JOURNAL OF THE CANADIAN SOCIETY OF EXPLORATION GEOPHYSICISTS
VOL. 18, NO. 1 (DEC. 1982), P. 23-33

ERROR CORRECTION IN FIELD RECORDED SEISMIC DATA[1]

M. GALBRAITH[2] and C.L. MACMINN[2]

ABSTRACT

A novel method is proposed to correct certain types of errors which sometimes arise in digitally recorded seismic data. During field recording, errors can occur because of equipment malfunction, improper gain settings, instrument noise, etc. This method corrects a particular class of such errors; namely, those that are random in nature and of short duration. Such errors are usually referred to as spikes, clipping, overflow or just plain glitches. In certain cases, these errors can cause severe difficulty in seismic data processing, particularly when data-dependent processes like spiking deconvolution are applied.

The proposed method uses a two-sided interpolation operator derived by using a least-squared error criterion. The operator is computed by using the autocorrelation of the data themselves and is designed to estimate one or more data samples given a number of samples on either side of the "bad" samples. A thresholding algorithm compares the estimates for each sample with their actual value and corrects only those samples severely in error.

A study using synthetic data with random errors confirmed that

a) The errors were successfully removed.
b) The original (correct) data were unchanged.
c) The method starts to break down in the presence of too many errors.

Examples of different types of errors found in field-recorded seismic data are presented. The results of applying spiking deconvolution to the corrected data showed considerable improvement over the original data processed with the same deconvolution parameters.

INTRODUCTION

Many errors that occur in field recording go undetected during seismic data-processing. Many others are noted but are ignored, because it is felt, quite justifiably, that they are the types of error that will be "filtered out" or "stacked out." We are concerned here with the types of error that show up in the final stack and cannot be removed by some of the more conventional methods such as muting or filtering. In fact, our work on the method presented below resulted from observations of the effect of deconvolution on some clipped field data. Each clipped peak or trough was split into two or sometimes three separate peaks and troughs. In addition, the amplitudes of these "extra" events were considerably higher than the original clipped events. As a result, the final stack contained significant errors.

As our work progressed, we found that other types of field recording error gave rise to even more severe errors after deconvolution than those that had occurred with the clipped data. In many cases the only solution available was to omit those traces with errors, or to mute out the errors.

Before proceeding further it is worth pointing out that the vast majority of seismic data have very few problems of this nature. It is certainly not the intent of this paper to suggest that errors of this type occur to the extent that we should be questioning field-recording procedures. However, in spite of the best efforts of all concerned, these errors can and do arise in practice and, accordingly, a satisfactory method of dealing with such problems is desirable. This paper will present such a method applicable to digitally recorded field data.

COMMON ERROR TYPES

Before outlining the method, it is useful to consider several types of real data error and the features that characterize them. Figure 1 shows three types — clipping, spike and gain errors.

The clipping error has the effect of "squaring off" peaks or troughs or, put another way, the samples near the peak or trough are held to some fixed value. Thus there may be one or more samples in error, depending on the wavelength of the data. For example, a 5Hz wave, clipped at about half height and sampled at 2 ms

[1]Presented at the CSEG National Convention, Calgary, Alberta, April 6, 1982.
[2]Veritas Seismic Processors Ltd., 200, 615-3rd Avenue S.W., Calgary, Alberta T2P 0G6.

rate could have as many as 30 consecutive samples in error. In real data, such as in Figure 1, most of the errors occurred at four or five consecutive samples. The size of the error will of course vary from one sample to the next, with the largest at the point where the peak or trough should be.

The spike error is characterized by one sample with an "unexpected" large value. In theory, any sample value consistent with the field filter settings should be expected, but in practice we learn to recognize that certain data samples are unreasonable. The size of the error in such cases is usually many times the value of neighbouring samples. Figure 1 shows such a case.

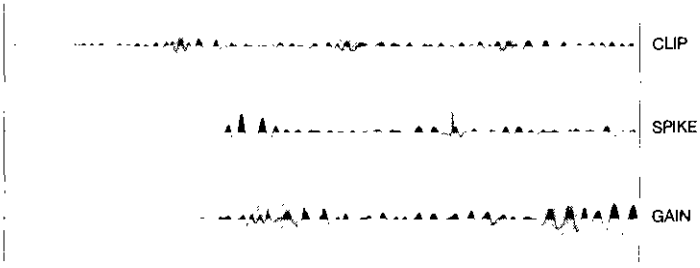

Fig. 1. Three types of error that can occur in field-recorded data.

The "gain" error shown in Figure 1 arose from some data where, because of some instrument problem, the field binary gain information was not properly recorded. This caused some samples to be recovered with half or double their true values. In contrast to the spike case, most of these errors occurred at two or three consecutive samples, with large numbers of "good" samples between errors. The size of the error, of course, was either 50 or 100% of the recovered sample value.

In all these cases the errors have been detected because one or more samples had unexpected or unreasonable values. Our notion of what is (or is not) reasonable comes from a sort of "eyeball interpolation" where we attempt to fill in certain values that we feel are reasonable in light of the samples on either side of the error. The method chosen to fix these errors, therefore, should somehow interpolate samples based on the samples on either side of the error.

Possible Solutions

Figure 2 presents a summary of possible solutions to the problem.

The first solution, muting, will work to the extent that a geophysicist can identify and code the shot, trace number and time of each error in order to remove (set to zero) the offending samples. This is likely to be an excessively tedious process and in some cases (e.g., clipping) will cause a more severe problem than existed to start with.

METHOD	ADVANTAGES	DISADVANTAGES
MUTES	EASY APPLICATION	TEDIOUS
FILTER	EASY APPLICATION	DISGUISES PROBLEM
POLYNOMIAL INTERPOLATION	CALCULATE OPERATORS ONCE	LESS ACCURATE ON MANY CONSECUTIVE ERRORS
ESTIMATOR (MINIMUM ERROR ENERGY)	ESTIMATES ARE "TRUE" TO DATA CHARACTER	FREQUENT COMPUTATION OF OPERATORS

Fig. 2. Possible solutions to the error problem.

The second solution, filtering, is the current "catch-all" method. If errors are not too severe, then bandpass filtering will smooth them out to the extent that they become negligible, particularly when stacked with other "good" samples. The method is certainly easy to apply and in fact, in many cases is simply a data-processing routine that would be carried out whether errors were present or not. However, there are cases when filtering is not entirely satisfactory, as shown in Figures 3, 4 and 5.

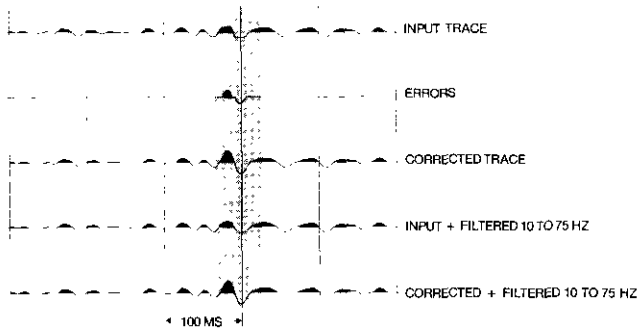

Fig. 3. Bandpass on clip errors.

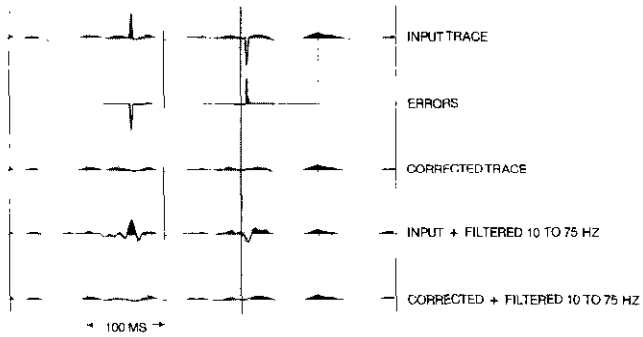

Fig. 4. Bandpass on spike errors.

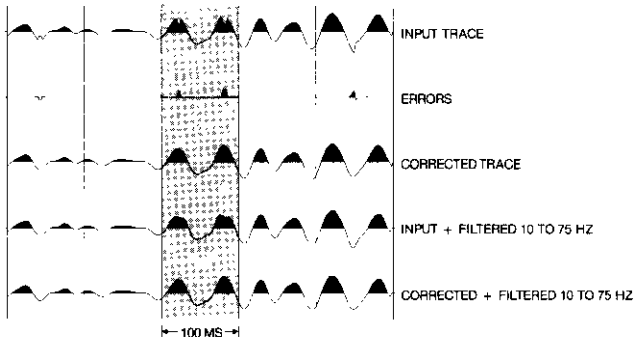

Fig. 5. Bandpass on gain errors.

In each case there is an input trace with a problem (clipping, spike and gain). The method to be described below was used to correct the data. Then a bandpass filter was applied to the original and the corrected trace. In all cases, the bandpass filter has no effect on the corrected trace. The effect on the bad samples is to "smooth out" the error, though not to the extent that the correction method did. In particular, on the gain errors in Figure 5, the bandpass caused a dimple in two peaks — a feature that might lead to an interpretation of two or more reflecting horizons at that time, whereas the correction method simply filled out the peaks. Thus the bandpass filter, although going some way toward solving the problem of sample errors, is clearly unsatisfactory in certain cases.

The third solution, polynomial interpolation, is at first sight a reasonable answer to our problem. In this method, the interpolated samples are taken to be the values of some polynomial function drawn through samples on either side of the error in question. The simplest type of polynomial is first-order, where a straight line is drawn from the last good sample before the sample(s) in error to the first good sample after the error. It is not necessary to compute a polynomial every time we have an error. It turns out that we can accomplish the effect of polynomial interpolation by constructing operators which are then applied to the data. These operators are completely independent of the data and need only be computed once before being applied to any data. The simplest operators will look like (½, 0, ½) which is an operator of length 3 corresponding to a first-order polynomial. A seventh-order polynomial operator will look like ($-\frac{1}{70}$, $\frac{4}{35}$, $-\frac{2}{5}$, $\frac{4}{5}$, 0 $\frac{4}{5}$, $-\frac{2}{5}$, $\frac{4}{35}$, $-\frac{1}{70}$).

However, we need to interpolate one or more consecutive samples. Thus if we have to correct an error of three consecutive samples, we need three different operators, one for each of the three samples in error. This point will be discussed further below. Polynomial interpolation can often be remarkably accurate close to points through which the polynomial passes, and can just as often be remarkably inaccurate at some distance from the "control" points. For this reason we add an extra level of complexity and use the fourth solution method, based on minimum error energy. The only disadvantage of the minimum error energy approach as compared with the polynomial approach is that the minimum error energy operators are data-dependent and therefore must be recomputed often.

Appendix I (after Nyman, 1977) discusses the derivation of minimum error energy operators and they are shown to depend solely on the data autocorrelation, the number of points of good data to use (corresponding to some equivalent polynomial order), and the number of consecutive samples to interpolate. Because the autocorrelation of seismic data depends largely on the average wavelet, it was felt that it would be necessary to recompute the minimum error energy operators only at every shot. The extra time required to compute the error operators at every shot is, in fact, small compared with the time required for the other computations common to both interpolation methods.

Comparison of Polynomial and Minimum Energy Operators

Figure 6 shows a comparison between a polynomial operator and a minimum error energy operator. Both operators were used to reconstruct seven consecutive samples of a 30 Hz Ricker wavelet sampled at 2 ms rate. Both operators, of course, used the same input data in reconstructing (interpolating) the samples, and the predominant data frequencies are well below the Nyquist or aliasing frequency. (Polynomials usually fare badly on frequencies close to the Nyquist — Nyman, 1977.)

The operators were designed to interpolate the midpoint of a 7-point error (*i.e.*, 7 consecutive erroneous samples). The midpoint was chosen because either method would have its poorest performance there.

The two operators, although similar in appearance, have significant differences. The minimum error energy operators tend to weight the "good" samples close to the error a little more heavily than a polynomial does.

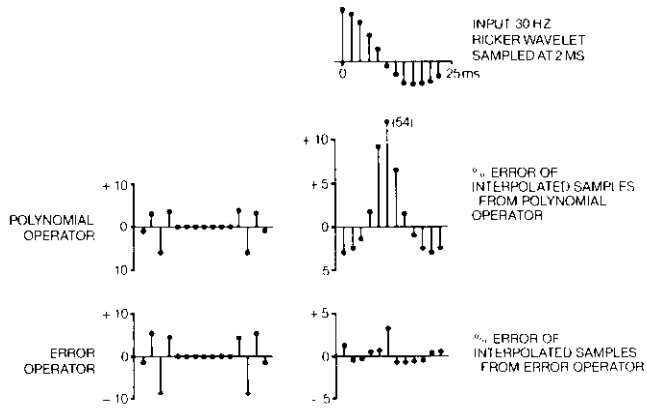

Fig. 6. Comparison of polynomial and minimum error energy operators.

Although this behaviour may not be true in general, it was observed in all seismic data cases.

Note that the input trace in Figure 6 shows correct samples; however, the operators were applied as if there was a 7-point error in the input data. The interpolated values, therefore, could be equal to the correct (input) values only if the operators were perfect.

The "errors" shown in Figure 6 are the percentage differences between the interpolated or estimated samples and the correct (input) values. As might be expected, the largest errors show up on the smallest samples. The error plots, however, indicate that the minimum error energy operator is in general more accurate than the polynomial operator. Excluding the one anomalous result (54% in the polynomial error), the average error for the minimum error energy operator is 0.6%, while the average error for the polynomial operator is 3.2%.

These results are, of course, data-dependent — on the autocorrelation of the Ricker wavelet, in fact. With different data having higher frequencies, and where there are errors of more than 7 consecutive samples, the two methods would show more divergent results.

Minimum Error Energy Operators

Appendix I gives the complete derivation of these operators — sometimes also called least-squared error operators. Figure 7 shows an example of some typical operators applied to a ramp. Here we see three sets of operators.

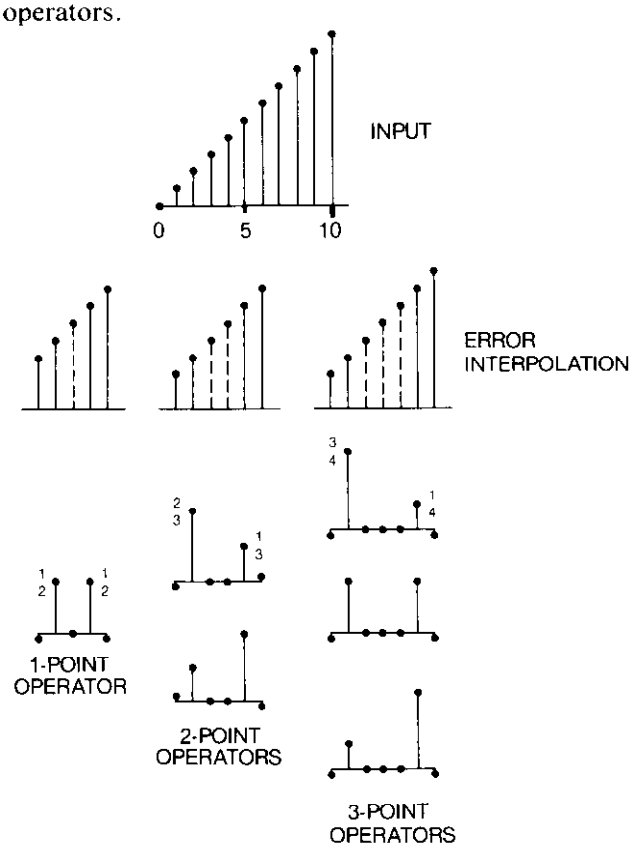

Fig. 7. Least-squared error operators.

The first set consists of one operator designed to interpolate or estimate a data sample using 2 data values on the left and 2 on the right of the error. The weights of the operator are (-0.0001, 0.5, 0.0, 0.5, -0.0001). The central zero shows the position of the point we are estimating. The fact that we have values like -0.0001 (the exact answer should be 0!) at the operator extremities is due mostly to not using enough time points to construct the autocorrelation of the ramp function properly.

The second set consists of two operators. The first of the two operators is designed to estimate the left-hand point of a two-point error. The second of the two operators is designed to estimate the right-hand point of a two-point error. The two operators are (-0.0001, 0.6666, 0.0, 0.0, 0.3333, -0.0001) and (-0.0001, 0.333, 0.0, 0.0, 0.6666, -0.0001).

There are now two zeros in the middle of the operator, indicating that there are two consecutive samples in error.

The third set consists of three operators. The first will estimate the left-hand point of a three-point error, the second the central point of a three-point error, and the third the right-hand point of a three-point error. The three operators are (-0.0001, 0.75, 0.0, 0.0, 0.0, 0.25, -0.0001) (-0.0001, 0.5, 0.0, 0.0, 0.0, 0.5, -0.0001) and (-0.0001, 0.25, 0.0, 0.0, 0.0, 0.75, -0.0001). There are three zeros in the middle of the operator, indicating that there are three consecutive samples to be estimated.

There is a certain symmetry about these operators. The operators that estimate the central point of an odd sequence of samples are completely symmetrical. All other operators can be grouped in pairs that are mirror images of each other. For example, the operator to estimate the second point of a seven-point error is the mirror image of the operator to estimate the sixth point of a seven-point error.

Error Detection and Correction Method

We shall now describe the full error detection and correction method using the minimum error energy operators. In Figure 8 is shown a Ricker wavelet with some sample errors near the peak. The error was created artificially by dividing three consecutive samples by 2 (simulating a gain error on real seismic data). The autocorrelation shown is the average of the autocorrelation of the displayed (erroneous) Ricker wavelet and nine identical autocorrelations of a correct Ricker wavelet — simulating a real data case where we do not have enough errors to seriously impair the shape of the average wavelet autocorrelation. Six least-squared error operators are also shown — one "1-point error" operator, two "2-point error" operators and three "3-point error" operators. In Figure 9 we show the result of convolving each of these six operators with

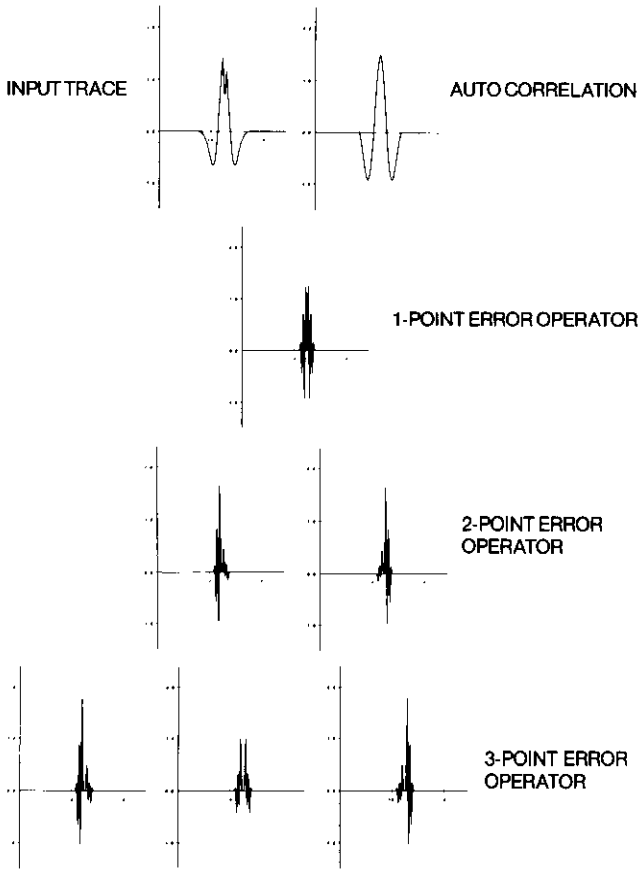

Fig. 8. Error operators.

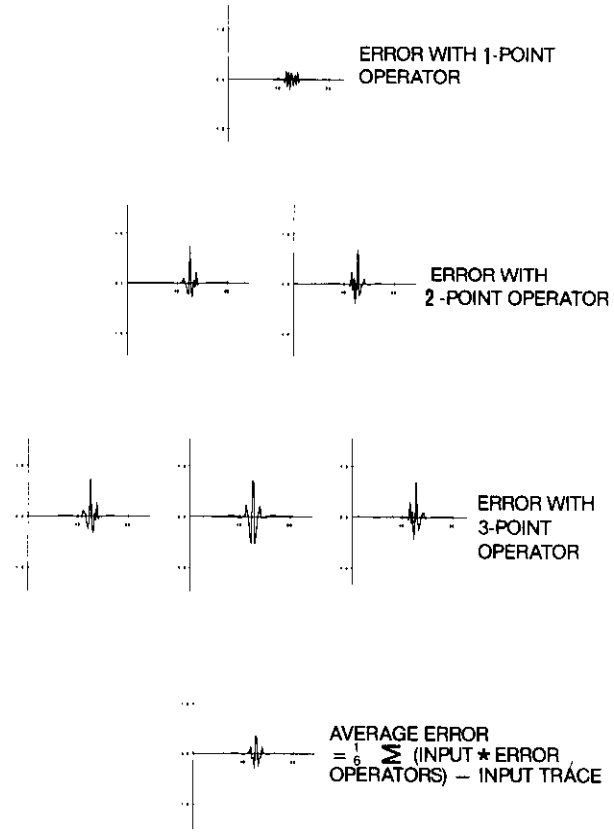

Fig. 9. Error components.

the input and subtracting the input. Finally, at the bottom of Figure 9 we show the average of the six error functions.

In real data it is impossible to know before hand how many consecutive samples there are in an error. Also, it is clearly undesirable to convolve the operators with bad data samples. Therefore, all cases are considered equally likely and the final average error function is used as a guide to the actual errors that exist in the data. First, each input data point is considered to be a single error and is estimated from the neighbouring samples by using the "1-point error" operator. Then samples of an error function are calculated as equal to the difference between the actual (input) samples and the estimates. The collection of all these samples is plotted as an error "trace" in Figure 9.

Second, each input data point is considered to be the left-hand point of a two-point error and is estimated from the neighbouring samples by using the first of the two "2-point" error operators. Then the error trace between the input and this estimate is calculated and displayed. Now, each input data point is considered to be the right-hand point of a two-point error and is estimated from the neighbouring samples by using the second of the two "2-point error" operators. The error trace is calculated and displayed.

Third, each input data point is considered to be the left-hand, central and right points in turn, of a three-point error and is estimated from the neighbouring samples by using the three "3-point error" operators. The three error traces are calculated and displayed. Finally, the average of these six error traces is calculated and displayed.

This rather exhaustive approach solves two difficulties:
i) Using an operator designed to estimate longer errors than are actually present leads to some inaccuracies in the estimates.
ii) Using bad data when estimating shorter errors than are actually present leads to inaccuracies.

The average error trace is now thresholded, by using a percentage of the mean of the input trace to arrive at the "error position" trace. This trace simply indicates which samples of the average error exceeded the threshold. The success of this error-detection method is due to the fact that the estimates are reasonably accurate when the operators are applied to "good" data (see Fig. 6). In this example it turned out that one three-point error (*i.e.*, three consecutive samples) was identified. Now the three "3-point" error operators were used in turn to construct the data estimates at the error position. The final corrected trace may be thought of as the sum of the input and the final error (Fig. 10).

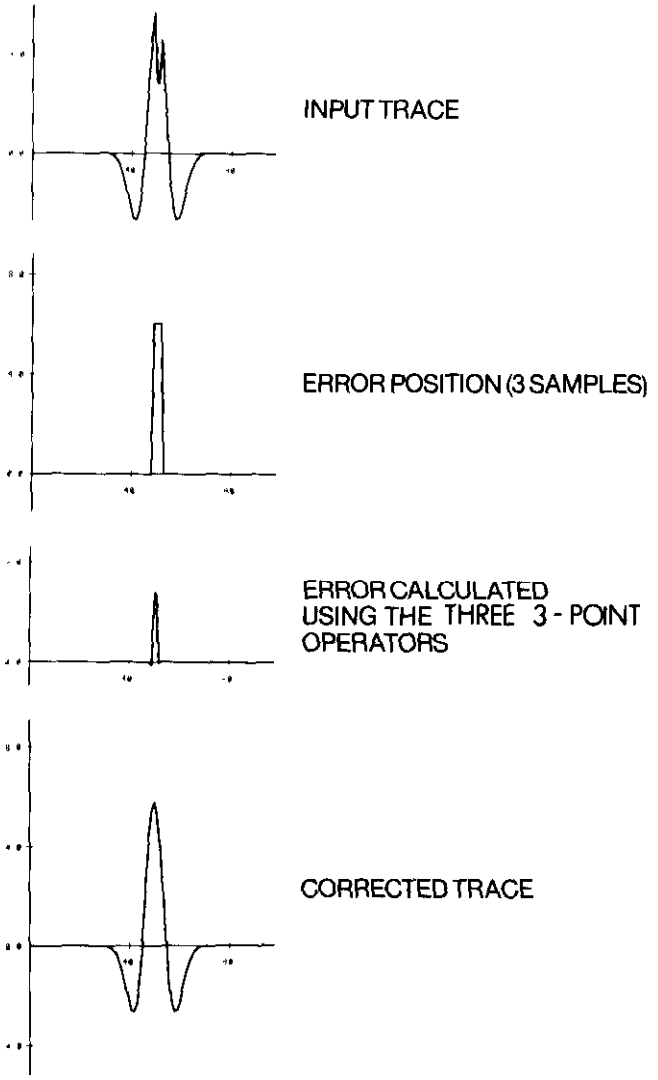

Fig. 10. Error correction.

Synthetic Data - Results

In this section we shall discuss the problems that arise when some sort of deconvolution is applied to seismic data with errors present. In some cases the result is almost tolerable; in others it is disastrous.

In Figure 11 we show a clipping error on one of ten identical Ricker wavelets. The second and third traces show the final error and the corrected trace respectively as computed by the method outlined above. The fourth and fifth traces show the results of applying conventional spiking deconvolution to the input and corrected trace respectively. The deconvolution operator's action on the Ricker wavelet can be seen to be the same everywhere except at the clipped data. Here the operator has caused some high-amplitude, high-frequency anomaly. Thus, although the input data look fairly reasonable, the small amount of clipping shown has produced a very large error after deconvolution. In contrast, the deconvolved corrected trace shows a barely perceptible change at the same position.

Fig. 11. Synthetic data — clip errors (1 clip/trace, 6 traces/shot).

Figure 12 shows a single spike error. The autocorrelation was computed from five "good" traces and this one "bad" trace. Thus the operators were accurate and have almost perfectly corrected the trace. The spiking deconvolution again has magnified the effects of the error on the input trace.

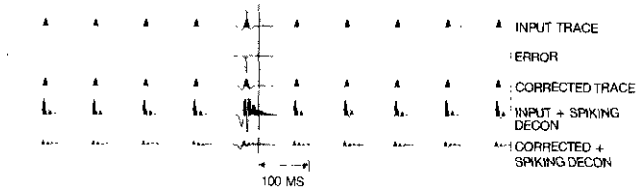

Fig. 12. Synthetic data — spike errors (1 spike/shot, 6 traces/shot).

Figure 13 shows a gain error of three consecutive samples on one Ricker wavelet out of ten. Although the corrected trace looks reasonable, the deconvolution has picked up some difference and magnified it, causing a small anomalous peak behind the first main peak.

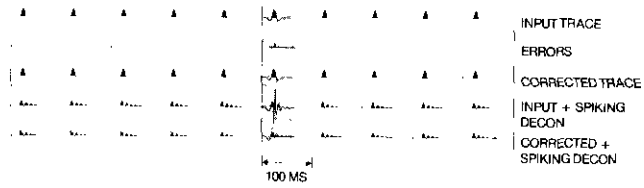

Fig. 13. Synthetic data — gain errors (1 error/trace, 6 traces/shot).

Figure 14 shows a simple spike error. The autocorrelation was computed from this trace. Thus Figure 12 has one spike every six traces, giving a reasonably accurate autocorrelation, whereas Figure 14 has one spike every trace, which gives rise to fairly significant differences. The difference in the autocorrelations and hence in the operators is easily seen in the final deconvolved trace. Whereas in Figure 12 the corrected trace was almost perfect, in Figure 14 the deconvolution is picking up on a small error and magnifying it as

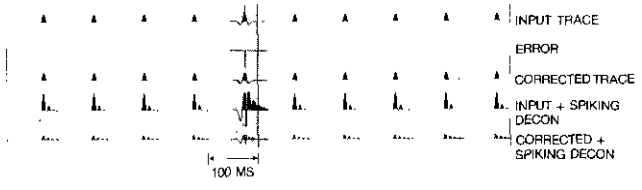

Fig. 14. Synthetic data — too many errors (1 spike/trace, 6 traces/shot).

shown. This example demonstrates how the method starts to break down in the presence of too many errors.

This synthetic data study demonstrated that:
i) The errors were successfully removed except where their number and magnitude affected the data autocorrelation.
ii) The data were unchanged except at the error positions.
iii) In all cases, the results of deconvolving the corrected trace are more accurate than the results of deconvolving the original input trace.

Real Data - Examples and Results

The following examples show actual seismic data from three different areas. In all three cases, the errors in the data were present on the tapes recorded in the field.

In all figures showing the effects of seismic processing (*e.g.*, deconvolution), the same processing parameters were used before and after correcting the errors.

A trace with spike errors is shown in Figure 15. The spikes were believed to be caused by random electrical noise. Each spike is one sample only. Taking the line as a whole, there were about 8 spikes per 48-trace shot. When zero-phase deconvolution is applied to the input trace, the result is greatly distorted. When the spikes are corrected first (*i.e.*, before deconvolution), the deconvolved trace retains its true character and the individual events are seen more clearly. The same effect is seen for spiking deconvolution. In this case, the effect of the spikes is so severe that, in conventional processing, it would have been necessary to mute the error or even kill the entire trace.

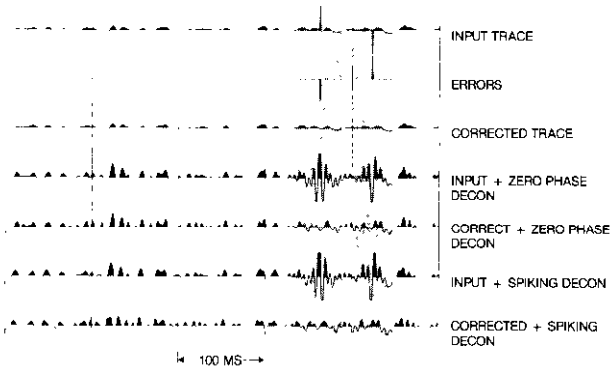

Fig. 15. Deconvolution on spike errors.

Two traces with gain errors are shown in Figures 16 and 17. These traces are from the same seismic line. The errors were caused by an equipment malfunction involving one bit in the binary gain code. Each erroneous sample is higher or lower than its correct value by a factor of two. The errors in this seismic line ranged in length from one to seven consective samples. Every seismic trace of the line had some errors, the average being about four per trace. Again, the errors cause great distortion when deconvolution is applied. When the errors are removed before deconvolution, the resolution is greatly improved. In this case, with errors on every trace, the only possible conventional remedy is bandpass filtering. It turned out in practice that the bandpass range required to produce a reasonable-looking stack was so restrictive as to render the stack unsuitable for any detailed interpretation.

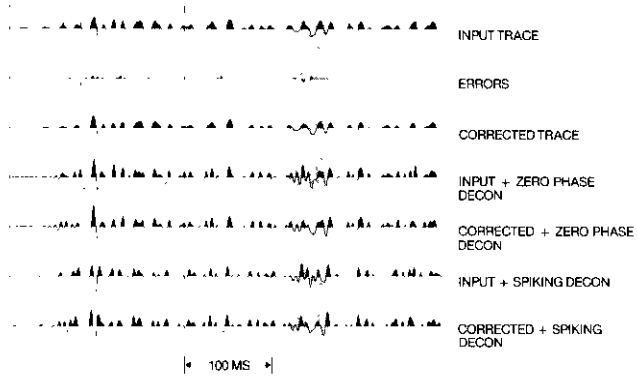

Fig. 16. Deconvolution on gain errors.

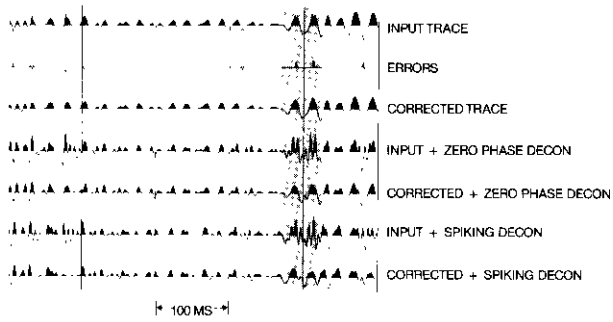

Fig. 17. Deconvolution on gain errors.

Part of a shot of clipped data is shown in Figure 18. The clipping occurred because the recording instruments were calibrated in the laboratory for a lighter charge of dynamite than was used in the field. Consequently, the dynamic range of the instruments could not accommodate all incoming energy, particularly in the region of the first arrivals and the first strong reflectors. The errors in this line ranged from three to over fifteen consecutive samples. Most traces had clipping in the first arrivals, and some on strong events down to about 1600 ms. The same data are shown again in Figure 19, after a number of seismic processes were applied. They show a slight improvement when the clipping is corrected first. It appears that clip errors are less detrimental than spike or gain errors. Even small errors, however, can cause important differences in the stacked data. Figures 20 and 21 show a portion of the stack made from the clipped data. There are significant differences between the stack of the original clipped data and that of the corrected data. Note

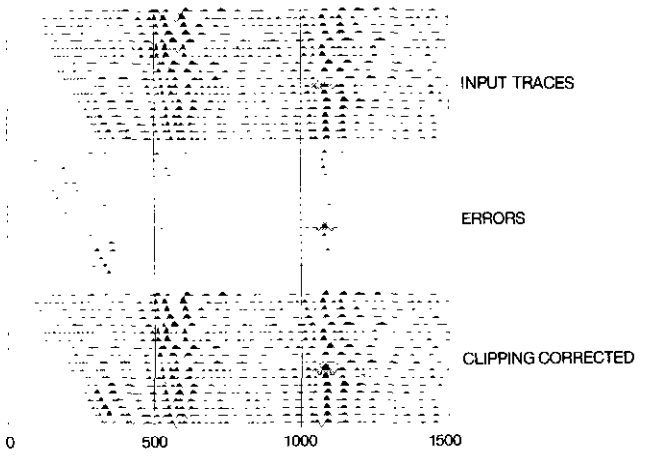

Fig. 18. Raw shot, with field-recorded gain and exponential scaling (Ate^{nt}) applied.

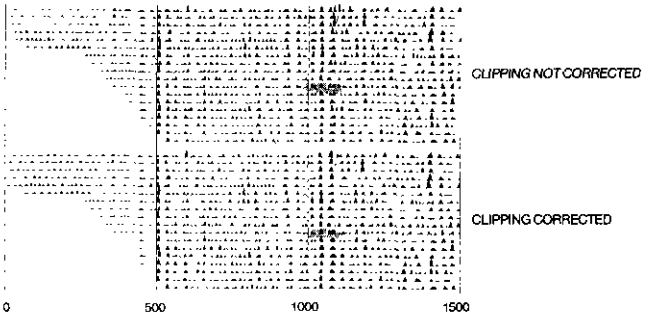

Fig. 19. Fully processed shot, with scaling, spiking decon, NMO, filter and LVL corrections applied.

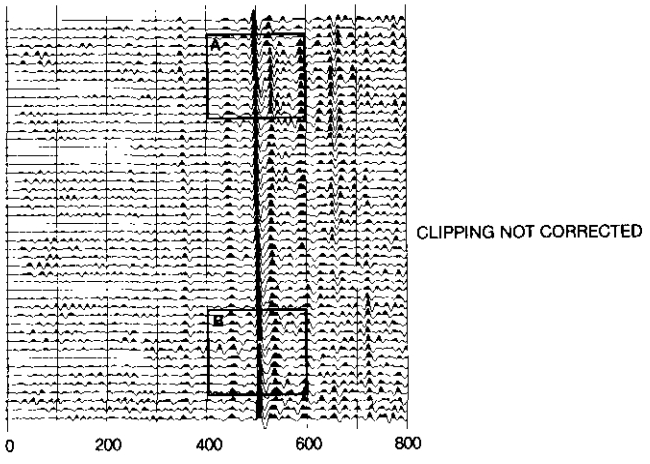

Fig. 20. Fully processed stack.

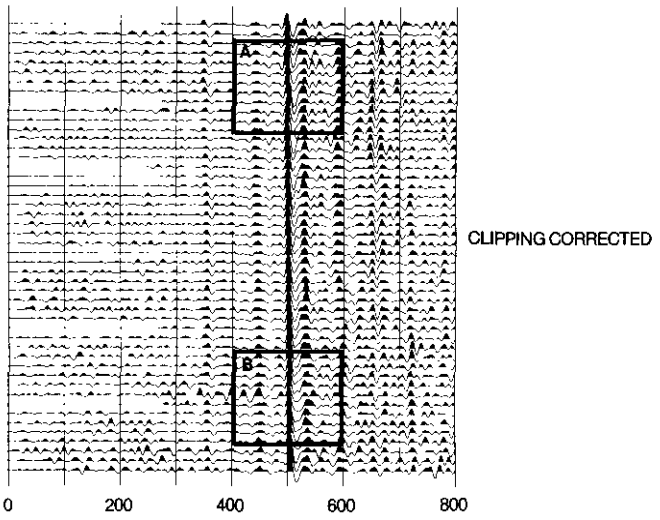

Fig. 21. Fully processed stack.

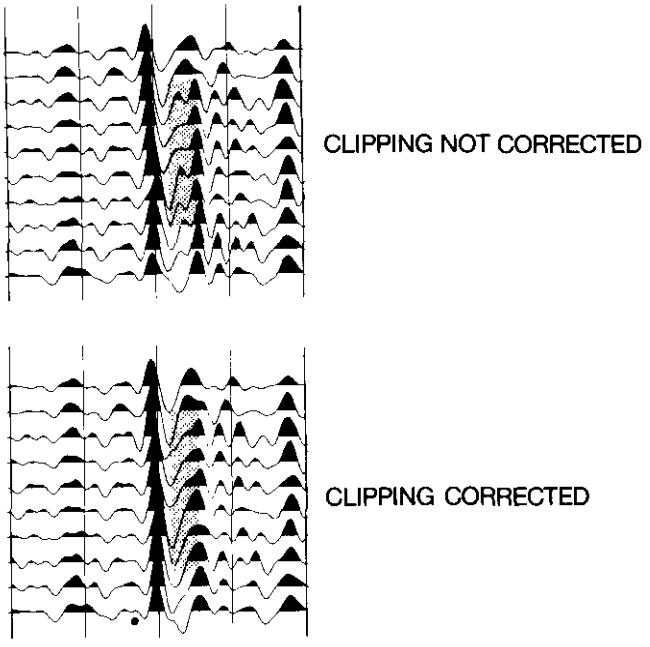

Fig. 22. Stack — inset A.

that both stacks were processed identically through the processes indicated on Figure 19.

Inset A from the stack is shown in Figure 22. On three of the stack "traces", a small anomalous peak is present. This anomaly, however, is actually a result of deconvolution: it does not occur when the clipping is corrected first. Furthermore, the small anomalous peaks each came from one or two input traces, despite having seven input traces at each stack trace position. The same thing occurs to a lesser extent in Inset B, Figure 23.

A deeper portion of the same stack is shown in Figure 24. When the clipping is not corrected, the second large peak tends to separate into a doublet, which could perhaps be mistakenly interpreted as some sort of stratigraphic anomaly. Again, this is due to deconvolving input data with errors present. The problem does not occur when the clipping is corrected first.

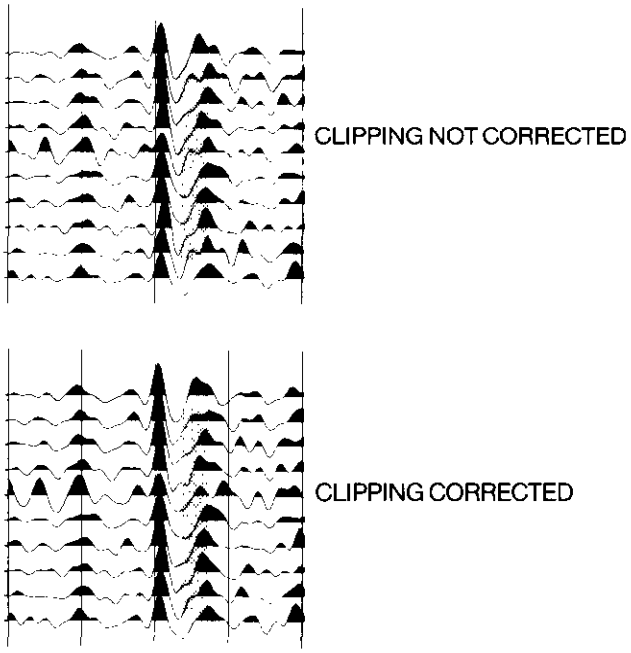

Fig. 23. Stack — inset B.

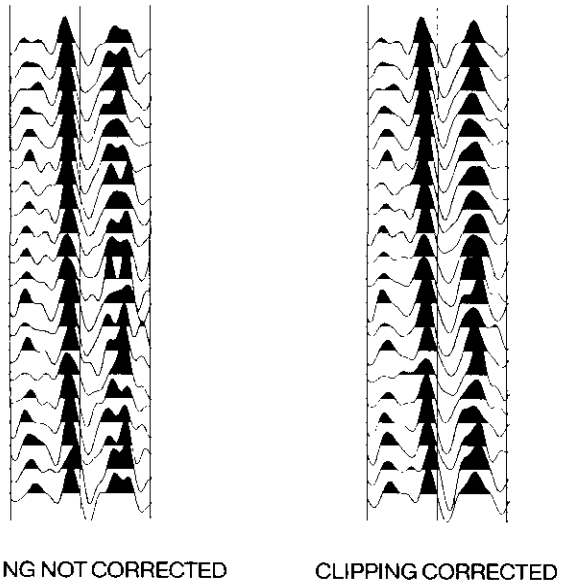

Fig. 24. Stack — deeper event.

CONCLUSIONS

Errors sometimes occur during field recording of seismic data. These errors are random in nature and usually affect several consecutive samples. When there *are* errors, they generally occur on most of the traces (and most of the shots) in the seismic line.

During the course of normal seismic processing, some processes such as bandpass filtering tend to lessen the effect of errors; others, such as deconvolution, amplify them. In some cases, part of the data might have to be discarded.

Consequently, a method was developed to correct the errors before other seismic processing. Of the two possible approaches to error correction, the minimum error energy method was chosen because of its higher accuracy compared with the polynomial interpolation method. In it, error operators are computed from the autocorrelation of the data, to minimize the error energy.

Errors are detected by estimating each sample, using the error operators, and comparing the estimate with the corresponding input sample. Then a thresholding algorithm identifies those samples that are actually in error. The error operators are sufficiently accurate that "good" samples are not mistakenly "corrected".

The results of applying deconvolution to the corrected data showed considerable improvement over the results of deconvolution on the input data.

This method, therefore, is a new and valuable tool for the processing geophysicist when faced with field recording errors of the types presented in this paper.

REFERENCES

Nyman, D., 1977, The interpolation error operator: time series error detection and correction: Geophysics, v. 42, p. 773-777.

APPENDIX

Let x_t be a given series $t = 0, 1, \ldots$

and ($d_{-k,i}$ and $d_{k,i}$) $k = 1, 2, \ldots m$

$i = 1, 2, \ldots n$

be a two-sided operator of length m to be applied to the data x_t.

Then the estimator (Interpolation or smoothing operator) of order (length) m and "multiplicity" n for the simultaneous estimation of n consecutive terms

$$x_t, x_{t+1}, x_{t+2}, \cdots x_{t+n-1}$$

is given by

$$\hat{x}_{t-1+i} = \sum_{k=1}^{m} d_{-k,i}\, x_{t-k} + d_{k,i}\, x_{t-1+n+k}$$

Error is $h_{t,i} = \hat{x}_{t-1+i} - x_{t-1+i}$

Minimizing the total interpolation error energy leads to

$$\frac{\delta}{\delta d_{k,i}} \left(\sum_{t} \sum_{i=1}^{n} h_{t,i}^{2} \right) = 0 \qquad \text{——————————} (1)$$

Define the jth lag of the autocorrelation of x_t as $r_j = \sum_t x_t x_{t+j}$

Equation (1) leads to

$$\left.\begin{aligned} r_{j+i-1} &= \sum_{k=1}^{m} \left(d_{-k,1}\, r_{k-j} + d_{k,i}\, r_{n-1+k+j}\right) \\ r_{n+j-i} &= \sum_{k=1}^{m} \left(d_{-k,i}\, r_{n-1+k+j} + d_{k,i}\, r_{k-j}\right) \end{aligned}\right] \begin{matrix} j = 1,m \\ i = 1,n \end{matrix}$$

-2 m.n. equations in 2 m.n. unknowns

It can easily be shown that $d_{k,i} = d_{-k,n+1-i}$

Thus we really have n/2 sets of 2m equations in 2m unknowns; *i.e.*, mn equations in mn unknowns.

$$\left.\begin{aligned} \sum_{k=1}^{m} d_{k,i}\, r_{n-1+k+j} + d_{k,n+1-i}\, r_{k-j} &= r_{j+i-1} \\ \sum_{k=1}^{m} d_{k,i}\, r_{k-j} + d_{k,n+1-i}\, r_{n-1+k+j} &= r_{n+j-i} \end{aligned}\right] \quad j = 1,m$$

$$\begin{bmatrix} r_{n+1} & r_{n+2} & r_{n+3} & \cdots & r_{n+m} & r_o & r_1 & r_2 & \cdots & r_{m-1} \\ r_{n+2} & r_{n+3} & r_{n+4} & \cdots & r_{n+m+1} & r_{-1} & r_o & r_1 & \cdots & r_{m-2} \\ r_{n+3} & r_{n+4} & r_{n+5} & \cdots & r_{n+m+2} & r_{-2} & r_{-1} & r_o & \cdots & r_{m-3} \\ \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot & \cdot & & \cdot \\ r_{n+m} & r_{n+m+1} & r_{n+m+2} & \cdots & r_{n+2m-1} & r_{-m+1} & r_{-m+2} & r_{-m+3} & \cdots & r_o \\ r_o & r_1 & r_2 & \cdots & r_{m-1} & r_{n+1} & r_{n+2} & r_{n+3} & \cdots & r_{n+m} \\ r_{-1} & r_o & r_1 & \cdots & r_{m-2} & r_{m+2} & r_{n+3} & r_{n+4} & \cdots & r_{n+m+1} \\ r_{-2} & r_{-1} & r_o & \cdots & r_{m-3} & r_{n+3} & r_{n+4} & r_{n+5} & \cdots & r_{n+m+2} \\ \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot & \cdot & & \cdot \\ r_{-m+1} & r_{-m+2} & r_{-m+3} & \cdots & r_o & r_{n+m} & r_{n+m+1} & r_{n+m+2} & & r_{n+2m-1} \end{bmatrix} * \begin{bmatrix} d_{1,i} \\ d_{2,i} \\ d_{3,i} \\ \cdot \\ \cdot \\ \cdot \\ d_{m,i} \\ \\ d_{1,n+1-i} \\ d_{2,n+1-i} \\ d_{3,n+1-i} \\ \cdot \\ \cdot \\ \cdot \\ d_{m,n+1-i} \end{bmatrix} = \begin{bmatrix} r_i \\ r_{i+1} \\ r_{i+2} \\ \cdot \\ \cdot \\ \cdot \\ r_{i+m-1} \\ \\ r_{n+1-i} \\ r_{n+2-i} \\ r_{n+3-i} \\ \cdot \\ \cdot \\ \cdot \\ r_{n+m-i} \end{bmatrix}$$

Matrix form of Normal Equations