Implementing an Intelligent Error Back Propagation (EBP) Relay in PSCAD^{TM}/EMTDC 4.2.1

E. William, *IEEE Student Member,* Brian K Johnson, *IEEE Senior Member, M. Manic, IEEE Senior Member*

***Abstract*— Power Systems Computer Aided Design (PSCAD^{TM}) is a graphical user interface for the Electromagnetic Transient Direct Current (EMTDC) type program. PSCAD^{TM}/EMTDC is used in this paper to simulate the injection of static single-line-to-ground (SLG) faults located at 120%, 100%, and 80% of the cable length on the ship, and protect the electrical system using the error back propagation (EBP) algorithm-based relay. Sequential and combinational digital logic is used to design and implement the EBP Relay in PSCAD^{TM}/EMTDC.**

***Index Terms*-- Algorithms, Artificial neural networks, Fault currents, Fault diagnosis, Fault location, Power system modeling, Power system stability, Protective relaying**

I. Introduction

THIS paper implements an error back propagation (EBP) relay in Power Systems Computer Aided Design (PSCAD^{TM}). PSCAD^{TM} is a graphical user interface for the Electromagnetic Transient Direct Current (EMTDC) program. The EBP algorithm-based relay is designed using sequential logic network diagrams in Fig. 3. These diagrams allow the design of the EBP-based digital logic circuit that is a function as a mho relay protection device in PSCAD^{TM}/EMTDC. This EBP-based relay is implemented for a shipboard electrical distribution system [1] and is designed to simulate responses to single-line-to-ground (SLG) faults as shown in Fig.1. The SLG faults are statically injected at 120%, 100%, and 80% of the shipboard electrical distribution (power system) line as described in Fig. 1. In addition, the EBP-based relay model can search for these static SLG fault locations used on simulated data in the PSCAD^{TM}/EMTDC simulation environment.

II. Technical Work Preparation

PSCAD^{TM}/EMTDC version 4.2.1 Educational used here. The design of the EBP relay is discussed in this section.

A. Sequential Logic Diagram

The first step is to define a series of inputs for the sequential logic network representing the artificial neural network (ANN) as in Table I. The resistance from is labeled "A" and the reactance from is labeled "B" as shown in Table I.

TABLE I
Variables for Sequential Logic Network Design

A	Resistance (R)
B	Reactance (X)
C	Fault at 80% of line
D	Fault at 100% of line
E	Fault at 120% of line
Y	Output

Table I displays the resistance "A' and reactance "B" inputs imported into the artificial neural network (ANN) diagram displayed in Fig. 2. In addition, Table I displays inputs "C", "D", and "E" are static single-line-to-ground (SLG) fault insertions at particular locations on the shipboard electrical distribution system. The input "C" is a fault inserted at 80% of the line, Input "D" is a fault insertion at 100% of the line, and input "E" is inserted at 120% of the line. Finally, "Y" is the output that controls the response. Each of these variables is passed to a sequential logic fault-locating diagram as in Fig. 3.

An artificial neural network (ANN) is a network of neurons that each have thresholds that control if a "1" or "0" is set. A neuron is very much like "AND", "OR" and "XOR" digital logic gates. Thresholds help determine how a neuron reacts, and simulates digital logic gate function. Moreover, digital logic gates thresholds define if the gate is a "AND", "OR, or "XOR" gates. High thresholds may render a "AND" truth table. In turn, low thresholds may render an "OR" [2]. The

This paper was financially supported by the University of Idaho, Moscow, ID 83844 USA

Affiliation footnotes:
Edward James William II is with the Department of Electrical and Computer Engineering at the University of Idaho, Moscow, ID 83844-1023 USA (e-mail: edward.william@vandals.uidaho.edu).
B.K. Johnson is with the Department of Electrical and Computer Engineering at the University of Idaho, Moscow, ID 83844-1023 USA (e-mail: bjohnson@ece.uidaho.edu).
M Manic is with the Computer Science Department at the University of Idaho, 1776 Science Center Drive, TAB Ste.#241
Idaho Falls, ID 83402 USA (e-mail: misko@uidaho.edu)

ANN network (displayed in Fig. 2) helped to derive a reduced state machine diagram in Fig. 3.

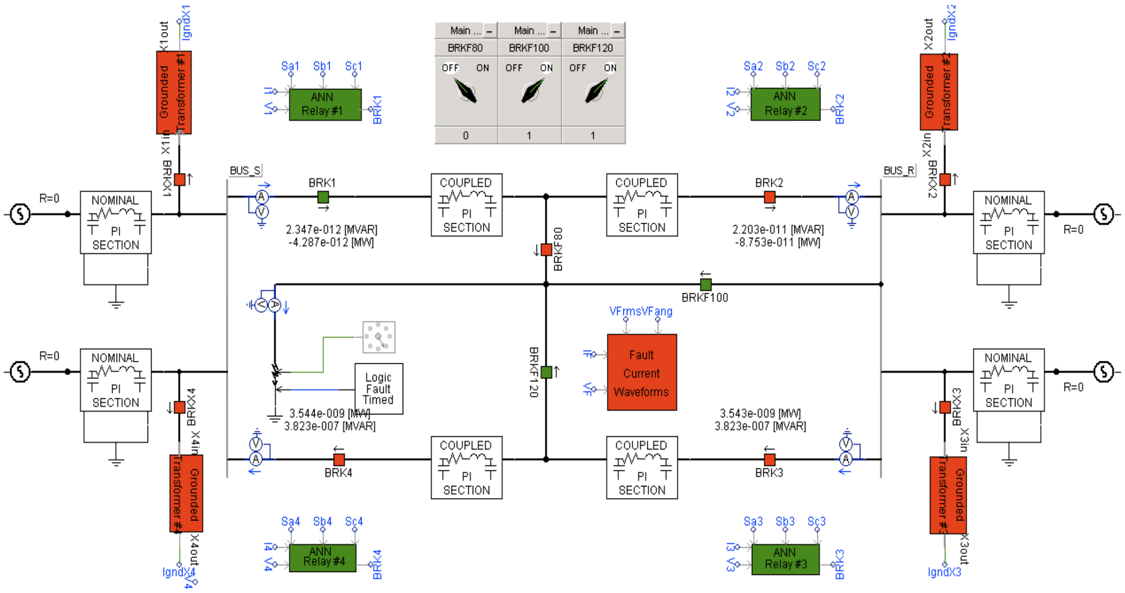


Fig. 1. Shipboard power system modeled in PSCAD^{TM}/EMTDC.

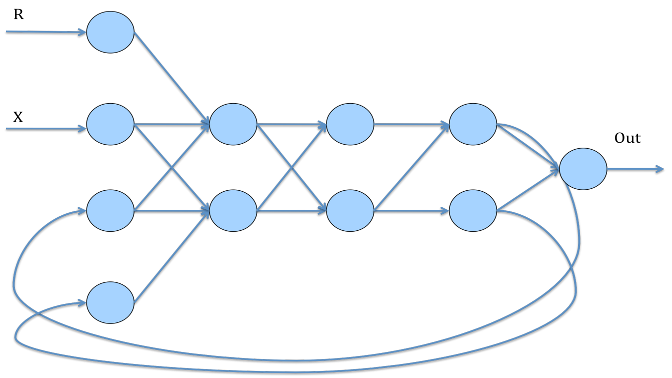


Fig. 2. The ANN diagram displays the error back propagation (EBP) algorithm. This uses the characteristic impedance values from PSCAD^{TM}/EMTDC.

Theoretically, The error back propagation (EBP) algorithm starts at a high error value at the initial state of the system, of S_0, in Fig. 3 as output "Y" is set to "0".

The reduced sequential logic network searches for a large increase in resistance or reactance values of inputs "A" and "B" from Table 1. If inputs "AB" has a large increase in value then value is "1" and state changes to S_1 in Fig. 3. If there is no change in and value is "0" for inputs "AB" the state returns to S_0 in Fig. 3. From S_1 output "Y" remains "0" and an option occurs of search for a fault at less than or equal to 80% of the line. If this condition is satisfied while in state S_1 then the state changes to state S_3 and output Y is set to "1". This means the fault is located at less than or equal to 80% of the line in Fig. 3. When Y is set to "1" a trip is generated. However, the other option is the diagram in Fig. 3 is if a fault exists at greater than 80% of the line. If this case is satisfied then the states change from S_1 to S_2 and output "Y" remains "0". When Y is set to "0" a no trip is generated. This means that although a fault exists it still needs to be located and the search for fault must continue resulting in "0" set to no trip status until fault is located within a smaller range. At S_2 if a fault occurs at less than 100% of the line the state changes to state S_3 and the output "Y" changes to "1". This means the fault is located between 80% and 100% of the line. However, the other option in state S_2 if a fault is located at greater than 100% of the line the state changes to S_4 and the output "Y" remains "0".

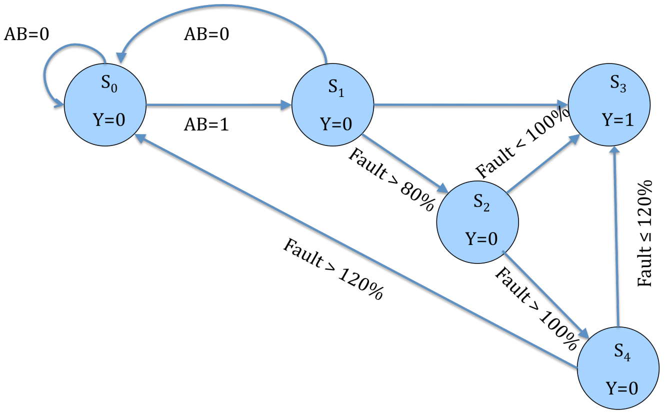


Fig. 3. Reduced and redesigned sequential logic diagram used as a state machine [2].

Finally, at state S_4 if the state changes to state S_3 the output "Y" changes to "1". This means that a fault exists between 100% and 120% of the line. On the contrary, at state S_4 if a fault exists at greater than 120% of the system line that is beyond the scope of the search. This results in the return to S_0 where the output “Y” remains “0” and the error remains constant resulting in a back propagating search for a fault. This means that no fault exists on the system and will result in a loop back to S_0 for the iterations of the search performed previously. The search will continue to propagate through each state until a fault is located. If no fault is located it will loop to state S_0 until a fault is found. The neurons are being trained to locate a SLG fault in real time and the errors will decrease as the system learns the EBP algorithm.

B. Truth Table

As described in the sequential logic diagram section all steps taken from states S_0, S_1, S_2, S_3 and S_4 can be tracked on Table II. The truth table of Table II contains inputs that are represented as: “A”, “B”, “C”, “D”, and “E” from Table I contribute to the state diagram of Fig. 2 and how each state transition in Table II. How this truth table works is the inputs are represented as “1” active, “0” inactive and “X” as don’t care.

TABLE II
SEQUENTIAL LOGIC TRUTH TABLE [2]

	A	B	C	D	E	S_0	S_1	S_2	S_0’	S_1’	S_2’	Y
S_0	1	1	X	X	X	0	0	0	0	0	1	0
	1	0	X	X	X	0	0	0	0	0	0	0
	0	1	X	X	X	0	0	0	0	0	0	0
	0	0	X	X	X	0	0	0	0	0	0	0
S_1	X	X	0	X	X	0	0	1	0	1	0	0
	X	X	1	X	X	0	0	1	0	1	1	0
S_2	X	X	X	0	X	0	1	0	1	0	0	0
	X	X	X	1	X	0	1	0	0	1	1	0
S_3	X	X	X	X	X	0	1	1	0	1	1	1
S_4	X	X	X	X	0	1	0	0	0	0	0	0
	X	X	X	X	1	1	0	0	0	1	1	0

The states S_0, S_1, and S_2, are also feedback inputs. These feedback inputs are the error back propagation (EBP) inputs that are present state. In turn, another set of outputs are stored as a result of these inputs changed and stored in memory as a result of the change in thresholds of the neurons. These inputs are called next state and are labeled, S_0‘, S_1’, and S_2’. A flip-flop function is needed to store these previous states and produce next states as demonstrated in Fig. 4. Flip-flops are explained in a later segment.

C. Boolean Algebra

Boolean algebra expressions help to design a digital logic circuit. The first step that is performed is to refer to Table II. Next, observe the next state column of S_0’ each expression is derived where:

$S_0' = 1$ When $S_0' = \overline{D}\overline{S_0}S_1\overline{S_2}$

The same steps can be applied to S_1’:

$$S_1' = \overline{C}\overline{S_0}\overline{S_1}S_2 + C\overline{S_0}\overline{S_1}S_2 + D\overline{S_0}S_1\overline{S_2} + \overline{S_0}S_1S_2 + ES_0\overline{S_1}\overline{S_2}$$

This expression can be reduced using theorems:

$$S_1' = (C + \overline{C})\overline{S_0}\overline{S_1}S_2 + \overline{S_0}S_1(D\overline{S_2} + S_2) + ES_0\overline{S_1}\overline{S_2}$$

Where $\left(C + \overline{C}\right) = 1$ [2] and

$$S_0S_1(DS_2 + S_2) = S_0S_1D\,[2]$$

$$\overline{S_0}\overline{S_1}S_2 + \overline{S_0}S_1S_2 = \overline{S_0}S_2$$

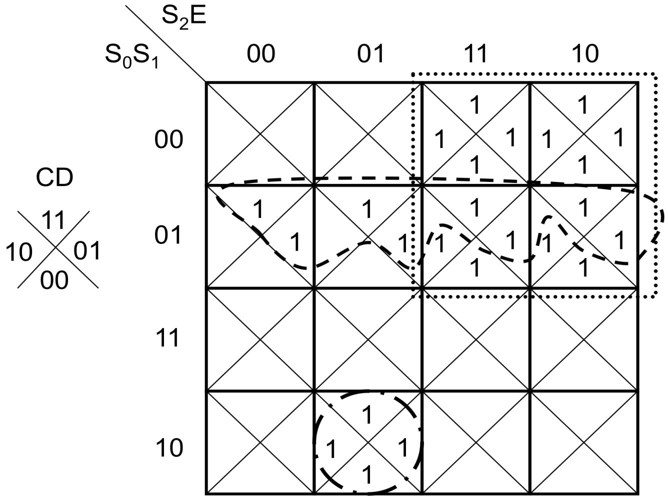
Therefore:

$$S_1' = \overline{S_0}S_2 + \overline{S_0}S_1D + ES_0\overline{S_1}\overline{S_2}$$

Also, the same expression can be reduced with a Karnaugh map (K-map) as shown in Table III.

Once the truth table of Table II is obtained its important to start entering results into a Karnaugh Map (K-Map) as shown in Table III.

TABLE III
KARNAUGH MAP (K-MAP)


S$_2$E
S$_0$S$_1$
CD
11
10
01
00

S$_0$S$_1$ \ S$_2$E	00	01	11	10
00			1 1 1 1	1 1 1 1
01	1 1	1 1	1 1 1 1	1 1 1 1
11				
10		1 1 1 1		

Using the sum of products (SOP) grouping method allows three grouped terms thus deriving the same Boolean algebra expression of:

$$S_1' = \overline{S_0}S_2 + \overline{S_0}S_1D + ES_0\overline{S_1}\,\overline{S_2}$$

Finally, the theorem or K-map steps can be applied to *S$_2$'* in order derive a Boolean algebra expression:

$S_2' = 1$ When:

$$S_2' = AB\overline{S_0}\,\overline{S_1}\,\overline{S_2} + C\overline{S_0}\,\overline{S_1}S_2 + D\overline{S_0}S_1\overline{S_2} + ES_0\overline{S_1}\,\overline{S_2} + \overline{S_0}S_1S_2$$

The inputs, previous states, and next states from Table I and Table II are finally expressed in logic form. These expressions allow for the logic gate circuit design.

D. Digital Logic Circuits

The reduced Boolean algebra expressions from the previous section helps to derive the following circuits as shown in Fig. 3, and Fig. 5:

$S_0' = 1$ when $S_0' = \overline{D}\,\overline{S_0}S_1\overline{S_2}$ as shown in Fig. 4.

$S_1' = \overline{S_0}\,\overline{S_1}S_2 + \overline{S_0}S_1\overline{S_2} + ES_0\overline{S_1}\,\overline{S_2}$ as shown in Fig. 5.

$$S_2' = AB\overline{S_0}\,\overline{S_1}\,\overline{S_2} + C\overline{S_0}\,\overline{S_1}S_2 + D\overline{S_0}S_1\overline{S_2} + ES_0\overline{S_1}\,\overline{S_2} + \overline{S_0}S_1S_2$$

$$Y = \overline{S_0}S_1S_2$$

$$S_2' = AB\overline{S_0}\,\overline{S_1}\,\overline{S_2} + C\overline{S_0}\,\overline{S_1}S_2 + DS_0S_1S_2 + ES_0\overline{S_1}\,\overline{S_2} + S_1S_2(DS_0 + S_0)$$

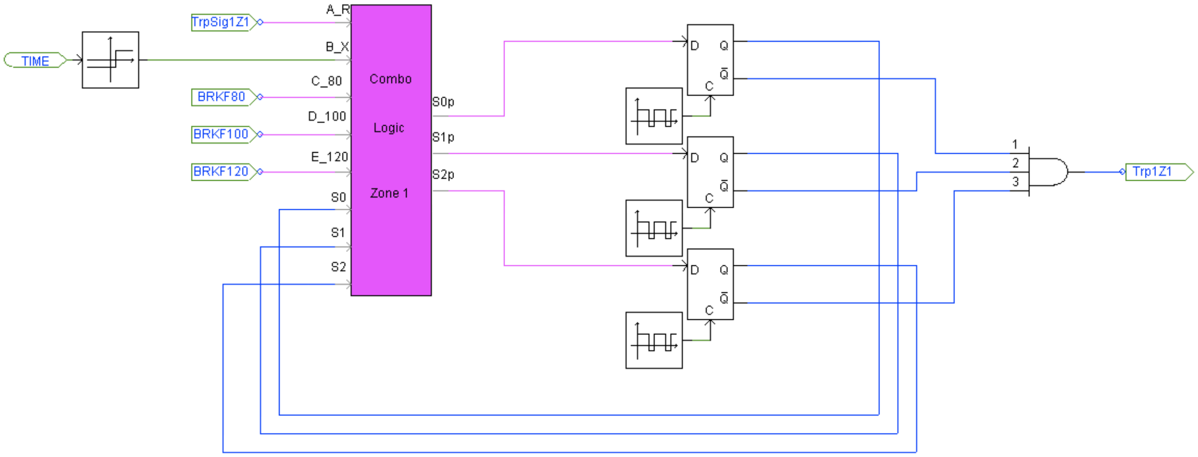


Fig. 4. Integrated digital logic circuit derived from Table II with flip-flops.

III. RESULTS

The digital circuit is now obtained and flip-flops are what help the circuit store and use previous state values as demonstrated in Fig. 4. Moreover, the flip-flops use the clock as a threshold for a neuron simulated digital logic network. The flip-flops produce next state values that help to locate the faults on the electrical distribution system. The simulated neuron trains itself as it learns the SLG fault locating patterns at 120%, 100%, and 80% of the line on the shipboard [3] electrical distribution system as shown in Figs. 5 and 6.

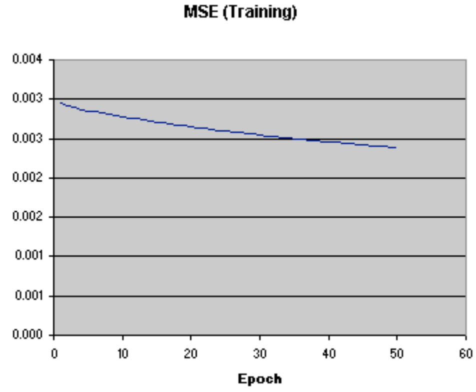
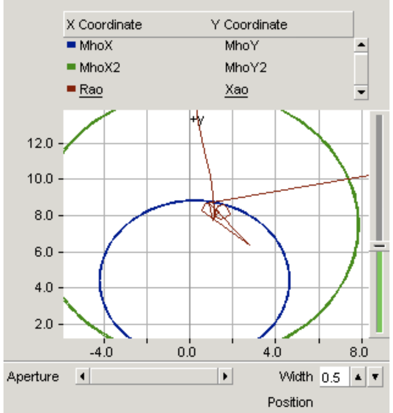


Fig. 5. Neuron is trained as it locates faults on line.



X Coordinate	Y Coordinate
MhoX	MhoY
MhoX2	MhoY2
Rao	Xao

Fig. 6. Mho zone protection as a function of EBP relay learns and locates fault.

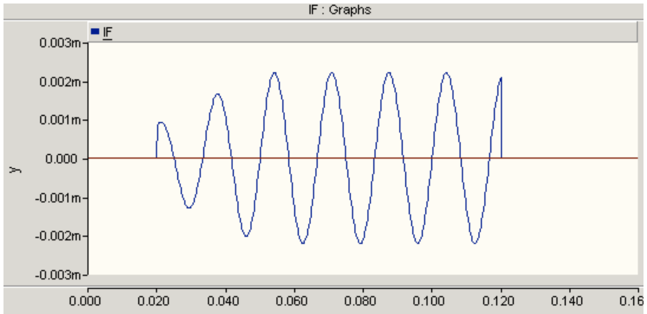


Fig. 7. Fault clears after 0.12s.

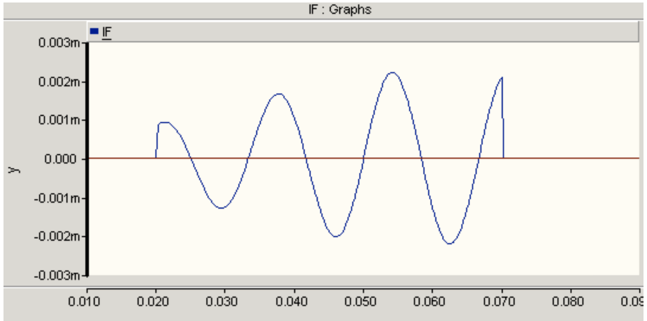


Fig. 8. Fault clears after 0.07s.

As the neuron learns the pattern the error back propagation (EBP) algorithm may improve single-line-to-ground (SLG) fault clearing times [4]. Theoretically, as the neurons are trained in the over 0.50s, the fault current clearing time improve from 0.12s to 0.07s each time the simulation runs as shown in Fig. 7 and 8. This enables the ship to isolate faults that may occur from random insertions at 88%, 101%, or 117% of the line while decreasing SLG fault clearing times each time a SLG fault is encountered.

IV. Conclusion

It is advantageous to consider using this error back propagation (EBP)-based relay. The EBP-based [5] relay successfully locates the single-line-to-ground (SLG) faults at 120%, 100%, and 80% of the shipboard power system line. The neurons learn the patterns of the SLG fault occurrences per simulation and fault clearing times improve as depicted in Fig. 7 and Fig. 8. For future works injecting SLG faults at random locations on the shipboard power system line will simulate adverse battle conditions.

V. Acknowledgment

The authors gratefully acknowledge the contributions of R. Wells, G. Donohoe, D. Billin, and H. Hess.

VI. References

[1] E.J. William, B. Johnson, M. Manic "ANN relay Protection for Shipboard Electrical Distribution Systems", 2007 North American Power Symposium Proceedings (NAPS), pp 143-147.

[2] C. Roth, *Fundamentals of Logic Design*, Boston: PWS, Fourth Edition 1995, p.33.

[3] Somani, B.K. Johnson, and H.L. Hess, "Evaluation of Grounding and Protection Methods for a Shipboard Power System," 2005 IEEE Electric Ship Technologies Symposium. pp 117-124, July 25-27, 2005.

[4] H. Khorashadi-Zadeh, M. R. Aghaebrahimi, "A Novel Approach to Fault Classification and Fault Location for Medium Voltage Cables Based on Artificial Neural Network" in *International Journal of Computational Intellegence*vol 2 no 2, 2005 pp 90-93.

[5] A. Apostolov, J Bronfeild, C.H. Saylor "An Artificial Neural Network approach to the detection of high impedance faults". *EPRI conference on Artificial Intellegence Applications in Power Systems*, Dallas TX, December 1992, pp 70-74.

VII. Biographies

Edward James William II, IEEE Student Member was born in San Diego, California, USA. He received his B.S in Electrical Engineering from Arizona State University and M.S. from the University of Idaho. He is currently a Doctoral student attending the University of Idaho. His employment experience included the Boeing Company, Honeywell Inc. His special fields of interest included Power System Protection. He received 1st place prize for the 2005 Minority Engineering Program Technical Paper competition for his research in Ion Channel Bio-sensors.

Dr. Brian K Johnson, (M92--SM07) received the Ph.D. degree in electrical engineering from the University of Wisconsin, Madison, in 1992. He is currently a Professor in the Department of Electrical and Computer Engineering, University of Idaho, Moscow. His interests include HVDC transmission, FACTS, custom power technologies, energy storage, and utility applications of superconductivity, power system protection, and electromagnetic transients in power systems. Dr. Johnson is a member of CIGRE and is a Professional Engineer in the States of Wisconsin and Idaho.

Dr. Milos Manic, IEEE Senior Member, received his Ph.D. degree in Computer Science from University of Idaho, Computer Science Dept. He received his M.S. and a Dipl.Ing. in Computer Science and Electrical Engineering from the University of Nis, Faculty of Electronic Engineering, Serbia. He is an assistant professor at the UI Computer Science Dept. and program director for Idaho Falls CS program. Dr. Manic serves as a chair of IES TCFA subcommittee, IES assoc. web editor, and on technical committees of various IEEE conferences. His areas of research include computational intelligence, intelligent control, modern heuristics, and performance based modeling of fault tolerant systems.