DATA REDUCTION AND
COMPUTER GROUP

TECHNICAL REPORT

TECHNIQUES IN DATA COMPRESSION

AND

MAXIMIZATION OF INFORMATION CONTENT

WHITE SANDS MISSILE RANGE
KWAJALEIN MISSILE RANGE
YUMA PROVING GROUND
ELECTRONIC PROVING GROUND
DUGWAY PROVING GROUND

NAVAL AIR WARFARE CENTER - WEAPONS DIVISION
ATLANTIC FLEET WEAPONS TRAINING FACILITY
NAVAL AIR WARFARE CENTER - AIRCRAFT DIVISION
NAVAL UNDERSEA WARFARE CENTER DIVISION, NEWPORT

45TH SPACE WING
AIR FORCE DEVELOPMENT TEST CENTER
30TH SPACE WING
CONSOLIDATED SPACE TEST CENTER
AIR FORCE FLIGHT TEST CENTER
AIR FORCE TACTICAL FIGHTER WEAPONS CENTER

TECHNIQUES IN DATA COMPRESSION AND MAXIMIZATION OF INFORMATION CONTENT

JANUARY 1977

DATA REDUCTION AND COMPUTING GROUP

RANGE COMMANDERS COUNCIL

Published by

Secretariat
Range Commanders Council
White Sands Missile Range
New Mexico 88002

TABLE OF CONTENTS

PAGE

CHAPTER 4 - STATISTICAL REPRESENTATION

CHAPTER 5 - OPTIMAL ESTIMATION TECHNIQUES

PAGE

LIST OF FIGURES

LIST OF TABLES

TECHNIQUES IN DATA COMPRESSION
AND
MAXIMIZATION OF INFORMATION CONTENT

AD HOC COMMITTEE

Participants	Range/Agency
Marvin E. Bauder, Chairman	SLA
Robert N. Barry	AFFTC
Barry L. Clark	NSWC
Richard H. Dale	WSMR
Thomas E. Ford	PMTC
Weldon R. Howell	WSMR
Robert N. Learn	NSWC
Clifford J. Lemieux	PMTC
Herbert S. Walters	ADTC
Danny B. Weddle	NATC

FOREWORD

At its 44th meeting in October 1975, the Data Reduction and Computing Group (DR&CG), of the Range Commanders Council (RCC), drafted a proposed task description entitled, "Techniques In Data Compression and Maximization of Information Content." This proposed task was submitted to the RCC Executive Committee for approval. Approval was granted and at the following DR&CG meeting an ad hoc committee was established, a detailed outline of the proposed document was developed, and specific assignments were made to accomplish the task. Committee members were instructed to conduct research into their specific areas and to come to a special work session in July 1976 prepared to write a first draft of the proposed document. At this special meeting the first draft was compiled and reviewed. Further refinements were made by individual participants and a final draft was submitted to the DR&CG membership for review at the September 1976 meeting. Additional minor changes and additions were made and the document was then published.

CHAPTER 1

DATA COMPRESSION AND MAXIMIZATION OF INFORMATION CONTENT

1.1 INTRODUCTION

Although this document primarily addresses problems at the various test ranges, it should provide applications for use throughout the scientific community. Much of the reference material used was derived from publications by other government agencies, contractors, universities, and private industry.

Within the past several years nearly all test ranges have experienced an exponential growth in the quantity of data being recorded and processed. Almost concurrently, large-scale, real-time data processing systems have been developed. It is reasonable to expect this trend to continue in the years to follow. Consequently, there is an increasing demand for minimizing redundancy and compressing the masses of data into forms which may be more quickly and easily assimilated.

Much effort has been expended at various locations in the development of reliable means for transmitting and retaining only significant changes in data instead of processing all that is generated. Considerable attention has also been given to presenting this data in forms conducive to early decision making. All this effort has produced a number of techniques which make up the field of data compression and maximization of information content.

1.2 SCOPE

This publication is intended as a single source document describing available techniques for reducing the quantity of data processed and for providing meaningful presentation. It includes mathematical, statistical, and graphical techniques which have been used successfully.

1.3 DATA COMPRESSION DEFINITION

The expression "data compression" has broad meaning and may encompass any or all of the following: data compaction, bandwidth compression, redundancy removal, redundancy reduction, adaptive sampling, parameter extraction, optimal estimation, and possibly a few other techniques. In general, data compression denotes operations which are performed to reduce the quantity of data prior to transmission, but which still preserve the minimum data elements of a measurement continuum such that the original information may be reconstructed within established limits of error.

CHAPTER 2

REDUNDANT DATA REMOVAL/USEFUL DATA SELECTION

2.1 INTRODUCTION

Data redundancy has been defined as "that fraction of a message or datum which is unnecessary and hence repetitive in the sense that if it were missing the message would still be essentially complete, or at least could be completed. Redundancy exists whenever the sampling rate... exceeds the frequency required to describe the input function in accordance with the accuracy requirements of the user."[2-6] The methods for retaining data which provide essentially all the information contained in the original message range from some simple visual and manual techniques to complex computer driven algorithms. However, the basis of all removal/selection techniques is the examination of each data sample and performing a comparison to preceding or succeeding samples in the context of some arbitrary reference pattern.[2-6] The choice of methods is extensive and may be adapted to virtually any set of circumstances or data.

2.2 EDITING

The editing processes involve the identification and subsequent removal of data estimates which are considered either erroneous or non-essential to the information content. Additionally, if erroneous data during "critical" intervals are edited those samples must sometimes be replaced by prediction/interpolation techniques. The implementation of these processes depends on the purposes and uses of the data, especially in the context that raw data are generally comprised of both expected and abnormal, or unexpected, input samples. The flight engineer will consider useful that data which shows missile performance characteristics only, whereas the instrumentation engineering will be interested in data showing instrumentation failures. In each case data compression may be accomplished through the reduction methods of editing.

2.2.1 MANUAL TECHNIQUES

Although generally more laborious and vulnerable to a certain amount of subjective judgement, manual editing is often employed to reduce the data volume and select useful data, especially during preliminary data processing stages. Two common data formats are lists and plots, and in each case techniques may be employed to facilitate the reduction process.

A simple method of editing printed data is to arrange it in a columnar format and sort it with respect to some key parameter, usually, but not necessarily, time. The reorganization of data in this way improves the capability to show data discontinuities and duplicated samples, which can be identified for removal. When samples are arranged in vertical juxtaposition with respect to previous and subsequent samples of the same function, simple trend analyses may be accomplished. These include such non-parametric tests as determination of zero crossings and the relative sizes (value) of the data estimates. In order to facilitate the editing of data lists the data parameters may be reconfigured by computing simple first or second differences between samples which will detrend the data and amplify data anomalies:

$$\begin{aligned} \Delta x_i &= x_i - x_{i-1} \\ \Delta^2 x_i &= \Delta x_i - \Delta x_{i-1} = x_i - 2x_{i-1} + x_{i-2} \\ &\vdots \\ \Delta^n x_i &= \Delta^{n-1} x_i - \Delta^{n-1} x_{i-1} \end{aligned} \qquad (2.2\text{-}1)$$

The data may be reconfigured into estimates of variances over short intervals which may be reviewed to determine quickly where data samples may be edited. These methods of reconfiguration may be combined to provide detrended variances in data which are changing in a polynomial fashion; the variances may be estimated from the differenced data, [2-2]

$$\sigma_x^2 = \frac{\sum_{i=1}^{m} (\Delta^n x)^2}{(m-n)\frac{(2n)!}{n!\,n!}} \qquad (2.2\text{-}2)$$

where σ_x = random error in the x coordinate

$\Delta^n x$ = the nth successive difference in x

m = the number of points used in each sample

When using this technique, $n \geq 3$ in order to at least eliminate quadratic trends.

The same general methods are used in analyzing and editing data in plotted formats. Trend analyses may be accomplished more readily because of the ability to review the data in a more condensed form. Sampling rates may be determined and useful data spans identified. Discontinuities in the data trend and spurious samples may be discerned, and in reviewing data in plotted form it is possible to determine patterns of abnormal data occurances. A drawback in editing the data from the graphical representation is the loss of a certain amount of data resolution, depending upon presentation scale factors.

2.2.2 BOUNDARY LIMIT EDITING

The simplest computer editing methods employ a selection process which compares the present data value to preset upper and lower limits:

$$k_1 \leq x_i \leq k_2, \qquad (2.2\text{-}3)$$

where the k's may be constant or even some function of (x). Under one option, acceptance occurs when the condition above is true; under the other if the statement is false. In either case the boundary limit test is designed to eliminate data which is considered unimportant.[2-10] Because the values of the k's must be predetermined, a priori knowledge of the nature of the data must be considered in planning for this type of editing.

Depending on the environmental, instrumentation, or processing characteristics which affect the nature of the data, the following factors may be utilized.[2-8]

a. Time Constraints. Data may be recorded only when it is within the time period covering a specified maneuver for a particular test. The most common procedure is simply to turn off the recorders, or edit recorded data using times found on operational notes to avoid processing data considered meaningless. Additionally, data sampling or compression may be initiated or discontinued on the basis of other events, which are measured or recorded.[2-1]

b. Physical Bounds. Variable which exceed known physical limitations, e.g., velocities of Mach 10, aircraft altitudes over a million feet, etc., need not be accepted or processed.

c. Calibration Limits. Telemetry functions, especially, those which exceed calibration limits will probably be outside the desired testing range.

d. Computer Table Limits. Editing criteria may be based on the amount of available computer core or tape storage whenever the data samples retained will meet conditions sufficiently to describe the entire population. This method of editing is usually employed if further data compression methods, viz, regression analysis or analysis of variance, will be used.

e. Detection Threshold Limits. The signal-to-noise level of all functions may be monitored to determine if any data are in fact being received.

f. Historical Limits. Based on the results of previous similar tests, expectation bounds may be determined to edit subsequent tests. The historical limits will usually be finer than the physical bounds.

g. Statistical Limits. Estimates of variance may be computed over short intervals and used to remove erroneous or meaningless data, or to sense signficant changes. The variation of the general test in this case would be

$$-ks_x \leq x_i \leq ks_x, \qquad (2.2\text{-}4)$$

where s_x is the estimate of the standard deviation in x.

h. Trend Limits. Based on the change in the trend of the data, a data sample, x_i, may be eliminated if

$$|x_i - x_{i-1}| > k \qquad (2.2\text{-}5)$$

where the boundary k is known a priori.

2.2.3 SOURCE SELECTION

When there are simultaneous measurements of a parameter by more than one instrumentation system, redundancy exists and a best estimate of that parameter may be made and all other measurements discarded.[2-12] Source selection may be accomplished in two ways.

a. Determination of the best source. A preliminary step here is to eliminate all data showing apparent malfunctions. This may be

accomplished by using the various methods of limit checking. Variance estimates may then be computed from each set of data and utilized as weighting criteria to determine the best data set,

$$W_j = \frac{1}{s_j^2} \qquad (2.2\text{-}6)$$

In practical application, changes in source selection should be made only when the weights change significantly.

b. Computation of a combined best source. The relative weights previously calculated in the determination of the best source may be used to compute a set of data which is a combination of all sources and which provides more confidence than use of any one set alone. This best estimate may be used as the data source during further processing. The pooled estimate may be computed:[2-11]

$$\tilde{x}_i = \frac{\sum_{j=1}^{n} W_j X_{ij}}{\sum_{j=1}^{n} W_j} \qquad (2.2\text{-}7)$$

2.3 SAMPLING

Redundancy removal through sampling is a direct data compression method which operates on the data in such a way that the output values are the actual sample values of the input data, or the actual sample values within a tolerance.[2-1] These data compression techniques can be divided into two classes; those which essentially destroy the time reference and transmit the significant samples at a constant rate, and those which transmit only significant samples as they occur in time.[2-5] The first method is termed fixed rate compression, the other, variable rate compression.

When using fixed rate sampling it is assumed that the data characteristics are constant and some optimum rate may be determined a priori. This is generally based on the highest frequency expected in the data. Since

the data sampling rate is known, the time tag need not be carried along but may be reconstructed after the essential processing is complete.

Variable rate compressors, on the other hand, have greater potential for redundancy reduction because the output sampling rates are keyed to variations in the data characteristics. However, because of this flexibility, each data sample requires a time tag, and in some cases, when combined with other compressed data, may result in more data bits being transmitted than were in the original data.[2-1]

2.3.1 FIXED RATE COMPRESSION

The technique most commonly used is simply to sample the data at a rate close to the Nyquist or folding frequency, (f_N), which is the maximum frequency which can be resolved for a given sampling rate, Δt. Generally most sensors are sampled at more than the theoretical minimum of twice the highest frequency component:

$$f_N = \frac{1}{2\Delta t} , \qquad (2.3\text{-}1)$$

and frequencies (f) which could be resolved are:

$$0 \leq f \leq \frac{1}{2\Delta t} \qquad (2.3\text{-}2)$$

If significant frequencies which are higher than $\frac{1}{2\Delta t}$ Hertz exist in the data they will appear as lower frequencies between 0 and $\frac{1}{2\Delta t}$ Hz. This is called aliasing, and must be considered in the determination of the sampling rate for data compression.[2-12] Figure 2-1 shows a spectral representation of typical data, where the sampling rate (Δt = 0.1 sec) was far greater than required to represent the highest signficiant frequency, in this case, $f_s \approx 0.5$ Hz. Spectral estimates over the frequency range f > 0.5 Hz are relatively low amplitude noise. If the data were sampled at a reduced rate, for example $\Delta t \leq 1$ second, that noise would be aliased into

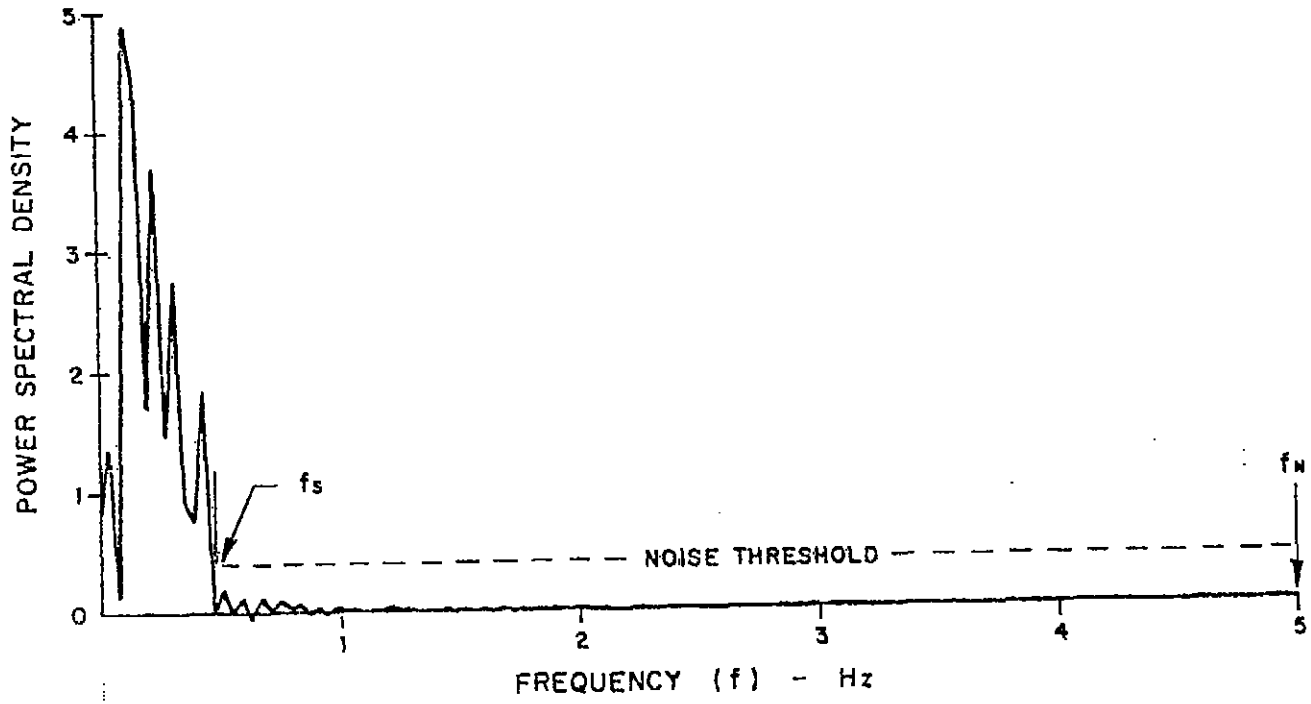


FIGURE 2-1. Estimate of the Spectral Density Function. The ordinate shows estimates of the variance per frequency increment.

the region f < 0.5 Hz with some adverse effect on the compressed data. To avoid aliasing, prefiltering the original data is necessary. This may be accomplished by using a low pass filter with a frequency cutoff equal to the highest significant frequency, or the highest desired Nyquist frequency.

2.3.2 VARIABLE RATE COMPRESSION

With this type of redundancy reduction the waveform is initially sampled at a constant rate and the nonessential samples are eliminated when the data change exceeds a predetermined tolerance with respect to a reference pattern. The choice of reference patterns used to detect redundancy is virtually unlimited. Examples are: polynomials, exponentials, and sine waves.[2-7] Of the many techniques the most widely used and discussed are the polynomial predictors and interpolators, since most data can be expressed or approximated in that form, especially over the data spans to be tested. A general description of these is as follows:[2-9]

- A tolerance window is placed about the data starting at the first data point.

- Succeeding points which fall within the tolerance window are considered redundant and are discarded.

- When a succeeding point falls outside the window, an appropriate point is saved and a new tolerance window is placed about the succeeding data.

- Each time a point falls outside the window, a new window is used for the succeeding data.

2.3.2.1 PREDICTORS

A predictor is an algorithm that estimates the value of each new data sample based on past performance of the data. If the

new data value falls within the tolerance range about the estimated new value, it is rejected as redundant since it is known that the data value can be constructed within that tolerance range.[2-7] A class of redundancy reduction techniques using predictors assumes that the sample will follow an n-th order polynomial of the form[2-1]

$$x_i = x_{i-1} + \Delta x_{i-1} + \Delta^2 x_{i-1} + \ldots + \Delta^n x_{i-1} \qquad (2.3\text{-}3)$$

where x_i is the predicted sample,

x_{i-1} is the previous sample,

$\Delta^n x_{i-1}$ are the successive differences as defined in subparagraph 2.2.1. A tolerance of $x_i \pm k$ can then be established about x_i.

2.3.2.1.1 ZERO-ORDER PREDICTORS

Commonly known as the "Step Method," the zero-order predictor is the simplest. For the zero-order predictor, $n = 0$, and equation (2.3-3) reduces to[2-1]

$$x_i = x_{i-1} \qquad (2.3\text{-}4)$$

and the redundancy test is

$$x_{i-1} - k \leq x_i < x_{i-1} + k \qquad (2.3\text{-}5)$$

Each x_i failing the test is saved as non-redundant and is used as the new reference for subsequent tests. This method is also known as the floating point aperature, simply because the tolerances follow the input values. An example is shown in Figure 2-2.

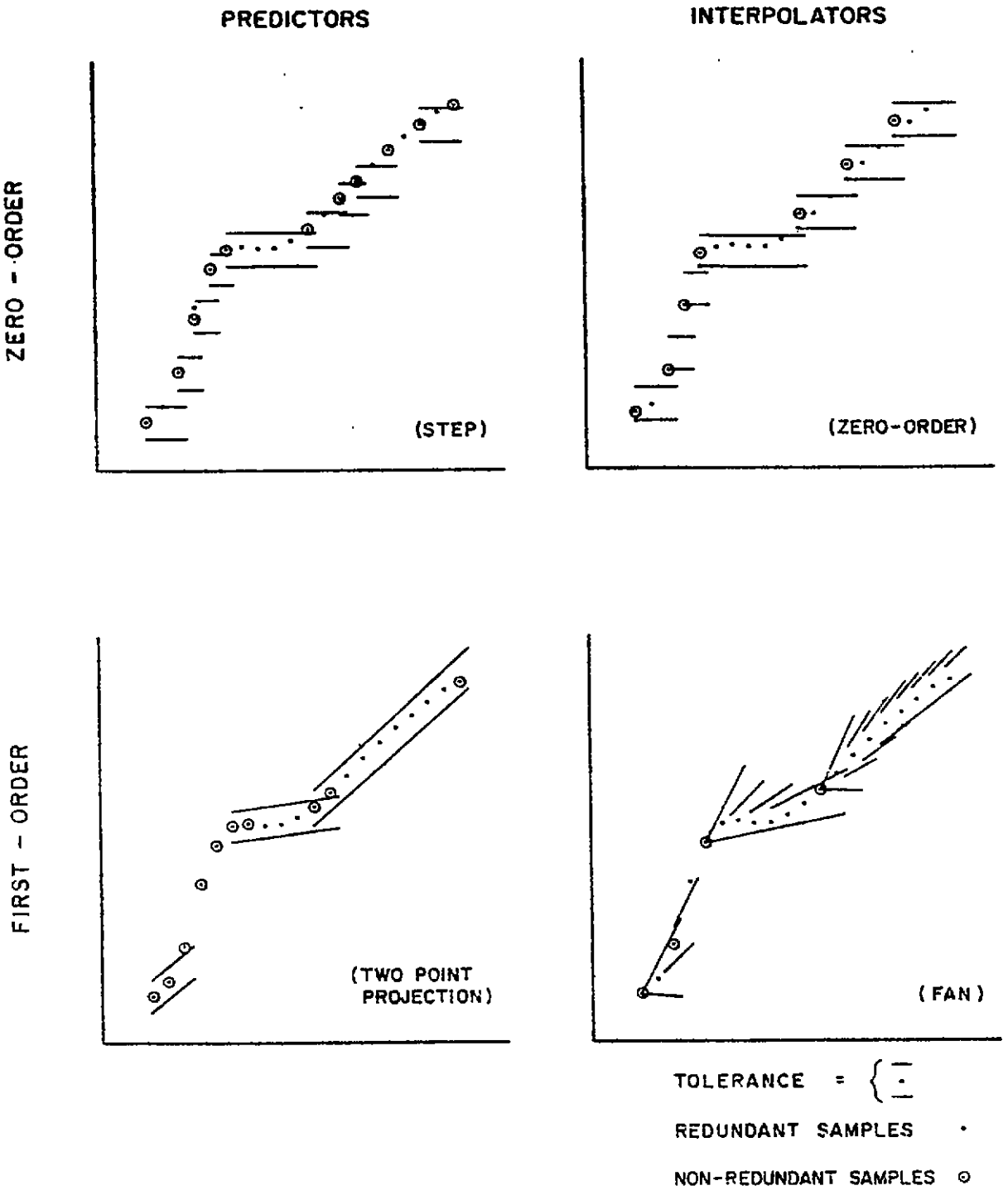


FIGURE 2-2. Variable rate compression methods.

2.3.2.1.2 FIRST-ORDER PREDICTOR

Setting n = 1 in equation (2.3-3), the first-order predictor

$$x_i = x_{i-1} + \Delta x_{i-1}$$

$$= 2x_{i-1} - x_{i-2} \qquad (2.3\text{-}6)$$

is obtained.[2-1] The extrapolation equation is a straight line drawn between the last two data points. Since Δx_{i-1} represents the change between the previous two samples, the predicted sample is the previous sample plus the change that occurred between the previous two samples.[2-5] The redundancy test is the same as that shown in expression (2.3-5). When a sample falls outside the tolerance, the preceding sample is considered nonredundant.[2-11]

It follows that higher order predictors can be built by considering more past data. Although the higher order predictors will tend to provide high compression efficiency on highly active data, experience has shown that a low order predictor will provide equal or greater compression efficiency for most telemetry data.[2-7]

The compression efficiency is basically the fidelity of reconstructing the original waveform with respect to the amount of redundancy reduction. Although there are certain trade-offs with respect to the variance in the data and the type of predictor to be used, general rules have been established. The zero-order predictor is prefectly matched to data which vary as step functions, such as data calibrations or discrete events. Because of horizontal tolerance limits the zero-order predictor is at a disadvantage where data activity is high with many vertical series of adjacent points. However, in the presence of noise only the zero-order predictor tends to set up strictly horizontal limit lines which are automatically parallel to the noisy, actionless data. In the presence of noise spikes, or wild points, the zero-order predictor works well since it

automatically keeps those points and does not have to keep a point for every vertical increment of one tolerance magnitude. The noise can then be effectively compressed if the tolerance limits are sufficiently wide.[2-9]

Because the first-order predictor is responsive to changes in the data it generally works best on data exhibiting a high level of vertical activity and relatively low noise. A disadvantage of the first-order predictor is the possibility of getting hung up on heavy noise, and while the zero-order predictor is handicapped by vertical variations of the data, noise tends to reduce the efficiency of the first-order to an even greater extent. When these conditions are mixed, i.e., high noise-high vertical activity, or low noise-little vertical activity, the two methods generally perform with equal efficiency.[2-9]

2.3.2.2 INTERPOLATORS

Prediction techniques are based on the assumption that the data will remain relatively constant from one time interval to the next. If the data vary continuously or are corrupted sporadically by noise, the redundancy reduction efficiency of the predictor generally will be reduced.[2-7] In such cases the compression efficiencies could be increased if both past and future data samples could be used. This process of determining redundancy after the sample has been examined is called interpolation.[2-5] Interpolators differ from predictors in that all sample values between the last transmitted value and the present value affect the interpolation.[2-1] Interpolation uses present samples to determine where past samples should have been and compares this prediction to the actual position of the past sample.[2-5]

2.3.2.2.1 ZERO-ORDER INTERPOLATOR

The zero-order interpolator, like the zero-order predictor, is a horizontal aperture device with "step-wise" tolerance

limits. However, whereas the predictor utilizes only knowledge of the initial sample value in locating the aperture, the zero-order interpolator operates by maximizing the length of time the original waveform stays within the aperture.[2-3] One method of implementing this is to place one of the tolerance bounds at the first point and consider this to be the maximum or minimum value in the redundant data set, depending on the slope of the curve. The aperture is initially centered at $x_o \pm k$ and the entire space is 2k. Whenever a sample exceeds the 2k limits, that sample is used to initiate the next tolerance band and the transmitted sample is the average of the maximum and minimum sample values in the tolerance band.[2-5]

$$x_t = \frac{x_{min} + x_{max}}{2} \qquad (2.3\text{-}7)$$

where x_t = transmitted sample,

x_{min} = smallest sample value in the redundant set,

x_{max} = largest sample value in the redundant set.

The spread that can be tolerated in the zero-order interpolator is strictly dependent upon the predefined error. The value transmitted is approximately the centroid of that redundant data set.

2.3.2.2.2 FIRST-ORDER INTERPOLATOR

The implementation of the first-order interpolator may take several forms; however, the most common is the "Fan Method" proposed by Gardenhire. This involves computing two slopes, both originating at the last transmitted sample, directed to the upper and lower tolerance limits of the next sample. These slopes are used to test the subsequent sample, and if it falls within the tolerance limits, a new, more restrictive fan, defined by the new tolerance limits, is used to test the subsequent point. As slopes are drawn from one sample around future samples, only the most restrictive slope above and the most restrictive one below are stored. The implementation of this is relatively simple and involves

little data storage since only five words of memory are necessary - the two slopes, the original sample, the last sample and the selection tolerance - regardless of how many samples are between the end points.[2-4] Whenever a sample exceeds the tolerance of the fan, the preceding sample is used as the origin of the next set of tolerance fans.

Since future samples must be examined to determine redundancy, the transmission of the non-redundant sample will be delayed. Thus there may be a major disadvantage in attempting to use interpolators for real-time processing.[2-5]

The predictors use only past, transmitted samples as a basis for future prediction to determine redundancy. However, since they use the set of future data to determine if a particular sample should be transmitted, they have a distinct advantage over the predictors. If the sample contains noise, the noise will be predicted to occur in the next sample. Therefore, that sample will probably fail the redundancy test. This pattern could continue at each succeeding sample making it difficult for the predictor to provide stable, non-redundant data. By using knowledge of future variations in the data, interpolators tend to reduce the effects of noise in transmitting non-redundant samples, and require a lower signal-to-noise ratio than the predictors.[2-5]

In making visual comparisons of the efficiency of the various redundancy reduction techniques on telemetry data, Lunsford observed that the first-order predictor tends to retain data peaks better than either the zero-order predictor or the first-order interpolator. The advantage of the first-order predictor over the zero-order algorithm is that the first-order limits generally have slope when approaching a data peak so that the upward or downward trend of points after the peak is picked up sooner than if the lines were horizontal. While the first-order interpolator has the same advantage as the first-order predictor, it does not define peaks as well for identical tolerances, because the limits for redundancy are wider.[2-9]

On the basis of examining sixteen telemetry functions with both predictors and the first-order interpolator, along with variations of these methods, Lunsford concludes that the zero-order interpolator should compress as efficiently, if not more, than any of the other three methods.[2-9] However, a major factor affecting the efficiency of each compression algorithm is the tolerance selected. Although Figure 2-2 illustrates each redundancy reducing method with essentially the same tolerance, the optimum tolerance is dependent upon the technique and data characteristics.

2.3.2.3 TOLERANCE

Once the decision concerning the type of compression method to be used is made, the size of the tolerance limits must be determined. Since noise is essentially random redundant data, the tolerance generally should be set large enough to enable the algorithm to suppress noise. Secondly, the tolerance should provide a relatively high compression ratio without significantly distorting the active data.[2-9] A priori knowledge of the data characteristics is necessary in choosing the optimum tolerance. The compression ratio which is an important factor in determining the effectiveness of the compression algorithm and tolerance is defined as:[2-5]

$$CR = \frac{\text{Total number of samples}}{\text{Number of significant samples}} \qquad (2.3\text{-}8)$$

Figure 2-3 shows the increase in compression ratio for different redundancy reduction methods over a tolerance range.[2-9] The tolerance limits are expressed in percentage of amplitude bandwidth, and the vertical scale shows the average compression ratio for 16 telemetry functions tested. These curves are intended to show approximate relative increases in compression ratios vs tolerances for each predictor.

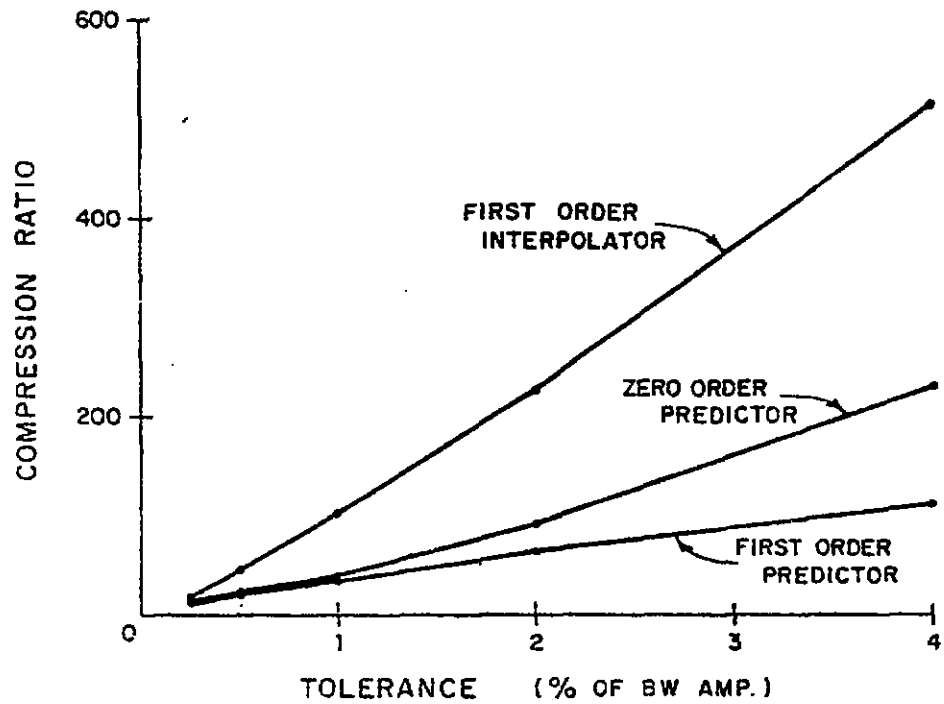


FIGURE 2-3. Compression Ratio Improvement for Different Tolerances.

2.3.2.4 ERRORS

Because of the various uncertainties in choosing the method and tolerance, information will be lost and fidelity of the original data will be affected. Gardenhire considers the tolerance to be an estimate of the maximum guaranteed error, and within this range provides typical error distribution curves for the redundancy reduction methods.[2-4] The results for 401 samples and a tolerance of .5% are shown in Figure 2-4 for the three methods. The curves show that the associated error distributions are far different from normal distribution curves. For the first-order interpolator the errors are more evenly distributed over the entire tolerance band while for the first-order predictor they peak at a very low error. The zero-order interpolator peaks at a higher error, but because of the relative distributions its mean error is lower than that of the first-order interpolator.

2.3.2.5 RECONSTRUCTION

Restoration of the data to its original form within the tolerances already determined may be necessary whenever further processing requires that the data be input at a fixed rate. This may be necessary at the receiving end when the data is compressed for transmission. Because of the differences in the algorithms used to compress the data, there are some considerations which affect the decompression, or reconstruction problem. Basically the reconstruction method is determined by the method of redundancy reduction.[2-6] Zero-order reconstruction fills in redundant samples which are equal to the last sample transmitted until a new sample is received. The first-order reconstruction process basically consists of connecting non-redundant values with straight lines through linear extrapolation.

In the real-time sense, the predictors, which were described herein, are relatively easy to decompress. With the zero-order case, all redundant samples, which must be reconstructed, will be within the original tolerance bounds but may not follow the original waveform,

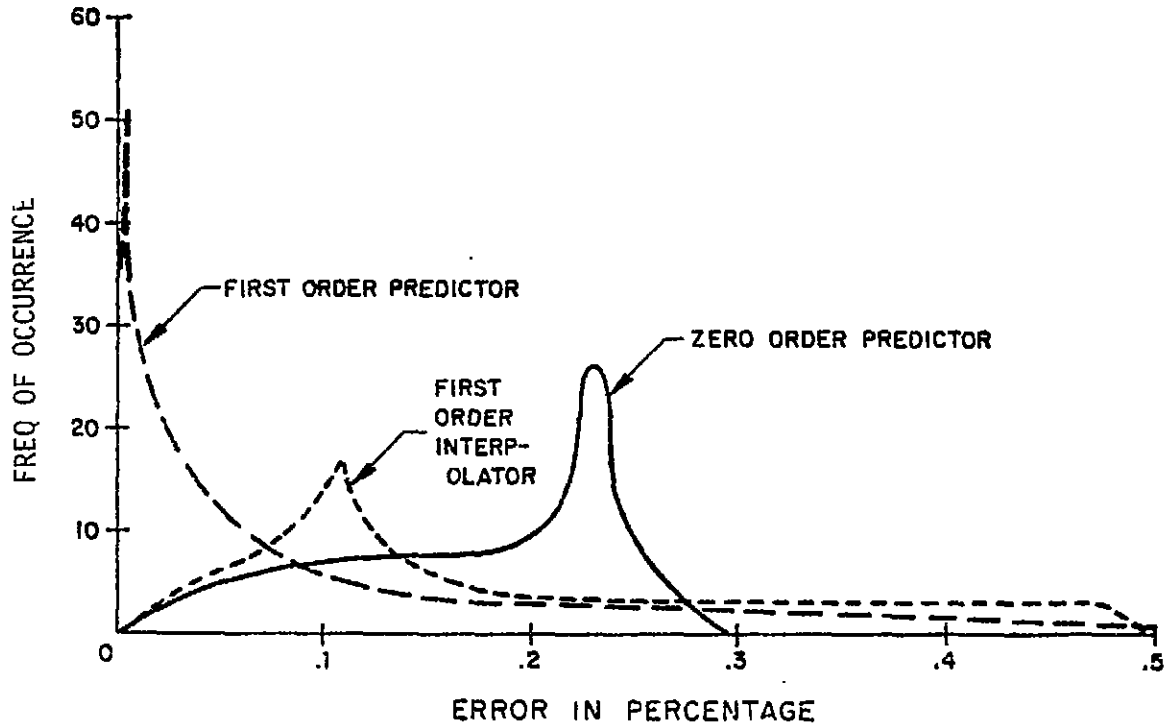


FIGURE 2-4. Typical Error Distribuiton Curves for Redundancy Reduction Methods.

especially if the tolerance bounds were large. The reconstruction of the redundant data removed by the first-order predictor uses the fact that after the first two samples are transmitted, Δx, the change in x which defines the slope of the tolerance bounds, is known and can be used to reconstruct the redundant samples until the next non-redundant sample is transmitted.[2-4]

The interpolators as described herein present a problem for reconstruction, especially when data are sampled for transmission. When using these algorithms, the non-redundant data values and slopes are not known nor transmitted until the longest possible line segment has been fitted to the data. This makes it impossible to reconstruct the original data without imposing some delay. The delay may be a major problem, especially if the data values remain within tolerance during the entire test.[2-6] However, because all the data variations on the compression end are known, the reconstructed samples tend to provide greater fidelity with the original data.

The zero-order interpolator transmits average estimates of the data in the tolerance bands. Therefore, the reconstructed data tend to follow the most likely estimate of the original redundant data. The first-order interpolator has a similar advantage. Since all sample points fall within the aperture space, there is no excess over the boundary as may exist with the two predictors.[2-3, 2-4]

REFERENCES

[2-1] Andrews, C. A., et al, "Adaptive Data Compression," Proceedings of the IEEE, vol 55, no. 3 (March 1967)

[2-2] Brunet, R. D., "DATVAL," Technical Note #3289-1044 (Pacific Missile Range, March 1965)

[2-3] Ehrman, L., "Analysis of Some Redundancy Removal Bandwidth Compression Techniques," Proceedings of the IEEE, vol 55, no. 3 (March 1967)

[2-4] Gardenhire, L. W. , "Redundancy Reduction - The Key to Adaptive Telemetry," Proceedings National Telemetering Conference, Section 1-5 (Los Angeles, CA, 1964)

[2-5] Gray, L. A., "Data Compression," a thesis submitted at the Dept. of Electrical Engineering, U. S. Naval Post Graduate School, Monterey, CA (June 1969)

[2-6] Kortman, C. M., "Redundancy Reduction - A Practical Method of Data Compression," Proceedings of the IEEE, vol 55, no. 3 (March 1967)

[2-7] Kortman, C. M., "Data Compression by Redundancy Reduction," IEEE Spectrum (March 1967).

[2-8] Lemieux, C., "Quality Assurance Processes," Technical Note #3420-2-72 (Pacific Missile Range, August 1972)

[2-9] Lunsford, B., "Feasibility of Data Compression," Technical Memorandum SC-TM-71-0493 (Albuquerque, NM: Sandia Laboratories, September 1971)

[2-10] Melton, R. A., "Data Compression Final Report," AFFTC (FTTSR) (August 1969)

[2-11] Smart, W. M., Combination of Observations (Cambridge, England; Cambridge University Press, 1958)

[2-12] _____, "Error Analysis and Methods for Estimating Errors in Position Velocity, and Acceleration Data," DR&CWG, IRIG, RCC Document 119-71 (May 1971)

CHAPTER 3

TRANSFORM METHODS/LUMPED PARAMETER TECHNIQUES

3.1 INTRODUCTION

Although the techniques described in this chapter are often thought of as analysis rather than data compression techniques, they can be used not only to compress data output from a computer, but also data stored internal to it. They also allow the user to make more intelligent conclusions than could be attained by simply inspecting the raw data.

In real-time data reduction it is imperative that the test conductor be presented with information he can assimilate in as short a time as possible. For example, is it desirable to reduce a long time history into a small number of computed parameters which characterize the complete time history, or to combine several parameters into one result upon which a decision might be based? The methods given here cannot only save considerable time, money, and paper when used judiciously in assessing the results of an experiment, but also will give more incisive pinpointing of what actually happened in the experiment. When properly utilized in real time, the test conductor or flight controller can leave the display room with full knowledge of his results rather than waiting several days for stacks of computer listings which are difficult to assimilate.

The Fourier Transform and Power Spectrum allow display of information related to the frequency content in the data. The Walsh Transform allows computation and display of information related to the number of zero crossings in the data. The transfer function allows representation of large quantities of data collected from a complicated system by a relatively small number of coefficients. Non-dimensionalized parameters allow combining of several parameters into one; both for reducing the output required, and for ease of assimilating information.

3.2 FOURIER TECHNIQUES

Fourier Analysis has been a rich area in applied mathematics for over 150 years. However, only in recent years, with the growth of digital computers and the introduction of the Fast Fourier Transform, is the full potential of this subject being realized. The ability to readily calculate the discrete Fourier Transform provides a very appealing data compression technique.

The definition of the Fourier Transform of a function, f(t), is given by the well-known integral

$$\bar{F}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t)\ e^{-i\omega t} dt \qquad (3.2\text{-}1)$$

with the reciprocal formula for the inverse.

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(\omega)\ e^{i\omega t}\ d\omega \qquad (3.2\text{-}2)$$

When t is time then ω is the frequency in radians/sec.

For digital data the Discrete Fourier Transform must be used and is defined as

$$F(\omega) = \frac{h}{\sqrt{2\pi}} \sum_{k=0}^{N} f(kh) e^{-i\omega kh} \qquad (3.2\text{-}3)$$

where h is the sampling interval.

The highest frequency discernable in descrete data equals $\frac{2}{h}$ and the finest resolution between frequencies equals $\frac{1}{Nh}$. Since $\omega=2\pi f$, where f is the frequency in cycles/sec, then the above definition becomes

$$F\ (\omega) = F\ \left(\frac{2\pi j}{nh}\right) = F(j) = \frac{h}{\sqrt{2\pi}} \sum_{K=0}^{N} f(kh)\ \ e^{\frac{-ikj}{N}}\ ,$$

$$j = 0,\ 1,\ 2\ \ldots\ N \qquad (3.2\text{-}4)$$

By appropriately rewriting equation (3.2-4) advantage may be taken of redundant calculations to significantly reduce the amount of computing required in calculating $F(\omega)$. Using this approach in the early 1960s, Blackmun and Tukey developed the Fast Fourier Transform (FFT). Today most computing organizations have software or hardware implementations of the FFT. Hence, it is possible to compute the Fourier Transform routinely on discrete sequences which would have been impossible before the FFT was developed. Not only is it possible to perform this computation in the batch mode, but it is also possible in many cases to perform it in near realtime.

A quantity closely related to the Fourier Transform is the Power Spectrum. This function is defined as the Fourier Transform of the autocorrelation function; however, it can be shown that this definition reduces to just the square of the absolute value of the Fourier Transform. That is

$$G_f(\omega) = R_e \{f(\omega)\}^2 + I_M \{f(\omega)\}^2 \qquad (3.2\text{-}5)$$

This function gives an indication of the distribution of the power as a function of frequency in the data being analyzed.

The Fourier Transform and Power Spectrum can be used for certain categories of experiments to greatly compress the amount of data input required to assess the results of the experiment. The most extreme example of this compression can be seen by considering the case of a pure sine wave, $f(t)=\sin \omega t$. The Power Spectrum for this case will be the delta function, $\delta(\omega-\omega_o)$; that is, all the power in the function is concentrated at ω_o. For the realistic case of finite data length, the Power Spectrum will be represented by a spike as shown in Figure 3-1.

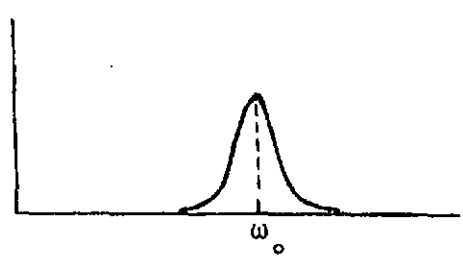


Figure 3-1

Hence, for this particular case, what could have been a sequence of thousands of points in the time domain is reduced to one pertinent point in the frequency domain.

Although it is very rare that a pure sign wave is encountered in practice, it is often true that most of the energy in a parameter is concentrated in a few narrow frequency ranges and that a good approximation of the parameter is given by a sum of sign waves in these ranges. With some a priori knowledge of the outcome of a test, a test controller can limit his output to cover the frequency range of interest and then not only significantly reduce the quantity of data output, but also have the results in a form from which conclusions can be drawn.

The Fourier Transform can also be used for saving computer storage requirements and for reducing the bit rate required in transmitting data. In many cases, the Fourier Transform of a signal or a curve is dominated by relatively few of the F(j) given in equation (3.2-4). In such cases only the F(j) which contribute significantly to the curve must be stored or transmitted. By storing or transmitting only those significant F(j) rather than the complete signal in the time domain, mass memory requirements or channel bandwidth requirements can be significantly reduced.

3.3 TRANSFER FUNCTIONS

The transfer function is defined as the ratio of the system input to the system output in the Laplace domain. It is usually used to characterize the frequency response of a system.

A constant coefficient linear system can be represented by the following vector differential equation.

$$\frac{d\bar{X}}{dt} = A\bar{X} + B\bar{U} \qquad (3.3\text{-}1)$$

Here $\bar{X}$ is the state or output vector, $\bar{U}$ is the control or input vector, A is the state transformation matrix and B is the control or input matrix. For a multiple-input multiple-output system the vectors $\bar{U}$ and $\bar{X}$ contain all pertinent input and output parameters, respectively.

In practice it is desirable to know the transfer function of one of the output parameters with respect to one of the input parameters. For such a case, it can be shown that the relationship between the input and output can be derived from equation (3.3-1) in the following form:

$$a_o \frac{d^n x}{dt^n} + a_i \frac{d^{n-1} x}{dt^{n-1}} + \ldots a_n X = b_o \frac{d^m u}{dt^m} + b_1 \frac{d^{m-1} u}{dt^{m-1}} + \ldots b_m u \qquad (3.3\text{-}2)$$

Here, x and u are particular components of $\bar{X}$ and $\bar{U}$.

The transfer function H(s) is obtained by taking the Laplace Transform of both sides of (3.3-2) and obtaining the ratio of X to U, to give the following:

$$H(s) = \frac{X(s)}{U(s)} = \frac{b_o s^m + b_1 s^{m-1} + \ldots + b_n}{a_o s^n + a_1 s^{n-1} + \ldots + a_n} \qquad (3.3\text{-}3)$$

Hence, the set of coefficients denoted by the a_i's and b_i's characterize the relationship between x and u. If i_ω is substituted for s, then H becomes the system frequency response function.

The transfer function can be calculated in several ways, among them: Fourier Transform, Z-transform, and parameter identification. In the Fourier Transform technique, the discrete Fourier Transform of both the input and output is taken and substituted into the left hand side of (3.3-3). Then a rational function numerical fit is made to the transformed data to give the a_i's and b_i's.

In the Z transform technique, the transformation $Z=e^{hs}$ is used to convert the differential equation (3.3-2) to a difference equation. Data values in the time domain are then substituted into the difference equation, usually giving an overdetermined system of linear equations with the a_i's and b_i's being the only unknowns. The a_i's and b_i's are then solved for by the method of least squares.

In the parameter identification method, the matrices A and B are usually determined by finding those respective values which will give the solution $\bar{X}(t)$ of (3.3-1) which most closely matches a set of measurements of $\bar{X}$, given, also, measurements for $\bar{U}$. There are several techniques used in parameter identification; among them are maximum likelihood, Newton-Rapheson, and Quasilinearization. The coefficients a_i and b_i can be easily determined from the matrices A and B. The details of parameter identification techniques are beyond the scope of this document. However, further detail may be found in references [3-3] and [3-4]

For systems which are approximately linear, the transfer function can be used to reduce a long time history of data for system output and input to a small set of coefficients which relate the two. Also, by looking at the roots of the numerator and denominator of the transfer function, we can determined the stability characteristics of the system. Hence, a test conductor who is analyzing his data in a near real-time mode will immediately have all the information needed to make decisions on the test. In this case, not only would a large stock of tabulated data be awkward to work with, it would also not provide him with the information needed to assess the results of the test. Hence, using the transfer function not only reduces the quantity of data output, but also provides the user with information in a form conducive to decision making.

3.4 WALSH TRANSFORMS

The Walsh Transform is analagous to the Fourier Transform in that a function or signal is represented by a series of orthogonal functions.

Just as the Fourier Transform is useful for representing signals composed of oscillatory components, the Walsh Transform is extremely useful in representing signals composed of a number of discrete level changes. The analog to the frequency for the Fourier Transform is the "sequency" or number of zero crossings for the Walsh Transform.

The orthagonal functions used in performing the Walsh Transform are known as Walsh Functions. The first 16 Walsh Functions are shown in Figure 3-2.

If f is a data vector of length N; then the one-dimensional Walsh Transform F of f is defined as

$$\bar{F} = \frac{1}{\sqrt{N}} \; W_n \bar{f} \qquad (3.4\text{-}1)$$

where W is an NXN matrix, the rows of which are the sampled Walsh Functions W_i. The inverse transform is given by

$$\bar{f} = \frac{1}{\sqrt{N}} \; W_n \bar{F} \qquad (3.4\text{-}2)$$

Hence, the forward and inverse transforms can be implemented by the same hardware and software.

The Walsh Transform can be used for data compression in a similar manner to the Fourier Transform. In certain cases only a relatively small number of the elements of $\bar{F}$ are significant. In such cases only these significant components need be retained for a large savings in memory or channel bandwidth to be achieved. The signal can be reconstructed using equation (3.4-2) with the insignificant components set to zero.

This technique has been especially useful in reducing bandwidth requirements for transmitting digitized video signals. In this case the screen image is composed of a relatively few discrete shades. The Walsh Transform is highly suited for representing the signal which generates these shades. References [3-5] and [3-9] give further details on the use of this technique.

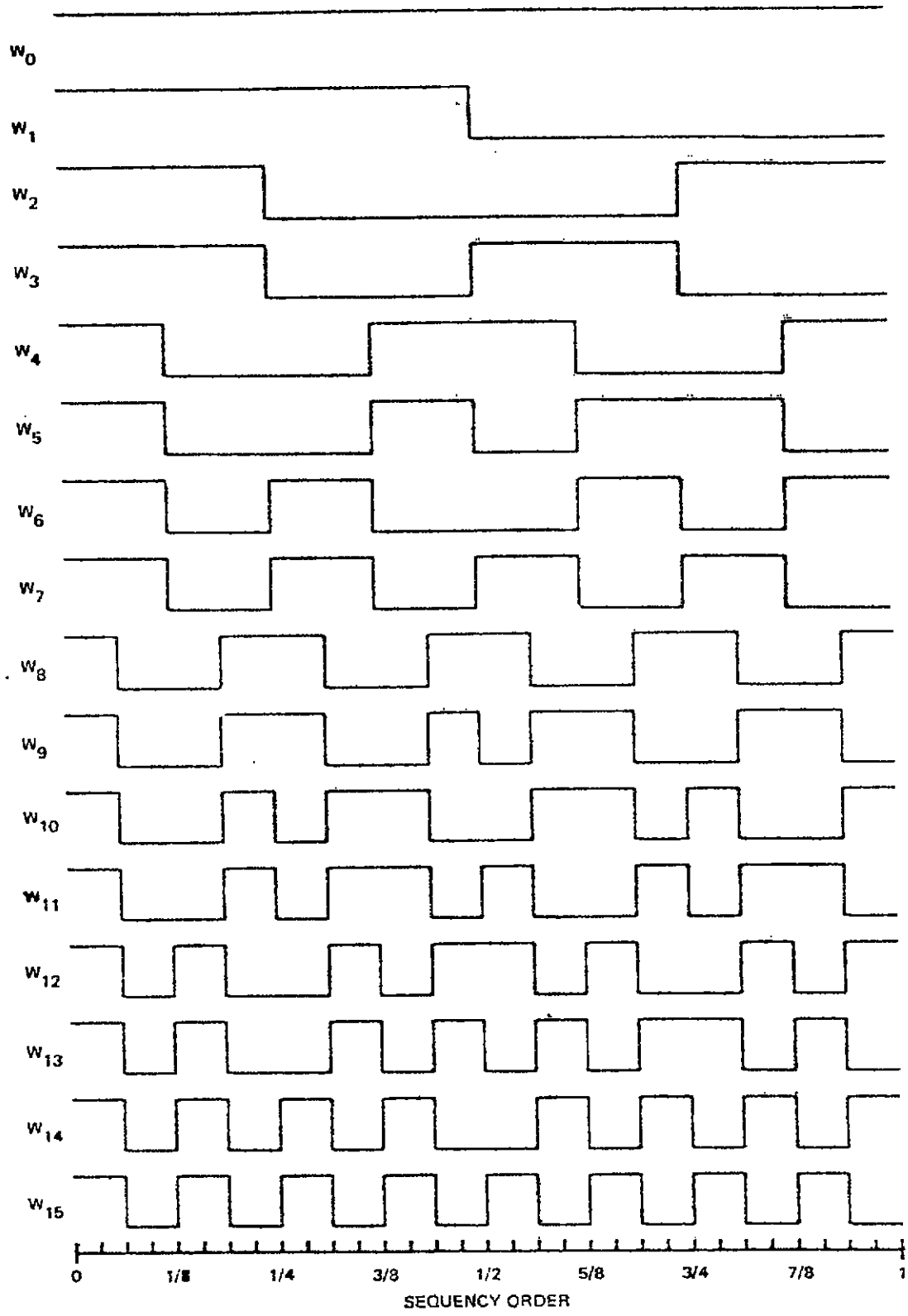


FIGURE 3-2. Sequency Order of First 16 Walsh Functions

3.5 NON-DIMENSIONALIZED PARAMETERS

Non-dimensionalized parameters have been used for many years by aerodynamicists for characterizing aerodynamic forces and moments in fluid flows. These non-dimensionalized parameters can be viewed as data compressors in that they lump together several parameters into one parameter. As in cases discussed previously, this reduction also usually means that the lumped parameter can be more easily interpreted than the several quantities could be separately.

For example, in incompressible viscous fluid flow through pipes, the Reynold's number, which is a non-dimensioned parameter made up of four physical quantities: density, viscosity, pipe diameter, and flow velocity; uniquely determines the value of the resistance coefficient for a given surface geometry of the pipe. Hence, there is no need to obtain data at all possible densities, pipe diameter, and flow velocities, but only to run experiments at varying values of the Reynold's number.

The extent to which a group of related quantities can be reduced to dimensionless parameters is governed by the Buckingham π theorem. This theorem states that, given a physical equation $f(X_1,X_2,X_3, \ldots X_N)=0$, where the X_i's are dimensional physical quantities related to the physical phenomenon of interest, that there can be N-M dimensionless quantities describing the same phenomenon, given, as follows:

$$f(X_1, X_2, X_3, \ldots X_N) = \emptyset\ (\pi_1,\pi_2,\pi \ldots \pi_{N-M}) = 0 \qquad (3.5\text{-}1)$$

where M is the number of fundamental physical dimensions in the problem. In pure mechanics problems, the fundamental units are mass, length, and time. Hence, by non-dimensionalizing, the number of quantities to be considered can be reduced by three.

As for the previous cases discussed, not only is a reduction in quantity of data achieved, but also it is easier to assimilate the results of a test by considering the reduced set of dimensionless

parameters instead of the complete set of physical quantities. Thus, for example, a test conductor should be able to gain considerably more information from viewing a force coefficient than from viewing separately the force, density, and velocity which constitute the force coefficient. In conclusion, before any new data analysis is set up, careful consideration should be given to using appropriate non-dimensionalized parameters for reducing the quantity of data to be output.

3.6 PITFALLS

The methods described in this chapter can be extremely useful in compressing data or increasing the information content of data to be presented. However, as is true with any mathematical technique, extreme care should be taken in using these methods. The user should be as familiar as possible with the physical phenomenon which is being represented and should make a careful assessment of whether the techniques here are applicable to his problem.

The Fourier Transform can give erroneous results when improperly used. When a truncated Fourier series is used on non-periodic data, spurious oscillations can be induced when the inverse is taken. This property, known as the Gibbs Phenomenon, is described in detail in any good reference on Fourier Transforms. Analogous errors are also introduced because of the finite data length in the time domain. If the Fourier Transform is blindly applied, the user may find that a significant compression ratio has been achieved at the expense of losing all the relevant information in the data. Similar pitfalls can occur in the use of the Walsh Transform.

The transfer function can also be abused as a data compression device. The most common pitfall occurs when the system from which the data is taken is not adequately described by a set of constant coefficient differential equations. For example, the system may contain significant non-linearities or time varying coefficients. In such cases, the coefficients in the transfer function will not give faithful reconstruction of

the original data and will give an erroneous picture of the process under test.

The only pitfall which can occur from use of non-dimensionalized parameters is incorrect modeling of the system under test. However, careful modeling should always be done, regardless of the data collection or data analysis technique to be used.

Avoidance of the pitfalls listed here is accomplished through careful study of a technique and how it applies to the physical process being tested. If possible, the system should be modeled and a simulation developed. The data compression technique being considered should then be tested on the simulated data. After the compression has been achieved, the data should be reconstructed to determine how much information was lost during the compression. The user should then choose the technique which gives the best compromise between compression ratio and fidelity of the reconstructed data to the original data.

REFERENCES

[3-1] BENDAT, J. S. and PIERSOL, A.G., Random Data Analysis and Measurement, (New York; John Wiley & Sons 1971)

[3-2] RABINER, L.R. and RADER, Charles, M., Digital Signal Processing, (New York; IEE Press, 1972)

[3-3] BURTON, R. A. and LATHAM, L. J., "Verification of S-3A Lateral-Directional Approach Characteristics Using a Maximum Likelihood Parameter Identification Technique," Naval Air Test Center Technical Report No. SAR-75, Patuxent River, Maryland (1976)

[3-4] CLIFF, Eugene M., "A Theoretical Review of Some Analytical Methods for Identification of Dynamical System Parameters," Air Force Flight Test Center Office Memo (1973)

[3-5] Symposium Proceedings, "Application of Walsh Functions," Washington, DC (1971)

[3-6] Grumman Aircraft Corporation, "The Z-Transform Method of Calculating Frequency and Damping," Internal Office Memorandum (New York: Calverton, 1968)

[3-7] WALSH, J. E., "A Closed Set of Normal Orthogonal Functions," American Journal of Mathematics, vol. 45, pp. 5-25 (1923)

[3-8] HARMUTH, H. F., "A Generalized Concept of Frequency and Some Applications," IEEE Trans. on Info. Theory, vol. IT-14, no. 3, pp 375-382 (May 1968)

[3-9] McCALL, D. C., "3-1 Data Compression Via Walsh Transform" Naval Electronics Laboratory Center, San Diego, California (1973)

[3-10] POWERS, J. W., "Image Data Compression Using Digital Holography," Rome Air Development Center, Technical Report 73-83 (New York: Griffiss Air Force Base, 1973)

[3-11] ESKANZI, S., Principles of Fluid Mechanics (Boston, Massachusetts; Allyn and Bacon, Inc., 1962)

[3-12] KOLK, W. R., Modern Flight Dynamics (Englewood Cliffs, New Jersey; Prentice-Hall, Inc., 1961)

CHAPTER 4

STATISTICAL REPRESENTATION

4.1 INTRODUCTION

Various statistical parameters are used to describe large groups of data. After the parameters are computed, the basic data may be stored or discarded. Other statistical techniques may be used to discard some individual pieces of data. The subjects in this chapter are discussed briefly. For details the reader is referred to the references.

4.2 PARAMETER ESTIMATION

In this paragraph, a group of data will be referred to as a sample. In order to summarize the information in a sample, certain representative values must be calculated. These representative values fall into two groups. One group measures the central tendency of the sample and the other measures the dispersion of the sample. Usually values from both groups are needed to summarize the sample.

4.3 MEASURES OF CENTRAL TENDENCY

The most common measure of central tendency is the arithmetic mean. If these are n values, X_1, $X_2 \ldots X_n$ in a sample the arithmetic mean, $\bar{X}$, is calculated by the formula

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i \qquad (4.3\text{-}1)$$

Two properties of the arithmetic mean are (1) the sum of the deviations from the mean are zero and (2) the sum of squares of the deviations from the mean is less than the sum of squares of the deviations from any other value. The arithmetic mean has the following advantages: (1) it is easily calculated, (2) it is easily understood, (3) it is commonly used, and (4) it lends itself to algebraic manipulation. On the other

hand, it has the disadvantage of being quite sensitive to extreme values and may be far from representative of the sample.

The midrange is a representative value which may be used to approximate the arithmetic mean. The midrange, M_R, is calculated by the formula

$$M_R = \frac{1}{2} (X_{MIN} + X_{MAX}) \qquad (4.3\text{-}2)$$

It is simply the arithmetic mean of the largest and smallest values in the sample.

It has the advantage of being easily and quickly calculated. Since it ignores the intermediate values, midrange has the disadvantage of being unrepresentative if either the maximum or minimum value is atypical of the values in the sample.

The median is often used to describe a sample. The median is that value for which half the values in the sample are less than the median value and half greater. When the sample values are arrayed in order of magnitude from lowest to highest, the median, M, is the (n+1)/2 value. If there are an even number of observations, the median is the arithmetic mean of the two middle values; i.e., for n values, X_i, where n is even, the median is

$$M = \frac{1}{2} (X_{n/2} + X_{n+1}) \qquad (4.3\text{-}3)$$

If there are an odd number of values, the median is the middle value; i.e., for n values, X_i, where n is odd, the median is

$$M = X_{\frac{n+1}{2}} \qquad (4.3\text{-}4)$$

The median is easy to calculate and is often more typical of the data than the arithmetic mean since it is not affected by extreme values. Some disadvantages are (1) that the values must be sorted and arrayed

before the median is computed, (2) it does not lend itself to algebraic manipulation, and (3) if the data fall into two distinct groups it could be misleading. Theoretically the probability is one half that an observation selected at random will be less than the median. The sum of the absolute values of the deviations from the median is less than the sum of absolute values from any other value. When there are several sample values which are identical, the median may not have half the samples below and above that theory indicates.

The data may be described by retaining only points which divide the sample into convenient groups. One such division is the division into percentiles. A percentile, P_p, is that value for which p% of the values are less than P_p and (100-p)% of the values are greater than P_p. When the values are arrayed in order of magnitude, then P_p is the p(n+1)/100th value if p(n+1)/100 contains a fraction; then the value is a linear interpolation between the two values on either side. If the value p(n+1)/100 falls outside the data, use the first or last value, whichever is appropriate.

As a simple example, consider the following set of data: 1, 2, 2, 3, 4, 5, 5, 5, 9, 11. The 95th percentile is the 95(11)/100 = 10.45 value or 11. The 80th percentile is the 80(11)/100 = 8.8 value or 5 + 0.8(4) = 8.2. The 20th and 25th percentiles are both 2. The median is the 50th percentile and in this example is 4.5. The 10 percentile numbers are referred to as deciles and the 25, 50, 75, and 100 percentile numbers are referred to as quartiles.

The mode is the most frequent value that appears in the sample. In the example in the previous subparagraph, the mode is 5. There can be several modes in a given sample. If all values in a sample are different, then there is no mode. When any value occurs more frequently than its neighbors, it is referred to as a relative mode. The most frequent value is called the absolute mode. There can be several absolute modes.

4.4 MEASURES OF DISPERSION

The measures of central tendency do not describe the spread of values. Three common values of dispersion are discussed here.

The Range, R, is the difference between the maximum and minimum values. It is calculated by the formula

$$R = X_{MAX} - X_{MIN} \tag{4.4-1}$$

The range is easy to calculate but has the disadvantage that it ignores intermediate values.

The variance and standard deviation may be considered together. The standard deviation is the positive square root of the variance. For this reason the variance is referred to by the symbol s^2 and the standard deviation by s. There are two ways to compute s^2. One uses n, the number of values in the sample, in the denominator. The other uses n-1. Both ways may be shown using the same formula. The method which uses n provides a biased estimator of the population variance while that with n-1 provides an unbiased estimator. An estimator is unbiased if its expected value is equal to the population parameter. The expected value of s^2, when n is used in the denominator, is $\frac{n-1}{n}\sigma^2$, where σ^2 is the population variance. The formulas given below are equivalent and selection of the one to use should be made by determining which one is the easiest to calculate. The first one given is usually the easiest for machine calculations. If the sample has n values, $X_1, X_2, \ldots X_n$ then

$$(1)\quad s^2 = \frac{\sum_{i=1}^{n} X_i^2 - \left(\sum_{i=1}^{n} X_i\right)^2/n}{\nu} \qquad \nu = n \text{ for biased estimator} \tag{4.4-2}$$

$$(2)\quad s^2 = \frac{\sum_{i=1}^{n} X_i^2 - \left(\sum_{i=1}^{n} X_i\right)^2/n}{\nu}, \quad \nu = n-1 \text{ for unbiased estimator} \tag{4.4-3}$$

$$(3)\quad s^2 = \frac{\sum_{i=1}^{n} (X_i - \overline{X})^2}{\nu}, \quad \overline{X} = \text{arithmetic mean of the sample.} \tag{4.4-4}$$

The coefficient of variation, C_v, is a measure of relative variation. It is computed from the formula $C_v = s/\bar{X}$ (4.4-5), where s is the sample standard deviation and $\bar{X}$ is the arithmetic mean of the sample. It has been observed that samples with numerically large values tend to vary widely and those with numerically small values tend to vary narrowly. In order to make a comparison of the variation among two groups of data with different magnitudes, the coefficient of variation may be used. It can be used to compare the variation in two samples which are measured in two different units; e.g., a comparison of variation in height with variation in weight.

4.5 COEFFICIENTS IN A MATH MODEL

A math model is simply an equation which relates an observed value, Y, to one or more known values, X_i. In practical cases most math models are linear. The reason is that linear equations are easy to manipulate and calculate. A math model is then of the form

$$Y = \sum_{i=0}^{n} b_i X_i \quad \text{where there are n+1 known values} \qquad (4.5\text{-}1)$$

The linear form can be used to deal with very general situations. In the case of a trajectory, position is represented by a second degree equation in time, viz,

$$y = b_o + b_1 t + b_2 t^2 \qquad (4.5\text{-}2)$$

If we let $X_o = 1$, $X_1 = t$, and $X_2 = t^2$, the second degree equation in one variable can be transformed into a first degree equation in three variables.

An equation of the form $y = ax^b$ can be transformed into a linear equation by taking logarithms. Specifically,

$$\log y = \log a + b \log X \qquad (4.5\text{-}3)$$

Here, $Y = \log y$, $b_o = \log a$, $b_1 = b$, $X_o = 1$, $X_1 = \log 4$.

A common method of determining the coefficients is to apply the method of least squares. To use the method, several observations of Y must be taken for various known values of the X_i's. Let

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \cdot \\ \cdot \\ \cdot \\ Y_n \end{bmatrix} \qquad X = \begin{bmatrix} X_{o1} & X_{11} & \cdots & X_{k1} \\ X_{o2} & X_{12} & \cdots & X_{k2} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ X_{on} & X_{in} & \cdots & X_{kn} \end{bmatrix} \qquad B = \begin{bmatrix} b_o \\ b_1 \\ \cdot \\ \cdot \\ \cdot \\ b_k \end{bmatrix}$$

$$\varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \cdot \\ \cdot \\ \cdot \\ \varepsilon_n \end{bmatrix}$$

Y is the matrix of n observed values.
X is the matrix of the n known points of the $k + 1$, X_i's.
B is the matrix of the $k + 1$ coefficients of the X_i's.
ε is the matrix of observational errors.

$$\text{Now } Y = XB + \varepsilon \qquad (4.5\text{-}4)$$

is the matrix equation of observations. The method of least squares assumes the errors to be independent, with mean zero, and a common variance, σ^2 (i.e., they are homoscedastic). The method of least squares finds the values of the coefficients which minimize the sum of the squares of the residuals. The symbol $\hat{}$, above a variable, will indicate that it is an estimate of that parameter. The sum to be minimized is

$$S = \hat{\varepsilon}'\hat{\varepsilon} = (Y - XB)'(Y - XB) \qquad (4.5\text{-}5)$$

A symbol, ´, indicates the transpose of the matrix.

When the partial derivatives are taken, set equal to zero and manipulated a bit, the following result is obtained:

$$X'X\hat{B} = X'Y \tag{4.5-6}$$

This is the matrix form of what are termed normal equations.

$$\hat{B} = (X'X)^{-1}(X'Y) \tag{4.5-7}$$

The estimate s^2 of σ^2 is given by

$$s^2 = \frac{1}{n-(k+1)} (Y'Y - \hat{B}'X'Y) \tag{4.5-8}$$

The matrix $(X'X)^{-1}$ is the variance-covariance matrix of the variances of the b's.

Let s^2_{bi} be the estimate of the variance of b_i and S_{bibj} be the covariance of b_i and b_j. Then the diagonal elements of

$$s^2(X'X)^{-1}$$

are the values of the S^2_{bi}

If $S^2_b =$ $\begin{bmatrix} S^2_{bo} \\ S^2_{bi} \\ \vdots \\ S^2_{bk} \end{bmatrix}$, then

$$S^2_b = s^2 \text{ diag } (X'X)^{-1} \tag{4.5-9}$$

The values of the variances may be used to determine confidence intervals for the estimates of the coefficients.

4.6 CORRELATION AND REGRESSION ANALYSIS

Correlation means the degree of association among variables. The quantities used to measure the correlation are termed correlation coefficients. Regression is a term for the methods used to determine the best functional relationship among variables. In statistics, when a dependent variable is expressed as a function of one or more independent variables, the function is termed a regression function. In other areas it is sometimes termed a response function. The statistical analysis of a regression function and the determination of the coefficients may not mean that a casual relationship must be made by a person well trained in the subject matter field in which the test was made.

A regression function is a math model. The discussion of least squares which appears in paragraph 4.5 also applies here. Polynomials of degree m may be considered as linear functions with the m+1 variables X_o, ... X_m where $X_o = 1$, and X_i is the i^{th} power of the variable. Nonlinear functions can often be linearized by a proper transformation. After the coefficients are computed they must be converted to the original terms. The example $y = a^b$ shown in paragraph 4.5 (equation 4.5-3) would have $a = 10^{bo}$, and $b = b_1$ where b_o and b_1 are obtained from the linear expression.

It is possible to determine the goodness of fit by examining the variance and sums of squares of the variables. Such an examination is called an Analysis of Variance. In the case of polynomials it is possible to decide whether the last term added has any significance. In general, it is possible to determine if several coefficients are significantly different from zero. The case of deciding whether a number of coefficients are different from zero is discussed here. An illustrative example is shown in the next paragraph. All the symbols and their

definitions are the same as those used in paragraph 4.5 concerning the coefficients of math models. Assume that it is desired to know if the last $p<k+1$ of the coefficients are significantly different from zero. To do this create two new matrices, X_ν and B_ν, where X_ν is the matrix formed by removing from X the p columns that correspond to the suspect coefficients and B_ν is the matrix formed by removing from B the appropriate p coefficients. Solve the reduced set of equations. This solution is:

$$\hat{B}_\nu = (X_\nu' X_\nu)^{-1}(X_\nu' Y) \tag{4.6-1}$$

with variance $S_\nu^2 = \frac{1}{n-(k+1-p)} (Y'Y - \hat{B}_\nu' X_\nu' Y)$ (4.6-2)

The following table should then be computed. This is called an analysis of variance table. The mean square column is the sum of squares divided by the degrees of freedom. The table is adapted from reference [4-9] as is the explanation following.

TABLE 4-1 ANALYSIS OF VARIANCE TABLE

Source of Variation	Degrees of Freedom	Sum of Squares	Mean Squares
Total	n	$Y'Y$	$\frac{1}{n} Y'Y$
Due to $k+1$ constants	$k+1$	$\hat{B}'X'Y$	K
Residual (from large solution)	$n-(k+1)$	$Y'Y-\hat{B}'X'Y$	s^2
Due to $k+1-p$ constants	$k+1-p$	$\hat{B}_\nu' X_\nu' Y$	A
Residual (from reduced solution)	$n-(k+1-p)$	$Y'Y-\hat{B}_\nu' X_\nu' Y$	s_ν^2
Due to additional p constants	p	$\hat{B}'X'Y-\hat{B}_\nu' X_\nu' Y$	p

$F = \frac{K}{s^2}$ is distributed as F with degrees of freedom $k+1$, $n-(k+1)$ and serves as a test of whether all $k+1$ constants account for a significant reduction in the error variance.

$F = \frac{P}{s^2}$ is distributed as F with degrees of freedom p, $n-(k+1)$ and serves as a test of whether the addition of the p coefficients accounts for a significant reduction in the error variance over that accounted for by the first $k+1-p$ constants.

The following illustrative example was adapted from reference [4-9]. The notation has been changed to conform to that used in this chapter. The numbers and computations are taken directly from the reference. The data are represented in tabular form below.

TABLE 4-2

Y	X_0	X_1	X_2
2	1	8	1
4	2	8	7
4	2	6	0
4	3	1	2
3	4	2	7
3	4	5	1

This corresponds to the situation

$$y = b_o x_o + b_1 x_1 + b_2 x_2 \qquad (4.6\text{-}3)$$

$$Y = \begin{bmatrix} 2 \\ 4 \\ 4 \\ 4 \\ 3 \\ 3 \end{bmatrix} \qquad X = \begin{bmatrix} 1 & 8 & 1 \\ 2 & 8 & 7 \\ 2 & 6 & 0 \\ 3 & 1 & 2 \\ 4 & 2 & 7 \\ 4 & 5 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix}$$

$$X'X = \begin{bmatrix} 50 & 67 & 53 \\ 67 & 194 & 85 \\ 53 & 85 & 104 \end{bmatrix} \qquad X'Y = \begin{bmatrix} 54 \\ 97 \\ 62 \end{bmatrix}$$

$$\hat{B} = (X'X)^{-1} X'Y = \frac{1}{239418} \begin{bmatrix} 12951 & -2463 & -4587 \\ -2463 & 2391 & -699 \\ -4587 & -699 & 5211 \end{bmatrix} \begin{bmatrix} 54 \\ 97 \\ 62 \end{bmatrix}$$

$$\hat{B} = \begin{bmatrix} 0.735 & 320 & 652 \\ 0.232 & 175 & 526 \\ 0.031 & 664 & 286 \end{bmatrix}$$

$$s^2 = \frac{1}{n-(k+1)}(Y'Y - B'X'Y) = \frac{1}{3}\,(5.808\ 473\ 038)$$

$$= 1.936\ 157\ 679$$

$$s = 1.391\ 4588$$

$$s_{bo} = 0.323\ 627$$

$$s_{b1} = 0.139\ 054$$

$$s_{b2} = 0.205\ 283$$

To test the significance of b_2, the last column is dropped from X.

$$\text{Then } X_v'X_v = \begin{bmatrix} 50 & 67 \\ 67 & 194 \end{bmatrix}$$

$$(X_v'X_v)^{-1} = \frac{1}{5211}\begin{bmatrix} 194 & -67 \\ -67 & 50 \end{bmatrix}$$

$$X_v'Y = \begin{bmatrix} 54 \\ 97 \end{bmatrix}$$

$$\hat{B}_v = (X_v'X_v)^{-1}\,(X_v'Y) = \begin{bmatrix} 0.763 & 193 & 245 \\ 0.236 & 422 & 951 \end{bmatrix}$$

$$P = \frac{1}{p}\,(\hat{B}'X'Y - \hat{B}_v X_v'Y) = \frac{1}{1}\,(64.191\ 527 - 64.145\ 461)$$

$$P = 0.046066$$

$$F = \frac{P}{s^2} = \frac{0.046066}{1.936158} = 0.024$$

The numerator has 1 and the denominator 3 degrees of freedom. At the 95% confidence level,

$$F_{0.95}(1,3) = 10.13$$

Since $F<10.13$, b_2 is not regarded as being significantly different, statistically, from zero. Therefore, it may be disregarded.

The preceeding discussion about least squares has been limited to the case where the variances of the observations were independent and equal. For a discussion of the cases where the observation errors are not equal and/or not independent the reader is referred to references [4-2] or [4-9].

Correlation only tells how well variables are related. The correlation coefficient, r, between two sets of data, each having n values, is computed by the following:

$$r = \frac{\sum_{i=1}^{n} (X_i-\bar{X})(Y_i-\bar{Y})}{\left[\sum_{i=1}^{n} (X_i-\bar{X})^2 \sum_{i=1}^{n} (Y_i-\bar{Y})^2\right]^{\frac{1}{2}}} \qquad (4.6\text{-}4)$$

where the X_i's and y_i's are the values in the two sets of data, $\bar{X}$ is the arithmetic mean of the X_i's and $\bar{Y}$ is the arithmetic mean of the Y_i's.

The range of r is $-1 \leq r \leq 1$. If the data are perfectly correlated $|r|=1$. If the data are uncorrelated $r=0$. Perfectly correlated means there is an exact linear relationship. If $r>0$, the slope of the fitted line will be positive. If $r<0$, the slope of the fitted line will be negative.

Of more importance in data compression is serial correlation. For a set of data which is not random there will be dependencies between

successive terms. Serial correlation is used to measure these dependencies. The coefficient of serial correlation of log k is the correlation coefficient between pairs of terms k units apart. Suppose a set of data contains n points X_1, $X_2 \ldots X_n$. The serial correlation coefficient of log k is given by

$$r_k = \frac{\sum_{i=1}^{n-k}\left[(X_i - \frac{1}{n-k}\sum_{i=1}^{n-k} X_i)(X_{i+k} - \frac{1}{n-k}\sum_{i=1}^{n-k} X_{i+k})\right]}{\left[\sum_{i=1}^{n-k}(X_i - \frac{1}{n-k}\sum_{i=1}^{n-k} X_i)^2 \sum_{i=1}^{n-k}(X_{i+k} - \frac{1}{n-k}\sum_{i=1}^{n-k} X_{i+k})^2\right]^{1/2}} \qquad (4.6\text{-}5)$$

Reference [4-6] shows how to use the correlogram of serial correlation coefficients to define envelopes of data. The reference shows that serial correlation preserves periodicity. The reference states that "the correlogram 'peaks out' on the positive side of zero whenever the input data completes a recognizable period of information." Tests may be applied to see if the various envelopes are statistically different. If they are not, the user has the option of discarding some. For details, the reader is referred to reference [4-6]. For other uses of the serial correlation see Chapter 2.

4.7 STATISTICAL SAMPLING

At times it is desirable to retain only a portion of the data available. The retained portion is called a sample. From the sample, inferences can be made about the collective properties of all the data. It is important to choose a sample that is large enough for valid inferences to be made and yet be small enough to meet considerations of time, computer storage limitations, ease of computation, cost, etc. Reference [4-9], pp 1-3, states, "Statistical inferences are basically predictions of what would be found to be the case if the parent populations could be and were fully analyzed with respect to the relevant characteristic or characteristics."

In order to draw correct inferences, the method by which a sample was chosen must be known. There are two general types of sampling: judgemental and chance. Samples selected by a chance method are called probability samples. If all the elements of a population have an equal chance of being selected, the sample is called a random sample. This is a necessary condition but is not sufficient for a sample to be a random sample. A sufficient condition for a sample to be random is that each possible sample must have an equal chance of being selected. Reference [4-9] notes, "experience teaches that it is not safe to assume that a sample selected haphazardly, without any conscious plan, can be regarded as if it had been obtained by simple random sampling. Nor does it seem possible to consciously draw a sample at random." The statistical techniques in this chapter are applicable to random samples and may or may not be applicable to other types of sampling.

One example of random sampling occurs when there is a block of data consisting of N points. A random sample may be obtained by assigning a number to each of the N values; then by using a random number generator, or random number table, to list a number, equal to the sample size, of different random numbers less than N. Select from the list of points only those whose position on the list corresponds to the random numbers.

Another example is the case when the data may be known to have occurred at different times. Suppose it is desired to estimate the turnaround time for jobs sent to a computer. Jobs sent to the computer are given a number which corresponds to the day, hour, and minute at which they are received. The same information is recorded when the job is finished. A random sample may be chosen by considering two digit random numbers in blocks of three. The first group will correspond to the day of the month, the next to the hour of the day, and the final to the minute. The job selected would be the job received closest to the random number and not previously selected.

Reference [4-9] gives two methods of determining the size of the sample to be drawn to estimate the mean of a population. It also lists

one method of determining the size of sample needed to estimate the standard deviation of a population to within a certain percent of its true value. One method is outlined below. For more details the reader is referred to the reference.

Assume it is desired to know the mean, m, of a population and that one is willing to take a risk, α, that the estimate is off by d or more. What size sample is needed? There is available an estimate, s, of the population standard deviation based on ν degrees of freedom.

From tables of the Student-t distribution, locate $t=t_{1-\alpha,\nu}$ for ν degrees of freedom. The sample size is then computed from the formula

$$n = \frac{t^2 s^2}{d^2} \qquad (4.7\text{-}1)$$

The value to use should be the smallest integer larger than or equal to n. If the mean, $\bar{X}$, of a sample of size, n, is computed, then with 100(1-α)% confidence, it can be said the interval from $\bar{X}$-d to $\bar{X}$+d includes the population mean, m.

4.8 ANALYSIS OF VARIANCE

Analysis of Variance is a technique used to separate variation in data into source components. The sources of variation considered in the Analysis of Variance are called variables or factors. The analysis of the variation depends on the particular grouping of the data or test design. An example of an analysis of variance procedure was shown in paragraph 4.6 of this chapter. That paragraph discussed the procedure to use to determine whether certain coefficients of a regression line were significant. Because of the large number of different applications, the reader is referred to the references for the particular technique to use in his application. References [4-9] and [4-10] give examples and work sheets to describe the various processes. Many of the books listed as references also describe work sheets and give examples.

4.9 SUMMARY

This chapter provided some statistical techniques which will allow a user to eliminate amounts of data. Everything described has been available for some time. The techniques may be termed merely classical statistics. Paragraph 4.2, which describes parametric estimation, mentions individual values which may be used to replace large groups of data. Paragraph 4.6, Correlation and Regression Analysis, gives techniques which enable the user to replace a large group of data with coefficients of a function or to eliminate one of two groups of data and replace it with a linear function which relates the remaining group to the one eliminated. Paragraph 4.7, Statistical Sampling, is presented because a smaller random sample may be taken from a larger group and allow inferences to be drawn about the collective properties of the larger group. Equation (4.7-1) shows how to compute the size sample to select if one desires to know the mean to within a given amount of uncertainty. Paragraph 4.8, Analysis of Variance, merely gives a definition and refers the reader to source documents.

REFERENCES

[4-1] Brunk, H.D. An Introduction to Mathematical Statistics, 2nd edition, (Waltham, Mass: Blaisdell Publishing Co., 1965)

[4-2] Chew, V. Regression Analysis with Correlated Errors, RCA TM Number 64-14 (AFETR, January 1964)

[4-3] Cochran, William G. and Cox, Gertrude M., Experimental Designs, 2nd edition, (New York NY: John Wiley and Sons, Inc., 1950)

[4-4] Draper, N.R. and Smith, H., Applied Regression Analysis (New York, NY: John Wiley and Sons, Inc., 1967)

[4-5] Freund, John E., Modern Elementary Statistics, 3rd edition (Englewood Cliffs, NJ: Prentice-Hall, Inc., 1967)

[4-6] Higgins, Ronald Clair, Statistical Procedures for Reducing the Amount of Technical Data That Must Be Collected and Stored During the Analysis of a Physical System, Dissertation submitted to the Graduate College of Texas A&M University (December 1974)

[4-7] Ostle, Bernard, Statistics in Research, 2nd edition, 3rd printing, (Ames, Iowa: Iowa State University Press, 1966)

[4-8] Tanabe, Marion K. Remote WRAP (Weighted Regression Analysis Program), Flight Test Technology Branch, AFFTC-TIM-75-9 (June 1975)

[4-9] _____, "Engineering Design Handbook - Experimental Statistics," AMCP 706-110, 706-111, 706-112, 706-113, 706-114, Headquarters, U.S. Army Materiel Command (December 1969)

[4-10] _____, Error Analysis and Methods for Estimating Errors in Position, Velocity, and Acceleration Data, Document number 119-71, Data Reduction and Computing Group, Range Commanders Council (May 1971)

CHAPTER 5

OPTIMAL ESTIMATION TECHNIQUES

5.1 INTRODUCTION

It is the intent of this chapter to consider data compression in relation to applied optimal estimation. In particular, this chapter will look at the implications of the use of such techniques in conjunction with discrete Kalman Filters. Starting with a statement of the discrete filtering problem, the compression problem will be set up and the objectives of its utilization discussed. For the most part, this chapter represents a survey of the use of data compression techniques in the area of applied recursive optimal estimation. It is not intended to be a theoretical treatise but rather a more practical approach oriented to problem solving. Both optimal and suboptimal compression techniques will be introduced along with a discussion of techniques for evaluating the suboptimal types.

"Optimal" data compression means that the data compression and corresponding estimation are performed in such a way as to minimize some selected measure of error and to utilize all information concerning the system dynamics, noise statistics and initial conditions. The optimal algorithms presented here calculate unbiased, minimum variance estimates and may, under certain conditions such as Gaussian error probability density functions, be optimal in several other senses such as least-squares, maximum likelihood, Bayesian *et al*.

An attempt has been made to include guidelines on such matters as compression design and recommended filtering and sampling rates. Generalized matrix forms and algorithms will be presented to the extent possible and a simple but illustrative scalar example will be carried throughout the section.

First consider the basic linear discrete model for which a multi-stage recursive data compression and estimation algorithm is to be constructed. The system is governed by the following equations:

$$\underline{x}(k+1) = \Phi(k+1,k)\underline{x}(k) + \underline{w}(k) \tag{5.1-1}$$

$$E[\underline{w}(k)] = 0 \tag{5.1-2}$$

$$E[\underline{w}(j)\underline{w}^T(k)] = Q(k)\ \delta_{jk} \tag{5.1-3}$$

where $\underline{x}$, the state vector, is propagated linearly by a transitition matrix Φ, and the state is corrupted by a zero-mean white process noise $\underline{w}$, with covariance Q. The observation equations are:

$$\underline{z}(k) = H(k)\underline{x}(k) + \underline{v}(k) \tag{5.1-4}$$

$$E[\underline{v}(k)] = 0 \tag{5.1-5}$$

$$E[\underline{v}(j)\underline{v}^T(k)] = R(k)\delta_{jk} \tag{5.1-6}$$

The observations $\underline{z}$ are linearly related to the state vector by the observation matrix H and are corrupted by zero-mean white noise with covariance R. In addition, the plant and observation errors are uncorrelated; i.e.,

$$E[\underline{v}(j)\underline{w}^T(k)] = 0 \tag{5.1-7}$$

The various assumptions, such as linearity and independent errors, can be (and have been) removed by investigators over the years but will be retained for purposes of simplicity and clarity in this treatment. Serial correlation of observation error will be considered later.

The optimal recursive estimation algorithm for this problem is well known as the Kalman Filter and was first published by Kalman [5-1,2] . The estimation error at time t(j), given observations through time t(k), is:

$$\epsilon(j|k) = \hat{\underline{x}}(j|k) - \underline{x}(j) \qquad (5.1\text{-}8)$$

where $\hat{\underline{x}}$ is the estimate of the true state $\underline{x}$. The state error covariance matrix is then defined as:

$$P(j|k) = E[\underline{\epsilon}(j|k)\underline{\epsilon}^T(j|k)] \qquad (5.1\text{-}9)$$

The Kalman Filter is then the linear, recursive minimum variance estimator for the above problem. It is, in fact, a set of rules for optimally combining the observations with a priori estimates of the state-given statistics of the relevant processes. The resulting algorithm - not derived here - is usually presented as a two-stage calculation.

Extrapolation Stage

State
$$\hat{\underline{x}}(k|k-1) = \Phi(k,k-1)\hat{\underline{x}}(k-1|k-1) \qquad (5.1\text{-}10)$$

Covariance
$$P(k|k-1) = \Phi(k,k-1)P(k-1|k-1)\Phi^T(k,k-1) + Q(k-1) \qquad (5.1\text{-}11)$$

Update Stage

Gain
$$G(k) = P(k|k-1)H^T(k)[H(k)P(k|k-1)H^T(k) + R(k)]^{-1} \qquad (5.1\text{-}12)$$

State
$$\hat{\underline{x}}(k|k) = \hat{\underline{x}}(k|k-1) + G(k)[z(k) - H(k)\underline{x}(k|k-1)] \qquad (5.1\text{-}13)$$

Covariance
$$P(k|k) = [I - G(k)H(k)]P(k|k-1) \qquad (5.1\text{-}14)$$

The implications and application of this algorithm are beyond the scope of this treatment, but the author highly recommends Gelb [5-3] as an excellent reference on the practical aspects of Kalman Filter design.

Example

A scalar example of such a filtering problem is the estimation of a first order Markov Process with exponential correlation, i.e.,

$$E[x(t)x(t+T)] = \sigma_x^2 \exp(-\omega T) \tag{5.1-15}$$

The state model is simply

$$x(k+1) = \gamma x(k) + w(k) \tag{5.1-16}$$

with observations

$$z(k) = x(k) + v(k) \tag{5.1-17}$$

where

$$\Delta t = t_{k+1} - tk \tag{5.1-18}$$

$$\gamma = \exp(-\omega \Delta t) \tag{5.1-19}$$

$$w \sim N(o,q) \tag{5.1-20}$$

$$v \sim N(o,r) \tag{5.1-21}$$

$$q = \sigma_x^2 (1-\gamma^2) \tag{5.1-22}$$

$$r = \sigma_v^2 \tag{5.1-23}$$

The corresponding Kalman Filter for this problem is then:

$$\hat{x}(k|k-1) = \gamma \hat{x}(k-1|k-1) \tag{5.1-24}$$

$$p(k|k-1) = \gamma^2 p(k-1|k-1) + q \tag{5.1-25}$$

$$g(k) = p(k|k-1) \Big/ [p(k|k-1) + \sigma_v^2(k)] \tag{5.1-26}$$

$$\hat{x}(k|k) = \hat{x}(k|k-1) + g(k)[z(k) - \hat{x}(k|k-1)] \quad (5.1\text{-}27)$$

$$p(k|k) = [1 - g(k)]p(k|k-1) \quad = g(k)\sigma_v^2(k) \quad (5.1\text{-}28)$$

Figure 5-1, a computer generated Gaussian white noise sequence, was utilized to drive equation (5.1-16) and thus simulate a typical Markov Process of this type using values of $\gamma^2 = 0{\cdot}9$ and $\sigma_x = 1{\cdot}0$. The same Gaussian random number generator was utilized to generate white observation errors with $\sigma_v = 0{\cdot}5$ resulting in the simulated observations of x - the z's. In Figure 5-2 these observations were introduced to the Kalman Filter. The resulting estimation errors, ε (after update), are plotted along with the associated error standard derivation σ_ε, calculated by the Kalman Filter. Notice the saw-tooth pattern of σ_ε caused by the time extrapolation which increases σ_ε followed by the update which decreases σ_ε because of the addition of measurement information.

Reformulate the basic recursive estimation problem into a multi-stage data compression and estimation problem. Suppose, as shown in Figure 5-3, that the filter is cycled once every $N\Delta t$ seconds but that it is desirable to process data at a rate N times the filter cycling rate. The integer N is often referred to as the compression ratio. Therefore at time t(k) there are N measurements, equally spaced Δt apart, that have been made since the last filter cycle at time t(k-N) which are to be processed at time t(k). This problem might be expected when the observation data are available at a rate higher than that rate which can computationally cycle the full filter or that rate which is necessary to recover the desired signal frequency. If the additional data is ignored, as is the case when using the conventional Kalman Filter since it accepts only a single observation, much useful information concerning the signal that would improve the accuracy of our estimation procedure is discarded.

The objective of optimal data compression techniques is to combine the N measurements in some manner into a single parameter (or set of

FIGURE 5-1

EXAMPLE: TRUE STATE AND OBSERVATIONS

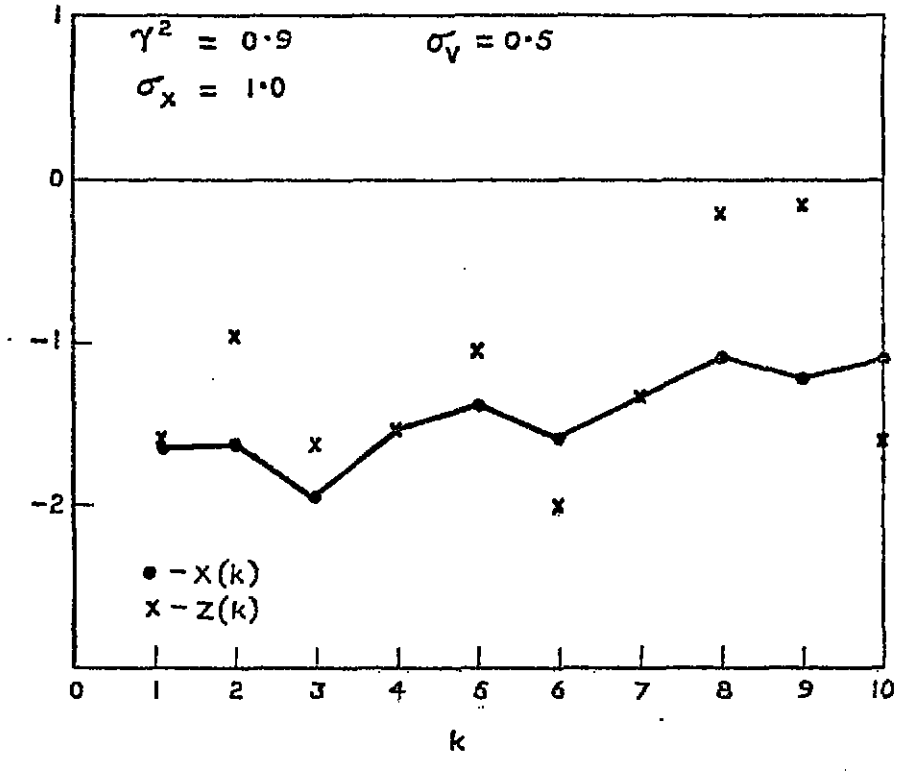


FIGURE 5-2

EXAMPLE: KALMAN FILTER ERROR STANDARD DEVIATION AND ERRORS

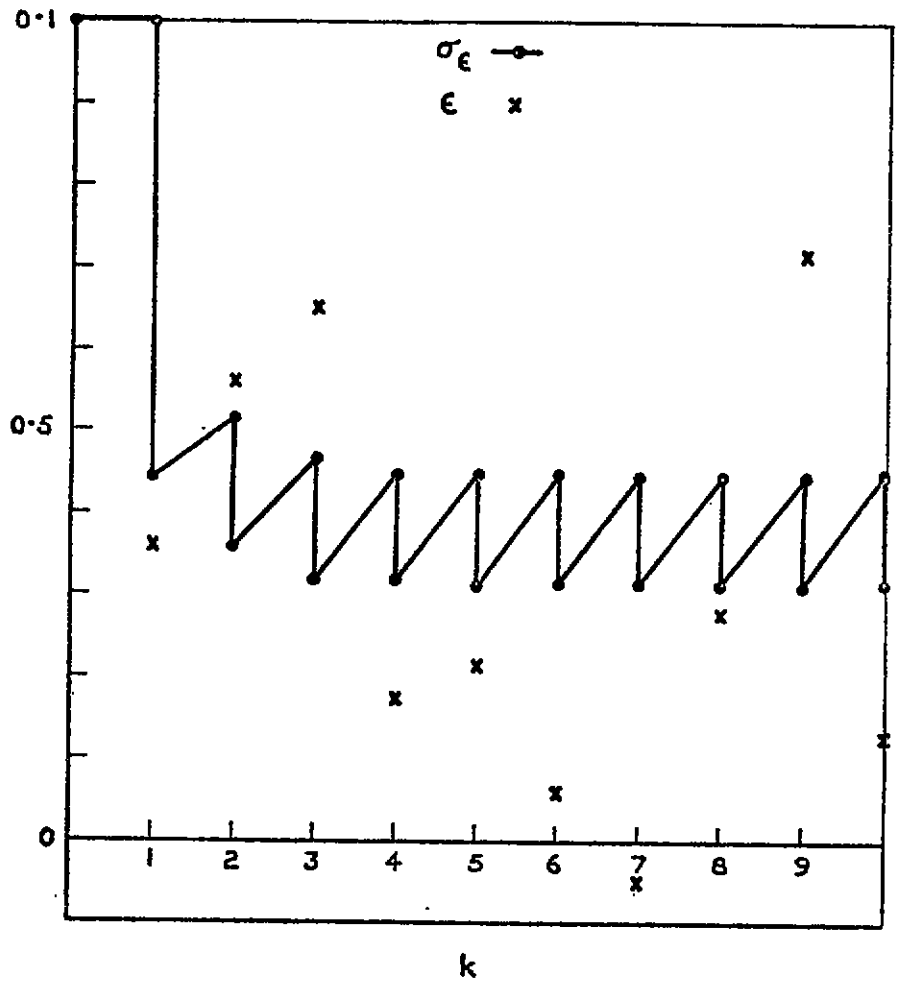
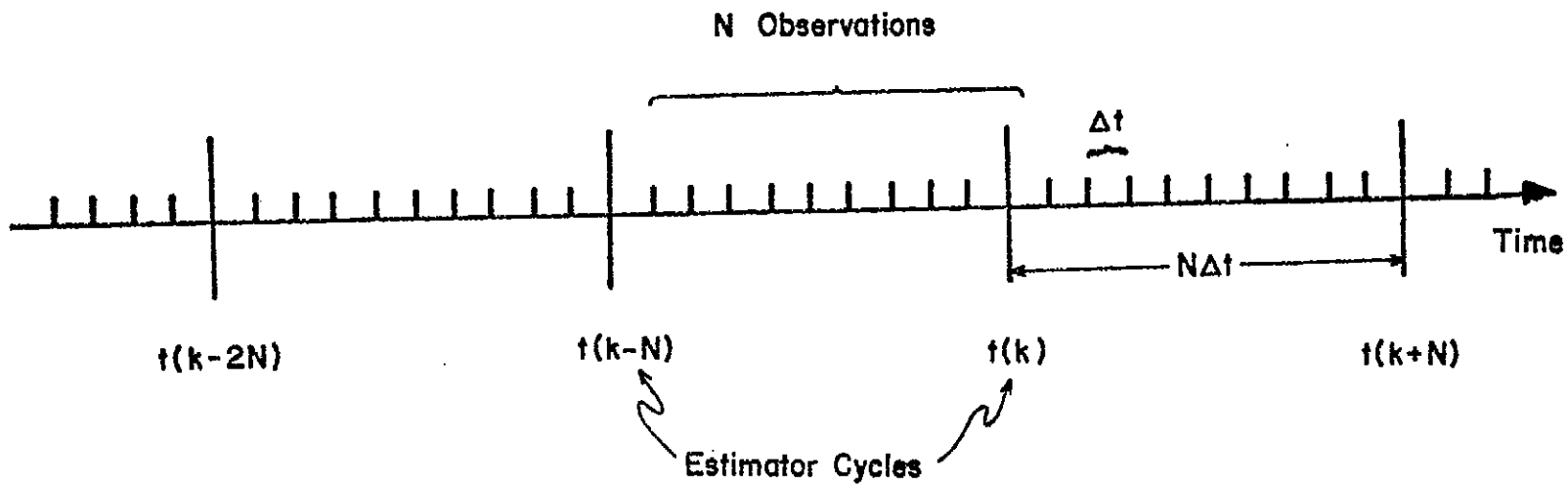


FIGURE 5-3

RELATION OF OBSERVATIONS AND ESTIMATOR CYCLES



parameters) in such a way as to minimize loss of accuracy while maintaining computational efficiency. The procedure which operates directly on the measurement is referred to as the "data compressor" or "prefilter" and that which operates more slowly on the compressed observation is referred to as simply the "filter" or "estimator." A Kalman Filter, such as described previously, operating directly on the measurements at the high-data rate and which contains all the correct model information and statistics will be "optimal." This filter represents the best available and thus is chosen as the standard for purposes of performance comparisons. The primary goal is to design a "suboptimal" data compression technique that degrades only slightly (or within acceptable limits) from the optimal. Besides the obvious advantage of computational efficiency, data compression can also be quite useful when dealing with multiple data rates and unevenly spaced data if an acceptable common estimation cycle time to which the data might be reflected (and compressed) can be determined.

Undoubtedly the best overall treatment of data compression and optimal estimation is that of Joglekar [5-4]. This work is comprehensive, covering optimal batch weighting as well as various averaging algorithms, covariance evaluation techniques and practical guidelines for design of multi-stage compression/estimation schemes. This work was conducted at the Stanford University Guidance and Control Laboratory and was sponsored by the Air Force Avionics Laboratory. Womble [5-5, 6] at Georgia Institute of Technology derived an optimal recursive prefiltering version of the Kalman Filter by determining a single discrete measurement that is equivalent to a set of discrete measurements.

Applications of various data compression techniques to estimation type problems are, of course, quite numerous and we will list only a select few here. Bar-Shalom [5-7] deals with the compression of data in real-time nonlinear estimation problems such as the linearized tracking filter for a re-entry vehicle. Clark [5-8] applied data compression techniques in the design of a real-time, dual-bandwidth, adaptive Kalman

tracking filter for high-speed maneuvering missiles and aircraft in a weapons control environment. Warren [5-9] derived a filter which provides optimal compensation for time lag and plant observation noise correlation. He applied the algorithm to position and velocity estimation for aircraft navigation. Kizner [5-10] utilized Chebyshev polynomial fits to derive an optimal data compression which he claims has better accuracy than the minimum variance estimate without data compression.

5.2 OPTIMAL DATA COMPRESSION TECHNIQUES

In a sense, the title of this paragraph might appear self-contradictory because, in application, data compression is never implemented optimally. If it is desirable to optimally process all the data, merely use the Kalman Filter. Optimal data compression is simply a restructuring of the Kalman Filter into the multi-stage problem of Figure 5-3. The restructuring is constrained such that the error covariance at the end of each multistage is equal to that of the optimal. The reason for doing this is to see the optimal data compressor and thus determine exactly what terms are neglected and test the validity of these simplifying assumptions.

Optimal data compression is a very important tool for designing such a system. Both Womble's optimal recursive prefilter and Joglekar's batch optimal compression algorithm will be presented, since, for any particular application and computer, one form may be preferable over the other. Both algorithms are optimal in the minimum variance sense and are exactly equivalent in covariance at the end of the compression intervals to the fast cycling conventional Kalman Filter.

The recursive prefiltering algorithm of Womble [5-5, 6] is presented in Table 5-1. It consists of a set of recursive matrix equations for the prefilter which must be cycled N times before the state and error covariance are updated by the estimator at the end of the interval. The prefilter can be cycled either as the measurements occur or delayed until the end of the interval and processed as a batch.

TABLE 5-1 OPTIMAL RECURSIVE DATA COMPRESSION ALGORITHM

Compression For $i = 1$, $N = k\text{-}N, k$

$$m(i) = H^T(i)R^{-1}(i)z(i) \tag{1}$$

$$J(i) = H^T(i)R^{-1}(i)H(i) \tag{2}$$

$$A'(i) = \Phi(i,i-1)A(i-1)\Phi^T(i,i-1) + Q(i) \tag{3a}$$

$$B(i) = I + J(i)A'(i) \tag{3b}$$

$$C(i) = I + A'(i)J(i) \tag{3c}$$

$$A(i) = A'(i)B^{-1}(i) \tag{3d}$$

$$\theta'(i) = \Phi(i,i-1)\theta(i-1) \tag{4a}$$

$$\theta(i) = [I - A(i)J(i)]\theta'(i) + A(i)m(i) \tag{4b}$$

$$\tilde{\Phi}(i) = C^{-1}(i)\Phi(i,i-1)\tilde{\Phi}\ i-1) \tag{5}$$

$$\tilde{J}(i) = \tilde{J}(i-1) + \tilde{\Phi}^T(i)J(i)C(i)\tilde{\Phi}(i) \tag{6}$$

$$\tilde{z}(i) = \tilde{z}(i-1) + \tilde{\Phi}^T(i)[m(i) - J(i)\theta'(i)] \tag{7}$$

Initialization:

$$A(o) = \tilde{J}(o) = o$$

$$\tilde{\Phi}(o) = I$$

$$\tilde{z}(o) = \theta(o) = o$$

Estimation

$$P'(k\text{-}N) = [I + P(k\text{-}N|k\text{-}N)\tilde{J}(N)]^{-1}\ P(k\text{-}N|k\text{-}N) \tag{8a}$$

$$\hat{x}'(k\text{-}N) = [I - P'(k\text{-}N)\tilde{J}(N)]\hat{x}(k\text{-}N|k\text{-}N) + P'(k\text{-}N)\tilde{z}(N) \tag{9a}$$

$$\hat{x}(k|k) = \tilde{\Phi}(N)\hat{x}'(k\text{-}N) + \theta(N) \tag{9b}$$

$$P(k|k) = \tilde{\Phi}(N)P'(k\text{-}N)\tilde{\Phi}^T(N) + A(N) \tag{8b}$$

Joglekar's [5-4] algorithm for optimal weighting of batch measurements is shown in Table 5-2. This algorithm is "cleaner" than the recursive algorithm in that the matrix algebra equations are not particularly more complicated than the original Kalman Filter. In fact, it is rather easy to see that the Kalman Filter for the trivial case of N=1 is recovered. This appearance of simplicity is misleading if the dimensions of the matrices used in the calculations are examined closely. The R* matrix, in particular, can get quite large - (MN x MN) where M is the dimension of the single observation. Unfortunately, it is necessary to invert this matrix.

In Tables 5-3 and 5-4, the recursive and batch optimal data compression algorithms were applied to the selected example problem presented previously. The substitution is rather straightforward. The resultant algorithms were applied with exactly the same set of parameters and observations used previously. The results, using a data compression ratio of N=5, are presented in Figures 5-4 and 5-5. Although each of the algorithms have different processing and covariance histories, it is important to emphasize that at the end of each compression interval; that is k=5 and 10, the error variances (or standard deviations) and actual estimates are identical to the original Kalman Filter presented in Figure 5-2. The optimal data compression algorithms are, in fact, merely the optimal Kalman Filter rearranged to account for the time delays and lumping, etc., occurring with the data compression approach. The principal difference in the error standard deviation histories of Figure 5-4 and 5-5 are caused merely by the order in which the extrapolation and update steps are taken. The recursive compressor reverses the more conventional order and updates before extrapolating.

Examination of either algorithm reveals a very significant problem that has not been discussed yet but which, in certain circumstances, can render data compression implementations either computationally impractical or seriously degraded in terms of performance. Since it is necessary to

TABLE 5-2 OPTIMAL BATCH DATA COMPRESSION ALGORITHM

Extrapolation

$$\hat{\underline{x}}(k|k-N) = \Phi(k,k-N)\hat{\underline{x}}(k-N|k-N) \tag{1}$$

$$P(k|k-N) = \Phi(k,k-N)P(k-N|k-N)\Phi^T(k,k-N) + Q^* \tag{2}$$

Update

$$K^* = [P(k|k-N)H^{*T} + T^*][H^*P(k|k-N)H^{*T} + R^* + H^*T^* + T^*H^{*T}]^{-1} \tag{3}$$

$$\hat{\underline{x}}(k|k) = \hat{\underline{x}}(k|k-N) + K^*[z^* - H^*\hat{\underline{x}}(k|k-N)] \tag{4}$$

$$P(k|k) = P(k|k-N) - K^*[H^*P(k|k-N) + T^*] \tag{5}$$

Batch Definitions

$$z^{*T} = [\underline{z}(1)\ z(2)\ \ldots\ \underline{z}(N)]^T \tag{6}$$

$$Q^* = E[w^*w^{*T}] \tag{7}$$

where

$$w^* = [\Phi(N,1)\ \ldots\ \Phi(N,N-1)\ \Phi(N,N)] \begin{bmatrix} \underline{w}(0) \\ \underline{w}(1) \\ \cdot \\ \cdot \\ \cdot \\ \underline{w}(N-1) \end{bmatrix}$$

$$= \sum_{j=1}^{j=N} \Phi(N,j)w(j-1) \tag{8}$$

$$H^* = [H(1)\Phi(1,N)\ \ H(2)\Phi(2,N)\ \ldots\ H(N)\Phi(N,N)]^T \tag{9}$$

$$R^* = E[v^*v^{*T}] \tag{10}$$

$$T^* = E[w^*v^{*T}] \tag{11}$$

Table 5-2 Optimal Batch Data Compression Algorithm (continued)

where

$$\underline{v}^* = [\underline{v}(1)\ \underline{v}(2)\ \ldots\ \underline{v}(N)]^T$$

$$= -\begin{bmatrix} H(1) & & & 0 \\ & H(2) & & \\ & & \ddots & \\ 0 & & & H(N) \end{bmatrix} \begin{bmatrix} 0 & \Phi(1,2) & \ldots & \Phi(1,N) \\ & 0 & & \\ & & \ddots & \\ 0 & & & 0 \end{bmatrix} \begin{bmatrix} \underline{w}(0) \\ \underline{w}(1) \\ \vdots \\ \underline{w}(N-1) \end{bmatrix} \qquad (12a)$$

$$\underline{v}_i^* = \underline{v}(i) - H(i) \sum_{j=i+1}^{j=N} \Phi(i,j)\underline{w}(j-1) \qquad (12b)$$

$$Q^* = \sum_{m=1}^{m=N} \Phi(N,m)Q(m-1)\Phi^T(N,m) \qquad (13)$$

$$R^*_{ij} = R(i)\,\delta_{ij} + H(i)\left[\sum_{k=\max(i,j)+1}^{k=N} \Phi(i,k)Q(k-1)\Phi^T(j,k)\right]H^T(j) \qquad (14)$$

TABLE 5-3 EXAMPLE: RECURSIVE COMPRESSOR

Compressor - For i = k-N to k

$$m(i) = z(i)/r \qquad (1)$$

$$J(i) = J = 1/r \qquad (2)$$

$$A'(i) = \Upsilon^2 A(i-1) + q \qquad (3a)$$

$$B(i) = 1 + A(i)/r \qquad (3b)$$

$$C(i) = B(i) \qquad (3c)$$

$$A(i) = A(i)/B(i) \qquad (3d)$$

$$\theta'(i) = \Upsilon\theta(i-1) \qquad (4a)$$

$$\theta(i) = [1-A(i)/r]\theta'(i) + A(i)m(i) \qquad (4b)$$

$$\tilde{\Phi}(i) = \Upsilon\tilde{\Phi}(i-1)/c(i) \qquad (5)$$

$$\tilde{J}(i) = J(i-1) + \Upsilon^2 c(i)/r \qquad (6)$$

$$\tilde{z}(i) = \tilde{z}(i-1) + \Upsilon[m(i) - \theta'(i)/r] \qquad (7)$$

$$A(o) = \tilde{J}(o) = \tilde{z}(o) = \theta(o) = 0 \quad \tilde{\Phi}(o) = 1$$

Update

$$p'(k-N) = p(k-N)/[1 + p(k-N)\,\tilde{J}(N)] \qquad (8a)$$

$$\hat{x}'(k-N) = [1 - p'(k-N)\tilde{J}(N)]\hat{x}(k-N) + p'(k-N)\tilde{z}(N) \qquad (9a)$$

Extrapolation

$$\hat{x}(k) = \tilde{\Phi}(N)\hat{x}'(k-N) + \theta(N) \qquad (9b)$$

$$p(k) = \tilde{\Phi}^2(N)p'(k-N) + A(N) \qquad (8b)$$

TABLE 5-4 EXAMPLE: BATCH COMPRESSOR

$$\phi(i,j) = \Upsilon^{i-j} \tag{1}$$

$$w^* = \sum_{i=1}^{i=N} \Upsilon^{i-N}\, w(i-1) \tag{2}$$

$$h_i^* = \Upsilon^{i-N} \qquad i = 1,N \tag{3}$$

$$v_i^* = v(i) - \sum_{j=i+1}^{j=N} \left(h_i^*/h_j^*\right) w(j-1) \qquad i = 1,N \tag{4}$$

$$Q^* = q \sum_{i=1}^{i=N} h_i^{*-a} \tag{5}$$

$$t_i^* = -q\, h_i^* \sum_{j=i+1}^{j=N} h_j^{*-a} \qquad i = 1,N \tag{6}$$

$$r^*_{ij} = r\delta_{ij} + q\, h_i^*\, h_j^* \sum_{k=\max(i,j)+1}^{k=N} h_k^{*-a} \qquad i,j = 1,N \tag{7}$$

$$z_i^* = z(i) \tag{8}$$

Extrapolation

$$\mathfrak{x}'(k) = \Upsilon^N\, \mathfrak{x}\,(k-N) \tag{9}$$

$$p'(k) = \Upsilon^{aN}\, p(k-N) + Q^* \tag{10}$$

Table 5-4 Example: Batch Compressor (continued)

Gain

$$d_{ij} = h_i^* h_j^* p'(k) + r_{ij}^* + h_i^* t_j^* + t_i^* h_j^* \qquad i,j = 1,N \qquad (11)$$

$$E = D^{-1} \qquad (12)$$

$$K_i^* = \sum_{j=1}^{j=N} \left[h_j^* p'(k) + t_j^* \right] e_{ji} \qquad i = 1,N \qquad (13)$$

Update

$$\hat{x}(k) = \hat{x}'(k) + \sum_{i=1}^{i=N} K_i^* \left[z_i^* - h_i^* \hat{x}'(k) \right] \qquad (14)$$

$$p(k) = p'(k) - \sum_{i=1}^{i=N} K_i^* \left[h_i^* p'(k) + t_i^* \right] \qquad (15)$$

FIGURE 5-4

EXAMPLE: RECURSIVE COMPRESSOR ERROR STANDARD DEVIATION AND ERROR

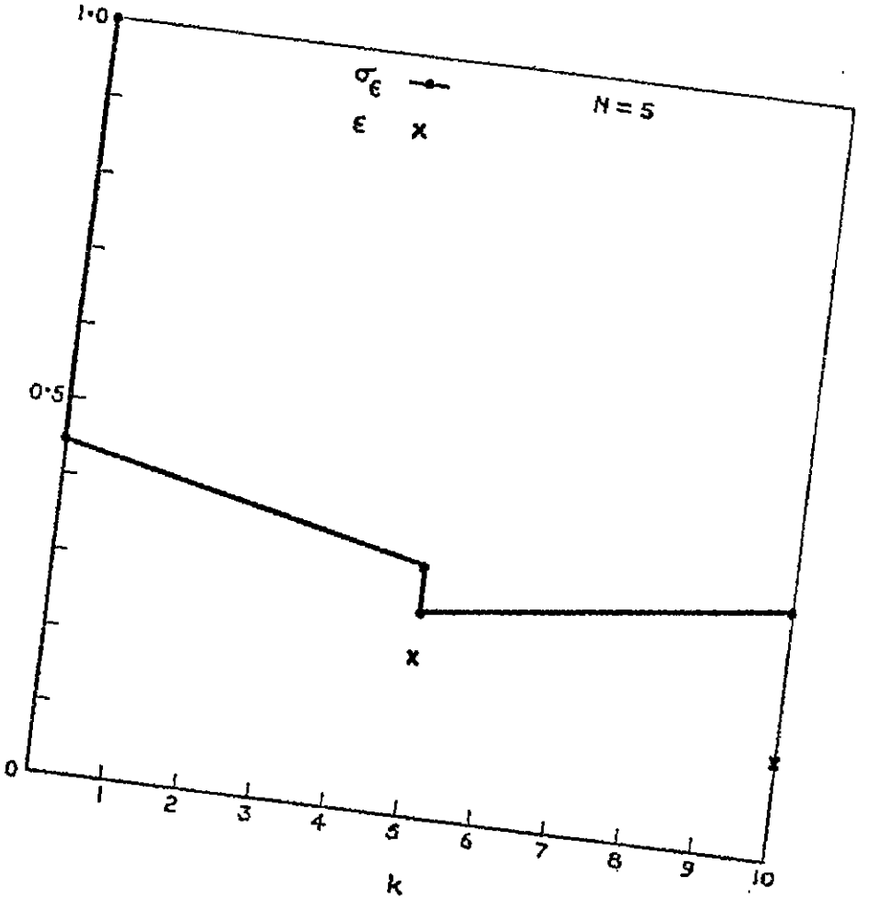
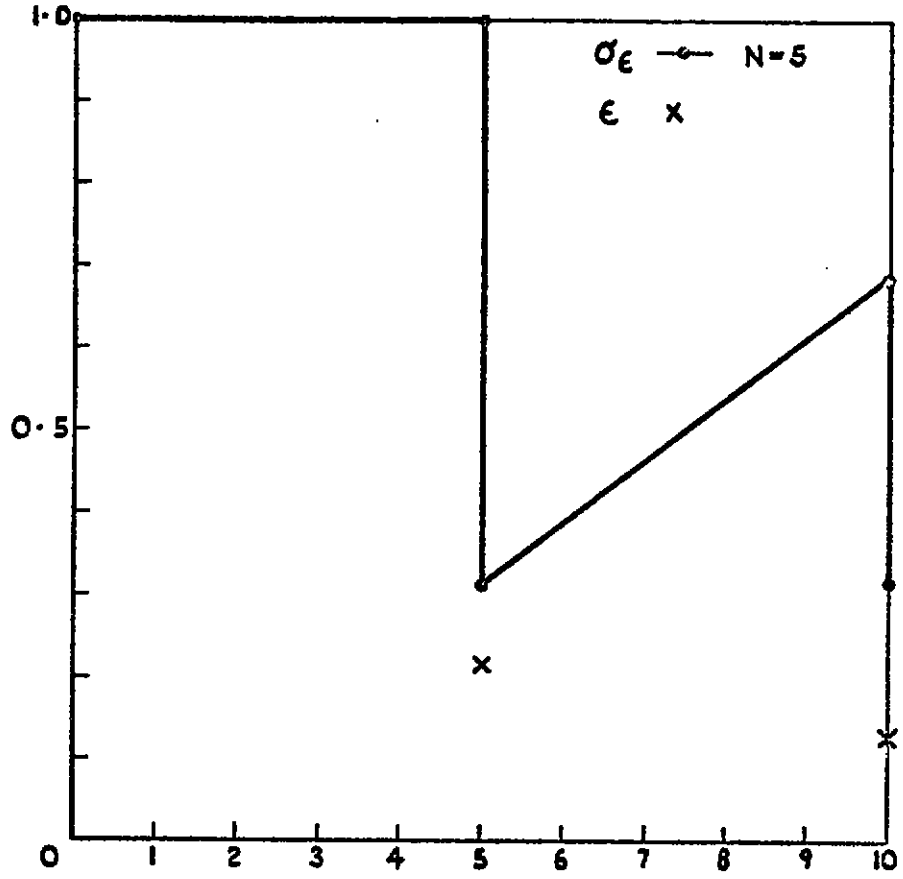


FIGURE 5-5
EXAMPLE: BATCH COMPRESSOR ERROR
STANDARD DEVIATION AND ERROR



refer all observations to a common time and this movement in time obeys the state dynamics equation, additional error is introduced due to the presence of plant process noise. In fact, the compressed observation error becomes correlated with the plant noise even though the original problem contained no such correlation. We see this correlation, for example, in the expression for v*, equation (12) in Table 5-2, and in the resulting equations in both algorithms. Consideration of this correlation was taken in the derivation of both optimal compression schemes. The presence of this effect results in the major addition of complexity to the data compression algorithms over that of the original Kalman Filter. The condition for neglecting this effect and the tremendous simpliciation that results is presented in paragraph 5.3.

There is another rather obvious approach to the optimal data compression problem that is somewhat simpler and should not be overlooked. This approach is to simply let the data compressor be a Kalman Filter with i=1 the first point and i=N the last. If the output state is evaluated at a time other than i=N, simply utilize optimal Kalman smoothing such as discussed by Gelb [5-3]. The output state vector from the compressor then becomes the input compressed observation for the slow Kalman Filter. If conditions are such (no process noise) to insure that the compressor estimate is totally independent of the slow filter and thus represents a new uncorrelated observation, the resulting combination of Kalman Filters will be equivalent to a single fast filter and will thus be optimal. This approach is favoured under these circumstances since it lends itself to analysis, implementation and evaluation easier than the other two approaches. If process noise is present, a conventional Kalman Filter which accounts for observation and plan correlation can be utilized. An example of such an algorithm can be found in Sage and Melsa [5-11].

5.3 SUBOPTIMAL DATA COMPRESSION

This paragraph will show how practical suboptimal data compression

algorithms can be derived from the optimal algorithms contained in the previous paragraphs. Essentially the arguments of Womble [5-5] will be reproduced and the problem progressively simplified by adding particular constraints to the original problem definition.

5.3.1 NEGLIGIBLE PROCESS NOISE

The greatest simplification that can be made to the data compression problem occurs when there is no process noise; i.e., Q=0. If the recursive algorithm of Table 5-1 is considered first, it is found that, using the initial conditions and letting $Q = 0$; $\tilde{A} = 0$ and $\Theta = 0$. As Womble points out, the prefilter transition matrix becomes the usual value

$$\tilde{\Phi}(i) = \Phi(i, 1) \tag{5.3-1}$$

and the prefilter equations reduce to

$$\tilde{J}(N) = \sum_{i=1}^{i=N} \Phi^{T}(i,1)J(i)\Phi(i,1) \tag{5.3-2}$$

and

$$\tilde{z}(N) = \sum_{i=1}^{i=N} \Phi^{T}(i,1)m(i) \tag{5.3-3}$$

This approach essentially results in the compressed observation being a rather simple weighted average which shows considerable computational advantage over the original Kalman Filter. Similarly, the batch algorithm simplifies since, for $Q = 0$, $Q^* = 0$, $T^* = 0$ and R^* reduces to a much simpler matrix involving only the original R matrices. It begins to become obvious that, in fact, the two algorithms actually end up processing the observations identically as stated previously. Joglekar points out that, rather than having Q vanish, Q should be negligible relative to the observation error; i.e.,

$$||H(i)\Phi(i,i+1)Q(i)\Phi^T(i,i+1)H^T(i)|| \ll ||R(i)|| \qquad (5.3\text{-}4)$$

where the double brackets denote the matrix norm.

Example: For the simple Markov example, equation (5.3-4) reduces to

$$\gamma^2\sigma_x^2(1-\gamma^2) \ll \sigma_v^2 \qquad (5.3\text{-}5)$$

Therefore, it is reasonable to expect to invoke this assumption if $\sigma_x^2 \ll \sigma_v^2$ when the process noise shows little variance relative to the observation noise. (The process begins to look like a constant zero.) Also for the limiting cases $\gamma \to 1$ and $\gamma \to 0$, the process looks like a constant bias or simply white noise like the observations. Of course, this last case makes the entire attempt of estimation ridiculous. Numerically, this particular example corresponds to $(0{\cdot}09) \ll 1$.

5.3.2 NEGLIGIBLE SYSTEM DYNAMICS

If over the compression interval, the system dynamics appear to be $\Phi \cong I$, additional simplification of the algorithms result. This condition means that it looks like the measurements occur at the same time or that no "reasonably accurate" dynamics can be resolved

over the interval due to observation error. In the recursive algorithm we find

$$\tilde{J}(N) = \sum_{i=1}^{i=N} J(i) \tag{5.3-6}$$

$$\tilde{z}(N) = \sum_{i=1}^{i=N} m(i) \tag{5.3-7}$$

In the batch algorithm, the H* matrix simplifies considerably.

5.3.3 STATIONARY OBSERVATION STATISTICS

Also, if over the compression interval, the observation statistics do not change; i.e., R(i) = R for all i, the recursive algorithm looks like the following:

$$\tilde{J}(N) = H^T \left[\frac{1}{N} R^{-1}\right] H \tag{5.3-8}$$

$$\tilde{z}(N) = H^T \left[\frac{1}{N} R\right]^{-1} \left(\frac{1}{N}\right) \sum_{l=1}^{l=N} m(i) \tag{5.3-9}$$

Joglekar [5-4] was motivated by the appearance of the weighted average measurement compression to construct "exact averaging" algorithms which are designed to give the best estimate of the state given that only equally weighted averaged measurements are available. He obtained an expression for the information loss due to "exact averaging." His exact averaging algorithms included consideration of process noise and associated correlations. The development is quite lengthy and will not be repeated here.

5.4 SENSITIVITY ANALYSIS

A particularly significant advantage for developing optimal data compression algorithms is that they provide a performance standard for comparison and evaluation of suboptimal realizeable approaches. As shown in paragraph 5.3, it is possible to determine exactly those terms that were chosen to be neglected and check the validity of the assumption. Unfortunately, when suboptimal, the associated error covariance calculations are incorrect since they are based upon simplifying assumptions. Therefore, the calculated suboptimal error covariance can no longer be used as a true measure of estimation performance. Fortunately, however, optimal estimation theory comes to the rescue by providing a means to calculate the actual error covariance of a suboptimal implementation and thus compare it with the optimal to determine the level of performance degradation. Again, Gelb's book [5-3] provides an excellent discussion of suboptimal filter design and sensitivity analysis.

In order to calculate the actual covariance of a suboptimal design, it is necessary to build a sensitivity algorithm tailored to fit the original compression approach. Therefore each of the three optimal algorithms in paragraph 5.2 must have their own associated sensitivity algorithm. In his report, Joglekar [5-4] derives equations for the actual covariance when using the averaging type compression algorithms he derived. The author does not, however, provide a sensitivity algorithm for the general batch compressor. Womble [5-5, 6] also fails to provide

a sensitivity algorithm for the prefilter. Sensitivity is therefore clearly an area of data compression requiring additional work if designers are to have a complete set of tools with which to develop practical data compression algorithms.

5.5 GUIDELINES FOR OPTIMAL DATA COMPRESSION DESIGN

5.5.1 ESTIMATION RATE AND SHANNON'S THEOREM

The first question to be considered involves how frequently to estimate the state of the system to specify accurately the state at all times. Shannon's Theorem - found in Monroe [5-12] - says that if a signal is bandlimited and contains no frequency greater than ω_{signal} (radians/ second) then it is possible, in principle, to recover completely the original signal from the sampled signal if sampled at a minimum rate of

$$\Omega_s = \omega_{signal} \Big/ \pi \text{ per second} \qquad (5.5\text{-}1)$$

This is to say in theory, no information is lost if the signal is perfectly sampled at that rate or faster. Since it is desirable to reconstruct the signal as accurately as possible and with a minimum loss of information, cycle the signal estimator (or slow filter) no slower than Ω_s. In fact, since there are no perfect samples or perfect estimators, estimate even faster than Ω_s - perhaps by a factor of two to ten. Another problem is that real-world signals are not often truly bandlimited but often only an accurate estimate of the lower frequency components is of interest. Shannon can still be used as a guideline to select the estimation rate but consideration must also be given to the affects of the higher frequency.

5.5.2 SAMPLING RATE AND THE NYQUIST FREQUENCY

The Nyquist Frequency or folding frequency is defined by Bendat and Piersol [5-13] as

$$\omega_N = \pi \Omega_{sampling} \text{(radians/second)} \qquad (5.5\text{-}2)$$

The similarity to equation (5.5-1) is undoubtedly not coincidental. If there is any frequency component in the signal - be it due to observation error or plant noise - there will be confusion between the higher frequency components and the lower frequency components that are presumably of interest. This problem is well known as aliasing or the "folding" of high frequency components into the low frequency. This is inherent in all analog-to-digital sampling systems. There are two practical methods of handling the aliasing problem. The first is to simply raise ω_N by raising the sampling frequency until there is no frequency component above ω_N. This technique is not always practical however. The second and more efficient method is to simply analog filter the data prior to sampling or digitally prefilter the data by simply averaging it in batches as in data compression. The analog and digital prefilters are in fact complimentary; the analog being preferred to remove very high frequency noise (relative to the signal) and the digital to remove noise which is not so high compared to the signal. Joglekar [5-4] discusses this in greater detail in his paper.

5.5.3 SERIALLY CORRELATED OBSERVATION ERROR

If the observation error has a bandlimited serial correlation, either naturally or due to the prefiltering, the effects on the information content of the observations as a function of the data rate and correlation should be considered. As an example, follow the arguments of Clark [5-8] and consider exponentially correlated observation error where the correlation coefficient of the original data is given as

$$\rho(T) = E[v(t)v(t+T)] \Big/ E[v^2(t)] \qquad (5.5\text{-}3)$$
$$= \exp\left(-|T| \Big/ \tau_v\right)$$

where τ_v is the correlation time constant. The discrete noise propaga-

tion equation for this stationary case is then

$$v(k) = \rho(\Delta t)v(k-1) + \sigma_{\xi}(\Delta t)\ \xi(k-1) \tag{5.5-4}$$

which is a simple linear system driven by white noise of variance σ_{ξ}^2 related to the ouput variance σ_v^2 by the relation

$$\sigma_{\xi}(\Delta t) = \sigma_v \sqrt{1 - \rho^2(\Delta t)} \tag{5.5-5}$$

Now assume that N measurements are again to be compressed utilizing an averaging technique to yield a single compressed observation. The variance and correlation time of the compressed measurement as a function of the original statistics and the compression ratio should now be determined. The compressed observation error v_c is given simply as the average

$$v_c(k) = \frac{1}{N} \sum_{i=1}^{i=N} v(k-N+i) \tag{5.5-6}$$

By substituting this into the appropriate definitions and taking expected values, it is easy to show.

$$\sigma_c / \sigma_v = \sqrt{S_1} \tag{5.5-7}$$

where

$$S_1 = \frac{1}{N^2} \sum_{i=1}^{i=N} \sum_{j=1}^{j=N} \rho^{|i-j|} \tag{5.5-8}$$

which can be simplified to

$$S_1 = \frac{1}{N} + \frac{2}{N^2} \sum_{i=1}^{i=N} (N-i)\ \rho^i \qquad (5.5\text{-}9)$$

The effective correlation time of the compressed observation τ_c is related to the original τ_v by

$$\frac{\tau_c}{\tau_v} = \frac{N \ln (1/\rho)}{\ln (1/\rho_c)} \qquad (5.5\text{-}10)$$

where

$$\rho_c = S_2 \Big/ S_1 \qquad (5.5\text{-}11)$$

and

$$S_2 = \frac{1}{N^2} \sum_{i=1}^{i=N} \sum_{j=1}^{j=N} \rho^{|i-j+N|} \qquad (5.5\text{-}12)$$

or

$$S_2 = \frac{\rho^N}{N} \left[1 + \frac{1}{N} \sum_{i=1}^{i=N} (N-i)(\rho^i + \rho^{-i}) \right] \qquad (5.5\text{-}13)$$

In Figures 5-6 and 5-7 the ratios are plotted as a function of the compression ratio for various levels of relative correlation. Large values of $\Delta t/\tau_v$ imply less correlation than small values. In Figure 5-6 we find (as we might expect) that for essentially uncorrelated error ($\Delta t/\tau_v = 10$) the error reduction behaves ideally as $1/\sqrt{N}$. As the correlation increases, the less independent information is received and improvement diminishes.

FIGURE 5-6

DATA COMPRESSION: ERROR REDUCTION WITH SERIALLY CORRELATED OBSERVATIONS

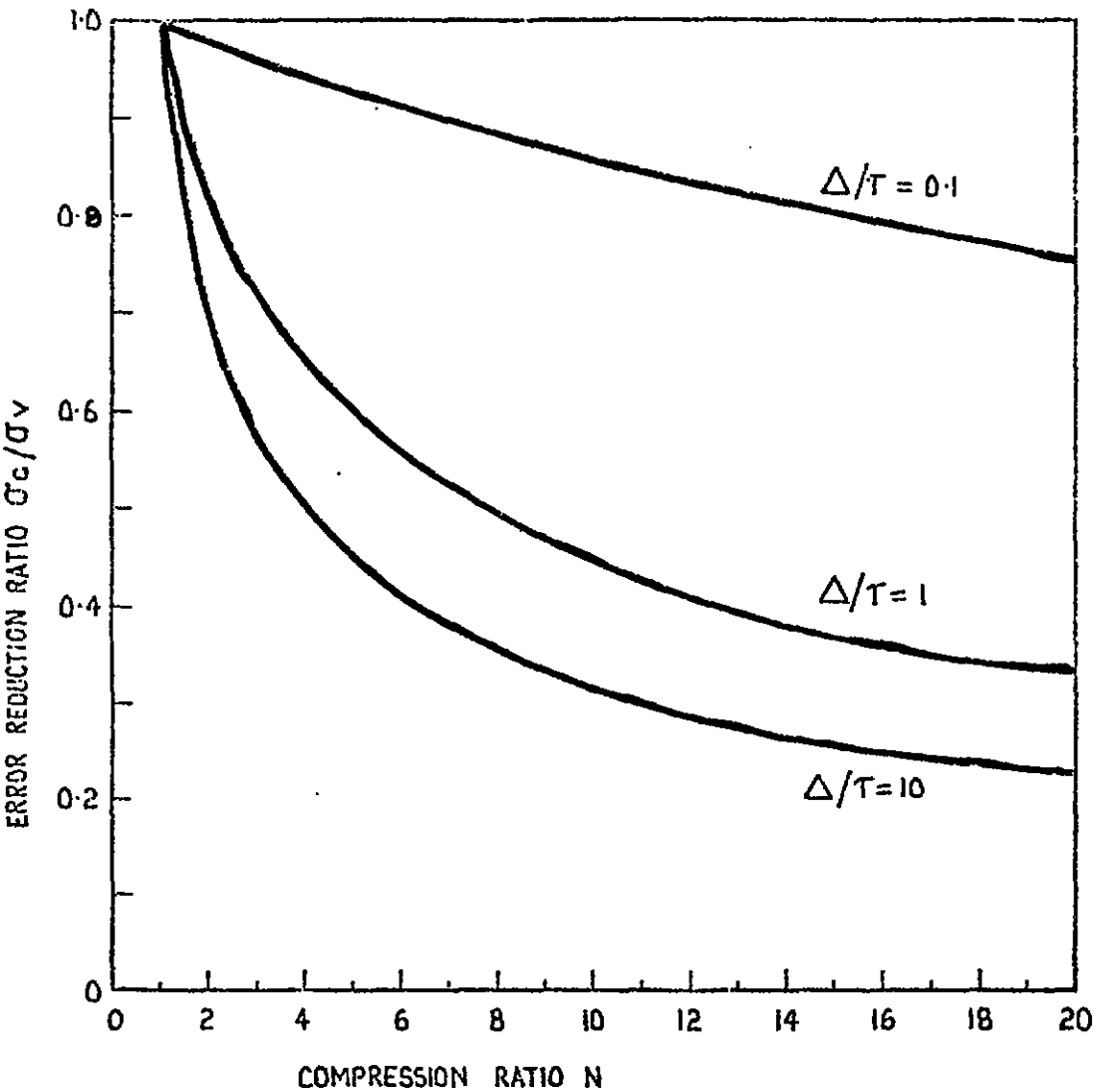
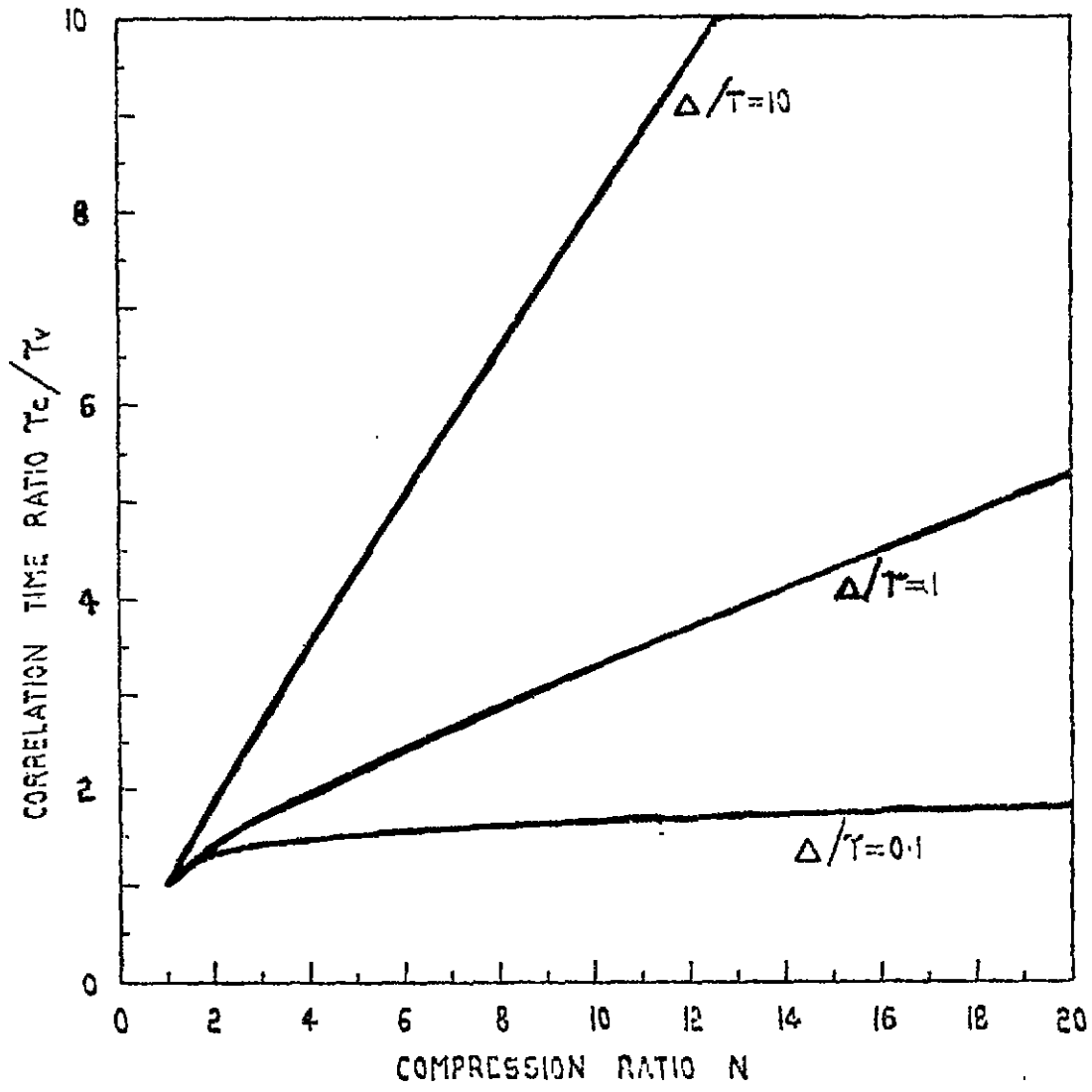


FIGURE 5-7

DATA COMPRESSION: COMPRESSED CORRELATION TIME RATIO WITH SERIALLY CORRELATED OBSERVATIONS



In fact, for high correlation ($\Delta t/\tau_v$ 0·1), little improvement is observed even after 20 samples. Figure 5-7, shows that for conditions of high correlation, the data compression process does not significantly increase the basic correlation time. However, a dramatic increase of correlation time is realized by compressing observations that originally contained little correlation. Joglekar [5-4] recommends a sampling rate such that

$$0{\cdot}25 \leq \Delta t/\tau_v \leq 1{\cdot}0$$

in order to efficiently recover most of the information.

REFERENCES

[5-1] Kalman, R.E., "A New Approach to Linear Filtering and Prediction Problems," Transactions of the ASME, Journal of Basic Engineering 82D, 35-45 (1960)

[5-2] Kalman, R.E., "New Results in Linear Filtering and Prediction Theory," Transactions of the ASME, Journal of Basic Engineering 83D, 95-108 (1961)

[5-3] Gelb, A. (ed), Applied Optimal Estimation (Cambridge, Mass: MIT Press, 1974)

[5-4] Joglekar A.N., "Data Compression in Recursive Estimation with Applications to Navigation Systems," SUDAAR No. 458 (Stanford University Dept. of Aeronautics and Astronautics, 1973)

[5-5] Womble, M.E. and Potter J.E., "A Preflitering Version of the Kalman Filter with New Numerical Integration Formulas for Riccati Equations," Proceedings of the 1973 IEEE Conference on Decision and Control, (San Diego, California, 1973)

[5-6] Womble, M.E., "Data Compression Via Prefilters." (1974)

[5-7] Bar-Shalom, Y. "Application of Data Compression to Real-Time Estimation," Presented at the IEEE Decision and Control Conference, Miami Beach, Florida (1971)

[5-8] Clark, B.L., "Development of an Adaptive Digital Target Tracking Filter and Predictor for Fire Control Applications," NSWC Technical Report TR-3445 (1976)

[5-9] Warren, A.W., A Continuous-Discrete Data Filter for Pre-Filtered Observations (Seattle, Washington: Boeing Computer Services, Inc., 1973)

[5-10] Kizner, W., "The Enhancement of Data by Data Compression Using Polynomial Fitting," JPL Technical Report 32-1078 (1967)

[5-11] Sage, A.P. and Melsa J.L., Estimation Theory with Applications to Communications and Control, (New York: McGraw-Hill, 1971)

[5-12] Monroe, A.J., Digital Processes for Sampled Data Systems, (New York: John Wiley and Sons, 1962)

[5-13] Bendat, J.S. and Piersol A.G., Random Data:Analysis and Measurement Procedures,:(New York: Wiley-Interscience, 1971)

CHAPTER 6

MAXIMIZATION OF INFORMATION CONTENT

6.1 INTRODUCTION

Sometimes it is necessary to restore, in total, exceedingly large amounts of data that have been collected. This is especially true of projects where data is collected by one responsible agency, stored and retrieved by another agency, and used by several different agencies for different purposes.

For example, live aircraft test data collected under varing environments may be desired by agencies interested in missile simulations, others interested in aircraft performance, still others interested in instrumentation accuracies, etc. Often the storage of such data is referred to as a "Data Base" or a "Data Bank." The designer of such a system encounters problems that do not normally arise when smaller amounts of data are involved.

It is not unusual for such data bases to contain several million to a billion or more words of data. The cost involved in the storage and retrieval of such data can be prohibitive if careful planning is not made in the design phases.

The purpose of this chapter is to suggest practical ways by which the sheer volume of the data can be reduced if tradeoffs in accuracy and retrieval costs can be accepted. Hopefully, this will give the designer a starting point when faced with a large volume of data to be stored and retrieved. Additionally, suggested ways for presenting the large amounts of data to the user will be discussed with a few general purpose graphic routines presented in paragraph 6.4.

6.2 VOLUME REDUCTION

For the purpose of discusion, consider the following example.

A particular project expects to fly 500 air - air combat training missions. It is desired to retain from each aircraft, in time-history form, the following parameters for future investigation.

Description	No. Parameters
Time	1
Position	3
Velocity	3
Acceleration	3
Attitude	3
Angle of Attitude/ Angle of Side Slip	2
Aiming Parameters	3
Aspect Parameters	3
Target ID	1
Power Setting	1
Fire Signal	1
Relative Winds	3
Total	27

If four aircraft participate for an average of 30 minutes per mission and the collection scheme is 10 samples/sec, the total number of data words collected would be $27 \times 4 \times 10 \times 60 \times 30 \times 500 \approx 972 \times 10^6$ words. With present storage devices, the cost of storage and retrieval would be prohibitive unless the volume could be reduced.

A first step in approaching the problem should be to investigate other methods discussed in previous chapters of this document for reducing the number of words that must be stored. For example, it may not be

necessary to retain 10 samples/sec for every parameter. Using the sampling techniques discussed in Chapter 2, alternate sampling rates can be derived which may reduce the total number of words by a factor of two or more.

Parameters such as fire signals, power settings and target identification change relatively few times during a given mission. These can be retained on a separate file, recording only the change and time of change.

6.2.1 RECOMPUTING

Investigation should be made into the need for retaining every parameter. Could some parameters be computed from others at retrieval time with acceptable computer costs?

In the example given, velocity and acceleration can be computed from position. Aiming and aspect angles can be derived from position and attitude. Inertial angle of attack/angle of side slip can be computed from relative winds, velocity, and attitude data. Relative winds can be derived from wind tables stored in a different file. Assuming that target ID, power setting, fire signals, and wind tables are stored on separate files (the magnitude of these files would be relatively small in comparison) and the parameters mentioned above can be recomputed, the number of words/sample becomes 6 instead of 27. The reduction factor $\approx 4.8:1$.

6.2.2 SCALING AND PACKING

Scaling a parameter simply means determining the absolute resolution that must be maintained when the data is retrieved. It is important because the resolution determines the minimum number of bits necessary to retain the parameter.

Assume, in the example given, that a stored position resolution of 1 foot with angular resolution of .1 degree is sufficient to retain the necessary accuracy when the data is reproduced. If the missions are to be flown in an airspace with a diameter of 50 miles, then the dynamic range of a position word is ±264000 ft. The number of bits necessary to represent a positional parameter to 1 foot resolution is 20 bits. The compression ratio for a CDC 6600 computer word is 3:1. For the 32-bit word machines the ratio is only 1.6:1.

Additional compression may be realized by making use of the fact that the dynamic range of the first difference in position is usually within $\pm 2000 \times \Delta t$ where Δt is the sampling interval in seconds. If $\Delta t = .2$, the first difference lies between ±400 which can be retained in 10 bits. If only the magnitude of the first differences were retained, 9 bits would suffice.

In order to retain 1 foot resolution, it is necessary to periodically record the full position word with intermediate updating of position from the first difference. If the first differences are retained to .1 ft resolution with rounding, the maximum error contributed by a single sample is .05 ft. Assuming that uniform distribution of error is between 0 ~ .05, the average error contributed would be .025 ft/sample. If the retained sample rate is 5/sec and the full position is recorded every 4 seconds, average cumulative error would be approximately .5 ft. (the 20th periodic samples would be the updated position). The additional compression realized by this scheme would be $(20X20:20+20X10) = 400:220 \approx 2:1$.

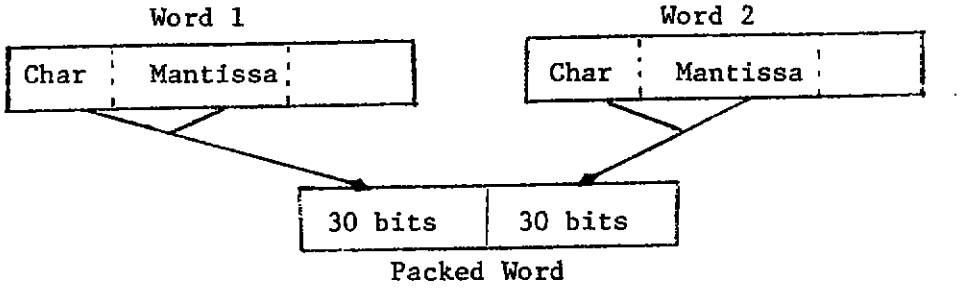
6.2.2.1 EXPONENTIAL PACKING

Sometimes, as in the case of radiometric data, the dynamic range of a variable is extremely large. Additionally the rate of change can be of such magnitude as to preclude using the first-difference technique described previously. Usually in such cases, absolute accuracy is not required. Instead a given number of significant digits of accuracy would be sufficient. Using this criteria, an exponen-

tial packing scheme could be devised such that the exponent of a variable could be retained in a few bits with the normalized (leading zeroes suppressed) variables presented to the desired number of significant digits.

This scheme can be very useful if the word size on the computer is relatively large and the computer contains floating point arithmetic. Consider the CDC 6600 computer word for example. The characteristic and sign are in 13 bits whereas 47 bits are used to represent the mantissa.

Making use of the CDC normalized floating arithmetic with shift and mask instructions, two words with six significant digits of accuracy may be packed into a single word. The compression ratio is 2:1.



An advantage of this scheme is that the data is already in acceptable floating point representation and does not need a separate table to retain scale factors.

6.2.2.2 FIXED-LENGTH MINIMUM BIT

A simple example of a fixed-length minimum bit scheme would be the use of a single bit to represent fire signal; zero = no fire, one = fire. For a CDC 6600 computer word the savings is 60:1.

Generally, however, the data cannot be represented by a single bit but in many cases there is a minimum number of bits

which can be used to represent the full dynamic range of data. (If the dynamic range is relatively large, an alternate scheme such as exponential packing, variable length minimum or a table of external scale factors would be desirable.)

Consider attitude, for example, with a full dynamic range 0-360 degrees. If .1 degree resolution with .05 degree accuracy is sufficient the dynamic range would be 0-3600 with a scale factor of 10 which could be retained in 12 bits. The storage savings would be 5:1 for a CDC 6600 computer word. The technique to pack words is simply to multiply the original word by 10, round, integerize and pack using shift and mask instruction. To unpack, simply mask, shift, and divide by 10.

The technique does not make full use of the storage capability of 12 bits. If the scaling factor were change to 4095/360, an accuracy of .0434 instead of .05 could be realized.

An alternate version of a fixed-length, minimum-bit scheme would be to retain a table of scale factors with sufficient additional bits allocated to each word for pointing to the correct entry in the table. This scheme allows for a broad dynamic range of a given variable.

6.2.2.3 VARIABLE-LENGTH MINIMUM BIT

An alternate form of the minimum bit scheme is to use a variable number of bits to represent a parameter with a broad dynamic range. A truly variable scheme would require an external table with entries pointing to the number of bits used to represent a parameter at a given time. There is the additional need for a pointer to point to the correct entry that must be retained with each sample.

A modified version of the variable bit scheme would be to divide the dynamic range into bands with a given number of bits allocated for each band. A pointer is retained with each word that would point to the correct band with an inherent number of bits.

Consider a variable with a dynamic range of 0-50000 with unity accuracy and resolution requirements. The data fell between 0-200 ninety percent of the time and was greater than 200 only ten percent of the time. If two bands were allocated containing 8 bits and 16 bits respectively with a single bit to point to the correct band, the savings over a fixed length minimum bit scheme would be

$$16:(.9X(8)+.1x16+1) = 16:9.8$$

For a CDC 6600 computer word, the compression ratio is

$$60:9.8 \qquad 6:1$$

Additional computer cost is involved to obtain the correct number of bits for shifting.

6.3 PRESENTATION

One of the most important and sometimes least emphasized areas of data reduction is data presentation. Often a simple change of an output format can mean saving many manhours in data analysis. Appropriate selection of numerical and graphical presentations can sometimes mean the difference between an accurate analysis or one that is biased by the analyst simply because he was not able to observe unexpected relationships or detect system errors.

6.3.1 NUMERICAL PRESENTATION

Numerical presentation implies presenting the data in a numerical format whether it be a simple printout of data or more sophisticated schemes of using numerics (or symbols) to represent various conditions or levels. Examples of such are digital pictures or number graphs.

6.3.1.1 THE TIME-HISTORY LIST

When presenting data in a time sequential format, a column presentation is usually preferred. Every effort should be made to output only a single parameter for a given field. This allows the user to scan a column and observe trends without having to search for the parameters in a maze of printouts.

Often the number of available print columns for a given listing is not sufficient for presenting all the desired parameters. In these instances, additional listings should be generated, usually with time on each listing for easy correlation. The simplest method involves generating the additional listings on separate files with disposition to a printing device.

If additional files are not available due to program limitations, the data can be written to a single file with appropriate code numbers to indicate separate listing. Before printing, the file can be sorted and printed by code number.

6.3.1.2 REDUCING PRINTOUT

As a general rule, a printout of every sample is neither desired nor needed. Selected samples that show significant levels, changes or samples at significant events are favored.

When the requirement is for data only during and after significant changes, the programming is easy to implement. When data prior to significant events is desired, the implementation is not as easy. If sufficient core storage is available, a rotary buffer could be maintained with sufficient past history retained to print the required data prior to events. The modular function available on most compilers is an excellent tool for retaining the current address in the rotary buffer. (Similar rotary schemes are often used when doing mid-point smoothing and editing of data.)

6.3.1.3 FIELD REDUCTION

Often the number of columns needed to represent a given parameter can be reduced by scaling techniques discussed previously. Other methods include printing the data in integer format with implied decimal. If a parameter, such as time, has columns that change infrequently (e.g., hours, minutes), these can be written at the top of each listing with less print columns assigned to the parameter itself. In any case, the columns assigned to a given parameter should remain constant to avoid confusion. It is not unusual for thirty or more parameters to be listed on a single page in column format with proper scaling and techniques.

6.3.1.4 MATRIX PRESENTATION

When data is of a matrix nature such as pictures, cell structures, etc., effort should be made to present the data in a matrix format. If all required columns (or rows) of the data cannot be displayed on a single line, additional listings should be generated such that the listings could be viewed together to observe the data in matrix format. If irrelevant data is contained in the matrix, these values should be set to blank for printout purposes.

6.3.2 GRAPHICAL PRESENTATIONS

Graphical presentations have advantages over numerical presentations in that much more data can be presented in an easily assimilated manner. A disadvantage is that more computer time is needed and additional and sometimes complex mathematics must be programmed to construct visual pictures.

There are numerous texts, papers, and articles devoted to all phases of computer graphics; from simple graphs to complex 3-D color movies and holograms. This discussion will mention the advantages of a few basic types of graphic presentations with simplified algorithms for

producing more complex plots such as 3-D and surface plots with hidden line removal.

6.3.2.1 RECTANGULAR PLOT

The rectangular plot is probably the simplest and most used of all types of plots. It simply involves plotting a dependent variable or variable on a vertical scale as a function of an independent variable on a horizontal scale. Uses include quick-look editing, observing trends and functional relationships.

6.3.2.2 POLAR PLOT

The polar plot is useful for pictorially representing the function $g = f(\Theta)$. To plot, the function should be mapped into rectangular plot coordinates (U,V) by the following:

$$U = g \sin(\Theta)$$
$$V = g \cos(\Theta) \qquad (6.3\text{-}1)$$

Useful examples of polar plots include vulnerability envelopes, antenna patterns, and radiation patterns.

6.3.2.3 HISTOGRAM

Histogram plots are used in determining the distribution of a given set of data. They are often used in conjunction with and in lieu of statistical measurements. A goodness of fit can often be inferred by a simple histogram.

6.3.2.4 TIME-HISTORY PLOT

The time-history plot is used for observing data trends or drifts, noise, biases, anamolies, timing problems and interrelationships between variables. The most uncomplicated time

history is a simple rectangular plot with time the independent variable and the test item the dependent variable.

As a general rule, however, the time of interst is of such magnitude as to preclude putting it on a single frame (plot). To achieve the continuous format, several frames must be abutted, sometimes requiring complex programming.

6.3.2.5 THE 3-D PLOT

The 3-D type plot is a plot whereby relationships in width, depth, and height may be observed in a single picture. An extension to this concept may be a family of functions displayed in some increment of a changing dependent variable.

There are many methods of constructing a 3-D picture using various gray-scale techniques, color schemes, and geometrics. This discussion will present three geometric methods for determining a given point represented by three coordinates (X,Y,Z) on a plotting plane in a 3-dimensional framework. A line can be drawn by determining the location of its two end points.

6.3.2.5.1 OBLIQUE METHOD

The oblique picture is one in which two of the axes are always at right angles to each other, being in a plane parallel to the image plane with the third or "depth" axis being at any angle (except 90 degrees) to the vertical (60 degrees or 45 degrees being generally used). The location of the point (X,Y,Z) can be found by going along one axis at a distance equal to the corresponding coordinate and then parallel to each of the other axes at distances equal to the corresponding coordinates. This can also be done mathematically by finding the horizontal and vertical distances from the point of the origin in the image plane to the point in question in the image plane.

As an example, consider the coordinate system drawn as indicated in Figure 6-1.

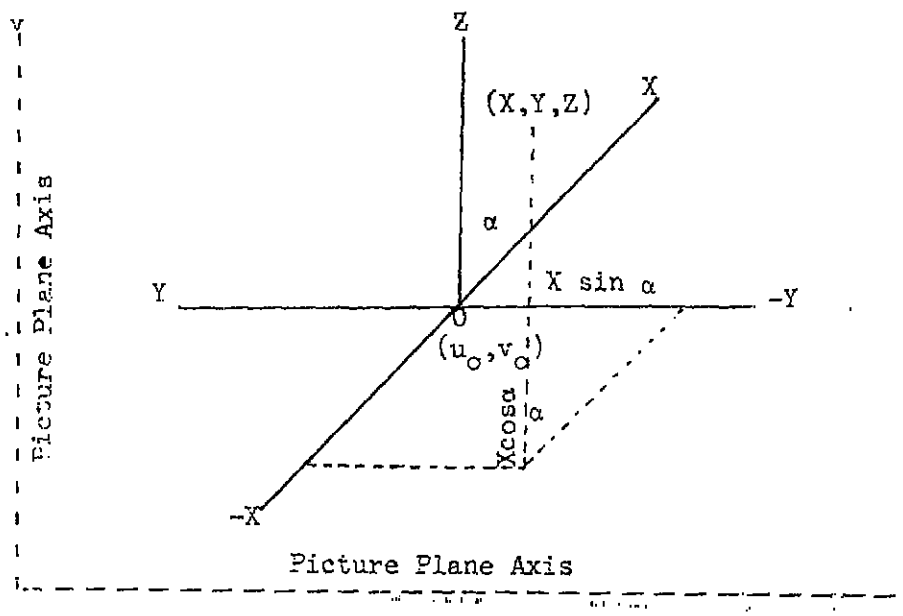


Figure 6-1

If α is taken as the positive angle between the positive X axis and the positive Z axis and (u,v) are the horizontal and vertical coordinates of the point (X,Y,Z) in the image plane relative to the picture origin,

$$u = X\text{Sin}(\alpha)-Y$$

$$v = X\text{Cos}(\alpha)+Z \qquad (6.3\text{-}2)$$

Although the oblique method is a relatively simple means of dipicting 3-D, a certain amount of distortion may exist if angle α is not carefully chosen.

6.3.2.5.2 ROTATION MATRIX

The following matrix is useful for rotating the point (X,Y,Z) through angles ψ, θ, ϕ (attitude angles of a viewer) for projection to a plane normal to a line of sight. The angles are defined in reference to the coordinate system depicted in Figure 6-2. Positive rotation is clockwise looking out the axis of rotation.

$$\Omega = \begin{vmatrix} C\Psi C\Theta & S\Psi\ C\Theta & -S\Theta \\ C\Psi S\Theta S\phi - S\Psi C\phi & S\Psi S\Theta S\phi + C\Psi C\phi & C\Theta S\phi \\ C\Psi S\Theta C\phi + S\Psi S\phi & S\Psi S\Theta C\phi - C\Psi S\phi & C\Theta S\phi \end{vmatrix} \quad (6.3\text{-}3)$$

C indicates Cosine function

S indicates Sine function

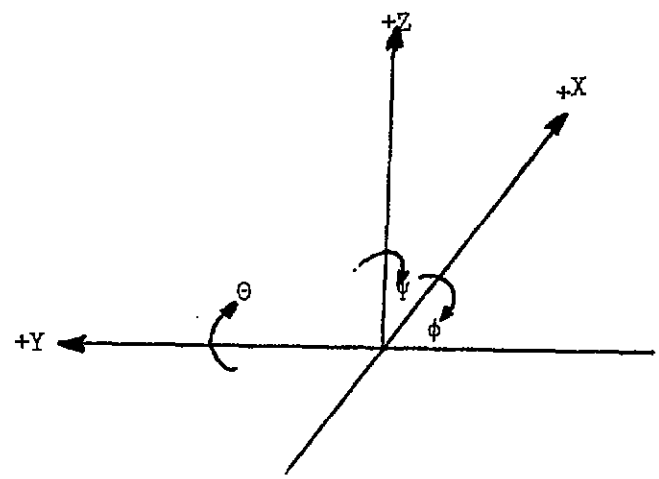


Figure 6-2

6.3.2.5.3 AXONOMETRIC METHOD

The axonometric method is theoretically an orthographic projection (parallel projection to the view plane) of an object to the image plane; the object being rotated such that three faces show. If ψ, ϕ, θ are the attitude angles of the viewer, the coordinates (u,v) in the picture plane of a given point (X,Y,Z) can be found as follows:

$$\begin{vmatrix} u \\ v \end{vmatrix} = \Gamma \begin{vmatrix} X \\ Y \\ Z \end{vmatrix} \qquad (6.3\text{-}4)$$

Where Λ is a 2x3 matrix defined as follows:

$$\Gamma_{i,j} = \Omega^{t}_{i+1,j} \qquad i = 1,2 \qquad j = 1,2,3 \qquad (6.3\text{-}5)$$

In the axonometric method, the picture plane does not need to be fixed but can be located anywhere along the line of sight (LOS). (See Figure 6-3.)

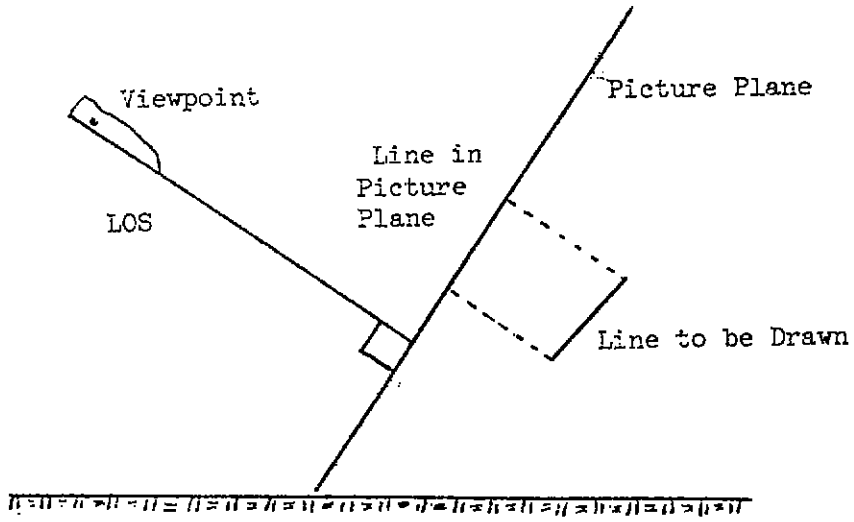


Figure 6-3

6.3.2.5.4 PERSPECTIVE METHOD

The perspective method is much like the axonometric method in that a viewpoint is specified and the object is rotated through the aspect angles of the viewer. Instead of orthographic projection, the rotated point is projected to a fixed image plane along a line from the point in question to the viewpoint (see Figure 6-4).

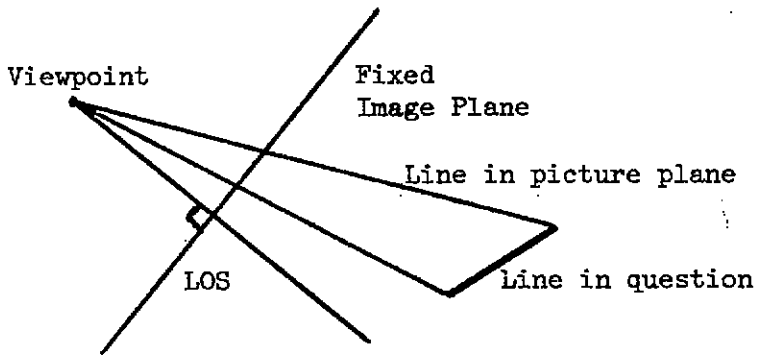


Figure 6-4

If ψ, ϕ, Θ are attitude angles of the viewer as defined previously, then the coordinate (u,v) in the picture plane can be found as follows:

$$\begin{vmatrix} X \\ Y \\ Z \end{vmatrix}_R = \Omega \begin{vmatrix} X \\ Y \\ Z \end{vmatrix} \qquad (6.3\text{-}6)$$

$$u = (Y_R)(R_V/X_R) \qquad v = (Z_R)(R_V/X_R) \qquad (6.3\text{-}7)$$

R_V is the normal distance from the picture frame to the viewpoint.

In equation (6.3-6) above, the coordinates (X_R, Y_R, Z_R) represent a point defined in a system where the origin is at the viewpoint with the positive X axis along the LOS, the positive Y axis to the left and the positive Z axis up. The equations for (u,v) in equation (6.3-7) result from a similar triangle relationship developed in basic projective geometry. To illustrate, let A be the plane normal to the LOS and passing through the point (X, Y, Z) and let B be the image plane (see Figure 6-5).

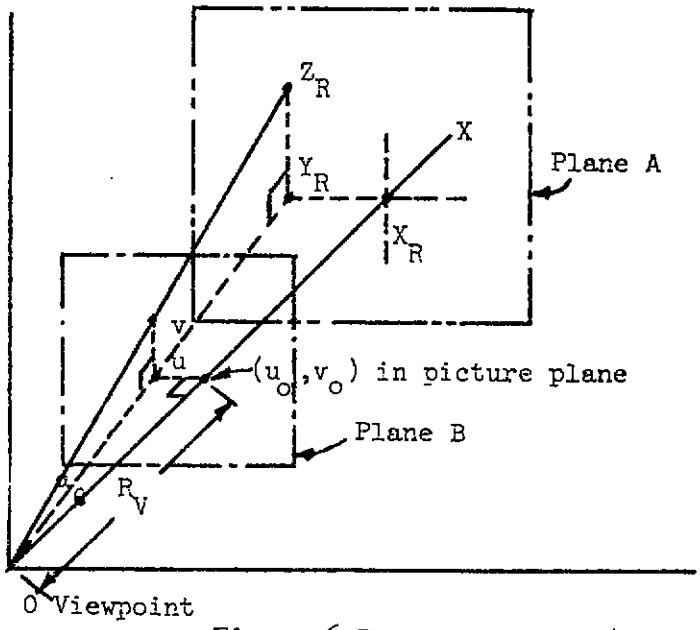


Figure 6-5

From Figure 6-5

$$u = R_v \tan\beta$$

$$X_R = R_R \tan\beta$$

or

$$\frac{u}{X_R} = \frac{R_v \tan\beta}{R_v \tan\beta}$$

$$\frac{u}{Y_v} = \frac{R_V}{X_R}$$

and

$$u = Y_v (R_v / X_R)$$

$$v = \frac{R_v}{\cos\beta} \tan\alpha$$

$$Z_R = \frac{X_R}{\cos\beta} \tan\alpha$$

$$\frac{v}{Z_R} = \frac{\frac{-R_v}{\cos\beta} \tan\alpha}{\frac{X_R}{\cos\beta} \tan\alpha}$$

$$\frac{v}{Z_R} = \frac{-R_v}{X_R}$$

$$v = Z_R (R_v / X_R)$$

R_v/X_R can be imagined as a variable scaling factor which scales the object image as a function of distance from the viewer. The perspective method, though more complex, provides a picture in which it is easier to visualize relative distances.

6.3.3 COMPUTER GENERATED MOVIES

An area of computer graphics worthy of mentioning is the use of the computer for generation of a series of pictures on film suitable for showing with a movie projector. The effect is an animated sequence approximating the dynamic actions of the objects in the pictures.

The steps for constructing a simplified movie of an object described by line structure is as follows:

1. Advance frame
2. Scale frame
3. Rotate all objects through view angles
4. Construct object on frame as per one of the previously described methods.
5. Advance frame

The steps are deceptively simple. The most difficult is usually the scaling of the frame and objects such that a realistic picture is achieved.

For further information, the reader is referred to the paper "Constructing 2-D Pictures of 3-D Objects With A Digital Computer" (ref 6.4), and Program P1932, "Generalized Movie Making Program" developed by the Directorate of Computer Sciences, ADTC, Eglin AFB, FL.

6.3.4 HIDDEN LINE ELIMINATOR

Numerous techniques have been developed for the removal of hidden lines (lines that are not seen when viewing an object or surface) in a picture by a computer. As a general rule, each technique has peculiar applications. The reader is referred to the bibliography for references on the various techniques.

One technique that is fairly easy to implement and has application when attempting to plot a surface, a matrix, or family of functions makes use of the following well known principle: If a surface can be described by a family of curves and the curves are ordered from the foreground to the background, a curve becomes invisible at points of intersection with curves that are further in the foreground. These points of intersection may be easily found if a "visibility" curve is established in the 2-dimensional plotting system consisting of the maximum (positive up) vertical plotting unit encountered for each horizontal plotting unit. The new curve to be plotted becomes invisible at all points where the vertical units of the new curve are below the corresponding vertical units on the visibility curve. A new visibility curve is established each time a new curve is plotted.

NOTE

The above assumes the surface does not become visible from the underside. If the surface is visible from the underside, a "minimum" visibility line may also be established consisting of the minimum vertical plotting unit encountered for each horizontal plotting unit. The curve in question is invisible at all points where the corresponding vertical units are below the maximum and above the minimum visibility curves.

The subroutine PLT3D1 included in paragraph 6.4 makes use of the above principle for making an oblique plot of a family of curves. PLTMTX may be used in conjunction with PLT3D1 for plotting a matrix.

6.4 USEFUL GRAPHIC SUBROUTINES

The following FORTRAN subroutines may be used to construct time-history plots and oblique 3-D surface plots. Use is made of an SC4020 Plotting Package which contains routines for constructing lines, scaling, and labeling. If the user does not have access to the SC4020 Plotting Package, appropriate routines will need to be substituted. The algorithm, however, will remain the same. Comments within each routine define the routine function and interaction with other routines.

```
      SUBROUTINE PLOTIT(TIME,NPGR,ISMB,DX,DY,DZ)
      DIMENSION DX(NPGR),DY(NPGR),DZ(NPGR) ,ISMB(NPGR)
     DIMENSION  CENT(3)
      DIMENSION SCAL(3)
     DIMENSION IBLINE (10),ITLINE(10)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     THIS IS A GENERAL PURPOSE ROUTINE FOR PLOTTING 1-3 GRAPHS ON
C     A CONTINUOUS ABUTTED FRAME OUTPUT.
C       IT USES AN SC-4020 PLOTTING PACKAGE TO CONSTRUCT LINES AND LABEL
C
C**** CONTROL VARIABLES
C      TIME =HORIZONTAL  VARIABLE
C     NPGR= NO. OF POINTS TO BE PLOTTED ON A SINGLE GRAPH.
C     ISMB= ARRAY CONTAINING INTEGERS TO SELECT PLOTTING SYMBOLS
C     DX=VERTICAL VARIABLE ARRAY FOR TOP GRAPH
C     DY= VERTICAL VARIABLE ARRAY FOR MIDDLE GRAPH. (IF REQUESTED)
C     DZ=VERTICAL VARIABLE ARRAY FOR BOTTOM GRAPH.( IF REQUESTED)
C               NOTE-  IF VARIABLE =-999999. THEN IT IS CONSIDERED
C                       AND IS NOT PLOTTED OR ANNOTATED.
C
C     NGRAPH= NO OF GRAPHS TO BE CONSTRUCTED. (MAX 3)
C     DELT=TIME RATE. (THIS IS USED FOR DETERMINING SCALE)
C     ICALL =1 IF BEGINNING OF NEW PLOT.  (NEW FRAME IS STARTED AND
C                 LABELING IS PERFORMED.)
C           = 0 IF NOT BEGINNING OF NEW PLOT
C     NPI= NO OF POINTS PER INCH TO BE PLOTTED.  THIS IS USED WITH
C           DELT TO DETERMINE SCALE.)
C     RANGE = ARRAY CONTAINING UPPER AND LOWER LIMITS OF DX,DY,DZ.
C     CENT = ARRAY CONTAINING  MEDIAN VALUE OF RANGES
C     VERT= BCD ARRAY CONTAINING VERTICAL LABELS.
C     HOR = BCD ARRAY CONTAININT 3 HORIZONTAL LABELS.  (1 AT TOP, 2 AT
C         BOTTOM)
C     TLAB= RATE AT WHICH TIME LABELING AND ANNOTATION OF CX, CY, CZ,
C         IS TO BE DONE.
C     TMHCK= RATE AT WHICH TIME HACK MARKS ARE TO BE INSERTED.
C
C**** LOCAL VARIABLES
C
C     NEWFRM=.TRUE. INDICATES TIME FOR NEW FRAME.
C     LABEL=.TRUE. INDICATES TIME FOR ANNOTATION OF TIME, CX, CY, CZ.
C     ILEFT= LEFT STARTING RASTOR POSITION. (90 FOR FIRST FRAME, ZERO
C           OTHERWISE)
C     XRAST= GRAPH HEIGHT IN RASTOR UNITS.
C     FMTIM = TOTAL TIME FOR A SINGLE FRAME.
C     TIMSCL = TIME SCALE FACTOR.
C     SCAL = ARRAY CONTAINING SCALE FACTORS FOR DX, DY, DZ.
C     TIML= TIME CORRESPONDING TO LEFT OF FRAME<
C     TIMR= TIME CORRESPONDING TO RIGHT OF FRAME.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      COMMON/LABEL/VERT(3),HOR(6,3)
      COMMON/PLOT/NGRAPH,DELT,ICALL,NPI,RANGE(2,3),  TLAB,TMHCK
       EXTERNAL TABL1V
       LOGICAL NEWFRM,LABEL
```

```
      IF(ICALL .EQ.0)GO TO 5
      XRAST=(850-(NGRAPH-1)*50.)/NGRAPH
      ILY=935.-(XRAST/2.)+70.
     ILEFT = 100
     FMTIM=(NPI*6.85)*DELT
     TIMSCL= 1023./FMTIM
      CALL BIGV
     CALL FRAMEV(3)
      CALL CHSIZV(2,2)
     CALL RITSTV(13,19,TABL1V)
      CALL RITE2V(10,1000,1023,90,1,56,1,HOR(1,1),IER1)
      CALL RITE2V(10,47,1023,90,1,56,1,HOR(1,2),IER2)
      CALL RITE2V(10,20,1023,90,1,56,1,HOR(1,3),IER3)
     CALL IHMS(TIME,IH,IM,SEC)
     ISEC = SEC/30.
     ISEC = ISEC * 30
     XTIME = IH * 10000 + IM * 100 + ISEC
     IXX = ILEFT
     CALL LABLV(XTIME,IXX,73,6,1,6)
     TT1 = IH * 3600000 + IM * 60000 + ISEC * 1000
     TT2 = TIME
     IF(ICALL.EQ.1) TIME = TT1
      NEWFRM=.TRUE.
     TIMR = 0.
5     CONTINUE
     IF(TIME.GT.TIMR)NEWFRM=.TRUE.
      IF(.NOT.NEWFRM)GO TO 100
     IF(ICALL.EQ.0)ILEFT=0
      IF(ICALL.EQ.0)CALL FRAMEV(3)
     IF(ICALL.EQ.0.AND.TIME.GT.(TIMR+FMTIM))ILEFT = 200
     IYBGN1=935.
     DO 20 I=1,NGRAPH
     IYBGN2=IYBGN1-XRAST/2.
     IYBGN3=IYBGN1-XRAST
      IF(ICALL.EQ.0)GO TO 10
     SCAL(I)=XRAST/(RANGE(1,I)-RANGE(2,I))
     CENT(I) = (RANGE(1,I) + RANGE(2,I))/2.
     CALL APRNTV(0,-14,10,VERT(I),6,ILY)
     ILY=ILY-XRAST-50.
      CALL LABLV(RANGE(1,I),18,IYBGN1,7,1,6)
     FVALUE = RANGE(1,I) - ((RANGE(1,I) - RANGE(2,I))/2)
     CALL LABLV(FVALUE,18,IYBGN2,7,1,6)
      CALL LABLV(RANGE(2,I),18,IYBGN3,7,1,6)
     CALL LINEV(ILEFT,IYBGN1,ILEFT,IYBGN3)
10    CONTINUE
     CALL LINEV(ILEFT,IYBGN1,1023,IYBGN1)
     CALL LINEV(ILEFT,IYBGN2,1023,IYBGN2)
     CALL LINEV(ILEFT,IYBGN3,1023,IYBGN3)
     YDEL=XRAST/10
     DO 15 JK=1,4
     IBLINE(JK)=IYBGN2-YDEL*JK
     ITLINE(JK) = IYBGN2 + YDEL*JK
     CALL LINEV (ILEFT,IBLINE(JK),1023,IBLINE(JK))
  15 CALL LINEV (ILEFT,ITLINE(JK),1023,ITLINE(JK))
     IYBGN1=IYBGN1-XRAST-50.
```

```
20      CONTINUE
        TIMR=TIME+FMTIM*(1-ILEFT/1023.)
        TIML=TIME
        NEWFRM=.FALSE.
C****   COMPUTE NEXT LABEL TIME
         TIMHCK=TIME + TMHCK-AMOD(TIME,TMHCK)
        XX = TIMHCK
        IL = ILEFT
        IF(ICALL.EQ.1) IL=100
   25   IX = IL + (XX-TIML) * TIMSCL
        IF(IX.EQ.1023.OR.IX.EQ.1022) IX = 1021
        IF(IX.GT.1024) GO TO 35
        IYH = 85 + XRAST
        CALL LINEV (IX,85,IX,IYH)
        IF(NGRAPH.EQ.1) GO TO 30
        IYTH = IYH + 50
        IYH = IYTH + XRAST
        CALL LINEV (IX,IYTH,IX,IYH)
        IF(NGRAPH.EQ.2) GO TO 30
        IYTH = IYH + 50
        IYH = IYTH + XRAST
        CALL LINEV (IX,IYTH,IX,IYH)
   30   CONTINUE
        XX = XX + TMHCK
        GO TO 25
   35   CONTINUE
95      TIMLAB=TIME+TLAB-AMOD(TIME,TLAB)
        IF(ICALL.EQ.1) TIME = TT2
100     CONTINUE
        LABEL=.FALSE.
145      CONTINUE
        IF(TIME.LT.TIMLAB)GO TO 150
        IF(TIMLAB.LT.TIML)GO TO 95
        CALL IHMS(TIMLAB,IH,IM,SEC)
        XTIME = IH * 10000 + IM * 100 + SEC
        IXX=(TIMLAB-TIML)*TIMSCL+ILEFT
        CALL POINTV(IXX,85,24,2)
        IXX=IXX-20
        IF(IXX.LT.8)IXX=8
        IF((IXX+48).GT.1023)IXX=975
        CALL LABLV(XTIME,IXX,73,6,1,6)
        LABEL = .TRUE.
        TIMLAB=TIMLAB+TLAB
        GO TO 145
150     CONTINUE
        IF(TIME.LT.TIMHCK)GO TO 153
         IX=(TIMHCK-TIML)*TIMSCL+ILEFT
        CALL POINTV(IX,85,5,1)
        IYH=85 + XRAST
        IF(NGRAPH.EQ.1)GO TO 151
        IYTH = 85 + XRAST + 50
        CALL POINTV(IX,IYTH,5,1)
        IYH=IYTH+XRAST
        IF(NGRAPH.EQ.2)GO TO 151
        IYTH = IYTH + XRAST + 50
```

```
      CALL POINTV(IX,IYTH,5,1)
      IYH=IYTH+XRAST
151   CONTINUE
      TIMHCK=TIMHCK+TMHCK
      GO TO 150
153   CONTINUE
      IX=(TIME-TIML)*TIMSCL+ILEFT
      IY0=935.-XRAST/2.
      DO 155 I=1,NPGR
      IF(DX(I).GT.RANGE(1,1).OR.DX(I).LT.RANGE(2,1)) GO TO 1531
      IY = (DX(I) - CENT(1))*SCAL(1)+ IY0
      CALL POINTV(IX,IY,ISMB(I),IDUM)
      CALL POINTV(IX,IY,ISMB(I),IDUM)
1531  IF(NGRAPH.LT.2) GO TO 155
      IF(DY(I).GT.RANGE(1,2).OR.DY(I).LT.RANGE(2,2)) GO TO 1532
      IY = (DY(I)-CENT(2))*SCAL(2)+(IY0-XRAST-50.)
      CALL POINTV(IX,IY,ISMB(I),IDUM)
      CALL POINTV(IX,IY,ISMB(I),IDUM)
1532  IF(NGRAPH.LT.3) GO TO 155
      IF(DZ(I).GT.RANGE(1,3).OR.DZ (I).LT.RANGE(2,3)) GO TO 155
      IY = (DZ(I) - CENT(3))*SCAL(3)+IY0-2.*XRAST - 100.
       CALL POINTV(IX,IY,ISMB(I),IDUM)
       CALL POINTV(IX,IY,ISMB(I),IDUM)
155   CONTINUE
       IY0=935
      IF(.NOT.LABEL)GO TO 205
      IXX=IX-20
      IF(IXX.LT.8)IXX=8
       IF(IXX+48.GT.1023)IXX=975
       CALL POINTV(IX,IY0,-24,2)
       DO 160 I=1,NPGR
       III=I*12
      IF(DX(I).NE.-999999.)CALL LABLV(DX(I),IXX,IY0+III,6,1,3)
160   CONTINUE
      IF(NGRAPH.LT.2)GO TO 205
      IY1=IY0-XRAST-50.
       CALL POINTV(IX,IY1,-24,2)
       DO 170 I=1,NPGR
       III=I*12
      IF(DY(I).NE.-999999.)CALL LABLV(DY(I),IXX,IY1+III,6,1,3)
170   CONTINUE
      IF(NGRAPH.LT.3)GO TO 205
      IY2=IY1-XRAST-50.
      CALL POINTV(IX,IY2,-24,2)
       DO 180 I=1,NPGR
       III=I*12
      IF(DZ(I).NE.-999999.)CALL LABLV(DZ(I),IXX,IY2+III,6,1,3)
180   CONTINUE
205   CONTINUE
      RETURN
      END
```

PAGE

```
      SUBROUTINE PLTMTX(A,NRA,NCA,IVIEW,IOPT,ICROSS,X1,X2,Y1,Y2)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C     THIS IS A GENERAL PURPOSE ROUTINE FOR MAKING AN OBLIQUE PLOT OF
C       A MATRIX OF VALUES.  IT USES PLT3D1 TO ACTUALLY PLOT THE DATA.
C
C       A = INPUT ARRAY TO BE PLOTTED.
C       NRA=NUMBER OF ROWS IN A
C      NCA= NUMBER OF COLUMNS IN A
C       IVIEW INDICATES VIEWING ANGLE
C              = 1, INDICATES VIEW FROM BOTTOM TO TOP
C              = 2, INDICATES VIEW FROM RIGHT TO LEFT
C              = 3, INDICATES VIEW FROM TOP TO BOTTOM
C              = 4, INDICATES VIEW FROM LEFT TO RIGHT
C       IOPT = 0, INDICATES USE DEFAULT SCALING FOR VERTICAL SCALE
C       IOPT = 1, INDICATES VERTICAL SCALE WILL BE SET BY PROGRAMMER (WTOP)
C       INTERVAL AT WHICH POINTS BETWEEN CURVES WILL BE CONNECTED (USUALLY 1)
C       X1 = VARIABLE ASSIGNED TO LAST ROW
C      X2=VARIABLE ASSIGNED TO FIRST ROW
C      Y1= VARIABLE ASSIGNED TO FIRST COLUMN
C      Y2= VARIABLE ASSIGNED TO LAST COLUMN
C                    NOTE- IF DO NOT WISH TO ASSIGN VARIABLES, THEN
C                         SET X1=NRA, X2=1, Y1=1, Y2=NCA.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
       DIMENSION A(NRA,NCA)
      COMMON/MAXVAL/UFOR,UBACK,VLEFT,VRIGHT,WBOT,WTOP,
     .THETA,UAXIS,VAXIS,UVPLNE,IU(2,2),IV(2,2),IW(2,2),JLAB
      DIMENSION V(150),W(150)
       PRINT 9000,NRA,NCA,IVIEW,IOPT,ICROSS,X1,X2,Y1,Y2
9000   FORMAT(1X,5I5,4F10.4)
      PRINT 9001,((A(I,J),J=1,NCA),I=1,NRA)
9001   FORMAT(1X,13F10.4)
      ZMIN = 9999999.
      DX=(X2-X1)/(NRA-1)
      DY=(Y2-Y1)/(NCA-1)
      JLAB=0
      ZMAX = -999999.
      DO 15 I=1,NRA
       DO 15 J=1,NCA
      ZMAX=AMAX1(ZMAX,A(I,J))
      ZMIN=AMIN1(ZMIN,A(I,J))
   15 CONTINUE
       IF(IVIEW.GT.4.OR.IVIEW.LT.1)IVIEW=1
      IF(IOPT.EQ.1)GO TO 3
      WTOP=2.*ZMAX
      WBOT=ZMIN
      THETA=.7854
3     CONTINUE
      UVPLNE=WBOT
      IF(ZMIN.LE.0.AND.ZMAX.GE.0)UVPLNE=0.0
      CLIPU=99999999.
      CLIPD=-999999999.
```

```
      XNOISE=0.0
      GO TO (11,12,13,14),IVIEW
11    UFOR=X2
      UBACK=X1
       DELU=-DX
      DELV=DY
      VLEFT=Y1
      VRIGHT=Y2
      GO TO 5
12    UFOR=Y2
      UBACK=Y1
        DELU=-DY
       DELV=-DX
      VLEFT=X2
       VRIGHT=X1
       GO TO 5
13    UFOR=X1
      UBACK=X2
       DELU=DX
       DELV=-DY
       VLEFT=Y2
       VRIGHT=Y1
      GO TO 5
14    UFOR=Y1
      UBACK=Y2
      DELU= DY
      DELV= DX
      VLEFT=X1
      VRIGHT=X2
5      CONTINUE
       UAXIS=VLEFT
      VAXIS=UBACK
      DO 10 I=1,150
10    W(I)=WROT
       NCURV=ABS((UFOR-UBACK)/DELU)+2.5
       NPTS=ABS((VRIGHT-VLEFT)/DELV)+2.5
      DO 700 I=1, NCURV
      U=UFOR+(I-2.)*DELU
      IF(I.EQ.1)U=UFOR
      DO 590 J=1,NPTS
      V(J)=VLEFT+(J-1.)*DELV
      IF(J.EQ.NPTS)V(J)=V(J-1)
      IF(I.EQ.1.OR.J.EQ.NPTS) GO TO 590
      GO TO (21,22,23,24),IVIEW
21    II=NRA-(I-2)
      JJ=J
       GO TO 25
22     JJ=NCA-(I-2)
      II=NRA-(J-1)
       GO TO 25
23    II=I-1
      JJ=NCA-(J-1)
       GO TO 25
24     JJ=I-1
       II=J
```

```
25      CONTINUE
        PRINT 9004,NCURV,NPTS,I,J,II,JJ ,IVIEW
9004    FORMAT(1X,10I5)
  580   CONTINUE
        W(J)=A(II,JJ)
         IF(W(J).GT.CLIPU)W(J)=CLIPU
         IF(W(J).LT.CLIPD)W(J)=CLIPD
        IF(ABS(W(J)).LT.XNOISE)W(J)=XNOISE
  590   CONTINUE
        IPASS=ICROSS
        IF(I.EQ.1) IPASS=-1

        CALL PLT3D1(U,V,W,IPASS,NPTS)
  700   CONTINUE
999     CONTINUE
        DELW=(WTOP-WBOT)/10.
         NLV=(NPTS-1)/15+1
         NLU=(NCURV-1)/20+1
        CALL PLT3D1(UBACK,DELW,0.0,-6,1)
        CALL PLT3D1(UBACK,DELV,0.0,-7,NLV)
        CALL PLT3D1(UBACK,DELU,0.0,-8,NLU)
        RETURN
         END
         SUBROUTINE PLT3D1(UA,VA,WA,IPASS,NPT)
         COMMON/CLASF/ICLASS,TITLE(7)
C
C**** THIS SUBROUTINE WILL MAKE OBLIQUE PLOT OF V VS W FUNCTION FOR
C     A GIVEN U.
C
C     U= VALUE OF INDEPENDENT VARIABLE U FOR A GIVEN V VS W FUNCTION.
C     VA = ONE DIMENSIONAL ARRAY CONTAINING INDEPENDENT VARIABLE V.
C     WA=ONE DIMENSIONAL ARRAY CONTAINING DEPENDENT VARIABLE W.
C                          WA(I)= F(U,VA(I))
C     NPTS= NO. OF PTS IN VA, WA ARRAYS.
C
C    NOTE : NPTS=IABS(NPT)
C                 IF(NPT.LT0) THEN WA(I)=F(UA(I),VA(I))
C
C     IPASS=-1 IF THIS IS FIRST CALL FOR NEW FRAME.
C      IPASS = -6 DRAW LINES FOR W SCALE AND LABEL W SCALE
C                 VA(1) IS     DELTA W FOR LABELING
C                 NPTS = LABEL EVERY NPTS LINE
C      IPASS = -7 DRAW LINES FOR V SCALE AND LABEL V SCALE
C                 VA(1) IS     DELTA V FOR LABELING
C                 NPTS = LABEL EVERY NPTS LINE
C      IPASS = -8 LABEL U SCALE
C                 VA(1) IS DELTA U FOR LABELING
C                 NPTS = LABEL EVERY NPTS LINE
C      IPASS = -2 HIDDEN LINES ARE NOT ELIMINATED,
C                 NOT CONNECTED TO PREVIOUS LINE
C      JLAB IS ALPHA LABEL FOR U SCALE
C                 JLAB = 0 NOT USED FOR THIS CALL
C     IPASS=N INDICATES EVERY NTH POINT ON THIS CURVE TO NTH POINT OF
C             PREVIOUS CURVE BE CONNECTED.
C
```

```
      COMMON/MAXVAL/UFOR,UBACK,VLEFT,VRIGHT,WBOT,WTOP,THETA,
     .  UAXIS,VAXIS,UVPLNE ,IU(2,2),IV(2,2),IW(2,2)
     .,JLAB

     MAXVAL COMMON IS USED FOR DETERMING SCALES.
     UFOR=EXTREME VALUE OF U FOR FOREGROUND.
     UBACK= EXTREME VALUE OF U FOR BACKGROUND.
     VLEFT= LEFT MOST VALUE OF V.
     VRIGHT= RIGHT MOST VALUE OF V
     WBOT=MIN VALUE OF W.
     WTOP=MAX VALUE OF W
     THETA=ANGLE OF OBLIQUE AXIS WITH VERTICAL (RADIANS).  NOTE- IF THIS
     IS ZERO, THEN PLOT WILL BE TWO DIMENSIONAL WITH U AND W AXES CO-
     INCIDENT.
     UAXIS= VALUE OF V AT WHICH UAXIS IS TO BE DRAWN.
     VAXIS= VALUE OF U AT WHICH V AXIS IS TO BE DRAWN.
     UVPLNE= VALUE OF W AT WHICH UAXIS AND VAXIS INTERSECT.

      DIMENSION IXSAV(601),IYSAV(601)
      DIMENSION UA(NPT),VA(NPT),WA(NPT)
      COMMON    MINY(1024),MAXY(1024)
      LOGICAL TOUT
      NPTS=IABS(NPT)
      PRINT 8880,UFOR,UBACK,VLEFT,VRIGHT,WBOT,WTOP,IPASS,NPTS
8880   FORMAT(1X,6F10.4,2I5)
C      PRINT 1000,  U,(VA(JJ),WA(JJ),JJ=1,NPTS)
1000   FORMAT(1X,F10.3/(11X,10F10.3))
      IF(IPASS.GE.0)GO TO 100
      IF(IPASS.EQ.-2)GO TO 100
      IF(IPASS.LE.-6) GO TO 305
C**** IF FIRST PASS, ADVANCE FRAME, SET SCALES, AND INITIALIZE MAX AND
C     MIN FUNCTIONS.   THE FIRST CURVE TO BE PLOTTED WILL BE THE FARTHEST
C     IN THE FOREGROUND.    SUCCEEDING VALUES OF U MUST BE EITHER
C     ASCENDING OF DESCENDING.
C
      CALL FRAMEV(3)
C**** FIND SCALE VALUES TO MAKE VARIABLES PROPORTIONAL
      ULNTH=ABS(UFOR-UBACK)
      VLNTH=ABS(VRIGHT-VLEFT)
      WLNTH=ABS(WTOP-WBOT)
      SCMIN=AMIN1(ULNTH,VLNTH,WLNTH)
      SCALU=SCMIN/ULNTH
      SCALV=SCMIN/VLNTH
      SCALW=SCMIN/WLNTH
        IF(IPASS.NE.-3)GO TO 19
      SCALU=1.
      SCALV=1.
      SCALW=1.
19    CONTINUE
      ULNTH=ULNTH*SCALU
       DIFX=ULNTH*SIN(THETA)
       IF(VRIGHT.LT.VLEFT)DIFX=-DIFX
      IF(THETA.GE.0.0)GO TO 20
      XRIGHT=VRIGHT*SCALV
      XLEFT=VLEFT*SCALV+DIFX
```

```
        GO TO 30
20      CONTINUE
        XLEFT=VLEFT*SCALV
        XRIGHT=VRIGHT*SCALV+DIFX
30      CONTINUE
        YTOP=(WTOP-WBOT)*SCALW+ULNTH*COS(THETA)+WBOT*SCALW
        YBOT=WBOT*SCALW
        YSCALE=YTOP-YBOT
        XSCALE=ABS(XRIGHT-XLEFT)
        CALL XSCALV(XLEFT,XRIGHT,74,50)
        CALL YSCALV(YBOT,YTOP,50,74)
        DO 60 I=1,1024
        MINY(I)=1000
        MAXY(I)=0
        IF(I.GT.500)GO TO 60
        IXSAV(I)=0
        IYSAV(I)=0
60      CONTINUE
C**** CONSTRUCT THE AXES.
C
C    (V AXIS)
        DELU=ABS(VAXIS-UFOR)*SCALU
        DX=DELU*SIN(THETA)
        IDELX=(DX/XSCALE)*900.
        DY=DELU*COS(THETA)
        IDELY=(DY/YSCALE)*900.
        IX1=NXV(VLEFT*SCALV)+IDELX
        IY1=NYV(UVPLNE*SCALW)+IDELY
        IX2=NXV(VRIGHT*SCALV)+IDELX
        IY2=IY1
        CALL LINEV(IX1,IY1,IX2,IY2)
        IV(1,1)=IX1
        IV(2,1)=IY1
        IV(1,2)=IX2
        IV(2,2)=IY2
C
C     (W AXIS)
        IX1=NXV(UAXIS*SCALV)+IDELX
        IX2=IX1
        IY1=NYV(WBOT*SCALW)+IDELY
        IY2=NYV(WTOP*SCALW)+IDELY
        CALL LINEV(IX1,IY1,IX2,IY2)
        IW(1,1)=IX1
        IW(2,1)=IY1
        IW(1,2)=IX2
        IW(2,2)=IY2
C
C     (U AXIS)
C
        IX1=NXV(UAXIS*SCALV)
        IY1=NYV(UVPLNE*SCALW)
        Y2=ULNTH*COS(THETA)+UVPLNE*SCALW
        X2=DIFX+UAXIS*SCALV
        IX2=NXV(X2)
        IY2=NYV(Y2)
```

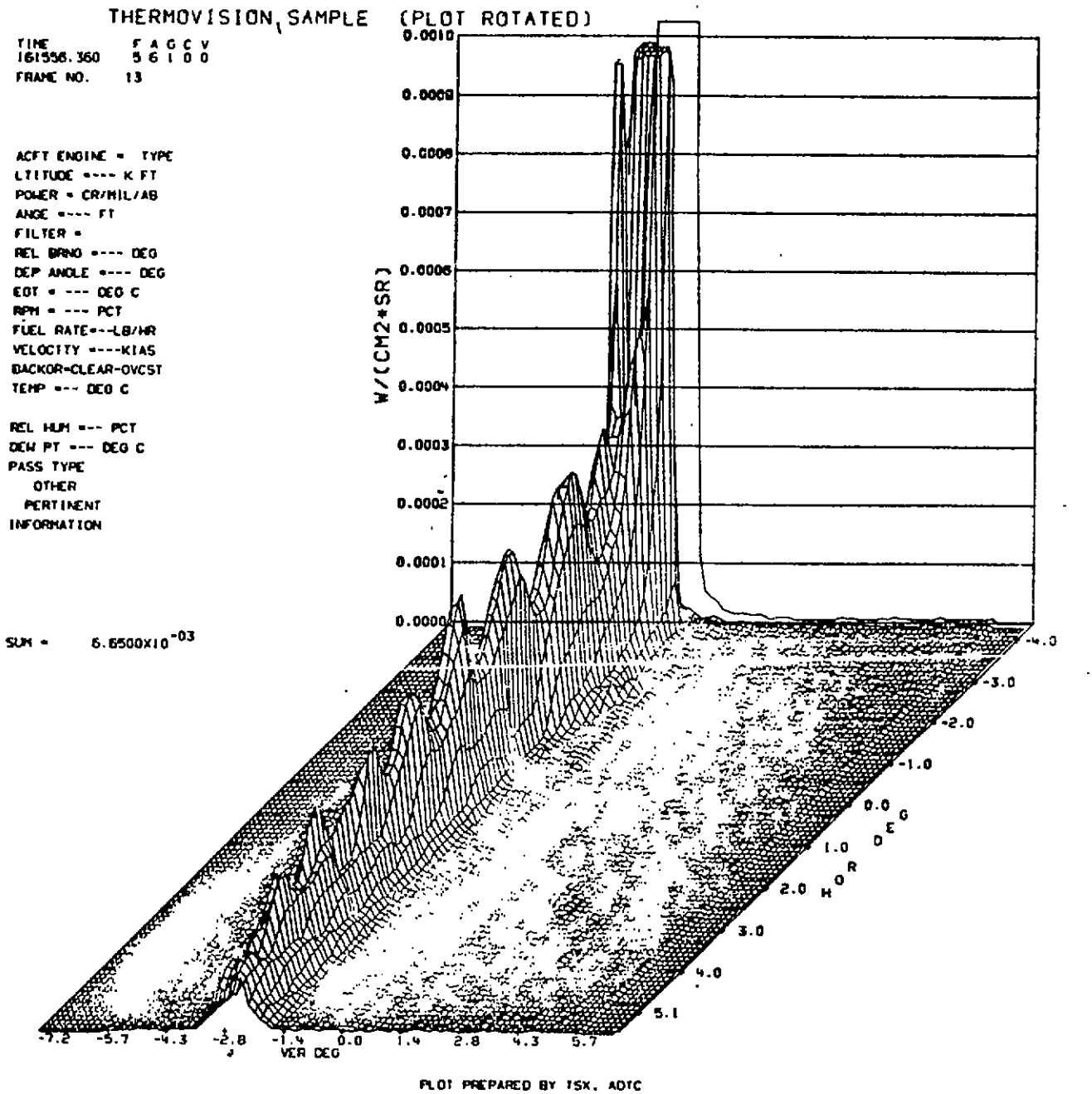
```
      CALL LINEV(IX1,IY1,IX2,IY2)
       IU(1,1)=IX1
      IU(2,1)=IY1
       IU(1,2)=IX2
       IU(2,2)=IY2
C
100    CONTINUE
       U=UA(1)
      IOUT=.TRUE.
      ICOUNT=ICOUNT+1
       J=NPTS+1
       DO 180 JJ=1,NPTS
       IF(THETA)110,120,120
110     J=JJ
      GO TO 125
120     J=J-1
125     CONTINUE
       IF(NPT.LT.0.)U=UA(J)
       IF(NPT.GT.0.AND.JJ.GT.1)GO TO 126
      DELU=ABS(U-UFOR)*SCALU
      DX=DELU*SIN(THETA)
      IDELX=(DX/XSCALE)*900.
      DY=DELU*COS(THETA)
      IDELY=(DY/YSCALE)*900.
126     CONTINUE
      IX2=IXV(VA(J)*SCALV)+IDELX
      IY2=IYV(WA(J)*SCALW)+IDELY
       IF(IPASS.LE.0)GO TO 170
      IF(MOD(J-1,IPASS).NE.0) GO TO 170
      NX=IABS(IX2-IXSAV(J))
       IF(NX.EQ.0)NX=1
       INCX=1
       IF(IXSAV(J).GT.IX2)INCX=-1
      DX=NX
       DY=IY2-IYSAV(J)
      RATIO=DY/DX
       IYST=IYSAV(J)
       IX=IXSAV(J)
       IN=0
       DO 165 I=1,NX
       IX=IX+INCX
       IY=IYST+I*RATIO
       IF(IY.LE.MAXY(IX).AND.IY.GE.MINY(IX))GO TO 130
      IN=0
       GO TO 160
130      IF(IN.EQ.1.OR.I.EQ.1)GO TO 135
      IXCK=IXSAV(J)
      IF(IY.GT.MAXY(IX).AND.IYSAV(J).LT.MAXY(IXCK)) IYSAV(J)=MAXY(IXCK)
       CALL LINEV(IXSAV(J),IYSAV(J),IX,IY)
135    IXSAV(J)=IX
       IYSAV(J)=IY
      IN=1
160    CONTINUE
      MAXY(IX)=MAXO(IY,MAXY(IX))
      MINY(IX)=MINO(IY,MINY(IX))
```

```
165     CONTINUE
      IF(IN.EQ.0)CALL LINEV(IXSAV(J),IYSAV(J),IX2,IY2)
170     CONTINUE
        IF(IPASS.NE.-2)GO TO 178
        IF(JJ.EQ.1)GO TO 175
        CALL LINEV(IX1,IY1,IX2,IY2)
175     CONTINUE
      IX1=IX2
      IY1=IY2
        IF(JJ.NE.NPTS)GO TO 180
        RETURN
178       CONTINUE
          IXSAV(J)=IX2
      IYSAV(J)=IY2
180     CONTINUE
      IX1=IXV(VA(1)*SCALV)+ICELX
      IY1=IYV(HA(1)*SCALW)+ICELY
      IF(IY1.LT.MAXY(IX1).AND.IY1.GT.MINY(IX1)) IOUT = .FALSE.
      DO 300 J=2,NPTS
        IX2=IXSAV(J)
      IY2=IYSAV(J)
      NX=IABS(IX2-IX1)
      IF(NX.EQ.0)NX=1
        INCX=1
      IF(IX2.LT.IX1)INCX=-1
      DX=NX
      DY=IY2-IY1
      RATIO=DY/DX
      IX=IX1
        IYST=IY1
      DO 230 K=1,NX
      IX=IX+INCX
      IY=IYST+K*RATIO
      INDX=IX
      MXY=MAXY(INDX)
      MNY=MINY(INDX)
        IF(IY.LT.MXY.AND.IY.GT.MNY)GO TO 200
      IOUT=.TRUE.
      GO TO 220
200     CONTINUE
      IF(.NOT.IOUT) GO TO 210
        CALL LINEV(IX1,IY1,IX,IY)
210     CONTINUE
      IOUT=.FALSE.
      IX1=IX
      IY1=IY
220     CONTINUE
        MAXY(INDX)=MAX0(IY,MXY)
      MINY(INDX)=MIN0(IY,MNY)
230   CONTINUE
        IF(.NOT.IOUT)GO TO 234
      CALL LINEV(IX1,IY1,IX2,IY2)
234   CONTINUE
      IND1=IX1
        MAXY(IND1)=MAX0(IY1,MAXY(IND1))
```

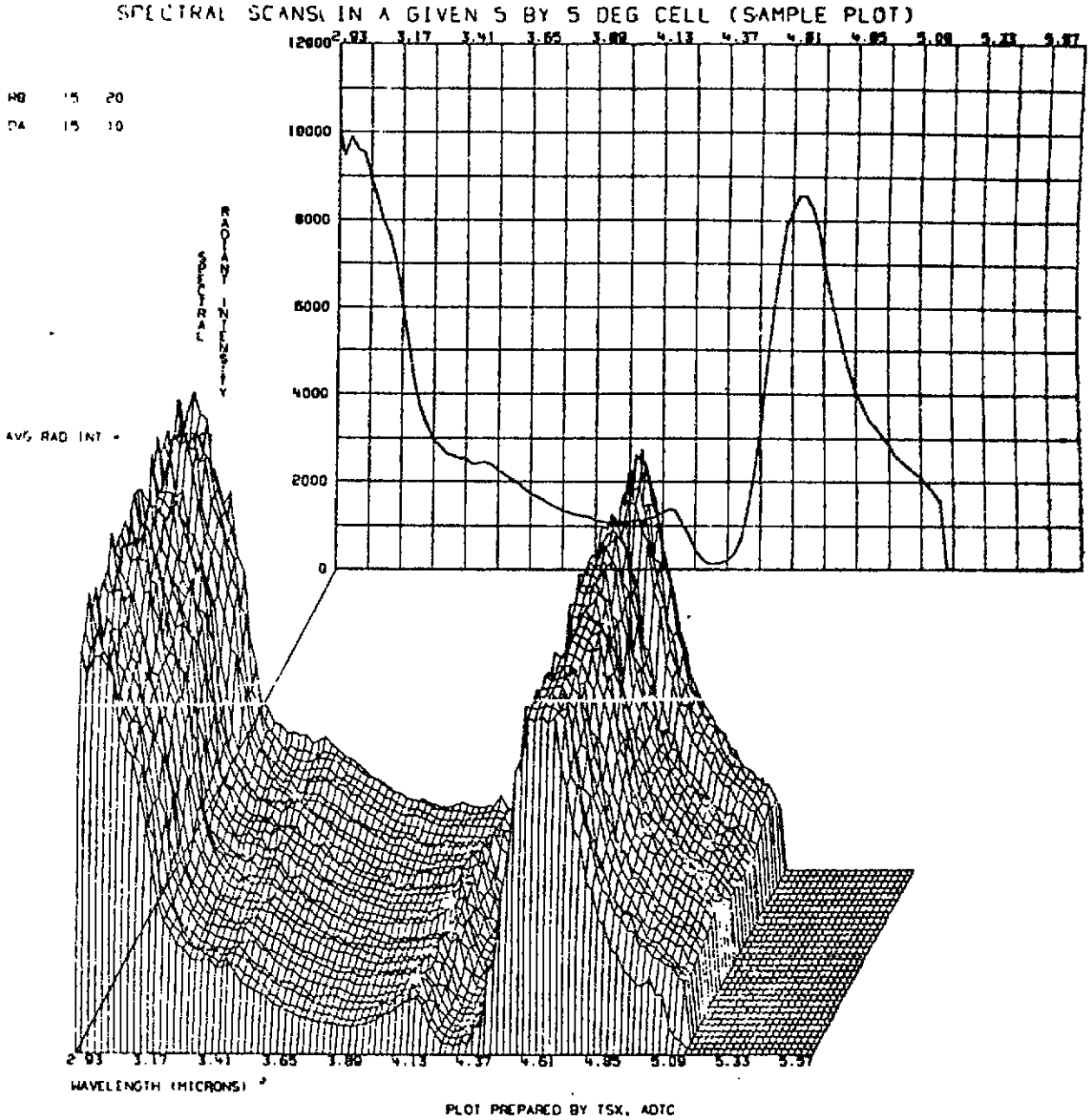
```
      MINY(IND1)=MIN0(IY1,MINY(IND1))
      IX1=IX2
      IY1=IY2
300   CONTINUE
      IF(JLAB.EQ.0)GO TO 301
C**** PRINT HOLERITH JLAB
      IXL=IXV(VRIGHT*SCALV)+IDELX+8
      IYL = IDELY + IYV(WBOT*SCALW)
      CALL PRINTV(8,JLAB,IXL,IYL)
      CALL LINE2V(IXL-12,IYL-1,5,0)
      CALL LINE2V(IXL-12,IYL-1,5,0)
  301 CONTINUE
      RETURN
  305 CONTINUE
      ILAB = 0
      IF(IPASS.EQ.-8) GO TO 350
      DELU = ABS(UBACK-UFOR) * SCALU
      DX = DELU * SIN (THETA)
      IDELX = (DX/XSCALE) * 900.
      DY = DELU * COS (THETA)
      IDELY = (DY/YSCALE) *900.
      IF(IPASS.EQ.-7) GO TO 330
      DELW = VA(1)
       XD=AMAX1(ABS(WTOP),ABS(WBOT))
       NDL=NDMAX(XD)
       NDIG=MAX0(5,NDL)
      W = WBOT
      IX1 = NXV(VLEFT  * SCALV)+ IDELX
      IX2 = NXV(VRIGHT  * SCALV)+ IDELX
  310 IY1 = NYV(W  * SCALW)+ IDELY
      IY2 = IY1
      CALL LINEV(IX1,IY1,IX2,IY2)
      IF(MOD(ILAB,NPTS).NE.0) GO TO 315
      CALL LABLV(W,IX1-52,IY1,NDIG ,1,NDL)
      CALL LABLV(W,IX1-52,IY1,NDIG ,1,NDL)
  315 CONTINUE
      ILAB= ILAB + 1
      W = W + DELW
      IF(W.GT.WTOP) GO TO 301
      GO TO 310
       XD=AMAX1(ABS(VLEFT),ABS(VRIGHT))
       NDL=NDMAX(XD)
       NDIG=MAX0(5,NDL)
  330 CONTINUE
      DELV = VA(1)
      V = VLEFT
       VR=VRIGHT+.01*(VRIGHT-VLEFT)
  340 IX1 = NXV(V  * SCALV)+ IDELX
      IY2 = NYV(WTOP  * SCALW)+ IDELY
      IY1 = NYV(WBOT  * SCALW)+ IDELY
      IX2 = IX1
      IX3 = IX1 - IDELX
      IY3 = IY1 - IDELY
      CALL LINEV(IX1,IY1,IX2,IY2)
      IF(MOD(ILAB,NPTS ).NE.0) GO TO 345
```

```
      CALL LABLV(V,IX2-28,IY2+6,NDIG ,1,NDL)
      CALL LABLV(V,IX2-28,IY2+6,NDIG ,1,NDL)
      CALL LABLV(V,IX3-28,IY3-6,NDIG ,1,NDL)
      CALL LABLV(V,IX3-28,IY3-6,NDIG ,1,NDL)
  345 CONTINUE
      ILAB = ILAB + 1
      V= V + DELV
       IF(VRIGHT.GT.VLEFT)GO TO 346
      IF(V.LT.VR    )GO TO 301
      GO TO 340
346   IF(V.GT.VR    )GO TO 301
      GO TO 340
  350 CONTINUE
       UB=UBACK+.01*(UBACK-UFOR)
      ULAB = UFOR
  351 CONTINUE
      DELAB = ABS(ULAB-UFOR) * SCALU
      DX = DELAB * SIN (THETA)
      DY = DELAB * COS (THETA)
      IDELX = (DX/XSCALE) * 900.
      IDELY = (DY/YSCALE) * 900.
      IX1 = NXV(VLEFT * SCALV) + IDELX
      IX2 = NXV(VRIGHT*SCALV)   + IDELX
      IY1 = NYV(WBOT * SCALW)   + IDELY
  355 CONTINUE
C     CALL POINTV(IX1,IY1,6,ANY)
C     CALL POINTV(IX2,IY1,6,ANY)
      IF(MOD(ILAB,NPTS ).NE.0) GO TO 358
C     CALL LABLV(ULAB,IX1-45,IY1,5,1,4)
C     CALL LABLV(ULAB,IX1-45,IY1,5,1,4)
      ICH = 4
      IF(ABS(ULAB).GT.9.) ICH = 5
      IF(ABS(ULAB).GT.99.) ICH = 6
      CALL LABLV(ULAB,IX2+4,IY1,ICH,1,ICH-2)
      CALL LABLV(ULAB,IX2+4,IY1,ICH,1,ICH-2)
  358 ILAB = ILAB + 1
      ULAB = ULAB + VA(1)
        IF(UBACK.LT.UFOR)GO TO 3441
      IF(ULAB.GT.UB    ) GO TO 301
      GO TO 351
3441    IF(ULAB.LT.UB    )GO TO 301
      GO TO 351
      END
       FUNCTION NDMAX(X)
C***** THIS FUNCTION RETURNS THE MAXIMUM DCCIMAL SCALE NEEDED
C      FOR USE WITH LABLV
        AX=ABS(X)*10.+.01
       DIG=ALOG10(AX)
       NDMAX=DIG
        IF(DIG.LT.0)NDMAX=0
       IF(NDMAX.GT.6)NDMAX=6
       RETURN
       END
      SUBROUTINE CLASS(ITI)
C**** THIS SUBROUTINE WRITES CLASSIFICATION AT TOP OF PICTIRE AND TITLE
```

```
C     IF REQUESTED.
      COMMON /CLASF/ ICLASS,TITLE(8)
       EXTERNAL TABL1V
      CALL CHSIZV(4,4)
      CALL RITSTV(23,33,TABL1V)
      IF(ICLASS-1)80,20,30
20     CALL RITE2V(412,1000,1023,90,2,12,-1,12HCONFIDENTIAL ,NE)
       CALL RITE2V(412,  20,1023,90,2,12,-1,12HCONFIDENTIAL ,NE)
      GO TO 80
30    CALL RITE2V(440,1000,1023,90,2,6,-1,6HSECRET ,NE)
      CALL RITE2V(440,  20,1023,90,2,6,-1,6HSECRET ,NE)
80    CONTINUE
      CALL CHSIZV(2,2)
      CALL RITSTV(13,18,TABL1V)
      IF(ITI.EQ.0) GO TO 90
      CALL RITE2V(100,989,1023,90,2,70,1,TITLE,NER)
90     CONTINUE
       RETURN
       END
```



OBLIQUE MATRIX PLOT WITH SAMPLE LABELING
PRODUCED USING PLT3D1



OBLIQUE PLOT OF FAMILY OF CURVES

PRODUCED USING PLT3D1

REFERENCES

[6-1] Yan-Yuen, Tao, "A Firmware Data Compression Unit," PB-228111. (Springfield, VA: National Technical Information Service, U.S. Dept. of Commerce)

[6-2] McWaters, Lynn, "Imagery Compression/Transmission Techniques," AD-770-300 (Springfield, VA: National Technical Information Services, U.S. Dept of Commerce)

[6-3] Alsberg, Peter A., "Space and Time Savings through Large Data Base Compression and Dynamic Restructuring," Proceedings of the IEEE, Vol. 63, No. 8, pp 1114-1116 (August 1965)

[6-4] Walters, H.S., "Constructing 2-D Pictures of 3-D Objects with a Digital Computer," pp 25-45 (ADTC, Eglin AFB, FL: Computer Sciences Laboratory, June 1973)

[6-5] Williamson, Hugh, "Algorith 420. Hidden-Line Plotting Program," Communications of ACM, Vol 15, pp 100-103, (February 1972)

[6-6] Sutherland, D.E., Sproul, R.F., Schumaker, R.P., "A Characterization of Ten Hidden-Surface Algorithms," ACM Computing Surveys, Vol. 6, pp 3-50 (March 1964)

[6-7] Chen, Tieu Chic, Ho, Irving T., "Storage-Efficient Representation of Decimal Data," Communication of the ACM, Vol 18, pp 49-52 (January 1975)

CHAPTER 7
SUMMARY

Data compression and maximization of information content has become a technology which can reduce: (1) computing costs, (2) data storage costs, (3) transfer time/costs, (4) hardware costs, and (5) response (decision-making) time. It is realistic to expect compression ratios in the range of 3:1 to 10:1 using techniques discussed in this document. Since any large data base problem may be amenable to more than one compression/maximization of information content technique, this document categorizes and describes individual techniques to aid the user in a choice for his application. In summarizing techniques, we may classify them as in the diagram, Figure 7-1.

7.1 REDUNDANT DATA REMOVAL TECHNIQUES

These techniques are successful if sampling rates are fixed and generally greater than the usual data information rates. They eliminate data samples that can be implied by examination of preceding or succeeding samples; or by comparison with arbitrary reference patterns.

7.2 TRANSFORM METHODS/LUMPED PARAMETER TECHNIQUES

This family of techniques operates on data samples via mathematical transformations whereby all the original data samples are irretrievably lost, but are represented by parameters in a domain other than time (such as frequency or sequency). The original data may be reconstructed within some error tolerance by the inverse transformations.

Figure 7-1

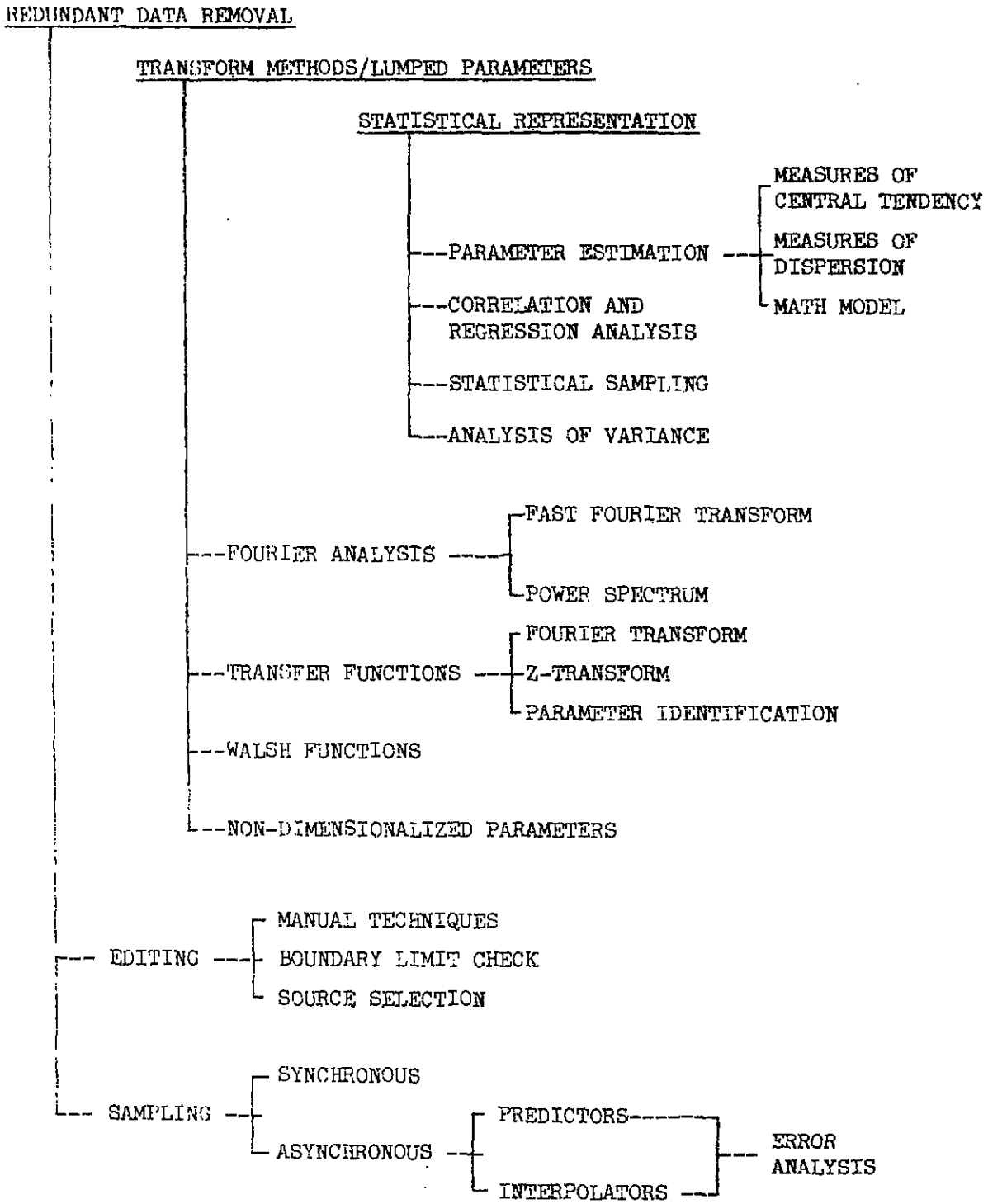
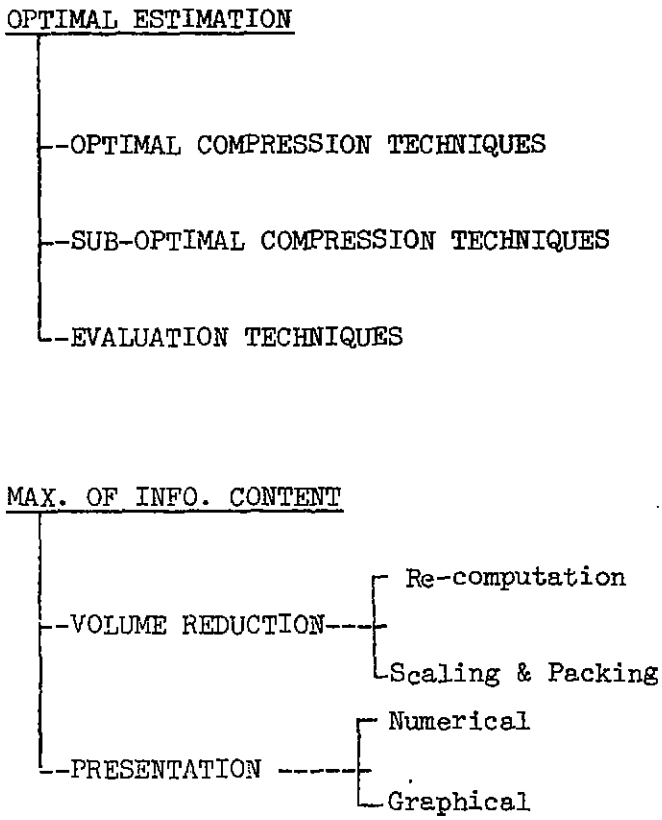

REDUNDANT DATA REMOVAL
TRANSFORM METHODS/LUMPED PARAMETERS
STATISTICAL REPRESENTATION
PARAMETER ESTIMATION
MEASURES OF CENTRAL TENDENCY
MEASURES OF DISPERSION
MATH MODEL
CORRELATION AND REGRESSION ANALYSIS
STATISTICAL SAMPLING
ANALYSIS OF VARIANCE
FOURIER ANALYSIS
FAST FOURIER TRANSFORM
POWER SPECTRUM
TRANSFER FUNCTIONS
FOURIER TRANSFORM
Z-TRANSFORM
PARAMETER IDENTIFICATION
WALSH FUNCTIONS
NON-DIMENSIONALIZED PARAMETERS
EDITING
MANUAL TECHNIQUES
BOUNDARY LIMIT CHECK
SOURCE SELECTION
SAMPLING
SYNCHRONOUS
ASYNCHRONOUS
PREDICTORS
INTERPOLATORS
ERROR ANALYSIS

Figure 7-1 (Continued)



7.3 STATISTICAL REPRESENTATION TECHNIQUES

As in the case of transform methods, the original data is lost and from there on is represented by other parameters such as statistical parameters, coefficients in a math model, or a smaller sampling. The original data may not be reconstructed.

7.4 OPTIMAL ESTIMATION TECHNIQUES

The objective of optimal techniques is to minimize some selected measure of error and to utilize all information concerning system dynamics, noise statistics, and initial conditions. An optimal technique provides a performance standard for comparison and evaluation of suboptimal approaches.

7.5 MAXIMIZATION OF INFORMATION CONTENT

Provides suggested practical techniques for reducing the sheer volume of data when trade-offs in accuracy and storage/retrieval costs can be accepted. Also, included are suggestions for presenting large amounts of data to the user via a few general purpose graphics routines.

DISTRIBUTION LIST

	COPIES
EXECUTIVE COMMITTEE	
Mr. O. E. Ova (KMR)	1
Dr. R. H. Duncan (USAWSMR)	1
Mr. F. W. Jones (USAYPG)	1
Col H. Taffet, USAF (AFETR)	1
Mr. F. Stoliker (AFFTC)	1
Mr. F. Kabasc (ADTC)	1
Col Robert H. Krumpe, USAF (AFSCF)	1
Mr. E. E. Clary (SAMTEC)	1
Col J. D. Salvucci, USAF (TFWC)	1
Mr. C. J. DiPol (NWC)	1
Mr. W. L. Miller (PMTC)	1
CAPT C. S. Wood, USN (AFWTF)	1
Mr. J. B. Paradis (NATC)	1
TECHNICAL REPRESENTATIVES	
Mr. J. L. Wymer (USAYPG)	1
Mr. C. V. Lochman (AFETR)	1
Mr. A. F. Miller (AFFTC)	1
LtCol K. H. Moses, USAF (ADTC)	1
Col J. W. Myers, USAF (AFSCF)	1
Mr. R. A. McLellan (TFWC)	1
Mr. W. T. Lamb (NWC)	1
Mr. K. I. Lichti (PMTC)	1
Mr. T. Ramirez (AFWTF)	1
Mr. J. Lynch (NATC)	1
Col M. J. Steger, USAF (SAMTEC)	1
DR&CG CHAIRMAN	
Mr. R. Barry (AFFTC)	2
DR&CG VICE CHAIRMAN	
Mr. J. Oddo (KMR)	2

	COPIES
DR&CG MEMBERS	
Mr. R. H. Dale (USAWSMR)	1
Mr. L. Hackamack (USAYPG)	1
Mr. W. M. Nolin (AFETR)	1
Mr. D. Weddle (NATC)	1
Capt A. G. Barry (6585th TG)	1
Mr. M. J. Smith (6585th TG)	1
Mr. H. S. Walters (ADTC)	1
Mr. D. E. Fletcher (SAMTEC)	1
Mr. T. Sushinski (TFWC)	1
Mr. I. B. Wertenberger (NWC)	1
Mr. J. M. Dorgan (NWC)	1
Mr. D. E. Ogden (PMTC)	1
Mr. A. J. Herleikson (PMTC)	1
Mr. T. E. Ford (PMTC)	1
LTJG W. Thompson, USN (AFWTF)	1
DR&CG ASSOCIATE MEMBERS	
Mr. J. A. Gilberto (USAOTEA)	1
Mr. R. Morton (TMSAEG)	1
Mr. R. N. Learn (NSWC)	1
LCDR R. H. Lee, USN (NPTR)	1
Mr. P. R. Harness (NASA/MSFC)	1
Mr. M. E. Bauder (Sandia Labs)	1
Mr. J. L. Rogers (Sandia Labs)	1
GROUP CHAIRMEN/VICE CHAIRMEN	
LtCol J. M. Young, USAF (AFFTC), DG Chairman	1
Mr. B. Krieger (SAMTEC), DG Vice Chairman	1
Mr. R. Thompson (ADTC), RSG Chairman	1
Mr. R. Geisinger (SAMTEC), RSG Vice Chairman	1
Mr. S. Garoutte (SAMTEC), JRIAIG Chairman	1
Mr. R. Acosta (USAYPG), ETMG Chairman	1
Mr. A. Kriss (TFWC), ETMG Vice Chairman	1
Mr. T. Frey (SAMTEC), FMG Chairman	1
Mr. E. Angle (PMTC), FMG Vice Chairman	1

	COPIES
Mr. R. Hixon (PMTC), MG Chairman	1
LtCol C. Bush, USAF (AFFTC), MG Vice Chairman	1
Mr. J. Cornell (SAMTEC), OSG Chairman	1
Mr. J. Wells (ADTC), OSG Vice Chairman	1
Mr. Q. Flint (USAWSMR), TCG Chairman	1
Mr. W. Steele (USAYPG), TCG Vice Chairman	1
Mr. G. Wooden (KMR), TG Chairman	1
Mr. H. Norfolk, Jr. (NATC), TG Vice Chairman	1
Mr. J. Broun (AFWTF), USG Chairman	1
Mr. D. Strietzel (KMR), USG Vice Chairman	1

CONTRIBUTORS

Mr. Barry L. Clark (NSWC)	1
Mr. Weldon Howell (USAWSMR)	1
Mr. Clifford J. Lemieux (PMTC)	1
DDC	12