World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:7, No:8, 2013

Steady State Simulation of Power Systems with Change in Topology

Aidil Azwin Zainul Abidin, Farrukh Hafiz Nagi, Agileswari K. Ramasamy, and Izham Zainal Abidin

Abstract**—**In power system protection, the need to know the load current together with the fault level detected by a relay is important. This is due to the fact that the relay is required to isolate the equipment being protected if a fault is present and keep the breaker associated with it closed if the current level is lower than the maximum load level. This is not an issue for a radial system. This is not the same however in a looped power system. In a looped power system, the isolation of an equipment system will contribute to a topology change. The change in the power system topology will then influence or change the maximum load current and the fault level detected by each relay. In this paper, a method of data collection for changing topology using Matlab and Sim-power will be presented. The method will take into consideration the change in topology and collect data for each possible topology.

Keywords**—**Topology Change, Power System Protection, Power System simulation, Matlab, Sim-power.

I. Introduction

THE need to know the load current and the minimum fault current that flows through a relay is important in setting the relay. Studies have been made by past researchers that require this information [1], [2]. In a radial system, this is not an issue due to the fact that the current can only flow in one direction and the problem of topology change does not occur. This is not the same however for a looped power system where the current can flow in both directions through a relay and topology change can take place [3]. In an event of a topology change, the maximum load current and the minimum fault current that is detected by the relays in the system will change [4]. This will cause coordination problems for the relays in the system. In this paper, using Matlab and Simulink, a data collection algorithm for power systems with changing topology will be presented. The algorithm will automatically change the topology, run a steady state analysis and store the current value of the relay for every possible topology that the system could have.

II. Algorithm

In this paper, the change in topology will be viewed as isolation of lines. If a system has a number of lines, then isolating one or more of the lines will be considered changing the topology of the system. The algorithm for this paper is shown below:

1. Identify the lines which exist in the system.
2. Identify breakers associated with each line in the system.
3. Identify the maximum number of lines than can be isolated in the system.(N)
4. Initialize i=1
5. Identify the possible isolation combination that could be done in the system for I lines isolated (M).
6. Initialize j=1
7. Perform isolation for combination j.
8. Perform steady state analysis for the system intact
9. Keep the data obtained in (8) in a structure for each relay.
10. Normalize the system such that all the lines are intact.
11. Increment j. If j>M, increment I, if i<N repeat starting from (5) if not terminate process. If j<M then repeat starting from (7).

The first step in the algorithm will have to be done manually. This is done by naming all the lines in the Simulink diagram. The second step is also done manually; the best practice is to have the names of the breakers in some way associated with the names of the lines connected to it. The 3rd step depends on how large the system is. The more lines available, the more the number of lines can be isolated in order to provide power to the load. This can be explained using an example. If a system has 5 lines, then isolating 4 lines would be pointless because the one line will not be of use in transferring power to a load from a source. In 5, the procedure is done using the *nchoosek* command in Matlab. The *nchoosek* command will list down all the possible combinations of choosing k elements out of the N elements in a set where N is the total elements in the set. In this command k is an integer given by the user. The flowchart for performing step 7 is shown in Fig. 1

Aidil Azwin is a senior lecturer in the Department of Electrical Engineering of University Tenaga Nasional (e-mail: aidilazwin@uniten.edu.my).

Farrukh Hafiz Nagi is an associate professor in the Department of Mechanical Engineering of University Tenaga Nasional.

Agileswari K. Ramasamy is an associate professor in the Department of Electrical Engineering of University Tenaga Nasional.

Izham Zainal Abidin is an associate professor in the Post Graduate Studies College for Universiti Tenaga Nasional.

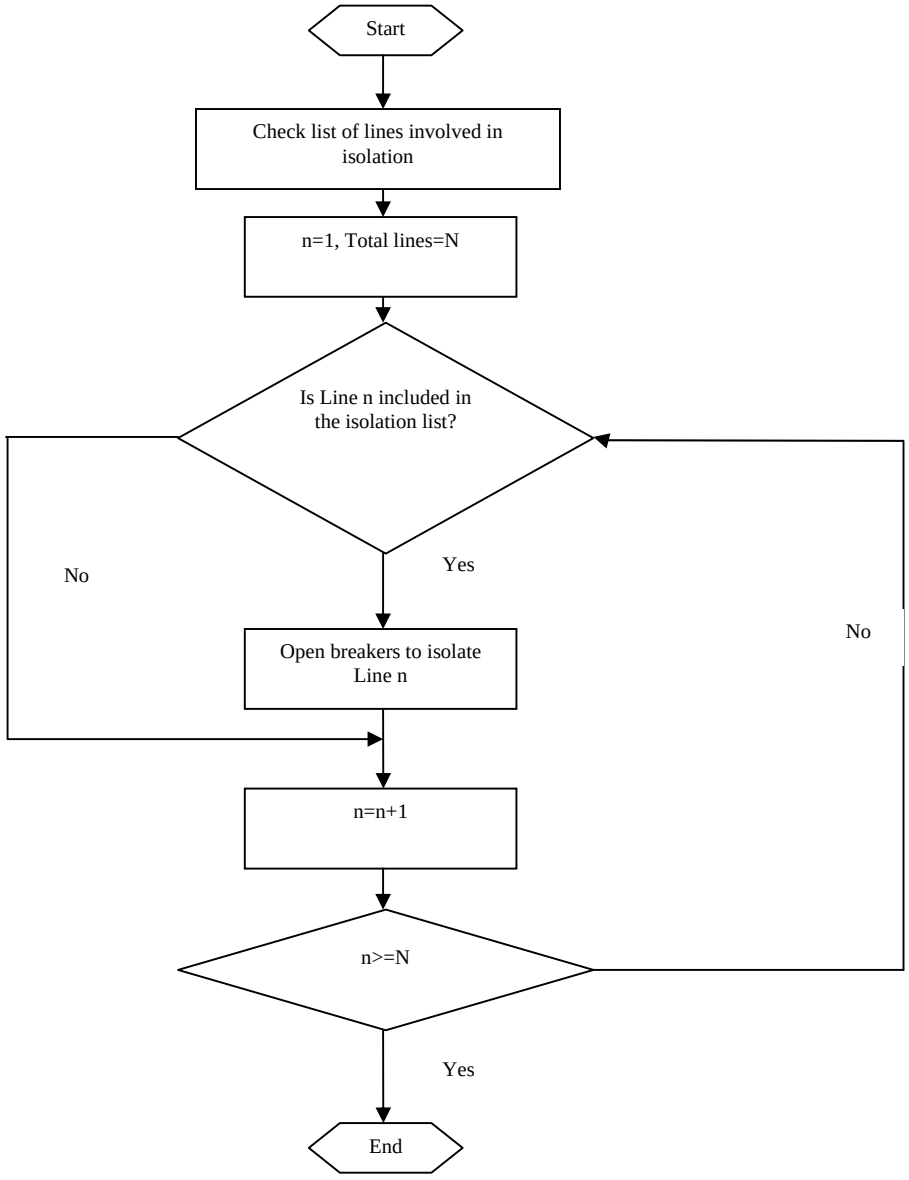

Fig. 1 Flowchart for program used to perform isolation of lines in the power system tested

Step 8 of the algorithm is implemented using the *power analyze* command in Matlab. The *power analyze* function will produce an output in a form of a structure which consist of the states of the power system, the inputs that are in the power system, the initial conditions of the power system and the steady state output of the power system. The names of each state and also outputs are also included in the structure produced. The current and voltage values which will be read by the relay in place needs to be extracted from this structure. The steady state output values are a vector of voltage and current values of all the measurement devices and circuit breaker voltages. For setting a relay, the voltage and current value corresponding to the relay needs to be known. This can be done by finding the correct index of this value in the vector. The name of the measurement device corresponding to a relay can be found in the vector containing the names of the outputs. The index of the name in this vector will be the same as the index of the voltage or current in the output steady state vector.

In step 9, a structure will be formed for each and every relay in the system. The readings for each relay will be put in these structures. Step 10 is important in order to ensure that the power system is put back in the original condition before the next topology can be simulated. This is due to the fact that in step 11, the next topology that needs to be simulated is taken from the list of possibilities that was generated by the *nchoosek* command in step 5.

III. Example of Implementation

The example given in this paper to illustrate this technique is a four bus system where 2 bus is the voltage source and another 2 bus is a static load. There are three lines which are connected between them. The lines are connected in such a

way that a minimum amount of load will lose power if one line goes off. In this example, the load current detected by each current transformer for situations where no lines were isolated to situations where two lines were isolated will be presented. The diagram for this system is shown in Fig. 2.

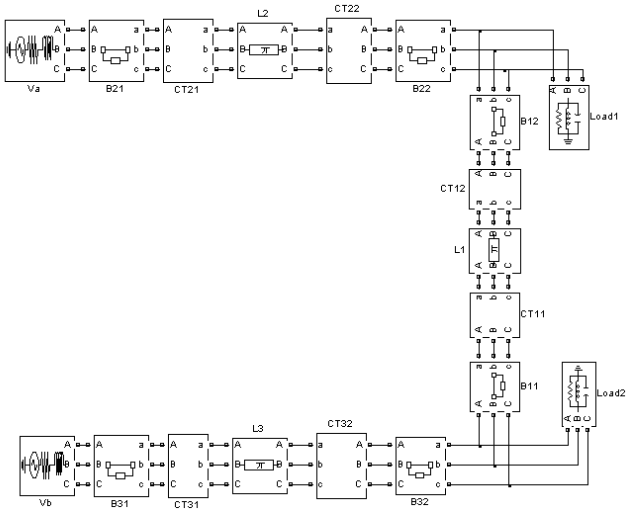

Fig. 2 Diagram of power system under study

In the example, 7 possibilities exist. This includes no lines out of service, 3 cases of one line out of service and three cases of two lines out of service. The first thing that was done is the naming of the lines. Each lines were given the name *L1*, *L2*, and *L3*. Corresponding to *L1*, there are two breakers which are *B11* and *B12*. Each of these breakers has two CTs which measure current namely *CT11* and *CT12*. The same can be seen in the diagram for the other two lines. The data which is relevant in this algorithm is the current detected by all the CT which is present in the system.

The first info that needs to be known is the location of these currents in the *power_analyze* steady state output structure. If a variable were to be assigned to the power_analyze command, the variable would then be a structure containing the steady state inputs and outputs. The location of these currents needs to be found in this structure. From the names of the output there are 54 outputs which include the voltages and currents of the measurement devices together with the voltages of the circuit breakers in the system. In this paper, the relay currents are of concern. Therefore from the output structure, it is found that there are 36 elements which are of interest. The index of these elements is then assigned to a structure which will gather all the current values for every possible topology.

The next information required is possible topology that the power system could have. The Line consists of three lines and the worst case scenario is just having one line energized. Therefore the possible conditions that exist are as follows:

1. No line isolated.
2. L1 isolated
3. L2 isolated
4. L3 isolated.
5. L1 and L2 isolated
6. L2 and L3 isolated
7. L1 and L3 isolated.

This can be done using the *nchoosek* command by defining a vector of [L1,L2,L3] as the set. The script for this example is shown below:

```
B4.Result(1).name='CT11'%structure for the relay using CT11
B4.Result(2).name='CT12'%structure for the relay using CT12
B4.Result(3).name='CT21'%structure for the relay using CT21
B4.Result(4).name='CT22'%structure for the relay using CT22
B4.Result(5).name='CT31'%structure for the relay using CT31
B4.Result(6).name='CT32'%structure for the relay using CT32
k=1
l=1
for c=0:1:2%Number of lines to be isolated
   m={'L1','L2','L3'}%Possible lines that can be isolated.
   N=nchoosek(m,c)%Command to find the possible combination of lines to be isolated.(Step 5 of algorithm)
B4.LineOut(l).lines=n%Putting possibilities in a structure.
Iflength(n)==0;%Simulation of non isolated topology.
Sps=power_analyze('bus4','structure')%step 8 of the algorithm
B4.Result(1).Data(:,k)=sps.yss(37:39)%extraction of current values of all phases for CT11(step 9)
B4.Result(2).Data(:,k)=sps.yss(40:42)%extraction of current values of all phases for CT12(step 9)
B4.Result(3).Data(:,k)=sps.yss(43:45)%extraction of current values of all phases for CT21(step 9)
B4.Result(4).Data(:,k)=sps.yss(46:48)%extraction of current values of all phases for CT22(step 9)
B4.Result(5).Data(:,k)=sps.yss(49:51)%extraction of current values of all phases for CT31(step 9)
B4.Result(6).Data(:,k)=sps.yss(52:54)%extraction of current values of all phases for CT32(step 9)
      k=k+1;
else%Simulation of isolated topology
for i=1:1:length(n)%Loop to execute step 11 of the algorithm.
      B=n(I,☺)
for j=1:1:length(b)%Loop to execute step 7 of the algorithm
        a=char(b(j))
isolate%Function used to execute isolation of lines
end
sps=power_analyze('bus4','structure')%step 8 of the algorithm
B4.Result(1).Data(:,k)=sps.yss(37:39)%extraction of current values of all phases for CT11(step 9)
B4.Result(2).Data(:,k)=sps.yss(40:42)%extraction of current values of all phases for CT12(step 9)
B4.Result(3).Data(:,k)=sps.yss(43:45)%extraction of current values of all phases for CT21(step 9)
B4.Result(4).Data(:,k)=sps.yss(46:48)%extraction of current values of all phases for CT22(step 9)
```

```
B4.Result(5).Data(:,k)=sps.yss(49:51)%extraction of current values of all phases for CT31(step 9)
B4.Result(6).Data(:,k)=sps.yss(52:54)%extraction of current values of all phases for CT32(step 9)
      k=k+1;
for j=1:1:length(b)%Loop to execute step 10 of the algorithm.
        A=char(b(j))
normalize%function used to normalize the system back to full topology
end
end
   l=l+1
end
end
```

In the program above, there are two other functions which were written which are *isolate* and *normalize*. The program for isolate is shown below:

```
if a==’L1’%Isolation of L1
set_param(‘bus4/B11’,’InitialState’,’Open’)%Opening breaker B11
set_param(‘bus4/B12’,’InitialState’,’Open’)%Opening breaker B12
elseif a==’L2’%Isolation of L2
set_param(‘bus4/B21’,’InitialState’,’Open’)%Opening breaker B21
set_param(‘bus4/B22’,’InitialState’,’Open’)%Opening breaker B22
elseif a==’L3’%Isolation of L3
set_param(‘bus4/B31’,’InitialState’,’Open’)%Opening breaker B31
set_param(‘bus4/B32’,’InitialState’,’Open’)%Opening breaker B32.
Else
end
```

For the normalization process the source code is shown below:

```
if a==’L1’%Normalize L1
set_param(‘bus4/B11’,’InitialState’,’Close’)%Closing breaker B11
set_param(‘bus4/B12’,’InitialState’,’Close’)%Closing breaker B12
elseif a==’L2’%Normalize L2
set_param(‘bus4/B21’,’InitialState’,’Close’)%Closing breaker B21
set_param(‘bus4/B22’,’InitialState’,’Close’)%Closing breaker B22
elseif a==’L3’%Normalize L3
set_param(‘bus4/B31’,’InitialState’,’Close’)%Closing breaker B31
set_param(‘bus4/B32’,’InitialState’,’Close’)%Closing breaker B32.
Else
End
```

The results of the program which was done for this example are shown in Table I.

III. Conclusion

From the program that was built and the results obtained, it is found that it is possible to automate the data collection of load currents and fault currents for a power system with changing topology using matlab and Simulink. With the algorithm presented in this paper, the setting of relays for changing topologies can be executed.

TABLE I
RESULTS OF CURRENTS FOR ALL RELAYS IN THE POWER SYSTEM SHOWN IN FIG. 1

Lines Isolated	RELAYS					
	CT11	CT12	CT21	CT22	CT31	CT32
None	1.79E-01	1.79E-01	7.40E+01	7.40E+01	7.40E+01	7.40E+01
L1	8.94E-03	8.94E-03	7.40E+01	7.40E+01	7.40E+01	7.40E+01
L2	7.36E+01	7.36E+01	8.99E-03	8.87E-03	1.47E+02	1.47E+02
L3	7.36E+01	7.36E+01	1.47E+02	1.47E+02	8.99E-03	8.87E-03
L1 and L2	8.95E-03	2.23E-04	8.97E-03	2.24E-04	7.40E+01	7.40E+01
L1 and L3	2.23E-04	8.95E-03	7.40E+01	7.40E+01	8.97E-03	2.24E-04
L2 and L3	5.40E-07	5.43E-07	8.97E-03	2.24E-04	8.97E-03	2.24E-04

References

[1] Zhongwei Li; Weiming Tong; Fengge Li; ShenghuFeng, ”Study on Adaptive Protection System of Power Supply and Distribution Line”, International Conference on Power System Technology, 2006. Power Con 2006, pp 1-6

[2] Fang, Y.J.; Xue, Y. ; Xu, J.B. ; Bao, Y.H. ; Mu, H. ; Lei, M. ; Qin, H. ; Zeng, Y. G., ”Implementation of an on-line pre-decision based system protection scheme ",International Conference on Power System Technology, 2002. Proceedings Power Con 2002, vol2, pp823-827.

[3] J. Lewis Blackburn,”Protective Relaying Principles and Applications” Marcel Dekker, Inc. 1998.

[4] Hadi Saadat, “Power System Analysis” Mc Graw Hill, Second Edition, 2004.