Interactive Protection System Simulation Using ATP MODELS and C++

X. Luo, *Student Member, IEEE*, and M. Kezunovic, *Fellow, IEEE*

***Abstract*—This paper presents a new approach for interactive protection system simulation. In this approach, the power system network is modeled by the ATP program while the “compiled foreign model” mechanism of MODELS language is employed to model the digital protective relay in C++ language. This allows “object-oriented” relay modeling as well as building a “seamless” interface between the power system network model and the relay model. An example is used to demonstrate the interactive protection system simulation developed using the new approach.**

***Index Terms*—power system, protection system, relay model, ATP, MODELS, C++**

I. Introduction

COMPUTER based interactive protection system simulation has been studied for years. The study generally covers power system network modeling, protective relay modeling and dynamic interaction between the power system network models and the relay models. The simulation is quite valuable for preliminary testing of relay algorithms, study of multi-terminal, coordinated relaying schemes, and evaluation of relay performance during cascade events [1], [2].

Previous research explored various options related to the software programs for modeling of power system networks and protective relays, and the schemes for interfacing the power system network models and the protective relay models. They generally fall into three categories.

The use of electromagnetic transients program (EMTP) for power system network modeling, and the transient analysis of control system (TACS) functions of EMTP for protective relay modeling is reported in the early literature [3]. Complied FORTRAN subroutine called from TACS in the EPRI/DCG version of EMTP is also used to develop protective relay models as reported in [4]. The MODELS language of the alternative transient program (ATP) version of EMTP, which is an enhancement to TACS, is employed for protective relay modeling as reported in [5], [6]. A prominent advantage of these approaches is the easy interfacing between the power system network models and the protective relay models because the TACS and MODELS are inherently embedded in EMTP/ATP [7-9].

A scheme which uses an “interaction buffer” for interfacing power system networks modeled by EMTP and protective relays modeled by MATLAB is described in [1]. Another method for establishing the link between EMTP and MATLAB is discussed in [10]. It is an interconnection where the internal computation engine of MATLAB is directly accessed by the FORTRAN code in EMTP. By these approaches, the high-level computation facilities of MATLAB can be utilized for protective relay modeling while the interconnection between the relay models and the power system network models is maintained.

An approach where power system network models are created in MATLAB/Power System Blockset and protective relay models are developed in MATLAB/SIMULINK is presented in [11].The interfacing is easily achieved since both the power system network models and the protective relay models are under the context of MATLAB/SIMULINK [12-14].

Despite the obvious advantages, the approaches discussed above have their inherent limitations. With respect to the first category of approaches, sophisticated relay models are difficult to be developed by TACS, MODELS and FORTRAN due to their limited flexibility and programmability. The “interaction buffer” and the programmed link discussed in the second category will cost excessive simulation time. They also cause the entire simulation program lack of integrity and portability. The problem of the third category lies in the slow simulation speed when the power system networks modeled by MATLAB/Power System Blockset are of large scale.

This paper outlines a new approach for interactive protection system simulation. In this approach, the power system network is modeled by the ATP program while the “compiled foreign model” mechanism of MODELS language is employed to model the digital protective relay in C++ language, which allows relay modeling in an “object-oriented” way as well as building a “seamless” interface between the power system network model and the relay model. The ATP/MinGW software package is used to facilitate the entire compilation and link process. As a result, the enhanced relay model representation, and the “seamless” interaction between the power system network model and the relay model make the overall protection system simulation more powerful.

In Section II, the “complied foreign model” mechanism of MODELS language is introduced. Section III gives the details of the modeling of a digital relay and its interfacing to the

The work reported in this paper is funded by NSF I/UCRC called Power Systems Engineering Research Center (PSERC) under the Project T-17 titled “Enhanced Reliability of Power System Operation Using Advanced Algorithms and IEDs for On-Line Monitoring”.

X. Luo and M. Kezunovic are with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843-3128, USA (e-mails: xuluo@ee.tamu.edu, kezunov@ee.tamu.edu).

power system network model. Section IV uses an example to demonstrate the interactive protection system simulation developed using the new approach. Section V draws the conclusions.

II. COMPLIED FOREIGN MODEL OF MODELS LANGUAGE

MODELS language is a general-purpose description language of the ATP program [8], [9]. It provides a format which focuses on description of the structure of a model, and the function of its elements. However, compared with high-level programming languages such as C/C++, its flexibility and programmability are relatively limited. To overcome the disadvantage, MODELS provides a "compiled foreign model" mechanism to expand its flexibility and programmability. This mechanism can be utilized for modeling a protective relay in high-level languages, and interfacing the relay model with the power system model.

A. Complied Foreign Model

MODELS provides a pre-defined interface to link procedures called a "foreign model" which is written in other programming languages to the ATP simulation program [9]. The interface is defined as four arrays carrying the values of data, input, output and history variables. Each "foreign model" should provide both an execution procedure and an initiation procedure corresponding to the EXEC procedure and INIT procedure of a model defined in MODELS.

A "foreign model" must be compiled and linked to the ATP simulation program before it can be called by MODELS. An interface routine in a FORTRAN file called "formod.for" is where the user registers the correspondence between the identification name used in the "foreign model" declaration in MODELS, and the actual names of the procedures in the "foreign model". Once declared and named, a "foreign model" can be used independently in as many separate uses as required. The inputs and outputs of the "foreign model", along with the directives controlling its simulation, are specified in a regular USE statement in MODELS.

B. Compilation and Link Facility

The newly developed ATP/MinGW program package has convenient tools to compile a "foreign model" written in FORTRAN and C/C++, and link it with the ATP simulation program [15]. The Minimalist GNU for Windows (MinGW) is a complier package for windows operating systems [16]. In the ATP/MinGW program package, the source code of the ATP program is complied by the FORTRAN compiler and C compiler to generate object files. The compilers are also used to compile the user-supplied source code of a "foreign model" written in FORTRAN or C/C++ to generate its object file. Then all the object files and libraries are linked together to produce a new executable ATP program, which takes the ATP data case file as input to run the simulation. Fig. 1 illustrates the whole make process which includes the compilation and link. It should be mentioned that the users can easily complete the make process in dialogs in the ATP/MinGW program package.

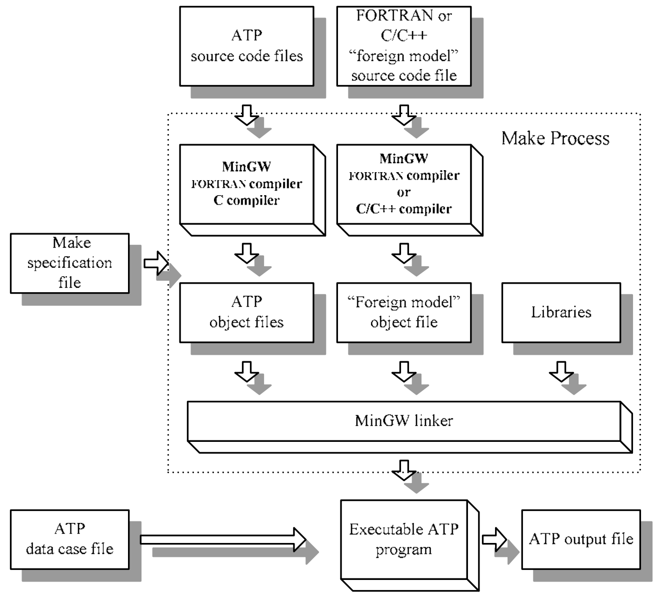

Fig. 1 Make process in the ATP/MinGW program package

III. MULTIFUNCTIONAL DIGITAL RELAY MODEL

The functional requirements for a digital protective relay model are dependent on the major purpose of the simulation study. Currently, an interactive protection system simulation is established for the demonstration of a novel relay data analysis application which aims to analyze large quantity of digital relay data. A multifunctional digital relay model is developed to serve as the data source for the application [17].

A. Main Features of the Relay Model

Besides the features which satisfy the common functional requirements to components, interface, and protection functions, the relay model is also capable of inserting user-defined errors and generating relay files and reports. Table I lists the main features of the relay model.

TABLE I: MAIN FEATURES OF THE RELAY MODEL

Requirements	Features
Components	• analog filter • A/D converter • implementation of signal processing and protection algorithms
Interface	• up to 4 channels of node voltages inputs and 8 channels of branch currents inputs • up to 2 channels of breaker status contact inputs • up to 2 channels of pilot signal inputs • up to 6 channels of trip signal outputs • up to 2 channels of pilot signal outputs
Protection Functions	• phase distance • ground distance • phase instantaneous over-current • ground instantaneous over-current • autoreclosing
Others	• user-defined error insertion • setting file reading • generation of oscillography files, fault reports and event reports

B. *Programming Structure of the Relay Model*

By virtue of the “compiled foreign model” mechanism, the advanced features of C++ language such as object-oriented concepts, direct access to windows libraries, and powerful file I/O capability can be utilized to model the relay. It is possible to realize all the functions of the relay model in the C++ “foreign model”. However, since the MODELS language itself has some unique features which facilitate modeling of some components of the relay, we adopt a hybrid approach to realize the relay functions in both the MODELS section of ATP data case file and the C++ “foreign model”. The interfacing to power system network model, analog signal filtering, and the A/D conversion is implemented in the MODELS section, while all other functions of the relay model are realized in the C++ “foreign model”. In order to reuse the “foreign model” with different configuration of inputs, outputs and sampling rates to represent different relay locations while applying the common analog signal filtering to all the reused “foreign models”, we also employ an “inheritance” modeling architecture. A model named “RELAY” which represents a generic relay model is declared in the MODELS section. A “foreign model” named “FM” which realizes the specific relay functions is further declared in the generic relay model. In the execution procedure of the generic relay model, the analog filtering function is defined, which is followed by the definitions of use of the “foreign model” with different inputs, outputs and sampling rates. Fig.2 illustrates programming structure of the relay model.

```
MODELS     ---MODELS section of the ATP data case file
INPUT  …   --- declare inputs from the power
               system network
OUTPUT …   --- declare outputs to the power system network

MODEL RELAY      ---declare the relay model
  INPUT …             --- declare the inputs for the relay model
  OUTPUT …            --- declare the outputs for the relay model
  MODEL FM FOREIGN FM {…}  --- declare the “foreign model”
  INIT                ---define the initiation procedure
  …
  ENDINIT
  EXEC                --- define the execution procedure
   ZFUN (y/x):=…          --- call Z-transform transfer function
                              for analog filtering
   USE FM AS FM           --- call the foreign model
     TIMESTEP MIN: …  ---define the sampling rate
                          for A/D conversion
     INPUT…           --- define the inputs for the “foreign
                          model”
     OUTPUT…          --- define the outputs for the “foreign
                          model”
   ENDUSE
  ENDEXEC
ENDMODEL
USE RELAY AS RELAY1
  INPUT…              --- define the inputs for the relay model
  OUTPUT…             ---define the outputs for the relay model
ENDUSE
USE RELAY AS RELAY2  --- reuse the relay model with different
                         inputs and outputs
  INPUT…
  OUTPUT…
ENDUSE
ENDMODELS
…

---C++ file for the “foreign model”
extern “C” {
      …   ---declare the initiation procedure and execution
              procedure
}
void relay_i_( double xdata_ar[],
               double xin_ar[],
               double xout_ar[],
               double  xvar_ar[]);
{ …        ---read relay setting file
}
void relay_m_( double xdata_ar[],
               double xin_ar[],
               double xout_ar[],
               double  xvar_ar[]);
{ …        ---implement protection algorithms
  …        ---generate relay files and reports
}
```

Fig. 2 Programming structure of the relay model

C. *Implementation of the Relay Model*

1） Interface to the Power System Network Model

The inputs from the power system network model are three phase voltages measured at bus nodes, three phase currents measured through circuit breaker switches, and statuses of circuit breaker switches. The outputs to the power system network model are control variables of the control nodes of circuit breaker switches. The names of these nodes and switches are declared in the INPUT and OUTPUT directives of the MODELS section. In the USE statement of the MODELS section, the inputs from the power system network model and outputs to the power system network model are referred by the generic relay model. In the USE statement of the generic relay model, the inputs and outputs associated with a specific relay location are further referred by a “foreign model”. In such a way, the interaction between the power system network model and the relay model associated with a specific location can be realized.

2) Analog Filtering

In order to meet the sampling theory, the sampling rate of the relay model should be twice the maximum frequency in the input analog signals. Sampling with a lower sampling rate will result in errors due to the aliasing effect in the discrete time signals. The anti-aliasing filters, which in practice are analog filters, should be used to minimize such aliasing effect as well as attenuate the high frequency components. In the relay model, analog second order Butterworth low-pass filter is employed. From a modeling point of view, such a filter can be represented by the Z-plane digital transfer function, which can be easily realized by the Z-transform transfer function of MODELS language.

3) A/D conversion

The sample and hold circuit of A/D converters is realized by the TIMESTEP MIN: “time step” directive in the USE statement of the generic relay model. This will actually perform the interpolation on the original simulation time-stamp at the rate of the specified time step.

4) Protection Algorithms

All the protection algorithms are implemented in the C++ “foreign model”. Fourier Transform is used to extract the fundamental frequency phasors for phase voltages and currents, line voltages and currents, and zero sequence currents. The phasors of phase currents and the phasors of zero sequence currents are used for comparison with the pickup thresholds of the Phase IOC Element and the Ground IOC Element respectively. The phasors for line voltages and currents are used to calculate the line impedances for comparison with the MHO characteristic of the Phase Distance Elements. The phasors for phase voltages and currents are used to calculate the phase impedances for comparison with the quadrilateral characteristic of the Ground Distance Elements. Timers are simulated to ensure the required time coordination between the pickup and the operation of protection elements. All the protection elements are programmed in an “object-oriented” way, which can greatly facilitate functional expansion in the future.

5) Relay File Generation

In the relay model, the analog signals of input voltages and currents, and digital signals representing current supervision, pickup and operation of protection elements are stored in the arrays for oscillography use. The status changes of digital signals are detected and used for event report generation. At the end of the simulation, the file I/O functions of C++ are employed to generate the event report and the oscillography files in COMTRADE format [18].

IV. EXAMPLE

In this section, an example is used to demonstrate the interactive protection system simulation. A substation with its transmission lines is modeled in the ATPDRAW program, by which an ATP data case file is generated [19]. A relay named DR01 is modeled by the MODELS language and C++ language. The MODELS section is inserted into the ATP data case file. The C++ “foreign model” is complied and linked to the object files of the ATP program to generate a new ATP executable program.

A. I/O Connection of the Relay Model

Fig. 3 illustrates the I/O connection of the relay model with the substation model. The relay DR01 is protecting the outgoing line L1 which is connected to the substation by one and a half breaker scheme. The relay takes node voltages on bus B1 as voltage inputs and branch currents through circuit breakers CB1 and CB2 as current inputs. The two circuit breakers are controlled by the relay by means of three-phase tripping scheme. The statuses of breakers are also monitored by the relay.

B. Relay Setting

Four protection elements including Phase Distance, Ground Distance, Phase Instantaneous Over-Current and Ground Instantaneous Over-Current are enabled in the relay model.

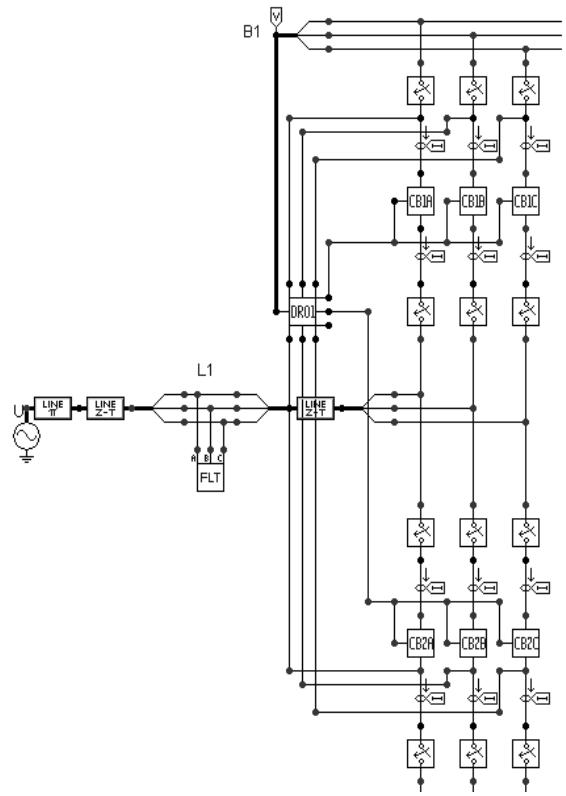

Fig. 3 I/O connection of the relay model with the substation model

The autoreclosing function is also enabled. Table II lists the major relay setting.

TABLE II: MAJOR RELAY SETTING

Elements		Range (% of the line length)	Coordination Time Delay (Second)
Phase Distance	Zone 1	75	0.008
	Zone 2	150	0.2
	Zone 3	230	1.0
Ground Distance	Zone 1	75	0.008
	Zone 2	150	0.2
	Zone 3	230	1.0
Phase IOC		N/A	0.3
Ground IOC		N/A	0.3
Autoreclosing		N/A	0.4

C. Fault Scenario

A temporary fault is assumed to occur on line L1. TABLE III lists the fault information.

TABLE III: FAULT INFORMATION

Fault Type	A-B
Fault Location (% of the line length)	80 (Zone 2)
Fault Inception Time (Second)	0.200
Fault Disappearance Time (Second)	0.600

D. Expected Protection Operation

The relay and associated circuit breakers should respond to the fault according to relay settings and performance specifications. TABLE IV lists the major characteristics of expected protection operation.

TABLE IV: EXPECTED PROTECTION OPERATION

Operated Element	Phase Distance Zone 2
Relay Trip Time (Second)	0.400
Circuit Breaker Opening Time (Second)	0.432
Current Interruption Time (Second)	< 0.448
Current Resume Time (Second)	< 0.848

E. Actual Protection Operation

A setting error of the characteristic of Phase Distance Zone 2 Element is deliberately introduced. After a simulation lasted for 1.1 second, the relay model generates an oscillography file and an event report. Fig. 4 and Fig. 5 show the oscillography file and event report displayed in the GUI of the relay data analysis application respectively [17]. It should be noticed that the oscillography shows the waveform of line currents of line L1, which is the sum of branch currents through circuit breakers CB1 and CB2. Table V lists the major characteristics of actual protection operation.

TABLE V: ACTUAL PROTECTION OPERATION

Operated Element	Phase IOC
Relay Trip Time (Second)	0.508
Circuit Breaker Opening Time (Second)	0.540
Current Interruption Time (Second)	0.550
Current Resume Time (Second)	0.940

As we can see, it was the Phase IOC Element instead of Phase Distance Zone 2 Element that triggered the relay trip since we deliberately introduce an error which made the Phase Distance Zone 2 Element failed to pick up. From such information, we may know that the Phase IOC Element functioned correctly as a backup for distance elements. Since the operating time delay of Phase IOC Element was set to be 0.1 second longer than that of Phase Distance Zone 2 Element, the relay trip, opening of circuit breakers, and reclosing of circuit breakers were delayed nearly 0.1 second. Another event which we should noticed in the event report was the pickup of Phase Distance Zone 1 Element for nearly 0.001 second. The reason for this abnormity was that the Zone 2 fault occurred near the boundary of Zone 1 and Zone 2. During the transient period, the impedance trajectory happened to fall into Zone 1 for a very short period of time. To prevent the operation of Phase Distance Zone 1 Element and Ground Distance Zone 1 triggered by such unexpected event, the coordination time delay for their operation had been set as 0.008 second, which significantly improved the security of Zone 1 operation.

V. CONCLUSIONS

Based on the discussion in this paper, conclusions are drawn as follows:

1) A relay model and its interfacing with a power system network model can be implemented in variety of options. However, most of the previously developed approaches have disadvantages which may limit their implementations.

2) The "compiled foreign model" mechanism of MODELS language provides convenient method to develop models for sophisticated dynamic systems such as digital relays based on high level languages such as C/C++ to facilitate the interface of dynamic system models and power system network models.

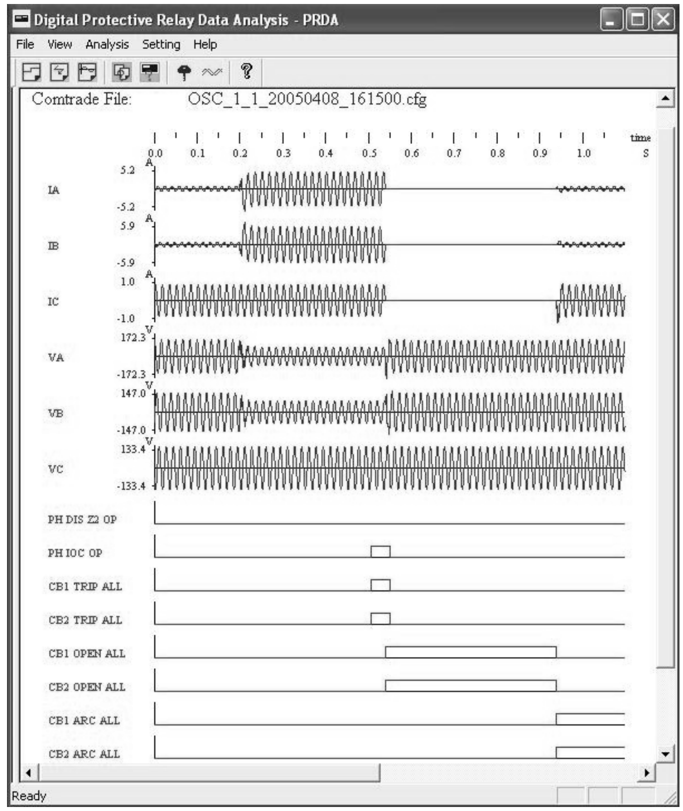

Fig. 4 Oscillography file generated by the relay model

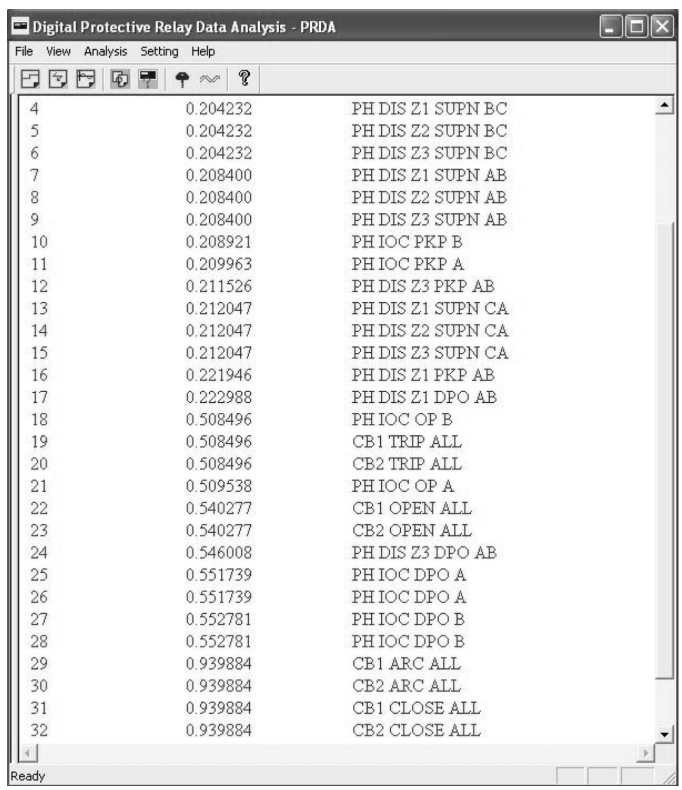

Fig. 5 Event report generated by the relay model

This makes the interactive simulation more flexible and powerful.

3) The flexibility of C++ language greatly facilitates the modeling of sophisticated digital relays. Its powerful file I/O capability is quite useful for relay file and report generation.

Some future work is proposed. MATLAB and some other intelligent system shells provide run-time access routine for C/C++ language. The digital relay model can be improved to utilize the functions in the MATLAB and the intelligent system shells such as an expert system shell. Thus a platform to study intelligent system application to analysis of protection system operation based on interactive simulation can be implemented.

VI. References

[1] M. Kezunovic, Q. Chen, "A novel approach for interactive protection system simulation", *IEEE Trans. Power Systems*, vol. 12, no. 2, pp. 668-674, April 1997

[2] M. Kezunovic, S. Vasilic, D. Ristanovic, "Interfacing protective relays and relay models to power system modeling software and data files", PowerCon 2002, Kunming, China, October 2002

[3] A. Domijan, M. V. Emami, "State space relaying modeling and simulation using the electromagnetic transient program and its transient analysis of control capability", *IEEE Trans. Energy Conversion*, vol. 5, no. 4, pp. 667-702, December 1990

[4] A. K. S. Chaudhary, Kwa-Sur Tam, A. G. Phadke, "Protection system representation in the electromagnetic transients program", *IEEE Trans. Power Delivery*, vol. 9, no. 2, pp. 700-711, April 1994.

[5] C. H. Kim, M. H. Lee, R. K. Aggarwal, A. T. Johns, "Educational use of EMTP MODELS for the study of a distance relaying algorithm for protecting transmission lines", *IEEE Trans. Power Systems*, vol. 15, no. 1, pp. 9-15, February 2000

[6] J. Y. Heo, C. H. Kim, K. H. So, N. O. Park, " Realization of Distance Relay Algorithm using EMTP MODELS", Internal Conference on Power System Transients-IPST 2003, New Orleans, LA, September 2003

[7] H. W. Dommel, *Electromagnetic Transients Program (EMTP) Theory Book*, Bonneville Power Administration, Portland, OR, 1986
Available: http://www.eeug.org/files/secret/EMTP_TheoryBook

[8] CanAm EMTP User Group, *Alternative Transient Program (ATP) Rule Book*, Portland, OR, 2001
Available: http://www.eeug.org/files/secret/ATP_RuleBook

[9] L. Dube, *MODELS in ATP, Language Manual*, 1996
Available: http://www.eeug.org/files/secret/MODELS

[10] J. Mahseredjian, G. Benmouyal, X. Lombard, " A link between EMTP and MATLAB for user-defined modeling", IEEE Trans. Power Delivery, vol. 13., no. 2, pp. 667-674, April 1998

[11] M. Kezunovic, B. Kasztenny, "New SIMULINK libraries for modeling digital protective relays and evaluating their performance under fault transients", International Conference on Power System Transients-IPST 1999, Budapest, Hungary, June 1999

[12] The MathWorks, Inc., *Using MATLAB*, Natick, MA, July 2002

[13] The MathWorks, Inc., *SimPowerSystems User's Guide*, Natick, MA, July 2002

[14] The MathWorks, Inc., *Using SIMULINK* , Natick, MA., July 2002

[15] Japanese ATP User Group, Introduction to ATPDraw+ATP/MinGW Easy Installer, http:// www.jaug.jp/mingw32/atpeasye.html

[16] MinGW User Group, Introduction to MinGW, http://www.mingw.org

[17] X. Luo, M. Kezunovic, "Automated analysis of digital relay data based on expert system", PowerTech 2005, St. Petersburg, Russia, June 2005

[18] *IEEE Common Format for Transient Data Exchange (COMTRADE) for Power Systems*, IEEE Standard, 1999

[19] SINTEF Energy Research, *ATPDRAW Version 3.5 Users' Manual*, Trondheim, Norway, 2002
Available: http://www.eeug.org/files/secret/ATPDraw

VII. Biographies

Xu Luo (S'05) received his B.E. and M.E. degrees from Xi'an Jiaotong University, Xi'an, China, both in electrical engineering in 1999 and 2002 respectively. He has been with Texas A&M University pursuing his Ph.D. degree since Aug. 2002. His research interests are power system protection, substation automation, and artificial intelligence applications in power system monitoring, control and protection.

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. 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 the IEEE.