New SIMULINK Libraries for Modeling Digital Protective Relays and Evaluating Their Performance Under Fault Transients

M. Kezunovic

Department of Electrical Engineering
Texas A&M University
College Station, TX 77843-3128, USA
kezunov@ee.tamu.edu

B. Kasztenny

Department of Electrical Engineering
Texas A&M University
College Station, TX 77843-3128, USA
bogdan.kasztenny@ieee.org

Abstract — **This paper presents five SIMULINK libraries for modeling, design, optimization and testing of digital protective relays. The new MATLAB based software package includes the following libraries: *Relay Elements*, *Relays*, *Protection Systems*, *Input Signals* and *Tools*. The paper presents the software package in detail. First, all the components are described including their functions and variety of available options. Next, sample studies are presented illustrating the usage of the package to simulate operation of digital relays in both open-loop and closed-loop modes.**

Keywords: Modeling and Simulation, Fault Transients, Protective Relaying, MATLAB

I. Introduction

The assessment requirements for protective relaying devices ask for an ability to verify both the relay design and application performance [1].

The relay design assessment is needed when a new relay design is introduced. In industry, a vendor or utility may want to evaluate the relay design characteristics to make sure that its performance is as expected [2]. In education, students learn by emulating an algorithm of a given relay (modeling) and testing it (simulation).

Typically, this type of evaluation is performed using phasors representing pre-fault, fault and post-fault waveforms. Once these waveforms are subjected to a relay, it is possible to get measurements of the corresponding relay operating characteristics. The measured characteristics can than be compared to the theoretical ones.

The application performance assessment is related to the relay behavior under actual fault conditions in a given power system [3]. This assessment is concerned with relay's ability to respond correctly under fault conditions, as well as its capability to achieve correct operation within a required time interval. In order to perform this evaluation, the relay has to be subjected to the transient waveform that closely resemble actual fault events.

The existing practice for relay testing is related to the use of phasors to perform design testing verifying the operating characteristic and calibrating the relay settings. The future relay testing practice may include extensive application testing using transients verifying relay performance under actual fault scenarios and conditions.

Significant design, evaluation and testing activities can be accomplished in the economical way using modeling and simulation of physical relays and power systems.

Designing, setting, testing and evaluating protective relays call for software tools capable of modeling the protective relays of various designs as well as the surrounding power system. Recently, significant effort has been put on developing both transient-based and phasor-based models of protective relays [4].

This paper presents new SIMULINK [5] libraries developed for modeling, testing and evaluating protective relays. These libraries include: **Relay Elements** (models of generic components of digital relays enabling one to create a detailed model of a given microprocessor based relay); **Relays** (models of protective relays such as overcurrent, impedance and differential); **Protection Systems** (models of complete protection terminals and/or schemes); **Input Signals** (specialized signal generators and file converters); **Tools** (specialized displays).

The paper starts with discussion of MATLAB as a simulation environment (sec.II); presents the basic concept of the new software (sec.III); introduces the new SIMULINK libraries (sections IV through VII) and closes with simulation examples (sec.VIII).

II. MATLAB — The Simulation Environment

MATLAB has been selected as a shell for the new protective relay modeling software [5].

MATLAB, a high-performance language for technical computation, integrates calculation, visualization and programming in an easy-to-use environment. At universities, MATLAB has become a standard instructional tool for introductory and advanced courses in mathematics, engineering and science. In industry, MATLAB is also a tool of choice for high-productivity research, development and analysis.

Particular factors that support the selection of MATLAB are:

(a) MATLAB is a de-facto standard that already dominates the university environment and is more and more recognized in industry including the power engineering field. Due to the high-volume distribution the package is practically available for all the targeted audiences and computer platforms.

(b) Flexible software structure of MATLAB comprising libraries, models and programs enables one to integrate different model components in one package conveniently.

(c) MATLAB and its time domain solver, SIMULINK, create a friendly and open system. New models and libraries may be just added to the package without a deep knowledge nor modification of the existing parts. This is very useful if the user is involved in continued development of the software (modifications, extensions or customization). This is particularly a case at universities where the students may be assigned, as a part of their education, to develop new or improve existing elements of the package [6].

(d) Fast-development with MATLAB using powerful calculation and visualization means of the package enables one to expand the software quickly and efficiently without developing any extra programming tools by using the existing MATLAB components.

(e) A wide selection of TOOLBOXes—comprehensive collections of pre-defined functions for solving application-specific problems—is already available with MATLAB and seems will be growing even faster in the future. The examples of toolboxes relevant for the protective relaying application include signal processing, optimization, neural network and wavelet toolboxes.

(f) Power System Blockset, one of the latest extensions of MATLAB enabling one to model the basic components of power systems, provides computations similar to EMTP/ATP [7] and introduces more flexibility particularly by permitting modeling of both the power system and its controls in the same environment facilitating closed-loop simulation.

III. The Basic Software Concept

The developed software emulates very closely the general philosophy of SIMULINK. The hardware and software elements of digital relays have been reduced to their generic models and implemented as blocks. The important issues that needed to be solved were:

- how to partition the hardware and software of a relay into blocks keeping the number of blocks at the minimum but provide maximum flexibility of simulation,
- what are the generic models of particular hardware and software blocks enabling accurate simulation,
- what is the variety of options and parameters to be implemented in particular blocks,
- what is the optimal and flexible representation of input/output signals in the relay model (the issue of connecting the blocks).

The aforementioned issues are addressed below.

A. Partitioning a digital relay blocks for modeling

When partitioning a digital relay (i.e. selecting the blocks to be used in modeling) the objectives were:

- to keep the number of blocks at minimum;
- the blocks should correspond to typical hardware and software modules of actual relays;
- the variations within a given block (digital filter, for example) should be accomplished by setting-up options and input data associated with the block, not by generating new versions of the same block.

As a result more than 10 blocks have been developed that can be combined (with occasional support of the standard SIMULINK libraries) into a model of almost any digital protective relay. The detailed review of these blocks is given in section IV.

B. Generic models of relay elements

For each block representing either a part of the relay hardware or software, complete relay or protection system, certain generic model has been assumed. For example, the model of the front end Data Acquisition Board (DAB) provides signal conditioning, analog filtering and sampling. A variety of options has been provided for all the operations included in the model. The structure of the model is fixed. The options of the block, however, enable one to modify the block significantly. For example, the analog filter may be disabled or the sampling frequency may be set very high so that the sampling gets practically disabled.

Similar options as to the functions, structures and options of all the blocks have been made available for the user to select.

C. Variety of options

A variety of options has been assigned to each block in order to model various relay designs. For example, the analog filter of the DAB can be enabled or disabled, its order can be set between 2 and 4, the cut-off frequency is a parameter, the filter's transfer function approximation can be assigned to be of four different types including a "free expression" transfer function (Fig.1). In the same block, the sampling frequency and conditioning gain are parameters. The vertical resolution of the A/D converter can be disabled or enabled. In the later case, the number of bits and the signal range are the parameters. This shows that each block in order to enable modeling of various digital relays can assume tens of different design characteristics depending on the selected options and data.

D. Input and output signals

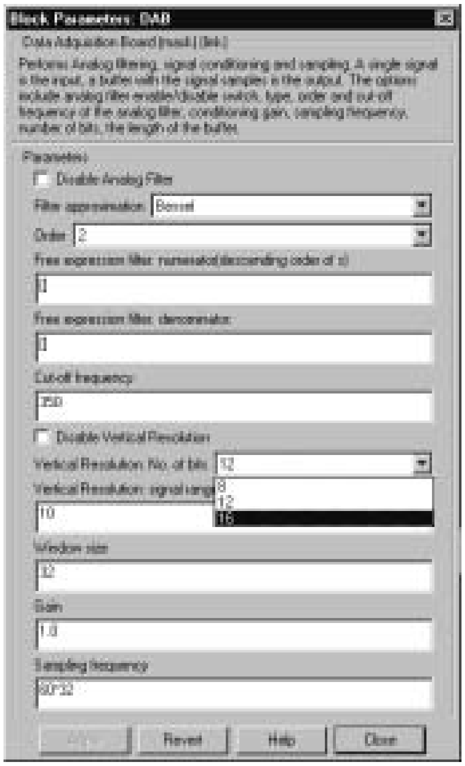
One of the important choices when developing specialized SIMULINK blocks in general, not only for modeling protective relays, is to assume optimal representation of the input/output signals.

In SIMULINK the signals (output-input connections between the blocks) are vectors by default [5]. The Multiplexer and Demultiplexer blocks are used to bundle and debundle several signals into a vector ("cable") when needed.

We followed the aforementioned approach. The following data structures are used for the signals:

- **real signals including logic signals** (time series of real numbers) are represented by regular SIMULINK signals;
- **complex signals** (time series of complex numbers denoting phasors and impedances) are represented by two-element vectors; the angular coordinates (magnitude, angle) have been assumed;

- **data windows** (time series of N real samples) are represented by N-element vectors with the newest sample as the first vector element.



Block Parameters: DAB

Data Acquisition Board (mask) (link)

Performs Analog filtering, signal conditioning and sampling. A single signal is the input, a buffer with the signal samples is the output. The options include analog filter enable/disable switch, type, order and cut-off frequency of the analog filter, conditioning gain, sampling frequency, number of bits, the length of the buffer.

Parameters

☐ Disable Analog Filter

Filter approximation: Bessel

Order: 2

Free expression filter: numerator(descending order of s): 1

Free expression filter: denominator: 1

Cut-off frequency: 350

☐ Disable Vertical Resolution

Vertical Resolution: No. of bits: 12 (8, 12, 16)

Vertical Resolution: signal range: 10

Window size: 32

Gain: 1.0

Sampling frequency: 60*32

Fig.1. Data entry window for the Data Acquisition Board.

In order to ensure connectivity of the blocks, certain data of the interconnected blocks must be compatible. For example, the Digital Fourier Transform (DFT) block has its input in the form of the data window; if the block is connected to the DAB, the later must provide the output window of at least the same length as the input window of the DFT (Fig.2).

E. Programming issues

The basic blocks (**Relay Elements**, **Tools** and **Input Signals**) have been developed as masked subsystems with s-functions [5] performing the required operations. The following approach has been taken:

- The s-functions have been optimized for fast execution: using matrix algebra, avoiding loops and other MATLAB preferred operations have been used.

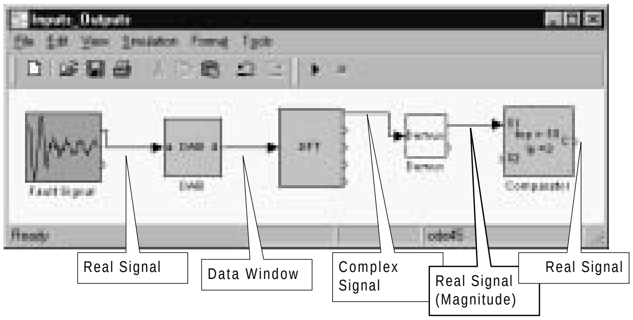


Fig.2. Input and output signals must fit to connect the blocks.

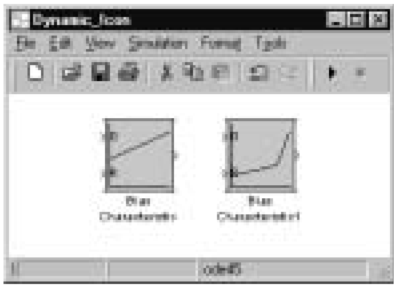


Fig.3. Two copies of the Bias Characteristic (BC) block with two different data have different icons reflecting the data.

- The icon of the mask [5] has been used in a dynamic way: the icon reflects the entered data to a great extent facilitating self-documenting (Fig.3, for example).
- The data entry mechanism of the mask [5] has been extensively used to input the block parameters. Consequently, the locked library contains both the blocks' structures and pre-defined default data.
- The help link of the mask [5] has been used to provide on-line help in the form of meta-text documents (html or reach text format). Fig.4 presents an example.

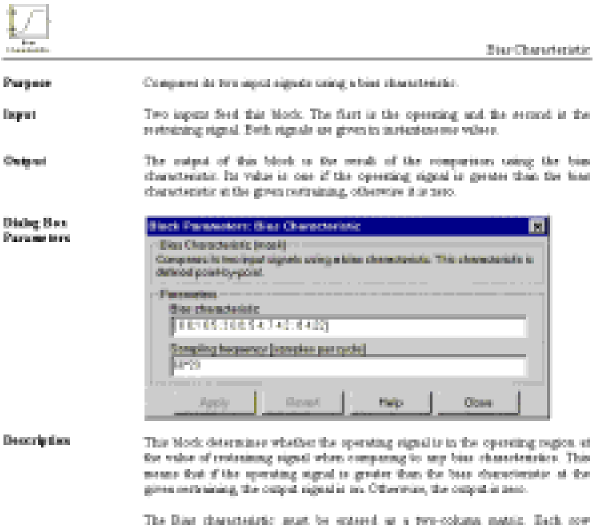
The more complex blocks (**Relays** and **Protection Systems**) have been developed as appropriate connections of the basic blocks. Again, masked subsystems [5] have been used to represent those complex structures.

Some blocks (file converters, for example) have been developed as MATLAB programs (m-files).

IV. RELAY ELEMENTS

This library includes models of the basic hardware and software components of a digital relay such as data acquisition board, digital filters, phasor computation algorithm, comparators, etc.

This SIMULINK library enables one to design complete relays by selecting and connecting models of appropriate elements from the library and setting the parameters. Table I and Fig.5 show the elements of this library.



Bias Characteristic

Purpose Compares its two input signals using a bias characteristic.

Input Two inputs feed this block. The first is the operating and the second is the restraining signal. Both signals are given in instantaneous values.

Output The output of this block is the result of the comparison using the bias characteristic. Its value is one if the operating signal is greater than the bias characteristic at the given restraining, otherwise it is zero.

Dialog Box Parameters

Block Parameters: Bias Characteristic

Bias Characteristic (mask)

Compares its two input signals using a bias characteristic. This characteristic is defined point-by-point.

Parameters

Bias characteristic: [illegible]

Sampling frequency [samples per cycle]: 32*20

Description This block determines whether the operating signal is in the operating region at the value of restraining signal when comparing to any bias characteristics. This means that if the operating signal is greater than the bias characteristic at the given restraining, the output signal is one. Otherwise, the output is zero.

The Bias characteristic must be entered as a two-column matrix. Each row

Fig.4. Sample documentation page (MathWorks' layout applied).

TABLE I. SUMMARY OF THE *RELAY ELEMENTS* LIBRARY

Element	Description
Data Acquisition Board (DAB)	An analog input signal is filtered for antialiasing, conditioned, sampled and forwarded as a data window of signal samples. The options include analog filtering and A/D vertical resolution enable/disable switches; type, order and cut-off frequency of the analog filter; range and number of bits of the A/D converter; conditioning gain and the length of the output data window.
Digital Filter (DF)	An input signal is filtered digitally. The pre-defined filters include Infinite Impulse Response (IIR) low-pass, high-pass, band-pass and band-stop filters of Butterworth, Bessel and Chebyshev approximation as well as Walsh and Fourier Finite Impulse Filters (FIR). "Free-expression" digital filter can also be set.
Digital Fourier Transform (DFT)	An input data window is captured and the phasors of up to five harmonics are calculated using the DFT technique. The options include selection of the frequency of the reference (first) harmonic and other requested harmonics.
Basic Measurement (BM)	The voltage and current phasors are captured and the amplitudes, impedance components and power are calculated. The post-filtering may be applied using either mean or median filters (individually for each output quantity).
Differential Equation based Impedance Measurement (DEIM)	The block measures the impedance based on the differential equation approach. The pre-filtering using either Walsh or Fourier filters of selecteble window length may be applied. The post-filtering using either mean or median filters may be applied. Either Euler or trapezoidal method of numerical differentiation may be used.
Universal Comparator (UC)	The block is fed by two signals and does a comparison between either the signal and a threshold, the signal and time or the two signals. The direction of comparison may be alternated and the standard time characteristics are included.
Zone Comparator (ZC)	The block is fed by the resistance and reactance values and emulates four forward and one reverse impedance zones. Either the mho or "free-expression" shapes may be set.
TRiggering element (TR)	The block is fed by the data window of a signal and acts as a transient detector. The implemented methods include sample-to-sample, cycle-to-cycle and value-to-threshold checking.
Symmetrical Components (SC)	The block is fed by three phase signals and produces three symmetrical components signals. Either phasors or instantaneous values are utilized.
Vector Group Compensator for two-winding transformers (VG-2)	The block captures instantaneous values of six phase currents of a three-phase two-winding transformer and computes the differential and restraining currents. The parameters include the CT ratios, transformer rated voltages, connection type and vector group.
VG-3	Similar to VG-2, but for three-winding transformers.
Bias Characteristic (BC)	The block is fed by the operating and restraining signal and applies a "free-expression" (point by point defined) bias characteristic.

TABLE II. SUMMARY OF THE *INPUT SIGNALS* LIBRARY

Sub-Category	Elements	Structure
Signal Generators	General specification generator, limited frequency spectrum generator, phasor generators, etc.	Library
Data File Converters	ATP to MATLAB, COMTRADE to MATLAB, DFR to MATLAB	Programs
Power System Transient Model	Power System Blockset, Instrument Transformers, internal fault models	Libraries

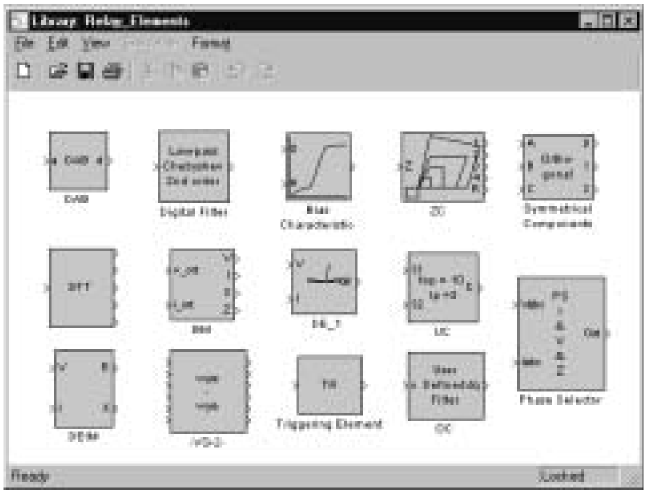


Fig.5. **Relay Elements** library.

V. INPUT SIGNALS

This part of the software enables generating the input signals for protective relays. This includes transient modeling of the power system for closed-loop simulation (link to ATP and MATLAB's Power System Blockset) as well as the analytical way of generating the signals. In addition, a file format conversion for data from major transient file formats (COMTRADE [8] and MATLAB's native format) is also provided.

This part of the software contains MATLAB programs (file converters), SIMULINK libraries (such as the Power System Blockset or models of instrument transformers) and SIMULINK blocks (such as specialized signal generators).

Table II provides a brief summary of this class while Fig.6 shows some of the SIMULINK blocks.

VI. TOOLS

This library contains elements that facilitate the analysis and testing procedures.

The Phasor Display is perhaps the best example (Fig.7). This block can be fed by up to six phasors (multiplexed into a single input vector). It displays them in both the graphical and numerical forms. The block can be disabled (present in the model but not active). The display can be of the transient type (the plot refreshed every given time in-

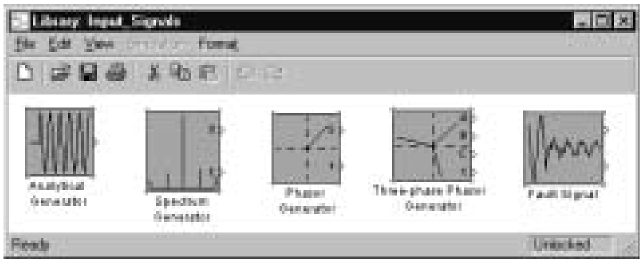


Fig 6. **Input Signals** library (signal generators only).

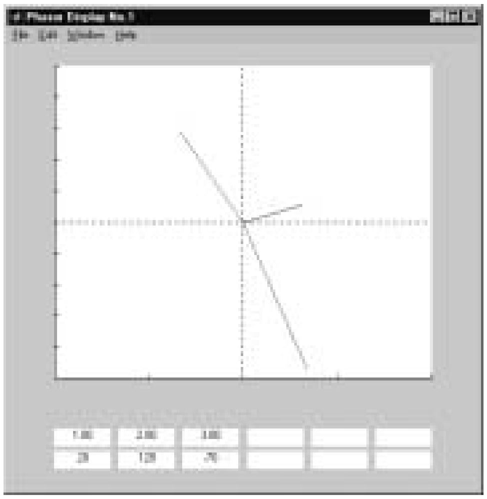


Fig.7. Window of the Phasor Display block (phase angles in degrees).

terval) or of the steady state type (the values are displayed when the simulation terminates). To use the display one puts its block in the model, connects the input and sets required options. Once the simulation is started and the block is set to active, a new window pops-up and the plot is created.

VII. Relays and Protection Systems

These two libraries contain pre-set models of complete digital relays and protection terminals such as overcurrent, impedance and differential relays.

Those models are composed from the blocks of the **Relay Elements** library and the general SIMULINK libraries. Some of the relay models emulate actual relays to the extent possible using the publicly available design details.

Figs.8, 9 and 10 present an example of a three-phase overcurrent relay featuring three-phase instantaneous and inverse-time elements as well as instantaneous residual overcurrent element. In this model:

- The DAB block is used to model analog filters, signal conditioners and the A/D converter (Fig.10). The following design parameters are entered in this block: parameters of the filters (order, cut-off frequency, type of approximation), signal conditioner (gain) and the A/D converter (sampling rate, number of bits, signal range).
- The Digital Filter (DF) block is used to filter the phase currents digitally. Type of the filter, its order and cut-off frequencies are specified in this block (Fig.10).
- The Symmetrical Components (SC) block is used to extract the residual current from the phase currents (Fig.9).
- The Universal Comparator (UC) block is used to specify both the instantaneous and time-delayed overcurrent characteristics. The relay settings are entered here (Figs.9 and 10).
- The standard SIMULINK elements are used to deal with the trip and target signals.

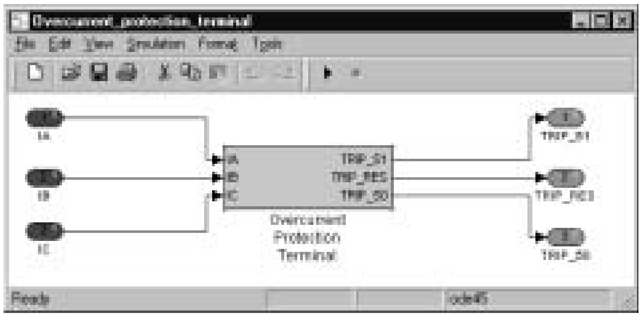

Overcurrent_protection_terminal
File Edit View Simulation Format Tools
IA
IB
IC
TRIP_51
TRIP_RES
TRIP_50
Overcurrent Protection Terminal
TRIP_51
TRIP_RES
TRIP_50
Ready
ode45

Fig.8. Overcurrent relay model (IA, IB, IC - input currents, TRIP_51, TRIP_50, TRIP_RES - targets).

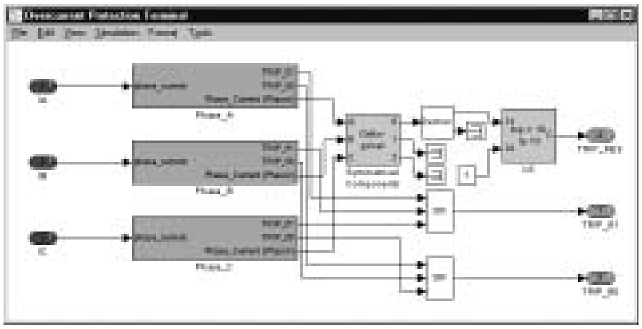


Fig.9. Content of the relay model of Fig.8.

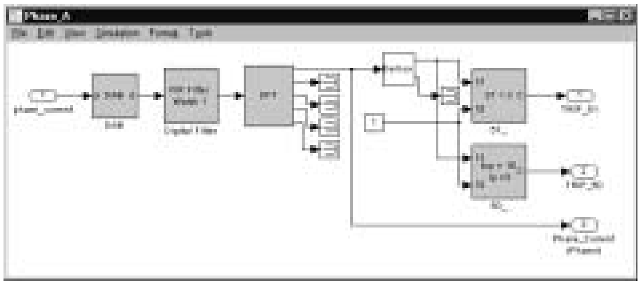


Fig.10. Content of the phase A channel.

As seen from the figures this model can be assembled from the available libraries in a very convenient and quick way. Modifications of its structure and parameters are very easy. Once developed, the model can be "zoomed-out" to a subsystem [5], get assigned a name and icon and may be included into a library of **Relays** as a new block (Fig.8). This approach speeds-up the model development phase significantly.

In the same way a larger structures of **Protection Systems** are built out of **Relays** and **Relay Elements**.

Protective relays or protection systems modeled using the developed libraries can be tested with the input signals from the following sources:

- signal generators of the Input Signals library
- PowerSystem Blockset of MATLAB
- ATP output files via the developed file converter
- COMTRADE [8] transient data files via the developed file converter

The results of testing (in terms of transient data) can be saved in native MATLAB format or in COMTRADE [8] via the developed output filter.

As of now, the closed-loop testing is possible only using the PowerSystem Blockset as shown in section VIII.

VIII. Simulation Examples

As an example, the relay of Fig.8 is tested in a simple power system shown in Fig.11.

The relay is fed with a phase-A current. Its TRIP signal is connected to open the circuit breaker and to stop the simulation two cycles later.

The line model has incorporated an internal fault. The fault location is a line model parameter. The fault model is attached to the line block and enables one to specify the fault resistance (Fig.11).

Fig.12 presents sample plots from a test — the relay trips in about 55msec.

IX. Conclusions

This paper presents new SIMULINK libraries for modeling, optimizing, setting and testing digital protective relays.

The developed blocks facilitate setting-up model of protective relays using generic representation of various hardware and software elements of actual relays.

Models of larger structures such as complete relays and protection systems can be set from the available block models. Such a hierarchical structure and openness of the software enable fast development of the user libraries with his/her own models and test studies.

Certain tools have been developed to facilitate the modeling and testing activities including specialized signal generators, specialized display and analysis tools.

The package is well documented using MathWorks' standards and layouts.

The software was tested in the university environment as a teaching tool [6] and the results are encouraging and very promising.

X. Acknowledgments

This project is jointly funded by the National Science Foundation, Grant No. ECS-96-19294 and the Electric Power Research Institute (EPRI). The project is also co-founded by the following industry partners: ABB-TTI, Bailey Network Management, Houston Lighting & Power, Incremental Systems, Pacific Gas & Electric, Puget Sound Power & Light, Schweitzer Engineering Laboratories, TU Electric. The project activities are a joint effort between Texas A&M University (Professors M.Kezunovic, A.Abur and G.Huang) and Washington State University (Professors A.Bose, K.Tomsovic, M.Venkatasubramanian). The participation of Texas A&M University students R.Martinez-Lagunes and N.Suphasan is also acknowledged.

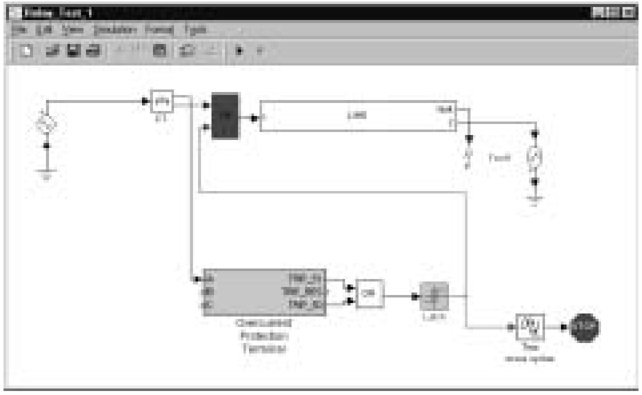


Fig.11. Closed-loop test of the relay of Fig.8 in a simple power system model.

XI. References

[1] M.Kezunovic, "Advanced Simulation And Testing Environment For Design, Application And Performance Assessment Of Protection Equipment", *Proceedings of the 6th International Conference on Developments in Power System Protection*, pp.394-397, March 25-27, 1997, Nottingham, UK.

[2] M.Kezunovic, Y.Q.Xia, Y.Guo, C.W.Fromen, D.R. Sevcik, "An Advanced Method for Testing of Distance Relay Operating Characteristic," *IEEE Transactions on Power Delivery*, Vol.11, No.1, pp.149-157, January 1996.

[3] M.Kezunovic, Y.Q.Xia, Y.Guo, C.W.Fromen, D.R. Sevcik, "Distance Relay Application Testing Using a Digital Simulator," *IEEE Transactions on Power Delivery*, Vol.12, No.1, pp.72-82, January 1997.

[4] IEEE Power System Relying Committee Working Group C1 chaired by P.McLaren, "Software Models of Relays", Draft No.7, December 1998.

[5] *MATLAB. Reference manual*, MathWorks, Inc., 1997.

[1] M.Kezunovic, B.Kasztenny, "New Educational MATLAB software for Teaching Protective Relaying Courses", *Proceedings of the IV Simposio "Iberoamericano Sobre Proteccion de Sistemas Electricos de Potencia*", pp.1-6, 1998, Monterey, Mexico.

[2] H.Dommel, *ElectroMagnetic Transient Program*, BPA, Portland, Oregon, 1986.

[3] IEEE Standard C 37.111-1991: "Common Format for Transient Data Exchange (COMTRADE) for Power Systems", *Guides and Standards for Protective Relaying Systems*, IEEE 1995.

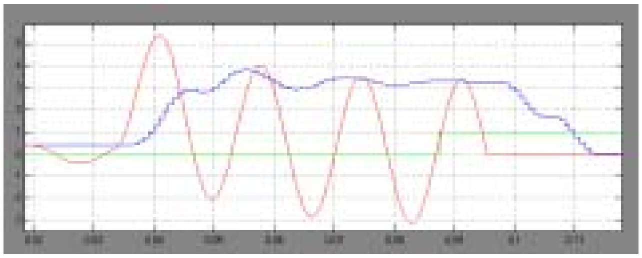


Fig.12. Sample test results for the relay of Fig.8 tested in the simple system of Fig.11 (phase A current, its magnitude as measured by the relay and the TRIP signal).