

Mimic Circuit Simulation in Real Time

by

Virgilio A. Centeno

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

APPROVED:



A.G. Phadke, Chairman



S. Rahman



J. De La Ree Lopez

June 1988

Blacksburg, Virginia

2

10
5655
V855
1998
C467
C.2

Mimic Circuit Simulation in Real Time

by

Virgilio A. Centeno

A.G.Phadke, Chairman

Electrical Engineering

(ABSTRACT)

9/16/2014 15:03

An algorithm is derived for the removal of the DC offset from a faulted current signal using a microprocessor sampling on real time. The algorithm is to be used instead of the analog Mimic circuit in distance computer relaying. Four variations of the algorithm were derived and tested to determine the best compromise between time response and noise sensitivity. The relay hardware is taken into consideration for the derivations to avoid adding any hardware to the relay. The graphical results of the test run in a analog simulator at the Virginia Tech Power Systems Laboratory are presented. Faults at different voltage angles were performed to determine the algorithm's performance at different levels of DC offset. From the graphical response obtained from the test and taking into consideration hardware and software limitations a preferred algorithm is selected with a good compromise between time response and noise sensitivity.

Acknowledgements

I am deeply indebted to Dr. A.G.Phadke for the help and guidance he has given me. I wish to thank Dr. S.Rahman and Dr. J.De La Ree for serving on my advisory committee.

Table of Contents

- CHAPTER 1** 1
- INTRODUCTION 1

- CHAPTER 2** 7
- ALGORITHM DERIVATION 7
 - 2.1 MIMIC CIRCUIT 7
 - 2.2 DISCRETE TIME ALGORITHM 10
 - 2.3 DESIRED FILTERING FACTORS 16

- CHAPTER 3** 18
- HARDWARE AND SOFTWARE CONFIGURATION 18
 - 3.1. RELAYING HARDWARE 19
 - 3.1.1 Analog Input Circuit 19
 - 3.1.2 Analog to Digital Converter 21
 - 3.1.3 Microprocessor Unit 23
 - 3.2 SOFTWARE CONSIDERATIONS 24
 - 3.2.1 Sampling Rate 24

3.2.2 Arithmetic Considerations	26
CHAPTER 4	28
EXPERIMENTAL RESULTS	28
4.1 TESTING HARDWARE	29
4.1.1 Relay Hardware	29
4.1.2 Simulator Hardware	29
4.2 SELECTED ALGORITHMS	31
4.2.1 3 Sample Algorithm	31
4.2.2 6 Sample Algorithm	32
4.2.3 3 Samples With Average	32
4.2.4 3 Samples Average to 6 Samples	33
4.3 TESTS	33
CHAPTER 5	61
CONCLUSION	61
5.1 PREFERRED ALGORITHM	61
5.2 ADVANTAGES AND DISADVANTAGES	63
5.3 FUTURE IMPROVEMENTS	65
5.4 CONCLUSION	66
APPENDIX A	67
A.1 6 Sample Algorithm Derivation	67
A.2 Three Sample with Average	69
BILIOGRAPHY	72

VITA 74

List of Illustrations

Figure 1.	Relay zones of protection.	2
Figure 2.	DC offset on current waveform.	4
Figure 3.	Mimic circuit on CT secondary.	8
Figure 4.	Parts of a computer relay	20
Figure 5.	Change from 3 sample average to six sample algorithm	34
Figure 6.	Simulated Transmission Line System.	36
Figure 7.	Input signal 0 degrees fault.	40
Figure 8.	Output analog mimic circuit 0 degrees fault.	40
Figure 9.	Output 3 sample algorithm 0 degrees fault	41
Figure 10.	Output 6 samples algorithm 0 degrees fault	41
Figure 11.	Output 3 samples average 0 degrees fault	42
Figure 12.	Output 3 samples average to 6 samples 0 degrees fault	42
Figure 13.	Input signal 90 degrees fault.	43
Figure 14.	Output analog mimic circuit 90 degrees fault.	43
Figure 15.	Output 3 sample algorithm 90 degrees fault	44
Figure 16.	Output 6 samples algorithm 90 degrees fault	44
Figure 17.	Output 3 samples average 90 degrees fault	45

Figure 18. Output 3 samples average to 6 samples 90 degrees fault 45

Figure 19. Input signal 135 degrees fault. 46

Figure 20. Output analog mimic circuit 135 degrees fault. 46

Figure 21. Output 3 sample algorithm 135 degrees fault 47

Figure 22. Output 6 samples algorithm 135 degrees fault 47

Figure 23. Output 3 samples average 135 degrees fault 48

Figure 24. Output 3 samples average to 6 samples 135 degrees fault 48

Figure 25. Input signal 180 degrees fault. 49

Figure 26. Output analog mimic circuit 180 degrees fault. 49

Figure 27. Output 3 sample algorithm 180 degrees fault 50

Figure 28. Output 6 samples algorithm 180 degrees fault 50

Figure 29. Output 3 samples average 180 degrees fault 51

Figure 30. Output 3 samples average 180 degrees fault 51

Figure 31. Input signal 225 degrees fault. 52

Figure 32. Output analog mimic circuit 225 degrees fault. 52

Figure 33. Output 3 sample algorithm 225 degrees fault 53

Figure 34. Output 6 samples algorithm 225 degrees fault 53

Figure 35. Output 3 samples average 225 degrees fault 54

Figure 36. Output 3 samples average to 6 samples 225 degrees fault 54

Figure 37. Input signal 270 degrees fault. 55

Figure 38. Output analog mimic circuit 270 degrees fault. 55

Figure 39. Output 3 sample algorithm 270 degrees fault 56

Figure 40. Output 6 samples algorithm 270 degrees fault 56

Figure 41. Output 3 samples average 270 degrees fault 57
Figure 42. Output 3 samples average to 6 samples 270 degrees fault 57
Figure 43. Input signal 315 degrees fault. 58
Figure 44. Output analog mimic circuit 315 degrees fault. 58
Figure 45. Output 3 sample algorithm 315 degrees fault 59
Figure 46. Output 6 samples algorithm 315 degrees fault 59
Figure 47. Output 3 samples average 315 degrees fault 60
Figure 48. Output 3 samples average to 6 samples 315 degrees fault 60

CHAPTER 1

INTRODUCTION

When a fault occurs in a power system its physical configuration changes leading to new steady state values of the system voltages and currents. The new steady state values are accompanied by transient components superimposed on the current and voltage waveforms. One of these transient components is an exponentially decaying DC offset in the currents caused by the sudden change of the current through the impedance of the system.

In a transmission line the voltage to current ratio of the line determines the fault location with respect to the zones of protection of an impedance relay (Figure 1). Zone one is a direct trip zone, while zone two overreaches the remote line terminal providing protection for the entire line. Zone three provides a remote back up function for neighboring circuits. To ensure relay selectivity a time delay

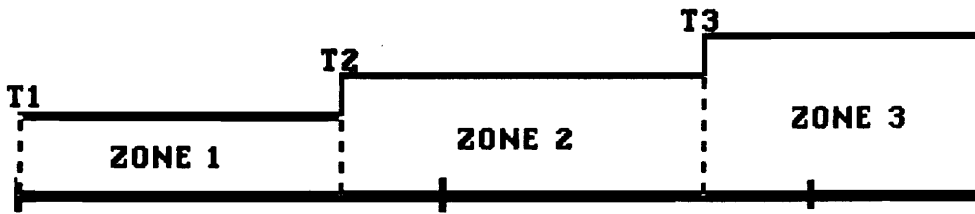


Figure 1. Relay zones of protection.

is introduced in the relay logic for zones two and three. Due to this delay the DC offset does not affect the fault impedance calculation for zones two and three since by then the DC offset has decayed considerably. This is not the case for faults in zone one, where the relay must act in about one cycle following the occurrence of the fault [1]. For the first few cycles the DC offset present in the current waveform may range from 0 to 100 percent of the peak value of the steady state fault current.

The presence of the DC offset in the current waveform reduces the magnitude of the impedance seen by the distance relay causing it to trip for faults outside the zone of protection. This phenomena is known as relay overreach and if uncorrected will cause unnecessary trips of the transmission line.

The exponentially decaying DC offset is characterized by its time constant and initial magnitude (Figure 2). The time constant is the same as the time constant of the faulted circuit which is usually a known value. The initial magnitude of the offset depends on the voltage phase angle at the instant of the fault. For angles around 90 degrees the DC offset is almost nonexistent, while for angles around zero degrees the DC offset is maximum making the total current about twice the magnitude of the peak of the new steady state current. It can be argued and proved statistically that most faults in a power system occur at voltage angles close to 90 degrees, when insulation breakdown is more likely. But faults, particularly during re-closing into faults, are possible at any voltage and do occur in transmission lines. The magnitude of the DC offset is thus a random value;

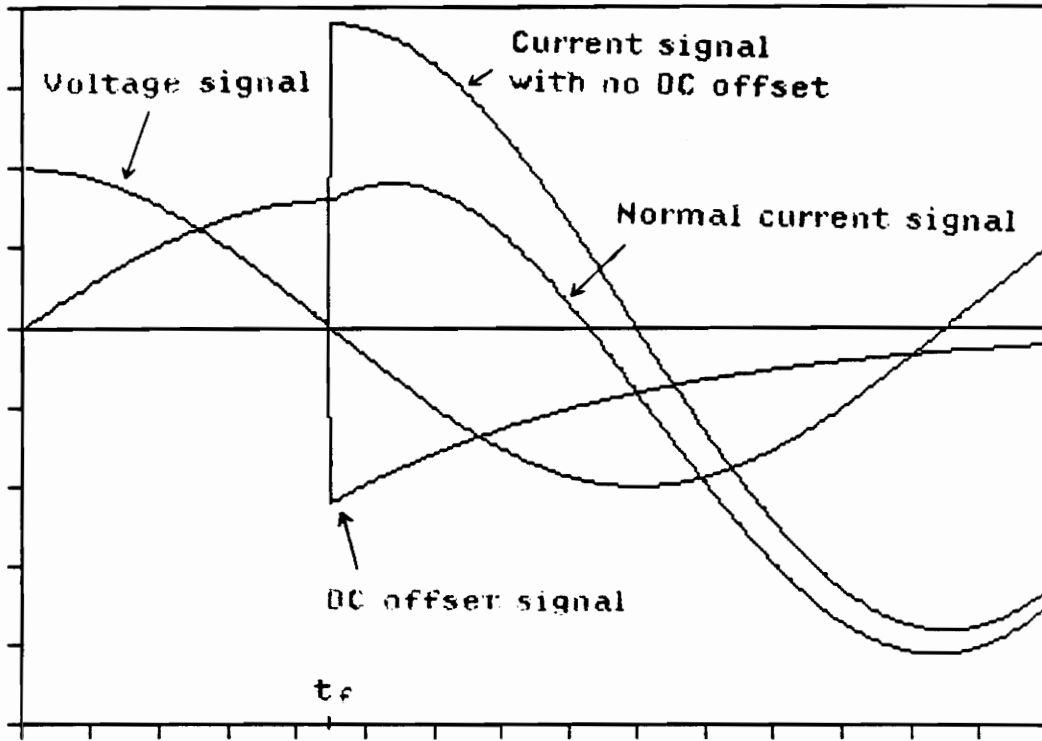


Figure 2. DC offset on current waveform.

therefore its influence on the distance relay performance is of extreme importance.

Traditionally, a mimic circuit in the secondary of the current transformer has been used to eliminate the DC offset from the current waveform. This mimic circuit is a replica impedance of the resistance and inductance of the faulted circuit in the secondary of the current transformer. When exactly matched to the transmission line the mimic circuit completely removes the DC offset from the secondary of the current transformer. But a perfect match is not always possible since the resistance and reactance of the faulted circuit are not exactly known. In addition the mimic circuit has the disadvantage of having to use analog components which must be adjusted when changes are made in the relay application. Also few settings for mimic circuit adjustments are possible in hardware.

With the recent developments in computer hardware the use of computers for relaying has become possible, offering several advantages over the electromagnetic and solid state relays [2]. Although few computer relays are commercially available, extensive research is being done by manufacturers, universities and utilities in this field and there is no doubt that relaying in the future will be performed by computers.

The most popular algorithms for computer relaying of high voltage transmission lines require that the decaying DC offset be removed from the current signal in order to achieve fractional cycle fault calculation [3,4,5]. A digital re-

removal of the decaying DC offset from the current signal will offer considerable advantages for the computer relay over the conventional mimic circuit. A digital mimic will have less time delay, can be more closely and easily matched to the transmission line, have no temperature or age dependency, be easily changed for new system configurations without any changes in the relay hardware and will not represent an added cost since it would be part of the computer relay.

This thesis presents the development and testing of an algorithm to remove the DC offset from the current signal with a microprocessor relay which uses real time samples of currents from a transmission line. The hardware used for computer relaying is briefly reviewed to determine its effects and limits in the digital removal of the exponentially decaying DC offset. Several variations of the algorithm are compared for accuracy and time delay. Experimental results obtained in the power system simulator at the Virginia Tech Power Systems Laboratory are presented and compared to select the most suitable algorithm. The advantages, disadvantages and possible improvements to the digital mimic algorithm are presented for possible use in computer relaying.

CHAPTER 2

ALGORITHM DERIVATION

2.1 MIMIC CIRCUIT

As mentioned in chapter one the mimic circuit is a replica of the impedance of the transmission line in the secondary of the current transformer. This circuit produces a voltage signal equivalent to the primary current but without a DC offset. (Figure 3)

To understand the operation of the mimic circuit consider the circuit in Figure 3. Let the fault current in the transmission line be denoted by i_f and be of the form:

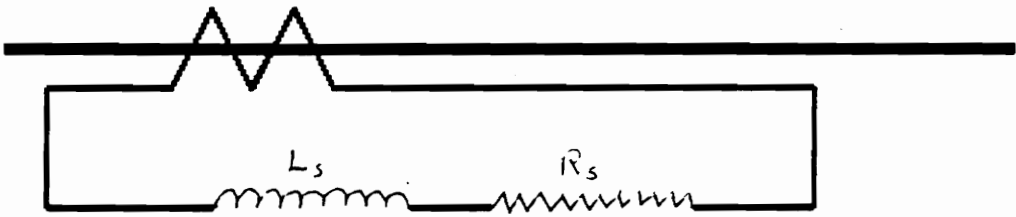


Figure 3. Mimic circuit on CT secondary.

$$i_f = I_m \sin(\omega t + \theta + \phi_l) - I_m \sin(\theta + \phi_l)e^{-\alpha t} \quad [2.1]$$

where:

$$I_m = V_m [R_l^2 + \omega L_l^2]^{-1/2} \quad [2.2]$$

$$\phi_l = \arctan(\omega L_l / R_l) \quad [2.3]$$

$$\alpha = R_l / L_l \quad [2.4]$$

and R_l and L_l are the corresponding resistance and impedance of the line.

The secondary voltage v_s is of the form:

$$\begin{aligned} v_s &= i_f Z_s \\ &= I_m Z_s \sin(\omega t + \theta + \phi_l + \phi_s) + I_m (R_s + d \frac{L_s}{dt}) \sin(\theta + \phi_l) e^{-\alpha t} \\ &= I_m Z_s \sin(\omega t + \theta + \phi_l + \phi_s) + I_m (R_s - \alpha L_s) \sin(\theta + \phi_l) e^{-\alpha t} \end{aligned} \quad [2.5]$$

The exponentially decaying component will be zero if:

$$R_s - \alpha L_s = 0 \quad [2.6]$$

but from equation 2.4

$$\alpha = R_l / L_l$$

which means that for the DC offset to be zero:

$$\frac{R_s}{L_s} = \frac{R_l}{L_l} \quad [2.7]$$

The impedance ratios of the primary and secondary circuit must be equal to make the DC offset zero. Therefore a resistance in series with an inductance in the secondary of the transformer filters the DC component if their ratio is the same as the transmission line impedance ratio.

2.2 DISCRETE TIME ALGORITHM

Let us consider the input and desired output of a digital mimic. The input current in a transmission line after a fault is of the form:

$$I = A \sin(\omega t + \phi) + B e^{-\alpha t} \quad [2.8]$$

where:

$$A = I_m$$

$$B = I_m \sin(\theta + \phi_l)$$

$$\phi = \theta + \phi_l$$

Decomposing the sine term into its sine and cosine terms the

following expression is obtained:

$$\begin{aligned}
 I &= A \sin \omega t \cos \phi + A \cos \omega t \sin \phi + B e^{-\alpha t} \\
 &= A_1 \sin \omega t + A_2 \cos \omega t + B e^{-\alpha t}
 \end{aligned}
 \tag{2.9}$$

where:

$$A_1 = A \cos \phi \quad \text{and} \quad A_2 = A \sin \phi$$

In discrete time form, starting at $t = 0$ with a sampling period T the expression becomes:

$$\begin{aligned}
 y_0 &= 0 + A_2 + B \\
 y_1 &= A_1 \sin(\omega T) + A_2 \cos(\omega T) + B\beta \\
 y_2 &= A_1 \sin(\omega 2T) + A_2 \cos(\omega 2T) + B\beta^2 \\
 &\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\
 y_n &= A_1 \sin(\omega nT) + A_2 \cos(\omega nT) + B\beta^n
 \end{aligned}$$

where:

$$\beta = e^{-\alpha T} \tag{2.10}$$

In matrix notation:

$$Y = \begin{bmatrix} 0 & 1 & 1 \\ \sin(\omega T) & \cos(\omega T) & \beta \\ \sin(2\omega T) & \cos(2\omega T) & \beta^2 \\ \cdot & \cdot & \cdot \\ \sin(n\omega T) & \cos(n\omega T) & \beta^n \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ B \end{bmatrix}$$

or

$$Y = JA \quad [2.11]$$

solving for A

$$A = (J^T J)^{-1} J^T Y \quad [2.12]$$

Where J is a nx3 matrix and n is the number of samples with a minimum value of 3 needed to compute the unknowns: A_1 , A_2 and B . A larger number of measurements will improve the accuracy of the results but will increase the computation window and decrease the response time of the algorithm as more computations are needed when the number of samples increases.

The desired output is a signal which is proportional to the input current but has no DC offset. Therefore the desired voltage on the secondary side is of the form:

$$\begin{aligned}
v_s &= I_m C \sin(\omega t + \phi + \phi_s) \\
&= I_m C \sin(\omega t + \phi_s) \cos \phi + I_m C \cos(\omega t + \phi_s) \sin \phi \quad [2.13] \\
&= CA_1 \sin(\omega t + \phi_s) + CA_2 \cos(\omega t + \phi_s)
\end{aligned}$$

Where a phase shift ϕ_s and a gain of C are assumed in the output signal. Using discrete time notation starting at $t = 0$ and with a sample frequency of $1/T$ the following equations are obtained:

$$\begin{aligned}
v_0 &= CA_1 \sin \phi + CA_2 \cos \phi \\
v_1 &= CA_1 \sin(\omega T + \phi) + CA_2 \cos(\omega T + \phi) \\
v_2 &= CA_1 \sin(2\omega T + \phi) + CA_2 \cos(2\omega T + \phi) \\
&\dots \dots \dots \\
v_n &= CA_1 \sin(n\omega T + \phi) + CA_2 \cos(n\omega T + \phi)
\end{aligned}$$

In matrix notation:

$$V = \begin{bmatrix} C \sin \phi & C \cos \phi \\ C \sin(\omega T + \phi) & C \cos(\omega T + \phi) \\ C \sin(2\omega T + \phi) & C \cos(2\omega T + \phi) \\ \dots & \dots \\ C \sin(n\omega T + \phi) & C \cos(n\omega T + \phi) \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$$

or

$$V = CA \quad [2.14]$$

Where A_1 and A_2 are the unknowns defined in equation 2.12. Now it is clear that

$$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ B \end{bmatrix}, \text{ or}$$

$$A = GA \quad [2.15]$$

therefore:

$$\begin{aligned} V &= CGA \\ V &= CG(J^T J)^{-1} J^T Y \\ V &= KY \end{aligned} \quad [2.16]$$

and

$$K = CG(J^T J)^{-1} J^T \quad [2.17]$$

The matrix K multiply by the input vector gives the desired output signal for n number of samples in the input vector.

For example, for $n=3$, a line impedance angle $\theta = 80$ degrees and a sampling rate of 720 Hz the following derivations are obtained:

$$C = \omega X/R = \tan 80 = 5.671$$

Therefore:

$$\alpha = R/L = \omega R/L = 66.474$$

For $f = 720$ Hz the period $T = 1/720$

$$\text{Then } e^{-\alpha T} = 0.9118$$

[2.18]

And:

$$Y = \begin{bmatrix} 0.000 & 1.000 & 1.000 \\ 0.500 & 0.866 & 0.912 \\ 0.866 & 0.500 & 0.831 \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ B \end{bmatrix}$$

$$(J^T J)^{-1} J^T = \begin{bmatrix} -2.095 & 2.629 & -0.363 \\ -2.967 & 6.871 & -3.967 \\ 3.967 & -6.871 & 3.967 \end{bmatrix}$$

$$CG = \begin{bmatrix} 376.991 & 0.000 & 0.000 \\ 326.484 & -188.496 & 0.000 \\ 188.496 & -326.484 & 0.000 \end{bmatrix}$$

Multiplying this matrix by an input vector of three consecutive samples an output vector corresponding to the three sampled inputs is obtained. The output vector has no DC offset a phase shift of 90 degrees and a gain of ω . If n samples are used K will be a nxn matrix which will eliminate the DC offset of and input vector of size n.

2.3 DESIRED FILTERING FACTORS

The K matrix provides the desired output for n samples, this is good for off line applications but not for on line applications where the output corresponding to the last acquired input is required. But if only the last row of matrix K is used and the previous n-1 samples are known, the filtered output corresponding to the last input will be obtained. Therefore the DC offset can be removed by the a discrete time algorithm of the form:

$$V_n = Y_n K_n + Y_{n-1} K_{n-1} + \dots + Y_1 K_1 + Y_0 K_0 \quad [2.19]$$

Where $K_0, K_1 \dots K_n$ are the elements of the last row of matrix K.

For the matrix obtained in the example for $n = 3$ using integer numbers the following equation is obtained:

$$V_0 = 12274Y_0 - 1748Y_{-1} + 574Y_{-2} \quad [2.20]$$

Therefore if the n-1 previous sampled inputs are known the filtered output corresponding to the last sampled input is obtained by using the derived algorithm. Although it is not recursive the use of the previous n-1 samples introduces a delay in the time response of the output signal. This delay increases as the number of samples used for the calculation increases. On the other hand the lower the number of samples used the higher the distortion due to noise in the sampled signal. A compromise between accuracy and the response time will dictate the number of samples to be used by the algorithm.

CHAPTER 3

HARDWARE AND SOFTWARE CONFIGURATION

The application of a digital mimic algorithm for computer relaying requires the use of the computer hardware. It is desired that no additional hardware be added to the computer relay. Knowledge of the computer relay hardware is then necessary for the correct implementation of the digital mimic algorithm. Limits introduced by the the relay hardware need to be recognized and taken into consideration. At the moment there are no standards for computer relaying hardware or software and there is a tendency for faster and more powerful microprocessors to replace the ones now available. The currently used hardware in computer relaying will be considered for the implementation of the digital mimic algorithm. It is assumed that correct operation of the algorithm with the available hardware

is a good indication that it can function satisfactorily now, and improve its performance with more advance hardware of the future.

3.1. RELAYING HARDWARE

The computer relay hardware can be divided in three major categories according to their functions: analog input circuit, analog to digital converter, and microprocessor unit (Figure 4). The components and functions of each of these categories will be briefly described to determine their effect, if any, on the digital removal of the DC offset from the sampled current.

3.1.1 Analog Input Circuit

The analog input circuit consists of current and voltage transformers, surge withstand capability (SWC) filters, anti-aliasing filters and signal conditioners. The voltage and current transformers take the high voltage and high current signals and transform them into acceptable levels for the relays. For the implementation of the digital mimic algorithm it is assumed that the gain errors in the voltage and current transformers are known and that there is no saturation error in the current transformer.

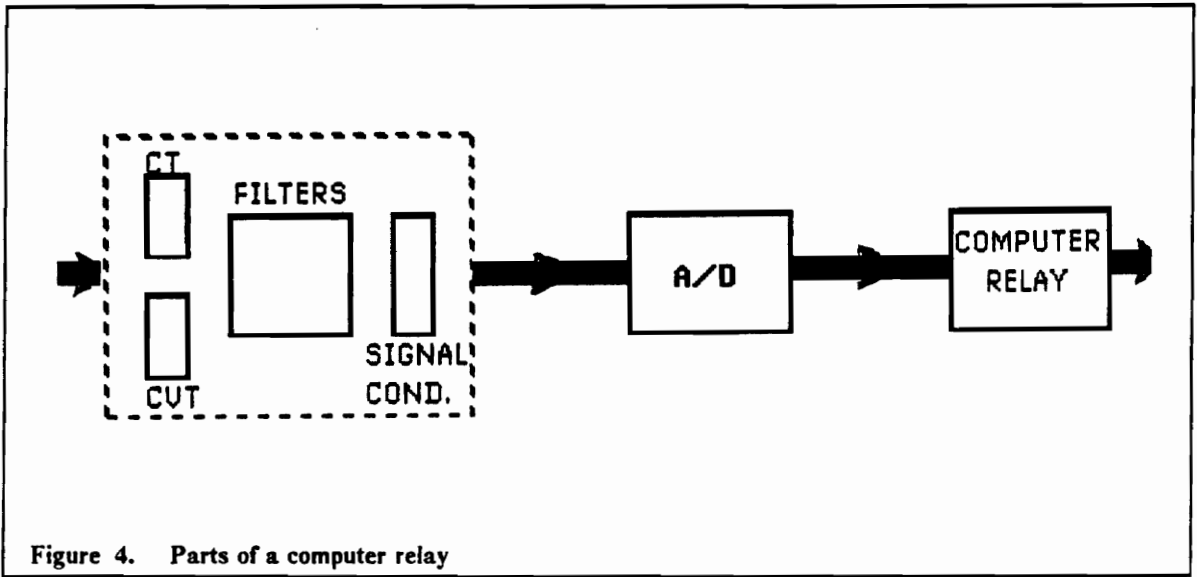


Figure 4. Parts of a computer relay

The SWC filter protects the computer hardware from surges in the power system which are deadly for digital equipment. These filters can be simple LC filters, metal oxide varistors or zener diodes. SWC filters should withstand surges of up to 4 KV at frequencies up to 1.5 MHz.[6]

Anti-aliasing filters limit the signal to a bandwidth of half the sampling frequency to avoid the appearance of aliased low frequency components which do not exist in the sampled signal but are created by the sampling process [7]. The most used filter for this purpose is a second order RC filter. Anti-aliasing filters introduce a delay of about one sample period in the input signal. This delay must be considered in the overall delay of the relay input signal.

Signal conditioners provide filtering of high frequency components and bring the CVT and CT outputs to levels acceptable by the A/D converters, usually $\pm 10V$. The signal conditioners filter a lot of the noise from the input signal but noise is not completely eliminated from the input signals; therefore noise in the sampled data should be considered in the implementation of the digital mimic algorithm.

3.1.2 Analog to Digital Converter

The analog to digital converter takes the current and voltage signal magnitude and transforms them into binary numbers. These binary numbers are the input

signal to the microprocessor. The accuracy of the converted samples is limited by the binary size of the A/D converter [8]. The sizes of commercially available A/D converters are 8, 12 and 16 bits. The 8 bit converter has the lowest acquisition time but it also has the lowest precision available with an acquisition error of 0.039 p.u. for a 10v input signal. It is meant to be used with the slower 8 bit microprocessors. The 16 bit A/D converter has the lowest acquisition error (0.00015 p.u. for a 10v input) and low acquisition time. Its high price makes it unattractive for computer relaying at this time but will probably become more attractive in the future. The 12 bit A/D converter offers a good compromise between price and accuracy and is the most commonly used converter for computer relaying at present. The 12 bit A/D converter has an acquisition error of 0.0024 p.u. for a 10V input signal with an acquisition time as low as 12 microseconds per channel taking into consideration the data acquisition software overhead. This time is part of the data window and must be considered when selecting sampling rates. It is assumed that a sample and hold circuit is used for the input signal to the A/D converter. If a sample and hold circuit is not used a difference in sampling of up to 2 degrees can be observed between the first and the last sample in a data scan.

The maximum input signal to an A/D converter is usually $\pm 10V$. This requires that the steady state input signal to the converter be less than 5 volts to avoid saturation of the A/D output when a maximum DC offset is created by a fault in the transmission line.

3.1.3 Microprocessor Unit

This unit consists of a microprocessor, RAM, ROM and serial communication chips incorporated in a single board. Microprocessors are rated by their speed and bit size of their data and address bus. At present there are three types of microprocessors commercially available with: 8, 16 and 32 bit words and the current tendency is for higher sizes. Eight bit precision is low for computer relaying applications and to obtain higher precision with an 8 bit microprocessor it becomes necessary to create higher computational burden. Therefore the 8 bit microprocessor will not be considered for the algorithm implementation. Thirty two bit microprocessors offer fast processing time and high precision but at the present its high cost makes it unattractive for computer relaying. The 16 bit microprocessor is the most commonly used microprocessors in computer relaying and is a good match for the 12 bit A/D converters. There is no doubt that as they become more readily available in the future, the price of 32 bit microprocessors will drop and will probably become more popular for computer relaying. However at the moment the 16 bit microprocessor is good enough and will be used for the implementation of the digital mimic algorithm.

The speed of a microprocessor depends on its clock speed which varies between 6 and 20 MHz. For a 16 bit microprocessor the clock speed is usually around 12.5 MHz. At this speed the common execution time for a 16 bit microprocessor is around one microsecond per instruction. The speed of the micro-

processor limits the number of operations that can be performed in each data window. A 16 bit microprocessor with a 12.5 MHz clock and one microsecond average instruction time will be used for the implementation of the digital mimic algorithm.

3.2 SOFTWARE CONSIDERATIONS

The relaying software and hardware introduce limits in the sampling rate and the numerical notation used by the digital mimic algorithm. The computational burden of the relaying program and the speed of the microprocessor clock limits the portion of the data window available for implementation of the digital mimic algorithm. Limitations on numerical notation (floating point or two's complement integers) are also introduced by the relay hardware and software. These limitations need to be known for the correct implementation of the digital mimic algorithm.

3.2.1 Sampling Rate

The minimum sampling rate of a relaying algorithm is determined by the Nyquist sampling theorem which establishes that in order to recover all the information of a sampled signal of frequency ' f ' it must be sampled at a frequency of at least ' $2f$ ' [9]. Therefore for a fundamental component at 60 Hz the mini-

imum sampling frequency is 120 Hz. Additional limits on the sampling frequency are established by the relaying algorithm. To make the mimic algorithm independent of the relay sampling frequency specially if a high sampling frequency is desired will require that the mimic algorithm be run in a separate microprocessor board. This means an addition to the relaying hardware increasing the price of the computer relay. This option may be economically feasible in the future but for the present it is preferred that the algorithm be ran in the same microprocessor.

The most popular algorithms for distance relaying use Fourier transform for the computation of the fault impedance [3,10,11]. For these algorithms the most common sampling frequency is 720 Hz since it reduces the computational burdens for algorithm implementation. For the implementation of the mimic circuit the sampling frequency must be at least 720 Hz so that the output of the filter can be made available to the relay algorithm at a rate of 720 Hz. Higher sampling rates must be multiples of 720 Hz to facilitate their output to the relay algorithm. The amount of time required by the relay and digital mimic algorithms determines the upper limit of the sampling frequency. For frequencies higher than 1440 Hz it is necessary to use a separate microprocessor for the digital mimic which has already been ruled out. For a frequency of 1440 Hz the time window available for computation is almost fully utilized precluding the addition of new features to the relay and mimic algorithms. In the future the use of faster microprocessors will allow for faster sampling rates. At the present a sampling fre-

quency of 720 Hz will be used for the implementation of the digital mimic algorithm.

3.2.2 Arithmetic Considerations

Factors such as numeric notation, register overflow and computational burden must be considered for the implementation of the digital mimic algorithm since they affect the precision and computational burden of the algorithm.

The numerical notation to be used will introduce limits in the precision and computational burden of the algorithm. Floating point operations can be performed in some 16 bit microprocessors without the need to add extra software or hardware. Floating point numbers eliminate the possibility of register overflow in the microprocessor registers, but increase the computational burden of the algorithm and introduce a conversion error when integer input numbers are transformed to floating point numbers.

When floating point notation is used a round-off error is introduced in the mantissa this error is smaller for larger mantissa size but it always exists. When integer numbers are used an overflow error occurs when the output size exceeds a maximum value. If this maximum value is not exceeded then this notation offers more precision than floating point numbers. For the digital mimic algorithm the inputs have a limited size of 12 bit (± 2048) and the factors are limited

to 16 bits (± 32768), which gives a maximum product of ($\pm 6.72 \times 10^8$). This number needs more than 16 bits in integer notation; thus a long word (32 bits) ($\pm 2.15 \times 10^9$) is required for the output. This guarantees that there will be no register overflow as long as the mimic algorithm uses less than 32 samples to filter the DC offset. At 720 Hz (12 samples per cycle) 32 samples will produce a long time delay in the algorithm response and increase the computational burden. Therefore 32 samples is well above the maximum number of samples to be used by the algorithm, then with an output 32 bits long (a long word) the register overflow problem is avoided.

Precision errors in the computation are introduced only if the relay algorithm requires a 16 bit integer input which will require a conversion of the long word (32 bits) into a word (16 bits). Therefore reducing the precision of the algorithm output. By using integer numbers the computational burden is kept to a minimum since the normal operational mode of the 16 bits microprocessor is used. At a speed of one instruction per microsecond the digital mimic computation time is under 100 microseconds. Therefore two's complement integer notation with an output word size of 32 bit (a long word) is chosen for the implementation of the digital mimic algorithm.

CHAPTER 4

EXPERIMENTAL RESULTS

Several variations of the algorithm were tested using a microprocessor connected to a power system simulator. This chapter describes the hardware of the computer relay and the simulator used to test the four algorithm variations. The algorithms tested and the graphical results of the tests are presented in figures which show the input and the output signal of the algorithms for different levels of DC offset. In addition the output of a digital mimic circuit for similar faults is also given.

4.1 TESTING HARDWARE

4.1.1 Relay Hardware

The algorithms were tested using a Motorola 68020 16 bit microprocessor with a 12.5 MHz clock and an estimated average speed of one microsecond per instruction. A Data Translation DT1402-F-16SE board was used for analog to digital conversion of the 4 input currents with an acquisition time of 12 microseconds per channel. An external 720 Hz pulse was used to trigger A/D conversion. The input signal to the microprocessor was passed through signal conditioners to provide high frequency filtering at appropriate voltage levels.

4.1.2 Simulator Hardware

The system simulator at the Virginia Tech Power Systems Laboratory uses three phase PI section for high voltage transmission line models, microprocessor simulated generators, signal conditioners, transformers and synchronous switches for fault simulation.

The PI section transmission line model simulates 345 or 765 KV three phase transmission lines with an impedance angle of about 80 degrees [12]. This value is needed to implement the algorithm. Each PI section is equivalent to 20 miles

of transmission lines; therefore any line length in multiples of 20 can be simulated.

The generator is simulated by a Motorola 68000 microprocessor. The microprocessor outputs a 3 phase sine wave at a rate of 48 points per cycle. The output of the generator goes through a RC filter to smooth the signal. The generator signal is amplified from 5 to 30 volts and feed to the line models through a transformer.

A subroutine in the generator program provides zero crossing detection for the generator sine wave. This information is used by the synchronous switch subroutine to trigger faults at desired voltage angles. Since the generator program uses 48 points per cycle, faults can be triggered only in steps of 2.5 degrees with angles ranging from 0 to 360 degrees. The synchronous switch hardware includes Mercury wetted relays used to perform the fault simulation [13].

The signal conditioner units transform the sampled current to voltages used by the A/D converter. In addition, a low pass filter in the signal conditioner filters the high frequency components of the current. The signal conditioners allow for variable gain, thus input to the microprocessor can be set below 5 volts to avoid saturation of the A/D input register. There is an optional analog mimic circuit in the signal conditioner units. This analog mimic was used to obtained the analog mimic response for comparison to the algorithms.

4.2 SELECTED ALGORITHMS

Two contradicting characteristics are desired from the digital mimic: Accuracy and Fast response. The tests performed in the simulator determine the number of samples to be used and the possible algorithm variations that will allow an acceptable compromise between accuracy and fast response. The following algorithm variations were tested:

4.2.1 3 Sample Algorithm

The three sample algorithm offers the fastest response time to a change in the input signal. After three samples (quarter cycle for a 720 Hz sampling rate) the algorithm produces the correct output for the sampled input. The derivation of the 3 sample formula was done in the example in chapter two (equation 2.20). The decimal point has been been drop since integer notation has been selected for the algorithm implementation.

$$V_0 = 1227Y_0 - 1748Y_{-1} + 574Y_{-2} \quad [4.1]$$

This algorithm has also the lowest computational burden since it requires two additions and three multiplications for its computation. It produces only two outputs in the transition region between the pre-fault and post fault period.

4.2.2 6 Sample Algorithm

The equation for the six sample algorithm is derived in appendix A in similar way to the example in chapter two with n equal 6. The equation obtained is:

$$V_0 = 303Y_0 + -13Y_{-1} - 217Y_{-2} - 244Y_{-3} - 76Y_{-4} + 255Y_{-5} \quad [4.2]$$

This algorithm produces five output points in the transition region between the pre-fault and the post fault region. It also has a higher computational burden than the 3 point algorithm since it requires six multiplications and 5 additions.

4.2.3 3 Samples With Average

This algorithm uses a 3 sample equation with a 15 degree shift of the output signal. The output obtained for the present sample is added to the value obtained from the last sample and the result is divided by two to obtain the average. This can be considered a four sample algorithm but its averaging process allows for a faster response than the 4 sample algorithm since when the average is taken for the six sample on of the averaging factor is out of the transition region between the pre-fault and post-fault signal. (see derivation in appendix A) The equation obtained for this algorithm is derived in appendix A. The equation obtained is:

$$V_0 = 1495 \frac{(Y_0 + Y_{-1})}{2} - 2590 \frac{(Y_{-1} + Y_{-2})}{2} + 1118 \frac{(Y_{-2} + Y_{-3})}{2} \quad [4.3]$$

The computational burden of this algorithm is lower than the 6 samples algorithm but higher than the 3 samples algorithm since an extra addition and a shift operations are required.

4.2.4 3 Samples Average to 6 Samples

This algorithm is a combination of three samples average and the six samples algorithm (Figure 5). The first algorithm is used for steady state operation and for the first half cycle after the fault when the program switches to the six sample algorithm. This requires the use of a transient monitor subroutine to determine when to switch to the six sample formula. This algorithm has a higher computational burden than all the others due to the use of the transient monitor subroutine to detect the fault.

4.3 TESTS

The simulated system is shown in Figure 6. Faults were assumed close to the end of the line at a section considered to be outside the first zone of protection of the relay. Samples were taken from phase 'a' and all faults simulated were

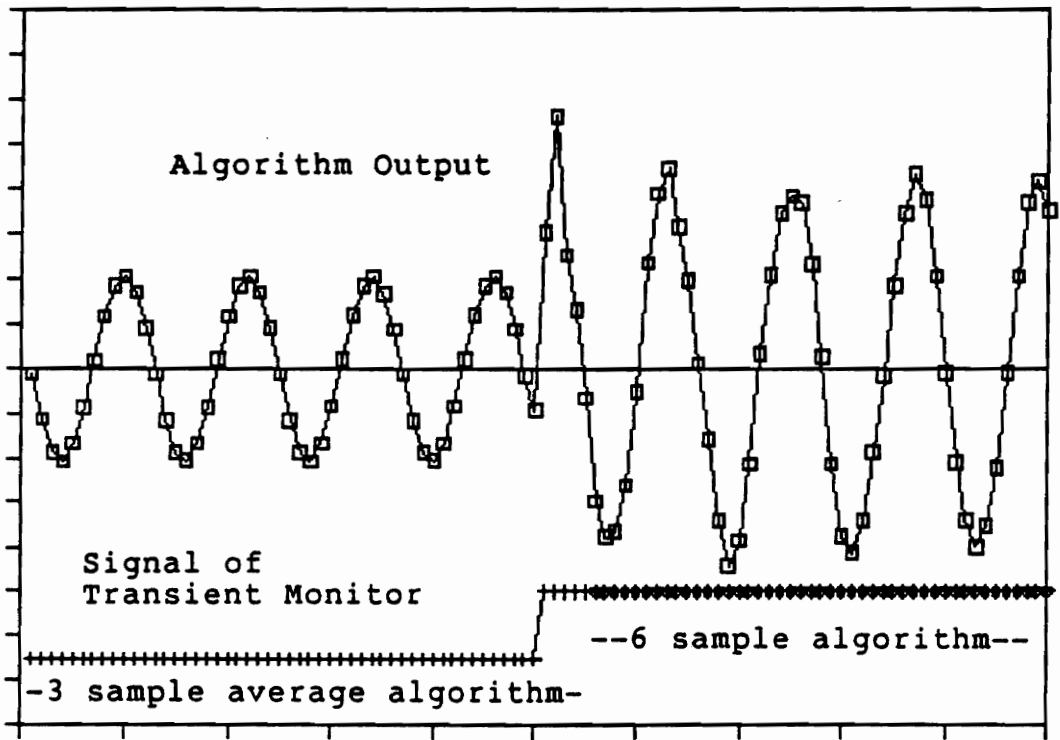


Figure 5. Change from 3 sample average to six sample algorithm

phase 'a' to ground faults. The four algorithms were executed at the same time to obtain an output signal from all algorithms corresponding to a common input signal. Samples from the analog mimic circuit were taken separately for faults at the same angles used to test the algorithms.

Faults were placed at voltages angles of 0, 90, 135, 180, -45, -90 and -135 degrees. The results are shown in groups of figures in the next pages. Each group of figures consists of an input plot and five output plots corresponding to the analog mimic and the four algorithms.

In Table 1 comments are given for the output plots from the tests run in the simulator. The program was arranged to perform all the algorithms for the same input to provide a better comparison. The plots appear in Figures 7 to 48.

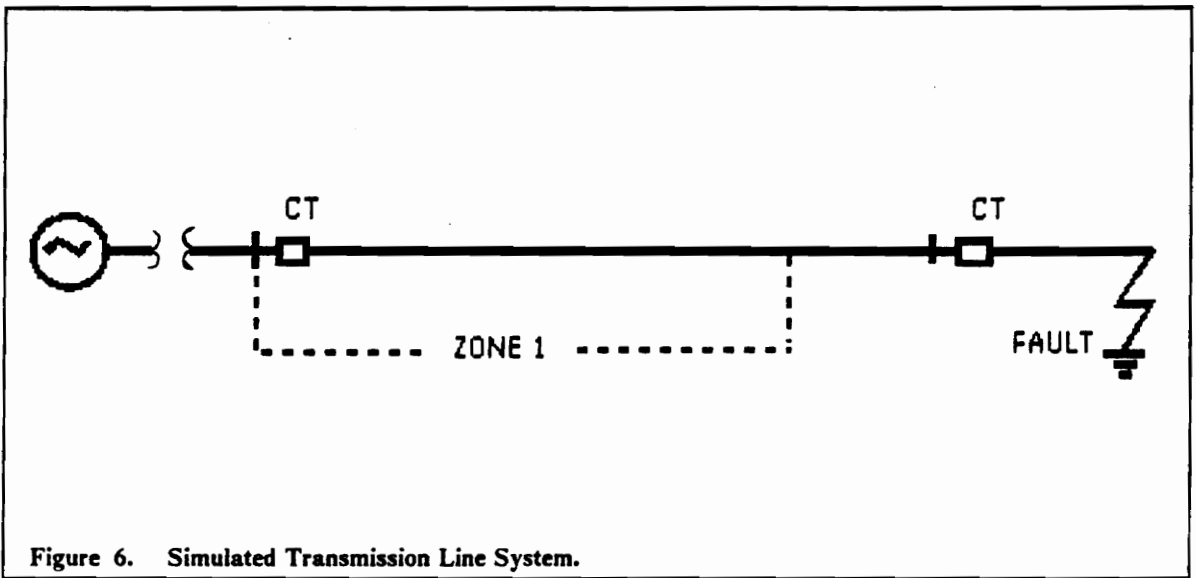


Figure 6. Simulated Transmission Line System.

Table 1. Tests Results (Comments)

CURRENT ANGLE	ALGORITHM /FIGURE	COMMENTS on signal output after the fault.
0 DEG.	Input (Fig. 7) Analog Mimic (Fig. 8) 3 Sample (Fig. 9) 6 Sample (Fig. 10) 3 Average (Fig. 11) 3 TO 6 (Fig. 12)	.Low +DC offset. Noise in first two cycles of the signal .Small distortion in first 2 cycles after the fault. .Highly distorted output first 2 cycles after fault. .Slow response but low distortion due to noise .Fast response some distortion on first cycle after the fault. .Fast response. Some distortion first cycle.
90 DEG.	Input (Fig. 13) Analog Mimic (Fig. 14) 3 Sample (Fig. 15) 6 Sample (Fig. 16) 3 Average (Fig. 17) 3 TO 6 (Fig. 18)	.Max +DC offset. Negligible noise in input signal. .Very small distortion in first cycle after the fault. .Fast response small distortion first 2 cycles. .Slow response first half cycle but no distortion. .Fast response small distortion first cycle. .Fast response small distortion first half cycle.
135 DEG.	Input (Fig. 19) Analog Mimic (Fig. 20) 3 Sample (Fig. 21) 6 Sample (Fig. 22) 3 Average (Fig. 23) 3 TO 6 (Fig. 24)	.High +DC offset. Some noise in first 2 cycles. .Some distortion in first cycle after the fault .Fast response high distortion in first cycle. .Slow response little distortion in first cycle. .Fast response small distortion first half cycle. .Fast response small distortion first half cycle.

Table 1. Cont.

CURRENT ANGLE	ALGORITHM /FIGURE	COMMENTS on signal output after the fault.
180 DEG.	Input (Fig. 25) Analog Mimic (Fig. 26) 3 Sample (Fig. 27) 6 Sample (Fig. 28) 3 Average (Fig. 29) 3 TO 6 (Fig. 30)	.Small -DC offset. Noise in the first three cycles. .Distortion in first 2 cycles after the fault. .Fast response high distortion in first 3 cycle. .Slow response little distortion in first cycle. .Fast response some distortion first cycle. .Fast response some distortion first half cycle.
225 DEG.	Input (Fig. 31) Analog Mimic (Fig. 32) 3 Sample (Fig. 33) 6 Sample (Fig. 34) 3 Average (Fig. 35) 3 TO 6 (Fig. 36)	.High -DC offset. High noise in the first cycle. .Minimum distortion in first cycle after the fault. .Fast response high distortion in first 2 cycles. .Slow response small distortion in first 2 cycles. .Fast response some distortion first 2 cycles. .Fast response some distortion first 1 and 1/2 cycles.
270 DEG.	Input (Fig. 37) Analog Mimic (Fig. 38) 3 Sample (Fig. 39) 6 Sample (Fig. 40) 3 Average (Fig. 41) 3 TO 6 (Fig. 42)	.Max -DC offset. Negligible noise in the signal. .Some distortion in first cycle after the fault. .Fast response small distortion first cycle. .Slow response first half cycle. .Fast response very small distortion first cycle. .Fast response very small distortion first 1/2 cycle.

Table 1. Cont.

CURRENT ANGLE	ALGORITHM /FIGURE	COMMENTS on signal output after the fault.
315 DEG.	Input (Fig. 43) Analog Mimic (Fig. 44) 3 Sample (Fig. 45) 6 Sample (Fig. 46) 3 Average (Fig. 47) 3 TO 6 (Fig. 48)	.High -DC offset. Noise in the first two cycles. .Distortion in first 2 cycles after the fault. .Fast response high distortion first 2 cycles. .Slow response small distortion in first cycle. .Fast response high pick first half cycle, distortion in first cycle.s .Fast response high pick first half cycle but no distortion after that.

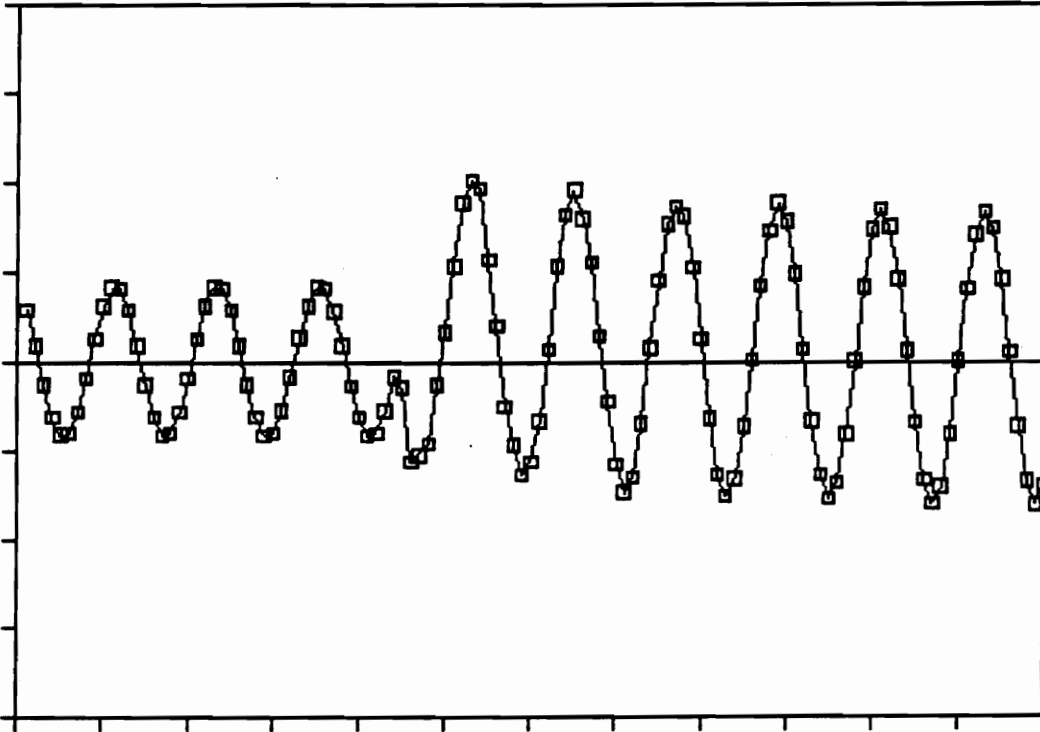


Figure 7. Input signal 0 degrees fault.

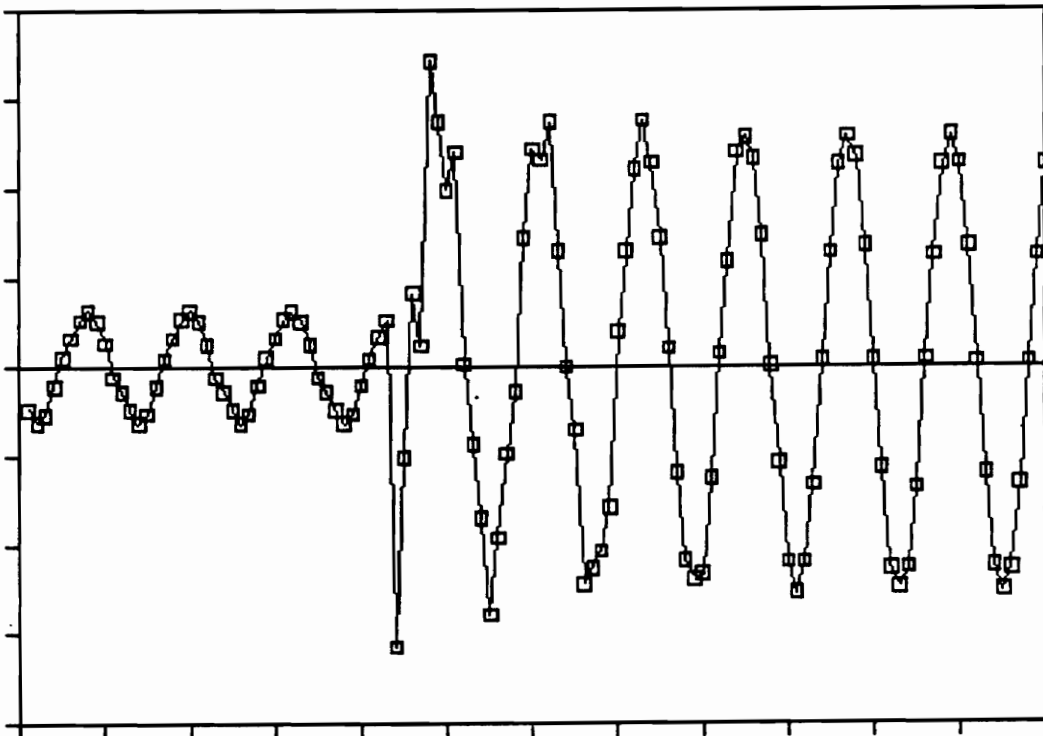


Figure 8. Output analog mimic circuit 0 degrees fault.

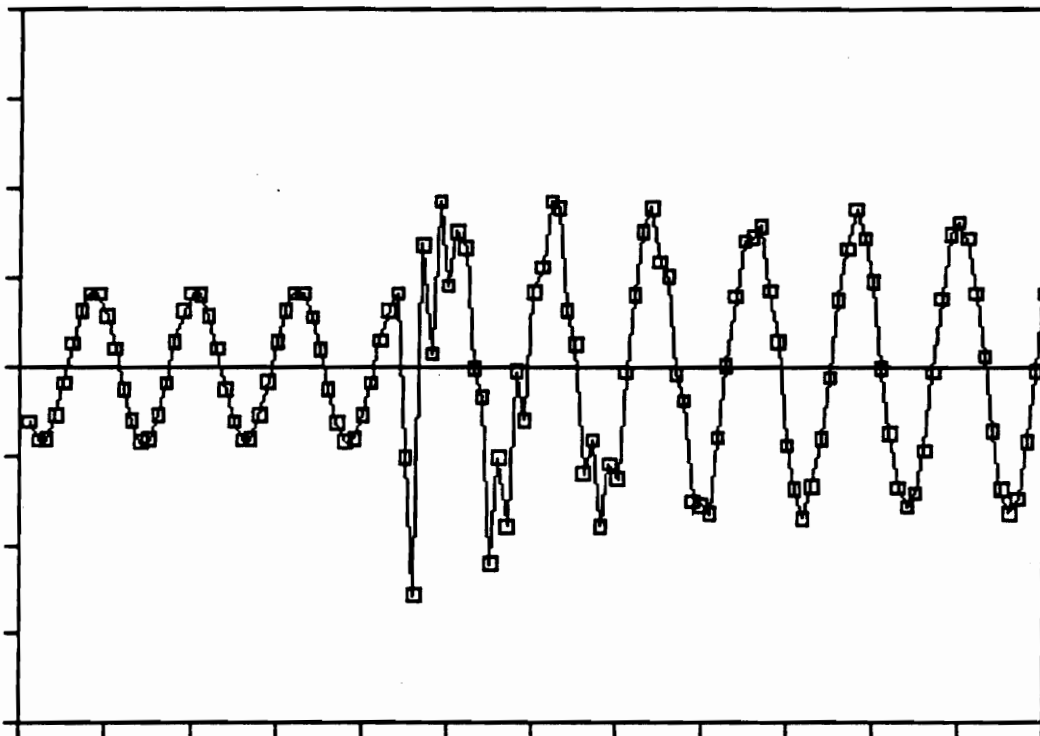


Figure 9. Output 3 sample algorithm 0 degrees fault

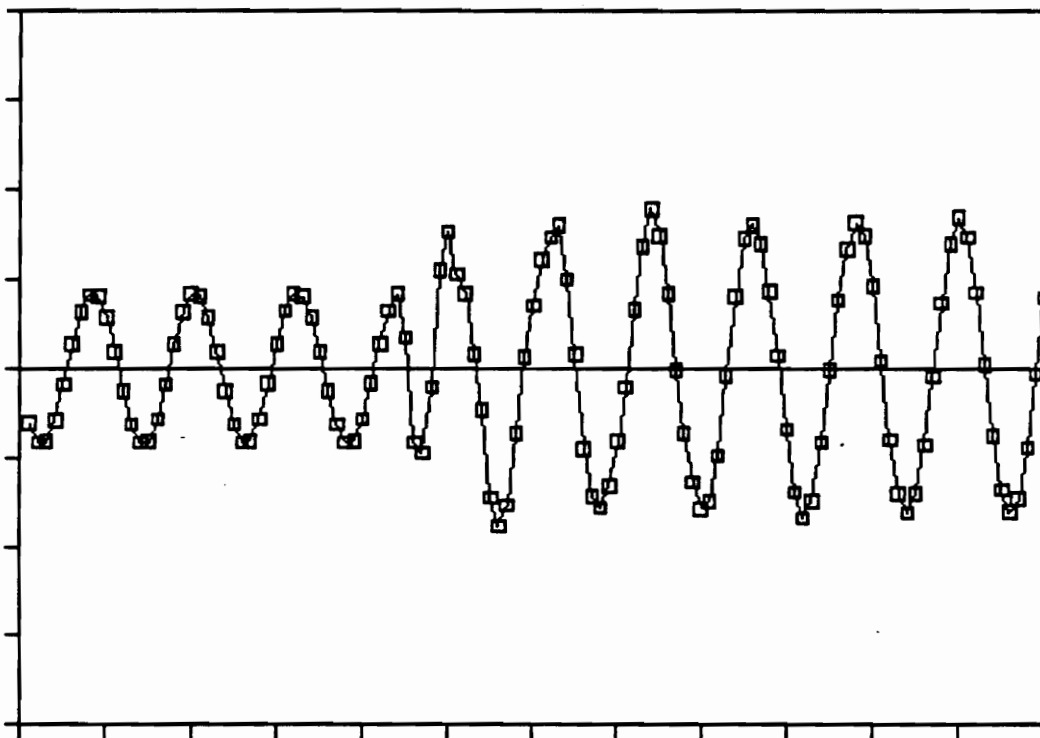


Figure 10. Output 6 samples algorithm 0 degrees fault

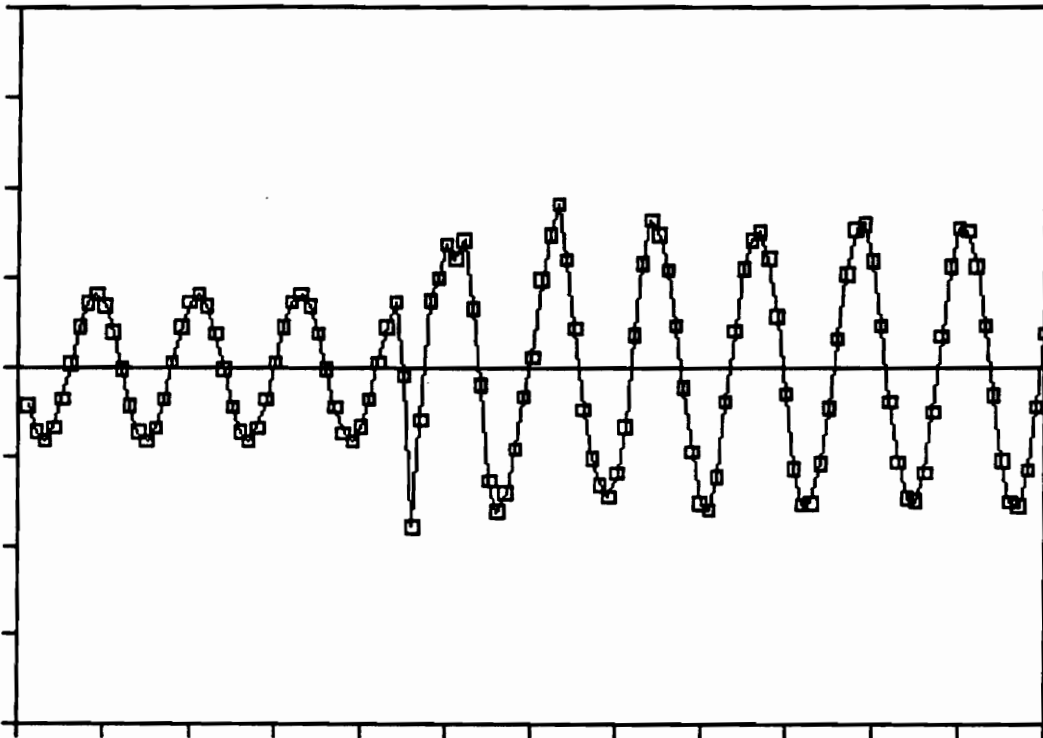


Figure 11. Output 3 samples average 0 degrees fault

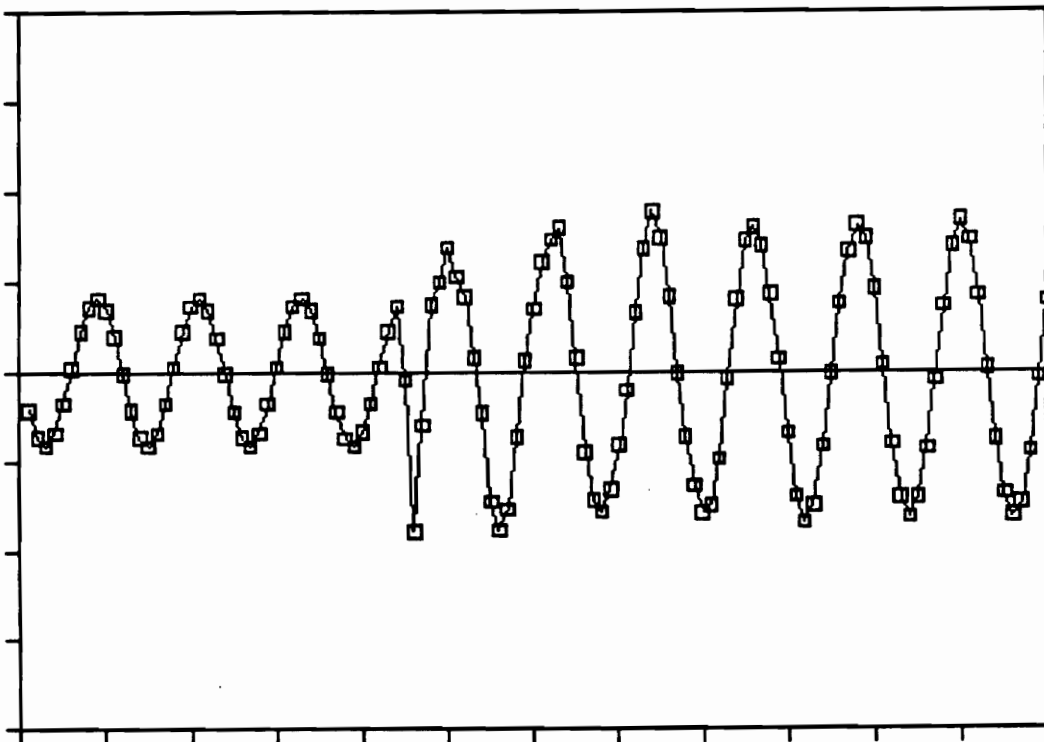


Figure 12. Output 3 samples average to 6 samples 0 degrees fault

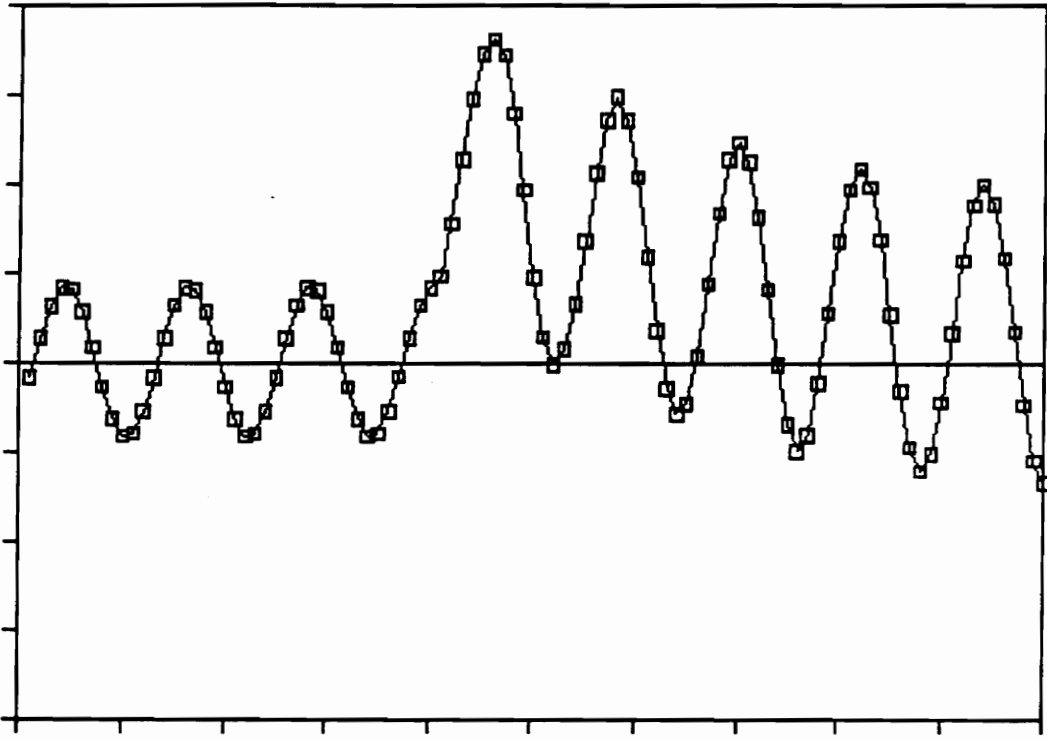


Figure 13. Input signal 90 degrees fault.

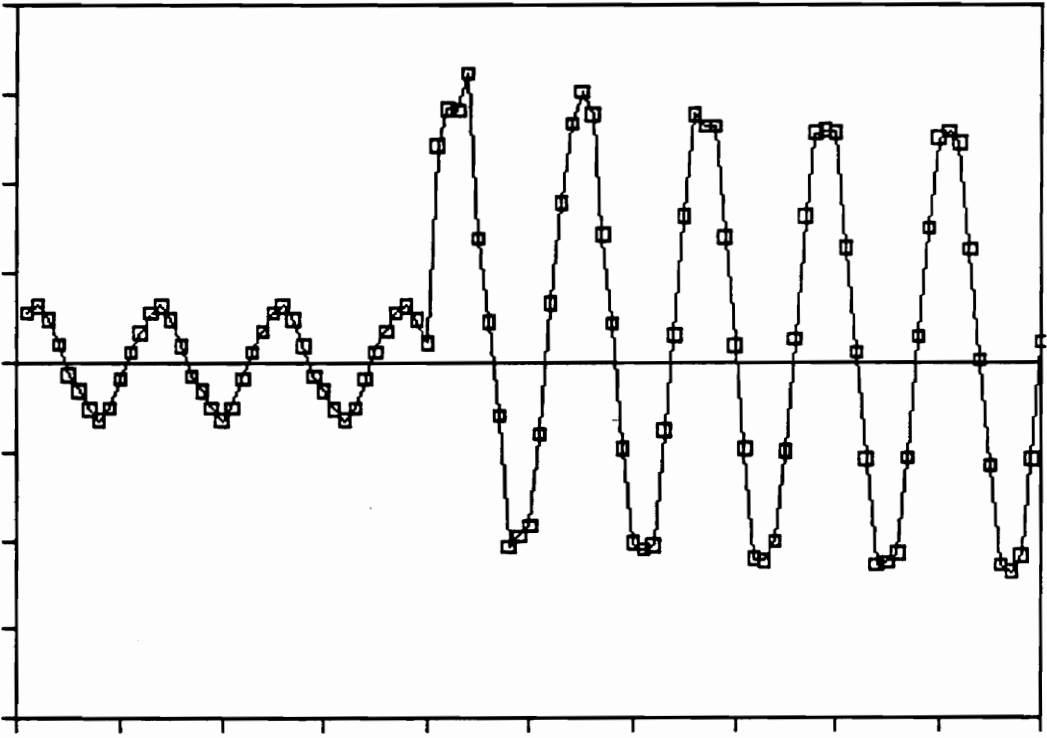


Figure 14. Output analog mimic circuit 90 degrees fault.

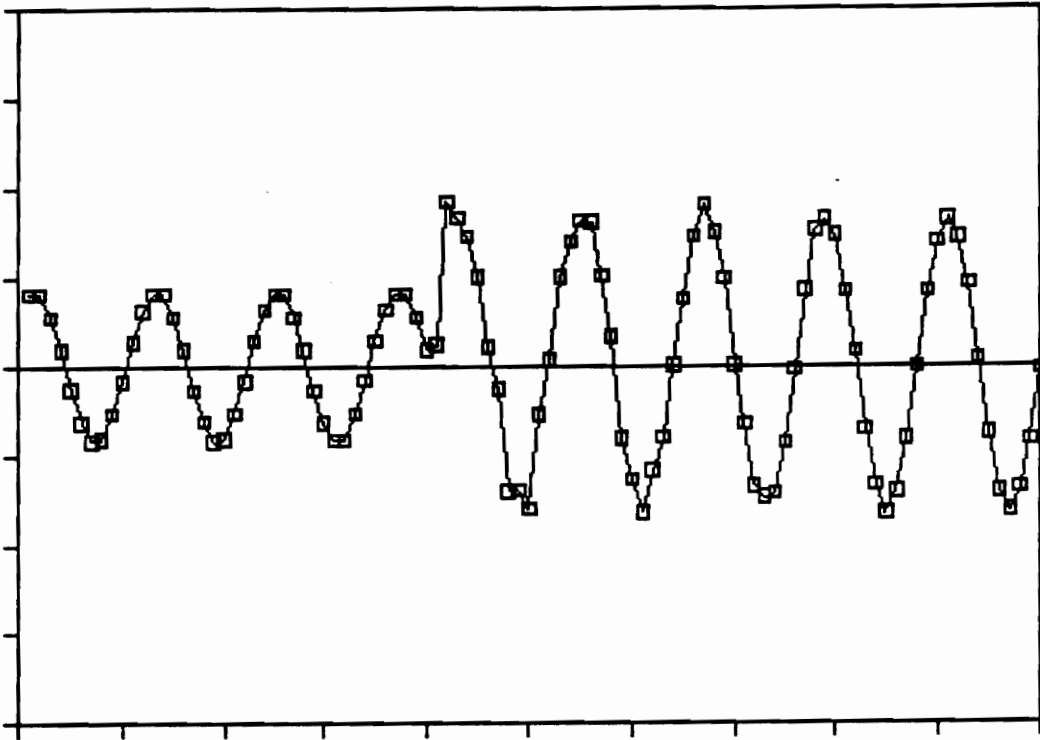


Figure 15. Output 3 sample algorithm 90 degrees fault

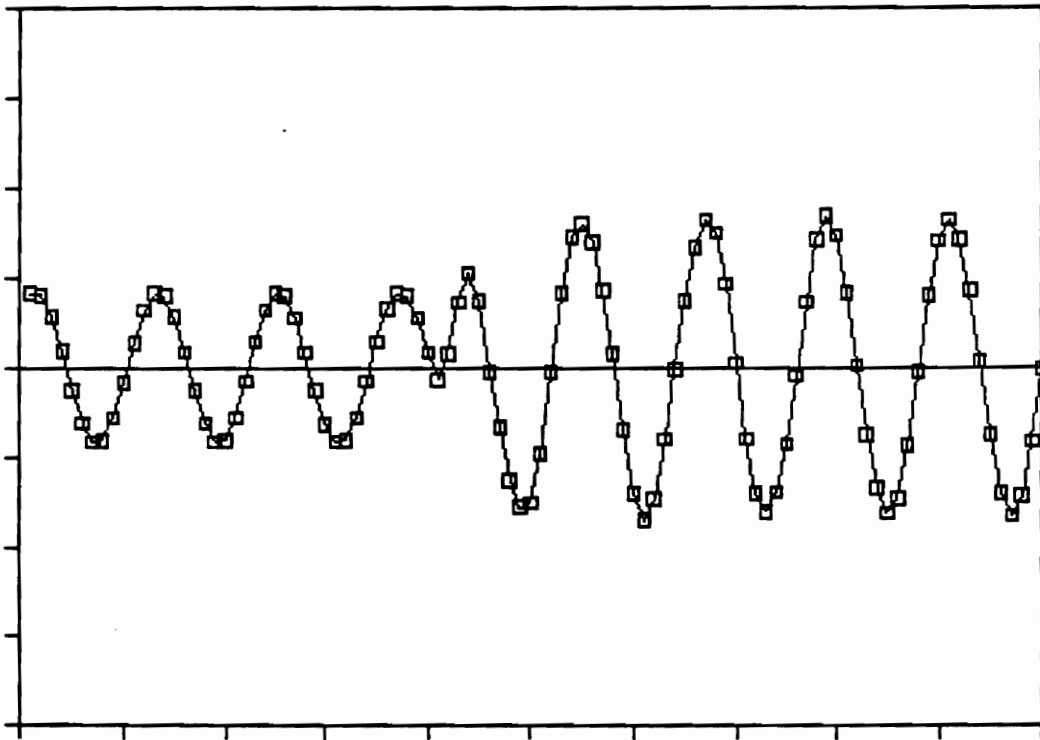


Figure 16. Output 6 samples algorithm 90 degrees fault

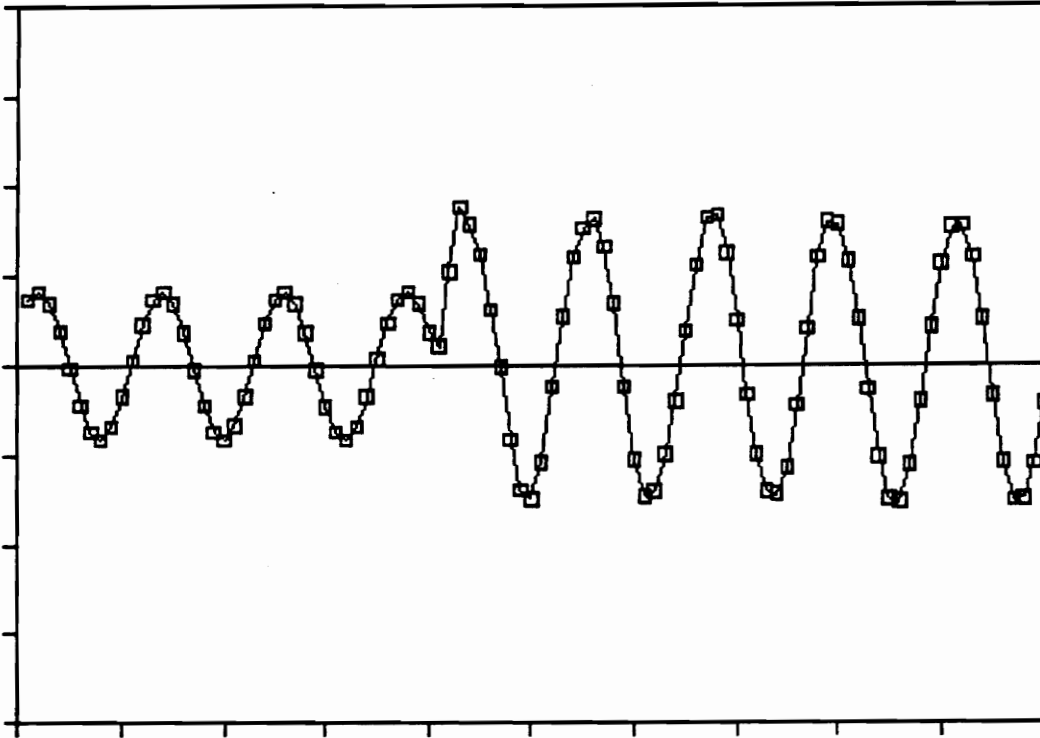


Figure 17. Output 3 samples average 90 degrees fault

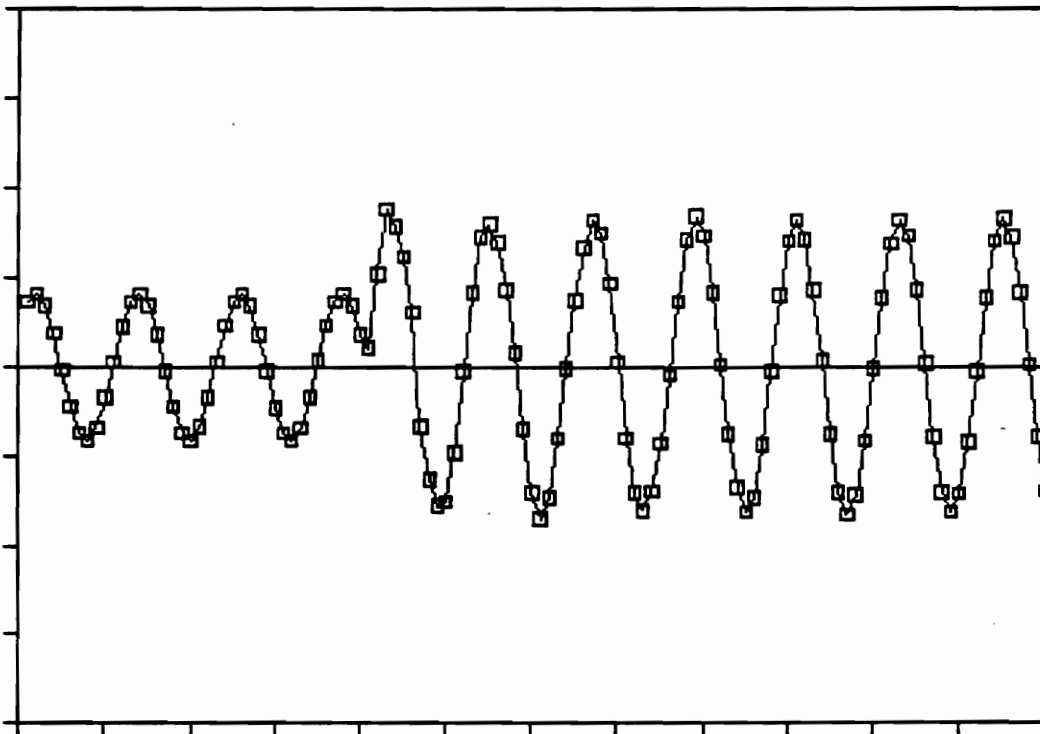


Figure 18. Output 3 samples average to 6 samples 90 degrees fault

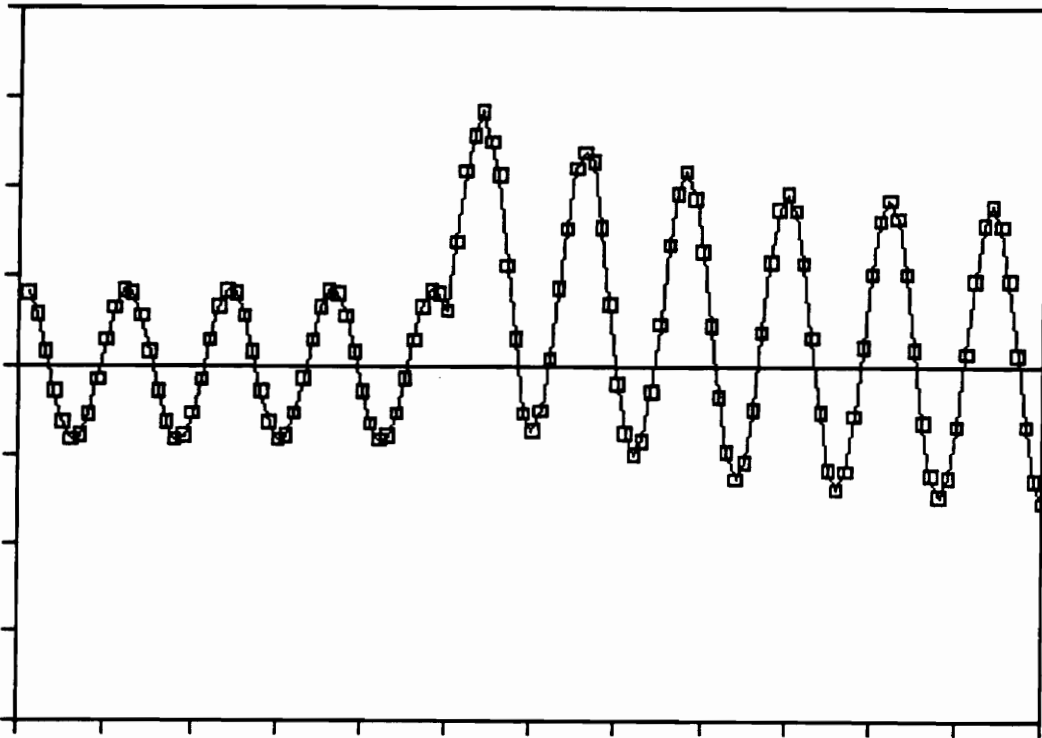


Figure 19. Input signal 135 degrees fault.

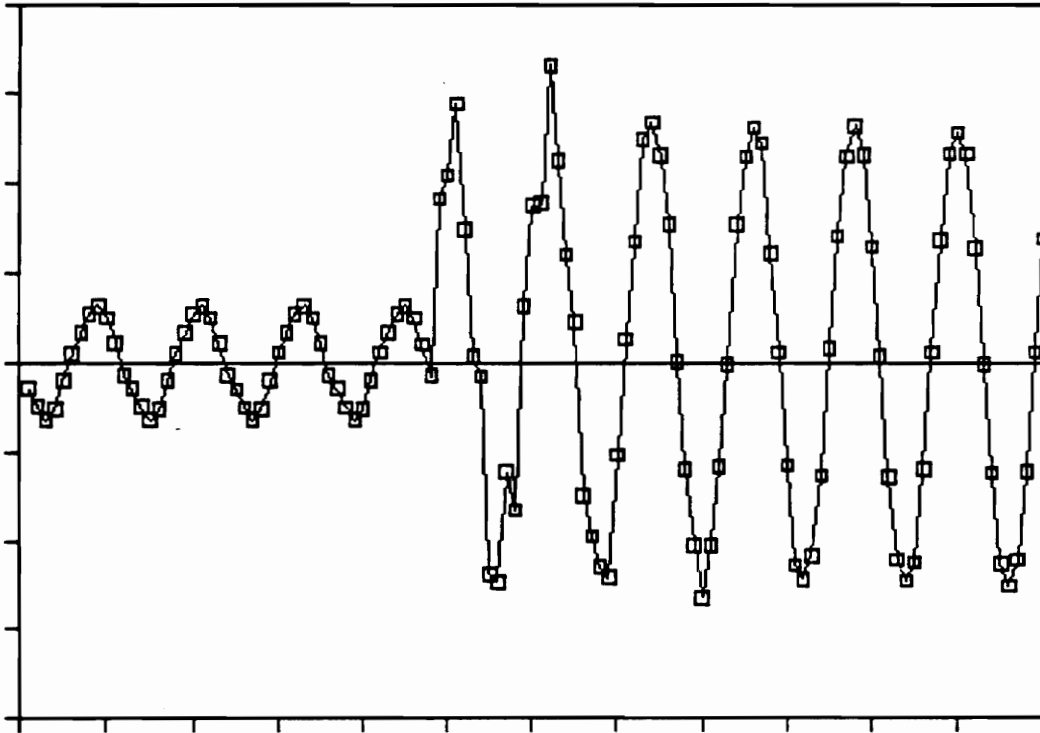


Figure 20. Output analog mimic circuit 135 degrees fault.

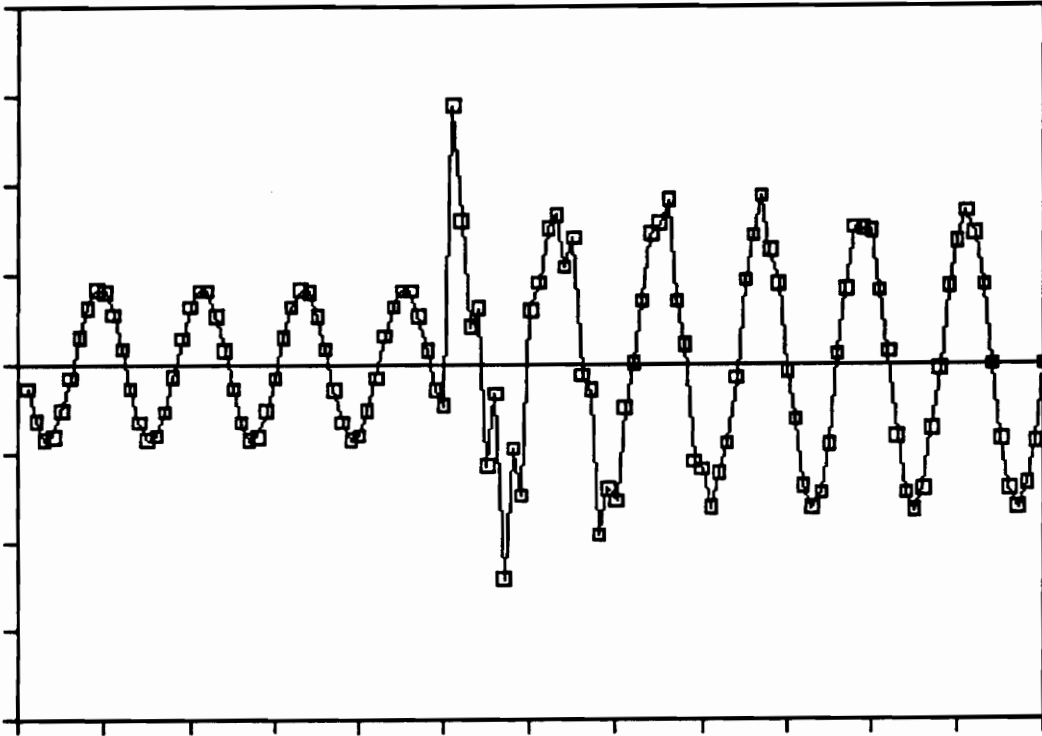


Figure 21. Output 3 sample algorithm 135 degrees fault

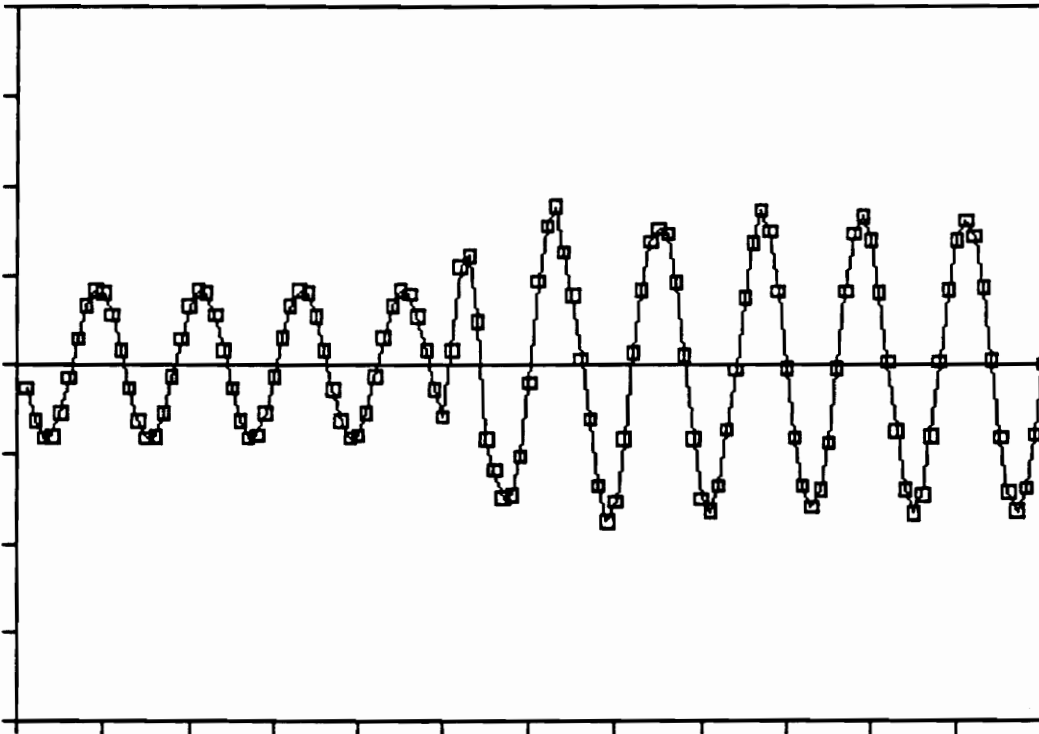


Figure 22. Output 6 samples algorithm 135 degrees fault

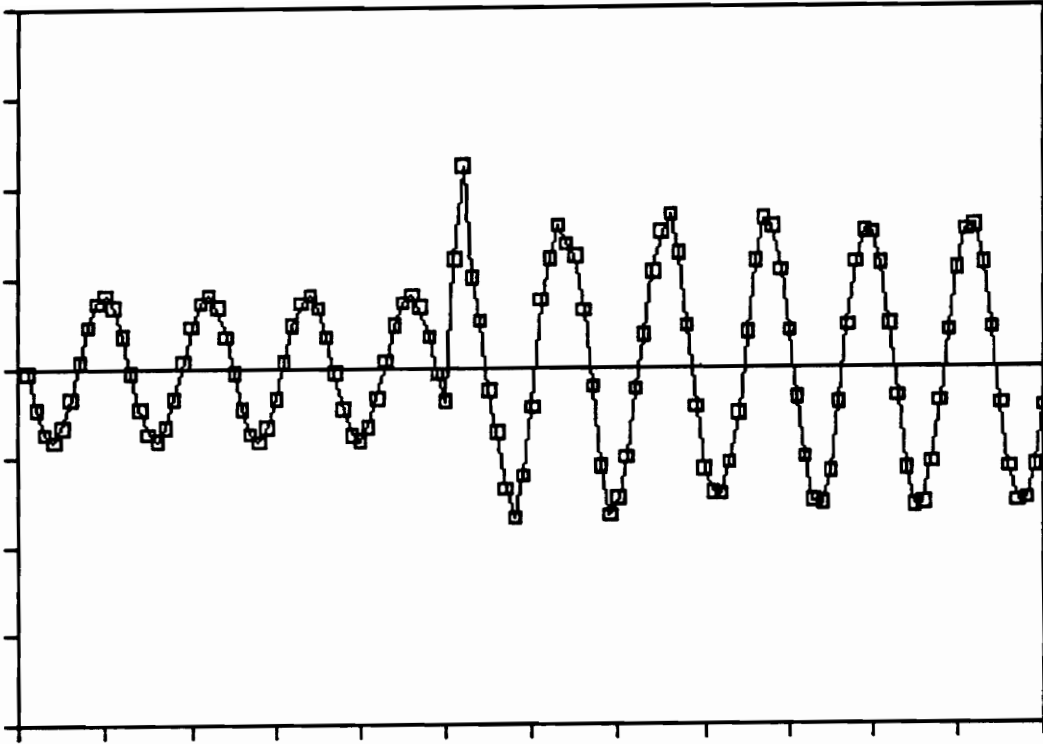


Figure 23. Output 3 samples average 135 degrees fault

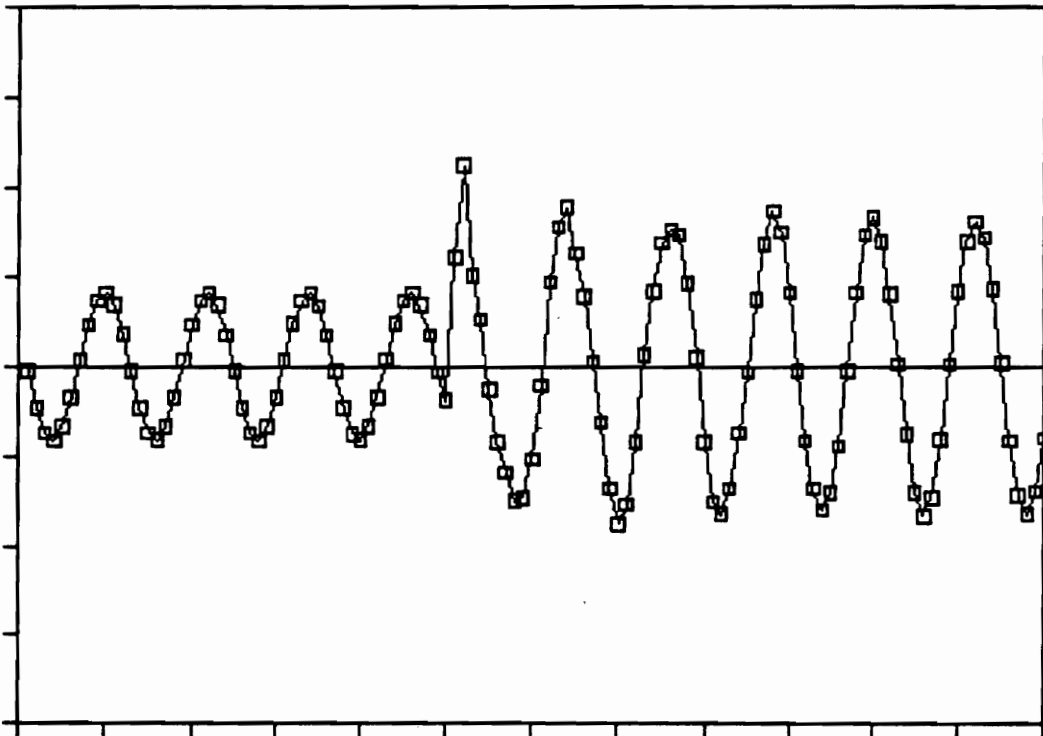


Figure 24. Output 3 samples average to 6 samples 135 degrees fault

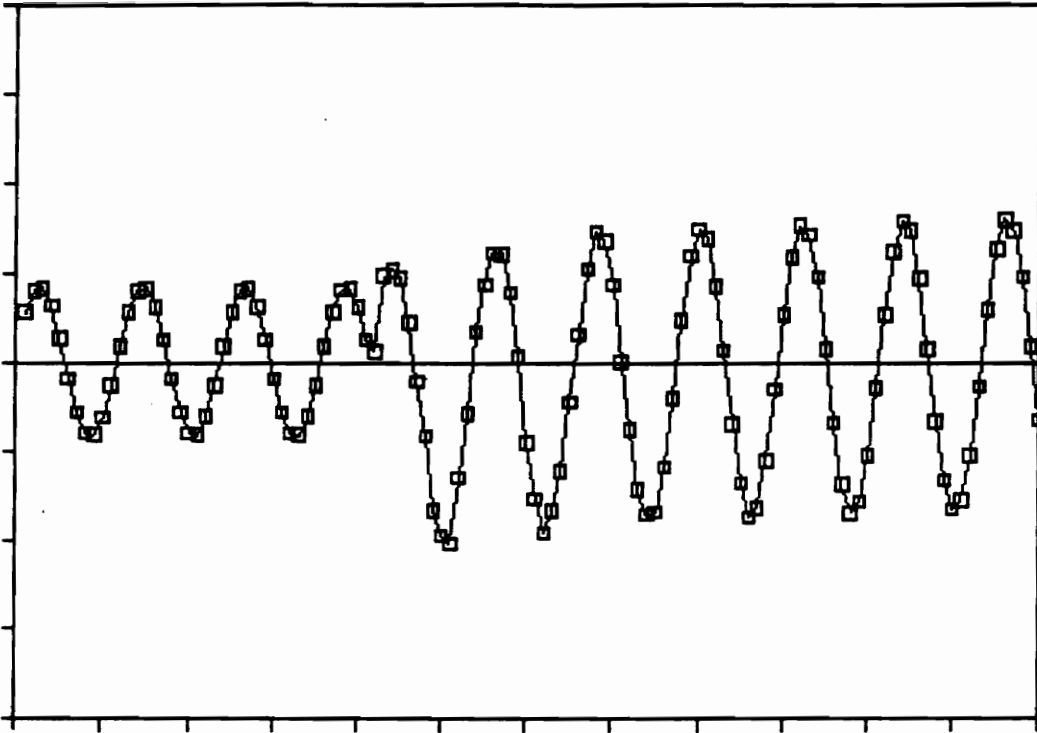


Figure 25. Input signal 180 degrees fault.

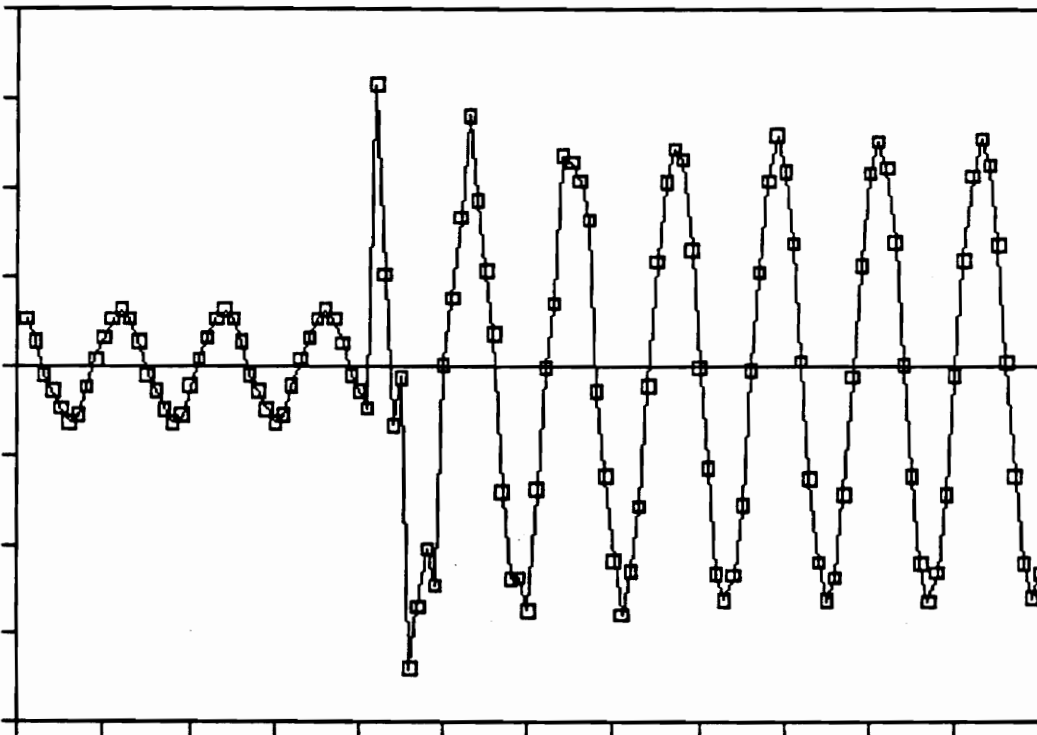


Figure 26. Output analog mimic circuit 180 degrees fault.

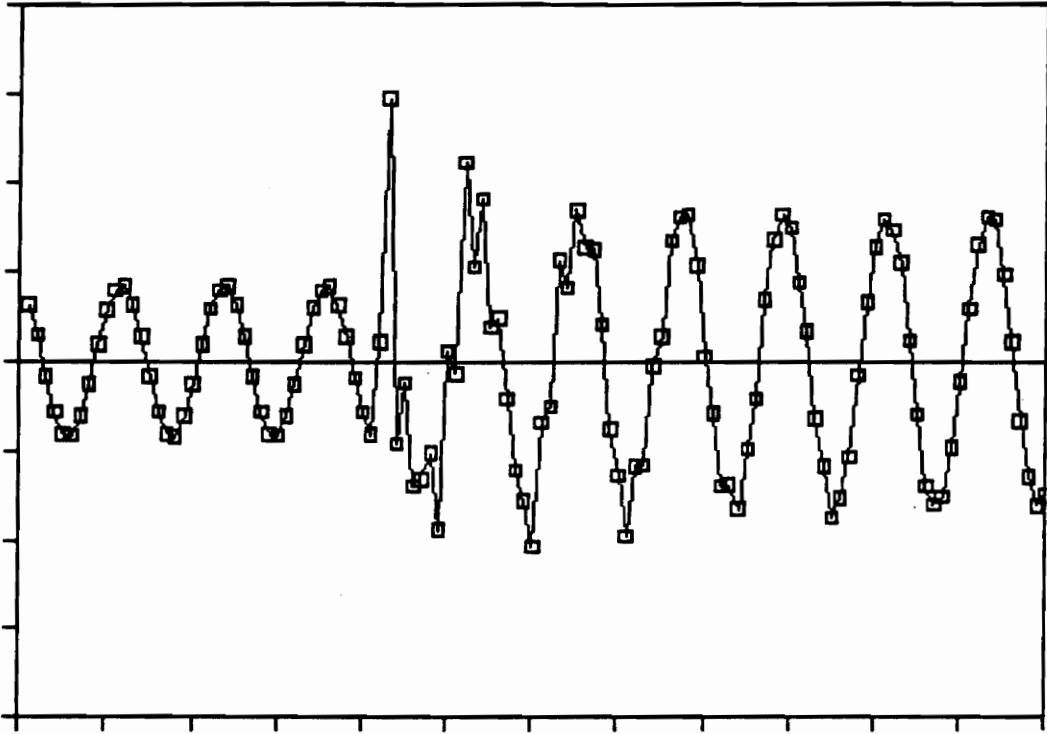


Figure 27. Output 3 sample algorithm 180 degrees fault

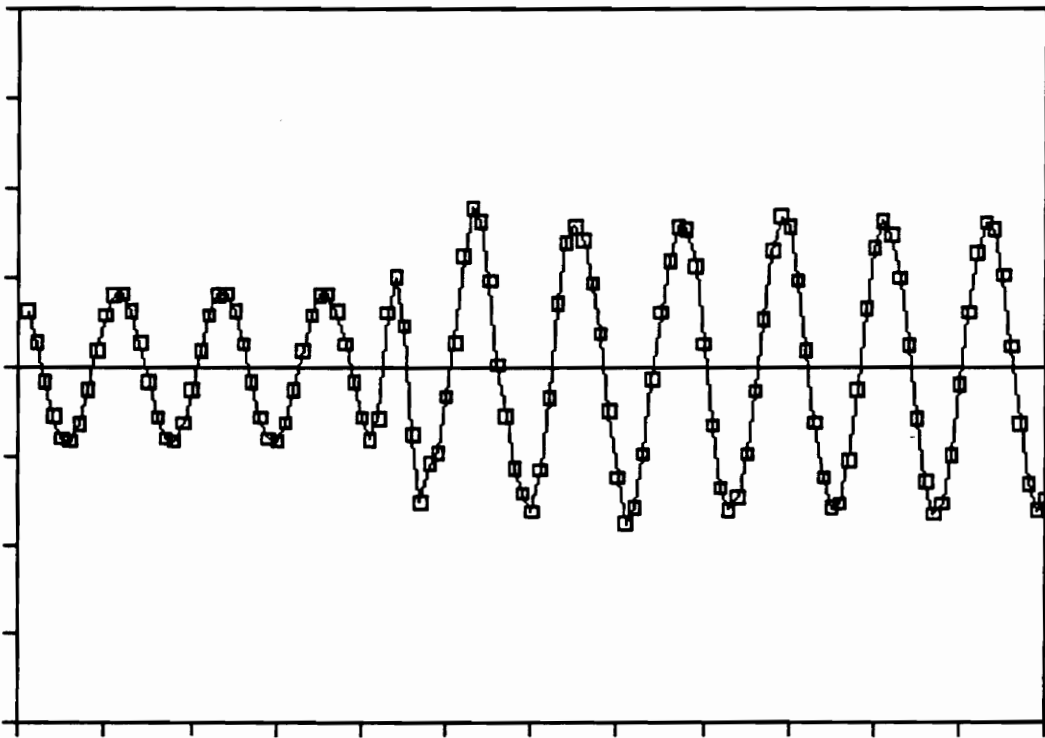


Figure 28. Output 6 samples algorithm 180 degrees fault

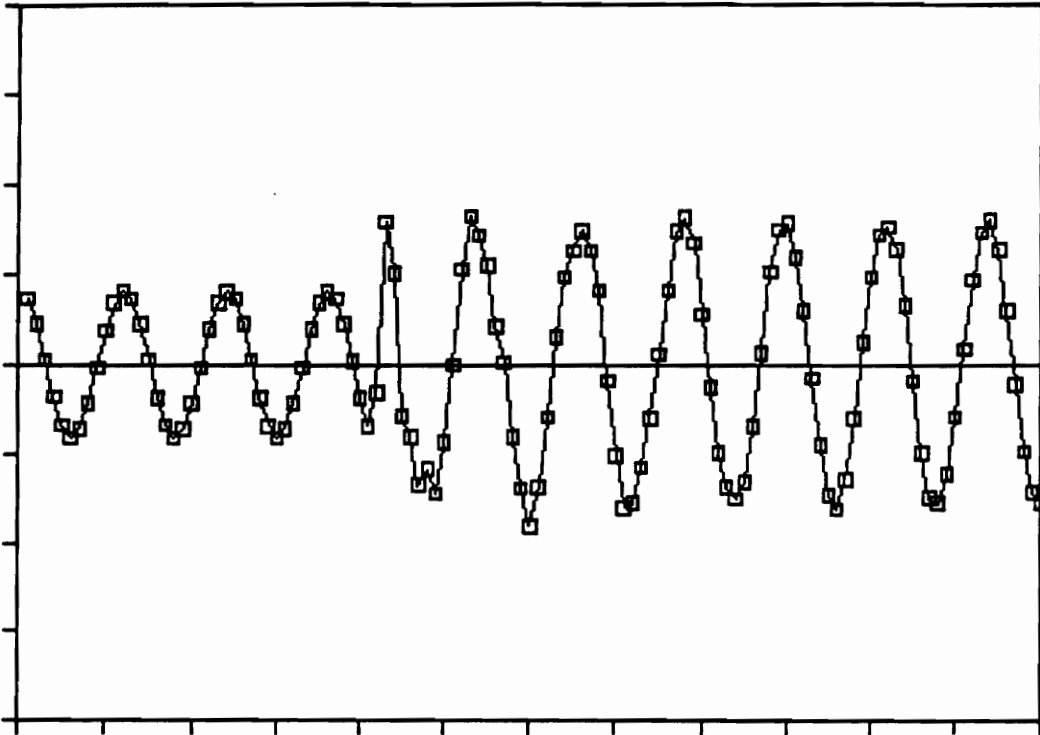


Figure 29. Output 3 samples average 180 degrees fault

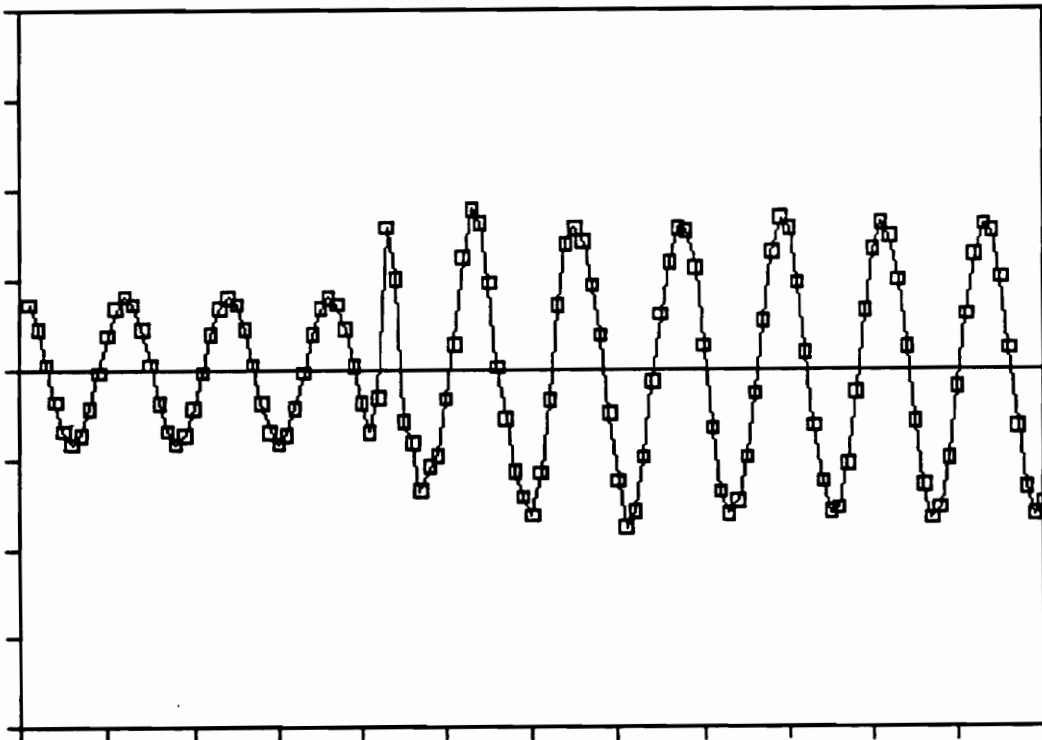


Figure 30. Output 3 samples average 180 degrees fault

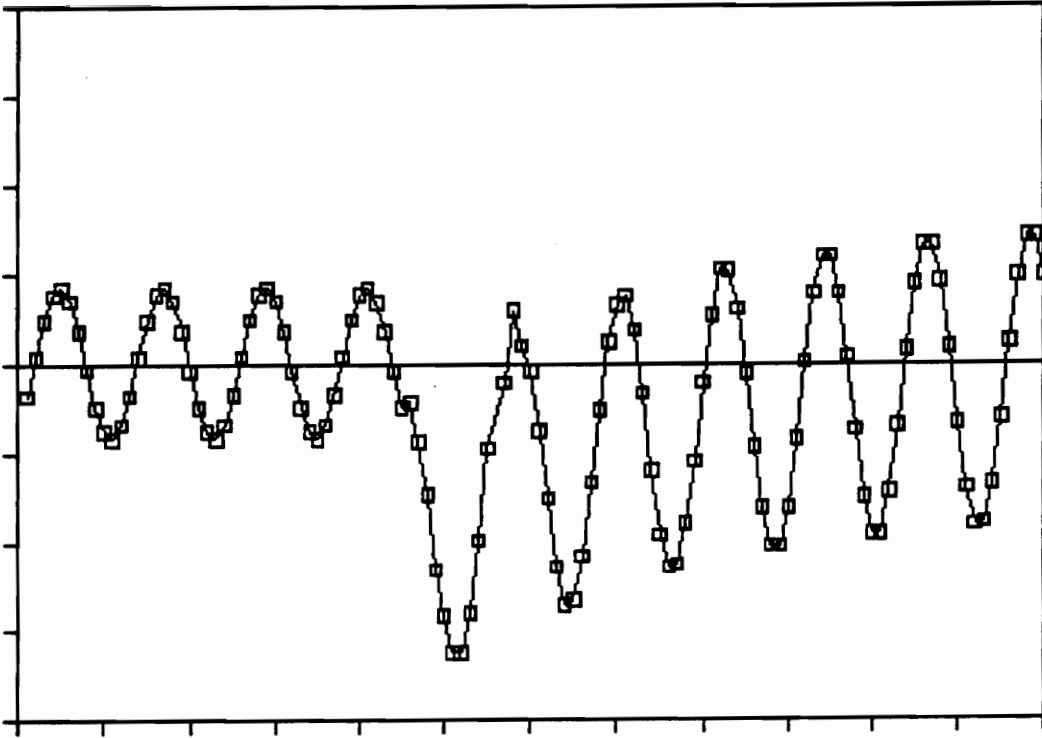


Figure 31. Input signal 225 degrees fault.

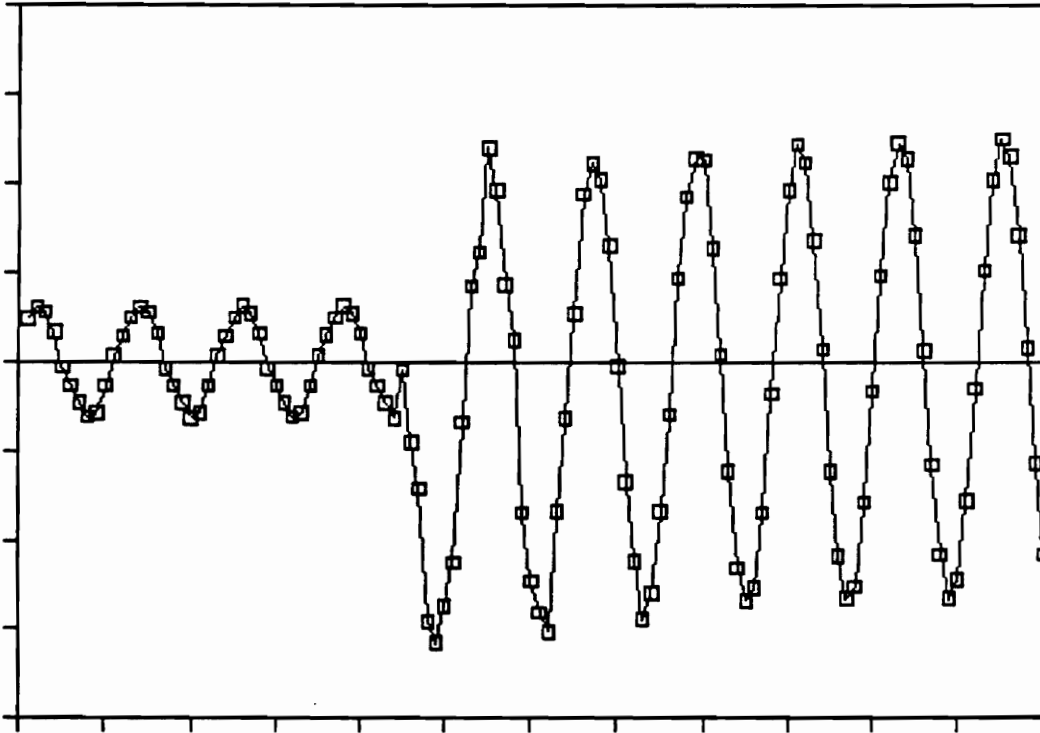


Figure 32. Output analog mimic circuit 225 degrees fault.

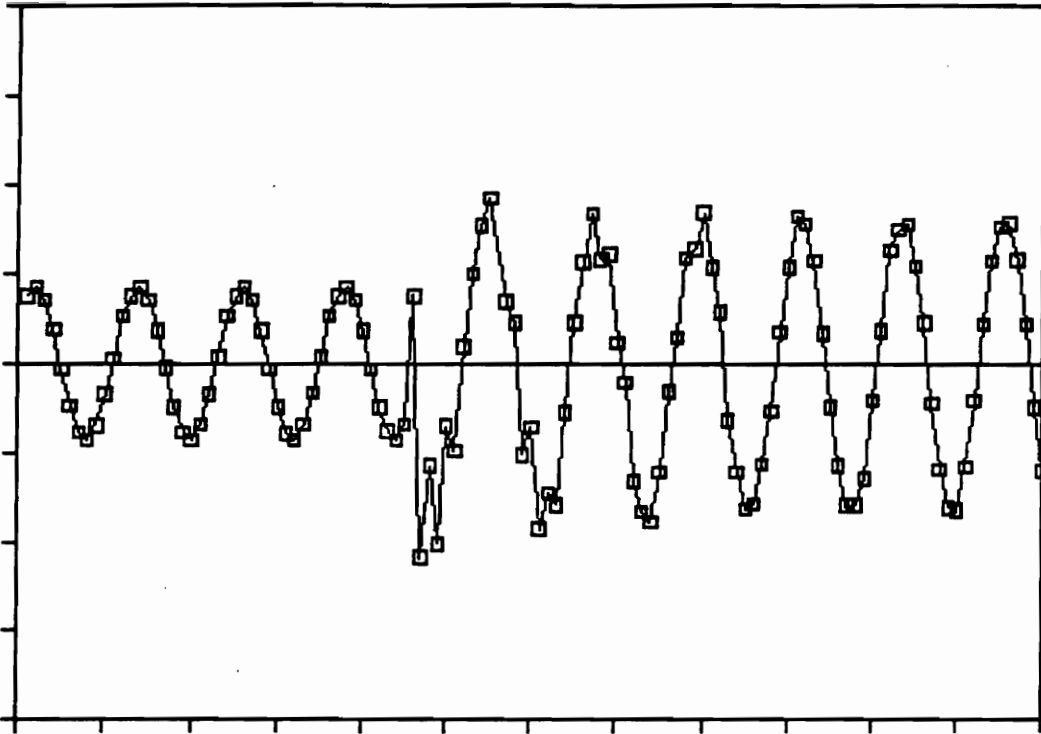


Figure 33. Output 3 sample algorithm 225 degrees fault

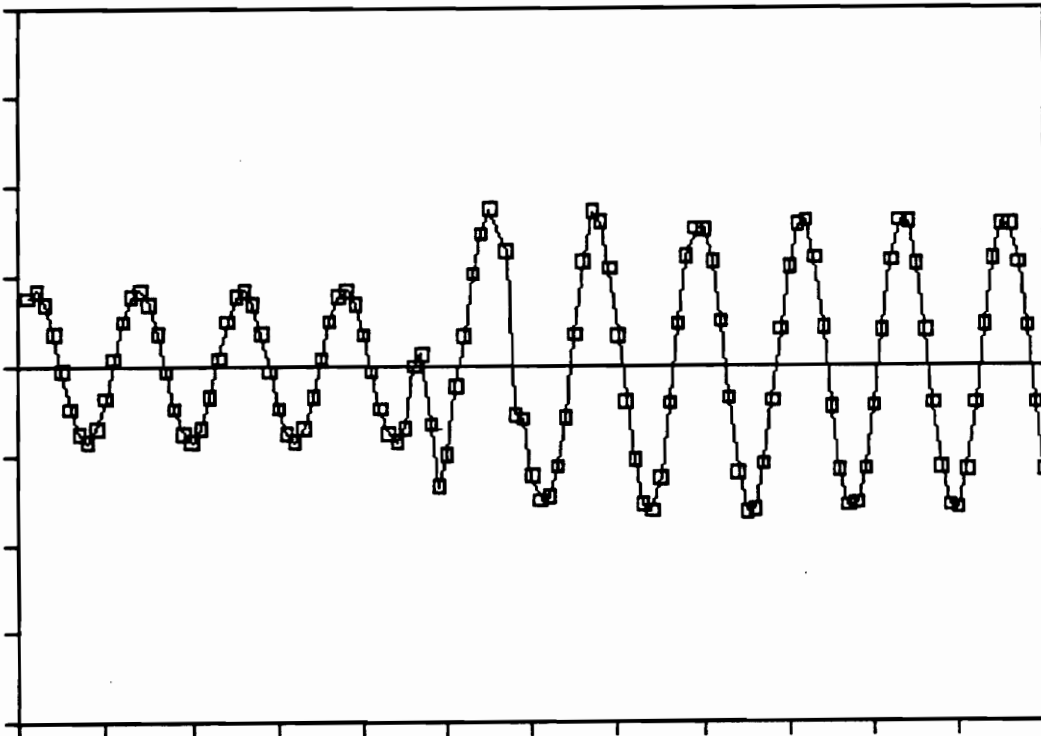


Figure 34. Output 6 samples algorithm 225 degrees fault

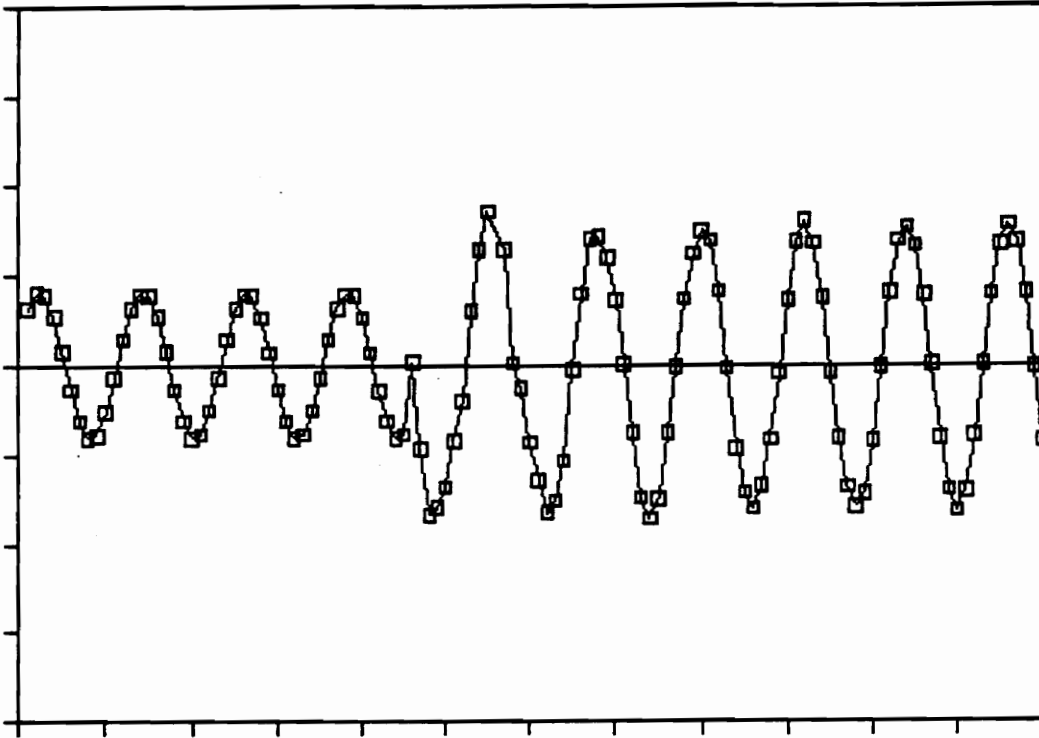


Figure 35. Output 3 samples average 225 degrees fault

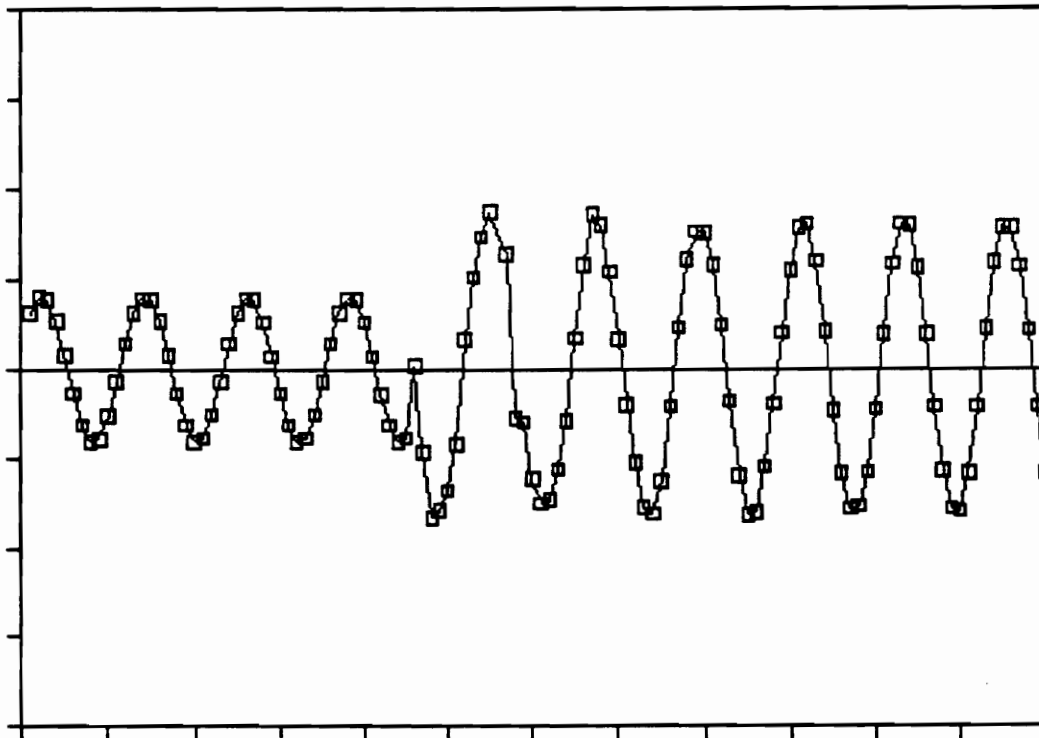


Figure 36. Output 3 samples average to 6 samples 225 degrees fault

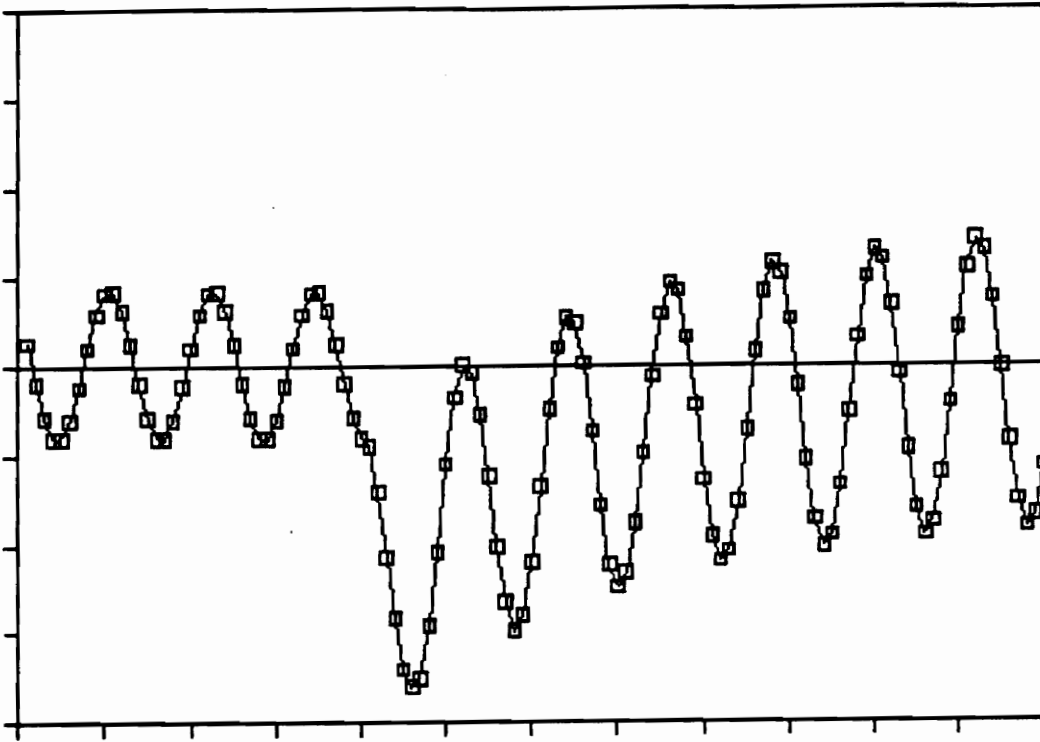


Figure 37. Input signal 270 degrees fault.

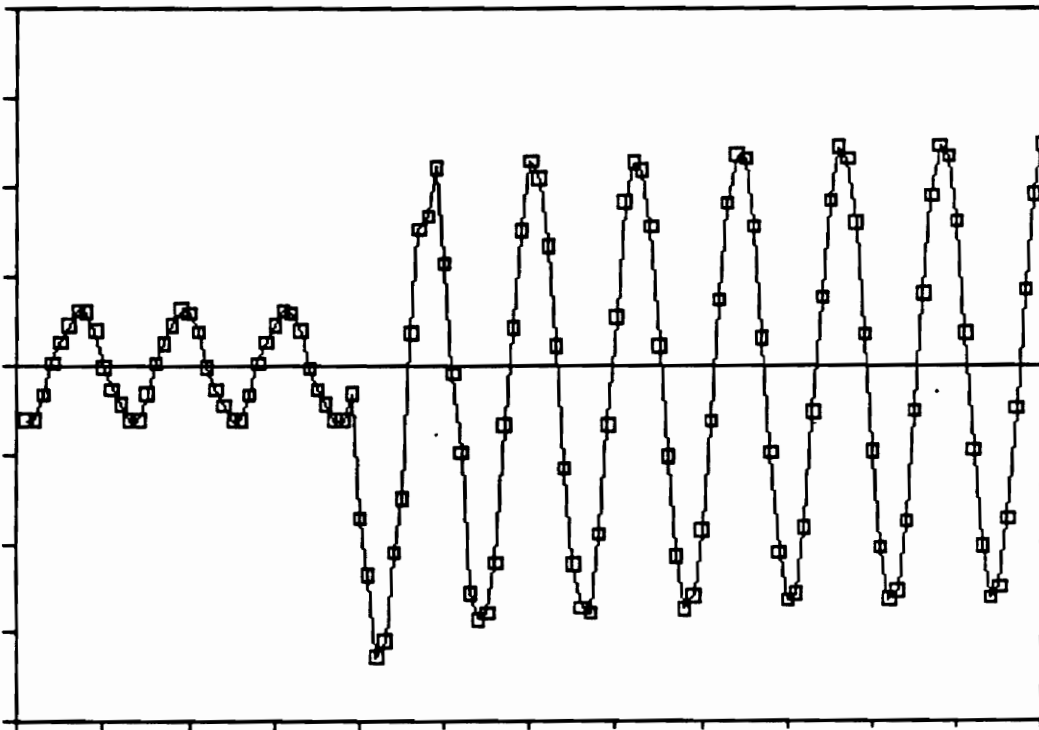


Figure 38. Output analog mimic circuit 270 degrees fault.

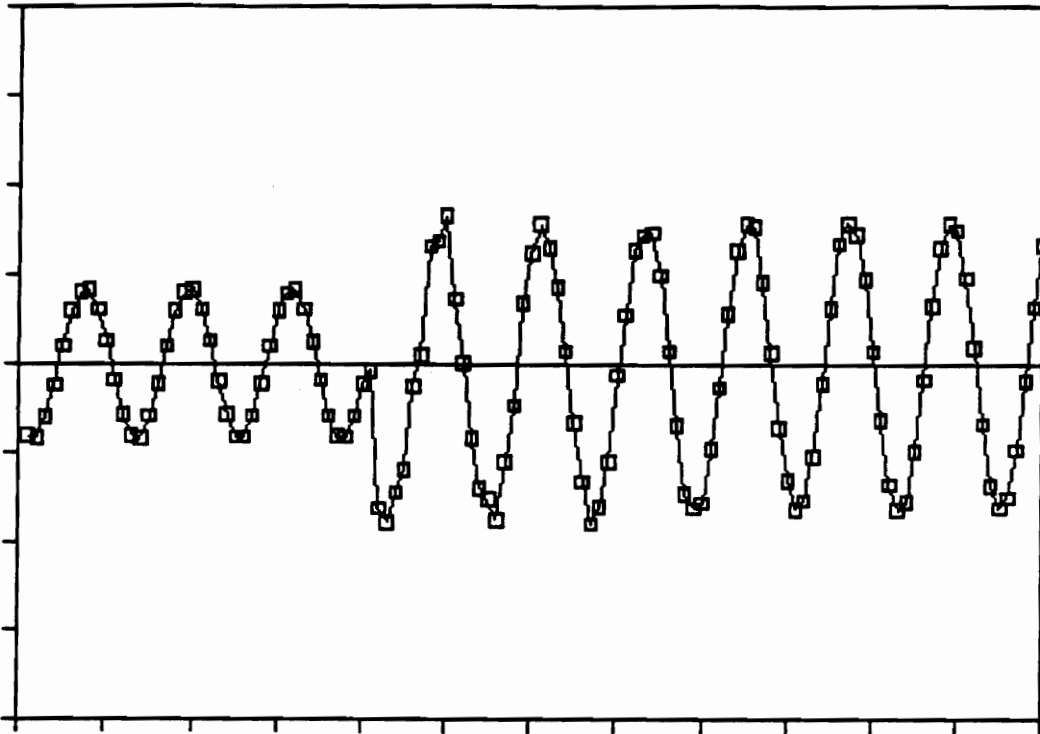


Figure 39. Output 3 sample algorithm 270 degrees fault

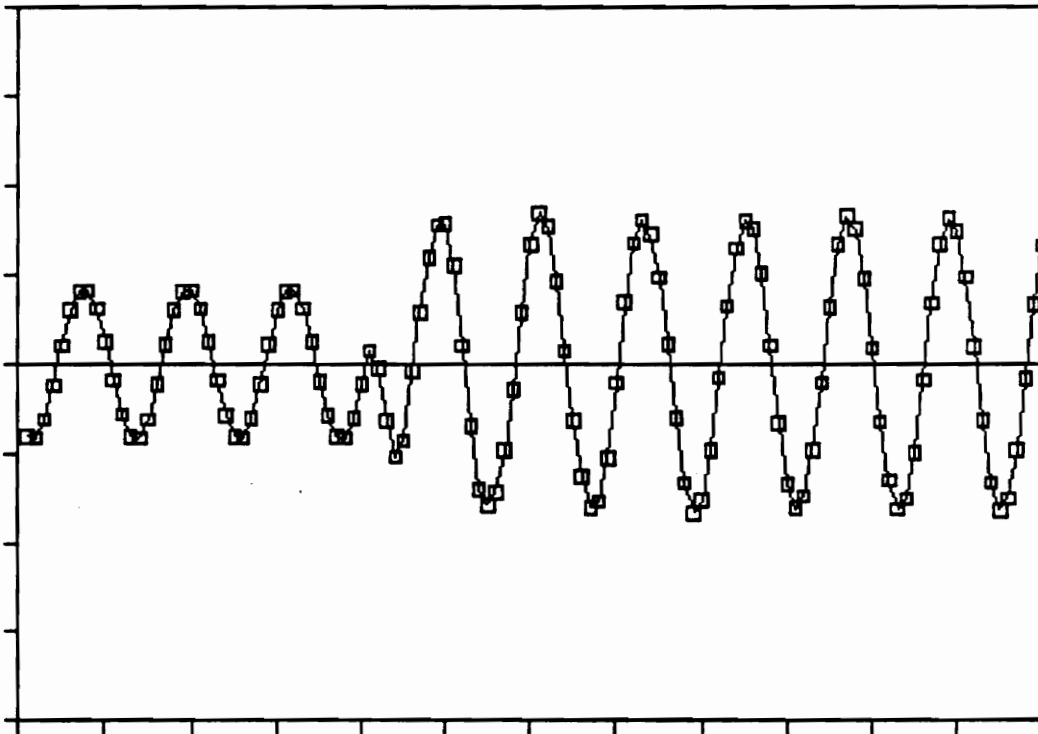


Figure 40. Output 6 samples algorithm 270 degrees fault

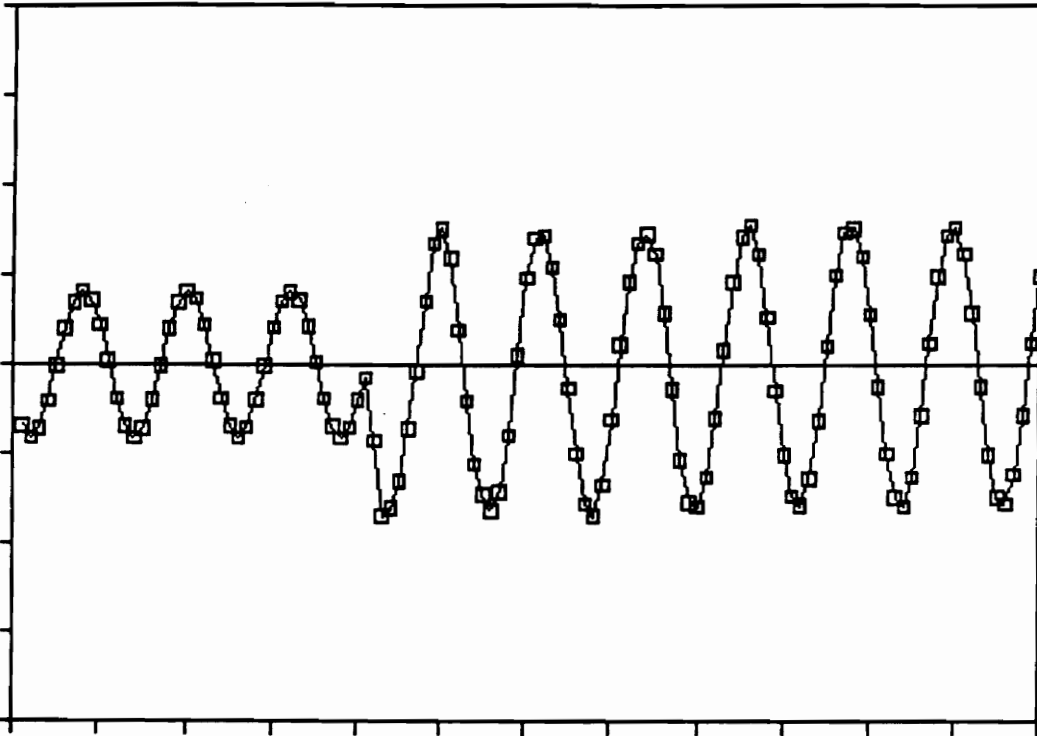


Figure 41. Output 3 samples average 270 degrees fault

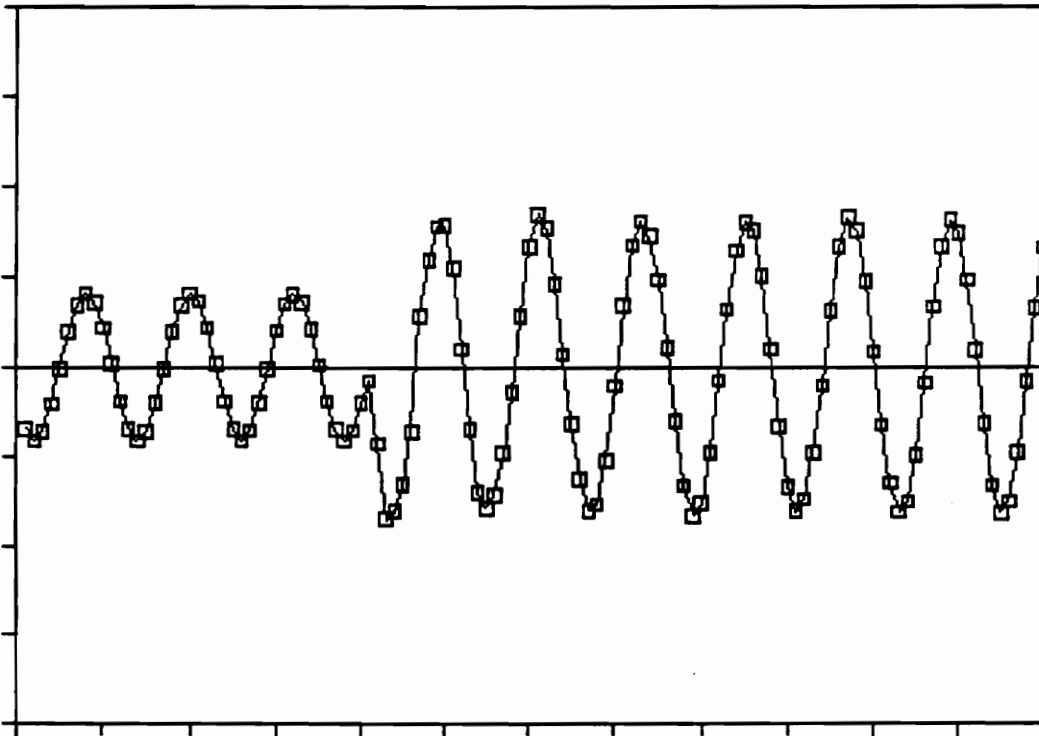


Figure 42. Output 3 samples average to 6 samples 270 degrees fault

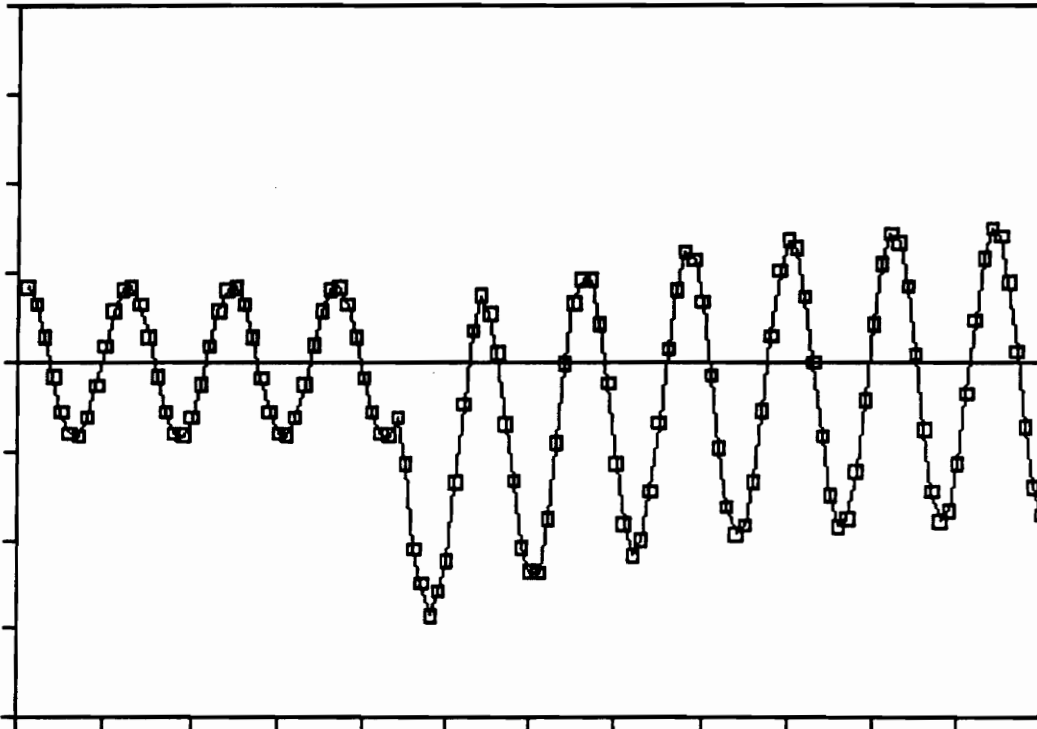


Figure 43. Input signal 315 degrees fault.

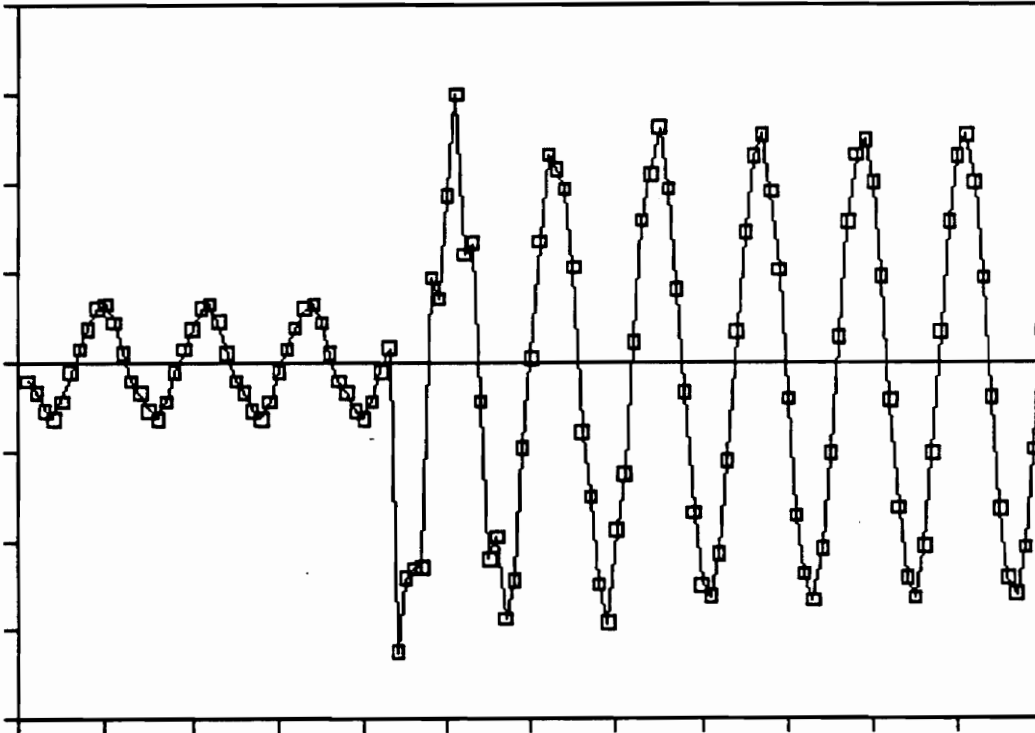


Figure 44. Output analog mimic circuit 315 degrees fault.

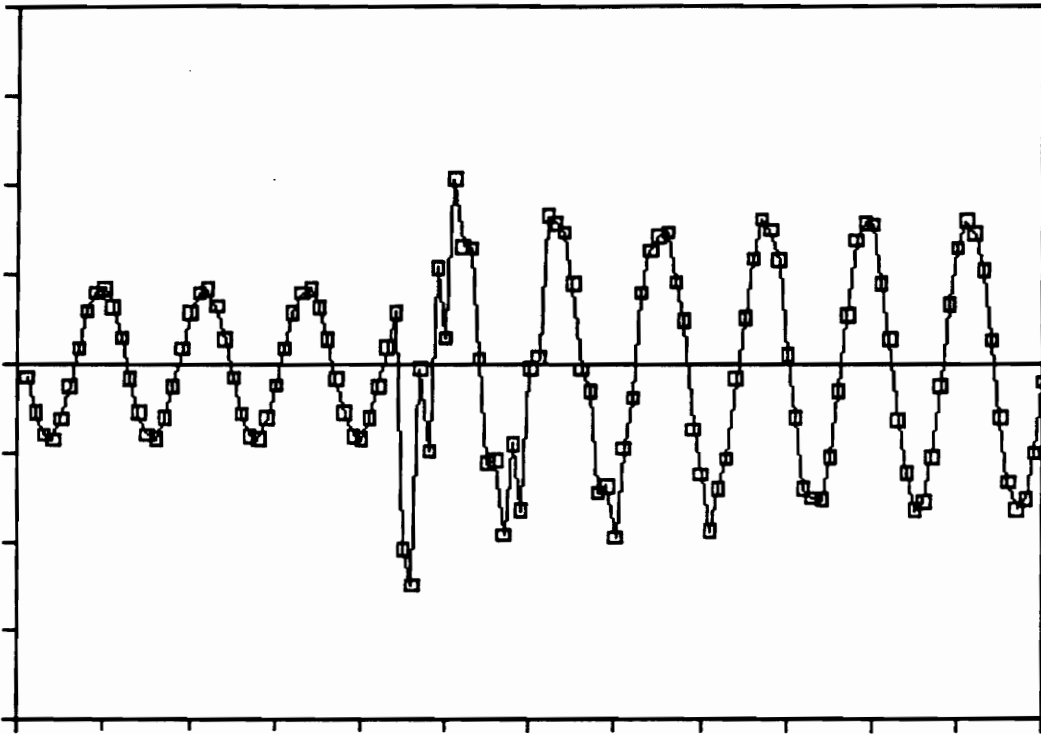


Figure 45. Output 3 sample algorithm 315 degrees fault

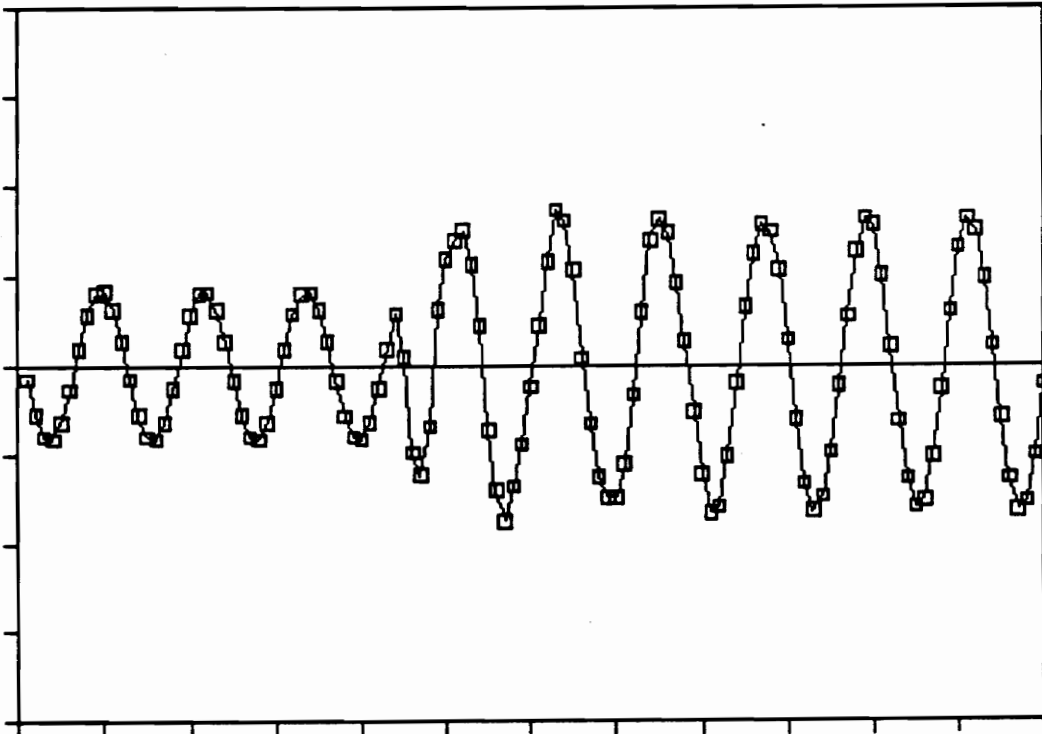


Figure 46. Output 6 samples algorithm 315 degrees fault

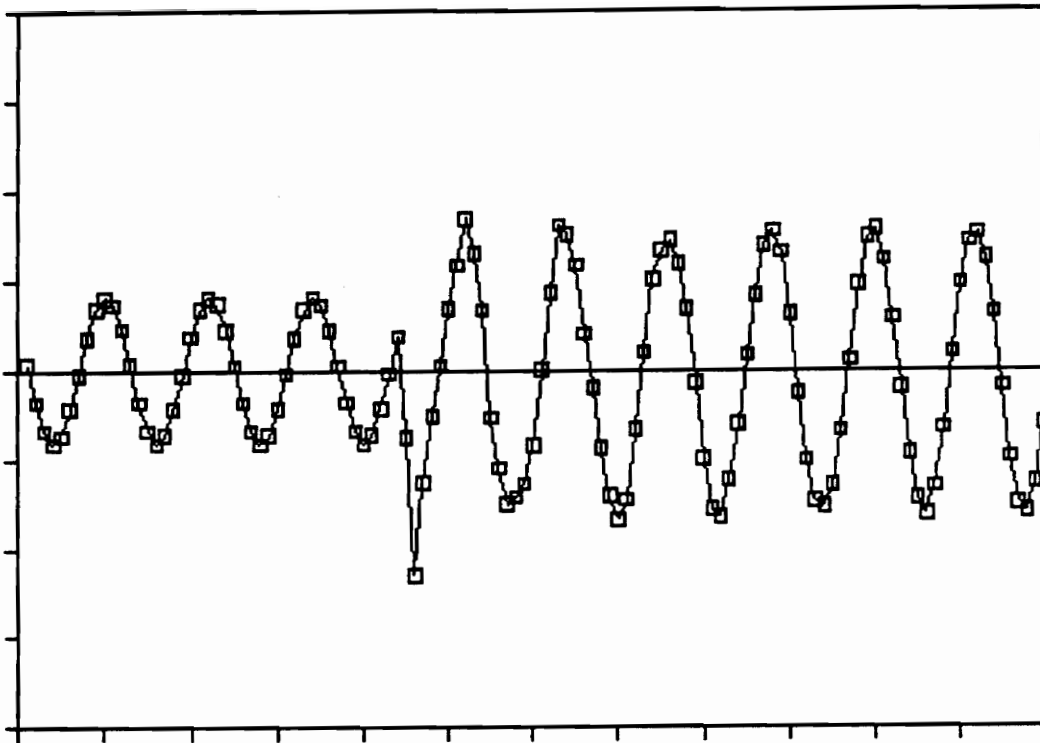


Figure 47. Output 3 samples average 315 degrees fault

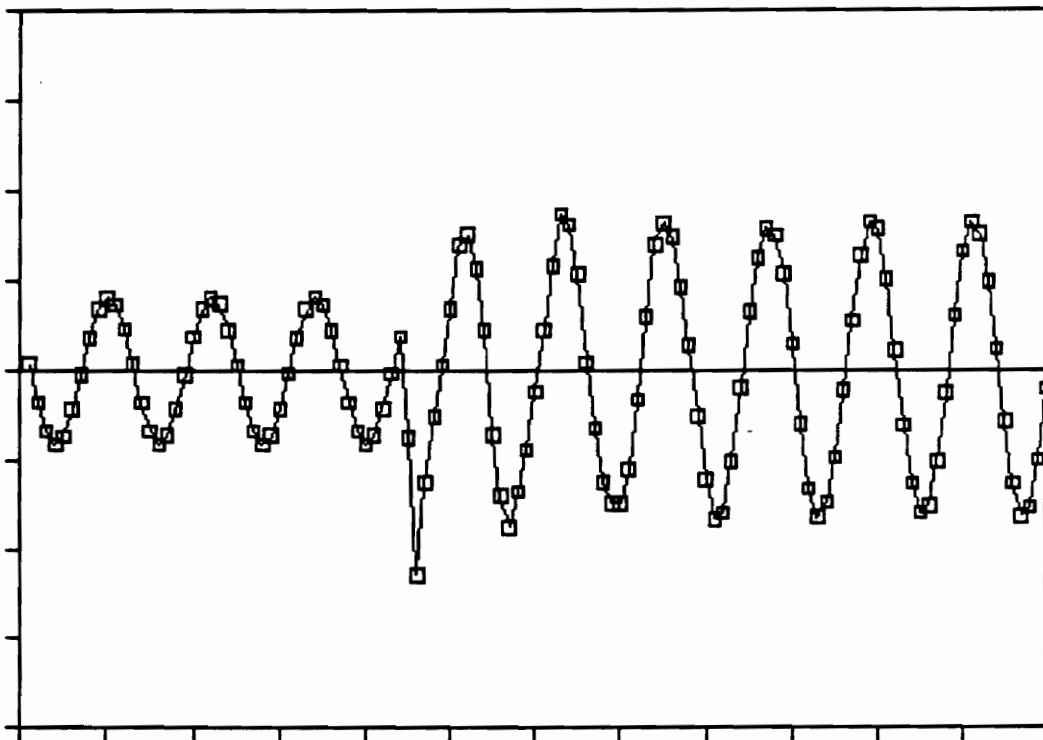


Figure 48. Output 3 samples average to 6 samples 315 degrees fault

CHAPTER 5

CONCLUSION

5.1 PREFERRED ALGORITHM

By studying the plots of the filter algorithm outputs shown in chapter four an evaluation of noise sensitivity and time response is possible for the digital DC offset removal algorithms. It is obvious from the output plots that all the algorithms except for the three sample algorithm perform better than the digital mimic circuit for most of the input signals. The following paragraphs describe the conclusion for each of the algorithms tested.

The three samples algorithm has the fastest response time (Figure 13) but is also the most sensitive to noise. The input signals at low DC offset have a higher

noise content than those close to maximum DC offset. For the low DC offset input the three point algorithm produces a very distorted output signal (Figures 8, 18, 23, 28, 38). The magnitude of the signal exceeds the new steady state value and the sinusoidal characteristic is completely lost for a few samples. This filter is obviously too sensitive to noise to be used for computer relaying.

The six samples algorithm has a slower response time but its noise sensitivity is very low for the same input signals that produced a distorted output for the 3 samples algorithm (Figures 9, 19, 24, 29, 39). The slower response is better seen in the output signal for the low DC offset cases (Figure 14) where the output signal gradually reaches its new steady state value through a period of half a cycle. A faster response around the quarter cycle point is preferable for transmission line relaying.

The three point algorithm with averaging is really a four point algorithm with additional filtering over the 3 samples algorithm. This algorithm averages the previous three point algorithm outputs with the one just obtained. When the averaging is performed an additional phase shift of -15 degrees is introduced due to the 30 degree separation between samples. To correct this phase shift the 3 sample algorithm used has a +15 degree shift, thus when the average is taken the additional phase shift is cancelled. It can be seen from the test results (Figures 10, 20, 25, 30, 40) that this algorithm filters more noise than the three point algorithm. But it is still sensitive to noise with some distortion in the output signal.

Its time response is half a sample period behind the three point algorithm but it is good considering the lower noise sensitivity.

The three sample with average algorithm which changes to six sample half cycle after the fault offers the best compromise among the algorithms results obtained in chapter four. For the operation of this algorithm a transient monitor subroutine is required to determine when to switch from one algorithm to the other. The use of the three point average for normal operation and for the first half of the faulted cycle allows for a fast response to the signal change with some filtering of the noise. The switching to the six point algorithm produces a less noise sensitive signal from which the relay can decide weather to continue, start or stop the trip signal to the circuit breaker (Figures 11, 16, 21, 26, 31, 36, 41).

The three samples with average algorithm which changes to six samples at half fault cycle offers the compromise between noise sensitivity and time response required for fast acting high voltage distance computer relays.

5.2 ADVANTAGES AND DISADVANTAGES

The digital removal of the DC offset by the digital mimic algorithm offers several advantages with few disadvantages over the mimic circuit in the emerging field of computer relaying.

Advantages:

1. It gives a better overall response than the analog mimic circuit.
2. The response time is close to a quarter of a cycle.
3. It has low sensitivity to noise after half a cycle and an acceptable sensitivity for the first half cycle of the fault.
4. Its response in the transition samples, first four samples, is in the same direction and close to the correct values.
5. Its filtering coefficients can be easily changed for new system configurations.
6. The algorithm can be used to filter all the lines monitor by the computer relay.
7. It is free of time or temperature dependency.
8. Does not add any hardware to the computer relay.

Disadvantages:

1. It has higher noise sensitivity for the first quarter of a cycle.
2. It requires the use of a transient monitor subroutine increasing its computational burden.

3. The use of a digital mimic increases the computational burden of the relay algorithm as a whole.

5.3 FUTURE IMPROVEMENTS

The algorithm developed can be improved to reduce its noise sensitivity if the overall time of the computer relay software is reduced allowing for a faster sampling rate. In addition, the availability of a communication port can be used to provide algorithm changes when changes in the system are registered by an external device. The time delay introduced by the algorithm can be taken into consideration as a part of the algorithm to reduce the overall software delay [14]. The algorithm was tested using the most available computer relay hardware and proven to work better than the analog mimic circuit. The performance of the digital mimic algorithm is expected to improve with future hardware which will allow for faster processing, higher sampling frequencies, larger register size and easier programming all of which contribute to a better compromise between response time and noise sensitivity.

If in addition to calculate the signal output for the last sample the algorithm calculates the output for the previous period using the latest samples, transients in the system can be detected. The transient will be determine by comparing the previous filter output with one period delayed output for the new sample. Test

under noisy conditions are required to determine the minimum difference to discriminate between a transient and noise.

5.4 CONCLUSION

The digital removal of the exponentially decaying DC offset from a sampled current signal is not only possible but offers several advantages over the use of analog mimic circuits. The algorithm selected adds an extra benefit to computer relaying whereby no extra hardware is required for its implementation.

APPENDIX A

A.1 6 Sample Algorithm Derivation

Using equation 2.17 with a matrix with $n = 6$ the 6 Sample algorithm is obtained. The same parameters used in the example in chapter two are use in this derivation.

From equation 2.18:

$$e^{-\alpha T} = 0.9118$$

Then

$$Y = \begin{bmatrix} 0.000 & 1.000 & 1.000 \\ 0.500 & 0.866 & 0.912 \\ 0.866 & 0.500 & 0.831 \\ 1.000 & 0.000 & 0.758 \\ 0.866 & -0.500 & 0.691 \\ 0.500 & -0.866 & 0.630 \end{bmatrix}$$

And

$$CG = \begin{bmatrix} 0.000 & 376.991 & 0.000 \\ -188.496 & 326.484 & 0.000 \\ -326.484 & 188.496 & 0.000 \\ -376.991 & 0.000 & 0.000 \\ -326.484 & -188.496 & 0.000 \\ -188.496 & -326.484 & 0.000 \end{bmatrix}$$

From equation 2.17

$$K = CG(J^T J)^{-1} J_T$$

$$K = \begin{bmatrix} -289.952 & 32.965 & 216.992 & 224.289 & 63.358 & -213.142 \\ -247.047 & -18.975 & 131.552 & 171.487 & 96.776 & -66.506 \\ -137.946 & -65.831 & 10.862 & 72.736 & 104.262 & 97.949 \\ 8.117 & -95.048 & -112.738 & -45.505 & 83.811 & 236.160 \\ 152.006 & -98.796 & -206.130 & -151.553 & 40.903 & 311.091 \\ 255.164 & -76.703 & -244.290 & -216.992 & -12.964 & 302.666 \end{bmatrix}$$

From the last row of matrix K we obtain the equation for the six sample algorithm by rounding off to the closest integer.

$$V_0 = 303Y_0 - 13Y_{-1} - 217Y_{-2} - 244Y_{-3} - 76Y_{-4} + 255Y_{-5}$$

A.2 Three Sample with Average

The derivation for this equation is similar to that of the example in chapter 2 with an additional 15 degree shift in matrix G.

From equation 2.18:

$$e^{-\alpha T} = 0.9118$$

Then

$$Y = \begin{bmatrix} 0.000 & 1.000 & 1.000 \\ 0.500 & 0.866 & 0.912 \\ 0.866 & 0.500 & 0.831 \end{bmatrix}$$

And

$$CG = \begin{bmatrix} 326.484 & -188.496 & 0.000 \\ 188.495 & -326.484 & 0.000 \\ 0.000 & -376.991 & 0.000 \end{bmatrix}$$

From equation 2.17

$$K = \begin{bmatrix} -124.857 & -436.713 & 629.126 \\ 573.645 & -1747.568 & 1226.612 \\ 1118.439 & -2590.163 & 1495.428 \end{bmatrix}$$

When the average is taken from two outputs of a three point sample algorithm the following equation is obtained:

$$V_0 = K_0 \frac{(Y_0 + Y_{-1})}{2} + K_{-1} \frac{(Y_{-1} + Y_{-2})}{2} + K_{-2} \frac{(Y_{-3} + Y_{-4})}{2}$$

Using the last row of the matrix K rounding the numbers to the nearest integer the following equation is obtained for the 3 sample average algorithm:

$$V_0 = 1495 \frac{(Y_0 + Y_{-1})}{2} - 2590 \frac{(Y_{-1} + Y_{-2})}{2} + 1118 \frac{(Y_{-2} + Y_{-3})}{2}$$

BIBLIOGRAPHY

1. J.S. Thorp, A.G. Phadke, S.H. Horowitz, J.E. Beehler, "Limits To Impedance Relaying " ; *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-98, No. 1, Jan./Feb. 1979, pp. 246-260.
2. "Microprocessor Relays and Protection Systems " IEEE Tutorial Course Text. Document 88 EH0269-1-PWR, IEEE Service Center Piscataway, N.J. 1987.
3. A.G. Phadke, T. Hlibka, M. Ibrahim, "A Digital Computer System For EHV Substations: Analysis and Field Tests " *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-95, No. 3, Jan./Feb. 1976 pp. 291-301.
4. P.J. Mann, I.F. Morrison, "Digital Calculation of Impedance for Transmission Line Protection " *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-90, No. 1, February, 1971 pp. 270-279.
5. J. Makino, Y. Miki, " Study of Operating Principles and Digital Filters for protective Relays With Digital Computer " Paper No. c75 197 9, presented at the IEEE Winter Power Meeting, New York, N.Y. January 1975, pp. 1-8.
6. ANSI/IEEE C37.90.1 IEEE Guide for Surge Whistand Capability (SWC) for Protective Relays.
7. "Computer Relaying " IEEE Tutorial Course Text Document 79 EH0148-7-PWR, IEEE Service Center Piscataway, N.J. 1979.

8. M.S. Sachdev D.G. Hunchuk, " Performance of Analog To Digital Converters Used In Computer Relaying " Paper No. A 76 154-5, presented at the IEEE Winter Power Meeting, New York, N.Y. January 1976, pp. 1-7.
9. A. Papoulis, " Signal Analysis ", McGraw Hill Book Company, New York, N.Y., 1977.
10. J.G. Gilbert, E.A. Udren, M. Sackin, " Evaluation of Algorithms For Computer Relaying " Paper No. A 77 520-0, presented at the IEEE Summer Power Meeting, Mexico City, Mexico, July 1976, pp. 1-8.
11. C.E. Kim, J.T. Cain, W.C. Guyker, " A Step Towards Identifying The Best Digital Algorithm For Transmission Line Distance Protection " Paper No. A 79 414-4, presented at the IEEE Summer Power Meeting, Vancouver, Canada, July 1979, pp. 1-8.
12. R.L. Dellinger, " Transmission Line Simulation For Transient Network Analyser At Virginia Tech " Thesis of Master of Science in E.E. Virginia Tech, Blacksburg, VA. December 1983.
13. A. Ahluwalia, " Sixteen Pole Synchronous Switch " Thesis of Master of Science in E.E. Virginia Tech, Blacksburg, VA. May 1983.
14. B.S. Ashok Kumar, " Time Delay Compensation For High Speed Digital Protection " ; presented at the IEEE Winter Power Meeting, New York, N.Y. Feb. 1986, pp. Wm 114-3.

VITA

V. Centeno was born in Tegucigalpa, Honduras on September the Twenty first, Nineteen Hundred and Sixty Three. He received the degree of Bachelor of Science in Electrical Engineering from Virginia Polytechnic Institute and State University, in June 1985, and joined as a graduate student at the same institution in July 1985.