Future Work

Development of a practical real time MT system is presently one of the Electrical Geophysics Laboratory's most vigorous efforts. Data acquired during the summer of 1977, being the most voluminous and the highest quality ever achieved by this laboratory, is being used for statistical studies of the measured spectra for the natural fluctuations in the earth's electric and magnetic fields. The goal of this effort is to determine the appropriate digital representation for data in the cascade decimation process for maximum procesing speed consistant with the significant accuracy of the rest of the MT system. Attention is also being given to additional hardware which can increase the efficiency of cascade decimation, such as the addition of a queue for incoming data points which could eliminate much of the control algorithm presently required.

As implemented on the HP-9825 for a five channel system, the maximum sample rate is 2 Hz. The primary limitation is the large time (avg. 11.5 ms) required for 64-bit floating point multiplies. Although the high level language of the HP-9825 was almost essential to the development of such a program, a much greater throughput rate can easily be achieved by implementing the developed concepts in machine language on a modest 16-bit digital processor. Second generation realization of cascade decimation is being implemented in this manner. Present system design specifications call for real time processing of up to 15 data channels at a sample rate of 20 Hz. The addition of remote E-field sensors with data telemetered back to the base station will make possible a much larger number if sites per day for large scale MT surveys.

The design and construction effort for this new system is underway and it is scheduled to be completed and tested by the end of the summer, 1979. Design of both the hardware and software for this system are drawing heavily on the concepts developed, and pitfalls discovered while implementing cascade decimation on the HP-9825.

APPENDIX 1

Flowcharts for Program DECIMATE

A) Main Program

B) Computation of Power Spectra Estimates

C) Interrupt Service Program

D) Decimation Control for Levels 4 and Higher

E) Odd Processing

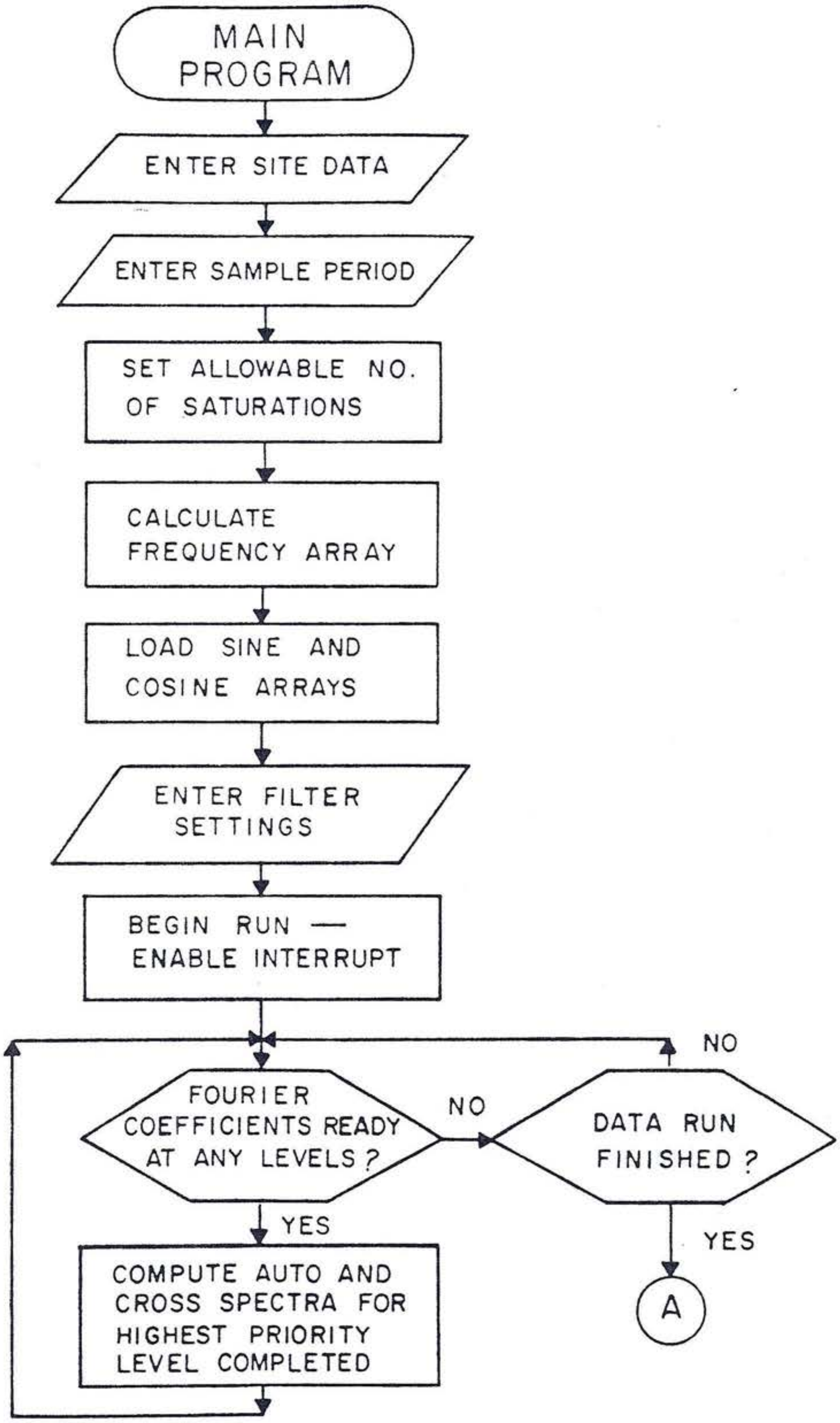
F) Even Processing

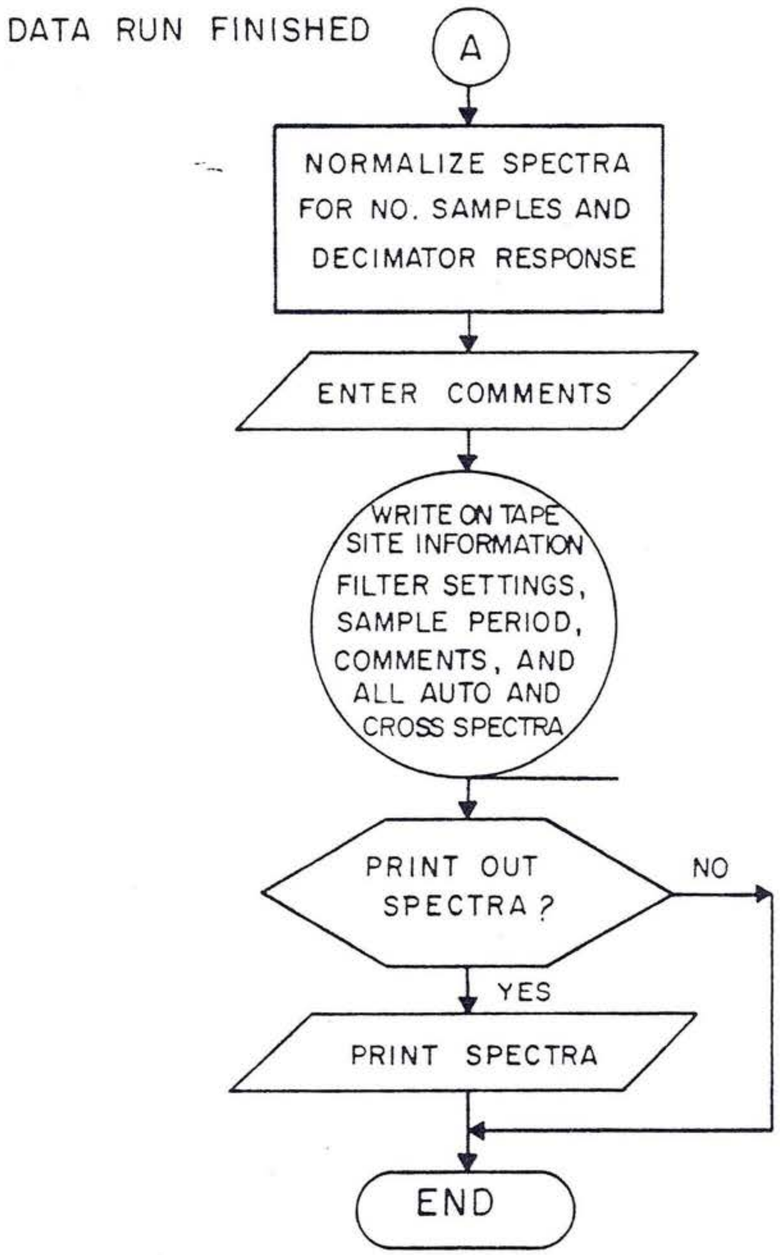
G) Computation of 6th Harmonic

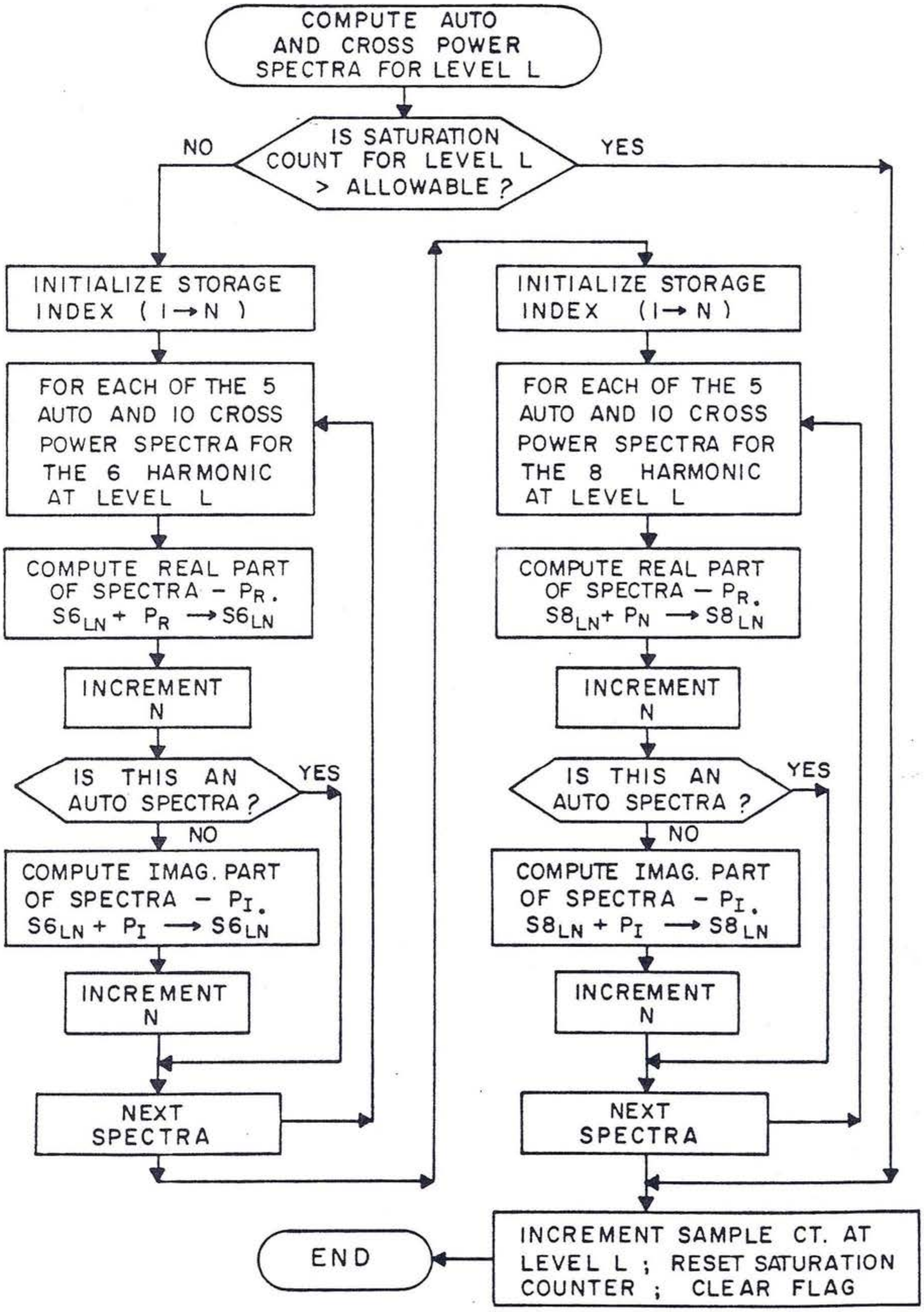
H) Computation of 8th Harmonic

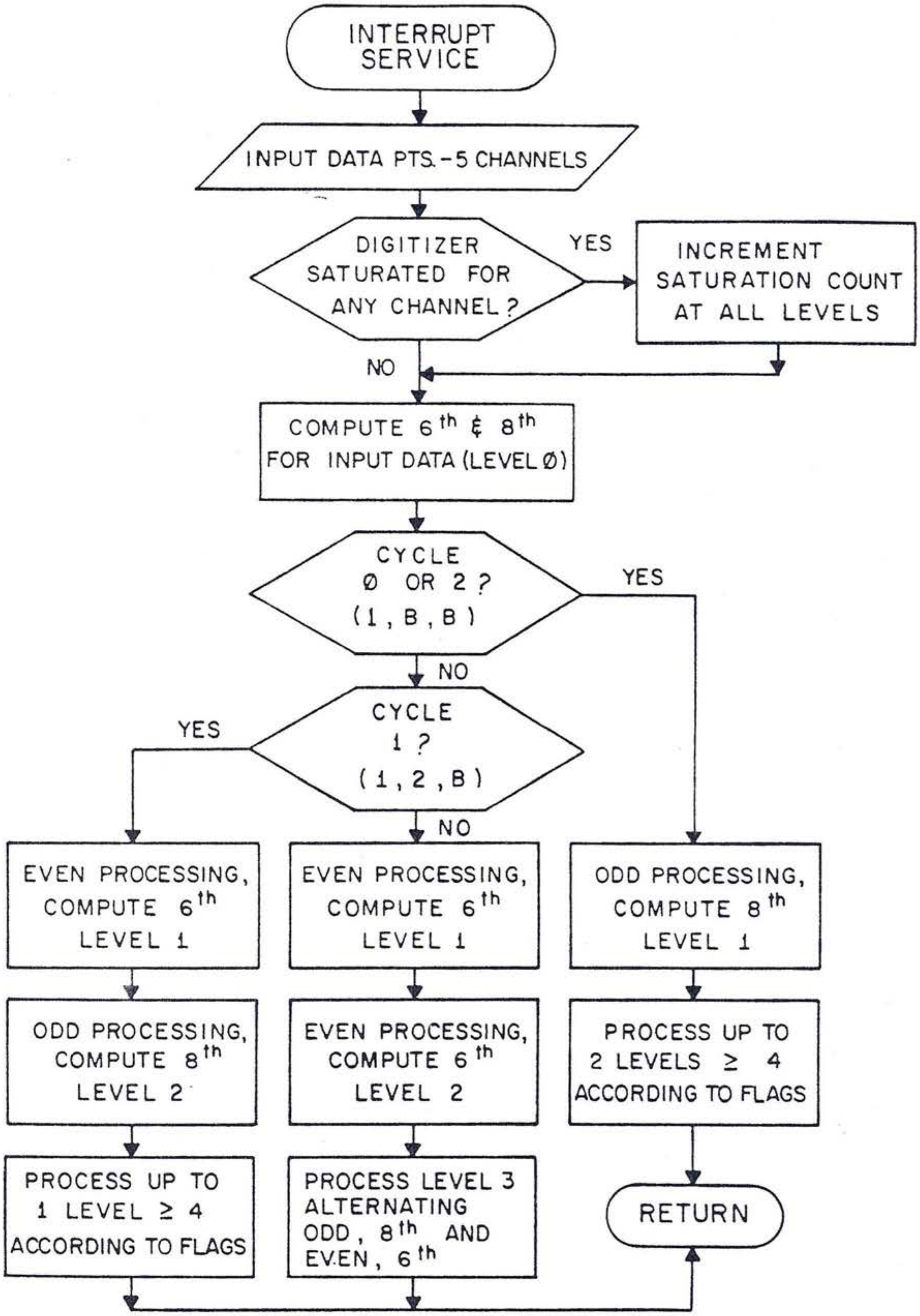
I) Computation of 6th at Interlaced Levels

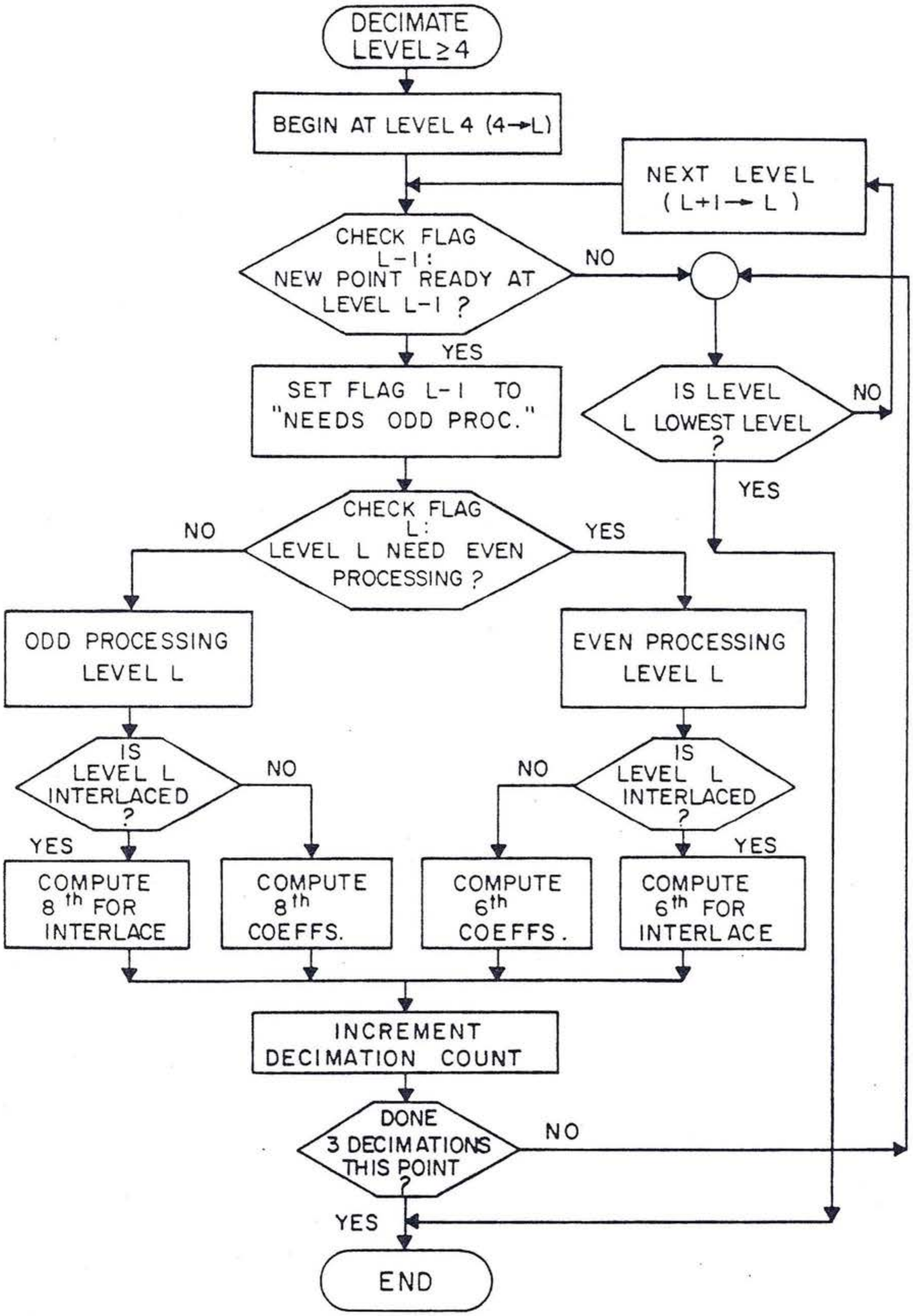
J) Computation of 8th at Interlaced Levels

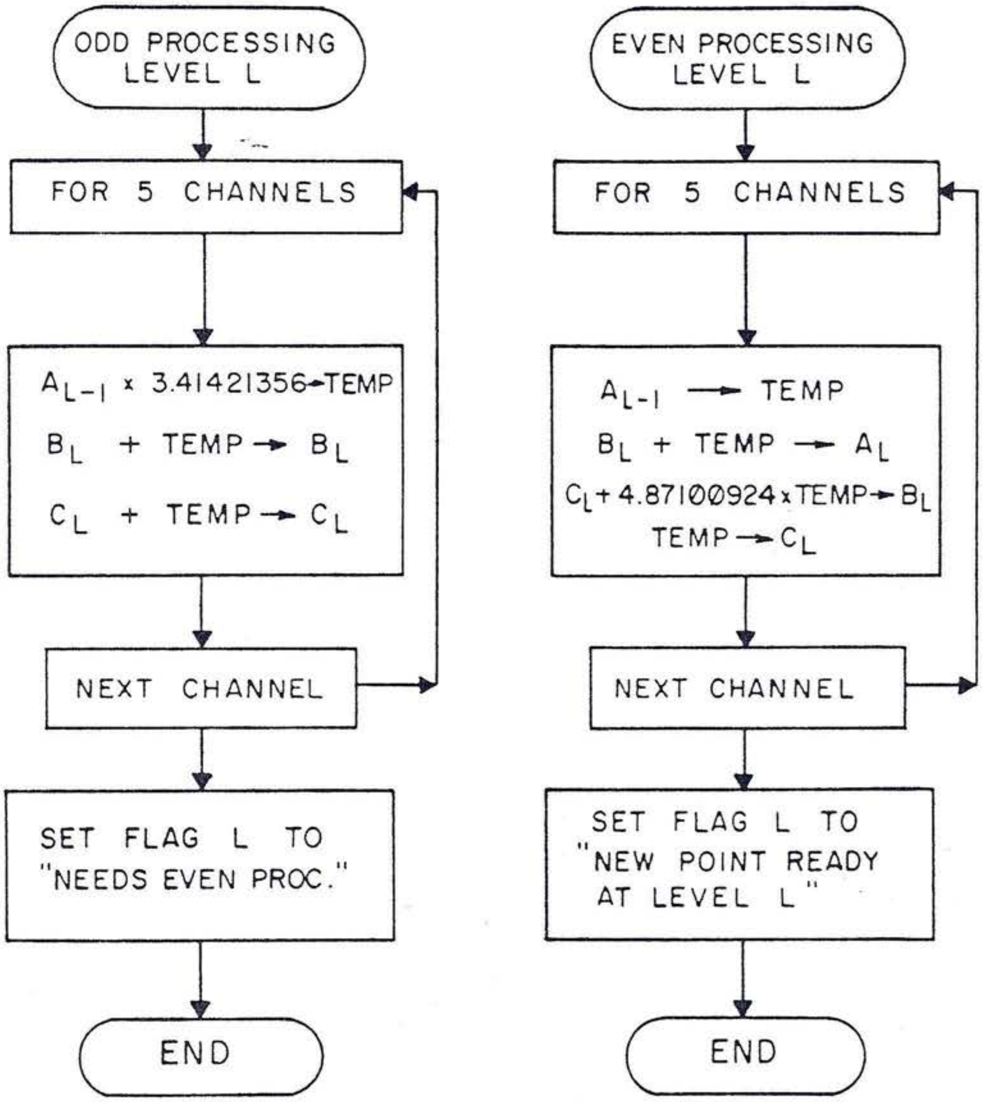

MAIN PROGRAM
ENTER SITE DATA
ENTER SAMPLE PERIOD
SET ALLOWABLE NO. OF SATURATIONS
CALCULATE FREQUENCY ARRAY
LOAD SINE AND COSINE ARRAYS
ENTER FILTER SETTINGS
BEGIN RUN — ENABLE INTERRUPT
FOURIER COEFFICIENTS READY AT ANY LEVELS?
NO
DATA RUN FINISHED?
NO
YES
COMPUTE AUTO AND CROSS SPECTRA FOR HIGHEST PRIORITY LEVEL COMPLETED
YES
A

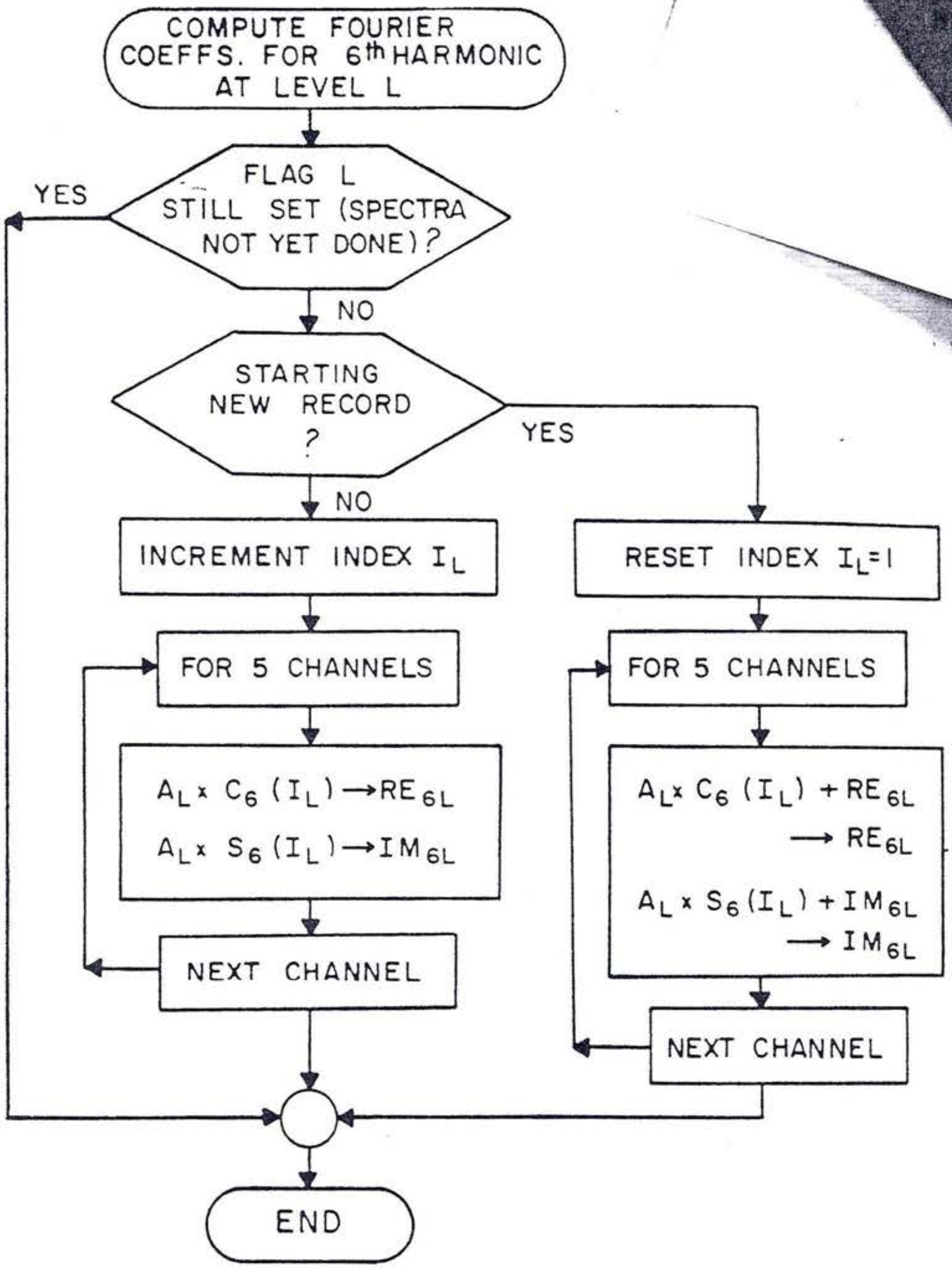

DATA RUN FINISHED
A
NORMALIZE SPECTRA FOR NO. SAMPLES AND DECIMATOR RESPONSE
ENTER COMMENTS
WRITE ON TAPE SITE INFORMATION FILTER SETTINGS, SAMPLE PERIOD, COMMENTS, AND ALL AUTO AND CROSS SPECTRA
PRINT OUT SPECTRA ?
NO
YES
PRINT SPECTRA
END

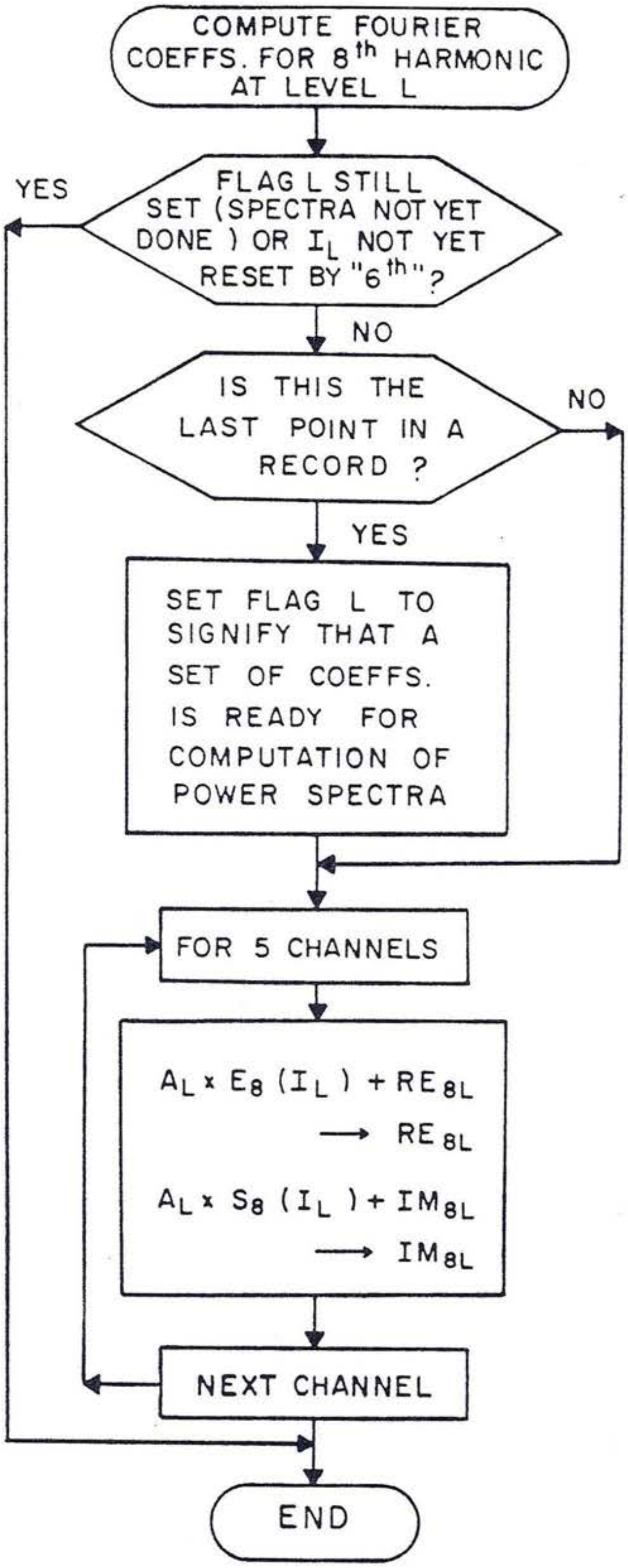

COMPUTE AUTO AND CROSS POWER SPECTRA FOR LEVEL L
IS SATURATION COUNT FOR LEVEL L > ALLOWABLE ?
NO
YES
INITIALIZE STORAGE INDEX (I→N)
FOR EACH OF THE 5 AUTO AND 10 CROSS POWER SPECTRA FOR THE 6 HARMONIC AT LEVEL L
COMPUTE REAL PART OF SPECTRA – P_R. $S6_{LN} + P_R \rightarrow S6_{LN}$
INCREMENT N
IS THIS AN AUTO SPECTRA ?
YES
NO
COMPUTE IMAG. PART OF SPECTRA – P_I. $S6_{LN} + P_I \rightarrow S6_{LN}$
INCREMENT N
NEXT SPECTRA
INITIALIZE STORAGE INDEX (I→N)
FOR EACH OF THE 5 AUTO AND 10 CROSS POWER SPECTRA FOR THE 8 HARMONIC AT LEVEL L
COMPUTE REAL PART OF SPECTRA – P_R. $S8_{LN} + P_N \rightarrow S8_{LN}$
INCREMENT N
IS THIS AN AUTO SPECTRA ?
YES
NO
COMPUTE IMAG. PART OF SPECTRA – P_I. $S8_{LN} + P_I \rightarrow S8_{LN}$
INCREMENT N
NEXT SPECTRA
INCREMENT SAMPLE CT. AT LEVEL L ; RESET SATURATION COUNTER ; CLEAR FLAG
END

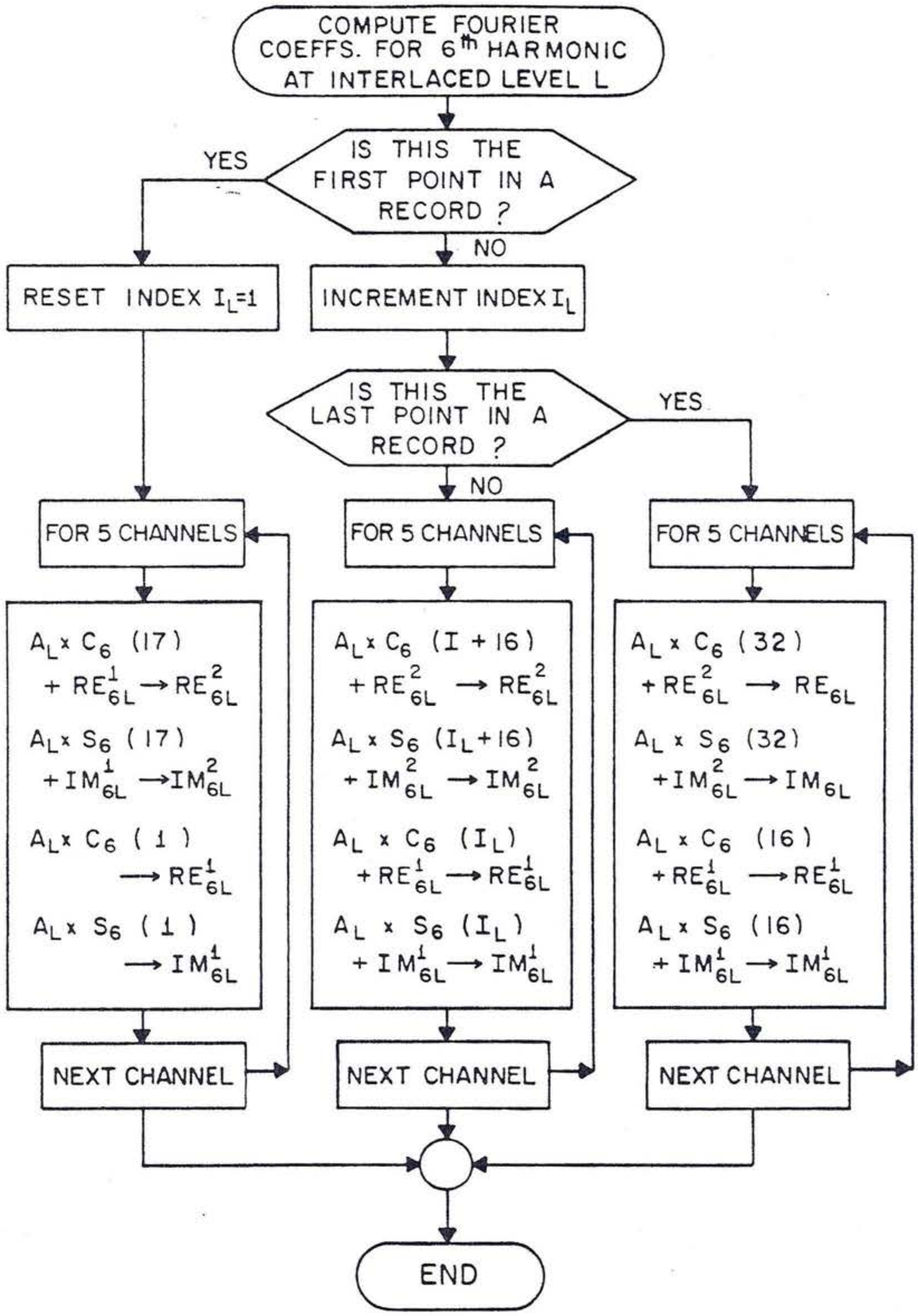

INTERRUPT SERVICE
INPUT DATA PTS.-5 CHANNELS
DIGITIZER SATURATED FOR ANY CHANNEL ?
YES
INCREMENT SATURATION COUNT AT ALL LEVELS
NO
COMPUTE 6th & 8th FOR INPUT DATA (LEVEL Ø)
CYCLE Ø OR 2 ? (1, B, B)
YES
NO
CYCLE 1 ? (1, 2, B)
YES
NO
EVEN PROCESSING, COMPUTE 6th LEVEL 1
EVEN PROCESSING, COMPUTE 6th LEVEL 1
ODD PROCESSING, COMPUTE 8th LEVEL 1
ODD PROCESSING, COMPUTE 8th LEVEL 2
EVEN PROCESSING, COMPUTE 6th LEVEL 2
PROCESS UP TO 2 LEVELS ≥ 4 ACCORDING TO FLAGS
PROCESS UP TO 1 LEVEL ≥ 4 ACCORDING TO FLAGS
PROCESS LEVEL 3 ALTERNATING ODD, 8th AND EVEN, 6th
RETURN

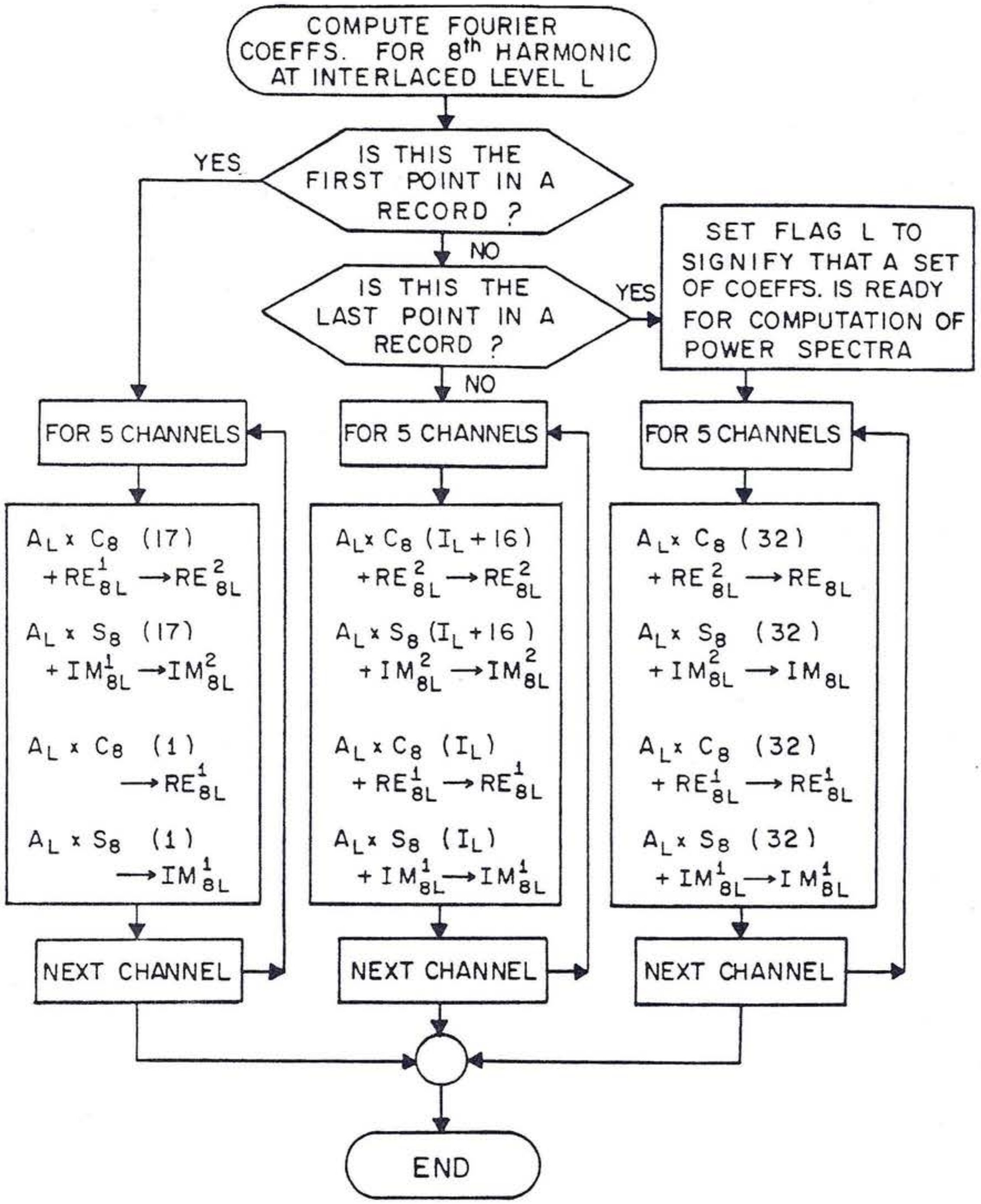

DECIMATE LEVEL≥4
BEGIN AT LEVEL 4 (4→L)
NEXT LEVEL (L+1→L)
CHECK FLAG L-1: NEW POINT READY AT LEVEL L-1 ?
NO
YES
SET FLAG L-1 TO "NEEDS ODD PROC."
IS LEVEL L LOWEST LEVEL ?
NO
YES
CHECK FLAG L: LEVEL L NEED EVEN PROCESSING ?
NO
YES
ODD PROCESSING LEVEL L
EVEN PROCESSING LEVEL L
IS LEVEL L INTERLACED ?
NO
YES
IS LEVEL L INTERLACED ?
NO
YES
COMPUTE 8th FOR INTERLACE
COMPUTE 8th COEFFS.
COMPUTE 6th COEFFS.
COMPUTE 6th FOR INTERLACE
INCREMENT DECIMATION COUNT
DONE 3 DECIMATIONS THIS POINT ?
NO
YES
END


ODD PROCESSING LEVEL L
FOR 5 CHANNELS
A_{L-1} x 3.41421356 → TEMP
B_L + TEMP → B_L
C_L + TEMP → C_L
NEXT CHANNEL
SET FLAG L TO "NEEDS EVEN PROC."
END
EVEN PROCESSING LEVEL L
FOR 5 CHANNELS
A_{L-1} → TEMP
B_L + TEMP → A_L
C_L + 4.87100924 x TEMP → B_L
TEMP → C_L
NEXT CHANNEL
SET FLAG L TO "NEW POINT READY AT LEVEL L"
END


COMPUTE FOURIER COEFFS. FOR 6th HARMONIC AT LEVEL L
FLAG L STILL SET (SPECTRA NOT YET DONE)?
YES
NO
STARTING NEW RECORD ?
YES
NO
INCREMENT INDEX I_L
RESET INDEX $I_L = 1$
FOR 5 CHANNELS
$A_L \times C_6(I_L) \rightarrow RE_{6L}$
$A_L \times S_6(I_L) \rightarrow IM_{6L}$
NEXT CHANNEL
FOR 5 CHANNELS
$A_L \times C_6(I_L) + RE_{6L} \rightarrow RE_{6L}$
$A_L \times S_6(I_L) + IM_{6L} \rightarrow IM_{6L}$
NEXT CHANNEL
END


COMPUTE FOURIER COEFFS. FOR 8th HARMONIC AT LEVEL L
FLAG L STILL SET (SPECTRA NOT YET DONE) OR I_L NOT YET RESET BY "6th"?
YES
NO
IS THIS THE LAST POINT IN A RECORD ?
NO
YES
SET FLAG L TO SIGNIFY THAT A SET OF COEFFS. IS READY FOR COMPUTATION OF POWER SPECTRA
FOR 5 CHANNELS
$A_L \times E_8(I_L) + RE_{8L} \longrightarrow RE_{8L}$
$A_L \times S_8(I_L) + IM_{8L} \longrightarrow IM_{8L}$
NEXT CHANNEL
END


COMPUTE FOURIER COEFFS. FOR 6th HARMONIC AT INTERLACED LEVEL L
IS THIS THE FIRST POINT IN A RECORD ?
YES
NO
RESET INDEX $I_L=1$
INCREMENT INDEX I_L
IS THIS THE LAST POINT IN A RECORD ?
YES
NO
FOR 5 CHANNELS
$A_L \times C_6$ (17) $+ RE^1_{6L} \rightarrow RE^2_{6L}$
$A_L \times S_6$ (17) $+ IM^1_{6L} \rightarrow IM^2_{6L}$
$A_L \times C_6$ (1) $\rightarrow RE^1_{6L}$
$A_L \times S_6$ (1) $\rightarrow IM^1_{6L}$
NEXT CHANNEL
FOR 5 CHANNELS
$A_L \times C_6$ (I + 16) $+ RE^2_{6L} \rightarrow RE^2_{6L}$
$A_L \times S_6$ (I_L+16) $+ IM^2_{6L} \rightarrow IM^2_{6L}$
$A_L \times C_6$ (I_L) $+ RE^1_{6L} \rightarrow RE^1_{6L}$
$A_L \times S_6$ (I_L) $+ IM^1_{6L} \rightarrow IM^1_{6L}$
NEXT CHANNEL
FOR 5 CHANNELS
$A_L \times C_6$ (32) $+ RE^2_{6L} \rightarrow RE_{6L}$
$A_L \times S_6$ (32) $+ IM^2_{6L} \rightarrow IM_{6L}$
$A_L \times C_6$ (16) $+ RE^1_{6L} \rightarrow RE^1_{6L}$
$A_L \times S_6$ (16) $+ IM^1_{6L} \rightarrow IM^1_{6L}$
NEXT CHANNEL
END


COMPUTE FOURIER COEFFS. FOR 8th HARMONIC AT INTERLACED LEVEL L
IS THIS THE FIRST POINT IN A RECORD ?
YES
NO
IS THIS THE LAST POINT IN A RECORD ?
YES
NO
SET FLAG L TO SIGNIFY THAT A SET OF COEFFS. IS READY FOR COMPUTATION OF POWER SPECTRA
FOR 5 CHANNELS
$A_L \times C_8$ (17) $+ RE^1_{8L} \rightarrow RE^2_{8L}$
$A_L \times S_8$ (17) $+ IM^1_{8L} \rightarrow IM^2_{8L}$
$A_L \times C_8$ (1) $\rightarrow RE^1_{8L}$
$A_L \times S_8$ (1) $\rightarrow IM^1_{8L}$
NEXT CHANNEL
FOR 5 CHANNELS
$A_L \times C_8$ $(I_L + 16)$ $+ RE^2_{8L} \rightarrow RE^2_{8L}$
$A_L \times S_8$ $(I_L + 16)$ $+ IM^2_{8L} \rightarrow IM^2_{8L}$
$A_L \times C_8$ (I_L) $+ RE^1_{8L} \rightarrow RE^1_{8L}$
$A_L \times S_8$ (I_L) $+ IM^1_{8L} \rightarrow IM^1_{8L}$
NEXT CHANNEL
FOR 5 CHANNELS
$A_L \times C_8$ (32) $+ RE^2_{8L} \rightarrow RE_{8L}$
$A_L \times S_8$ (32) $+ IM^2_{8L} \rightarrow IM_{8L}$
$A_L \times C_8$ (32) $+ RE^1_{8L} \rightarrow RE^1_{8L}$
$A_L \times S_8$ (32) $+ IM^1_{8L} \rightarrow IM^1_{8L}$
NEXT CHANNEL
END

APPENDIX 2

Program DECIMATE

A) Variables used by DECIMATE

B) Program Listing for DECIMATE

Variable Allocation for Program DECIMATE

Simple Variables

A	Total number of levels
B	Level at which interlacing begins
C	Channel index
D	Number of decimations
I	Index for computing spectra
J	Index for computing spectra
L	Level index in interrupt service routine
M	Level index in main program
N	Index for storing spectra
Q	File number
R	X azimuth
S	Y azimuth
T	Temporary storage
U	Temporary storage
V	Temporary storage
W	Index for processing of upper levels
X	X line length
Y	Y line length
Z	Sample period

String Variables

S$ (20)	Site name - run number
D$ (10)	Date
C$ (60)	Comments
R$ (5,10)	Filter settings and gains

Subscripted Variables

A	(O:A,5)	Decimation register A	
B	(A,5)	Decimation register B	
C	(A,5)	Decimation register C	
D	(O:4)	Alias filter coefficients	
E	(O:A)	Index for sine and cosine arrays	
F	(A)	Flags for decimation	
G	(O:A,5)	6th real	Fourier coefficients
H	(O:A,5)	6th imag	
I	(O:A,5)	8th real	
J	(O:A,5)	8th imag	
K	(O:T,5)	6th real	Registers for "building" first and second halves of interlace records T=2(A-B)+1
L	(O:T,5)	6th imag	
M	(O:T,5)	8th real	
N	(O:T,5)	8th imag	
O	(32)	cos 6th	Stored sine and cosine arrays
P	(32)	-sin 6th	
Q	(32)	cos 8th	
R	(32)	-sin 8th	
S	(O:A)	Flags for "Fourier coefficients ready"	
T	(O:A)	Saturation count	
U	(O:A)	Saturations allowed	
V	(O:A)	Number of complete spectra records	
W	(O:A)	Frequency array - 6th	
X	(O:A)	Frequency array - 8th	
Y	(O:A,25)	Spectra - 6th	
Z	(O:A,25)	Spectra - 8th	

```
0: "DECIMATE
   6/10/77":
1: 9→A;5→B
2: dim A[0:A,5],
 B[A,5],C[A,5],
 D[0:4],E[0:A],
 F[A],G[0:A,5],
 H[0:A,5]
3: 2(A-B)+1→T
4: dim I[0:A,5],
 J[0:A,5],K[0:T,
 5],L[0:T,5],
 M[0:T,5],N[0:T,
 5]
5: dim O[32],
 P[32],Q[32],
 R[32],S[0:A],
 T[0:A]
6: dim U[0:A],
 V[0:A],W[0:A],
 X[0:A],Y[0:A,
 25],Z[0:A,25]
7: dim S$[20],
 D$[10],C$[60],
 R$[5,10],Q,R,S,
 X,Y,Z
8: fxd 1
9: spc 3;prt "
 MT-DECIMATE";
 spc 2;ent "ente
 r site name /
 run number",S$
10: prt S$;spc ;
 ent "enter date
  - mo/day/yr",
 D$;prt D$;spc
11: ent "enter
 line length -
 Lx",X;prt "Lx="
 ,X;spc
12: ent "enter
 X bearing (-
 180<angle<180)"
 ,R;prt "angle(x
 )=",R;spc
13: ent "enter
 line length -
 Ly",Y;prt "Ly="
 ,Y;spc
14: ent "enter
 Y bearing (-
 180<angle<180)"
 ,S;prt "angle(y
 )=",S;spc
15: "period":ent
  "enter sample
 period in secon
 ds",Z;fxd 4
16: if Z=0;dsp
 "period can
 not be zero";
 wait 1000;gto
 "period"
17: prt "period=
 ",Z
18: 0→U[0]→U[1]→
 U[2]→U[3];8→U[4
 ];16→U[5];for
 T=B to A;2↑(1+
 T)→U[T];next T
19: "setsat":dsp
  "EXEC to chang
 e sat level or
 CONT";rdb(0)→T
20: if T#10;gto
 "nochange"
21: fmt 9,"allow
 ed saturations-
 level",f2.0,
 "=",f4.0
22: for U=0 to
 A;wrt .9,U,U[U]
 ;wait 2500;ent
 "enter new valu
 e or CONT",U[U]
23: next U;gto
 "satprt"
24: "nochange":d
 sp " ";if T#25;
 gto "setsat"
25: "satprt":fxd
  0;spc 2;prt
 "Allowable Sat"
 ;spc ;for U=0
 to A;prt U[U]
26: next U
27: for U=0 to
 A;6/32Z2↑U→W[U]
 ;8/32Z2↑U→X[U];
 1e10→T[U];next
 U
28: rad;for U=1
 to 32;U-1→V;
 .5(1-cos(2πV/
 32))→T
29: Tcos(12πV/
 32)→O[U];-Tsin(
 12πV/32)→P[U];
 Tcos(16πV/32)→Q
 [U]
30: -Tsin(16πV/
 32)→R[U];next U
31: trk 0;ldk 0;
 rew
32: spc 2;prt
 "Filter Setting
 s";spc ;fmt 1,
 c2,2x,c10
33: ent "enter
 Ex filter /
 gains",R$[1];
 wrt 16.1,"Ex",
 R$[1]
34: ent "enter
 Ey filter /
 gains",R$[2];
 wrt 16.1,"Ey",
 R$[2]
35: ent "enter
 Hx filter /
 gains",R$[3];
 wrt 16.1,"Hx",
 R$[3]
36: ent "enter
 Hy filter /
 gains",R$[4];
 wrt 16.1,"Hy",
 R$[4]
37: ent "enter
 Hz filter /
 gains",R$[5];
 wrt 16.1,"Hz",
 R$[5];spc 2
38: dsp "CONT
 to begin run";
 stp
```

```
39: lkd ;cfg 13;
 oni 2,"point";
 eir 2
40: "flgscan":if
  flg5;5→M;gto
 "spectra"
41: if flg6;6→M;
 gto "spectra"
42: if flg7;7→M;
 gto "spectra"
43: if flg8;8→M;
 gto "spectra"
44: if flg9;beep
 ;9→M;gto "spect
 ra"
45: if flg4;4→M;
 gto "spectra"
46: if flg3;3→M;
 gto "spectra"
47: if flg2;2→M;
 gto "spectra"
48: if flg1;1→M;
 gto "spectra"
49: if flg0;0→M;
 gto "spectra"
50: if flg13;
 gto "record"
51: gto "flgscan
 "
52: "spectra":.
53: if T[M]>U[M]
 ;0→S[M]→T[M];
 cfg M;gto "flgs
 can"
54: 1→N;for I=1
 to 5;for J=I
 to 5
55: G[M,J]G[M,
 I]+H[M,I]H[M,
 J]+Y[M,N]→Y[M,
 N]
56: I[M,J]I[M,
 I]+J[M,I]J[M,
 J]+Z[M,N]→Z[M,,
 N];N+1→N
57: if I=J;gto
 "autospec"
58: H[M,I]G[M,
 J]-G[M,I]H[M,
 J]+Y[M,N]→Y[M,
 N]
```

```
59: J[M,I]I[M,
 J]-I[M,I]J[M,
 J]+Z[M,N]→Z[M,
 N];N+1→N
60: "autospec":n
 ext J;next I;
 V[M]+1→V[M]
61: fmt 8,f4.0,
 x,f3.0,x,f3.0,
 x,f3.0,x,f2.0,
 x,f2.0,x,f2.0,
 x,f2.0,x,f1.0,
 x,f1.0
62: wrt .8,V[0],
 V[1],V[2],V[3],
 V[4],V[5],V[6],
 V[7],V[8],V[9]
63: "endspec":0→
 S[M]→T[M];if
 M<B;0→S[M]
64: cfg M;gto
 "flgscan"
65: "record":eir
  2,0;dsp "Load
 data tape then
 continue";stp
66: 1→D[0]→D[4];
 3.41421356→D[1]
 →D[3];4.8710092
 4→D[2];1→E→F;
 rad
67: for P=0 to A
68: 1/8E→M;1/
 8F→N
69: if V[P]=0;
 for T=1 to 25;
 0→Y[P,T]→Z[P,
 T];next T;gto
 "remove"
70: for T=1 to
 25;(MM/V[P])Y[P
 ,T]→Y[P,T];(NN/
 V[P])Z[P,T]→Z[P
 ,T];next T
71: "remove":0→I
 →J→K→L;3/32/
 2↑P→G;4/32/2↑P→
 H
```

```
72: for U=0 to 4
73: D[U]cos(2πUG
 )+I→I
74: D[U]sin(2πUG
 )+J→J
75: D[U]cos(2πUH
 )+K→K
76: D[U]sin(2πUH
 )+L→L
77: next U
78: E√(II+JJ)→E;
 F√(KK+LL)→F;
 next P
79: ent "Comment
 s ?",C$
80: trk 1;ldf 0,
 Q
81: Q+1→Q;rcf 0,
 Q;fdf Q;idf U,
 V;if V#0;dsp
 "file-trk1 not
 empty",Q;stp
82: mrk 1,5000;
 trk 0;fdf Q;
 idf U,V;if V#0;
 dsp "file-trk0
 not empty",Q;
 stp
83: mrk 1,5000;
 trk 1
84: rcf Q,U[*],
 V[*],W[*],X[*],
 Y[*],Z[*],S$,
 D$,C$,R$,Q,R,S,
 X,Y,Z
85: trk 0;rew
86: prt "No. of
 Records";spc ;
 fxd 0;for U=0
 to A;prt V[U];
 next U;spc 2
87: prt "File
 Number=",Q;spc
 2
88: dsp "CONT
 to print raw
 spectra";stp
89: prt "Raw
 Spectra";spc
90: for T=0 to A
```

```
91: spc ;fxd 5;
 prt "freq=",
 X[T];flt 6
92: for V=1 to
 25;prt Z[T,V];
 next V
93: fxd 5;spc ;
 prt "freq=",
 W[T];flt 6
94: for V=1 to
 25;prt Y[T,V];
 next V
95: next T
96: stp
97: "


      ":
98: "point":for
 C=1 to 5;if
 abs(rdb(2)→A[0,
 C])>2046;sfg 11
99: next C;if
 flg11;cfg 11;
 for T=0 to A;
 T[T]+1→T[T];
 next T
100: 0→L;cll
 '6th';cll '8th'
 ;if W=0 or W=2;
 gto "1"
101: if W=1;gto
 "1,2"
102: "1,2,3":1→L
 ;cll 'even';
 cll '6th';2→L;
 cll 'even';cll
 '6th';3→L
103: cmf 12;if
 flg12;cll 'odd'
 ;cll '8th';gto
 "123ret"
104: cll 'even';
 cll '6th'
105: "123ret":0→
 W;wtc 2,1;eir
 2;iret
106: "1":1→L;
 cll 'odd';cll
 '8th';1→D;cll
 'flags';W+1→W
```

```
107: wtc 2,1;
 eir 2;iret
108: "1,2":1→L;
 cll 'even';cll
 '6th';2→L→D→W;
 cll 'odd';cll
 '8th'
109: cll 'flags'
 ;wtc 2,1;eir 2;
 iret
110: "flags":4→L
111: "startflg":
 if F[L-1]=2;
 gto "servflg"
112: "flgloop":i
 f L=A;ret
113: L+1→L;gto
 "startflg"
114: "servflg":0
 →F[L-1]
115: if F[L]=1;
 cll 'even';gto
 "int6?"
116: cll 'odd';
 if L>=B;cll
 '8thInt';gto
 "done"
117: cll '8th';
 gto "done"
118: "int6?":if
 L>=B;cll '6thIn
 t';gto "done"
119: cll '6th'
120: "done":D+
 1→D;if D=3;ret
121: gto "flgloo
 p"
122: "even":for
 C=1 to 5;A[L-1,
 C]→T;B[L,C]+
 T→A[L,C];C[L,
 C]+4.87100924T→
 B[L,C]
123: T→C[L,C];
 next C;2→F[L];
 ret
124: "odd":for
 C=1 to 5;3.4142
 1356A[L-1,C]→T;
 B[L,C]+T→B[L,
 C];C[L,C]+T→C[L
 ,C]
```

```
125: next C;1→F[
 L];ret
126: "6th":
127: if flgL;
 ret
128: if S[L]=1;
 gto "6norm"
129: 1→E[L]→S[L]
 ;for C=1 to 5;
 A[L,C]→T;TO[1]→
 G[L,C];TP[1]→H[
 L,C]
130: 0→I[L,C]→J[
 L,C];next C;
 ret
131: "6norm":E[L
 ]+1→E[L]→U;for
 C=1 to 5;A[L,
 C]→T;TO[U]+G[L,
 C]→G[L,C]
132: TP[U]+H[L,
 C]→H[L,C];next
 C;ret
133: "8th":
134: if flgL or
 S[L]=0;ret
135: if (E[L]→U)
 =32;sfg L
136: for C=1 to
 5;A[L,C]→T;TQ[U
 ]+I[L,C]→I[L,
 C];TR[U]+J[L,
 C]→J[L,C];next
 C;ret
137: "6thInt":if
 S[L]=1;gto
 "6Intnorm"
138: L-B→V;1→E[L
 ]
139: for C=1 to
 5;A[L,C]→T;TO[1
 7]+K[V,C]→K[L,
 C];TP[17]+L[V,
 C]→L[L,C]
140: TO[1]→K[V,
 C];TP[1]→L[V,
 C];next C;ret
141: "6Intnorm":
 if (E[L]+1→E[L]
 →U)=16;gto "6In
 tlast"
```

```
142: L-B→V
143: for C=1 to
 5;A[L,C]→T;TO[1
 6+U]+K[L,C]→K[L
 ,C]
144: TP[16+U]+
 L[L,C]→L[L,C];
 TO[U]+K[V,C]→K[
 V,C];TP[U]+L[V,
 C]→L[V,C]
145: next C;ret
146: "6Intlast":
 if flgL;dsp
 "LOST SYNC";
 stp
147: L-B→V
148: for C=1 to
 5;A[L,C]→T;TO[3
 2]+K[L,C]→G[L,
 C];TP[32]+L[L,
 C]→H[L,C]
149: TO[16]+K[V,
 C]→K[V;C];TP[16
 ]+L[V,C]→L[V,
 C];next C;ret
150: "8thInt":if
  S[L]=1;gto
 "8Intnorm"
151: L-B→V;1→S[L
 ]
152: for C=1 to
 5;A[L,C]→T;TQ[1
 7]+M[V,C]→M[L,
 C];TR[17]+N[V,
 C]→N[L,C]
```

```
153: TQ[1]→M[V,
 C];TR[1]→N[V,
 C];next C;ret
154: "8Intnorm":
 if (E[L]→U)=16;
 gto "8Intlast"
155: L-B→V
156: for C=1 to
 5;A[L,C]→T;TQ[U
 +16]+M[L,C]→M[L
 ,C]
157: TR[U+16]+
 N[L,C]→N[L,C];
 TQ[U]+M[V,C]→M[
 V,C]
158: TR[U]+N[V,
 C]→N[V,C];next
 C;ret
159: "8Intlast":
 sfg L;0→S[L];L-
 B→V
160: for C=1 to
 5;A[L,C]→T;TQ[3
 2]+M[L,C]→I[L,
 C]
161: TR[32]+N[L,
 C]→J[L,C];TQ[16
 ]+M[V,C]→M[V,
 C];TR[16]+N[V,
 C]→N[V,C];next
 C
162: ret
163: end
*30388
```

BIBLIOGRAPHY

Bailey, R.C., "Inversion of the Geomagnetic Induction Problem", Proc. Roy.Soc.Lond, A. 315, (1970), 185-194.

Becher, W.D., and C.B. Sharpe, "A Synthesis Approach to Magnetotelluric Exploration", Radio Science, Vol. 4, No. 11 (Nov. 1969), 1089-1094.

Blackman, R.B., and J.W. Tukey, The Measurement of Power Spectra, New York:Dover Publications, 1959.

Boehl, J.E., and F.X. Bostick, "An Application of the Hilbert Transform to the Magnetotelluric Method", Ph.D. dissertation, The University of Texas at Austin, 1977.

Bostick, F.X., H.W. Smith, and J.E. Boehl, "Magnetotelluric and DC Dipole - Dipole Soundings in Northern Wisconsin," Austin, Texas: The University of Texas Electrical Geophysics Research Laboratory, 1977.

Bostick, F.X., and H.W. Smith, "Investigations of Large-Scale Inhomogenieties in the Earth by the Magnetotelluric Method", Proceedings of the IRE, Vol. 50 (Nov. 1962), 2339-46.

Brigham, E.O., The Fast Fourier Transform, Englewood Cliffs, N.J.: Prentice-Hall, 1974.

Cagniard, L., "Basic Theory of the Magnetotelluric method of Geophysical Prospecting," Geophysics, Vol. 18 (July 1953), 605-35.

Cooley, J.W., and J.W. Tukey, "An Algorithm for Machine Calculation of Complex Fourier Series,"Math. Computation,Vol. 19 (April 1965), 297-301.

Cantwell, T., "Detection and Analysis of Low-Frequency Magnetotelluric Signals," Ph.D. dissertation, Massachusetts Institute of Technology, 1960.

Franklin, R.N., Personal communication regarding material in forthcoming M.S. Thesis, The University of Texas at Austin, May, 1977.

Hewlett-Packard Company, Hewlett-Packard 9825 Calculator Operating and Programming, Loveland, Colo.: Hewlett-Packard Company, 1976.

Hewlett-Packard Company, Hewlett-Packard 9825A Calculator Advanced Programming ,Loveland, Colo.: Hewlett-Packard Company, 1976.

Hewlett-Packard Company, Hewlett-Packard 9825A Calculator General I/O Programming, Loveland, Colo.: Hewlett-Packard Company, 1976.

Hewlett-Packard Company, Hewlett-Packard 9825A Calculator Extended I/O Programming, Loveland, Colo.: Hewlett-Packard Company, 1976.

Hewlett-Packard Company, Hewlett-Packard 9825A Calculator String Variable Programming, Loveland, Colo.: Hewlett-Packard Company, 1976.

Hewlett-Packard Company, Hewlett-Packard 98032A 16-Bit Interface Installation and Sevice Manual, Loveland, Colo.:Hewlett-Packard Company, 1976.

Johnson, I.M., and D.E. Smylie, "An Inverse Theory for the Calculation of the Electrical Conductivity of the Lower Mantle", Geophys. J.R. Astr. Soc., Vol. 22 (1970), 41-53.

Laird, C.E., and F.X. Bostick, "One-Dimensional Magnetotelluric Inversion Techniques", The University of Texas Electrical Geophysics Research Laboratory Technical Report No. 101, Austin, Texas: The University of Texas, 1970.

Oppenheim, A.V., and R.W. Schafer, Digital Signal Processing, Englewood Cliffs, N.J.: Prentice-Hall, 1975.

Patrick, F.W., "Magnetotelluric Modeling Techniques", Ph.D. Dissertation The University of Texas at Austin, 1969.

Sims, W.E., and F.X. Bostick, "An Analysis of Dynamic Power Spectra," The University of Texas Electrical Engineering Research Laboratory Technical Report No. 139, Austin, Texas: The University of Texas 1965.

Sims, W.E., and F.X. Bostick, "Methods of Magnetotelluric Analysis," The University of Texas Electrical Geophysics Research Laboratory Technical Report No. 48, Austin, Texas: The University of Texas 1969.

Sims, W.E., F.X. Bostick, and H.W. Smith, "The Estimation of Magnetotelluric Impedance Tensor Elements from Measured Data," Geophysics, Vol. 36 No. 5 (Oct. 1971), 938-42.

Stanley, W.D., J.E. Boehl, F.X. Bostick, and H.W. Smith, "Survey of Magnetotelluric Soundings in the Snake River Plain-Yellowstone Region", J. Geophys. Res., Vol. 82, No. 7 (June 1977).

Swift, C.M., "A Magnetotelluric Investigation of an Electrical Conductivity Anamoly in the Southwestern United States", Ph.D. dissertation, Massachusetts Institute of Technology, 1967.

Walker, B. "An Analysis of the Design of Nonrecursive Digital Filters from Frequency Domain Specification," M.S. Thesis, The University of Texas at Austin, 1977.

Word, D.R., H.W. Smith, and F.X. Bostick, "An Investigation of the Magnetotelluric Tensor Impedance Method", The University of Texas Electrical Geophysics Research Laboratory Technical Report No. 82, Austin, Texas: The University of Texas, 1970.

Wu, F.T., "The Inverse Problem of Magnetotelluric Sounding," Geophysics, Vol. 33, No. 6 (1968), 972-79.