

USER-FRIENDLY, OPEN-SYSTEM SOFTWARE FOR TEACHING PROTECTIVE RELAYING APPLICATION AND DESIGN CONCEPTS

Mladen Kezunovic, IEEE Fellow

Abstract--This paper describes modeling and simulation software developed specifically for teaching protective relaying application and design concepts. The emphasis was on implementing user-friendly and open-system solution that will allow an easy use and straight-forward future expansion. This is achieved by introducing new libraries of signal sources and relay elements developed for the SIMULINK environment of MATLAB. Combined with the Power Block Set (PBS) toolbox of MATLAB, the mentioned libraries allow for a variety of studies aimed at better understanding protective relay design approaches and related applications.

Index Terms: Protective Relaying, Faults, Transients, Relaying schemes, Time-domain simulation, Electromagnetic transient program

I. INTRODUCTION

Protective relaying is a complex electrical engineering field requiring knowledge from a number of diverse engineering disciplines such as electromagnetics, circuits, power systems, computers, communications, signal processing, controls, intelligent systems, etc. Teaching protective relaying application and design concepts involves a multidisciplinary approach where the physics of the fundamentals together with diverse mathematical formulations need to be merged in a common framework. One convenient and yet powerful way is to use modeling and simulation as the common framework [1]. To be efficient, such a framework needs to be user-friendly and amenable to future expansions as the continued needs to incorporate new developments grow. This paper describes a development of the software aimed at meeting such requirements.

Recently published IEEE survey paper discussing different approaches to modeling protective relays and related power system events indicates a variety of possible software tools that may be used for this purpose [2]. Well known software packages can be used to simulate power

system faults in both the time-domain (electromagnetic transients) [3,4,5] and phasor (unbalanced steady state) [6,7,8] modes. Some of the packages have the provision for incorporating protective relay models developed by the user [9,10] while the others have the generic relays models already included [6,7,8]. In all the mentioned cases it is difficult to add the modeling and simulation features to teach specific protective relaying concepts that go beyond the level of detail originally provided by the software. Additions of new relay models and implementations of specific fault scenarios as well as the flexibility of changing the way the models of relays and power system interact is constrained by the specific user and programming interface rules embedded in the existing software.

To facilitate the mentioned user tasks, a specialized teaching software environment is developed using a commercial software that was built to allow for flexibility in providing user interfaces and implementing custom models. The MATLAB software package with SIMULINK support and Power Block Set toolbox [11,12,13] is utilized to develop customized model libraries for teaching protective relaying concepts. This paper provides details of this development and indicates how the user-friendly interface and open-system goals are achieved.

The paper starts with discussion of the reasons for using MATLAB and associated tools. Developments of new libraries are described next. The use of the overall solution in teaching the relaying concepts is also presented. Conclusions are given at the end.

II. MODELING PROTECTIVE RELAYS IN MATLAB

MATLAB has been selected as the programming shell for the new protective relay modeling software. Particular factors that support the selection of MATLAB are [11-13]:

- (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.
- (b) A 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,

This work was supported in part by the National Science Foundation under Grant ECS-96-19294 and in part by EPRI under Contract WO 8618-02. Contribution of Dr. B. Kasztenny and Dr. Z. Galijasevic, while they were at TAMU as Research Staff, is also acknowledged.

M. Kezunovic is with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843-3128 (email: kezunov@ee.tamu.edu).

create a friendly and open system. New models and libraries may be just added to the package without deep knowledge nor modification of the existing parts. This is very useful if the user is also involved in continued development of the software.

- (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.
- (e) A wide selection of TOOLBOXes, comprehensive collections of pre-defined functions for solving application-specific problems, is already available with MATLAB and is likely to grow even faster in the future.
- (f) Power System Blockset, one of the latest extensions of MATLAB, is enabling modeling the basic components of power systems. It provides computations similar to EMTP/ATP permitting modeling of both the power system and its controls in the same environment, and thus, facilitating closed-loop simulation.

II. 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 while providing 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, over a dozen blocks have been developed so far. They may 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 III.

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

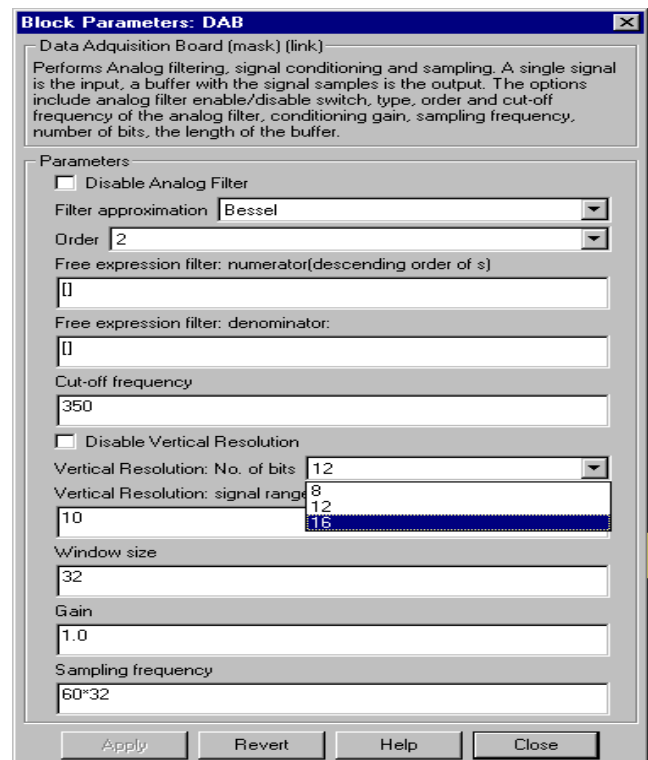


Fig. 1. Data entry window for the Data Acquisition Board.

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 the parameters. The vertical resolution of the A/D converter can be disabled or enabled. If enabled, 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 dozens 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 [11]. The Multiplexer and Demultiplexer blocks are used to bundle and de-bundle several signals into a vector ("cable").

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.

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 latter must provide the output window of at least the same length as the input window of the DFT (Fig.2).

III. PROGRAMMING ISSUES AND SOFTWARE LIBRARIES

The basic blocks (**Relay Elements, Tools and Input Signals**) have been developed as masked subsystems with s-functions performing the required operations. The following approach has been taken [11,12]:

- The s-functions have been optimized for fast execution: using matrix algebra, avoiding loops, and other MATLAB preferred operations have been used.
- The icon of the mask 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 has been extensively used to input the block parameters. Consequently, the locked library contains both the blocks' structures and pre-defined default data.

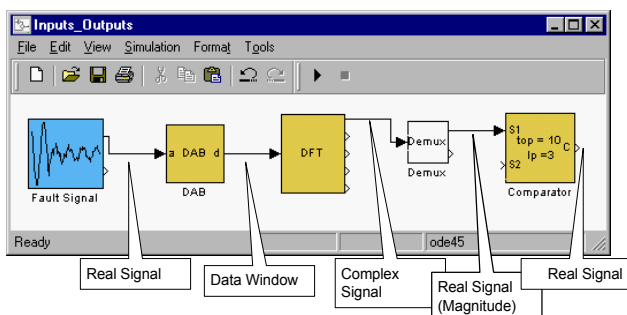


Fig.2. Input and output signals must fit to connect the blocks.

- The help link of the mask has been used to provide on-line help in the form of meta-text documents (html or rich 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 have been used to represent those complex structures.

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

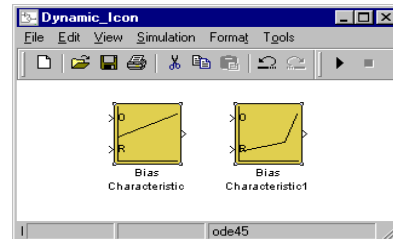


Fig.3. Two copies of the Bias Characteristic (BC) block with two different data have different icons reflecting the data.

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	
Description	<p>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 on. Otherwise, the output is zero.</p> <p>The Bias characteristic must be entered as a two-column matrix. Each row</p>

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

A. 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.

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 selectable 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 impedance zones and a reverse one. 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

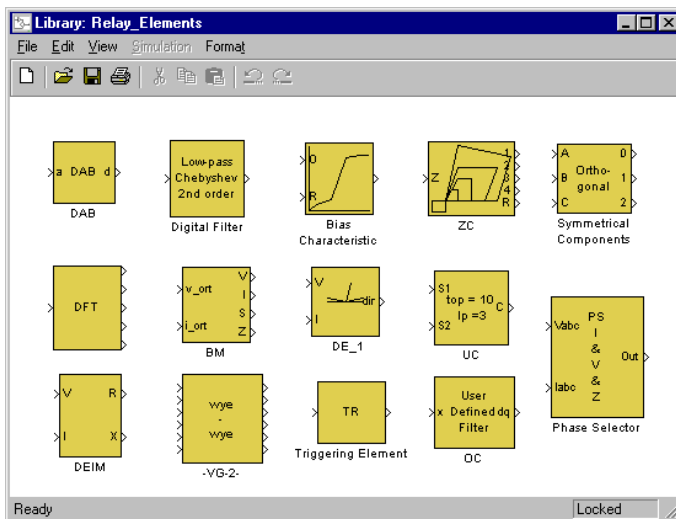


Fig. 5. Relay Elements library.

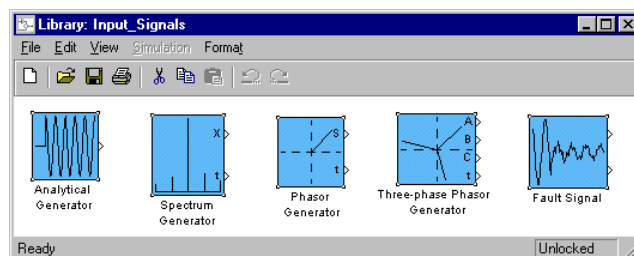


Fig. 6. Input Signals library (signal generators only).

B. 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 [14] 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.

C. 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 interval) 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.

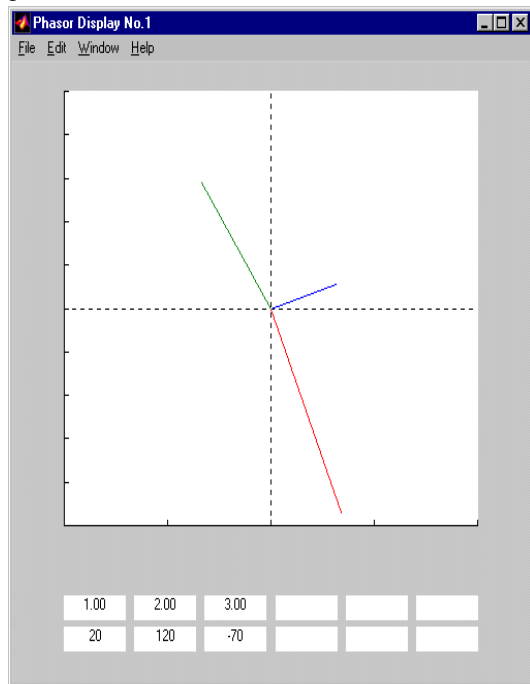


Fig. 7. Window of the Phasor Display Block (phase angles in degrees).

D. Relays and Protection Systems

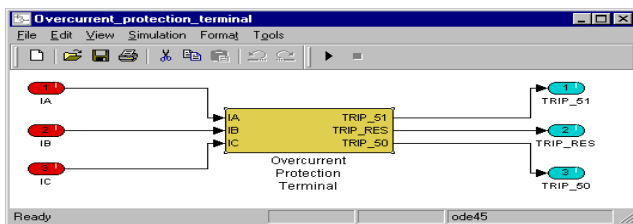


Fig.8. Overcurrent relay model (IA, IB, IC - input currents, TRIP_51, TRIP_50, TRIP_RES - targets).

These two libraries contain pre-set models of complete 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:

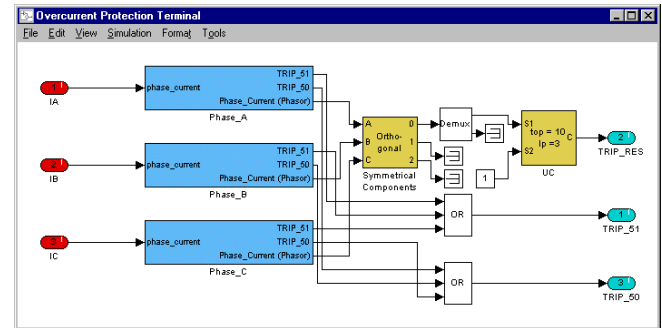


Fig. 9. Content of the relay model of Fig. 8.

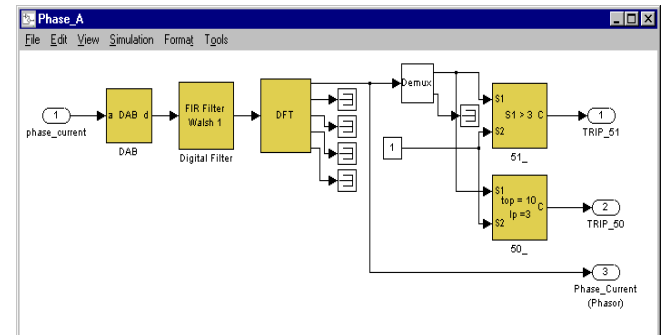


Fig. 10. Content of the Phase A Channel.

- The DAB block is used to model analog filters, signal conditioners and the A/D converter (Fig 10). The following parameters of the filters (design parameters are entered in this block: 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.

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 [11], 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 transient data files via the developed file converter

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

IV. TEACHING EXAMPLES

Having the package developed and capable of performing emulation of both the power system and related protective relays, its usage in education and training needs to be carefully designed and planned.

This section presents teaching examples of different complexity levels and different perspectives.

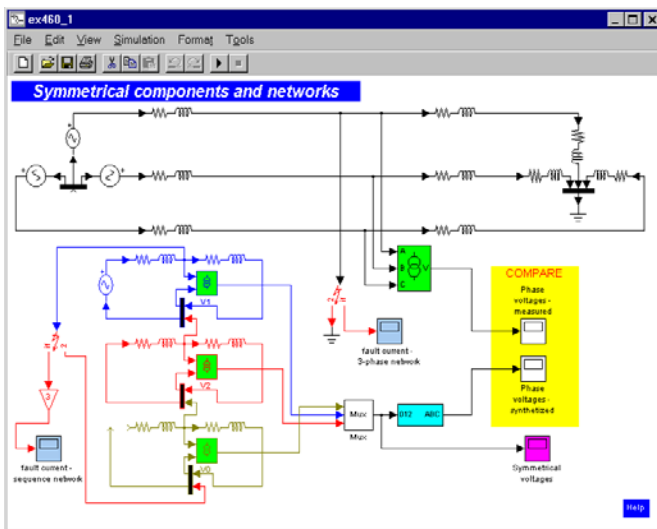


Fig. 11. Teaching example for symmetrical components and networks.

A. Symmetrical components and sequence networks

Fig.11 presents a pre-defined SIMULINK model for the topic of symmetrical components and networks within the class *Input Signals*. In this model:

1. The three-phase one-machine power system is set-up using the Power System Blockset.
2. Three sequence networks are created and connected to reflect the type and location of a fault occurring in the three-phase system.
3. Measuring devices are placed in both the original system and sequence networks to facilitate comparison between the signals from the original three-phase system and sequence networks. This includes ideal voltage transformers, a converter from the phase values to symmetrical components, and displays.

The configuration and parameters of the entire model may be changed by cut-and-paste and drag-and-drop manipulations with the icons.

The student assignments for this example include:

- Selecting the three-phase system in terms of parameters and connections (especially the way of grounding the neutral points); changing the type, parameters and location of the fault and observing the values and relations between the symmetrical components (studying the power system in unsymmetrical fault states).
- Editing the original three-phase system, changing the fault type and parameters and editing the sequence component networks in terms of parameters and connections to reflect the changes in the original system. Comparing the results from the original system and its sequence-network-based "solver" (learning how

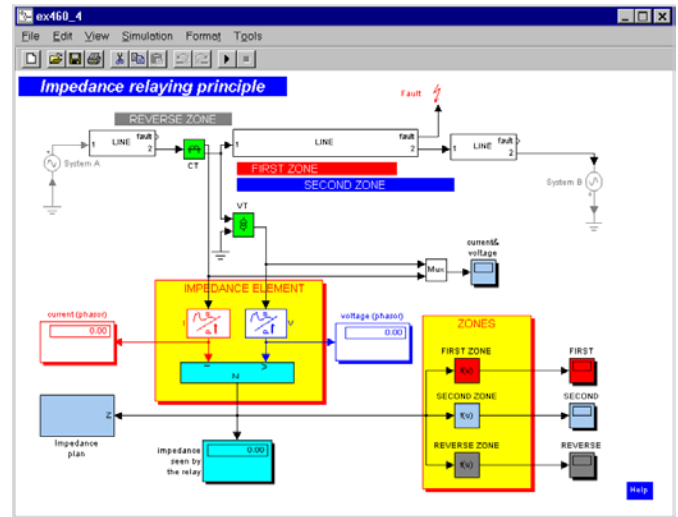


Fig. 12. Teaching example for impedance relaying principle.

to use sequence networks).

By this example the students learn:

- (a) The relation between the type and parameters of a fault and values of symmetrical components.
- (b) The rules of using the sequence networks for analysis of three-phase systems.

B. Impedance relaying principle

Fig. 12 introduces a pre-defined SIMULINK model for the topic of impedance relaying principle within the class *Relaying Principles*. The model consists of:

1. A single-phase model of a two-machine three-line power system with models of the fault and instrument transformers.
2. Measuring blocks estimating the voltage and current phasors as well as impedance components.
3. Zone comparators checking the impedance location against three impedance zones.
4. Displaying elements.

The student assignments for this example include:

- Placing faults at different locations and observing the value of the impedance. Introducing the ideas of a *relaying point*, a *protected line* and *impedance seen* by the relay.
- Simulating faults at the far end busbar with non-zero fault resistance and different pre-fault power flows.

Explaining the in- and out-feed effects. Recommending the relay settings.

- Simulating faults at the relaying point. Recommending the relay settings. Explaining the need for a directional element.
- Setting the first zone of the relay. Testing the relay. Introducing the *sensitivity* and *selectivity* concepts.
- Simulating external faults. Understanding the idea of a *back-up protection*. Setting the second and reverse zones.

By this example the students learn:

- (a) The basics of the impedance relaying principle.
- (b) Certain relay design issues.
- (c) Basics of impedance relay setting.

The two examples were successfully utilized in an

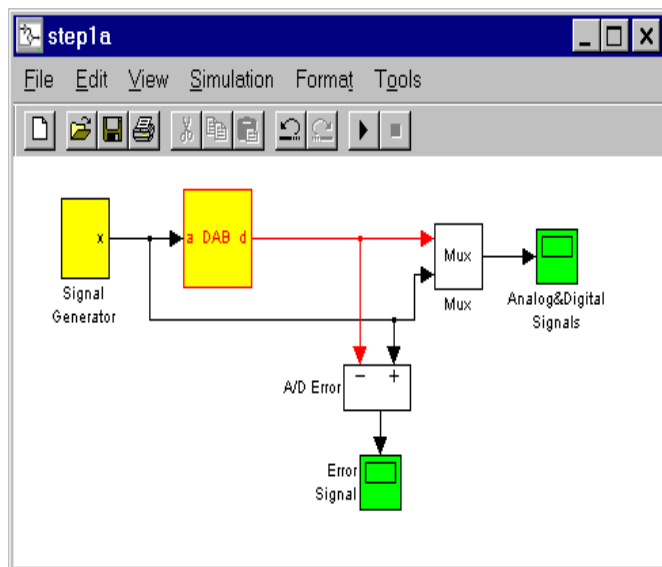


Fig. 13. Teaching example for analog filtering and sampling.

undergraduate course at Texas A&M University in the form of a two-hour supervised laboratory exercises and take-home assignments.

C. Analog filtering and sampling

Fig. 13 presents a pre-defined SIMULINK model for the topics of analog filtering, signal conditioning and sampling within the class *Design Principles*. This model consists of the Data Acquisition Block (DAB - see Table I) and few auxiliary elements.

The student assignments for this example include:

- Simulating and explaining the aliasing phenomenon in both the time and frequency domains.
- Selecting an analog filter. Investigating its effects on both the noise and information.
- Selecting the sampling frequency.
- Simulating and explaining the phenomenon of vertical resolution of an A/D converter.
- Testing the complete front-end part of a digital relay with analytical and EMTTP-generated signals.

This example was successfully utilized in the short-course activities at Texas A&M University in the form of

computer-assisted lecture and a supervised hands-on laboratory exercise.

V. CONCLUSIONS

Discussion given in the paper explains the following major outcomes of this development:

- The use of MATLAB and SIMULINK has enabled user friendly, open-system environment.
- The user friendly features are facilitated with the use of a simple drag-and-drop and cut-and-paste approach to building the exercises out of pre-defined library elements.
- The open-system capability is enhanced through possibility of adding newly developed library elements to the existing MATLAB or the new customized protective relaying libraries.

VI. REFERENCES

- [1] M.S. Sachdev, M. Nagpal, and T.Adu, "Interactive Software for Evaluating and Teaching Digital Relaying Algorithms," *IEEE Transactions on Power Systems*, Vol 5, No. 1, Feb. 1990, pp 346-352.
- [2] P.G. McLaren, C. Henville, V. Skendzic, A. Girgis, M. Sachdev, G. Benmouyal, K. Mustaphi, M. Kezunovic, Lj. Kojovic, M. Meisinger, C. Simon, T. Sidhu, R. Marttila, D. Tziouvaras, "Software Models for Relays," *IEEE Transactions on Power Delivery*, Vol. 16, No.2, pp. 238-246, April 2001.
- [3] CanAm EMTTP User Group, *Alternative Transient Program (ATP) Rule Book*, Portland, 1992.
- [4] EMTTP Reference Manual, BPA, August 1986.
- [5] EMTDC User's Manual, Manitoba HVDC Resource Center, 1996.
- [6] CAPE User's Manual, Electrocon International, 1997.
- [7] Tractabel-EDF, "EUROSTAG Software," Release Notes, ver. 4.1, December 2000.
- [8] ASPEN User's Manual, ASPEN Inc., 1995.
- [9] Chul-Hwan Kim, Myung-Hee Lee, Raj K. Agarwal, Allan T. Johns, "Educational Use of EMTTP MODELS for the Study of a Distance Relaying Algorithm for Protecting Transmission Lines," *IEEE Transactions on Power Systems*, Vol. 15, No. 1, February 2000.
- [10] Richard W. Wall, Brian K. Johnson, "Using TACS Functions within EMTTP to Teach Protective Relaying Fundamentals," *IEEE Transactions on Power Systems*, Vol. 12, No. 1, February 1997.
- [11] The MathWorks, Inc., *Using MATLAB*, January 1999.
- [12] The MathWorks, Inc., *Using Simulink*, Version 3, 1990.
- [13] The MathWorks, Inc., *Power System Blockset User's Guide*, Version 1, 1999.
- [14] 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.

Mladen Kezunovic (S'77, M'80, SM'85, F'99) 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 relay testing as well as application of intelligent methods to power system monitoring, control and protection. Dr. Kezunovic is a registered professional engineer in Texas. He is also a Fellow of IEEE and member of CIGRE-Paris.