Design Tool Generates Substation Interlock Schemes

Mohindar S. Sachdev, Pramod Dhakal,* Tarlochan S. Sidhu**

Balancing load and generation and having adequate transmission capacity in service is essential for maintaining continuity of power supply to customers. This is achieved by reconfiguring the transmission and distribution systems by opening and closing circuit breakers and isolators for connecting and disconnecting lines, generators, and loads. These actions redistribute the current flows and change the voltage profile in the system. An improper switching operation can damage equipment and/or can affect the quality of supply to the customers.

Normally, switching operations are performed either by operators or by substation controllers. The operators use predefined guidelines that include line loading limits and interlocking of switches. The use of computerized switching schemes has become essential for avoiding human errors and expediting control actions. A major handicap in designing interlocking logic is that each scheme is designed specifically for the substation in which it is to be implemented.

The introduction of object-oriented languages combined with the Windows environment has provided tools for developing graphical user interfaces (GUI) that allow intuitive and visual communication with the application programs. Availability of advanced features, such as inheritance, that are not available in traditional computer languages has allowed the development of applications that were tedious to develop using traditional programming approaches. The power of object-oriented programming in a Windows environment has been used to design a software application that automatically generates switching schemes that satisfy specified interlocking criteria.

The application reduces the time required to write the interlocking logic and provides control for modifying the logic to meet special needs

The application has two components, one for generating interlocking schemes from substation data and the other for generating switching instructions and implementing them. The connectivity of substations, parameters of switches and components, and interlocking constraints are used by the software as input data. While this generalized tool expedites the development of interlocking schemes, its use is not limited to specific substation configurations.

* University of Saskatchewan

ISSN 0895-0156/00/$10.00©2000 IEEE

Three out of the many designs prepared are presented in this paper to demonstrate its usefulness.

System Overview

The software is made up of a graphical interface for interacting with the users, application software, and a database. The graphical interface provides a facility for drawing one-line diagrams of substations and for entering substation data and interlock constraints with the help of dialog boxes. The data and logic entries are validated as they are entered, and the validated information is then transferred to the application software.

The application software generates the interlocking scheme without the user having to write the detailed logic. The user, however, must understand the operation of the substation and identify the operating constraints of each component, one component at a time.

The generated logic of the interlocking scheme is stored for future use in a permanent memory such as a hard drive. When the stored interlocking scheme is retrieved, it is interpreted and the substation diagram is drawn on the monitor. The status of all switches (and other components) is displayed in this diagram. The graphical environment also includes facilities for testing the interlocking scheme and confirming that the logic satisfies the constraints and the switching operations are viable.

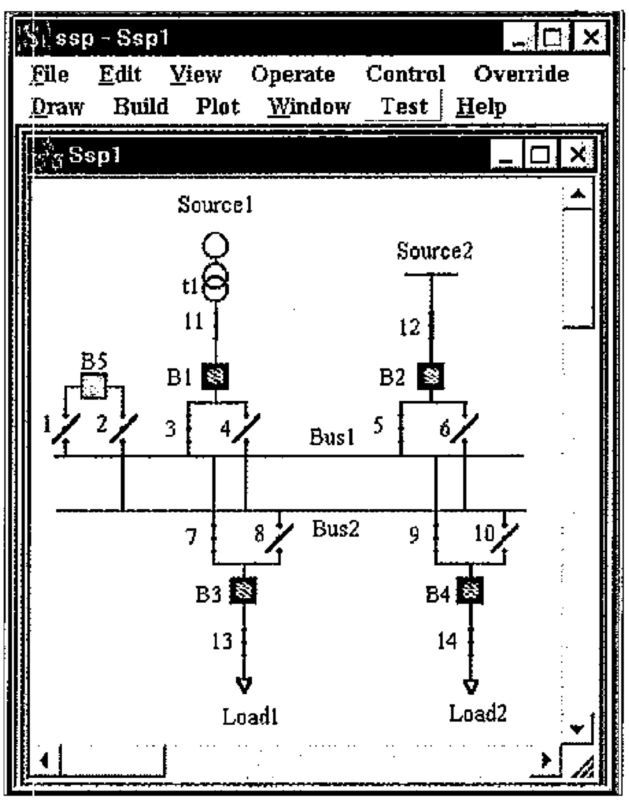

Figure 1. Part of the one-line diagram of a double bus-bar substation

Graphical User Interface

The GUI helps in preparing the information, keeping it organized, and presenting it in a user-friendly manner. The circuit diagram of a substation is drawn by dragging symbols of components from a toolbox and releasing them at the appropriate coordinates of the screen of the monitor. When a component is brought to the screen, a dialog box is automatically displayed. At other times, the dialog boxes can be activated by selecting appropriate commands from the menu. The specifications of the components are entered via dialog boxes. The connections can be either drawn on the screen or entered through dialog boxes. The software automatically creates appropriate representations of the components. Figure 1 shows a part of the one-line diagram of a double bus substation prepared in this manner. An operator can interact with the one-line diagram by using a pointing device, such as an optic pointer or a mouse, for communicating with the software.

Application Software

The information received by the application software is kept in lists and arrays of appropriate data structures. The switches are sorted according to their types and kept in an array. All other components, such as nodes, transformers, and lines, are collected in another array and are sorted. An identification number is assigned to each circuit element.

A significant proportion of computations for checking interlocks is common irrespective of the switching operations to be performed. To avoid processing the same information repeatedly, the data of a substation is processed, and intermediate results are stored for future use. When an interlock check is to be performed, processing starts from the intermediate results. This increases the efficiency of the application and expedites the decision-making process.

Major Functional Blocks

The application software consists of eight major functional blocks, as is shown in Figure 2. The task manager controls the operation of the interlock-design software. The path tracer determines the energy-flow-paths to ensure that power flow is not interrupted by switching operations unless it is so intended. The system-level interlocks detector inhibits the operations if they adversely affect the integrity of the system. The topology detector keeps up-to-date the information on the topology of the substation.

The task manager, path tracer, and topology detector use information generated by the remaining five modules. The primary circuit element data collector receives the data from the user interface and data files, creates software models of the components, and organizes the objects in sorted lists and arrays. The branch and node extractor creates objects representing branches and nodes of the substation circuit. The component interlock detector compiles the interlock code of each object and attaches it to the object. The network graph generator prepares a mathematical representation of the substation configuration. The branch interlock detector monitors the interlocks within a branch and between different branches of the circuit.

Interlock Constraints

The constraints for the operation of switches and sequence of operations were determined from the properties of switches and ratings of the equipment. Some of the constraints used in developing interlocks are as follows.

- Protection systems must be allowed to open circuit breakers for clearing faults.
- Only one switching operation should be performed at a time when a sequence of operations is to be performed.
- The most recent status of all switches must be available.
- An isolator must not make or break load or fault current, but its operation should be permitted if a shunt path (across it) has been established.
- An isolator adjoining a circuit breaker should be operated only if the circuit breaker is open.
- A load-break switch must not be operated to interrupt fault current.
- A circuit breaker must not be opened if it interrupts the flow of power to a circuit unless it is so intended.
- A circuit breaker must not be closed if an isolator, in the circuit it controls, is in motion.
- A live circuit must not be grounded.
- The current and voltage ratings of lines, transformers and other equipment should not be exceeded.
- For parallel lines, the opening of a circuit breaker must not cause overloading of the circuits remaining in operation.

Additional constraints were developed for bus-coupler circuit breakers, bus sectionalizers, and bus configurations. These constraints work at the object level such as isolator, circuit breaker, busbar, line, load, and grounding switch. When two live circuits are to be connected, synchrocheck must confirm that the systems are synchronized. If the test shows that the two systems are not in synchronism, the procedure of synchronization is included in the switching operations.

Implementation

The application implements its major functional blocks by software modules organized as object-oriented programs written in C++ for the Microsoft Windows environment. Some of the major modules out of the many that form the software application are as follows.

Constraints for the operation of switches and sequence of operations are determined from the properties of switches and ratings of the equipment

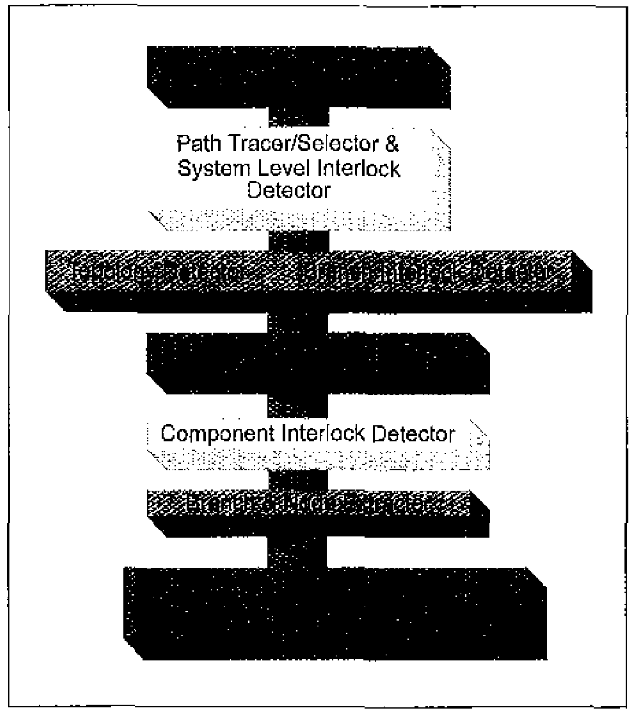

Figure 2. Major functional blocks of the interlock-design software application

- Interlock management follows a protocol for accepting, storing, and processing information on all objects. The text-based logic statements of interlock constraints are converted to machine code and vice versa. The status of components is updated.
- Status management continuously monitors the parameters of electrical components that change from time to time. It stores, updates, and compares status information and communicates it to various components. It also evaluates the states of all components at the time of making switching decisions.
- Topology determination finds which circuit elements are connected and how they are connected. It describes the interconnection of lines, busbars, generators, transformers, and loads.

- State estimation evaluates the state of the power system that would result from the execution of a switching operation. The present state of a system is known through actual measurements, and the future state is estimated by conducting a load-flow study.
- Path-tracer determines alternative power flow paths between two locations of a substation. The module considers the operating constraints of the components and the viability of the path. When multiple paths are available, the module selects the one with minimum switching operations. The path tracer excludes the components that are locked, stuck, unhealthy, or are under repair.

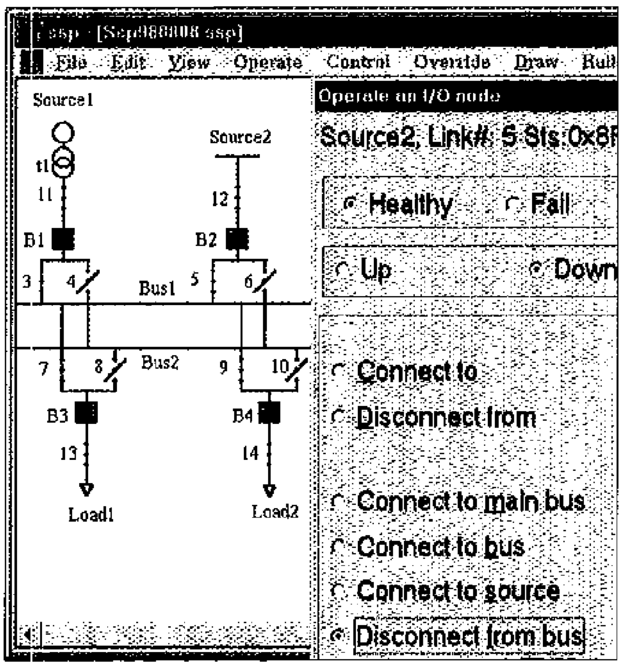

Figure 3. One-line diagram of a substation and the dialog box for issuing the "Disconnect from bus" command

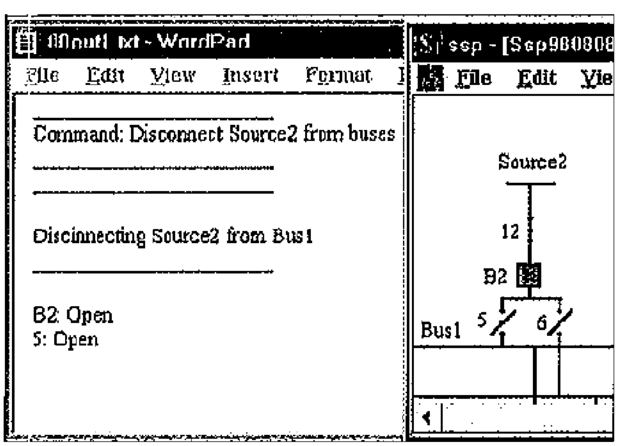

Figure 4. Sequence of operations selected by the application and the status of switches after implementing the command of Figure 3

Examples

This interlock-design application was applied to several configurations of substations. It produced functional interlocking schemes that worked correctly. Three cases are presented in this section. The first case disconnects source 2 from bus 1 in the substation shown in Figure 1. The second case takes out of service a circuit breaker of a transfer busbar substation. The third case transfers all circuits from one busbar of a double-bus substation to another busbar.

Case 1: Disconnect Source from Bus

A part of a double-bus substation used in case 1 and the dialog box that is displayed when source 2 is double-clicked are shown in Figure 3. Since the objective in this case was to disconnect source 2 from bus 1, the button "Disconnect from bus" was selected. The "OK" button, which is at the bottom of the dialog box, was then selected. The interlocking software application prepared the switching sequence and executed the operations it had identified. The sequence of switching operations determined by the application and the status of the substation after the operations were performed is shown in Figure 4.

Case 2: Circuit Breaker Maintenance

The objective of the second case was to test the performance of the interlocking software application applied to a transfer busbar substation. A part of the one-line diagram of the substation is shown in Figure 5. The operation selected in this case was to take circuit breaker B3 out of service for maintenance. Double-clicking on the circuit breaker brought up a dialog box on the monitor. The "Isolate" box was checked, and the "OK" button was clicked. The application prepared the switching operations and performed them. A part of the dialog box and the final status of the substation are shown in Figure 6.

Case 3: Transfer Circuits to Auxiliary

The objective in the third case was to test the ability of the application for transferring all the circuits of the main busbar of a double-bus substation to the auxiliary busbar. A part of the selected substation is shown in Figure 7. Double-clicking on "Bus1" brought up a dialog box on the screen. Three items were selected in this dialog box: the "Transfer bus to" command, then "Bus2" (the bus to which all circuits would be transferred), and finally the "OK" button.

The application prepared the switching sequence and executed the operations contained in the sequence. The final status of the substation after executing the determined switching operations is shown in Figure 8. The operations carried in this sequence include the closing of the bus-coupler-B5 and the adjoining isolators to connect bus 2 with bus 1. This also connects sources 1 and 2 and loads 1 and 2 to bus 2 through the bus-coupler circuit. Bus 2 is then connected to bus 3 via bus-coupler-B10 circuit and the bus-sectionalizing isolator 16. Bus 1 is then isolated from the system. These operations isolated bus 1 from the system without interrupting the power supply to any of the circuits.

Several software modules participated in making the switching decisions. The following sequence indicates major functions performed by the modules:

1. Selection of the appropriate command and pressing of "OK" button closed the dialog box and transferred the command to the task manager module.
2. The task manager module invoked the path tracer/selector and system level interlock detector (PTSLID) to check the impact of removing bus1 from service.
3. The PTSLID performed the following tasks:
    a. With the help of the topology detector, it found that the connections between sources 1 and 2 and the loads 1 and 2 will be broken.
    b. It checked the generation data and found that sources 1, 2, 3, and 4 were supplying 150, 50, 80, and 70 MW, respectively.
    c. It checked the load data and found that loads 1, 2, 3, and 4 were receiving 59, 70, 175, and 48 MW, respectively.
4. The flow balance module determined that the generation and load do not balance if bus 1 is disconnected from bus 3. The module returned this information to the task manager module.
5. The task manager module divided the task in the following five subtasks.
    a. Connect bus 1 to bus 2 via the bus coupler.
    b. Connect lines connected to bus 1 to bus 2.
    c. Connect bus 3 with bus 2.
    d. Disconnect lines from bus 1.
    e. Disconnect bus 1 from bus 3 and bus 2.
6. The task manager module again invoked the PTSLID that invoked the topology detector and branch interlock modules.
7. Using the information received from these modules, the PTSLID determined the optimal switching sequences for the subtasks and passed them on to the task manager module.
8. The task manager module performed simulation of the switching operations.

Applications

There are two applications of this design tool. The first application is to design interlocking logic for incorporating in automatic substation control packages. This application substantially reduces the time required to write the interlocking logic. The tool also provides substantial control for modifying the logic to meet with the special needs of each application.

The software is also suitable for training operators working in control rooms of substations and generating stations and those working in dispatch centers

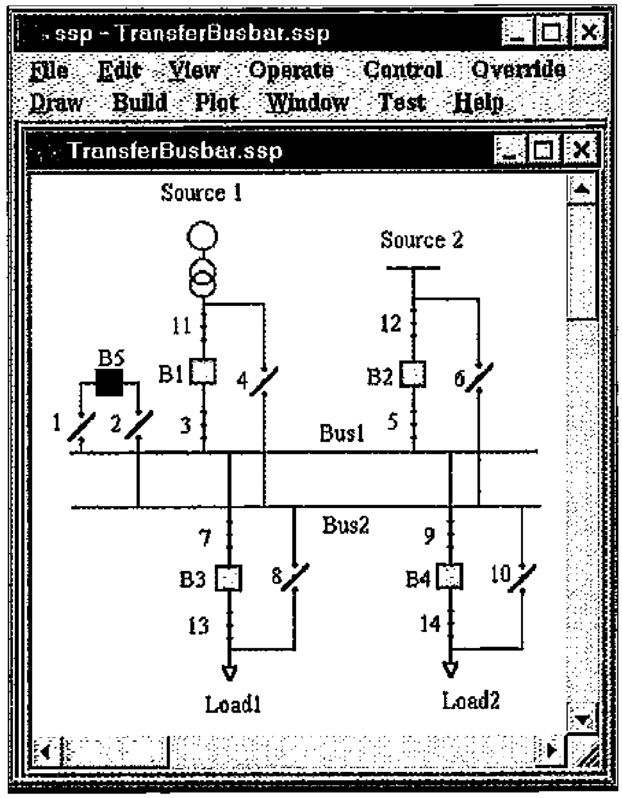

Figure 5. Substation with transfer bus configuration

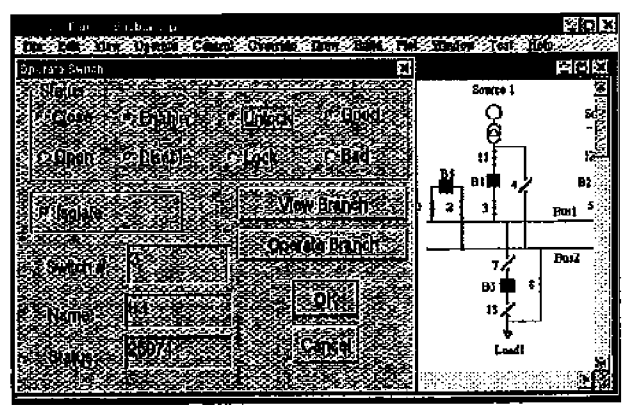

Figure 6. Dialog box for issuing the "Isolate" command and the status of the substation after the switching operations are performed

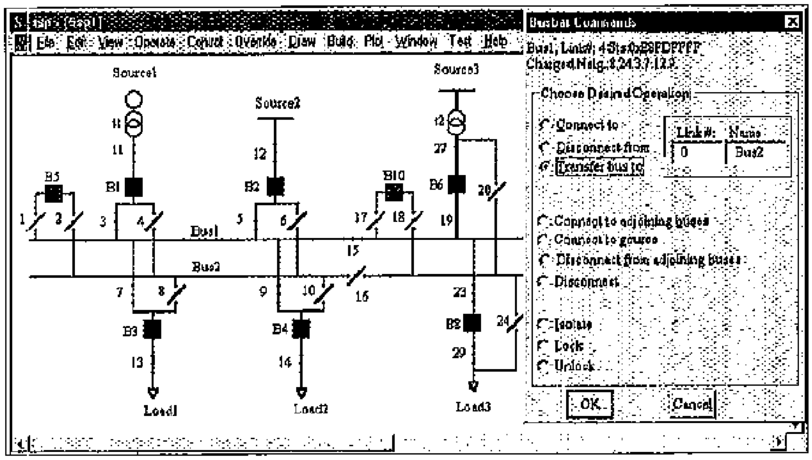

Figure 7. One-line diagram of a double-split busbar arrangement and dialog box showing the "Transfer bus to" command

The second application of this tool is operator training. The software produced by this tool incorporates facilities to draw one-line diagrams of substations and accepts switching tasks through a graphical interface. To perform the necessary task, the software generates the required switching sequences and implements the switching operation in the determined sequence. In real life, switches take finite time for opening and closing. The software application models these times, and the simulated operations mimic the operations that take place in utility substations. These features make the software suitable for training operators working in control rooms of substations and generating stations and those working in dispatch centers.

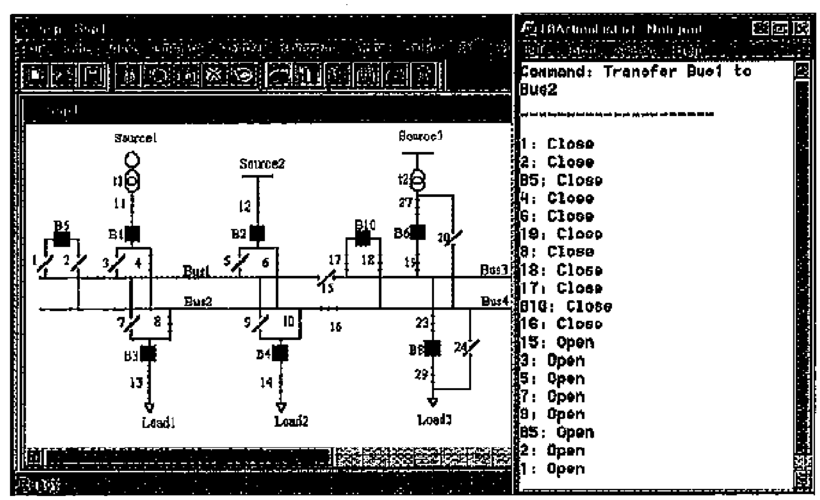

Figure 8. List of operations selected for isolating busbar 1 and the double-split busbar substation configuration after the selected operations are performed

For Further Reading

V.R.D. Hutchinson, "Interlocking in large electricity-supply substations: a fundamental approach," *Proceedings of the IEE*, Volume 130, Number 6, June 1966, pp. 1063-74.

K.P. Brand, J. Kopiansky, W. Wimmer, "Topology-based interlocking of electrical substations," *IEEE Trans. on Power Delivery*, Volume 1, Number 3, July 1986, pp. 118-126.

E.N. Dialynas, A.V. Machias, "Interactive modeling of substation switching operations following a failure event," *IEE Proceedings*, Volume 134, Part C, Number 2, March 1987.

G. Booch, *Object Oriented Analysis and Design with Applications*, second edition, Benjamin Cummings, Redwood City, USA, 1994.

N.N. Schulz, B.F. Wollenberg, "Expert system to aid evaluation and generation of switching procedures," *Proceedings of the 57th American Power Conference*, Part 2, pp. 1481-1486, 1985.

Biographies

Mohindar S. Sachdev received the B.Sc. from the Baneras Hindu University, India, the M.Sc. degrees from the Punjab University, Chandigarh, India and the University of Saskatchewan, Saskatoon, Canada, and the PhD and DSc degrees from the University of Saskatchewan, Saskatoon, Canada. He worked for the Punjab P.W.D. Electricity Branch and the Punjab State Electricity Board, India, from 1950 to 1968 in System Operation, Design and Planning. In 1968, he joined the University of Saskatchewan, where he is currently professor emeritus of electrical engineering. His areas of interest are power system analysis and power system protection. He is an IEEE Fellow, a Fellow of the Institution of Engineers (India), and a Fellow of the Institution of Electrical Engineers, London (England). He is also a registered professional engineer in Saskatchewan and a chartered engineer in the UK.

Pramod Dhakal obtained his MS degree from the University of Saskatchewan in 1994 and a bachelor of engineering degree with honors from the University of Roorkee in 1988. He is near completion of his PhD degree from the University of Saskatchewan, where he worked in the field of computer-aided design of substation switching systems. He worked for Nepal Electricity Authority as an operation engineer in 1988-1989. He then taught electrical engineering courses as a lecturer at the Institute of Engineering, Nepal, between 1989-91. He joined LGS Group Inc. as an analyst in 1998 and became a consultant in 1999. With the LGS group, he developed electronic commerce applications for large organizations. He recently joined Nortel Networks as a software engineer, and he is working in the development of a distributed operating system for ATM communication networks.

Tarlochan S. Sidhu received the B.E. (Hons.) degree from the Punjabi University, Patiala, India, in 1979 and MS and PhD degrees from the University of Saskatchewan, Saskatoon, Canada, in 1985 and 1989, respectively. He worked for the Regional Computer Center, Chandigarh, India, from 1979 to 1980 and developed software for computer-based systems. He also worked for the Punjab State Electricity Board, India, from 1980 to 1983, in distribution system operation and thermal generating station design. He joined Bell-Northern Research Ltd., Ottawa, Canada and worked on a software development project. In 1990, he joined the University of Saskatchewan, where he is presently professor and graduate chair in the Department of Electrical Engineering. He is a senior member of the IEEE and a member of the IEE (UK). He is also a registered professional engineer in Saskatchewan and a chartered engineer in the UK. His areas of research interest are power system protection, monitoring, and control.