Real-time Distance Protective Relay on FPGA

Yifan Wang, *Student Member, IEEE,* and Venkata Dinavahi, *Senior Member, IEEE*

***Abstract*— The paper proposes a real-time hardware digital distance protective relay. Taking advantage of inherent hardwired architecture of the Field Programmable Gate Array (FPGA), the proposed hardware distance relay design is paralleled and fully pipelined to achieve low latencies in various relay modules which are developed in textual VHDL language. This hardware distance relay is tested in real-time by feeding it with generated faulted current and voltage data for typical faults, and the relay response was recorded. The results demonstrate the effectiveness of the hardware distance relay.**

***Index Terms*— Digital hardware distance relay, Field programmable gate arrays, Parallel processing, Power system protection.**

I. Introduction

In the context of the smart grid with emphasis on wide-area monitoring, protection and control, there is a growing need for a high-capacity, high-bandwidth protective relay that can cope with the demand of signal processing, intelligence, and communication functions [1]-[3]. In such applications, field programmability (FP) is a desirable feature to have in a protective relay. A FP device, whether analog or digital, is an integrated electronic circuit whose firmware can be reprogrammed in the field after manufacture. Currently, FPGAs are making significant inroads in many applications related to power systems such as real-time simulation, and the control of industrial and commercial systems [4]-[8]. The characteristics of the FPGA that are germane for its use in protection relay application include inherent parallel hardwired architecture allowing an ultra-low latency realization of complex algorithms, very large capacity devices providing substantial hardware resources, mature and advanced design and development tools for rapid prototyping, and fast clock speed and high-speed transceivers to communicate with external devices.

In the industry, protective relays have experienced mainly three generations of evolution [9]: electromechanical (EM) relays, solid-state (SS) relays, and digital relays. EM relays were based on moving parts to perceive abnormal changes of current or voltages to generate the mechanical torque. SS relays based on analog electronic devices such as transistors, diodes, and other electronic components, were the static replacements of EM relays. Currently, most commercial relays are digital or numeric relays based on microprocessor technology, and sequential software programmability. Whilst there may be use of some auxiliary processors, the majority of the functions including the human-machine interface (HMI) of the relay are software programmable, and are implemented on a CPU or DSP [10]. The sequentiality of software architecture inherently limits the operating speed of the designs. Recently new types of protective relays are reported in the literature. In [11] a distance protective relay implementation in field-programmable analog arrays (FPAA) is presented. This approach using a combination of FPAA and digital technologies has the potential to provide better performance than the present-day numerical relays; however, the use of analog blocks in the relay may have limitations on device size, switching noise and sample-rate related bandwidth problems compared with digital technology. In the literature, few works have been done with hardware design of the protective relay on FPGA. Reference [12] designs a hybrid protection scheme for HVDC line implementation, and [13] implemented the wavelet-based directional relay for transformer protection. In these works, the implementation is carried out using vendor-specific schematic blocks, which preclude a generalized and portable floating-point design. The hardware design details are omitted from their description.

In this work, we propose an FPGA-based low-latency high-resolution distance protective digital relay. The design consumes a small percentage of Xilinx Virtex-7 FPGA resources, and it can process data in a few microseconds. It not only uses the parallelism of hardware to speed up but also breaks up the sequentiality in the software-based numeric relay to provide much higher reliability. The organization of this paper is as follows. The operation philosophy and design details of the relay modules on FPGA are provided in Section II. Section III presents a case study for testing the designed hardware distance relay, and experimental results are shown to verify the relay operation. Finally, Section IV presents the conclusions of the paper.

II. Distance Protective Relay Hardware Architecture and Parallel Modules

The overall architecture of the FPGA-based hardware distance relay is shown in Fig. 1. It consists of four main hardware modules: the Discrete Fourier Transform (DFT) processing module, the Coordinated Rotation Digital Computer (CORDIC) processing module, the fault detection module, and the protection elements module. First, by taking in fault voltage and current data through the I/O interface, the DFT module estimates fundamental amplitude and phase of the signals. The CORDIC processing provides trigonometric values that are needed for

Financial support from the Natural Science and Engineering Research Council of Canada (NSERC) is gratefully acknowledged.

The authors are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta T6G 2V4, Canada. Email: {yifan17, dinavahi}@ualberta.ca

978-1-4799-1303-9/13/$31.00 ©2013 IEEE

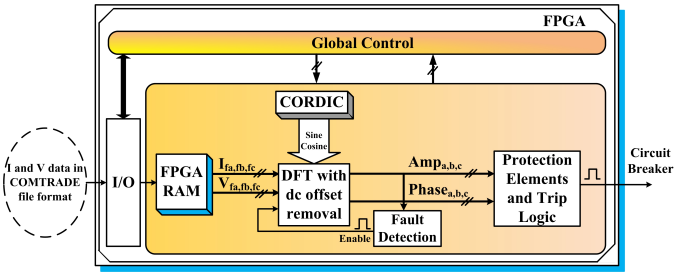

Fig. 1. Overall architecture of the FPGA-based hardware distance relay: functional block diagram.

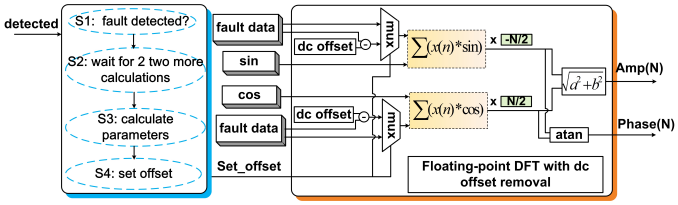

Fig. 2. Pipelined DFT module with dc offset removal.

arithmetic computations. The fault detection module utilizes the DFT results of the currents to detect the inception of a fault based on an over-current protection mechanism, while the protection element module calculates the fault impedance and decides the trip logic. If a fault is detected, and the calculated impedance falls into the protection zone, a trip signal for the associated circuit breaker is sent to isolate the fault. In addition, a global control module is designed to control all the operations in the whole design. In the following subsections, we will discuss each module in detail.

A. DFT Module with dc offset removal

The operation of the distance relay relies on the fundamental phasor estimation of the voltage and current signals at the relay location. Thus, the very first step is to obtain the signal fundamental frequency components. Among the existing filtering algorithms [13]-[18] for fundamental frequency component extraction in digital distance relays, the DFT which has improved harmonic immunity is one of the most widely used methods to obtain the quantities of interest. However, dc offset components contained in fault signals can lead to inaccurate estimation of fundamental phasors by DFT. Therefore, the distance relay tends to overreach or underreach due to the interference of dc offset components. Therefore, the dc offset component has to be removed [20]. Here the algorithm put forward by [21] is adopted as it only requires one cycle plus two samples to finish full-cycle DFT (FCDFT) calculation with dc offset removal. The effect of the additional two calculations due to the two samples can be offset by the low-latency of the FPGA hardware module.

The conventional FCDFT algorithm calculates the fundamental component of a sinusoidal discrete time signal can be described as:

$$\begin{aligned} X_{(1)} &= \frac{2}{N}\sum_{n=0}^{N-1} x(n) \times (\cos\omega_1 n\Delta T - j\sin\omega_1 n\Delta T) \\ &= \frac{2}{N}\sum_{n=0}^{N-1} x(n) \times (\cos\frac{2\pi n}{N} - j\sin\frac{2\pi n}{N}), \end{aligned} \tag{1}$$

where $x(n)$ is the discrete-time sinusoidal input signal, ω_1 is the fundamental angular frequency, N is the number of samples in a fundamental cycle, and ΔT is the sampling interval.

Then, with real and imaginary parts of the fundamental phasor $X_{(1)}$ known, the amplitude and phase of the phasor could be worked out. For dc offset removal strategy in [21], the input signal contains a decaying dc offset $Ae^{-t/\tau}$ (original dc offset amplitude A, time constant $\tau = q\Delta T$). After computing $X_{real(N)}$, $X_{real(N+1)}$, $X_{real(N+2)}$ which are real parts of three FCDFT fundamental phasor calculation results, the two parameters of the dc offset are obtained as:

$$e^{-1/q} = \frac{(X_{real(N+2)} - X_{real(N+1)})cos(2\pi/N)}{(X_{real(N+1)} - X_{real(N)})cos(4\pi/N)}, \tag{2}$$

$$A = \frac{N(X_{real(N+1)} - X_{real(N)})}{2cos(2\pi/N)e^{-1/q}(e^{-N/q} - 1)}. \tag{3}$$

The dc offset then can be subtracted from the sampled signal $x(k)$ as:

$$z(k) = x(k) - Ae^{-k/q} \tag{4}$$

where $z(k)$ is the discrete signals without dc offset.

Fig. 2 shows the FPGA hardware design of DFT module with dc offset removal. The DFT calculation module gets the *set_offset* control signal from the finite state machine controller which has four transition states going from $S1$ to $S4$. The controller senses the fault *detected* signal, and waits for two more sample calculation cycles to finish offset parameter calculation. Once the calculation is done, a *set_offset* signal is sent to the DFT calculation module. In each DFT computation cycle, with collection of one full-cycle window of fault data, and *sin* or *cos* values, imaginary and real part of the fundamental are calculated. After taking square root of the sum of imaginary and real part squares, and taking the *arctangent* of imaginary part over real part, the amplitude and phase of the set of data are generated. Once the calculation is completed, the data window moves forward by one sampling point to the next computation cycle. Within the module, all the computation elements use IEEE single-precision floating-point 32-bit format and are fully pipelined to enhance throughput.

B. CORDIC-based Trigonometric Function Evaluation Module

The trigonometric function values required by the DFT process are generated from the iterative CORDIC module [22] [23]. The CORDIC method has the advantage of higher precision over the conventional look-up-table (LUT) based method. Because usually the desired trigonometric or nonlinear function values are stored in LUTs

whose length and precision are limited by the volume of ROM limits.

Vectoring and rotation modes are generally the two ways to implement the CORDIC algorithm [22]. In this paper, the rotation mode is chosen to estimate trigonometric values of the input angles. In the rotation mode, the initial vector (x_0, y_0) starts aligned with the $x-$axis and is rotated by a micro angle of θ_i which has the tangent value of 2^{-i} in every iteration. After n iterations, the desired angle is achieved, and the y and x coordinate of the vector at that moment reflect the cosine and sine value of the angle respectively. The following equations can be computed in each iteration:

$$\begin{cases} \sigma_i & = & \pm 1 \\ x_{i+1} & = & x_i - \sigma_i y_i 2^{-i} \\ y_{i+1} & = & y_i + \sigma_i x_i 2^{-i} \\ z_{i+1} & = & z_i - \sigma_i \arctan 2^{-i} \\ i & = & i+1, \end{cases} \tag{5}$$

where x_i, y_i are the x and y coordinates of the i_{th} vector, z_i is the angle in i_{th} iteration, σ_i is the sign of angle needed to rotate in i_{th} iteration, decided by the difference between the desired and current angles.

In addition, there's a scaling effect in the rotation process in each iteration which can be represented in factor K as:

$$K = \prod_{i=0}^{n} K_i = \prod_{i=0}^{n} 1/\sqrt{(1+2^{-2i})}. \tag{6}$$

Value of K is used to multiply the final x and y at the end of the iteration.

The design of the CORDIC algorithm on FPGA has a pipelined architecture to improve speed and throughput. Fig. 3 shows how each CORDIC iteration is implemented in hardware. Scale factor and arctangent values of 2^{-i} are pre-calculated and stored in memory. Then, in each iteration, there are only addition/subtraction and shifting operations (multiplication by 2^{-i} can be done by 1-bit shifting to the right), which can be easily achieved in hardware as shown in Fig. 3. The iterative stage n can be set manually. In this design, n is set to 20 for higher accuracy.

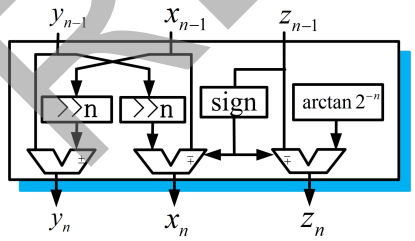

Fig. 3. FPGA design of CORDIC module in one iteration.

C. Fault Detection

The fault detection module detects the initiation of the fault. The DFT filtering based method is adopted to apply the detection algorithm. Although the algorithm suffers from some drawbacks such as its sensitivity to even harmonics and decaying dc components in the fault signals, the presence of harmonics does not significantly affect the fault detection [24] [25]. If the fundamental amplitude of any phase current extracted from DFT exceeds a certain threshold three times consecutively, this indicates that the transmission line is exposed to an abnormal situation, and a detection signal is sent out.

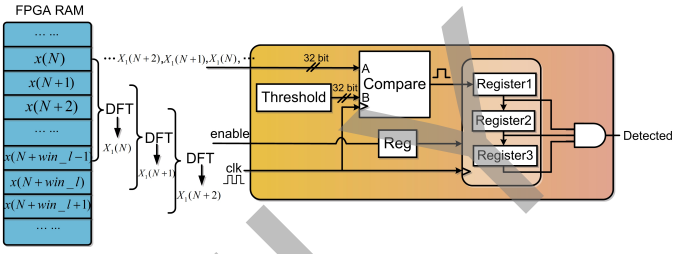

Fig. 4. Fault detection hardware module.

The hardware implementation details of the fault detector are shown in Fig. 4. Previous DFT amplitude results enter into the module, and are compared with the threshold value. If three successive amplitudes exceed the threshold, the three single-bit registers which record the comparison results will become '1'. Then a fault detection signal would be generated.

D. Distance protection elements

The FPGA-based hardware digital relay was designed to protect a three-phase transmission line with six impedance measuring elements: three phase-to-ground relays and three phase-to-phase relays. Table I presents apparent impedance calculations for different fault types.

TABLE I
IMPEDANCE EQUATIONS BASED ON DIFFERENT FAULT TYPES

Relay elements	Impedance formula
A-ground	$V_A/(I_A + 3k_0 * I_0)$
B-ground	$V_B/(I_B + 3k_0 * I_0)$
C-ground	$V_C/(I_C + 3k_0 * I_0)$
A-B	$(V_A - V_B)/(I_A - I_B)$
B-C	$(V_B - V_C)/(I_B - I_C)$
C-A	$(V_C - V_A)/(I_C - I_A)$

I_0 is zero sequence current calculated from $(I_A + I_B + I_C)/3$; Z_0 and Z_1 are zero and positive sequence line impedances from relay location to protection zone respectively; $k_0 = (Z_0 - Z_1)/Z_1$.

The impedance calculated from these six relay elements is then processed by a mho characteristic [26] relay element which is based on phasor comparison to decide which protection region it is in. Fig. 5 shows the mho relay characteristic with three protective zones.

Applying different zone protection can be achieved by setting different diameter of the circles. The relay calculates the angle between polarizing vector V_p and operating vector $V_o = IZ_R - V_p$, where V_p equals to fault voltage and acts as reference, I is fault current and Z_R is the relay setting. Consider Zone 1 protection. As seen from

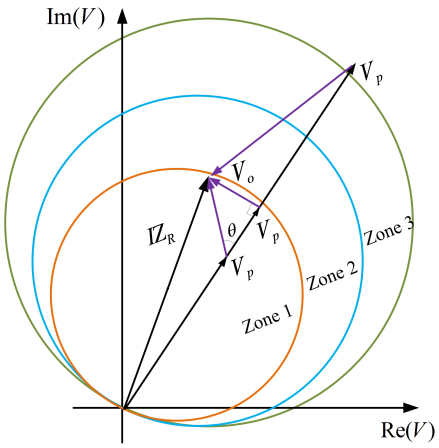

Fig. 5. Mho characteristic for 3 zone protection.

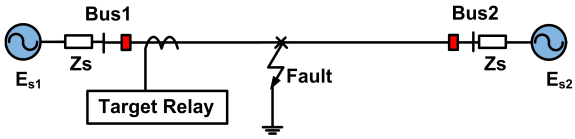

Fig. 6. Single line diagram of test power system.

Fig. 5, if the angle between the two vectors is less than or equal to 90°, the fault impedance locates within or on the characteristic circle meaning the measured impedance is under reaching the zone. Then the relay will trip the corresponding circuit breaker of the transmission line. The FPGA implementation of mho elements can be straightforwardly achieved by applying complex number computation and angle comparison.

III. Case Study and Experimental Results

A. Test Setup and hardware utilization

The full hardware distance relay design was targeted to a Xilinx® Virtex®-7 XC7VX485T FPGA, using Xilinx ISE®14.1 tools to synthesize and implement the architecture. The hardware resource consumption of the entire relay design is 8% slice registers, 2% DSP, and 7% slice LUTs. A 16-bit 4 channel DAC board is connected to the FPGA board with an FMC-DAC-Adapter. The output of the DACs are captured by a 500 MHz Tektronix DPO7054 4-channel oscilloscope.

Based on the previous hardware design details of each module on FPGA, the DFT module consumes the largest latency of 83 clock cycles, while the fault detection module has the smallest latency of 3 clock cycles. In addition, the CORDIC module, the dc offset parameter calculation and the mho relay element utilize 30, 51, and 42 clock cycles respectively, giving a total latency of 209 clock cycles. Based on the FPGA clock frequency of 100MHz, the whole design has a overall latency of 2.09 μs. Since all the relay algorithms must be completed before acquiring the next sample to achieve real-time operation, the low latency of the design allows a much higher sampling frequency which is not easy to achieve in pure sequential processing.

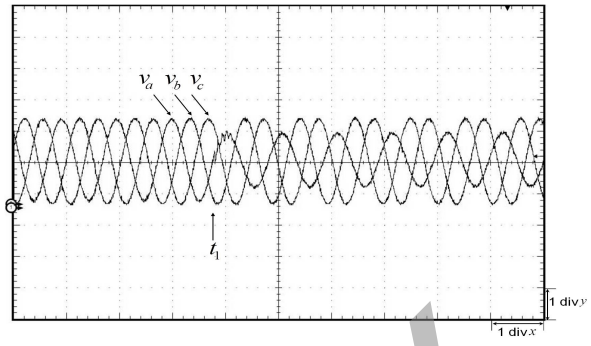

Fig. 7. Real-time voltage waveform for a Phase A to ground fault measured at relaying point. Scale: 1 div. x= 50 μs, 1 div. y= 140.0 kV.

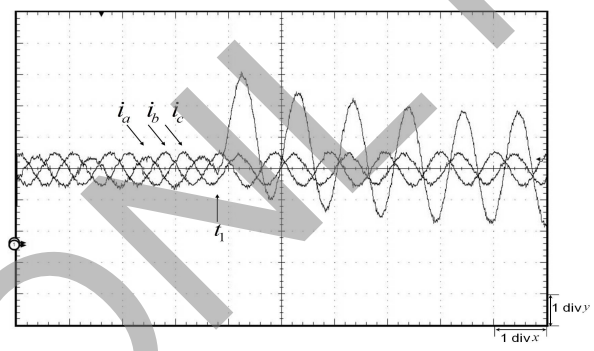

Fig. 8. Real-time current waveform for a Phase A to ground fault measured at relaying point. Scale: 1 div. x= 50 μs, 1 div. y= 0.6 kA.

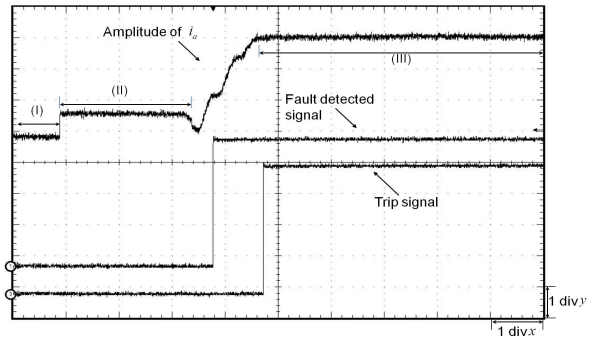

Fig. 9. Fault current amplitude, fault detected and trip signal. Scale: 1 div. x= 50 μs, 1 div. y= 0.42 kA.

To test the effectiveness of proposed distance relay hardware design, a typical single-phase-to-ground fault was simulated on a test power system using PSCAD/EMTDC to generate the fault data that the relay needs. These data are then fed into the target hardware distance relay. The tested power system consists of two synchronous generators and a 210 km long transmission line is shown in Fig. 6 with parameters given in the Appendix.

B. Results and discussion

The fault event for this case study which is a single-phase-to-ground fault occurs on the transmission line 90 km away from the relay location. The oscilloscope traces real-time simulation results of fault data transients at the

relay location which are given in Fig. 7 and Fig. 8. The fault in phase-a is initiated at time t_1, when the consequent decrease in v_a, and increase in i_a, and its dc offset can be observed. The DFT processing results, fault detection and zone 1 trip signals are captured and depicted in Fig. 9. In this figure, during Segment I the i_a amplitude stays zero until one full cycle of fault data collection is finished. Then the value stays constant under normal operating condition as shown by Segment II. After that, due to the inception of fault, according to the DFT calculation, the i_a amplitude starts to increase. The fault detection module operates to send the fault detected signal. The smooth Segment III indicates the effectiveness of the dc offset removal of fault current in phase-a. Based on the above results, the proposed distance relay has achieved its fundamental objectives.

IV. Conclusion

This paper presents the digital hardware distance relay on the FPGA. All the relay submodules are described in VHDL which can be easily transplanted to different development environments. Taking advantages of inherent parallel architecture of FPGA, the proposed hardware is paralleled and fully pipelined to achieve high operating efficiency and speed. The case study shows the effectiveness of the proposed distance relay. This demo hardware relay is a simpler implementation of distance relaying on the FPGA. Commercial distance protection systems have many more functions, and complex algorithms and features such as superior phase selection, power swing blocking, auto-reclosing, etc. Nevertheless, with plenty of hardware resources available on the FPGA board, this approach can also be updated to a multi-function hardware relay once more functional elements and features are added in the future. Furthermore, depending on the size of the system and the time-step required for a certain transient, more submodules can be replicated or added to the existing FPGA hardware distance relay to make it a relay cluster that can handle several different protection scenarios.

Appendix

I. Testing power system parameters

Source parameters: $Z_s = 9.2 + j52\ \Omega$, $E_{s1} = 230\angle 0°$ kV, $E_{s2} = 230\angle 20°$ kV.

Transmission line sequence impedance (Ω/km): $Z_0 = 0.363 + j1.326$, $Z_1 = 0.0357 + j0.5078$, $Z_2 = 0.0357 + j0.5078$.

References

[1] H. Qi, X. Wang, L. M. Tolbert, F. Li, F. Z. Peng, P. Ning, M. Amin, "A resilient real-time system design for a secure and reconfigurable power grid", *IEEE Trans. on Smart Grid,* vol. 2, no. 4, pp. 770-781, Dec. 2011.

[2] S. M. Amin, "Toward more secure, stronger and smarter electric power grids", *IEEE PES Gen. Meeting,* pp. 1-4,24-29, July 2011.

[3] D. Anderson, C. Zhao, C. Hauser, V. Venkatasubramanian, D. Bakken, A. Bose, "A virtual smart grid", *IEEE PES Mag.,* vol. 10, no. 1, pp. 49-57, Jan.-Feb. 2012.

[4] Y. Chen , V. Dinavahi, "FPGA-based real-time EMTP", *IEEE Trans. on Power Delivery,* vol.24, no.2, pp.892-902, April 2009.

[5] Y. Chen , V. Dinavahi, "An iterative real-time nonlinear electromagnetic transient solver on FPGA", *IEEE Trans. on Ind. Electr.,* vol.58, no.6, pp.2547-2555, June 2011.

[6] Y. Chen , V. Dinavahi, "Digital hardware emulation of universal machine and universal line models for real-time electromagnetic transient simulation", *IEEE Trans. on Ind. Electr.,* vol.59, no.2, pp.1300-1309, Feb. 2012.

[7] E. Monmasson, L. Idkhajine, M. N. Cirstea, I. Bahri, A. Tisan, M.W. Naouar, "FPGAs in industrial control applications", *IEEE Trans. on Industrial Informatics,* vol.7, no.2, pp.224-243, May 2011.

[8] E. Monmasson, M. N. Cirstea, "FPGA design methodology for industrial control systems - A review", *IEEE Trans. on Industrial Electronics,* vol.54, no.4, pp.1824-1842, Aug. 2007.

[9] S. H. Horowitz, A. G. Phadke, Power system relaying, 3rd ed. , *Research Studies Press Ltd. and John Wiley & Sons, Ltd. ,* May 2008.

[10] GE Multilin D60, *Line Distance Protection System Brochure,* pp. 1-8, 2012.

[11] M. R. D. Zadeh, T. S. Sidhu, and A. Klimek, "Field-programmable analog array based distance relay", *IEEE Trans. on Power Delivery,* vol. 24, no. 3, pp. 1063-1071, July 2009.

[12] X. Liu, A. H. Osman, and O. P. Malik, "Real-time implementation of a hybrid protection scheme for bipolar HVDC line using FPGA", *IEEE Trans. on Power Delivery,* vol. 26, no. 1, pp. 101-108, Jan. 2011.

[13] S. P. Valsan, K. S. Swarup, "Protective relaying for power transformers using field programmable gate array", *IET Trans. on Electric Power Applications,* vol. 2, no. 2, pp. 135-143, March 2008.

[14] H. J. Altuve Ferrer, I. Diaz Verduzco, and E. Vazquez Martinez, "Fourier and Walsh digital filtering algorithms for distance protection", *IEEE Trans. on Power Systems,* vol. 11, no. 1, pp. 457-462, Feb. 1996.

[15] A. H. Osman, O. P Malik, "Transmission line distance protection based on wavelet transform", *IEEE Trans. on Power Delivery,* vol. 19, no. 2, pp. 515-523, Apr. 2004.

[16] T. S. Sidhu, X. Zhang, F. Albasri, and M. S. Sachdev, "Discrete-fourier-transform-based technique for removal of decaying DC offset from phasor estimates", *IEE Proceedings- Generation, Transmission and Distribution,* vol. 150, no. 6, pp. 745-752, Nov. 2003.

[17] A. A. Girgis, "A new kalman filtering based digital distance relay", *IEEE Trans. on Power Apparatus and Systems,* vol. PAS-101, no. 9, pp. 3471-3480, Sept. 1982.

[18] P. K. Dash, A. K. Pradhan, and G. Panda, "Application of minimal radial basis function neural network to distance protection", *IEEE Trans. on Power Delivery,* vol. 16, no. 1, pp. 68-74, Jan. 2001.

[19] L. L. Lai, W. L. Chan, C. T. Tse, and A. T. P. So, "Real-time frequency and harmonic evaluation using artificial neural networks", *IEEE Trans. on Power Delivery,* vol. 14, no. 1, pp. 52-59, Jan. 1999.

[20] N. T. Stringer, "The effect of DC offset on current-operated relays", *IEEE Trans. on Industry Applications*, vol. 34, no. 1, pp. 30-34, Jan./Feb. 1998.

[21] J. C. Gu, S. L. Yu, "Removal of DC offset in current and voltage signals using a novel Fourier filter algorithm", *IEEE Trans. on Power Delivery*, vol. 15, no. 1, pp. 73-79, Jan. 2000.

[22] J. E. Volder, "The CORDIC trigonometric computing technique", *IRE Trans. on Electronic Computers,* vol. EC-8, no. 3, pp. 330-334, Sept. 1959.

[23] Y. H. Hu, "CORDIC-based VLSI architectures for digital signal processing", *IEEE Signal Processing Magazine,* vol. 9, no. 3, pp. 16-35, July 1992.

[24] T. S. Sidhu, D. S. Ghotra, and M.S. Sachdev, "An adaptive distance relay and its performance comparison with a fixed data window distance relay", *IEEE Trans. on Power Delivery,* vol. 17, no. 3, pp. 691-697, July 2002.

[25] T. S. Sidhu, X. Zhang, and V. Balamourougan, "A new half-cycle phasor estimation algorithm", *IEEE Trans. on Power Delivery,* vol. 20, no. 2, pp. 1299-1305, Apr. 2005.

[26] T. S. M. Rao, Power system protection : Static relays with microprocessor applications, 2nd ed. *New Delhi, India: Tata McGraw-Hill, 1989.*