

Programmability of Numerical Relays: A Busbar Protection Relay Serves as a Traditional RTU

Casper Labuschagne
Schweitzer Engineering Laboratories, Inc.

Izak van der Merwe
Eskom

Revised edition released November 2004

Originally presented at the
2004 Southern African Power System Protection Conference, November 2004

Programmability of Numerical Relays: A Busbar Protection Relay Serves as a Traditional RTU

Casper Labuschagne, *Schweitzer Engineering Laboratories, Inc.*, Pullman, WA, USA
Izak van der Merwe, *Eskom Resources and Strategy Division*, Cape Town, South Africa

Abstract -- Numerical relays have revolutionized protection, control, metering, and communication in power systems. Functional integration, new methods of communication, reduced physical size, and an enormous amount of available information are but a few of the benefits of this revolution.

Having made the initial conceptual adjustment of relating objects from electromechanical technology such as rotating discs and moving armatures to such electronic technology as analog-to-digital converters and comparators, protection practitioners then must deal with programming a relay. Initially, programming was no more than selecting values for relay settings. Further advancement in digital technology, however, has made possible advanced and sophisticated programming of logical functions and analog quantities.

A good understanding of relay programming is necessary to take full advantage of the many functions integrated into numerical relays and use these functions in different applications to enhance operation of a power network. Unfortunately many users avoid relay programming, considering it too complex. Because of this perceived complexity, not all users investigate the use of relay programming to realize automation and control applications. Many cost-saving opportunities and simple engineering solutions to automation applications are reliably achieved by using the protection relay programming features.

This paper focuses on the programmability of modern protection relays and how these modern relays can incorporate functions traditionally performed by a remote terminal unit (RTU), but with greater hardware reliability. The paper shows examples of programming tools available in a modern protection relay and discusses separation of protection and automation program areas in the same relay. We briefly discuss different communications topologies, such as star and multidrop, and we show the advantage of using a central device such as a bus protection relay instead of these communications topologies. As an example of protection relay programmability, we show how a busbar protection relay serves as a traditional RTU while still performing primary functions of busbar protection. As an example of programming protection functions, we show how to program a definite-time earth-fault element. This paper describes the key features of this unique project.

I. PROGRAMMING TOOLS

Before investigating cost-saving programming opportunities, we need to first look at the available programming tools. Modern microprocessor relays offer, at a minimum, the following programming tools:

- Control equation elements
- Boolean operators
- Math operators
- Binary elements (Word bits)
- Analog quantities

II. CONTROL EQUATION ELEMENTS

Control equation elements are a collection of storage locations (both volatile and nonvolatile), timers, and counters that you use to customize relay operation and use the relay to automate substation operation. Table 1 summarizes the elements available for use in control equations.

TABLE 1
ELEMENTS AVAILABLE FOR USE IN CONTROL EQUATIONS

Element	Description
Control equation variables	Storage locations for the results of Boolean control equations
Control equation math variables	Storage locations for the results of math control equations
Latches	Nonvolatile storage for the results of Boolean control equations
Conditioning timers	Pickup and dropout style timers
Sequencing timers	On-delay timers as used in programmable logic controllers
Remote bits	Storage locations for status communicated from other devices
Counters	Counters that count rising edges of Boolean value inputs

III. BOOLEAN OPERATORS

Use Boolean operators to combine values with a resulting Boolean value. The arguments of the operator may be either numbers or Boolean values, but the result of the operation must be a Boolean value. Combine the operators to form statements that evaluate Boolean logic. Table 2 contains a summary of typical Boolean operators available in the relay.

TABLE 2
BOOLEAN OPERATOR SUMMARY

Operator	Description
OR	Logical OR
AND	Logical AND
NOT	Logical inverse
()	Parentheses
R TRIG	Rising-edge trigger
F TRIG	Falling-edge trigger
>, <, =, <=, >=, <>	Comparison of values

IV. MATH OPERATORS

Use math operators when writing mathematical control equations involving analog values. Math control equations manipulate numerical values. Table 3 summarizes the operators available for math control equations.

TABLE 3
OPERATORS AVAILABLE FOR MATH CONTROL EQUATIONS

Operator	Description
()	Parentheses
+, -, *, /	Arithmetic
SQRT	Square root
LN, EXP, LOG	Natural logarithm, exponentiation of e, base 10 logarithm
COS, SIN, ACOS, ASIN	Cosine, sine, arc cosine, arc sine
ABS	Absolute value
CEIL	Rounds to the nearest integer towards infinity
FLOOR	Rounds to the nearest integer towards minus infinity
-	Negation

V. BINARY ELEMENTS (WORD BITS)

Binary elements are the results of relay logic, the logic being either pre-programmed or programmable. Fig. 1 shows a logic diagram consisting of an AND gate, an OR gate, a timer, two inputs, and an output. The Initiate input could be either pre-programmed or programmable. When the input is programmable, you define other binary element(s) as an input to the logic. The output of the logic is a binary element that has a value of either 1 or 0. When used in pre-programmed logic, binary elements have familiar labels such as TRIP or CLOSE. General control equation elements have labels that describe their functions. For example, the output (Q) of Protection Conditioning Timer 10 has the label PCT10Q. With modern relays you can define aliases for functional names to make logic interpretation more intuitive.

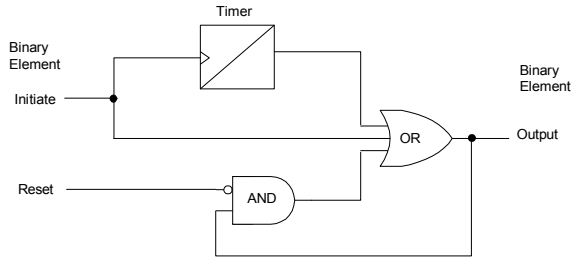


Fig. 1. Example of Relay Logic

VI. ANALOG QUANTITIES

Analog quantities are the received, measured, and calculated current and voltage quantities. Normally, analog values are available at various points in the processing sequence. For example, Table 4 shows the instantaneous as well as the average values of the current and voltage quantities. In this paper, angle I01FA is the one-cycle current angle of the A-phase current of Terminal I01 with reference to the A-phase voltage (V01FM), i.e. I01FA is the phase angle of Terminal I01 with respect to Voltage V01. We use math operators to program the analog quantities.

TABLE 4
EXAMPLE OF SOME AVAILABLE ANALOG QUANTITIES

Label	Description	Units
InnFIM	Phase filtered instantaneous current magnitude	A (secondary)
InnFIA	Phase filtered instantaneous current angle	degrees
InnFM	Phase one-cycle average current magnitude	A (primary)
InnFA	Phase one-cycle average current angle	degrees
VmmFM	Phase one-cycle average voltage magnitude	kV (primary)
VmmFa	Phase one-cycle average voltage angle	degrees

VII. SEPARATION OF PROTECTION AND AUTOMATION PROGRAMMING

In general electric utility practice, protection and automation (control) are typically two separate disciplines. Normally, each discipline uses its own, separate devices for protection and automation. The relay we discuss in this paper continues this practice by providing separate storage (programming) areas for protection and automation. With this segregation, automation programming is completely separate from protection programming, and vice versa. Protection programming need not affect automation programming. However, protection and automation areas are available in the same device and can interact and exchange information as necessary.

Fig. 2 illustrates how the relay separates protection and automation programming while still exchanging information between the two areas. The arrows indicate data flow between components. The binary elements and analog quantities are available for protection, automation, and output programming. Protection programming uses the binary elements, analog quantities, protection variables, and automation variables as inputs, but protection programming only writes and stores information to protection variables. Similarly, automation programming uses data from all parts of the relay, but automation programming only stores data in the automation variables [1].

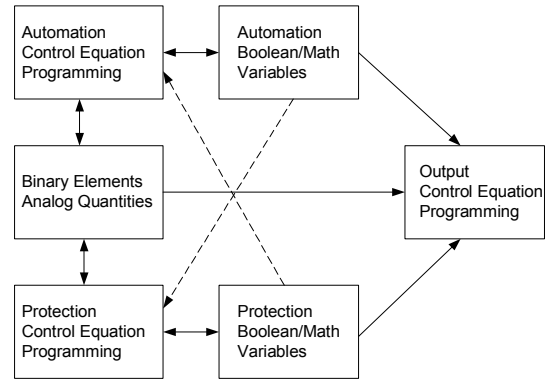


Fig. 2. Protection and Automation Separation

The relay provides both fixed result programming capability and free-form programming capability. Fixed result settings are control equations with the left side of the control equation pre-defined (fixed). For example, the Reset input in Fig. 1 is pre-defined. With fixed result settings, you cannot change the

left side of the control equation; you can only enter a setting on the right side of the control equation. When using free-form programming, you can define both the left and right side of the control equation to create customized automation and protection.

Fig. 3(a) and Fig. 3(b) contrast the two programming forms.

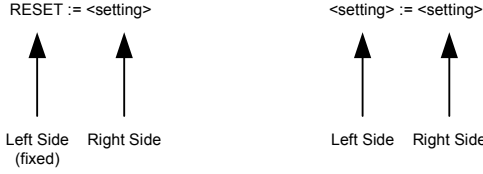


Fig. 3. (a) Fixed Settings

(b) Free-Form Settings

The relay provides fixed result and free-form programming capability for three different instances. The first instance is for protection storage and consists of two components: a storage area for protection control equations that use fixed result settings, and a separate storage area for protection control equations that use free-form programming. The second instance is for automation storage that is entirely free form. The third instance is for output control equations with which you configure the output contacts of the relay through use of fixed result control equations [2].

VIII. PROTECTION AND AUTOMATION FREE-FORM TIMING AND MEMORY ISSUES

Once you have programmed the relay, you must use some of the available execution time, execution storage, and settings storage space available in the relay to execute control equations. Because the primary function of the relay is protection, the majority of resources in a protection relay are dedicated to protection. However, a large subset of execution time, execution storage, and settings storage space is available for customized automation programming. Execution time is the time between successive processing of control equations.

Protection and automation solutions require both real-time and near real-time operations. Two separate regions of isolated settings are created and referred to as protection settings and automation settings. For near real-time automation control equations, the execution time is not absolute time. This time is, instead, deterministic in that the relay will process the control equation within one second. For real-time protection control equations, the execution time is the same as that for the protection elements. Because execution time is not absolute for automation programming, execution storage and settings space (rather than execution time) is an issue. Conversely, execution time, rather than execution storage and settings space, is an issue for protection programming.

Because of the different time and storage requirements, the relay calculates the available programming capacity for the automation and protection control equations separately. To provide you with information regarding available execution storage and setting storage space for automation programming, the relay reports the value of execution storage as execution availability. To provide you with information regarding avail-

able protection processing time, the relay reports the value of execution time availability as execution availability.

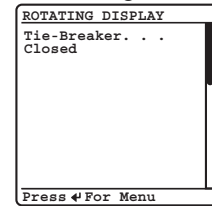
IX. CUSTOMIZED LOCAL USER INTERFACE

One major shortcoming of many RTUs is the lack of a local user interface. The user cannot program or monitor the RTU without a separate device, usually a computer. RTUs are normally installed without this computer and the only way to locally monitor, control, or diagnose the RTU is with a portable computer or laptop.

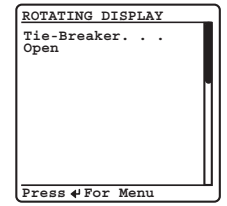
In sharp contrast to this RTU shortcoming, where all data are invisible inside the RTU, is the innovative front panel of modern protective relays. Analog and status values scroll across the display screen and present operators with the values of control permissives, lockouts, and alarms, and replace panel meters and paper operational tags. In addition, the front-panel also displays settings values and protection indications (see Fig. 4). You can program single-function pushbuttons to perform automation logic, bay control, and other commanded control as well as make situational changes such as set tags and lockouts.

X. DISPLAY POINTS

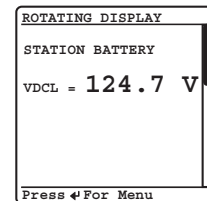
Display points are customizable text and numeric strings that are programmed into the relay to display descriptive information to users. Display points are messages that appear on the relay front-panel display when selected binary elements assert based on the argument of programmable control equations. For example, assume a substation where a feeder can be closed only when the tie breaker is open. Wire a 52A auxiliary circuit breaker contact to a relay input. When the tie breaker is closed, the input asserts, and the relay displays the message shown in Fig. 4(a) on the front-panel display. When the tie breaker is open, the input deasserts, and the relay displays the message shown in Fig. 4(b) on the front-panel display. Fig. 4(c) shows the battery voltage at the station on the front-panel rotating display; other information available for display includes information such as the current in each terminal and the busbar voltage.



(a) Tie Breaker Is Closed



(b) Tie Breaker Is Open



(c) Battery Voltage

Fig. 4. Example Display Points

XI. SEQUENTIAL EVENTS RECORDER (SER)

Sequential Events Recorder (SER) applications monitor input contacts and elements of logical and arithmetic equations to timestamp their Change of State (COS) and then create and queue an SER message for each occurrence detected. Configure the SER to monitor status, health, and performance of the power system as well as monitor communications access and security. The relay monitors COS of up to 250 calculated, instrumented and output status as well as settings changes, power ups, and binary elements selected by the user. Customized settings allow the user to create easily understood aliases using text and numerals. These SER messages are stored locally and sent to appropriate destinations for alarming, monitoring, analysis, and diagnostics. The SER application also provides a method of testing the accuracy and sequence of customized real-time and near real-time equations.

XII. COMMUNICATION

Most substation automation designs rely on a master to share data among intelligent electronic devices (IEDs). The master collects information from the controllers and other IEDs into one large database. The master can then send data from one IED to another. Should this master connection fail, the IEDs become stranded and do not work in a coordinated manner. Often, the master providing this data exchange is an otherwise occupied SCADA (supervisory control and data acquisition) master. Use of such a master for substation automation designs further dilutes the performance of the master and could reduce reliability. The reliability of this remote host-oriented substation automation (SA) system is reduced drastically by the possibility of host failure or failure of a host connection [3].

Direct connect and multidrop are two types of data link connections to protective relays. In a multidrop connection (Fig. 5), several devices can be connected physically in a bus network, and control of the transmit and receive conductors must be negotiated. For a multidrop connection, only one relay can communicate at a time. Software and hardware determine which device has permission to transmit, preventing data collisions on the conductor. Because several devices are connected, addressing is necessary within the protocol to identify the source and destination of data being communicated. This addressing adds overhead in the form of processing time and amount of information that needs to be transmitted, thus reducing the data that can be transferred at a given speed. Devices compensate for this reduction in data by increasing the speed at which they communicate and by increasing the amount of communications processing they perform [3].

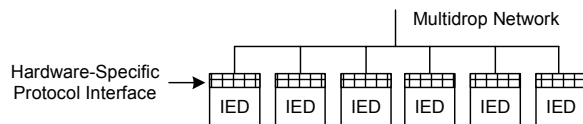


Fig. 5. Multidrop Network IED Interface

In a direct connection, only two devices connect via a transmit and receive pair of conductors. Use each conductor to transmit from one device and receive by the other device. Because there are only two devices, each device constantly controls the conductor on which the devices transmit and both know implicitly to which other device they are connected. Direct connections to many relays allow each of them to communicate simultaneously. In a star topology, such as that in Fig. 6, many direct connections originate from one device. Many star networks can be connected in a parallel or vertical hierarchy. Any protocol can be used in this configuration. Virtually all microprocessor-based relays have a simple EIA-232 serial port connection to support direct connections. Any of the other communications methods can also be used in a direct connection [3].

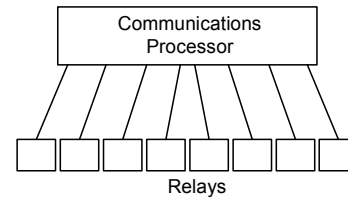


Fig. 6. Star Topology

With a multidrop network such as shown in Fig. 5, we can connect many IEDs to a single RS-485 port on an RTU; we are not limited by the number of available RTU serial ports. A drawback of the multidrop network is that most RTUs are forced to reduce their communications speed (at the same baud rate) each time a new IED is added to the multidrop. Another drawback to this topology is that you need an additional RS-485 bus; therefore, each IED must have an additional serial port, should you want engineering access from one central point. This hardware addition increases wiring and creates another point of failure in the system. Each IED connected to the RS-485 bus needs configuring to prepare the analog and binary data in the device for RTU collection. You must also configure the RTU itself to forward the data collected from the IEDs on to the SCADA master.

In contrast with multidrop networks, star topologies need only one serial connection between the communications processor and each IED, thereby reducing wiring between IED and RTU. With this point-to-point connection, an interruption of one communications channel affects only one IED, whereas all IEDs beyond the break are affected in a multidrop topology. As with multidrop, you must configure each IED as well as the communications processor to allow control and data collection from a central point.

We can eliminate many of these communications complications by using a single device to perform most RTU or communications processor functions.

XIII. RTU APPLICATION

One cost-saving opportunity exists with the traditional RTU. Fig. 7 shows typical RTU data concentration and control topology. Each bay control unit (BCU) receives current transformer (CT), voltage transformer (VT), and breaker status

information, and connects to the RTU to communicate this information and perform trip and close control functions. Either a BCU or RTU calculates real power (P), reactive power (Q), apparent power (S), and power factor (pf) from the current and voltage inputs.

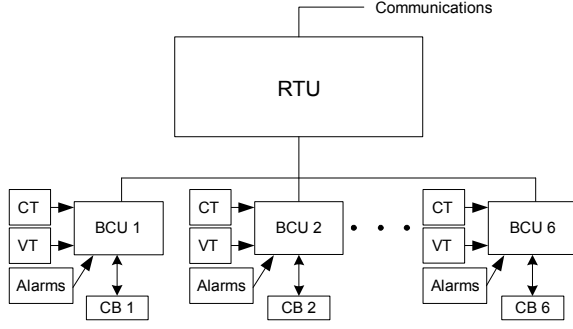


Fig. 7. RTU Topology in a Substation

XIV. SELECTION OF APPROPRIATE DEVICE

Because an RTU collects data from each substation IED, a replacement device must, at a minimum, connect each IED to a central unit. One such device is the busbar protection relay. To provide busbar protection (low impedance) for a substation, we must wire CTs from each terminal to the busbar protection relay and trip outputs from the busbar relay to all circuit breakers connected to that busbar. Fig. 8 shows typical connections among a busbar relay and substation terminals.

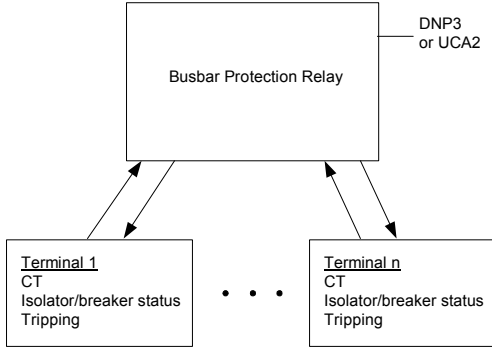


Fig. 8. Busbar Protection

Through a low-impedance busbar protection system, one can obtain information about and access the following:

- All currents at the voltage level where the busbar protection is installed
- All trip coils at the voltage level where the busbar protection is installed

XV. REPLACE A TRADITIONAL RTU WITH BUSBAR PROTECTION

Compare the inputs in Fig. 7 (CTs and VTs, digital inputs, alarms, and control) to the inputs necessary for low-impedance busbar protection (CTs, digital inputs, and tripping) as shown in Fig. 8. If we add VT inputs, alarms, and circuit breaker close control to the busbar protection in Fig. 8, we emulate a station RTU.

Fig. 9 shows the busbar protection relay with the information necessary to emulate a station RTU. The currents from all station terminals as well as the busbar voltage are available in the relay. With this information, we can use the mathematical functions in the relay to calculate real power, reactive power, apparent power, and power factor inside the bus protection relay. After calculating these values, we can map these values into the DNP map together with the other analog quantities and binary points to make this information available to the SCADA master.

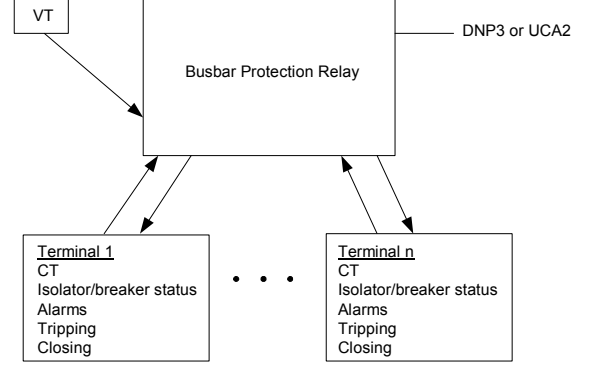


Fig. 9. Busbar Protection With Equivalent RTU Information

From a communications viewpoint, using the busbar protection relay as an RTU requires no communications protocols between an IED and the central unit; we have eliminated the communications complications associated with traditional RTUs. The only remaining uncertainty is the communications protocol between the busbar protection relay and the SCADA master. Most modern relays provide, at a minimum, a choice between DNP3 and UCA2 to cover most non-proprietary communications between RTUs and SCADA masters.

XVI. RTU FUNCTIONAL PROGRAMMING

To provide the busbar protection relay with RTU functionality, we must program the busbar protection relay to calculate the necessary quantities and perform the required control functions. No additional programming is necessary; the programming previously done in the RTU is now done in the protection relay. For near real-time functions, we use the automation programming functions in the relay. Table 5 shows the quantities typically necessary for SCADA, and the quantities directly available in the relay. According to the table, only the RMS currents and voltages, and the phase angle are available; we must calculate the apparent power, real power, and reactive power.

TABLE 5 QUANTITIES REQUIRED FOR SCADA, AND QUANTITIES DIRECTLY AVAILABLE IN THE RELAY

Quantity	Directly Available In Relay
RMS Phase current (I)	Y (InnFM, in primary Amps)
RMS Phase voltage (V)	Y (VmmFM, in primary kV)
Phase Angle (cos f)	Y (InnFA, in degrees)
Real Power (P)	N
Reactive Power (Q)	N
Apparent Power (S)	N

nn = 01 through 18
mm = 01 through 03

To calculate real power, reactive power, and apparent power, we select InnFM, the phase one-cycle average current (primary ampere) and VmmFM, the phase one-cycle average voltage (primary voltage in kV), from Table 4 (both quantities are available in the polar form). From Table 3 we select the COS and SIN operators, and from Fig. 2 (automation control equation programming), we assign the following functions to Automation Math Variables (AMVxxx) AMV001 through AMV009 for Transformer 1:

AMV001: Real power (single phase)
 AMV002: Reactive power (single phase)
 AMV003: Power factor
 AMV004: Intermediate step
 AMV005: Intermediate step
 AMV006: Apparent power (single phase)
 AMV007: 3-phase real power
 AMV008: 3-phase reactive power
 AMV009: 3-phase apparent power

Assuming balanced loading, select the A-phase current of Transformer 1 (see Fig. 10) and the A-phase voltage from the busbar VT to calculate the per-phase real power through use of the well-known equation $P = V \cdot I \cdot \cos \phi$ (multiply angle I01FA by -1 (AMV002) to obtain the conjugate):

AMV001 := V01FM * (I01FM/1000) * COS(I01FA)# REAL POWER (P), MW

(Note: the relay does not process text following the comment symbol (#); include programming information and clarification here)

Similarly, use $Q = V \cdot I \cdot \sin \phi$ to calculate the per-phase reactive power as follows:

AMV002 := V01FM * (I01FM/1000) * SIN(I01FA*-1)# REACTIVE POWER (Q), MVAR

Calculate the power factor as follows:

AMV003 := COS(I01FA)# POWER FACTOR (PF)

Calculate the per-phase apparent power ($S = \sqrt{P^2 + jQ^2}$) as follows:

AMV004 := AMV001 * AMV001# SQUARE THE REAL COMPONENT

AMV005 := AMV002 * AMV002# SQUARE THE IMAGINARY COMPONENT

AMV006 := SQRT(AMV004 + AMV005)# APPARENT POWER (S), MVA

Calculate the three-phase real power, reactive power, and apparent power as follows:

AMV007 := 3 * AMV001# BALANCED THREE-PHASE REAL POWER

AMV008 := 3 * AMV002# BALANCED THREE-PHASE REACTIVE POWER

AMV009 := 3 * AMV006# BALANCED THREE-PHASE APPARENT POWER

XVII. SUBSTATION AUTOMATION

Building on the previous example, consider a case in which we want to control substation loading. Fig. 10 shows the maximum permissible load on each transformer and each

feeder at the substation, as well as the current allocation of each terminal. For example, the three current inputs from Transformer 1 are I01, I02, and I03.

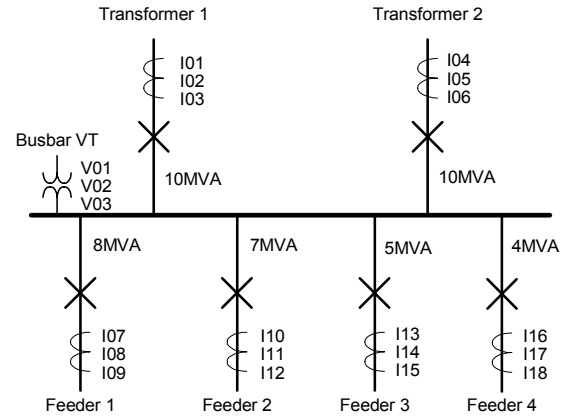


Fig. 10. Maximum Permissible Load on Each Transformer and Feeder

Assume now that we want to control the station loading in the following fashion:

- Trip each feeder individually when the load on the feeder exceeds the corresponding MVA rating.
- Trip Feeder 4 when the MVA sum of the four feeders exceeds 20MVA.

Following the same programming steps for Transformer 1, we can calculate the MVA in each terminal through use of the A-phase (I01, I04, I07, etc.) of each terminal and the A-phase (V01) of the busbar VT. For brevity, the following shows only the relevant information for each terminal:

Feeder 1

AMV025 := 3 * AMV019# BALANCED THREE-PHASE REAL POWER

AMV026 := 3 * AMV020# BALANCED THREE-PHASE REACTIVE POWER

AMV027 := 3 * AMV024# BALANCED THREE-PHASE APPARENT POWER

Feeder 2 through Feeder 4 have similar equations.

Calculate the total MVA at the station as follows:

AMV055 := AMV025 + AMV034 + AMV043 + AMV052 # SUM OF REAL VALUES

AMV056 := AMV026 + AMV035 + AMV044 + AMV053 # SUM OF IMAGINARY VALUES

AMV057 := AMV055 * AMV055 # SQUARE THE REAL COMPONENT

AMV058 := AMV056 * AMV056 # SQUARE THE IMAGINARY COMPONENT

AMV059 := SQRT(AMV057 + AMV058)# TOTAL APPARENT POWER (S), MVA

From Fig. 2, we select control variables ASV001 through ASV005, and program these variables as follows:

ASV001 := AMV027>8# ASSERTS WHEN FEEDER 1 MVA EXCEEDS 8 MVA

ASV002 := AMV036>7# ASSERTS WHEN FEEDER 2 MVA EXCEEDS 7 MVA

ASV003 := AMV045>5# ASSERTS WHEN FEEDER 3 MVA EXCEEDS 5 MVA

ASV004 := AMV054>4# ASSERTS WHEN FEEDER 4 MVA EXCEEDS 4 MVA

ASV005 := AMV059>20# ASSERTS WHEN TOTAL MVA EXCEEDS 20 MVA

XVIII. TRIP EQUATIONS:

The final step is to include the variables in the output of the relay, programmed as follows:

Feeder 1

OUT101 := (...other trip variables) OR ASV001 (trips Feeder 1 when load exceeds 8 MVA)

Feeder 2 and Feeder 3 have similar equations.

Feeder 4

OUT104 := (...other trip variables) OR ASV004 OR ASV005 (trips Feeder 4 when load exceeds 4 MVA, or when the MVA sum of the four feeders exceeds 20 MVA)

XIX. ALARMS AND MONITORING

Because busbar protection must protect complex busbar layouts such as double bus with transfer, the relay has a sufficient number of digital inputs for isolator inputs to perform zone selection for these complex busbars. However, distribution busbar layouts are not as complex, and you can use the unused digital inputs for alarm functions. Furthermore, most modern relays also include a dc monitor to report on the station battery health by measuring the dc voltage, ac ripple, and voltage between each battery terminal and earth.

XX. CONTROL FUNCTIONS

To perform remote open and close commands, we use dedicated relay variables (Table 7) or general relay variables (Table 6). For this example, we assume that DNP is the protocol between the RTU and the SCADA master, and we apply the settings as shown in Table 6 and Table 7.

TABLE 6
CLOSE OPERATION

SCADA Master			Relay Variable
Object 12	Index 24	Close (pulse)	RB2 (Transformer 1)
Object 12	Index 25	Close (pulse)	RB4 (Transformer 2)
Object 12	Index 26	Close (pulse)	RB6 (Feeder 1)
Object 12	Index 27	Close (pulse)	RB8 (Feeder 2)
Object 12	Index 28	Close (pulse)	RB10 (Feeder 3)
Object 12	Index 29	Close (pulse)	RB12 (Feeder 4)

TABLE 7
TRIP OPERATION

SCADA Master			Relay Variable
Object 12	Index 32	Trip (pulse)	OC1 (Transformer 1)
Object 12	Index 33	Trip (pulse)	OC2 (Transformer 2)
Object 12	Index 34	Trip (pulse)	OC3 (Feeder 1)
Object 12	Index 35	Trip (pulse)	OC4 (Feeder 2)
Object 12	Index 36	Trip (pulse)	OC5 (Feeder 3)
Object 12	Index 37	Trip (pulse)	OC6 (Feeder 4)

For brevity, we do not show the relay output equations to close the circuit breakers, but the output equations are similar

to the Trip Equations (and may include supervision by variables ASV001 through ASV005) this paper just discussed.

XXI. ADDITIONAL FUNCTIONS AND BENEFITS

Using a busbar protection relay as an RTU provides additional functions such as event reporting and battery monitoring. Because of the cost, only transmission stations or high-profile distribution substations have digital fault recorders (DFR) installed. In addition to the cost of the DFR equipment, you have to wire CTs, VTs, and the necessary digital inputs to these DFRs, and then still configure the DFR for each station.

After wiring the inputs necessary to serve as an RTU into the busbar protection relay, we have also wired all inputs necessary for a station DFR into the relay. Station-wide DFR functionality is possible because most modern relays permit you to initiate an event report from an external input. With this external initiation, you can trigger event reports in the busbar protection relay for faults other than busbar faults, such as line faults and transformer faults.

In addition to the hardware of a protection device being more suitable for a substation environment than the hardware of an RTU, a single device has fewer components, thus reducing overall system failures [3] [4] [5].

XXII. BUSBAR PROTECTION PROVIDES OVERCURRENT AND EARTH-FAULT PROTECTION

Because currents from all terminals are available in the busbar protection relay, overcurrent and earth-fault protection is readily available. In this particular instance, the relay provides overcurrent protection only, and we need to program a definite-time earth-fault element. Here we use protection variables instead of automation variables, because the relay processes protection variables together with all other protection functions. For Transformer 1, we use I01FIM (angle I01FIA), I02FIM (angle I02FIA), I03FIM (angle I03FIA), and the phase-filtered instantaneous currents (see Fig. 10). For the definite-time earth-fault element, we use the following general equation:

Where:

IN = Neutral current

3I0 = Three times the zero-sequence current

IA = A-phase current

IB = B-phase current

IC = C-phase current

Therefore, to program an earth-fault element, we sum the three phase currents. However, because the relay provides the current values in polar form, we must first convert the current values to rectangular form before addition. We calculate the real values and add the three phases in one programming step as follows:

$$PMV01 := I01FIM \cdot \cos(I01FIA) + I02FIM \cdot \cos(I02FIA) + I03FIM \cdot \cos(I03FIA)$$

Similarly, we calculate the imaginary values and add the three phases as follows:

$$PMV02 := I01FIM \cdot \sin(I01FIA) + I02FIM \cdot \sin(I02FIA) + I03FIM \cdot \sin(I03FIA)$$

Calculate the square root of the sum of the squares as follows:

PMV03 := PMV01*PMV01# SQUARE THE REAL COMPONENT

PMV04 := PMV02*PMV02# SQUARE THE IMAGINARY COMPONENT

PMV05 := SQRT(PMV03 + PMV04) #TAKE THE ROOT OF THE SUM OF THE SQUARES

PMV05 is IN, the neutral current. For a definite-time element, we must start a timer when the current value exceeds a specified threshold.

PSV10 := PMV05 > 0.2 #SET THE PICKUP VALUE TO 0.2A

PCT10PU := 50 #SET THE TIMER TO 50 CYCLES (1 SECOND)

PCT10DO := 0 #SET THE DROPOUT TIME TO 0

PCT10IN := PSV10#INITIATE THE TIMER WHEN PSV10 ASSERTS

Include the timer output (PCT10Q) in the trip equation of each terminal. For example, the trip equation for Feeder 1 may be the following: OUT101 := (... other trip variables) OR ASV001 (trips Feeder 1 when load exceeds 8 MVA) OR PCT10Q (definite-time earth-fault element)

XXIII. SUMMARY

- We have shown examples of programming tools available in modern protection relays.
- We have recognized the importance of separating automation and protection variables, and we have shown the difference in execution time, execution storage, and settings storage space when using automation and protection variables.
- We used automation variables to program a busbar protection relay to serve as a traditional RTU, while the relay still performs the primary function of busbar protection.
- We used protection variables to program a definite-time earth-fault element, using the analog quantities available in a busbar protection relay.
- We have shown that by using a busbar protection relay to serve as a traditional RTU, we eliminate intervender protocol incompatibilities and avoid other network complications, such as IED failure and lack of network bandwidth.
- We have shown that, in addition to serving as a traditional RTU, busbar protection is also a digital fault recorder, as well as a battery-monitoring device.

XXIV. CONCLUSIONS

1. You can use a modern numeric relay to provide not only core protection functionality but also eliminate additional automation hardware, such as an RTU, by using the flexible logic and functionality within the relay.
2. With separate automation and protection functions, the relay shares information between automation and protection program areas, while maintaining relay security and dependability.
3. Although automation and protection control equations use the same relay element variables as those used in the pre-programmed functions, using these variables in programming has no effect on protection functions.

4. Mathematical programming is in a format that is both familiar and easy to use.
5. Using the programming operators and mathematical functions provides many cost-saving opportunities such as capacitor control, transformer tap changing, breaker and disconnect interlocking, etc.
6. Use of protection grade equipment for automation reduces the failure rate by an order of magnitude. RTUs generally have a mean time between failures (MTBF) of between 12 and 15 years, while some protective relays have an MTBF of 100 to 300 years.
7. Using a single robust and reliable protection relay to replace several BCUs, relays, and an RTU greatly increases the reliability of the solution. Fewer, more reliable products mean less risk of failure.

XXV. REFERENCES

- [1] SEL-487B Instruction Manual
- [2] SEL-487B Relay Software Specifications
- [3] D. Dolezilek, "Case Study Of A Large Transmission and Distribution Substation Automation Project," Proceedings of the 1999 Western Power Delivery Automation Conference.
- [4] D. Dolezilek, "Choosing Between Communications Processors, RTUs, and PLCs as Substation Automation Controllers," SEL White Paper, October 17, 2000.
- [5] P.M. Anderson, B. Fleming, T.J. Lee, and E.O. Schweitzer, III, "Reliability Analysis of Transmission Protection Using Fault Tree Methods," Proceedings of the 24th Annual Western Protection Relay Conference, Spokane, Washington, October 21-23, 1997.

XXVI. BIOGRAPHIES

Casper Labuschagne earned his Diploma (1981) and Masters Diploma (1991) in Electrical Engineering from Vaal Triangle Technicon, South Africa. After gaining 20 years of experience with the South African utility Eskom, where he served as Senior Advisor in the protection design department, he began work at SEL in 1999 as a Product Engineer in the Substation Equipment Engineering group. Presently, he is Lead Engineer in the Research and Development group. He is registered as a Professional Technologist with ECSA, the Engineering Counsel of South Africa.

Izak van der Merwe obtained his B. Eng (Electrical) degree from the University of Stellenbosch in 1991. He started to work for Eskom in 1993 as a Protection Field Engineer. He is currently a Chief Engineer: Protection Specialist in Resources and Strategy, a Corporate Division of Eskom Holdings Limited. He is responsible for Distribution Division National Contracts for protection schemes and equipment, general protection technology direction setting and technology management, and the implementation of Distribution Automation and Substation Automation in Eskom's Distribution Division. He has authored a number of protection and substation automation related papers. He is a Registered Professional Engineer in South Africa.