Development of a Fault Locating System Using Object-Oriented Programming

Mladen Kezunovic, Fellow, IEEE   Yuan Liao *, Student Member, IEEE

Texas A&M University
Department of Electrical Engineering
College Station, TX 77843-3128

Abstract: Object-oriented programming has received wide acceptance in power system applications. The main advantages achieved by using the object-oriented techniques include easy maintenance, enhanced expandability and inherent implementation flexibility. This paper presents recent software developments of a fault location system implemented using object-oriented approaches and C++ programming language. The proposed new algorithms, implementation techniques, and evaluation studies are addressed. The developed system aims at pinpointing faults on transmission systems by utilizing recorded data coming from digital fault recorders (DFRs) sparsely located at various substations. The concept of "waveform matching" is implemented by taking advantage of the power system model. The mathematical formulation of the problem is presented. A unique genetic algorithm based search engine for pinpointing the most probable fault location is proposed. The suitability of applying the object-oriented programming for fault location software developments is demonstrated.

Keywords: Object-Oriented Techniques, C++ Programming Language, Fault Location, Power System Protection, Genetic Algorithm, Digital Fault Recorder

I. INTRODUCTION

Prompt and accurate location of the faults in a large-scale transmission system is critical to the system reliability and usually is the first step in the system restoration [1-6]. Although tremendous research efforts have been made on this topic in the past, the solutions with improved performance are [1]still sought.

Various one-end, two-end, or three-end algorithms utilizing the voltage and current quantities for estimating the fault location have been proposed previously [2-4]. The one-end algorithm is the simplest and does not require the communication of the data between the monitoring devices located at different buses. Its accuracy may be adversely affected by the fault resistance [3]. Two-end and three-end algorithms are essentially independent of the fault resistance and can yield quite accurate estimates [4]. While synchronized sampling may obtain better results than unsynchronized sampling, it entails additional system requirements like Phasor Measurement Units for interfacing the timing signals from the GPS [5].

In general, there have not been satisfactory algorithms for the case in which only recorded data at limited substation locations are available. Our work aims at proposing a better solution for such a general case. Specifically, we are investigating the feasibility of the “waveform matching” based concept for fault location by utilizing limited data. By taking advantage of the model of the power system, and carrying out simulation studies, it is hoped that better results may be obtained. We have developed a genetic algorithm (GA) based approach for implementing this concept [6]. The fault location estimation is mathematically formulated as an optimization problem of which the fault location and fault resistance are unknown variables. By iteratively posing faults in the system, running simulations, and comparing the simulated waveforms with the recorded ones, an optimal estimate of the fault location is determined. It is defined as the one that allows simulating the waveforms that best match the recorded ones [6].

As an example for implementing the proposed approach, a fault location system is developed for the Reliant Energy HL&P transmission system. The recorded data captured by digital fault recorders (DFR) and system related data (i.e., the system topology, load and generation data) are utilized. Existing software package Power System Simulator for Engineering (PSS/E) is used to carry out the needed simulation studies [2, 6-9].

Object-oriented (OO) techniques using C++ language are utilized to facilitate the development of the fault locating system [10-11].

In the rest of the paper, the problem considered by the paper and the proposed new approaches are illustrated. The overall structure of the proposed new fault locating system using Reliant Energy HL&P transmission system as an example is described. Each part of the solution is explained. Various implementation issues are illustrated.

II. PROBLEM STATEMENT AND PROPOSED NEW APPROACH

A. Problem Statement

The proposed approach will make use of the “waveform matching” based methodology. The two key concepts, namely the “sparse data” and “waveform matching” are illustrated as follows.

Yuan Liao is presently employed by ABB ETI, Raleigh, NC

Sparse data, in our work, are referring to the data obtained from recording devices sparsely located at various substation locations. Examples of recording devices may include digital fault recorders (DFR), digital relays, or other intelligent electronic devices (IED) [2]. The data captured by recording devices may include analog quantities such as voltage and current waveforms and digital quantities such as breaker status and relay operation status. Both analog and digital quantities may be useful for locating the fault [1-2]. To solve the fault location problem utilizing sparse data, the "waveform matching" based approach may be used as illustrated next.

The model of the power system is utilized to carry out simulation studies. The matching is made between the voltage and current waveforms obtained by recording devices and those generated in simulation studies. The fault is searched through the system by utilizing an iterative searching process. The searching process may consist of the following steps. First an initial fault location is assumed. Second, the simulation studies are set up according to the specified fault. Third, the simulation studies corresponding to the specified fault are carried out utilizing appropriate simulation tools. Fourth, simulated waveforms of quantities of interest are obtained. Fifth, the simulated waveforms are compared with recorded ones, and the matching degree of the simulated and recorded waveforms is evaluated by using appropriate criteria. Sixth, the initial fault location is modified according to certain approaches, and then the process proceeds to the second step and continues. The above steps are iterated until the simulated waveforms that best match the recorded ones are produced. The fault location will be determined as the one specified in the simulation studies when generating simulated waveforms that best match the recorded ones [6].

To evaluate the matching degree of the simulated and recorded waveforms, two different criteria may be employed. The first one utilizes phasors for matching. The other one utilizes transients for matching. Only the phasor matching is investigated in this paper. The short circuit studies may be carried out to obtain the phasors under the specified fault conditions [6-7]. Then the phasors are compared with those derived from the recorded waveforms.

To run simulation studies, short-circuit model of the system is needed. Among other existing software packages, Power System Simulator for Engineering (PSS/E) may be used for carrying out short circuit studies [7].

So far, the "waveform matching" based approach for fault location has already been employed by some engineers. The matching is made at the phasor level. It is known that presently the searching steps, as described previously, are carried out manually by some engineers. Manual searching may be time consuming, tedious and prone to mistakes. Our research aims at proposing a new implementation method for automated "phasor matching" that may facilitate the entire searching process and lead to improved accuracy.

To effectively guide the searching process, a genetic algorithm (GA) based searching approach based on "phasor matching" is proposed, as illustrated next [12-14].

B. Proposed New Optimization Based Approach

In the fault location problem discussed so far, there may be two possible unknown parameters, namely the fault location and the fault resistance. In our work, the type of the fault is assumed to be obtained by another fault type classification program.

To search for the fault location by phasor matching, the two unknown parameters, i.e. fault location and resistance, need to be varied. By specifying faults with various fault resistances and locations, a number of short-circuit studies can be carried out and the corresponding voltage and current phasors obtained. The most probable solution to the problem may be determined as the one specified in the simulation studies when producing the phasors that best match those derived from the recorded waveforms.

The fault location problem can be formulated as an optimization problem: Find the value of x and R_f that minimize

$$f_c(x, R_f) = \sum_{k=1}^{N_v}\{|V_{ks} - V_{kr}|\} + \sum_{k=1}^{N_i}\{|I_{ks} - I_{kr}|\} \qquad (1)$$

or maximize

$$f_f(x, R_f) = -f_c(x, R_f) \qquad (2)$$

where

$f_c(x,R_f)$: the defined cost function

$f_f(x,R_f)$: the defined fitness function. The larger the value of the fitness function, the better the matching of the simulated and recorded phasors, and the better the solution is.

x : the fault location

R_f : fault resistance

V_{ks} and V_{kr} : the during-fault voltage phasors obtained from short-circuit simulation studies and from recorded waveforms respectively.

I_{ks} and I_{kr} : the during-fault current phasors obtained from short-circuit simulation studies and from recorded waveforms respectively.

k : the index of the voltage or current phasors

N_v and N_i : the total number of the voltage and current phasors respectively.

It is noted that the largest possible fitness value defined by (2) is equal to zero and can be reached if the phasors obtained from simulation studies exactly match those obtained from recorded waveforms. Therefore, the best fault location estimate would be the one that maximizes (2). Appropriate optimization techniques need to be selected to solve this problem.

It has been shown in [6] that the fitness surface versus the fault location and resistance is not regular and contains saddle points and local maximum points. It is rather difficult to use the gradient-based method to find the global maximum point. Exhaustive search through every possible solution may also be too time-consuming to be practical [12-14].

The GA based optimization approach is good at finding the globally optimal solution and avoiding the local optima. The nature of the fitness function prompts the attempt of the GA based optimization method as described next [12].

In GA based optimization approach, (2) will be chosen as the fitness function. Fault location x and the fault resistance R_f are the two variables. The flow chart of the algorithm is shown in Fig. 3. The ranges for varying x and R_f can be decided as follows. R_f can be selected according to the typical possible fault resistance values. x can be simply selected as ranging from zero to the sum of the length of all the possible faulty lines, or estimated by other suitable algorithms [2, 4].

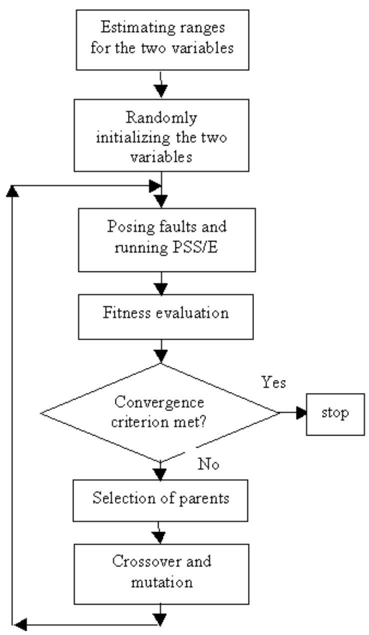

Fig. 1. The flowchart for the GA based fault location estimation

In the step "Posing faults and running PSS/E", the short circuit studies are carried out by using the software package PSS/E according to the specified fault conditions [7]. In "Fitness evaluation", (2) is applied to evaluate the matching degree of the simulated and recorded phasors. Steps related to initialization, selection of parents, crossover and mutation are standard GA steps, as illustrated in [13-14].

By iteratively posing faults, running short circuit simulations, evaluating the fitness value, and updating the fault location and resistance, the GA based searching engine guides the searching process for a globally optimal solution.

Evaluation studies have been carried out to verify the accuracy of the proposed approach. More detailed results are referred to [7].

IV. IMPLEMENTATION OF THE PROPOSED APPROACH USING OBJECT-ORIENTED TECHNIQUES

To implement the proposed algorithms using computer languages, various programming paradigms are available. Two basic alternatives, namely structural programming and object-oriented programming (OOP) may be used [15]. The structural programming method was originated in 1960s and formed the cornerstone of the modern software engineering. In structural programming, a complex system is divided into subtasks and sub-modules that are easier to control and manage. Each module has a clear interface for convenient use. However, structural programming is a data and procedure oriented design method, where the data and procedure are independent entities. Programmers have to bear in mind the data format when dealing with a procedure, which hinders code reuse.

To solve the above problem, OOP assimilates all the benefits of the structural programming, and provides encapsulation. In OOP, the data and the operation on the data (or called methods) are considered as an inseparable part using data abstraction and hiding. The object and the method are abstracted into a new data type called class. The relationship between different classes and class reuse are fully considered. OOP is promising for generating reusable and maintainable codes, reducing the software complexity and improving development efficiency. The software systems built on an object structure are more robust and expandable. OOP paradigm is embodied by OOP languages. Examples of OOP languages include LISP family, Simula, Smalltalk, and C++ family. The C++ class provides desirable encapsulation, abstraction, inheritance and polymorphism [10-11]. This makes C++ language an excellent candidate for implementing object-oriented designs.

In our work, a software package for implementing the proposed approach has been developed using object-oriented techniques written in C++ language. The developed system utilizes data coming from DFRs installed in the system. PSS/E developed by PTI has been utilized to carry out load flow and short circuit studies. The system model is given in PSS/E format. The existing commercial software DFR Assistant developed by TLI is employed to obtain the fault type and initial faulty branches [2].

This section illustrates the overall architecture of the developed software, presents the major class designs, highlights application of IPLAN language for interacting the PSS/E activities and the C++ programs, and illustrates the benefits of OO programming to fault location applications.

A. Overall Architecture

Fig.2 shows the overall architecture of the developed software. DFR Assistant converts the DFR raw data into COMTRADE format and obtains the fault type and faulty branches based on individual DFR recording. Based on such information, the phasors of the monitored voltage and current quantities, and pre-fault breaker status can be obtained. Based on the topology data and a number of measurements available, the fault location software determines the type of the fault location algorithm to be utilized. If special cases like two-end, three-end, one-end algorithms apply, the fault location can be obtained directly based on the related voltage and current phasors without running any PSS/E simulation studies [3-4]. If the case belongs to the general case, then the following steps are to be taken.

(1) The current phasors and/or the pre-fault breaker status can be utilized to tune the system topology (update the static system model). Only the service status of the branch is updated. If the magnitude of the pre-fault current phasor of the monitored branch is smaller than a pre-specified value, then the branch is considered as out of service. Branch status is also updated according to the pre-fault breaker status. If both the pre-fault breaker status and the current of a branch are monitored, the recorded current overrules the breaker status for determining the service status of the branch [7, 9].

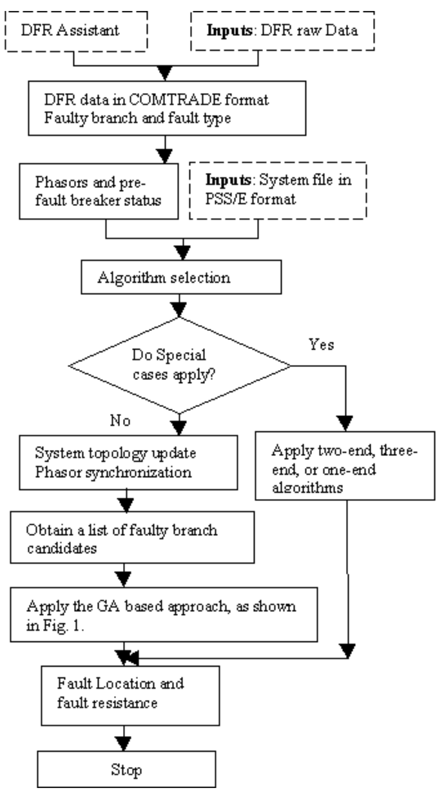

Fig. 2. Overall architecture of the software modules

(2) After the system topology is updated, the PSS/E load flow study is carried out to obtain the pre-fault phasors of the related bus voltages and branch currents. Then the phasors derived from the recorded waveforms are synchronized by rotation with reference to the phasors obtained by the load flow study.

(3) Based on the faulty branches given by DFR Assistant and the system topology, the software derives the list of total faulty branch candidates for posing faults [2].

(4) The GA algorithm based approach is activated for searching the faults.

B. Class Hierarchical Architecture

This section illustrates the hierarchical architecture of the main classes designed for the software package. Fig. 3 shows the inheritance relationship between the classes.

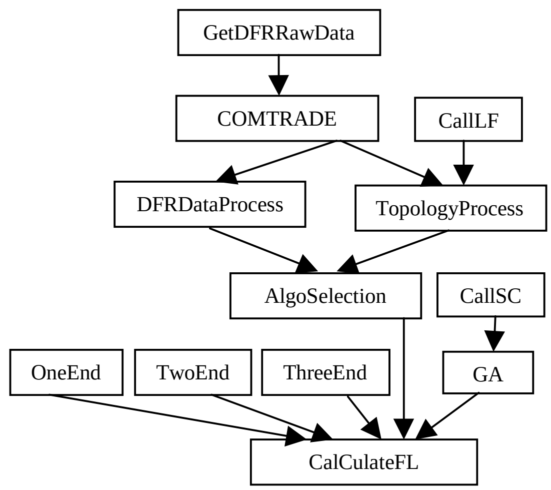

Fig. 3. The class hierarchical architecture

The class GetDFRRawData interfaces with the existing software package DFR Assistant, developed by TLI, and prepares the set-up for importing DFR data. The COMTRADE class imports the DFR data from COMTRADE format, allocates appropriate memories and saves the data in the memories. The class DFRdataProcess inherits from class COMTRADE. It processes the imported DFR data including filtering the data using low-pass filter, obtaining phasors using signal processing techniques, etc. TopologyProcess class extracts the topology information from the given power system model saved in the PSS/E format and upgrades the topology data. AlgoSelection inherits from both DFRDataProcess and TopologyProcess. It determines the type of the fault location algorithm to be used according to the topology information and the actual number of DFR recordings available. The class CallLF implements the functionality of load flow studies. The class CallSC implements the functionality of short circuit studies. Both CallLF and CallSC interfaces with the existing software package PSS/E, developed by TLI. The class OneEnd, TwoEnd and ThreeEnd realizes one-end, two-end and three-end fault location algorithms respectively. The GA class implements the genetic algorithm based fault location approach. The class CalculateFL inherits from OneEnd, TwoEnd, ThreeEnd, GA and AlgoSelection. It selects the suitable fault location algorithm and obtains the fault location results.

As an example, part of the interface description of the GA class is presented as follows.

```
class GA : public CallSC
{
        int out_iteration;//outer_loop iteration
                           //number
        int in_iteration;//inter_loop iteration
                          //number
        int n_var; //number of variables
        int binaryle[10];//length of the binary
                          //coding
        int noelitism;//number of elitism strings
        int n_parent;//number of population
        double Pc;//cross-over probability
        double Pm;//mutation probability
        ...
public:
        void fitness(double y[]);
        void initi();
        void GArepe();
        void GAresult();
        ...
        GaPro(char *pathfile);
        ~GaPro();
};
```

where, out_iteration, in_iteration, n_var, etc. are appropriate genetic algorithm parameters. In our work, standard genetic algorithms with elitism are selected [14]. Function fitness() evaluates the fitness of the specified population. Initi() initializes the genetic algorithm. Garepe() carries out the iterative steps for searching the most probable fault location. Garesult() analyzes the most probable fault location and prepares a fault report. An example fault report is presented as follows.

```
*****Fault Location Report*****

The Genetic-Algorithm based Approach is selected by
the program

The estimated fault type is: bg
The most possible fault location is obtained as:
On the line from 4278(GRNBYUE8) to 4746(WITTER 8)
(circuit# is 21), and is 0.835971 p.u. (45.602150
miles) from the bus 4278(GRNBYUE8).
The fault resistance is 0.000000 p.u.

The voltages at the fault point are:
Phase A Magnitude(p.u.): 1.257122, Angle(degrees):
24.828695
Phase B Magnitude(p.u.): 0.000000, Angle(degrees):
14.036243
Phase C Magnitude(p.u.): 1.225940, Angle(degrees):
65.951640

The currents through the fault path are:
Phase A Magnitude(p.u.): 0.000032, Angle(degrees):
67.924648
Phase B Magnitude(p.u.): 30.387240, Angle(degrees):
121.720273
Phase C Magnitude(p.u.): 0.000041, Angle(degrees):
165.878602
```

C. Interaction between C++ and PSS/E Software Package

It is seen from Fig. 1 that the interaction between the C++ program and the PSS/E application software is required for fitness evaluation. In our software development, all the running of the PSS/E load flow and short circuit studies is invoked within the C++ program. The following C++ functions illustrate the adopted approaches.

```
void callexe(char *prog, char *args)
{
      _spawnl(_P_WAIT, prog, prog, args,
      NULL);
}
Void invoke_PSSE()
{
callexe("x:\\pti\\psse26\\pssexe\\psslf4", "-gnikool
off -buses 4000 -inpdev
g:\\users2\\liaoy\\ly\\psse\\flocini.idv");
}
```

where psslf4 is the name of the PSS/E application package. Flocini.idv is the PSS/E response file that contains the appropriate commands to be implemented.

Various activities performed within PSS/E environment are accomplished by using IPLAN commands. IPLAN is a stand-alone programming language associated with PSS/E main application package. It aims at facilitating carrying out load flow or short circuit studies automatically in a batch mode. All the load flow and short circuit studies utilized in our work are realized through IPLAN programs. Some common IPLAN commands are illustrated as follows.

Commands used for retrieving load flow study results include BUSDAT, BRNCUR and BRNFLO. Commands used for retrieving short circuit study results include SCINIT, SCDONE, SCBRN2 and SCBUS2. Activity SCMU performs unbalanced short circuit studies. It allows the posing of faults with different fault location, fault type, and fault resistances. This is the key command utilized for performing short circuit studies. Command FNSL solves the load flow case using the Newton-Ralphson method. Command CHNG modifies the system topology specified by the user. More detailed illustration on the IPLAN language and commands is referred to [7].

D. Benefits of Utilizing OO Programming for Fault Location Software Development

This section illustrates the advantages achieved by utilizing OO techniques for developing fault location software.

One of the main benefits resulting from OO techniques is to promote code reuse. As discussed previously, the developed software utilizes the existing software package DFR Assistant for acquiring the DFR raw data and converting the data to COMTRADE format. The PSS/E software package is utilized for performing load flow and short circuit studies. Classes GetDFRRawData, CallLF and CallSC smoothly interface these existing software packages and the developed software.

Directly using existing software packages reduces the development time and costs. This allows us to devote more time and efforts to focus on the core algorithm development, i.e., the genetic algorithm in our work. After the software is developed and tested, additional appropriate modules may be developed to improve the performance of the software.

By utilizing OO techniques, the interfaces between different modules can be carefully designed such that the modules (called client) that invoke the PSS/E and DFR Assistant packages (called server) can still work properly even if in the future these servers are changed or replaced. For example, if a more appropriate short circuit package is developed in the future, it could replace PSS/E package for the developed fault location software. The fault location software would need no or little modification in such case. This not only facilitates code reuse, but also reduces the maintenance costs.

Various types of fault location algorithms have been published in the past, such as the one-end, two-end, three-end and genetic algorithms [1-6]. It is desirable to utilize the most of these approaches when developing a fault location system. An efficient and systematic way for doing this is needed. OO based technique seems as a promising tool for the solution. To do this, these algorithms are encapsulated in individual classes, implemented, and tested respectively. Once these algorithms are successfully verified by evaluation studies, they can be readily incorporated into any fault location software package.

Component Object Model (COM) techniques, originated by Microsoft company, provide a new promising way for promoting code reuse [16]. Diverse fault location algorithms may be further wrapped into appropriate COM components. With appropriate designed interfaces, they can greatly facilitate code reuse and system expandability. Such new techniques may be attempted in the future work.

V. CONCLUSIONS

This paper introduces a novel genetic algorithm based approach for implementing the "waveform matching" concept for transmission line fault location. An actual system has been developed by utilizing the HL&P transmission system as an example. The object-oriented techniques have been applied for the software developments by using C++ language. It is shown that C++ language and object-oriented programming facilitate expandability, maintenance and flexibility of the developed software. Application of IPLAN language for automating the PSS/E load flow and short circuit activities has also been presented.

VI. ACKNOWLEDGEMENT

Software developments reported in this paper are funded by the Texas Higher Education Coordinating Board Advanced Technology Program. The co-funding is provided by TXU Electric and Reliant Energy HL&P.

VII. REFERENCES

[1] A. A. Girgis and M. B. Johns, "A hybrid expert system for faulted section identification, fault type classification and selection of fault location algorithms", *IEEE Transactions on Power Delivery*, vol. 4, no. 2, pp. 978 – 985, April 1989.
[2] M. Kezunovic, "Automation of fault analysis using DFR data", in *Proceedings of the 60th American Power Conference*, Chicago, IL, April 1998, pp. 91-98.
[3] M. S. Eriksson, and G. D. Rockfeller, "An accurate fault locator with compensation for apparent reactance in the fault resistance resulting from remote-end infeed", *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-104, no. 2, pp. 424-436, February 1985.
[4] A. A. Girgis, D. G. Hart, and W. L. Peterson, "A new fault location technique for two- and three-terminal lines", *IEEE Transactions on Power Delivery*, vol. 7, no. 1, pp. 98 –107, January 1992.
[5] Gopalakrishnan, M. Kezunovic, S.M. McKenna, and D.M. Hamai, "Fault location using distributed parameter transmission line model," *IEEE Transactions on Power Delivery*, in press, 2000.
[6] M. Kezunovic and Y. Liao, "Fault location estimation based on matching the simulated and recorded waveforms using genetic algorithms", to be presented at the 7th International Conference on Developments in Power Systems Protection, Netherlands, 2001.
[7] Power Technologies, Inc., *PSS/E 25 Program Operation and Application Manuals*, December 1997.
[8] IEEE, *COMTRADE Standard* (C37.111), 1991.
[9] M. Sasson, S. T. Ehrmann, P. Lynch, and L. S. V. Slyck, "Automatic power system network topology determination", *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-92, pp. 610-618, March/April 1973.
[10] Z. L. Gaing, C. N. Lu, B. S. Chang, and C. L. Cheng, "An object-oriented approach for implementing power system restoration package", IEEE Transactions on Power Systems, vol. 11, no. 1, February 1996, pp. 483-489.
[11] Manzoni, A. S. e Silva, and I. C. Decker, "Power systems dynamics simulation using object-oriented programming", IEEE Transactions on Power Systems, vol. 14, no. 1, February 1999, pp. 249-255.
[12] R. E. Miller, *Optimization: Foundations and Applications*, Wiley Press, New York, NY, 2000.
[13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, MA, 1989.
[14] C. S. Chang, W. Q. Jiang, and S. Elangovan, "Application of genetic algorithms to determine worst-case switching overvoltage of MRT systems", *IEE Proceedings, Part C: Generation, Transmission, and Distribution*, vol. 146, no. 1, pp. 81-87, January 1999.
[15] Bjarne Stroustrup, *The C++ Programming Language*, Addison-Wesley, 2000.
[16] John E. Swanke, *COM Programming by Example*, R&D books, Lawrence, Kansas, 2000.

VIII. BIOGRAPHIES

Mladen Kezunovic (S'77, M'80, SM'85, F'99) received his Dipl. Ing. Degree from the University of Sarajevo, the M.S. and Ph.D. degrees from the University of Kansas, all in electrical engineering, in 1974, 1977 and 1980, respectively. Dr. Kezunovic's industrial experience is with Westinghouse Electric Corporation in the USA, and the Energoinvest Company in Sarajevo. He also worked at the University of Sarajevo. He was a Visiting Associate Professor at Washington State University in 1986-1987. He has been with Texas A&M University since 1987 where he is the Eugene E. Webb Professor and Director of Electric Power and Power Electronics Institute. His main research interests are digital simulators and simulation methods for equipment evaluation and testing as well as application of intelligent methods to control, protection and power quality monitoring. Dr. Kezunovic is a registered professional engineer in Texas, and a Fellow of IEEE.

Yuan Liao (S'99) received his B.S. and M.S. degrees from Xi'an Jiaotong University, P. R. China, the Ph.D. degree from Texas A&M University, all in electrical engineering, in 1993, 1996 and 2000, respectively. Presently, he works for ABB ETI, Raleigh. His research interests are applications of wavelets, signal processing and artificial intelligence to power system protection and power quality studies.