Phasor-assisted automated topology processing for state estimators

Conference Paper · August 2013

DOI: 10.1109/EPEC.2013.6802914

CITATIONS
6

READS
38

2 authors:

Mostafa Farrokhabadi
University of Waterloo
20 PUBLICATIONS **774** CITATIONS
SEE PROFILE

Luigi Vanfretti
Rensselaer Polytechnic Institute
319 PUBLICATIONS **3,654** CITATIONS
SEE PROFILE

Phasor-Assisted Automated Topology Processing for State Estimators

Mostafa Farrokhabadi, *Student Member, IEEE,* and Luigi Vanfretti, *Member, IEEE*

***Abstract*— This article provides a robust network topology processor for state estimators that utilizes both conventional and/or PMU measurements. The emphasis is on the involvement of PMU measurements in topology processing. Building on top of the state of the art, an algorithm is proposed in a way to cover the limitations of current approaches and at the same time to suggest new features. The topology processor was coded in MATLAB and is tested using a modified version of the IEEE Reliability Test System 1996. The topology processor is intended to provide network topologies to PMU-only sate estimators, so the test system is simulated in real-time using the eMegaSim Opal-RT real-time simulator that generates synthetic data mimicking a real PMU. Different test scenarios are carried out and the topology processor efficiency and robustness is verified by the test results. Specifically, it is testified through a discussion that the proposed method is fast enough to support PMU-only state estimators, as well as conventional or phasor-assisted state estimators.**

***Index Terms*— State Estimator, Topology Processor, Real-Time Simulation, Phasor Measurement Unit**

I. Introduction

Topology processing is a crucial step for EMS applications –particularly state estimators- to properly function. However, the openly available documented evidence on the implementation of the current topology processors (TP) is limited and associated researches shows a lack of rigor in the description of the their algorithms. That makes a practical implementation by an independent party to be very difficult.

Sasson *et al.* [2] proposes a general algorithm to carry out topology processing. Although it is the corner stone for several other works, it is exposed to several drawbacks. For example, for the algorithm to function properly, it should determine the topology for all the substations that any of their breakers have a different status compared to a pre-defined initial statuses; instead, it is more efficient to define the topology just for those substations which have suffered changes compared to previous measurements cycle. In addition, [2] may show inconsistency when it comes to splitting/merging incidents due to the lack of rigorous number assignation algorithm for the nodes. [3] uses the same algorithm as [2] with the difference that it puts forward amethod to track the changes in the system topology; this tracking is then exploited for matrix de-factorization to save in computation time for EMS applications. However, [3] is exposed to the same problems of [2] as the algorithm to define the topology remains intact. [5] make uses of the graph theory for topology processing. However, the information provided regarding the technical side of the algorithm is vague, and there is no detail testing results. Other related works along with those discussed here are scrutinized in details in [11].

Moreover, since the introduction of PMUs, there have been plenty of proposals to utilize phasor measurement data for state estimation [10]. In particular, two novel approaches have been introduced, both of them relying only on PMU data to perform state estimation [1, 6]. However, to the knowledge of the author, [4] is the only work that considers the PMUs for the aim of topology processing. However, the algorithm provided in [4] is a rudimentary one and has overlooked different aspects of a realistic TP. This algorithm is discussed in detail in further sections of the current paper.

Therefore, there is a need for a topology processor that is robust enough to not only function with conventional data but also has the ability of integrating PMU data whenever needed. This article proposes a topology processor that has the ability to work with both conventional data and those received from PMUs. Hence, the network topologies can be provided to PMU-only state estimators such as the ones in [1, 6].

The reminder of this article is organized as follows, in Section II the inputs and outputs of the proposed topology processor algorithm are described, while Section 3 provides a description of the algorithm execution procedure. In Section IV the algorithm is extended to exploit the availability of PMUs, either in combination with conventional measurements or when PMUs are the unique source of the measurements. Section V describes the results of different testing scenarios. Finally, in Section VI, conclusions are drawn and future work is outlined.

II. Automated Topology Processing: Algorithm and Implementation

Generally, the TP must be developed to meet the following requirements:

1) It must support EMS applications that focus on the HV backbone of the network, i.e. transmission networks [1, 6].
2) It must be capable of deriving positive sequence network representations for (balanced) 3 phase switching - as a minimum requirement.

The proposed TP is built upon the work in [2]. However, it has been thoroughly modified in order to provide needed functionalities that: (a) can deal with any type of substation configuration, and (b) to include all functions prevalent in typical topology processors. For the algorithm to execute properly, the following rules should be followed.

A. TP Algorithm Rules

1) Every substation should be assigned a number from 1 to n, i.e. the number of substations.
2) Every switch should be assigned a number from 1 to m, i.e. the number of switches in the system.
3) Every line should be assigned a number from 1 to k. Then, the bus bars in the system are assigned numbers from k+1 onwards.
4) A switch status is either 1 or 0. 1 represents the closed condition, while 0 means that the switch is open.
5) A switch can't be connected to more than two circuits. Here, circuit is defined as either a line or a bus bar. If a switch is connected to more than two circuits, phantom switches should be introduced that are always closed.
6) Switches are categorized into 4 different types:

- *Type 1:* Switches located on a transmission line connecting a generator to a substation.
- *Type 2:* Switches located on a transmission line connecting two different substations.
- *Type 3:* Switches that are part of a substation configuration such as ring, double-bus-double-breaker, etc.
- *Type 4:* Switches located on a line connecting shunt elements to a substation.

B. Inputs and Outputs of the TP

The inputs and outputs for this TP are provided in matrix form. There are two input matrices, the *"Switch Table Matrix"* and the *"Configuration Matrix"*. The later contains information regarding the circuits in the network and their respective measurements, while the former carries the switches and their respective circuit information. Suppose that there are N switches in the power network; the Switch Table matrix would be a $N\times 8$ matrix. Each column holds specific information, defined as follows:

1) Switch Numbers: The first column contains the numbers assigned to the switches from 1 to N.
2) Near Substation Number: refers to the closest substation to the switch. For type 1, 3 and 4 switches it is the substation to which they belong. For switches type 2, the near substation is entered.
3) Far Substation Number: Similarly, for type 1, 3 and 4 switches it is the substation to which they belong. For type 2 switches, the further substation's number is entered.
4) Switch Type.
5) Switch status.
6) Circuit analysis: Based on the type of switches, it determines if the switch forms part of a single circuit (types 1, 2, 4) or if the circuit is part of the substation configuration and is connected to two different circuits (type 3). For type 2 switches, the entry in the 6th column is its corresponding line number and the entry in the 7th column is 0. For types 3 switches the corresponding circuits' numbers are entered in columns 6 and 7. If a circuit is connected to more than one switch, a minus sign is introduced before its number. Finally, for type 1 and 4 switches the entries of both the 6th and 7th columns are 0.
7) Refer to 6 (the order of entries in the 6th and 7th columns is not important for type 3 switches).
8) Original Substation Number: For type 3 switches it is the number of the substation to which they belong (when all the switches are closed). For other types, the entry is zero.

If there are M circuits in the network, i.e. M different lines and buses, the *"Configuration Matrix"* would be an $M\times 11$ matrix, as follows:

1) Column 1: Circuit Numbers - The first column contains circuit numbers from 1 to M.
2) Columns 2, 3, 4, 5, 6, and 8: A circuit can be either a line or a bus bar. In addition, a line may have: (a) no measurements, (b) measurement instruments are available on both terminals and (c) just on one terminal. A measurement may be available or lost. Hence, a single circuit can be categorized into different cases as shown in Fig. 1. The entries of the columns 2 to 8 are directly related to these cases. Columns 2 to 4 carry information regarding the circuits' substations, and columns 5 to 8 carry information regarding the circuits' measurements. It can be seen from Fig. 1 that there are 4 major cases. Additionally, there are two and four sub-classes for cases 3 and 4 respectively. Therefore, the entries of the columns 2 to 8 are determined as follows:

- Case #1: The circuit is a bus bar. The bus bar's station is entered in the 2nd column, and the entries of columns 3 to 8 are zero.

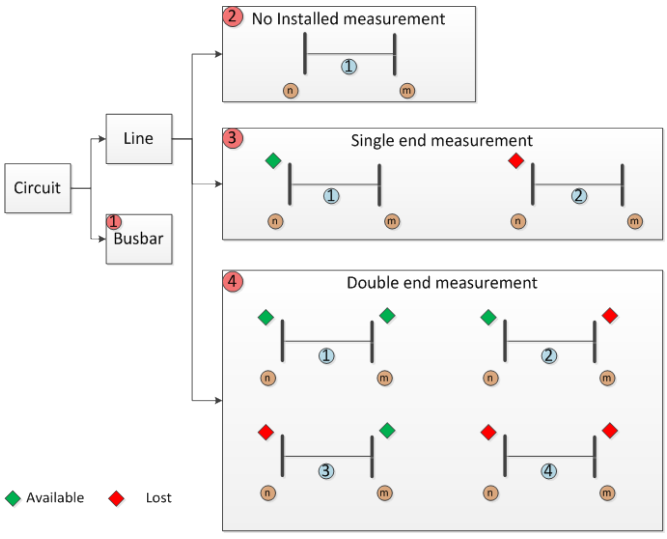

Fig. 1. Different circuit and measurement configurations

- Case #2: The circuit is a line with no installed measurement instruments. The entries of the 2nd and 3rd columns are *n* and *m* (regardless of the order), and the entries of columns 4 to 8 are zero.
- Case #3.1: The circuit is a line with an available measurement at one terminal. The entries of the 2nd and 3rd columns are *n* and *m*.
  The entry of the 4th column is 0, the 5th column is 1, column 6 is zero, column 7 is 1, and column 8 is zero.
- Case #3.2: The circuit is a line with a measurement at one terminal. However, the measurement is lost due to a technical reason. The entries of the 2nd and 3rd columns are *n* and *m*. The entries of the columns 4 to 8 are zero.
- Case #4.1: The circuit is a line with available measurements installed at both ends. The entries of the 2nd and 3rd columns are *n* and *m*. The entry of 4th column is the same as the entry of the 2nd column. The entries of all columns 5 to 8 are 1.
- Case #4.2: The circuit is a line between stations *n* and *m*. The line has installed measurement at both ends, but the one at station *m* is not available due to technical reasons. The entry of the 2nd column is *n*, the entry of 3rd column in *m*, the entry of 4th column is *n*, the entry of the 5th column is 1, the entry of the 6th column is 0, the entry of the 7th column 7 is 1, and the entry of the 8th column is 0.
- Case #4.3: The circuit is a line between stations *n* and *m*. The line has installed measurement at both ends, but the one at station *n* is not available due to technical reasons. The entry of the 2nd column is *n*, the entry of 3rd column in *m*, the entry of 4th column is *n*, the entry of the 5th column is 0, the entry of the 6th column is 1, the entry of the 7th column 7 is 1, and the entry of the 8th column is 0.
- Case #4.4: The circuit is a line with installed measurements at both ends. However, both measurements are lost due to some technical issues. The entries of the 2nd and 3rd columns are *n* and *m*. The entry of the 4th column is the same as the entry of the 2nd column. The entries of the columns 5 to 8 are zero.

3) Column 9: The content of this column is the number assigned to the island to which the substation entry of the 2nd column belongs to.
4) Column 10: The content of this column is the number assigned to the island to which the substation entry of the 3rd column belongs to.
5) Column 11: In a power network there are original substations when all the breakers are closed. However, due to switching actions there may be a moment when original substations are split into two or more stations (nodes). The content of this column is the number assigned to the original substation to which the substation of the 2nd column belongs to (in the case that it is a split node).
6) Column 12: The content of this column is the number assigned to the original substation to which the substation of the 3rd column belongs.

C. *Functions of the TP*

Schematically the TP algorithm has six different functions as shown in Fig. 2:

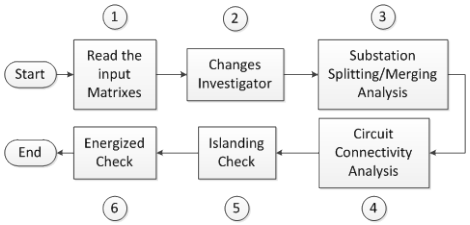

Fig. 2. *Algorithm Flowchart*

1) *Read the Input Matrix*

The statuses of the all the switches are the inputs in each snapshot.

2) *Changes Investigator*

The switches' statuses are compared with the previous snapshot. For type 3 switches, if any change is identified, both the switch number and its corresponding original substation number are saved. For type 1 or 4 switches, there is no need to track changes as they have no effect on the system topology. For type 2 switches, if any change is identified, the switch's number and its corresponding line number are saved.

The output of this function is a "sub matrix". This matrix reports the substation numbers in which any switching have occurred.

3) *Substation Splitting/Merging Analysis*

The topology engine executes this function only if there is any type 3 switch that faced a status change. Otherwise it bypasses the function.

The topology engine goes through all substations that have suffered changes, sequentially. It searches for open switches within the station. If there are no open switches, i.e. there is just one single closed path within the substation; a list is created with all the substation's circuits.

If any open switch is found, the TP engine saves its number. Then it starts a list with one of the two circuits connected to the open switch. Next, it scrutinizes the "Switch Table Matrix" to add all the other circuits that are connected via closed switches to the one that is already in the list. All the switches' numbers that are through the found paths are saved separately. This procedure is repeated sequentially for all circuits. The whole action is repeated then for the next open breaker until all the open switches are scrutinized.

This function outputs a "List Matrix". Each row depicts a created list, i.e. represents a closed path within a substation. The numbers assigned to circuits of each path are the entries in each row.

If there is more than one list, each contains more than one circuit, and the substation is split. An efficient numbering

algorithm is needed to assign numbers to newly formed nodes. The algorithm therefore outputs a "Substation Number Matrix". If there are L original substations in the system, the matrix dimension is $L \times K$. Each row represents an original substation, and the entries of each row are the numbers assigned to the nodes originated from the corresponding original substation.

The first step of the numbering algorithm is to build a vector of empty entries termed "Possibility Vector". It is a vector from 1 to m (a number that is sufficiently greater than the total number of original stations, usually 3 times larger). When a station splits, the corresponding row of the "substation Number Matrix" is scrutinized to find out the numbers assigned to split stations of the previous snapshot. Afterwards, those numbers are replaced by 0 in the Possibility Vector (which means these numbers are released to be assigned in further steps).

After the Possibility Vector is modified, the newly formed nodes are assigned a number from the next available number onwards.

The ultimate output of this function is a matrix called "Heart Matrix". The dimension of this matrix depends on the changes that have occurred in the system. Hence, the matrix is described through an example. Suppose that in a system with 10 substations, the 3rd and 7th stations are exposed to changes in their breakers' statuses. The changes result in three separate closed paths within the 3rd station, and two separate closed paths in the 7th. The circuit numbers that belong to each path in substation 3 are (4, 5), (8, 10, 14, 16), and (13); the circuit numbers that belong to each path in substation 7 are (20, 23, 27, 28), and (30, 31, 34, 36, 40, 41). The corresponding Heart Matrix will be formed:

3	3	2	4	1	4	5	8	1	1	1	1	0	0
								0	4	6	3		
7	2	4	6	2	2	2	2	3	3	3	3	4	4
				0	3	7	8	0	1	4	6	0	1

The first column shows the station with changes. The second column shows the number of separate closed paths in each station. For station 3, columns 4, 5 and 6 show the number of circuits located in each of the station's (three) closed paths. For station 7, columns 4 and 5 show the number of each closed path circuits. The other columns display the circuits that belong to each closed path. For example, in station 3 the first closed path has 4 circuits, so columns 7, 8, 9, and 10 carry the circuits' number that belong to this closed path.

4) Circuit Connectivity Analysis

The task performed in two different functions. One function is embedded within the Splitting/Merging analysis. This is executed when the list matrix is scrutinized to see if any splitting has occurred. If a list that consists of single circuits is found, the TP engine marks the circuit as disconnected.

The other function performing this task is independent and investigates type 2 switches. If a switch's status is 0, the topology engine finds its corresponding line through a search in the Switch Table Matrix, and marks it as disconnected. If the status has changed from 0 to 1, the engine looks for all the other type 2 switches that are related to the same line. If all of them are closed, the engine marks the line as being closed.

5) Islanding Check

The topology engine starts a list with a random node and adds all other nodes that are connected via closed lines. Whenever a new node is added, a redundancy check is performed so that there are no multiple entries in the same station number in the same list. The number of generated lists represents the number of islands in the system.

The output of this function is the "Island Matrix". For a system that has L separated islands, the matrix has L rows; each corresponds to one island. In each row, the first column is the number assigned to the island, while the other columns display the substations that belong to it. Finally, the last column shows whether the islands are energized or not.

6) Energization Check

The TP engine searches for any closed type 4 switches in an island. If there is at least one closed type 4 switch, the engine marks the island as energized. Otherwise, the island is de-energized. If the island is energized, 1 is the entry of the "island matrix" last column, otherwise it is 0.

The input and output matrices of the algorithm and their relationships are summarized in Fig. 3:

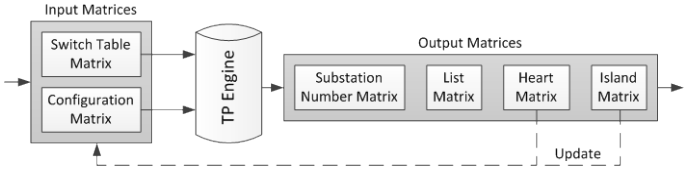

Fig. 3. *Input and Output matrices and their inter-correlation*

III. PHASOR-ASSISTED TOPOLOGY PROCESSING

The algorithm should be developed to be capable of working with data from PMUs as well. To the knowledge of the authors, [4] is the only work reporting the usage of PMU data to define systems topology. However, the approach in [4] is not well-elaborated and is exposed to several drawbacks. In this section, [4] is briefly discussed and it is shown that its proposed methodology is ineffective. With that to keep in mind, a new algorithm that exploits the PMU data is developed and fully described.

A. A New Algorithm of Topology Analysis Based on PMU Information [4]

[4] checks if a line is connected or not by comparing the current magnitude flowing through the line with a pre-defined threshold. However, there may be situations in which the disconnection of a line results in other lines to have very low current flow as well; as a consequence, those lines will be wrongly reported as being disconnected.

Moreover, to detect the changes in switches statuses, it compares the difference of the current magnitude between two consecutive timestamps, k and $k+1$. In the case that the difference is bigger than a predefined threshold, it concludes

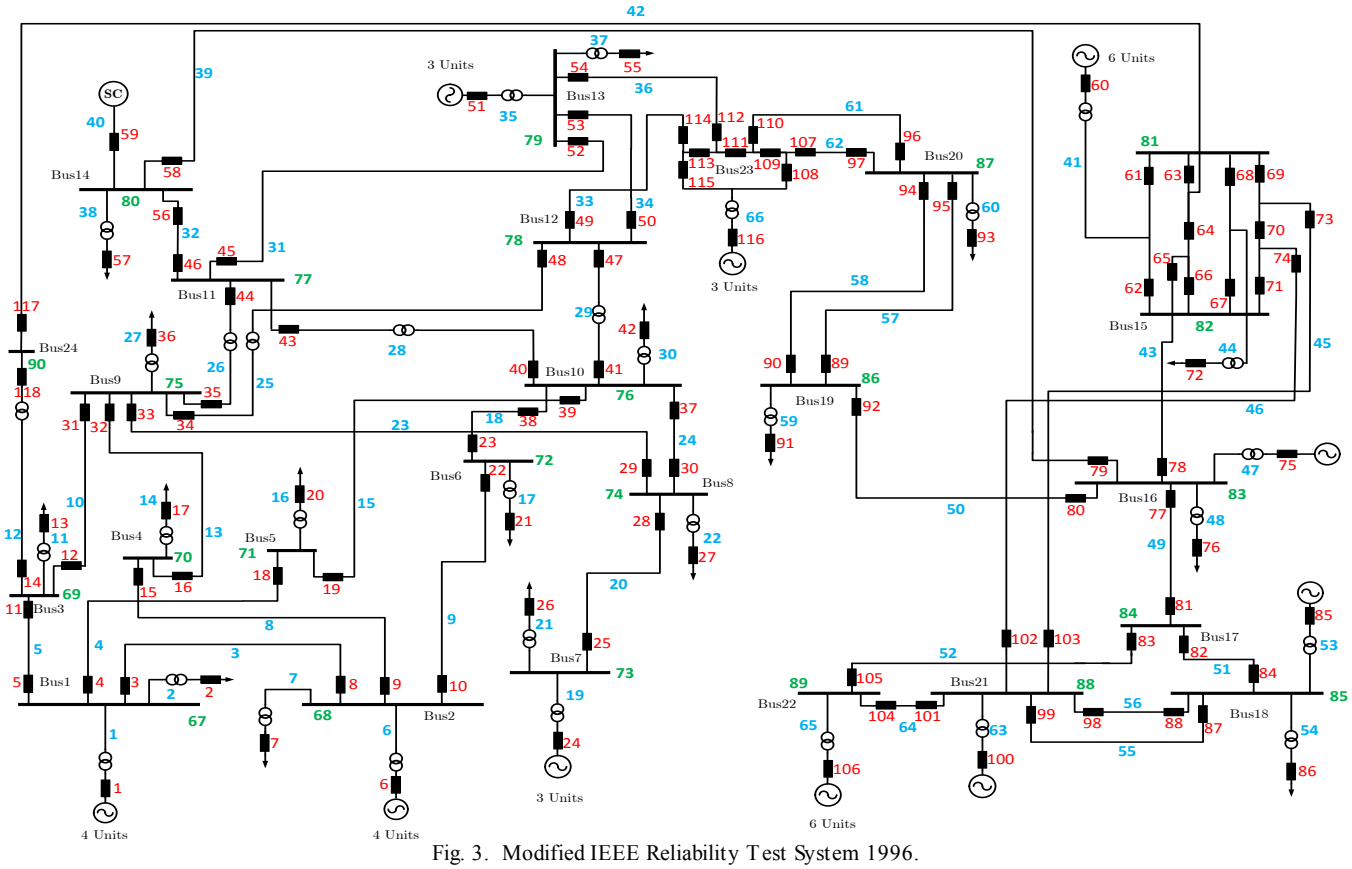

Fig. 3. Modified IEEE Reliability Test System 1996.

that the switch status has changed. However, there may be transient situations in which the current magnitude difference between two consecutive timestamps is bigger than the threshold. In this case, false switching will be reported as well.

The drawbacks mentioned above are verified through testing results, and possible remedies are suggested in the coming subsection.

B. Proposed Method

Every TP needs the switches statuses in order to determine the network topology. Traditionally, the switch statuses are telemetered using the TCP/IP or ICCP protocol. In the case of PMU, two different types of data can be used. First is the digital data which directly gives the status of the system switches which is included in the IEEE C37.118-2011 [9]. Second, the current magnitude through the lines as well as the substation voltages may be used.

The latter measurements can be used in parallel to digital data for the aim of verification. However, an independent procedure is elaborated that is based solely on PMU data.

To verify if a switch status has changed, the current magnitude passing through that switch utilized according to the following equation:

$$\left|I^{(k+1)} - I^{k}\right| > 100 \times \left|I^{k} - I^{(k-1)}\right| \quad (1)$$

Here, $k+1$ is the timestamp at which the switching has occurred for the first time. If the difference of the current magnitudes for two consecutive cycles $k+1$ and k is more than 100 times greater than the difference of current magnitudes of k and $k-1$, it can be concluded that a breaker status change has occurred. The fact that two sets of consecutive timestamps are involved compensates for transients in which large current magnitude variations may occur.

An additional verification may be carried out using the current flow phase angle through the line that corresponds to the switch:

$$\left|\Phi^{(k+1)} - \Phi^{k}\right| > 100 \times \left|\Phi^{k} - \Phi^{(k-1)}\right| \quad (2)$$

Please note that the time span between two consecutive measurements depends on the PMU measurement rate. The tests in this article are carried out using synthetic measurements at 50 samples per second. Therefore, the time span for two consecutive timestamps is 0.02 sec. Hence, in normal operating conditions the variation of current magnitude and phase for two consecutive measurements is sufficiently low. The coefficient of 100 is chosen according to a set of different simulation scenarios. Due to space limitations, it is not possible to include all those test scenarios.

After the switching has been detected (according to the above procedure), or in the case that the digital data is not available temporarily, current and voltage phasors can be

utilized to check if a line is connected or not.

Ideally, whenever a breaker is open no current is passing through its corresponding line. However this is not the case in reality, as the open line still shows some current flow. However, if the current magnitude is higher than a threshold, the line is certainly closed. The threshold is considered to be equal to 0.08 p.u. [4]:

$$I > 0.08\,pu \tag{3}$$

But if the current was below the threshold, additional check should be carried out in order to find the line status. Suppose that the line connects substation *n* to substation *m*. The difference between the voltage magnitudes of two substations is calculated; if it was above the voltage threshold, the line is disconnected. Otherwise it is connected.

$$\left\| V_m \right| - \left| V_n \right\| > 0.005\,pu \tag{4}$$

When a line has a flow below *0.08p.u.* but is not practically open, the voltage loss through the line is very low; this results in a very low voltage difference between the stations that are connected by that line. Therefore, by comparing the corresponding station voltages, the line status can be defined.

The threshold of 0.005 pu is defined based on several tested scenarios. The pre-defined threshold may be defined differently for each network. One future work can be to correlate the threshold value with the network parameters and properties.

IV. RESULTS

This section provides rigorous testing results on the new topology processing algorithm, as well as the PMU measurement extension. A modified version of the IEEE Reliability Test System 1996 [7] is simulated in real-time using the eMegasim Opal-RT real-time simulator. This has provided the possibility of having the simulation values as synthetic measurements very similar to PMU data, with a rate of 50 samples per second. As the modeling platform for this real-time simulator is MATLAB/Simulink, the model is implemented in Simulink with several modifications.

In this section, the test system is introduced providing enough information about the simulation set-up to generate synthetic PMU measurements. Then consecutive switching scenarios are carried out to show the efficiency of the proposed TP algorithm. Finally, the procedures that exploit the use of PMU measurements are tested, and the results are discussed.

A. Modified IEEE Reliability Test System 1996

As shown in Fig. 3, this system consists of 24 substations, 90 circuits (66 lines and 24 buses) and 119 switches. This test system has all 4 types of switches, and the most common configurations for a substation, i.e. One and a Half Switch, Double Bus Double Switch, and Ring. Note that this is a modification of IEEE Realiability Test System 1996. The original version has three areas (all of which are identical); just one of these areas is considered here.

TABLE I
TOPOLOGY PROCESSOR TESTING: SCENARIOS' DESCRIPTION AND RESULTS

Scenario	Description	Results
1	5 switches are opened simultaneously. 2 Stations suffer changes inside.	Stations 15 and 23 are reported with changes inside. For station 15, 2 closed paths are found. One of them has two circuits 42 and 43 (node 25) while the other one has the remaining 6 circuits of the station (node 15). For station 23, 2 closed paths are found. One of them has two circuits 36 and 61 (station 26), while the other the one contain 3 circuits 62, 33, and 64 (node 23). Line 28 is reported as disconnected. There is just one island in the system containing all of the system substations. Computation time: 5.4 ms
2	7 switches change status simultaneously. 1 Stations suffer changes inside.	Station 23 is reported as the only one with changes inside. There is one single closed path containing all of station 23 circuits. That means nodes 23 and 26 are merged back to original station (23). Lines 23, 9, 29, 15, and 19 are reported as disconnected. Lines 28 were disconnected previously.

B. Switching Scenarios

Initially, all the breakers in the system are closed. In the first scenario, breakers 43, 63, 66, 109 and 113 are opened simultaneously. This means that stations 15 and 23 have suffered changes within their substation configuration.

In the second switching patter, breakers 109 and 113 are closed. Also, breakers 22, 24, 33, 39 and 47 are opened. All these changes are made simultaneously. Please note that breakers 63, 66 and 43 are still open.

Due to space limitations, no synthetic measurements from these scenarios are shown. The results of carrying out TP for these two scenarios are shown in Tab. 1.

From the results of Tab. 1, it can be observed that the proposed algorithm of this article is quite efficient in dealing with several different topological situations. This includes the ability to perfectly detect and report splitting or merging nodes, and the well-operating number assignation.

The proposed TP has all the necessary parts of a typical topology processor, and no part is neglected. Moreover, it performs topology processing just for those parts of the system which have suffered changes compared to the previous execution cycle.

The computation time is in order of milliseconds. However, the complexity of the scenarios under test is higher than in normal operation as several breakers are switching at the same

time. Also, simultaneous changes happen at different parts of the system. In addition, the CPU used for performing TP is Intel® Core™ i7-2600 CPU @ 3.40GHz. of a typical PC; we are using MATLAB in a Windows 7 OS. The computation time will further decrease using a production-grade platform for performing TP calculations.

C. *Phasor-Assisted TP Results*

To scrutinize the validity of the proposed method to exploit PMU measurements, breaker 118 is opened. First, the current phasor through line 12 is shown in Fig. 4.

The breaker status changes at *t=33.96*. As the breaker opens, sudden changes occur in its corresponding line current magnitude and phase. The current magnitude suddenly decrease to almost 0; in the case of current phase, it shows a sudden decrease at the same time that the breaker opens; then

Fig. 4. Current flow through line 12

it has an increase step in the next snapshot, i.e. *t=33.98*.

- Magnitude Verification:

$$I^{33.96} - I^{33.94} = 1.208 - 2.563 = -1.355\ pu$$
$$I^{33.94} - I^{32.92} = 2.563 - 2.563 = 0 \quad (5)$$

The current magnitude difference in time stamp 33.96 with its previous time stamp 33.94 is greater than 100 times of difference between magnitude at instances 33.94 and 33.92, which means that switching has ocurred at time 33.96, as expected from equation (1).

- Phase Verification:

$$\Phi^{33.94} - \Phi^{33.92} = 2.495 - 2.498 = -0.003$$
$$\Phi^{33.96} - \Phi^{33.94} = -6.89 - 2.495 = -9.385 \quad (6)$$
$$\Phi^{33.98} - \Phi^{33.94} = 76.85 - 2.495 = 74.355$$

Phase verification is carried out for both the instant of the switching and also for one time stamp later. As it can be seen from the above calculations, equation (2) is verified by an acceptable margin, so it helps in certifying that the switch status has changed.

Next step is to define if line 42 is connected. Because Line 42 is in series with line 12, its current shows similar behavior. This is shown in Fig. 5.

It can be observed that in addition to the current magnitude in line 12 (which is shown in Fig. 4), the current magnitude of line 42 is also below the threshold of 0.08 pu (satisfying equation (3)).

Therefore, additional check should be performed using equation (4). From Fig. 3, it can be seen that line 12 connects stations 3 and 24, while line 42 connects stations 15 and 24. Therefore, the voltage difference between stations 24 and 15, and also 24 and 3 is scrutinized. The result is shown in Fig. 6.

As it can be observed in Fig. 6, the voltage difference before switching is almost 0.001pu for stations 3 and 24, and is almost 0.021pu for stations 24 and 15. However, after the switching, the voltage difference is almost 0.002p.u. for stations 15 and 24 (remains below threshold); for stations 3 and 24, it reaches almost 0.08pu which is far above the threshold of 0.005pu; this shows that the link between stations 3 and 24 is disconnected, while the one between 15 and 24 is still connected.

These examples show the efficiency of the proposed method to effectively exploit PMU data for topology processing.

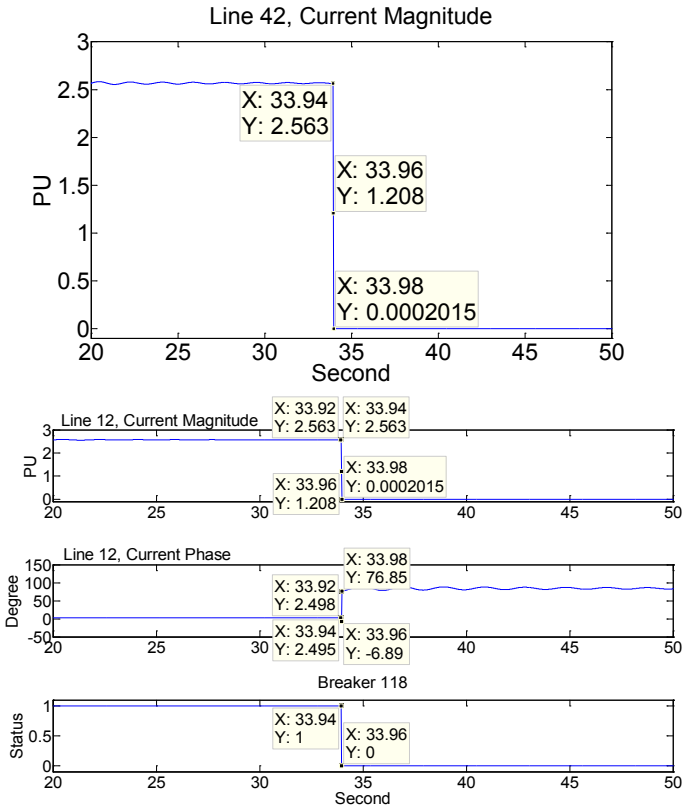

Fig. 5. Current flow through line 42

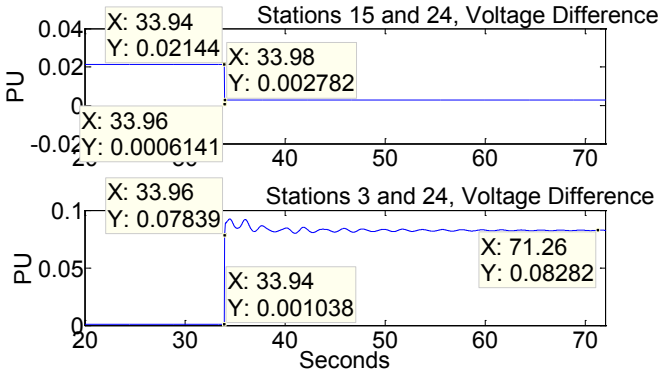

Fig. 6. Station Voltages' Difference

V. CONCLUSION

The results presented in this paper show that the proposed algorithm for TP is efficient in dealing with several different operation scenarios with complex topological changes. This includes the ability to perfectly detect and report splitting or

merging nodes, and a well-operating number assignation.

The proposed TP has all the necessary parts of a typical topology processor, and it is capable of performing topology processing just for those parts of the system which have suffered changes compared to the previous execution cycle. The outputs are in a form which makes the changes in the topology to be easily traced.

The exploitation of PMU data for enhancing the quality of the TP output through data verification is described. The illustration of different scenarios by real-time simulation waveforms used as synthetic PMU data makes the explanations and procedures rigorous. The proposed extensions for exploiting PMU data are used to verify the traditional telemetered data. This ability can be used in parallel with the main TP or as an independent function.

In the hypothetical case that PMUs become omnipresent in power systems (being capable of transmitting all breaker status in their digital channels), the method in this article can then also work using PMU data only, and the verification rules can be used to check the data being sent in the digital channels of the sychrophasor stream.

As a result, the algorithm is a quite robust TP, which is able to deal with any kind of topological changes in the system. It is fast enough to work with both conventional and PMU data-based SEs. Please note that the TP algorithm is executed on a typical PC using Intel® Core™ i7-2600 CPU @ 3.40GHz. The computation time will further decrease using a production-grade platform for performing TP calculations. In addition, parallelizing the TP execution so that the tasks are separated to independent cores (multi-core computations) will significantly decrease the computation time.

ACKNOWLEDGMENT

The literature, comments and suggestions provided by Mr. Magnus Johansson from Svenska Kraftnät, the Swedish Transmission System Operator (TSO), are sincerely acknowledged.

REFERENCES

[1] L. Vanfretti, J.H. Chow, S. Sarawgi, and B.B. Fardanesh, "*A Phasor-data-based State Estimator incorporating phase bias correction*", IEEE Transactions on Power Systems, Vol. 26, pp. 111-119 April, 2010.

[2] A.M. Sasson, S.T. Ehrmann, P. Lynch, and L.S. Van Slyck, "Automatic Power System Network Topology Determination," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-92, no. 2, pp. 610-618, 1973.

[3] A. Bose and M. Prais, "A Topology Processor That Tracks Network Modifications Over Time," IEEE Transactions on Power Systems, vol. 3, no. 3, pp. 992-998, 1988.

[4] C. Qian, Z. Wang, and J. Zhang, "A New Algorithm of Topology Analysis Based on PMU Information," *IEEE 5th International Conference on Critical Infrastructure (CRIS), 2010*, pp. 1-5, 2010.

[5] R. Kussel, R. Chrustowski, and C. Jaborowicz, "The Topology Engine: A new Approach To Initializing And Updating The Topology Of An Electrical Network", ABB Netzleittechnik GmbH, Germany

[6] A.G. Phadke, J.S. Thorp, and K. Karimi, "State Estimation with Phasor Measurements," *IEEE Transactions on Power Systems*, vol. 1, no. 1, pp. 233-238, 1986.

[7] C. Grigg et al, "The IEEE Reliability Test System-1996. A report prepared by the Reliability Test System Task Force of the Application of Probability Methods Subcommittee", *IEEE Transactions on Power Systems*, vol. 14, no. 3, pp. 1010-1020, 1999.

[8] eMegaSim Simulator documentation from OPAL-RT available at: http://www.opal-rt.com/sites/default/files/brochure-eMEGAsim-120507.pdf

[9] "IEEE Standard for Synchrophasors for Power Systems", IEEE C37.118 – 2011.

[10] A.G. Phadke and J.S. Thorp, "*Synchronized Phasor Measurements and their Applications*", Springer, New York, 2008.

[11] M. Farrokhabadi and L. Vanfretti, "State-of-the-Art of Topology Processors for EMS and PMU Applications and their Limitations", *IEEE IECON 2012*, pp. 1422-1427, Oct. 2012.

Mostafa Farrokhabadi (Student Member 2011) obtained the BSc in Electrical Engineering from Tehran Polytechnic University in 2010. He recently obtained the MSc in Electrical Power Engineering degree from KTH Royal Institute of Technology, Stockholm, Sweden in April 2012. During his studies he focused on electrical power systems and real- time simulation, and wrote his MSc thesis on topology processing for EMS and PMU applications. In addition, he was involved in an educational research project on constructive alignment implementation in power system analysis courses from 2011 to 2012. He has started his PhD studies at University of Waterloo, Canada, in May 2012. His research interests are applications of PMUs in power systems, power system stability and smart grids.

Luigi Vanfretti (Student Member 2003, M'2010) is a tenured Associate Professor at the Electric Power Systems Department, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden. He was conferred the Swedish title of Docent in 2012 and was an Assistant Professor in the same department from 2010 to 2013.

Since 2011, he serves as Special Advisor for Strategy and Public Affairs at the R&D Division of Statnett SF, the Norwegian power system operator. His duties include architectural analysis for sychrophasor data transfer, communications and application systems for Smart Grid applications; as well as providing inputs into R&D strategy development and aiding in the excecution of collaborative projects with universities, TSOs and R&D providers.

He received the Electrical Engineering Degree from Universidad de San Carlos de Guatemala in June 2005. He was a visiting researcher at the Department of Electronics and Electrical Engineering of The University of Glasgow, Scotland in 2005. He received his M.Sc. in 2007 and Ph.D. in 2009 both in Electric Power Engineering from Rensselaer Polytechnic Institute (RPI), Troy, NY, USA. For his research and teaching work towards his Ph.D. he was awarded the Charles M. Close Award from Rensselaer Polytechnic Institute in 2010.

Since 2009, he has served as Secretary of the IEEE Task Force on Open Source Software and as WG Organizer for the IEEE PSDP Working Group on Power System Dynamic Measurements. His research interests are in the general area of modeling, dynamics, stability and control of power systems; while his main focus is on the development of applications of PMU data. He is an evangelist of Free/Libre and Open Source Software for power system engineering education, and the application of constructive alignment theory for enhancing power engineering education.