

5 RSS System Model

The complete model to test the operation of the RSS consists of two parts: the existing relay system model to adapt and the RSS model. Both systems are modeled in Matlab. The existing relays are modeled so they are explicitly told when to trip for a fault. The RSS model is a complete model, and uses actual EMTP data to perform relay calculations. Trip signals from the existing relays trigger the RSS into action. This signal is supplied by an auxiliary contact from the existing relay.

The general flow of the system model is shown in Figure 10. For each sample of an EMTP case study, the model sets trips from existing relays (in the Faultsys routine), performs supervision of the existing relays (in the RSS routine), and sets any breaker trips. The model continues until all samples of the case study are exhausted. Reading one sample at a time permits time synchronization between the existing relay models and the RSS.

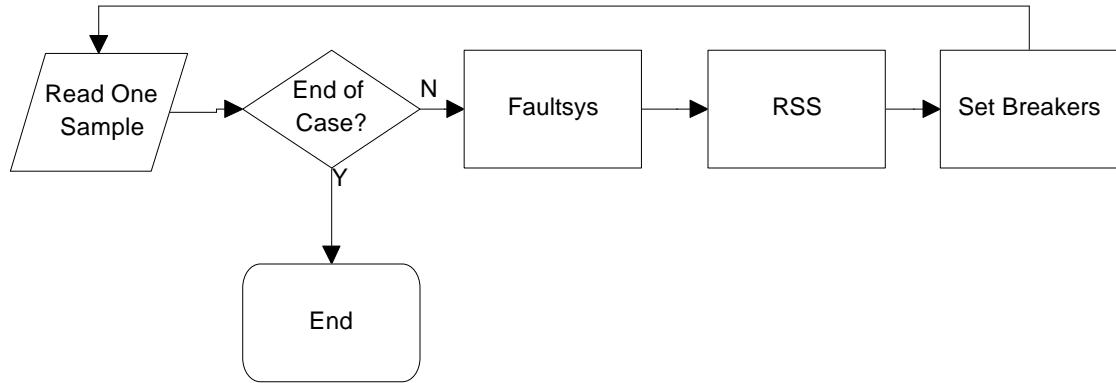


Figure 10: Device Model Flowchart

5.1 Breakers

Breaker status is determined by a series combination of any breaker operations provided by existing relays and from the RSS. The breakers modeled take 1 cycle to open. Setting a breaker to trip initiates the breaker failure relay scheme. This scheme works on a timer. As long as the breaker should trip, and remain closed, the breaker failure timer increments every sample. When this value reaches the timer setting, the breaker failure scheme sends a trip signal to all breakers included in the breaker failure scheme. When the breaker opens, the breaker failure timer is reset to 0.

5.2 Existing relay models

The existing relays are explicitly told which faults to trip for. It is assumed that the relays always work correctly for all faults within their zone of protection. Hidden failure modes change the zone of protection. The Faultsys routine tests the existing relay models. Based on information supplied for the test case, the routine checks to see if the fault causes any relays to trip. This includes the effects of any hidden failure modes. The routine returns the state of all existing relays to the main program. Every EMTP case has a switch to identify fault inception. No relay operates until this switch closes.

Transmission lines, breakers, and transformers are all object types in the Matlab code. Each object has a method routine to test the relays associated with the device. Each object also contains a list of breakers to trip should the device operate. The methods follow the protection schemes listed in Table 14. The Faultsys routine tests the existing relay model. Based on information supplied for a test case, the routine checks to see if the fault causes any relays to trip. This includes the effects of hidden failure modes.

Table 14: Existing Protection Schemes

Line Protection	Phase	Ground
Primary	Step-Distance	Step-Distance
Backup	Directional Comparison Blocking	Phase Comparison Blocking
Backup	Phase Comparison Blocking	Directional Overcurrent

Transformer Protection	Phase	Ground
Primary	Differential	Differential

Breaker Failure Protection	Phase	Ground
Primary	Overcurrent Breaker Failure	Overcurrent Breaker Failure

Table 15: Model Operating Timer Settings for Relays

Relay	Operating Time (cycles)
Zone 1	2
Zone 2	2 + Timer setting (0.3 sec * 60 Hz)
Zone 3	2 + Timer setting (0.6 sec * 60 Hz)
Directional Comparison Blocking	Timer setting (0.3 sec * 60 Hz)
Phase Comparison Blocking	1.5
Directional Overcurrent	Time Dial Setting (0.4 sec * 60 Hz)
Transformer Differential	1
Breaker Failure	1
Other Devices	
Contacts	0.5
Breakers	1

The methods used to test existing relays are all similar. Each method determines if the fault is in the operating direction, and if this specific relay is operating under a hidden failure mode. The next step determines if the fault is within the relay zone of protection. For some relays, the information needed to determine this is the distance to the fault, in percent of line length. For other relays, the information needed is the steady state fault current magnitude in secondary amps. If the fault is in the zone of protection, the method starts a timer. This timer models the length of time the actual relay takes to recognize and operate for a fault. Table 15 lists the operating time for each existing relay model. Once this timer expires, the method returns a trip signal.

The existing relay system models require the following information: Faulted line, location of fault in percent of line length, and whether it is a phase or ground fault. Relays that work on fault current magnitude, such as overcurrent relays, require a steady-state fault current value.

The Faultsys routine tests the existing relay model. For each line, transformer, and breaker, the routine checks to see if the fault is in the zone of protection for the relay, or whether a hidden failure causes the fault to be within the relay zone of protection.

5.3 RSS Model

The RSS is the model of the substation computer implemented in Matlab. The inputs to the model are supplied by the existing relay model, and by EMTP. Anti-aliasing filter of the

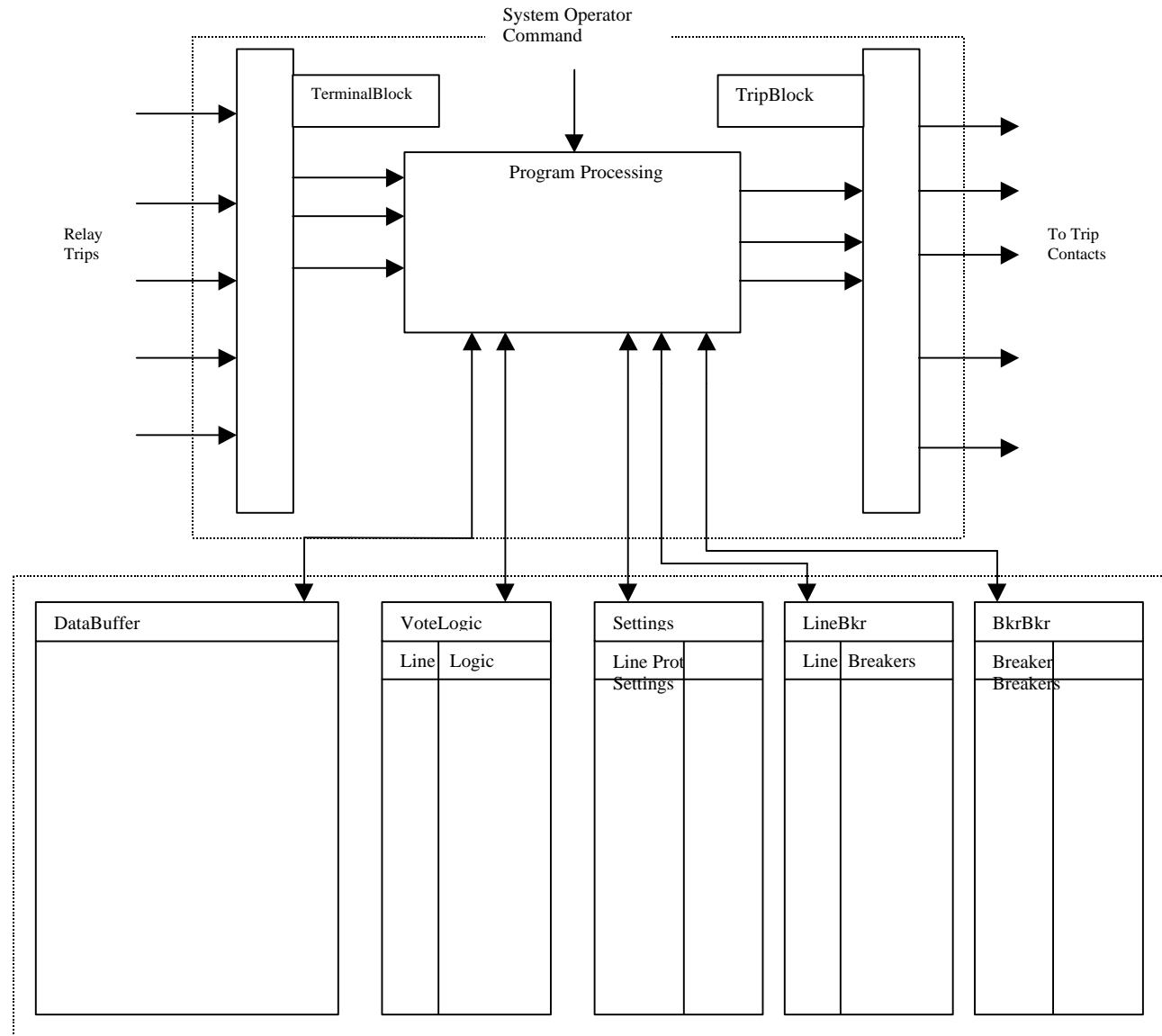


Figure 11: RSS Schematic

voltage and current waveforms are modeled in EMTP. The RSS samples waveforms at 12 times per cycle, as represented in the EMTP model. The existing relays provide trip signals to trigger the RSS into action.

The RSS has three operating modes: Normal, Emergency, Out of Service. These modes only supervise relays at the local substation. For any relay, any one of these modes must return a trip result for 3 consecutive samples to allow a device operation. This ensures the reliability of the fault calculation.

5.3.1 General Flow

The RSS continuously samples and stores voltage and current waveforms for all devices. To simulate time flow through the fault case, the model actually works step by step through an array of sample data. This achieves time synchronization between the existing relays, the RSS, and the breakers. The RSS works on trip signals, which enter through TerminalBlock. Each element in TerminalBlock represents a contact input from a specific protection scheme for a specific device. When a trip signal enters, the RSS chooses an algorithm based upon which of the three operating modes the RSS is in: Normal, Emergency, or Out of Service. The RSS always operates in Normal mode, with two exceptions. An external command from the system operator can place the RSS in Emergency mode. An external command from the system operator can place the RSS in Out of Service mode as well, or the RSS can place itself in Out of Service mode due to component failure. Communications signals from the system operator, and the RSS self-test, are modeled as user input.

Normal mode determines performs hidden failure supervision for any protection scheme with a trip signal. To perform the hidden failure supervision, the algorithm retrieves information for this line stored in Settings. Settings contains the type of hidden failure supervision scheme to use, and the actual relay settings for this scheme. The specific supervision algorithm uses the relay settings to perform a fault calculation. If the fault is in the zone of protection for the scheme, the routine increments the Trip status in TimerBlock corresponding to this protection and device by one. If the fault is outside the zone of protection, the algorithm resets the Trip status in Timerblock to zero. Three consecutive successful fault calculations are necessary to actually trip a breaker, so the value of Trip must equal three to actually send a trip output signal. When the value of Trip reaches three, the TripBlock output contact is set to trip for all breakers associated with this device.

Emergency mode also determines which specific protection schemes for specific devices have trip signals. First, this routine retrieves the logic information of the vote scheme for this line, stored in VoteLogic. If the logic scheme in VoteLogic has three or more protections, the routine applies the logic scheme. If at least two protection schemes for the line included have trip signals, the vote logic scheme decides there is a trip. For a trip, the routine increments Trip in TimerBlock for this line. Otherwise, Trip is reset to zero. As in Normal mode, there must be three consecutive successful vote calculations to actually set an output trip. When the value of Trip equals three, the TripBlock output contact is set to trip for all breakers associated with this line. Lines or devices that don't have a logic scheme stored in VoteLogic, such as transformers and breakers, are pushed into normal mode hidden failure supervisor.

Out of Service mode models the situation where the RSS is removed from service. This can occur due to failure of the RSS during self-test, removal due to maintenance, etc. In this case, the TripBlock output contact is set to trip all breakers.

5.3.2 Hidden Failure Supervision

This section describes in detail the implementation of all hidden failure supervision

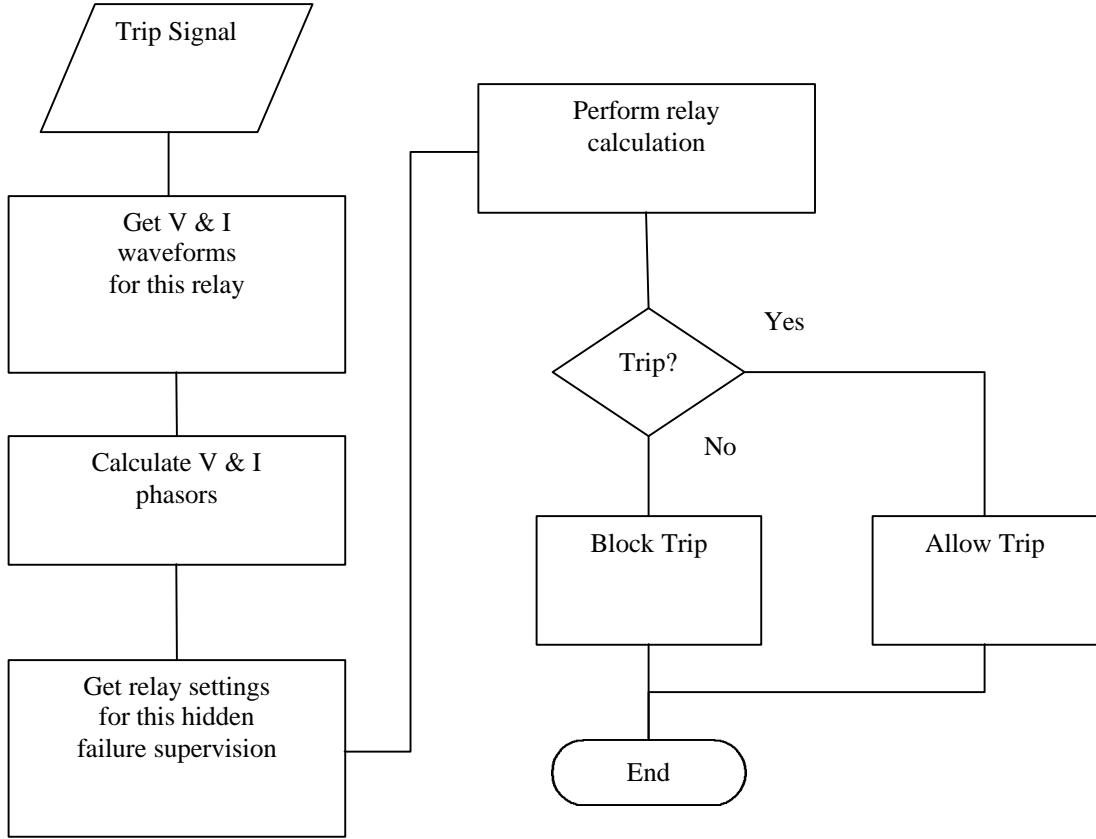


Figure 12: General Hidden Failure Algorithm

schemes. All schemes follow the same general program flow as illustrated in Figure 12.

5.3.2.1 Zone

The Zone algorithm models a distance relay using a mho circle characteristic. The algorithm is used exactly as is for the Step-Distance supervision method, and is a building block for other supervision methods. The Zone algorithm is called once for each zone of a distance

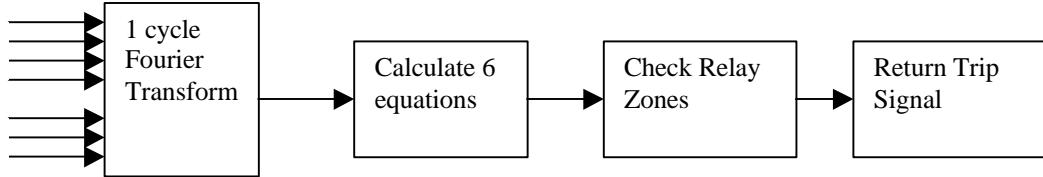


Figure 13: Distance Relay Algorithm

relay. This algorithm can handle relay zones that operate instantaneously or operate on a coordination timer. The basic algorithm is the same in either case. The algorithm calculates 6 impedance equations, and compares each of these equations to the zone setting.

The routine first uses the 12 most recent data samples for every current and voltage waveform to perform the one cycle Fourier transform. The resulting fundamental phasors for the voltages and currents are used to in the 6 fault impedance equations.

$$\begin{array}{ccc} \frac{E_a - E_b}{I_a - I_b} & \frac{E_b - E_c}{I_b - I_c} & \frac{E_c - E_a}{I_c - I_a} \\ \frac{E_a}{I_a + mI_0} & \frac{E_b}{I_b + mI_0} & \frac{E_c}{I_c + mI_0} \end{array}$$

(m is the zero sequence compensation factor, defined as: $\frac{Z_0 - Z_1}{Z_1}$). The algorithm then

compares the 6 impedances to the zone reach setting. If any of the 6 impedances are within the mho circle, the algorithm returns a trip.

For cases where the relay zone uses a coordination timer, the algorithm is performed twice. The first instance verifies a fault still exists, and the second instance supervises the zone time by verifying the fault existed to actually start the timer. The first calculation is performed exactly as above, using the 12 most recent samples to convert to phasors. If this calculation returns a trip, indicating a fault currently exists, the algorithm runs a second time. In this case, the algorithm scrolls backwards through the stored data samples for an amount of time equivalent to the timer setting. The 12 previous samples from this point in the buffer are used to determine the phasors and calculate the fault impedance. For example, if the timer setting is 0.3 seconds, the data from $[0.3\text{sec} \times 60^{\text{cycles}}/\text{sec} \times 12^{\text{samples}}/\text{cycle}] = 216$ samples previous is used.

If this second calculation returns a trip, then the whole algorithm returns a trip. Table 16 shows the required settings for the Zone relay. Actual settings for each line are shown in Appendix IV.

Table 16: Zone Relay Settings

Setting	Units
Positive sequence impedance	secondary ohms
Zero sequence impedance	secondary ohms
Maximum torque angle	radians
Zone reach setting	secondary ohms
Timer setting	seconds

5.3.2.2 AccelZ

The AccelZ routine is a Zone 1 extension scheme, used to supervise hidden failure modes that depend on carrier signals. This routine first performs a zone 1 (under-reaching) distance calculation, using the Zone algorithm. If this calculation returns a trip, the whole algorithm returns a trip. If this distance calculation doesn't return a trip, the routine waits for a specified length of time and performs a zone 2 (over-reaching) distance calculation. This time is stored in TimerBlock. If this zone 2 calculation returns a trip, the whole algorithm returns a trip. Every sample increments the timer in TimerBlock.

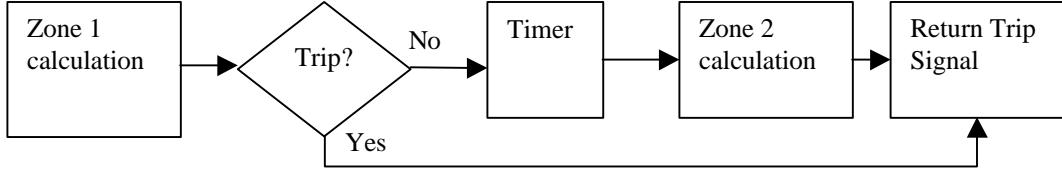


Figure 14: Zone 1 Extension Algorithm

Table 17: AccelZ Relay Settings

Setting	Units
Positive sequence impedance	secondary ohms
Zero sequence impedance	secondary ohms
Maximum torque angle	radians
Zone 1 reach setting	secondary ohms
Zone 2 reach setting	secondary ohms
Delay timer setting	seconds

5.3.2.3 DirOC

The DirOC routine performs voltage and current polarization to use with directional

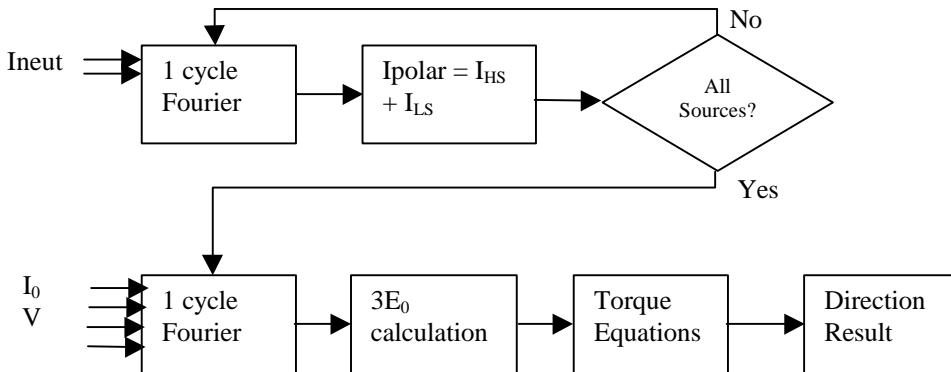


Figure 15: DirOC Relay Algorithm

overcurrent relays. Current polarization uses the neutral current from the high side and low side grounded-wye windings of the three-winding transformers. It uses at least one, but can use more transformers as a source for the polarizing current. All the relays in this system use both transformers as polarizing current sources. The 12 most recent samples are used for the neutral current waveform. These samples are converted to phasors using a one cycle Discrete Fourier Transform. The high side neutral current is reflected to the low side base. The algorithm also mathematically matches any difference between the high side and low side CT ratios. The polarizing current is the sum of all the high side and low side neutral currents.

Voltage polarizing compares the zero sequence line voltage to the zero sequence line current. The algorithm uses the 12 most recent samples of the zero sequence current and the line