Node Breaker Model Based Transient Stability Simulations including Protective Devices Modeling and Time Coordination

Vibhuti Sahu, Gurunath Gurrula, Sanjeev Bangali and Sarasij Das

Department of Electrical Engineering
Indian Institute of Science, Bangalore, India

Abstract—Modeling protection devices in the Transient Stability Assessment (TSA) of power systems results in an accurate estimation of system stability and behaviour. Miscoordination/misoperations of the relays and circuit breakers (CBs) result in higher order contingencies, $N > 1$. Substation (SS) configurations and associated CB operations are usually ignored in TSA, even though the relays are modeled in detail. Replicating such scenarios needs manual fabrications in Bus-Branch (BB) models used in commercial TSA tools, which can be eliminated using Node-Breaker (NB) models. Also, incorporating relay algorithms, including the realistic CB operations and coordinating their timings, increases the complexity of the TSA code for large systems. The development of TSA tools to simulate the practical operation of CBs due to protective relay decisions similar to the physical substation operation is essential. This paper attempts in this direction. This paper uses a systematic approach to convert BB models to NB models. The Sparse Tableau Approach (STA) is used to represent the system in the NB model. It provides an algorithmic approach for automatically placing differential, distance and under/over frequency relays. It also gives programming logic to coordinate the timings between relays and the associated CBs. The program complexity increases with the size of the system as the number of CBs and the associated timers significantly increase based on the station configuration. The simulation results demonstrate the scalability and effectiveness of the proposed framework for the WECC 9 bus, New England 39 bus, and Polish 2383 bus systems. The WECC 9 bus system results are compared and validated with the commercial software PSS/E.

Index Terms—Sparse Tableau Approach, Node-Breaker, Transient Stability Simulation, Relay Time Coordination

I. LIST OF ABBREVIATIONS

$N_{st}$: Number of stations
$N_{s}$: Total number of series elements
$N_{sh}$: Total number of shunt elements
$N_{se}$: Total number of source elements
$N_{ele}$: Total number of elements except circuit breakers in a station
$N_{node}$: Total number of elements connected to a node
$N_{line}$: Total number of lines in the system
$N_{trd}$: Total number of transformers in the system
$N_{cum}^{(k)}$: Total number of cumulative nodes till $k^{th}$ station
$N_{phy}$: Number of physical buses in a station
$N_{int}$: Number of intermediate nodes created

This work is carried out under the project “BLACK OUT LOOK-AHEAD SIMULATION TOOL (BLAST)” funded by National Supercomputing Mission, a joint mission by Department of Science and Technology (DST) and Department of Electronics and Information Technology (DeitY), India, DST/NSM/R&D/HPC/Applications/2021/03.35.
Depending on the substation configuration, the BFP can lead to additional elements tripping and delay in clearing the fault. Similarly, a busbar relay clearing a fault in a substation with a breaker and half scheme differs from a main and transfer scheme. In the former case, bus-splitting will happen; in the latter case, all the substation elements will be tripped. To create such realistic scenarios and study their impacts, one must understand the end effect of these scenarios and make suitable modifications to the Y-Bus matrix in the existing commercial TSA tools. This is because these tools do not explicitly model the CBs and their timing, even though they take station configurations as input [10].

In [11]–[14], the importance of considering breaker failures for system reliability assessment has been addressed. Neural networks are employed to model the protection systems in [11], dealing with the uncertainties involved with relay and circuit-breaker operation messages to identify fault sections after contingencies. Two types of protection failures, undesired-tripping mode and fail-to-operate mode, and their impact on reliability modeling are discussed in [12]. In [13], an event-based dual-timer BFP scheme is implemented with an additional signal depending on the distance of the multiphase fault from the busbars of the power plant for transient stability simulations. A study in [14] proposes a generalized analytical methodology to identify the failure events due to the stuck breaker condition through search algorithms and repeated matrix operations. In [15], the importance of using NB models for steady state cascade analysis has been examined for a practical system to meet regulatory compliance. In [16], NB modeling for optimal power flow using Sparse Tableau Approach (STA) was proposed. In [17], we have proposed NB model for Transient Stability Analysis (TSA) using STA for the first time in the literature. However, the relay and CB timings coordination is not included.

Existing TSA tools have limitations to accommodate all the above facilities at once. Typically, utilities possess the NB data of their network in the Energy Management System (EMS) in CIM format. EMS platforms convert the NB model to the BB Model via Network Topology Processing (NTP) [15] which is further used by other analysis tools in EMS. However, these converted BB models are not accessible directly to external TSA tools. A compatible CIM file or custom data files in NB format must be manually created for NB models to be used in TSA tools. However, as they use Y-Bus/ power flow-based network solutions, they can only mimic the end effect of CB operations in a station due to relay decisions. Moreover, most test systems used in TSA by researchers today are available as BB models only. So, creating an NB model from a BB model in an automated manner is essential. In [18], we have proposed an algorithmic approach to convert BB models to NB models for different types of station configurations.

Developing a generalized transient stability program including conversion of BB model to NB model, populating all the relays automatically, associating the CBs with the relays and coordinating the timings for large systems to replicate the practical scenarios is a non-trivial task [15]. The following are the contributions of the paper to address the limitations of existing TSA tools.

- A unified framework based on the methods proposed in [17], [18] and discussion on the programmatic approaches to automatically populate relays (distance relays, differential relays and under/over frequency relays), CBs associated with each relay and timers of the relays and CBs in time domain simulation for TSA.
- Systematic approach for time coordination between the simulation time step, protective relays trip time in differ-
ent zones and their associated CBs operating time.
- A comparison of the results and execution time using the proposed framework for WECC 9 bus system with the commercial software PSS®E.
- Simulation results showcasing the flexibility and scalability of the framework on the New England 39 bus system and the Polish 2383 bus system.

The proposed framework will help in new commercial TSA tools development which will alleviate the manual diligence required for creating practical cascade scenarios due to relay and CB operations in substations.

III. PROPOSED SIMULATION FRAMEWORK

This section explains the proposed simulation framework, from data preparation to time domain simulation. Fig. 1 describes the framework in form of a block diagram. The proposed approach shown in Fig.1 builds the NB model of the system in the form of Sparse Tableau, which allows the modeling of CB operations that occur due to relay decisions realistically. In this paper, we use the methodology proposed in [18] for BB to NB model conversion, as shown in Fig.1. The test system data is taken from MATPOWER in BB format. A CIM interface can also be used here. After processing this BB data, input files containing series, shunt and source elements data, as shown in Table I - Table III, will be generated. Using data, input files containing series, shunt and source elements information about the generators (source elements). The connecting nodes and CBs operating time.

An NB model for a substation with Breaker and Half (BAH) configuration is shown in Fig. 2. The final output of the algorithms in [18] gives NB data as in Table IV. File 1a contains information about loads (shunt elements), File 2 contains information about loads (shunt elements), and File 3 contains information about the generators (source elements). The connecting nodes of all the CBs in the network are stored in File 4. After the NB model of the network is developed, the NB data is presented.

In transient stability simulation, \( M_{\text{STF}} X = I \) is solved to give \( X \). Here, \( I \) is the current injection vector, and \( X \) is a vector containing node voltages and voltages and currents at each port of elements. The resultant vector \( X \) is sent to the Protect Devices module. This module’s structure and functioning are described in detail in the next section. The relay algorithm processes the measurements and trips CBs, if any. The change in relay status is reflected via changing the corresponding \( F_v \) and \( F_i \) entries as shown in (2) in \( M_{\text{STF}} \) in the next time loop.

### Table I: Input file 1 format

<table>
<thead>
<tr>
<th>MATPOWER Bus Number</th>
<th>Configuration</th>
<th>( N_{\text{el}} )</th>
<th>( N_{\text{sh}} )</th>
<th>( N_{\text{so}} )</th>
</tr>
</thead>
</table>

### Table II: Input file 2 format

<table>
<thead>
<tr>
<th>MATPOWER Bus Number</th>
<th>Configuration</th>
<th>Indexes of Series Element connected</th>
<th>Indexes of Shunt Element connected</th>
<th>Indexes of Source Element connected</th>
</tr>
</thead>
</table>

### Table III: Input file 3 format

<table>
<thead>
<tr>
<th>Series Element Index</th>
<th>MATPOWER From Bus</th>
<th>MATPOWER To Bus</th>
</tr>
</thead>
</table>

### Table IV: NB Data format

<table>
<thead>
<tr>
<th>File</th>
<th>Type</th>
<th>From Node</th>
<th>From Station</th>
<th>To Node</th>
<th>To Station</th>
</tr>
</thead>
<tbody>
<tr>
<td>File 1a</td>
<td>Line</td>
<td>From Node</td>
<td>Station</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File 1b</td>
<td>Trafo</td>
<td>From Node</td>
<td>To Node</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File 2</td>
<td>Load</td>
<td>Station</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>File 3</td>
<td>Gen</td>
<td>Station</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>File 4</td>
<td>CB</td>
<td>To Node</td>
<td>Station</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table V: Node Data

<table>
<thead>
<tr>
<th>Node Number(n)</th>
<th>Total Elements Connected ( \left( n.eleType(n) \right) )</th>
<th>( n.eleType(n) )</th>
<th>( n.eleNo(n) )</th>
<th>( n.eleNo(n) )</th>
</tr>
</thead>
</table>

---

23rd Power Systems Computation Conference
Paris, France — June 4 – 7, 2024

PSCC 2024
Algorithm 1: Busbar differential relays ($R_{87BB}$)

1: Assume: $A_{num}^{index}(0) = 0$;
2: $r_{num} \leftarrow 1$; // index for relay number
3: for $i = 1:N_{ele}$ do
4: for $j = 1:N_{ele}$ do
5: $CB_{num} \leftarrow 0$; // index that stores associated CB number to relay
6: $R_{87BB}(r_{num}).Node \leftarrow (A_{num}^{index}(i - 1) + j) / \text{physical buses in a station are numbered first}$
7: for $k = 1:R_{87BB}(r_{num}).Node.N_{ele}$ do
8: if $R_{87BB}(r_{num}).Node.eleType(k) == CB$ then
9: $CB_{num} \leftarrow CB_{num} + 1$
10: $R_{87BB}(r_{num}).CB(CB_{num}) \leftarrow R_{87BB}(r_{num}).Node.eleNo(k)$ // CB number is associated to relay
11: end if
12: end for
13: $R_{87BB}(r_{num}).CB_{total} \leftarrow CB_{num}$
14: $r_{num} \leftarrow r_{num} + 1$
15: end for
16: end for

Algorithm 2: Line Distance Relays ($R_{21L}$)/ Transformer differential relays ($R_{87T}$)

1: $r_{num} \leftarrow 1$
2: For $R_{87T}$, replace line-trafo & $R_{21L} \rightarrow R_{87T}$
3: for line = 1:$N_{line}$ do
4: $R_{21L}(r_{num}).Node \leftarrow line.FromNode
5: $R_{87T}(r_{num}).Node \leftarrow trafo.ToNode // skip for $R_{21L}$
6: $R_{21L}(r_{num} + 1).Node \leftarrow line.ToNode // skip for $R_{87T}$
7: $CB_{num} \leftarrow 0$
8: for $k = 1:line.FromNode.N_{ele}$ do
9: if $line.FromNode.eleType(k) == CB$ then
10: $CB_{num} \leftarrow CB_{num} + 1$
11: $R_{21L}(r_{num}).CB(CB_{num}) \leftarrow line.FromNode.eleNo(k)$
12: end if
13: end for
14: $R_{21L}(r_{num}).CB_{total} \leftarrow CB_{num} // skip for $R_{87T}$
15: $r_{num} \leftarrow r_{num} + 1 // skip for $R_{87T}$
16: $CB_{num} \leftarrow 0 // skip for $R_{87T}$
17: for $k = 1:line.ToNode.N_{ele}$ do
18: if $line.ToNode.eleType(k) == CB$ then
19: $CB_{num} \leftarrow CB_{num} + 1$
20: $R_{21L}(r_{num}).CB(CB_{num}) \leftarrow line.ToNode.eleNo(k)$ // consecutive relay for pilot protection only for $R_{21L}$
21: end if
22: end for
23: $R_{21L}(r_{num}).CB_{total} \leftarrow CB_{num}$
24: $r_{num} \leftarrow r_{num} + 1$
25: end for

Algorithm 3: Load Frequency Relays ($R_{81L}$)/ Generator Frequency Relays ($R_{81G}$)

1: $r_{num} \leftarrow 1$
2: For $R_{81G}$, replace load-gen & $R_{81L} \rightarrow R_{81G}$
3: for load = 1:$N_{load}$ do
4: $R_{81L}(r_{num}).Node \leftarrow load.Node
5: $CB_{num} \leftarrow 0$
7: if load.Node.eleType(k) == CB then
8: $R_{81L}(r_{num}).CB(CB_{num}) \leftarrow load.Node.eleNo(k)$
9: end if
10: end for
11: $R_{81L}(r_{num}).CB_{total} \leftarrow CB_{num}$
12: $r_{num} \leftarrow r_{num} + 1$
13: end for

In the format shown in Table IV. Using this data, relays can be placed for elements in network. All the data in File(1)-(4) is stored in arrays. The given data form a new vector of structures that stores node information as given in Table V. Here, $n$ is the node number in the system, $n.N_{ele}$ gives the total number of elements (CB/line/trafo/load/generator) connected to node $n$. The members $n.eleType(k)$ and $n.eleNo(k)$ give element type and element number of the $k$th element connected to the node.

For this paper we have considered differential relays for busbars, distance relays with $Mho$ characteristics for lines, differential relays for transformers and under/over frequency relays for load/generator. Since these relays protect all the components, other relays are not considered. Although any other relay functions can be employed. Relay information is stored in arrays denoted via ANSI code for each type of relay. For example, $R_{87BB}$, $R_{21L}$, $R_{87T}$, $R_{81L}$, $R_{81G}$ are the relay arrays that store information of busbar differential, distance, transformer differential, under and over frequency relays respectively. The length of an array $R_X$ is $N_X$ where $X$ is the ANSI code. Each element of the array is a structure. For example, $R_{87BB}(k)$ is the structure for $k$th differential relay. $R_{87BB}(k).Node$ stores the node number of busbar protected, $R_{87BB}(k).CB_{total}$ stores total number of CBs connected to the node and $R_{87BB}(k).CB(i)$ for $R_{87BB}(k).CB(CB_{total})$ stores CBs that are associated with the relay. The same convention is used for other relays $R_{21L}$, $R_{81L}$, and $R_{81G}$. Algorithms are provided for the placement of relays in the system. Placement of relays means storing relay number, type, element it is protecting and the CBs associated with it. The key idea behind algorithms is to identify the node to which the element to be protected is connected from Table V. Each element connected to that node is spawned. If it is a CB, the CB number is stored as one of the associated CBs to the relay protecting the element. Then, same is repeated for next element with an incremented relay number. The algorithms provided in this paper currently populate 3 types of relays automatically in the following manner:

- **Differential relays for busbars are populated bus number-wise using Algorithm-1.**
- **Distance relays for lines and differential relays for transformers are populated line/transformer number-wise using Algorithm-2.**
- **Frequency relays are populated load/generator number-wise using Algorithm-3.**

Algorithm-1 shows the steps for placement of differential relays for busbar protection in the network. The first consecutive node numbers in each station are assigned to physical buses in the node data (Table V). Each busbar differential relay is associated to these physical buses in the Algorithm; step 6. All the CBs connected to the nodes are associated to the relays in step 10. Algorithm-2 shows the steps for placement of distance relays and differential relays for lines and transformers connected at each station in the network. Here, the relays associated with the two ends of the line are consecutively numbered; see steps 3 and 5. This gives flexibility in identifying the CBs that should be tripped for pilot protection. The CBs connected to the $FromNode$ and $ToNode$ are associated with the respective relays in steps 10 and 19 respectively. The same logic can be used to place
differential relays for transformers by replacing the term line with trafo and associating both ends to a single relay. Step 4 can be performed instead of step 5 and steps 12, 13 and 14 can be skipped. Algorithm-3 gives placement of frequency relays for transformers by replacing the term differential characteristics [19] for $R_{57BB}$, Mho characteristics and adjacent line impedances for $R_{21L}$ [20] and frequency settings for $R_{81LG}$ [21].

B. Relay and Circuit Breaker Trip-Time Coordination

This section provides the framework for time coordination between relays and CBs in simulation. In simulation, the network solutions give voltages at each node and current at each port ($M_{STF} X = I$). These node voltages and current measurements are further processed to give two arrays $V_R$ and $I_R$ respectively. $V_R$ stores voltage measurements for relays that require voltage as an input. $I_R$ stores current measurements for relays that require current as an input. Since the identification number for the relays is already assigned via Algorithm 1-3, the relay measurements are arranged in the following form, so that it is easier to send arrays to the respective relay functions:

$$V_R = \begin{bmatrix} V_{R21L} \\ V_{R81L} \\ V_{R81G} \end{bmatrix}; \quad I_R = \begin{bmatrix} I_{R87BB} \\ I_{R21L} \\ I_{R81L} \\ I_{R81G} \end{bmatrix}$$

Each section of $V_R$ and $I_R$ is sent to respective relay algorithms sequentially (Fig. 1). For a particular time loop, $V_R$ and $I_R$ will be same for any relay, hence the sequence of relays does not matter. In implementation, we have considered sequence Differential $\rightarrow$ Distance $\rightarrow$ Under/Overfrequency. The differential and frequency relays are definite time relays.

The distance relays have different zones of operation and trip times.

Fig. 3 shows time coordination between different zones of a distance relay and its associated CBs. The logic starts with the first distance relay and calculates measured impedance $Z$ from corresponding entrance in $V_{121}$ and $I_{121}$. First, it checks if $Z$ comes inside Zone 1 impedance circle. If true, it trips the relay instantaneously updating Relay Trip Signal (RTS) to 1 and starts the CB timer. If false, it checks if $Z$ comes inside Zone 2 impedance circle. If it comes in this zone and the relay has already detected fault in Zone 3 (RTS=5) or not detected at all (RTS=0) in previous time step, the relay updates RTS to 4 i.e. in Zone 2. If the relay has already detected fault in Zone 2, it increments the relay timer $T_R$ with one time step and checks if it is equal to Zone 2 timer $t_{Z2}$. If $T_R \neq t_{Z2}$, it trips the relay in Zone 2, updates RTS=2 and starts the CB timer. If $Z$ does not come inside Zone 2 as well, it is checked for Zone 3. If the relay has already detected a Zone 2 fault or nothing in previous time loop then RTS is updated to 5, i.e. detects fault in Zone 3. If the relay has already detected a Zone 3 fault then it simply increments $T_R$ with one time step. If $T_R$ reaches Zone 3 timer then RTS is updated to 3, i.e. relay is tripped and the CB timer starts. If $Z$ is out of all the impedance circles at any time loop, the CB timer, relay timer and RTS are set to 0. This part of the logic is depicted in the left portion of Fig. 3 i.e. Relay Operation. This part is identical for both the Y-Bus and Sparse Tableau based approaches. In Y-Bus approach each relay is considered to have only two hypothetical CBs with a common timer to open/close at both ends. An average delay is added to account for the CB opening time before modifying the Y-Bus. Timing coordination program is trivial in Y-Bus approach.

In the proposed approach we explicitly model the CB operations and detailed station configurations. The CB operation is achieved in the network matrix ($M_{STF}$) by changing the

---

**Fig. 3:** Simulation Time Coordination between different Zones of Distance Relay and associated Circuit Breakers

23rd Power Systems Computation Conference

Paris, France — June 4 – 7, 2024

PSCC 2024
corresponding $F_c$ and $F_i$ entries. The time coordination is essential between transient simulation time step, relay operating time and their associated CBs operation time. The right part of Fig. 3 i.e. Circuit Breaker Operation gives the flow chart for the proposed approach. In Fig.3, while the Relay Trip Signal (RTS) reaches 1/2/3 values, the next step is to open the associated CBs. The program must keep a vector of associated CBs ($R_{21L}(i_{rela}),CB(n))$ with each relay and the total number of CBs to be tripped ($R_{21L}(i_{rela}),CB_{total}$) based on the station configuration. For a BAH scheme, a minimum of 2 CBs must be tripped for each relay, and a pilot trip command must be used to open 2 CBs at the other end of the station. As soon as the CB timer for associated $i^{th}$ CB to relay i.e. $T(i_{CB})$ starts to increment due to relay tripping in any of the zones, the relay condition $Z < Z_{1/2/3}$ does not matter. The CB statuses are updated to open CB as soon as the CB timer saturates ($T(i_{CB}) = t_{CB}$) and RTS is updated to 6. If tripped in Zone 1, CBs are opened for both sides of the line (pilot protection) otherwise CBs are opened for one side of the line. This process is repeated for all the distance relays in the system in a time loop incremented with the simulation time step. The Circuit Breaker Operation block will remain same for definite time relays with $R_{21L}$ changed to the respective relay array. Meanwhile, the Relay Operation block will have a single zone of operation.

IV. RESULTS

This section shows various scenarios that can be simulated using the proposed framework with relays and CBs in TASA of power system networks. WECC 3 generator 9 bus system is employed to validate the results of the proposed framework with commercial software PSS®E 34. After that, the proposed framework is applied to the New England 10 generator 39 bus system and the Polish 327 generator 2383 bus system. Fault conditions with the normal operation of CBs are compared with breaker failure operations after relay tripping. Since BAH is the most popular configuration used for transmission substations, each bus in the system is expanded to a BAH configuration. In simulations, we use a Zone 2 timer of 0.26667s and Zone 3 timer of 0.5s and Zone 1 as instantaneous. We also used a fixed time delay of 0.05s (3 cycles) to account for the finite CB opening time after the trip command is issued.

For each case, the substation diagram for the concerned substations are provided with relay location assigned by algorithms discussed in Section III-A. In substation diagrams, differential relays are denoted by $D_f$ and distance relays are denoted by $D_s$. To keep the diagrams simple and understandable the relays that are not operating in fault case are not shown (transformer differential, under-over frequency for loads and generators). The neighbor substations are shown as buses. Bus fault and line fault are created and noteworthy relays and CB currents are plotted. In all the cases fault impedance $Z_f$ is taken as 1e-4 pu.

Table VI shows the attributes for distance and load relays selected in PSS®E. The settings are calculated for these attributes based on [20] and relays are populated manually in PSS®E. Only one shed point at a frequency of 59 Hz is selected for load relays. Since PSS®E does not facilitate differential relays, the differential relay operation is emulated in PSS®E by self clearing fault. PSS®E does not calculate settings instead it takes them as input. Hence, the settings are calculated via a separate program and given as input to both PSS®E and the proposed approach (PA).

Table VII shows the total number of buses, nodes, CBs, relays and the associated timers to be coordinated for the test systems considered. In the Y-Bus based approach since there are no CBs associated with the model, the number of timers will be the same as the number of relays. It can be observed from the Table that, due to an increase in the intermediate nodes, physical buses, and CB elements the number of timers to be coordinated also increases with system size significantly in NB models. The proposed algorithms automatically populate, maintain, and update protective relays, CBs, and timers in the transient stability simulations.

### A. WECC 9 bus system: Bus Fault (Fig. 4)

First, we discuss fault creation with PA. A 3φ balanced fault is created at bus $B_{6a}$ at 0.1s. The relay protecting $B_{6a}$ is $D_f$ and the associated CBs are $CB_{21}$, $CB_{24}$. $D_f$ trips instantaneously. $D_s$, $D_{s3}$ detect Zone 2 fault. At 0.15s, fault is cleared by opening the CBs $CB_{21}$, $CB_{24}$. Fig. 4b (top) shows the currents for $CB_{21}$, $CB_{22}$, $CB_{24}$ went up during fault duration. After $CB_{21}$, $CB_{24}$ are open, the current through them becomes 0, and the current through $CB_{22}$ comes to a post-fault value. Now, to create the same conditions in PSS®E, a bus fault is created at bus 6 with same

<table>
<thead>
<tr>
<th>Relay</th>
<th>Characteristic</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance</td>
<td>Mho</td>
<td>Zone 1/2/3 pick-up time, Zone 1/2/3 reach, Zone 1/2/3 center line angle, Zone 1/2/3 centre distance, self/transfer trip breaker time, self/transfer trip reclosure time</td>
</tr>
<tr>
<td>Load</td>
<td>Under-frequency load shed</td>
<td>first load shed point (Hz), first point pick up time, first fraction of load shed, Breaker time</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>Buses</th>
<th>Nodes</th>
<th>CBs</th>
<th>CB Timers</th>
<th>Distance Relays</th>
<th>Distance Relay Timers</th>
<th>Differential Relays</th>
<th>U/OFLS Relays</th>
<th>U/OFLS Relay Timers</th>
</tr>
</thead>
<tbody>
<tr>
<td>WECC 9Bus</td>
<td>18</td>
<td>48</td>
<td>45</td>
<td>45</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>New England</td>
<td>78</td>
<td>218</td>
<td>210</td>
<td>210</td>
<td>68</td>
<td>68</td>
<td>78</td>
<td>29</td>
<td>29</td>
</tr>
<tr>
<td>Polish</td>
<td>4766</td>
<td>14302</td>
<td>14304</td>
<td>14304</td>
<td>5452</td>
<td>5452</td>
<td>4766</td>
<td>2245</td>
<td>2245</td>
</tr>
</tbody>
</table>
A 3ϕ balanced fault is created at one end of Line 5 near station 5 at 0.1 s. $D_{s8}$ picks up in Zone 1 and $D_{s9}, D_{s10}$ are picked up in Zone 2. $D_{s8}$ trips instantaneously and sends a pilot trip command to $D_{s9}$ opening $CB_{15}, CB_{16}, CB_{30}, CB_{31}$ at 0.15 s. The currents for CBs can be seen from Fig. 5 (top). As soon as CBs are opened and Line 5 is tripped, $D_{s10}$ gets reset. Fig. 5 (bottom) shows relay currents for $D_{s8}, D_{s9}, D_{s10}$ with the PA as well as PSS®E. The current seen by $D_{s8}, D_{s9}$ becomes 0 while $D_{s10}$ comes to a post fault value at 0.151 s. The similarity in waveforms of PA and PSS®E from the figures confirms the accuracy of PA. The time required for 10 s of simulation with PSS®E is $\sim 0.068$ s and that with PA is $\sim 0.27$ s. The runtime of PSS®E is less as it solves the network via Y-Bus not considering CBs whereas PA solves it with Sparse Tableau considering all the CBs.

With same fault conditions, one of the CBs, say, $CB_{75}$ gets stuck while operating. In that case, $D_{s41}, D_{s42}, D_{s46}$ trip in Zone 2 at 0.368 s; The associated CBs in Station 18 i.e. $CB_{76}, CB_{77}, CB_{78}$ are opened at 0.481 s tripping $L_{21}, L_{22}, L_{25}$. As soon as Line 22 is tripped $D_{s46}$ gets

C. New England system: Bus Fault (Fig. 6)

A 3ϕ balanced fault is created at bus $B_{18a}$ at 0.1 s. The relay protecting $B_{18a}$ is $D_{f34}$ and the associated CBs are $CB_{75}, CB_{78}$. $D_{f34}$ trips instantaneously. $D_{s41}, D_{s42}, D_{s48}$ detect Zone 2 fault and $D_{s46}$ detects Zone 3 fault. At 0.15 s the CBs associated to $D_{f34}$ are opened and $B_{18a}$ is isolated. $D_{s41}, D_{s42}, D_{s46}, D_{s48}$ reset at 0.151 s. The system returns back to original. This scenario is called bus splitting. The CB and relay current waveforms are shown in Fig. 6b.

With same fault conditions, one of the CBs, say, $CB_{75}$ gets stuck while operating. In that case, $D_{s41}, D_{s42}, D_{s48}$ trip in Zone 2 at 0.368 s; The associated CBs in Station 18 i.e. $CB_{76}, CB_{77}, CB_{78}$ are opened at 0.481 s tripping $L_{21}, L_{22}, L_{25}$. As soon as Line 22 is tripped $D_{s46}$ gets

D. Conclusion

The proposed approach has shown promising results in terms of accuracy and speed. The comparison with commercial software like PSS®E and PA has shown that the approach is capable of handling complex scenarios like bus splitting and CB stuck. The use of sparse tableau and Y-bus network models in PA and PSS®E, respectively, has shown that PA requires less runtime while maintaining accuracy.

E. Future Work

Further research is needed to extend the approach to handle more complex scenarios like power flow restoration after a fault and the integration of renewable energy sources. The integration of artificial intelligence techniques like machine learning and deep learning can also enhance the accuracy and speed of the approach.
reset. The stuck breaker caused $N-3$ contingency (excluding $Ld6$).

E. Polish system: Bus Fault (Fig. 8)

A 3φ balanced fault is created at bus $B5a$ at 0.1s. The relay protecting $B5a$ is $Df8$ and the associated CBs are $CB24, CB27, CB30$. $Df8$ trips instantaneously. $Ds16, Ds18$ detect Zone 2 fault and $Ds681$ detects Zone 3 fault. At 0.15s the CBs associated to $Df8$ are opened and $B5a$ is isolated. $Ds16, Ds18$ and $Ds681$ reset at 0.151s. Fig. 8b shows the current waveform of CBs and relays for the scenario. Fig. 8c shows the waveforms when $CB24$ gets stuck while operating. $Ds16, Ds18$ trip in Zone 2 at 0.368s and open $CB25, CB26$ at 0.418s. Hence, the current through $CB24$ gets reduced at 0.418s but the fault is still being fed via $T8, T9$. $Ds681$ trips at 0.601s in Zone 3, tripping $Line341$ at 0.651s, further reducing current through $CB24$. $Ds879$ trips in Zone 3 at 1.153s, tripping $Line440$ at 1.203s. Note that only $CB27, CB30$ are opened in Station 5 so the fault is still getting fed via $CB24$ from $B6$.

D. New England system: Line Fault (Fig. 7)

A 3φ balanced mid-line fault is created at $Line10$ at 0.1s. $Ds18, Ds19$ trip instantaneously in Zone 1 at 0.1s. $Ds20$ and $Ds22$ detect Zone 2 faults. The CBs associated with $Ds18$ i.e. $CB135, CB136$ and $Ds19$ i.e. $CB141, CB142$ open at 0.15s. This isolates $Line10$ from the system and the power flowing through $Line10$ now transfers to $Line11$. So generator current at $B9$ flows through $CB143$ to $Line11$. At 1.167s, $Ds21$ detects Zone 2 fault and trips at 1.24s. $CB143$ opens at 1.29s isolating $Line11$. This isolates Station 29 from the system.

Another case is considered where $CB136$ gets stuck while operating. This makes $Ds22$ trip in Zone 2 followed by tripping of $Ds21$ isolating Station 29. But in this case the isolation happened much earlier than the normal operation with an increased contingency. The waveforms and timings can be seen from Fig. 7c.

F. Polish system: Line Fault (Fig. 9)

A 3φ balanced mid-line fault is created at $Line18$ at 0.1s. $Ds34, Ds35$ detect faults in Zone 1 and trip instantaneously. $Ds7, Ds17, Ds28, Ds30, Ds32, Ds36$ detect Zone 3 faults. The CBs associated with $Ds34, Ds35$, i.e. $CB49, CB50$ and $CB165, CB166$ open at 0.15s,
resetting all the other relays. Fig. 9b shows waveforms for the case.

Fig. 9c shows currents for a case where CB49 gets stuck. All the other relays detecting fault in Zone3 trip at 0.601s, opening associated CBs in their station at 0.651s. While all the lines connected to Station 7 are tripped, T12 still connects B7a to B322. This is still feeding current to fault via CB49.

From the above result cases it is evident that incorporating protective relays and CBs in TSA plays an important role in TSA. The CBs, when mal-operate, can result in a higher level of contingency than that in normal operations. It can also result in sustained faults in system which may lead to cascade events. With proposed framework, it is very convenient to represent the system in NB model with all the CBs, automatically place relays for each element and get associated CBs and trip any element of the system by opening the CBs. This can further help to perform substation level analyses and cascading analyses.

The simulation was carried out with a cpp code on Intel® Xeon® Silver 4110 CPU @ 2.10GHz 32GB RAM. Table VIII shows the break-up of time required for various functions for each time loop. The symbolic and numeric factorization of $M_{STF}$ takes place before the time loop starts or when $M_{STF}$ needs to be refactorized. The other functions in Table VIII are called in every time loop hence, contribute to the total simulation time. The majority of time is taken for backward/forward substitution to solve $M_{STF}X = I$. This is due to increased size of Sparse Tableau i.e. $M_{STF}$. The computational enhancement of Sparse Tableau is discussed in [22]. However, this paper does not focus on this. The time taken for relay computations are also provided. The time taken for a 10s simulation of the Polish system using the NB model with relays is 98s, whereas, for the BB model without any relays, it is 19.2s regarding the same code setup. Although the simulation time with proposed approach is higher, it provides extreme flexibility for the operators to analyze the system. Parallelization approaches can be adopted to reduce the timings of the simulation further.

G. Parallel Processing Capability of Proposed Framework

Multiple cores can be employed to break down the proposed framework methodology to make full use of modern supercomputers. Since the placement of relays is performed only once before simulation, parallelization of the same will not affect much. Still, it could be done by assigning one type of relay placement to one CPU. The significant time reduction can be achieved by reducing the time of each loop. The Protection Devices Module can be decoupled from the differential and algebraic equations for transient simulation. Input vector $X$ can be used from the previous time step to process the Protection Device Module in parallel. The drawback with this approach would be a lag of one time step in Protection Module action. However, it can be neglected as the expected nature of the waveform for TSA will remain the same, and all the timers will still be synchronized. With this modification, the extra computation time required for relays can be removed entirely from the simulation. The simulation time for New England system can be reduced by half and for Polish system by 34%. Also, parallel methods can be explored to reduce the forward/backward substitution, as it is the most time consuming part of the time loop.

<table>
<thead>
<tr>
<th>Function</th>
<th>39 Bus</th>
<th>2383 Bus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbolic factorization</td>
<td>0.4</td>
<td>7.15</td>
</tr>
<tr>
<td>Numeric factorization</td>
<td>0.6</td>
<td>36.157</td>
</tr>
<tr>
<td>Forward/Backward substitution</td>
<td>0.17208</td>
<td>6.531</td>
</tr>
<tr>
<td>Differential relays computation</td>
<td>0.1428</td>
<td>1.94</td>
</tr>
<tr>
<td>Distance relays computation</td>
<td>0.035</td>
<td>0.32</td>
</tr>
<tr>
<td>Frequency relays computation</td>
<td>0.00428</td>
<td>0.0595</td>
</tr>
<tr>
<td>Total time for relays</td>
<td>0.182</td>
<td>2.3195</td>
</tr>
<tr>
<td>Differential Equation solver</td>
<td>0.0102</td>
<td>0.11712</td>
</tr>
<tr>
<td>States update</td>
<td>0.0013</td>
<td>0.0102</td>
</tr>
<tr>
<td>Total time (10s)</td>
<td>1.16s</td>
<td>98s</td>
</tr>
</tbody>
</table>

Fig. 9: Polish system - Line fault at Line 18 (a) Station Configuration (b) Normal operation (c) CB49 stuck
V. CONCLUSION

This paper provides a simulation framework to include protective relays and CBs in transient stability assessment. The Sparse Tableau Approach is used to represent the system in Node-Breaker model. Algorithms are derived for automatic placement of differential, distance and under/over-frequency relays in the system and CBs associated to them. The timer logic implemented for coordinating different zone timers of relays in the system and CBs associated to them. The timer placement of differential, distance and under/over-frequency relays in the system and CBs associated to them.

REFERENCES


