GER-3962

Advancements in Adaptive Algorithms for Secure High-Speed Distance Protection

Advancements in Adaptive Algorithms for Secure High Speed Distance Protection

M. G. Adamiak
G. E. Alexander
GE Power Management
Malvern, PA

Dr. W. Premerlani
GE Corporate Research and Development
Schenectady, NY

INTRODUCTION

Numeric transmission line distance protection systems have been widely applied in recent years primarily because of their monitoring and communications capabilities rather than for improved performance of the protection functions. Typical tripping times for digital distance relays range from one to 3 cycles, while state of the art relays using analog signal processing techniques offer trip times of one-quarter to one cycle. A previous paper [1] discussed some of the time delays associated with typical digital processing used in digital distance relays as well as the affect of higher sampling rates.

Recent developments in adaptive algorithms and the use of higher sampling rates combine to provide secure high speed protection not available with previous implementations. These advancements are in both the area of phasor calculation and the protective algorithm implementation.

TRADITIONAL FOURIER CALCULATION

Full Cycle Discrete Fourier Transform

Many analog and digital distance relays use phasors as the operating signals in the distance functions. The phase angle comparator is a well known operating principle that uses the phasor information contained in the input signals. A digital filter that both removes the non-fundamental frequencies and also provides phasor information is therefore desirable for a digital implementation of a phase angle comparator distance relay. One such filter which is widely used in digital distance relays is the Discrete Fourier Transform which is considered in this paper.

A steady state voltage signal in the time domain can be described by the equation:

$$v(t) = V_{peak} \bullet \cos(\omega t + \theta)$$

In a digital relay, this signal is sampled N times per cycle. Thus the input signal can be represented by a series of samples, V_k, where k = 0 to N-1.

Digital filters, such as those discussed in this paper, process the sampled data points, V_k, by multiplying each sample by one or more coefficients determined by the type of digital filter employed. In the traditional Fourier calculation, each sampled value is multiplied by a sine term and a cosine term. The Discrete Fourier Transform Calculation of the fundamental components can be defined by the following equations.

$$\text{Vreal} = \left(\frac{2}{N}\right) \bullet \sum_{k=0}^{N-1} \left[\text{Vk} \bullet \cos\left(2 \bullet p \bullet \frac{k}{N}\right)\right]$$

$$\text{Vimag} = \left(\frac{2}{N}\right) \bullet \sum_{k=0}^{N-1} \left[-\text{Vk} \bullet \sin\left(2 \bullet p \bullet \frac{k}{N}\right)\right]$$

The magnitude of the voltage phasor can be calculated by the following equation.

$$\text{Vmag} = (\text{Vreal}^2 + \text{Vimag}^2)^{1/2}$$

The phase angle of the voltage phasor can be calculated by the following equation:

$$\text{Vangle} = \arctan(\text{Vimag}/\text{Vreal})$$

$$\text{Vangle} = \theta_V$$

Note that depending upon the design of the digital filter, the angle θ_V may be constant, or it may rotate 360/N degrees with each new sample.

With these definitions, the Fourier Transform Calculation is able to convert the sinusoidal voltage waveshape to a phasor. The phasor is represented by two forms, the first form is the rectangular form where the real and imaginary components define the phasor; the second form is the polar form where the magnitude and the phase angle define the phasor.

Recursive Vs Non-recursive Filters

There are two methods of calculating the Discrete Fourier Transform: Recursive and Non-recursive. The Non-recursive method requires that each sampled data point be saved in memory (amount of data is determined by the "window" size) and that the entire coefficient multiplication and summation process be performed every sample. The newest sample becomes the Nth sample, the oldest sample is dropped from the calculation. The real and imaginary terms must be recalculated from the beginning. One implementation of the Recursive method requires that the product of the sine and cosine coefficients and the sample data values used to generate the sums are saved (the amount is still determined by the "window" size) and an abbreviated summation process is performed. In this method, the oldest product is removed from the sum and the newest product is added into the sum. With this Recursive implementation, only the product for the newest sample needs to calculated instead of recalculating the values for all the samples in the "window". This reduces the amount of calculations performed. Therefore, the time required to complete this process is also reduced enabling the relay to perform additional tasks or increase its sampling rate. The Non-Recursive method, on the other hand, requires more time and/or computing speed to complete.

Half Cycle vs. Full Cycle Fourier Calculation

The Discrete Fourier Transform has the capability of working on different sized "windows". The Full Cycle window generates the sums using all the sampled data collected over the last cycle. This means that the "window" includes the last full cycle's worth of data. The Half Cycle window generates the sums using the sampled data collected in the last half cycle. Therefore, the data "window" is a half cycle. Using a Half Cycle window allows the Discrete Fourier Transform to more quickly track a change in the sampled data than is possible with a Full Cycle window. However, there are differences in the filtering actions of the Half and Full Cycle filters. For example, the Half Cycle Fourier is subject to errors due to dc offset and even harmonics of the fundamental frequency. Both the Full and Half Cycle Fourier may be implemented as either a Recursive or Non-recursive filter.

A NEW APPROACH TO PHASOR CALCULATION

High Speed Sampling

The effect of the sample rate on the operating time of a generic distance relay was discussed in [1]. It was noted that the improvement in the relay operating times for sampling rates higher than 16 samples per cycles was not substantial. In addition, the higher sampling rate required more computing power (higher cost) in order to process the data in less time. At a sampling rate of 4 samples per cycle on a 50 Hz. system, 5 milliseconds is available between samples to process the data and run the protection algorithms; if the sample rate is increased to 64 samples per cycle, the time between samples is reduced to only 0.3125 ms.

A raw sampling rate of 64 samples per cycle is now feasible, and desirable for the increased fidelity of the oscillography data included with the digital relays. This raises the issue of what to do with the extra data samples if the protection is run only 16 times per cycle. The first option was to use every fourth sample for protection, and to use the remaining samples only for the oscillography data. This approach could result in aliasing problems due to the higher order frequency components resulting from the increased sampling rate. As a result, it was decided to use all of the sampled data, but to run the protection algorithms only 16 times per cycle. Sets of four data samples are processed together to form a “mini phasor” or “phaselet”.

Phaselets

Phaselets are partial sums of the product of the waveform samples and the sine/cosine coefficients. Groups of phaselets may be scaled and added together to create a phasor. Phaselets enable the efficient computation of phasors over sample windows that are not restricted to an integer multiple of a half cycle at the power system frequency. In the case of a data window that is a multiple of a half cycle, the computation is exactly equal to the Discrete Fourier Transform. In the case of a window that is not a multiple of a half-cycle, there is an additional correction that results from the sine and cosine functions not being orthogonal over such a window. However, the computation can be expressed as a two by two matrix multiplication of the sine and cosine weighted sums. A detailed description of the phaselet calculations is presented in Appendix A.

Digital Mimic

The inductive behavior of power system transmission lines gives rise to decaying exponential offsets during transient conditions. Processing of data containing this offset component will result in an oscillation of the phasor (magnitude and angle) [1]. It is therefore very desirable to remove the exponential offset prior to the phaselet calculation.

Using analysis, simulations, and several references, several alternatives for the removal of the decaying exponential offset were considered, including various least squares fits and a digital mimic. Simulations and analysis showed that the mimic is the best approach, effectively removing the decaying offset without introducing noise artifacts.

The mimic computation is applied to the raw data samples for each phase current. The output of the mimic calculation is the input for the phaselet computation.

Variable Window

The traditional approach to the calculation of the Fourier employs a "sliding data window" (typically a half or full cycle). When a fault occurs, the sliding data window includes pre-fault data along with fault data. Subsequently, the phasor estimation, (and the distance relay) will have an inherent transient time delay that is a function of the window size, as discussed in [1]. The concept of a "variable window" has been developed to improve the response of the phasor estimation, and as a direct result to speed up the operating time of the distance relay.

In steady state, non-fault, conditions, a one cycle window of data is used. The phaselets are summed over one cycle, creating the equivalent of a one cycle window DFT. When a disturbance is detected on the power system, the window size is dynamically reduced to the width of a single phaselet. As new phaselets are obtained, the window size is increased to include the new data. Because all of the pre-fault data has been removed from the window, the phasor estimate responds more quickly to the state of the power system, and the accuracy of the estimate improves as each new phaselet is added. The data window continues to expand until it reaches a full cycle at which point the window reverts to a sliding window similar to the conventional DFT.

ADAPTIVE TECHNIQUES

The use of numeric techniques for protective relays has allowed the protection algorithms to become adaptive to conditions on the power system. The use of these adaptive algorithms can increase both the security and the dependability of the protective system compared to similar electronic and electromechanical relay systems. Several of these adaptive techniques will are discussed in the following sections.

Adaptive Polarizing Memory

Some form of polarizing voltage memory is a common feature for mho distance relays. The memory voltage serves three main functions: it allows the distance relays to operate for zero voltage three phase faults in front of the relay, it prevents the relay from operating for zero voltage three phase faults behind the relay, and it give the relay a variable characteristic [3]. In the past, the duration of the memory voltage was typically for a predetermined time, or for an "infinite" time. One possible problem area with the fixed memory time is for faults beyond the reach of the Zone 1 functions. On lines with high source to line ratios, magnitude of the steady state fault voltage at the relay for three phase faults at the remote end of the line may be less than the voltage required for the relay to operate. For these conditions, the overreaching step distance backup functions may not operate if the time delay is greater than the fixed memory time. In digital relays, the memory time can be made adaptive based on the fault duration. One possible adaptive memory logic scheme is described below.

If the positive sequence voltage is less than 10% of rated during a fault, the relay will continue to use the pre-fault memory voltage to polarize the distance functions. The pre-fault memory voltage will be used until the positive sequence voltage increases above 10%, or until the fault detector resets. Note that the fault detector is sealed-in when any distance function is picked up. If the relay uses positive sequence voltage polarizing for all distance units, this change will not affect the performance for other than three phase faults because the magnitude of the positive sequence voltage will be above 10% of nominal (≈7 V rms) for all unbalanced faults.

Adaptive Reactance Supervision

Reactance characteristics must be supervised to prevent operation under load conditions. This supervision may be provided by a mho distance characteristic, or the reactance characteristic may be one boundary of a quadrilateral characteristic. The quadrilateral characteristic limits the resistive reach by the use of resistive blinders. Traditionally, these supervising functions are set based on the minimum load impedance that the relays might see in service. An example of this setting is shown in Figure 1; where the reach is limited by the minimum load impedance which plots at point X. Under light load conditions, such as indicated by the load impedance plotted at X′, the resistive coverage is still limited by the expected maximum load flow. With digital relays it is now possible to modify the supervising functions based on the load flow existing on the line rather than the expected maximum load flow. This allows the function to have the maximum fault resistive coverage for any given load flow. On approach using a mho distance function with an adaptive reach is shown in Figure 1. The relay continually monitors the load impedance and adjusts the reach setting of the supervising mho distance function to provide the maximum resistive coverage while maintaining a safe margin from the load impedance.

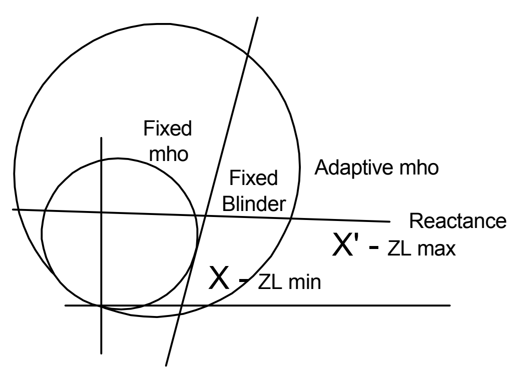

Figure 1

Adaptive Zone 1 Reach

Because a Zone 1 distance function is an unsupervised direct tripping unit, it must be designed to have minimum transient overreach so that it can be set to cover the largest possible percentage of the protected line, typically 90%. Capacitive coupled voltage transformers (CCVTs) are a common source of transients in the voltage signal which may cause overreach of a Zone 1 distance function. Oscillations in the initial Fourier calculations for sub one cycle windows may also introduce errors in the current and voltage phasors. These errors in turn may cause transient overreach. Various solutions to the problem have been used in the past such as additional filtering or added time delay. Typically, this minimum overreach requirement will result in operating times for severe close in faults that are longer than desired. In order to overcome these errors and at the same time provide the fastest possible operating times, an adaptive Zone 1 distance function has been introduced.

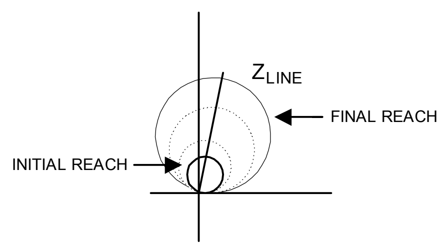

Figure 2

Unlike a traditional distance relay, the reach of the adaptive Zone 1 distance function is not constant. When a disturbance is detected on the power system, the reach of the Zone 1 is set to a minimum value (0-35% of the line impedance). As each new phaselet of current and voltage are added to the Fourier calculation, the reach of the Zone 1 function is increased as the confidence in the measured phasors increases. This approach produces high speed operation for faults occurring near the relay which will typically have the greatest affect on system stability, and slower operation for remote faults which are less likely to affect the system stability.

COMPUTER SIMULATION

Initial verification of the performance of new Fourier calculation method and of the adaptive Zone 1 reach algorithm was accomplished using a C++ model of the algorithms and fault data captured from an analog model power system. The power system model is shown in Figure 3. Over one hundred faults were studied. The operating times of the Zone 1 distance function provided sub one cycle relaying for fault locations less than 75% of the line. Average operating times for the simulations are shown in Figure 4.

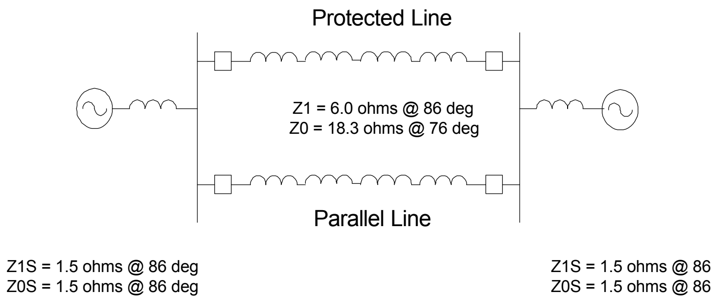

Figure 3

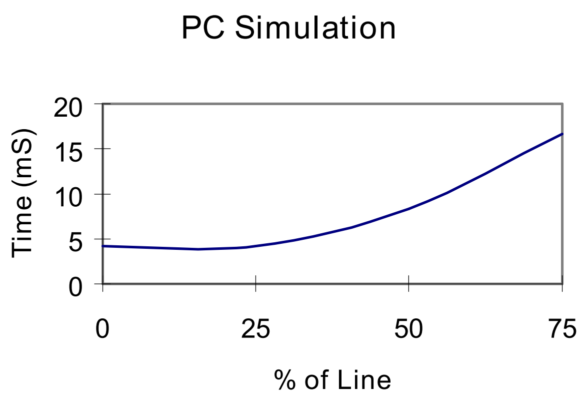

Figure 4

MODEL POWER SYSTEM VERIFICATION

A relay incorporating the new Fourier calculation method and the adaptive Zone 1 reach algorithm was tested on an Analog Model Power System as shown in Figure 3 and subjected to a wide assortment of fault types, fault locations, and load flows. The average operating times for the Zone 1 distance functions are shown in Figure 5. The actual operating times of the relay compare favorably with the operating times predicted by the PC simulation.

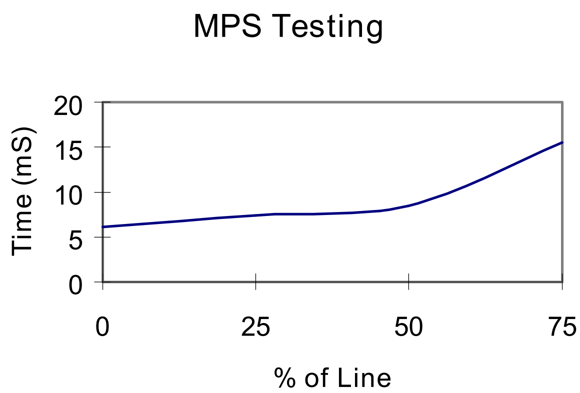

Figure 5

CONCLUSION

Adaptive techniques can now be applied to the calculation of the current and voltage signals used in digital distance relays as well as to the measuring algorithms used in the protection functions. The adaptive features can improve the performance in speed, security, and dependability.

The combination of the Variable Window Fourier calculation and the Adaptive Zone 1 distance function provides substantially faster operating times for previous designs of digital distance relays. The operating times of this new generation of digital relays are now approaching the times available with state of the art analog designs.

REFERENCES

[1] J. M. Kennedy and G. E. Alexander, " Variable Digital Filter Response Time in a Digital Distance Relay", Twentieth Annual Western Protective Relaying Conference, October 1993.

[2] DLP- Digital Line Protection, GE technical publication GET-8037.

[3] G. E. Alexander, J. G. Andrichak, W. Z. Tyska and S. B. Wilkinson, "Effects of Load Flow on Relay Performance", Thirty-Ninth Annual Texas A &M Relay Conference, April 1986.

APPENDIX A

Phaselets enable the efficient computation of phasors over sample windows that are not restricted to an integer multiple of a half cycle at the power system frequency. Determining the fundamental power system frequency component of current data samples by minimizing the sum of the squares of the errors gives rise to the first frequency component of the Discrete Fourier Transform (DFT). In the case of a data window that is a multiple of a half cycle, the computation is simply sine and cosine weighted sums of the data samples. In the case of a window that is not a multiple of a half-cycle, there is an additional correction that results from the sine and cosine functions not being orthogonal over such a window. However, the computation can be expressed as a two by two matrix multiplication of the sine and cosine weighted sums.

Phaselets and sum of squares are computed for each current from the output of the mimic computations as follows:

$$\text{Phaselet Real}_p = \sum_{k=p\cdot P-P+1}^{p\cdot P} C_k \cdot \text{Imimic}_k$$

$$\text{Phaselet Imaginary}_p = \sum_{k=p\cdot P-P+1}^{p\cdot P} - S_k \cdot \text{Imimic}_k$$

Where:

PhaseletReal_p = the real part of the pth phaselet

$\text{PhaseletImaginary}_p$ = the imaginary part of the pth phaselet

N = Number of samples per cycle

p = phaselet index: there are N / P phaselets per cycle

P = the number of samples per phaselet

Imimic_k = kth sample of mimic output; taken N samples per cycle

The computation of phaselets and sum of squares is basically a consolidation process.

Until a disturbance is detected, phaselets will be combined to form a one cycle sliding window DFT. For a one cycle DFT, the process for computing phasors from is simple, as shown in the following equations:

$$\text{Phasor Real}_n = \frac{2}{N} \cdot \left[\sum_{p=n-\frac{N}{P}+1}^{n} \text{Phaselet Real}_p \right]$$

$$\text{Phasor Imaginary}_n = \frac{2}{N} \cdot \left[\sum_{p=n-\frac{N}{P}+1}^{n} \text{Phaselet Imaginary}_p \right]$$

Phasor Real_n = the real part of the nth phasor

$\text{Phasor Imaginary}_n$ = the imaginary part of the nth phasor

n = the phasor index; there are N / P phasors per cycle

The above equations are defining. The sums involved are not actually computed in the order shown, but are computed recursively. That is, after initialization, the sums at one value of "n" is computed from the previous sums by adding the newest terms of the new sums and subtracting the oldest terms of the old sums.

Converting phaselets to phasors can also be done for other window sizes by adding phaselets and then multiplying by a normalization matrix. First the phaselets are added together over the desired window:

$$\text{PhaseletSumReal}_n = \sum_{p=n-\frac{W}{P}+1}^{n} \text{Phaselet Real}_p$$

$$\text{PhaseletSum Imaginary}_n = \sum_{p=n-\frac{W}{P}+1}^{n} \text{Phaselet Imaginary}_p$$

Where:
W = window size in samples; W/P is the window size in phaselets

Phaselet sums are converted into stationary phasors by multiplying by a pre-computed matrix:

$$\begin{bmatrix} \text{Phasor Real}_n \\ \text{Phasor Imaginary}_n \end{bmatrix} = \begin{bmatrix} T_{RR}(n,W) & T_{RI}(n,W) \\ T_{IR}(n,W) & T_{II}(n,W) \end{bmatrix} \bullet \begin{bmatrix} \text{PhaseletSum Real}_n \\ \text{PhaseletSum Imaginary}_n \end{bmatrix}$$

Note that the matrix elements depend on W, n, P, and N. P and N are design constants. W and n are variables. Matrix elements are pre-computed for each combination of n and W. With 8 phaselets per cycle, there are 64 different matrices.