Feeder Topology Identification

Raffi Avo Sevlian and Ram Rajagopal

Abstract—Advanced control of distributed energy resources at the consumer level requires full situational awareness of the distribution system. One important problem is that of feeder topology identification which involves detecting changing switch configurations given a sparse number of measurements. We formulate the problem for residential feeders as a spanning tree identification problem over a general graph. Given a set of power flow measurements and load pseudo measurements, we show that the underlying graph structure is crucial in defining identifiability of the correct spanning tree on the graph. First we solve the deterministic case of known true loads and measured power flow. We show that the placement of sensors on the network alone determine whether the set of spanning trees can be correctly identified. In the stochastic case where loads are we are given only load forecasts, we present a locally optimal sensor placement algorithm.

I. INTRODUCTION

The need for advanced controls in the distribution system is an emerging topic in power system and the controls community. For example, the advanced dispatching of distributed energy resources [1], [2], and the operational aspects of maintaining a stable grid under this new regime of deep renewable penetration is a recent topic of interest [3], [4], [5], [6]. Crucial to the operation of these advanced methods and verification of its efficacy is situational awareness of the distribution system. The growth of sensing and communication infrastructure is key in providing real time insight into the state of the distribution system.

Historically, work on topology detection in the power systems community has focused on the transmission system, where the focus was detecting the switching of protection devices given measurements on the network [7]. In such a scenario, the network before and after a protection event is a connected graph. These methods have relied on running a least squares based state estimator usually using voltage magnitude as the primary measurement. The accuracy of these methods relies mostly on the state estimators accuracy.

Topology identification on the distribution system differs from the transmission system, since the network must be radial at all times. In a feeder system, there are generally multiple power sources which can be connected and disconnected to ensure a connected network [8]. The problem of topology detection of a distribution feeder reduces to that of identifying a spanning tree on a larger graph given a set of measurements.

This paper is organized as follows. Section II formulates the problem of topology detection where we show how the detection of switch configurations of a feeder can be converted to that of spanning tree identification. Then we present the load and sensor models used in the identification problem along with the sensor placement optimization. Section III presents some needed graph theory definitions used in the problem solution. Section IV presents the metrics used to evaluating sensor placement in the deterministic and stochastic setting. Sections V, VI solves the placement problem in both settings with numerical demonstrations given in Section VII.

II. TOPOLOGY IDENTIFICATION PROBLEM FORMULATION

We start by proposing a simple graph theoretic representation of the valid configurations of a feeder for topology identification goals then use this model to develop the detection and placement problems.

A. Extended Island Graph: representing feeder configurations

We describe a method of representing the set of switch operations of a typical distribution system. We construct a reduced island graph which captures each feasible topology of a reconfigurable distribution grid. As an illustrative example, consider the IEEE 123 node feeder in Figure 1(a). There are multiple loads connected to each other and separated by switches. These switches demarcate various islands of the distribution system. We can then think of the network...
as a set of connected islands. The typical feeder will have multiple power sources i.e. feeders. The set of switches are appropriately set so that each group of islands is connected to a load while ensuring no flow from one feeder to another.

### TABLE I

| Island Graph | IEEE Test Feeder |
|--------------|------------------|
| switch       | nodes            | Load ID |
| w1           | F2(v8) - w1      | (250 – 251) |
| w2           | v1 - v2          | (18 – 135) |
| w3           | v2 - v3          | (151 – 300) |
| w4           | v4 - v5          | (97 – 197) |
| w5           | v4 - F3(v2)      | (450 – 451) |
| w6           | v3 - v4          | (54 – 94) |
| w7           | v1 - v3          | (13 – 152) |
| w8           | F1(v6) - w1      | (149 – 150) |
| w9           | F1(v6) - F4(v9)  | (95 – 195) |

The standard IEEE network in Figure 1(a) can be reduced to a more simplified diagram as shown in Figure 1(b). This reduced feeder simply displays the various switches and load islands of the original network. The distribution system must always maintain a tree structure so power always flows from feeders to loads. The connection of two feeders, for example by both w1 and w8 being closed will cause instability in the network and is thus always to be avoided.

**Extended Island Graph.** A simple representation can be constructed for capturing switch configurations. We can represent all valid topologies where all islands are connected to a load with no direct path between feeders. This is done by constructing what we refer to as an extended island graph in Figure 2(a).

This is done by reducing each of the feeders F1 . . . F4 and islands v1 . . . v5 to vertices of the graph. Each switch w_i in the reduced feeder corresponds to edge e_i in the island graph. A complete mapping between the reduced feeder and the extended island graph is given in Table I. Given a total of |W| switches and |F| feeders, edges e1|W|+1 . . . e|W|+|F| are added to connect the feeders to the virtual node vr.

The extended island graph G = (V, E) is the network used in the remaining analysis. First we define a directed subgraph of G, τ0 = { (v_r, F_i) | F_i feeders. This construction leads to a simple method for enumerating each valid topology of the extended island graph.

**Proposition 1:** Every spanning tree of the extended island graph which includes τ0 represents a valid topology.

Abusing notation we refer to the set of spanning trees containing subtree τ as Τ(τ(G)). Therefore enumerating every spanning tree of the island graph which includes τ0 will generate the set of topologies corresponding to valid switch configurations. Figure 2 represents an example spanning tree that can be constructed. Τ(G) is the set of all valid spanning trees that can be generated by graph G and is a proper superset of the valid topologies in Proposition 1.

The number of possible spanning trees |Τ(G)| = T. Therefore, all valid switch configurations must map to some Τ ∈ Τ(G). Note the naive method of testing all 2^|W| switch positions and testing for full connectivity of all loads as well as invalid feeder loops is ill defined and impractical. Therefore this method provides a structured approach to analyzing the problem.

### B. Load Model

Each node v_n excluding the virtual root and feeder nodes in the graph has a consumption load x_n and pseudo-measurement $\hat{x}_n$. We denote the error by $\epsilon_n = x_n - \hat{x}_n$. We assume that the loads are single phase real power quantities and the errors are mutually independent random variables: $\epsilon_n \sim N(0, \sigma_n^2)$ and $x_n \sim N(\hat{x}_n, \sigma_n^2)$. The $\mathbf{x}$ and $\hat{\mathbf{x}}$ represent the vector of true loads and that of load pseudo-measurements; thus $\mathbf{x} \sim N(\hat{\mathbf{x}}, \Sigma)$. Covariance matrix $\Sigma$ being diagonal.

### C. Measurement Model:

For any edge e, we denote by s the power flow on it to all active downstream loads. The measured flow depends on unknown network topology and true loads.
The sensor placement $\mathcal{M} \subset E$ is a subset of edges of the network. For instance the sensor placement in Figure 2(a) correspond to $\mathcal{M} = \{e_1, e_9\}$. In general for a topology $\mathcal{T} \in \mathcal{T}(G)$ and measurements $s = \{s_1, \ldots, s_{|\mathcal{M}|}\}$ the $k$th flow is $s_k(x) = \sum_{v_j \in V_k^+(\mathcal{T})} x_j$. Where $V_k^+(\mathcal{T})$ is the subset of downstream nodes for a particular placement and spanning tree. Given a placement, we can represent the observations with

$$s = \Gamma(\mathcal{T}, \mathcal{M})x$$

(1)

where $\Gamma \in \{0, 1\}^{|\mathcal{M}| \times |V|}$.

D. Topology Detection and Sensor Placement

Given an unknown set of switch positions $w = \{w_1, \ldots, w_n\}$ and the resulting spanning tree $\mathcal{T}$, we measure a set of flows in real time as well as set of load pseudo measurements. Depending on whether spanning tree, we must infer the topology from the measured flows and load pseudo measurements. Depending on whether the pseudo measurements are exact or not, we have the following two cases.

- Deterministic case: Choose the topology matching the measured flows under the known loads.
- Stochastic case: Choose the most likely topology via general MAP detector given by

$$\hat{\mathcal{T}} = \arg \max_{\mathcal{T} \in \mathcal{T}(G)} \Pr(\mathcal{T} | \hat{x}, s).$$

(2)

The goal of the optimal sensor placement is to enable the topology detection and provide the best tradeoff between the number of sensors and reduction in the detection error. Based on characterizing the performance of optimal detection and the minimal identifiability requirements we can develop an optimal sensor placement problem formulation for both the deterministic and stochastic cases.

III. GRAPH THEORY PRELIMINARY

We review some key concepts in graph theory used in this work following [18].

1) A cycle $c = \{e_1, \ldots, e_N\}$ is a connected subgraph where vertex has degree 2. The set $B(E)$ is the power set over the edge set $B(E) = \{0, 1\}^{|E|}$. Any cycle $c$ is a vector in the space defined by $B(E)$.

2) The space of all possible cycles in a particular graph is $\mathcal{C}(G) \subset B(E)$. A basis $B_C \subset B(E)$ is the smallest set of cycles whereby all other cycles can be constructed via symmetric difference operator $E \oplus E' = (E \cup E') \setminus (E \cap E')$. So for any cycle $c = \bigoplus_{e \in B_C} e'$.

3) The dimension of the cycle basis $\dim \mathcal{C}(G) = \mu(G)$, is referred as the cyclomatic number of the graph. For any graph $\mu(G) = |E| - |V| + n(G)$ where $n(G)$ is the number of connected components of the graph. For example the extended island graph in Figure 2(a) has $n(G) = 1$, and $\mu(G) = |[13] - [10]| + 1 = 4$.

4) A fundamental cycle basis is a set of cycles $c_1, \ldots, c_{\mu(G)}$ where there exists an edge $e_i \in c_i$ which is unique to that cycle and is in no other cycle in the set.

IV. TOPOLOGY IDENTIFICATION: IDENTIFIABILITY AND OPTIMAL DETECTION

A. Deterministic Case: Tree Identifiability

In the case of perfect knowledge of loads and power flows, we require a sensor placement such that different spanning trees lead to unique measured flow values.

The following definition is of use:

**Definition 1:** The set $\mathcal{T}(G)$ is identifiable if $\beta \mathcal{T}, \mathcal{T}' \in \mathcal{T}(G)$ with $s = s'$. That is we never have two unique spanning trees that map to the same measured flow $s$.

In the deterministic case, we desire a placement $\mathcal{M}$ such that $\mathcal{T}(G)$ is identifiable. Checking this condition for a general graph and placement in a naive manner involves enumerating each spanning tree and evaluating the measured flows, which will have $O(|V| \cdot T^2)$ complexity. Theorem 1 provides a simple condition to verify on $\mathcal{M}$ which provides a guarantee that $\mathcal{T}(G)$ be identifiable. The condition can be evaluated without spanning tree enumeration.

B. Stochastic Case: Minimax and Mean Error

Since the true loads at anytime are not known with complete certainty, we must deal with uncertainty in the pseudo-measurements. Therefore the detector in eq. 2 will not only deal with undetectability, but also have a finite error rate. Given the MAP detector in eq. 2, we can compute the missed detection error for any hypothesis $\Pr(\hat{\mathcal{T}} \neq \mathcal{T} | \mathcal{T}; \mathcal{M})$. This allows us to evaluate a placement by minimizing one of two commonly used objective functions:

1) The maximum error over all possible spanning trees in the network is given by:

$$\mathcal{M}^* = \arg \min_{\mu(G) \leq |\mathcal{M}| \leq M} \max_{\mathcal{T} \in \mathcal{T}(G)} \Pr(\hat{\mathcal{T}} \neq \mathcal{T} | \mathcal{T}; \mathcal{M}).$$

(3)

The placement constraints $\mathcal{M} : \mu(G) \leq |\mathcal{M}| \leq M$ is placed since any $\mu(G) \geq |\mathcal{M}|$ leads to an unidentifiable network (Theorem 1) whereby the maximum error is always 1. This placement is very conservative since it is a guarantee on the worst possible detection error. If all spanning trees are possible in the network, this may be the best option.

2) An alternative placement criterion is to minimize the mean missed detection error over all hypothesis. This is given by the following optimization:

$$\mathcal{M}^* = \arg \min_{|\mathcal{M}| \leq M} \sum_{\mathcal{T} \in \mathcal{T}(G)} \Pr(\mathcal{T}) \Pr(\hat{\mathcal{T}} \neq \mathcal{T} | \mathcal{T}; \mathcal{M})$$

(4)

Although $|\mathcal{M}| \leq \mu(G)$ will result in a non-degenerate value, the error is extremely high since a large number of trees are unidentifiable. A realistic constraint is again $\mu(G) \leq |\mathcal{M}| \leq M$.
V. Deterministic Placement

A. Cycle Space Observability

For a placement \( \mathcal{M} \) on \( G \), we can define the following.

**Definition 2:** A placement \( \mathcal{M} \) covers \( C(G) \) if \( \forall \mathcal{E} \in C(G) \) where \( E(\mathcal{E}) \cap E(\mathcal{M}) = \emptyset \).

That is, all cycles formed on the graph must have at least one edge in \( \mathcal{M} \). If we can find a cycle where \( E(\mathcal{E}) \cap E(\mathcal{M}) = \emptyset \) then we know for certain that \( \mathcal{M} \) does not cover \( C(G) \).

Conversely, \( \mathcal{M} \) covers \( C(G) \) if \( |\mathcal{M}| \geq \mu(G) \), where \( \dim C(G) = \mu(G) \) and \( |\mathcal{M}| \) is the size of the placement. Since \( \mathcal{M} \) can be any size greater than \( \mu(G) \), we would like to restrict our attention to the following:

**Definition 3:** A placement \( \mathcal{M} \) is a minimal cover of \( C(G) \) if \( \mathcal{M} \) covers \( C(G) \) and \( |\mathcal{M}| = \mu(G) \).

The primary result we wish to show is the following.

**Theorem 1:** \( \mathcal{T}(G) \) is identifiable if and only if \( \mathcal{M} \) covers \( C(G) \).

**Proof:** The proof is quite involved and is in the Appendix.

The intuition of Theorem 1 is that to have observability of all spanning trees, there must be observability of the cycle space of the graph. The minimum guarantee for cycle observability is the determinant of the cycle space. Therefore in the deterministic case, topology identification under some placement problem.

B. Enumerating Valid Placements

The set of all valid placements for a graph \( G \), (i.e., satisfies definition 3) is denoted by \( \mathcal{M}(G) \). Definition 3 provides the conditions under which any two spanning trees are identifiable but does not provide a way in which to construct the set of placements. Given a valid spanning tree \( T \), we define \( f(T) = E(G) \setminus E(T) \).

We can now state the following:

**Theorem 2:** The function \( f : \mathcal{T} \rightarrow \mathcal{M} \) is a bijection (one-to-one and onto) between the set \( \mathcal{T}(G) \) and the set of all valid minimal covers \( \mathcal{M}(G) \).

**Proof:** See Appendix.

**Remark 1:** Theorem 2 implies that \( |\mathcal{M}(G)| = |\mathcal{T}(G)| \).

Theorem 2 is quite important from a placement perspective since it actually yields a method to generate a valid placement in the deterministic case. Also, it allows us to enumerate all valid placements for a graph. This is important when dealing with stochasticity in the detection and placement problem.

VI. Stochastic Placement

For simplicity, we redefine the objective functions in (3) and (4) as:

\[
\begin{align*}
\mathcal{g}_1(\mathcal{M}) &= \max_{\mathcal{T} \in \mathcal{T}(G)} \Pr(\hat{\mathcal{T}} \neq \mathcal{T} \vert \mathcal{T}; \mathcal{M}) \\
\mathcal{g}_2(\mathcal{M}) &= \sum_{\mathcal{T} \in \mathcal{T}(G)} \Pr(\mathcal{T}) \Pr(\hat{\mathcal{T}} \neq \mathcal{T} \vert \mathcal{T}; \mathcal{M})
\end{align*}
\]

A. Monotone Decreasing Objective

We can show that the placement objective functions \( \mathcal{g}_1(\mathcal{M}) \) and \( \mathcal{g}_2(\mathcal{M}) \) are monotonically decreasing functions. This guarantees that a greedy placement algorithm, such as Algorithm 1 is guaranteed to find a locally minimum solution.

Given a multivariate hypothesis testing problem with

\[
\mathbf{s} \mid H \sim \begin{cases} 
N(\mu_i, \Sigma_i), & H = T_i \\
N(\mu_j, \Sigma_j), & H = T_j 
\end{cases}
\]

where \( \mu_i = \Gamma_i \hat{x} \) and \( \Sigma_i = \Gamma_i \Sigma_i^T \).

We have the following detector:

\[
\hat{\mathcal{T}}(s) = \begin{cases} 
T_i, & s \in R_{ij} \\
T_j, & s \notin R_{ij}
\end{cases}
\]

(8)

Where the regions \( R_{ij} \) is specified by the maximum likelihood criterion. The acceptance region for deciding \( H = \mathcal{T} \) or \( H \neq \mathcal{T} \) can be constructed by \( R_i = \bigcap_j R_{ij} \).

For any pairwise test \( T_i \) vs. \( T_j \) we can compute the pairwise error probability, \( \Pr(\hat{T}(s) = h_j \mid H = h_i; \mathcal{M}) \). This is used to compute the missed detection probability

\[
\Pr(\hat{T}(s) \neq T_i \mid H = T_i; \mathcal{M}) = \sum_j \Pr(\hat{T}(s) = T_j \mid H = T_i; \mathcal{M}).
\]

(9)

**Lemma 1:** For all \( T_{ij} \) pairwise tests \( \Pr(\hat{T}(s) \neq T_j \mid H = T_i; \mathcal{M}) \leq \Pr(\hat{T}(s) = T_j \mid H = T_i; \mathcal{M}) \) whenever \( \mathcal{M} \subset \mathcal{M}' \).

**Proof:** See Appendix.

**Theorem 3:** Under different sensor placements \( \mathcal{M} \subset \mathcal{M}' \) we have \( g_K(\mathcal{M}) \geq g_K(\mathcal{M}') \) for \( k = 1, 2 \).

**Proof:** See Appendix.

B. Greedy Placement

We propose Algorithm 1 as a greedy method for minimizing either eq. (3) or eq. (4). The greedy placement is initialized with a valid placement which minimizes the objective. At every subsequent iteration, we evaluate the objective over all the remaining edges in the graph \( E(G) \setminus E(\mathcal{M}_{GD}) \).

**Algorithm Initialization:** Any placement with \( |\mathcal{M}| \leq \mu(G) \) will result in a deficient starting point. Therefore if the objective is the maximum error \( g_1(\mathcal{M}) \) we must initialize the algorithm with line 1. If we aim to minimize the mean error \( g_2(\mathcal{M}) \), then an alternative initialization is \( \mathcal{M} \leftarrow \{\emptyset\} \).

VII. Numerical Experiment

We conduct a number of numerical experiments evaluating the general performance of the placement methods...
presented. (1) We evaluate the deterministic placement for some randomly generated planar graphs with high spanning tree counts. (2) Then we evaluate the stochastic placement for both max and mean placement error on a graph with high edge counts and low cyclomatic number and compare the result to a randomized placement. (3) We evaluate the stochastic placement, on the IEEE 123 Node Test Feeder. (4) We present a numerical counterexample illustrating the optimization problem is not sub/sup modular.

### A. Deterministic Placement

We test the placement problem on a set of planar graphs, shown in Figure 3. In both graphs, designate a single vertex as the source, which is indicated in the dashed horizontal lines. Graph $G_1$ has $v_{\text{root}} = v_1$, $G_2$ has $v_{\text{root}} = v_4$ and $G_3$ has $v_{\text{root}} = v_1$.

![Sample graphs](image)

To test the deterministic placement, we enumerate the set of spanning trees for each of the graphs. The method relies on the backtracking method developed in [19]. The simulation was implemented in MATLAB and deemed correct by checking that each spanning tree was unique and the number of test trees corresponded to those calculated from the matrix-tree theorem [18], where $\tau(G) = \det(L_v)$ and $L_v$ is the minor of the laplacian matrix. The result is invariant to $v$.

For the three graphs in Figure 3, the graph statistics and experiment results are shown in Table II. We evaluate the confusion matrix for the detection problem as:

\[
b_{i,j} = \begin{cases} 
1, & s(T_i) = s(T_j) \\
0, & \text{else}
\end{cases}
\]

And error rate $\epsilon = \frac{1}{2}1^T B 1$ which computes the total missed detection, correcting for double counting.

From Theorem 1, the missed detection error must be zero. The computed $\epsilon$ in Table II verifies this.

We also show the error if the direction of flow is missing. This is a subtle point that is used in the proof of Theorem 1, therefore violating this assumption leads to Theorem 1 failing to hold. The computed values for $\epsilon'$ are shown in Table II. We evaluate each valid placement in $\mathcal{M}(G)$ with Table II reporting the mean $\pm$ the standard deviation of the error rate. This verifies that different placements result in different unsigned missed detection rates. We see that if the direction of flow is not known, around 10% of the spanning trees are indistinguishable on average.

### B. Stochastic Sensor Placement

![Graphs](image)

![Algorithm 1: Greedy Sensor Placement](image)

**Algorithm 1:** Greedy Sensor Placement

| Input: | [1] Graph $G$  
|        | [2] Nominal Load Statistics $L$, $\Sigma$  
|        | [3] Maximum Sensors $M$  
| Output: | Greedy Placement $\mathcal{M}_{GR}$  
| 1 $\mathcal{M}_{GR} \leftarrow \arg \min_{\mathcal{M} \in \mathcal{M}(G)} g_i(\mathcal{M})$  
| 2 while $|\mathcal{M}_{GR}| \leq M$ do  
| 3 $e^* \leftarrow \arg \min_{e \in E(G) \setminus E(\mathcal{M}_{GR})} g_i(\mathcal{M}_{GR} \cup e)$  
| 4 $\mathcal{M}_{GR} \leftarrow \mathcal{M} \cup \{e^*\}$  
| 5 end  

**Table II**

|\begin{tabular}{|c|c|c|c|c|}
|----------------|----------------|----------------|----------------|
|\mu(G) & \tau(G) & $\epsilon$ & $\epsilon'$ & $|E|/\mu(G)$ |
|---|---|---|---|---|
|G_1 & 5 & 391 & 0 & 56.9 $\pm$ 26.7 & 2.8 |
|G_2 & 5 & 830 & 0 & 139.3 $\pm$ 72.2 & 3.2 |
|G_3 & 2 & 375 & 0 & 95.9 $\pm$ 32.5 & 17.5 |
|\end{tabular}|

We evaluate the performance of Algorithm 1 for both the max-error and mean-error objectives. The algorithm is tested on Graph $G_3$ with each node having an identical $\mu_i = 1, \sigma = 0.1$. For comparison, we evaluate the mean of 100 randomly allocated placements for each size $\min(100, \left|\mathcal{M}\right|)$. The performance is indicated in Figure 4 for both metrics. The graphs indicated a clear improvement as opposed to randomized placement. For the mean error
metric, $|{\cal M}_{GR}| \geq 6$ has an error rate less than 0.005 which is a sensor density of 17%. A randomized method has much poorer performance on average. The results are much worse in the max error case, which is expected. For very large sensor densities, the maximum error is still quite high. For the max error metric, $|{\cal M}_{GR}| \geq 6$ has an error rate less than 0.005 which is a sensor density of 17%.

C. Supermodularity Counterexample

Sub/Super modularity is a property commonly exploited in many combinatorial optimization problems (see [20]) for more details. It is useful since it guarantees that a greedy algorithm is within a factor of $(1 - \frac{1}{e})$ of the optimal value.

Definition 4: For every $M \subset {\cal M} \subset E(G)$ we have $\forall e \in E(G) \setminus B$ with $g(M \cup \{e\})-g(M) \leq g(M' \cup \{e\})-g(M')$. \hspace{1cm} (12)

Intuitively the relative decrease in the objective function must be larger for the smaller set under all subsets $M, M'$ and additional element $e$.

\[ \Delta_k(e) = g_k(M \cup \{e\}) - g_k(M) \] \hspace{1cm} (11)

\[ \Delta'_k(e) = g_k(M' \cup \{e\}) - g_k(M'). \] \hspace{1cm} (12)

For super modularity to hold, we must have $\Delta_k(e) \leq \Delta'_k(e)$ for all $e$. However, as the example shows, in a certain set of $e$ we have that $\Delta'_k(e) < \Delta_k(e)$ for $k = 1, 2$.

D. 123 Test Feeder

As presented in Section I we introduced a method of transforming the IEEE 123 test feeder into a reduced feeder (Figure 1(b)) and an island graph (Figure 2(a)).

The results we present allow us some insight into the topology detection problem. From Theorem 1 the minimum number of flow measurements is $\mu(G) = 4$. Since the actual set of allowable trees is $T_\tau(G)$, this restricts the set of trees to search over, in evaluating $g_1(M), g_2(M)$.

The set of valid placements is restricted since it cannot have edges in $\tau$. These edges have no physical meaning. The restricted set is given by $M_\tau(G) = \{f(T)|T \in T_\tau(G)\}$. We can now evaluate Algorithm 1 with the augmented sets, $M_\tau(G)$ and $T_\tau(G)$.

For simplicity assume a single phase flow of only real power. Given the three phase complex power consumption of each load, we reduce this to a single value power value. This is the case, if the measurement on a line sum and report the three phases. For a rough estimate of the forecast error, we follow the forecasting model in [21] where the day ahead forecasts are computed for varying aggregation levels. The forecast errors are used to construct the following scaling law for coefficient of variation:

\[ CV(W) = \sqrt{3562/W} + 41.9. \] \hspace{1cm} (13)

This gives us a realistic understanding of distribution system forecasting errors. Since the loads of each individual island is quite large and beyond the critical load reported in [21], the CV of each island is close to the irreducible error of 6.3%.

Figure 6 shows the performance of an initial stochastic placement for the IEEE123 test feeder. We evaluate the set of restricted placements $M_{\tau}(G)$ and spanning trees $T_{\tau}(G)$, where $|{\cal M}_{\tau}(G)| = 44$. Notice that for almost half of the placement the maximum error is negligibly small. Since the pairwise error probabilities are computed via monte carlo, they are small positive values that experimentally evaluate to 0.

VIII. Conclusion

We formulate the problem of detecting switch configurations on residential feeders as a spanning tree detection problem on an 'extended island graph'. The detection problem relies on power flow measurements on edges as well as load information at nodes. The deterministic case leads to notion of tree identifiability and sensor placement conditions.
to ensure identifiability. For the stochastic case, we propose a greedy algorithm which leads to a locally optimal solution.

REFERENCES

[1] J. Lavaei, D. Tse, and B. Zhang, “Geometry of power flows in tree networks,” in Power and Energy Society General Meeting, 2012 IEEE. IEEE, 2012, pp. 1–8.

[2] A. Lam, B. Zhang, and D. Tse, “Distributed algorithms for optimal power flow problem,” arXiv preprint arXiv:1109.5229, 2011.

[3] M. Farivar, R. Neal, C. Clarke, and S. Low, “Optimal inverter var control in distribution systems with high pv penetration,” in Power and Energy Society General Meeting, 2012 IEEE. IEEE, 2012, pp. 1–7.

[4] A. Lam, A. Dominguez-Garcia, B. Zhang, and D. Tse, “Optimal distributed voltage regulation in power distribution networks,” Tech. Rep., 2012.

[5] P. Jahangiri and D. C. Aliprantis, “Distributed volt/var control by pv inverters,” Power Systems, IEEE Transactions on, vol. 28, no. 3, pp. 3429–3439, 2013.

[6] J. Smith, W. Sunderman, R. Dugan, and B. Seal, “Smart inverter,” in Power and Energy Society General Meeting, 2012 IEEE. IEEE, 2012, pp. 1–8.

[7] A. Baran, M. Gahr, A. von Meier, G. Cavraro, M. Ruh, and G. Andersson, “Topology detection in microgrids with micro-synchrophasors,” arXiv preprint arXiv:1502.06938, 2015.

[8] W. H. Kersting, Distribution system modeling and analysis. CRC press, 2012.

[9] G. N. Korres and N. M. Manousakis, “A state estimation algorithm for monitoring topology changes in distribution systems,” in Power and Energy Society General Meeting, 2012 IEEE. IEEE, 2012, pp. 1–8.

[10] R. Arghandeh, M. Gahr, A. von Meier, G. Cavraro, M. Ruh, and G. Andersson, “Topology detection in microgrids with micro-synchrophasors,” arXiv preprint arXiv:1502.06938, 2015.

[11] M. Ruh, A. Borer, and G. Andersson, “Condensed grid topology detection for a fully transparent distribution management system.”

[12] Y. Gao, Z. Zhang, W. Wu, and H. Liang, “A method for the topology identification of distribution system,” in Power and Energy Society General Meeting (PES), 2013 IEEE. IEEE, 2013, pp. 1–5.

[13] M. Baran, J. Jung, and T. McDermott, “Topology error identification using branch current state estimation for distribution systems,” in Transmission & Distribution Conference & Exposition: Asia and Pacific. 2009. IEEE, 2009, pp. 1–4.

[14] Y. Sharon, A. M. Annaswamy, A. L. Motto, and A. Chakraborty, “Topological identification of power distribution network with limited measurements,” in Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES. IEEE, 2012, pp. 1–6.

[15] S. Bolognani, N. Bof, D. Michelotti, R. Muraro, and L. Schenato, “Identification of power distribution network topology via voltage correlation analysis,” in Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on. IEEE, 2013, pp. 1659–1664.

[16] M. C. Deepjyoti Deka, Scott Backhaus, “Structure learning and statistical estimation in distribution networks - part i,” http://arxiv.org/abs/1501.04131, 2015.

[17] ———, “Structure learning and statistical estimation in distribution networks - part ii,” http://arxiv.org/abs/1502.07820, 2015.

[18] R. Diestel, Graph Theory (Graduate Texts in Mathematics; 173). Springer-Verlag Berlin and Heidelberg GmbH & Company KG, 2000.

[19] H. N. Gabow and E. W. Myers, “Finding all spanning trees of directed and undirected graphs,” SIAM Journal on Computing, vol. 7, no. 3, pp. 280–287, 1978.

[20] G. L. Nemhauser and L. A. Wolsey, Integer and combinatorial optimization. Wiley New York, 1988, vol. 18.

[21] R. Sevlian, S. Patel, and R. Rajagopalan, “Distribution system load and forecast model,” arXiv preprint arXiv:1407.3322, 2014.

APPENDIX

A. Proof of Proposition 2

Proof: We have that

\[ \mu(G \setminus E(M)) = |E| - |V| + n(G \setminus M) \]  \hspace{1cm} (14)

\[ = |E| - |V| + 1 = 0 \]  \hspace{1cm} (15)

Therefore, \(|V| = |E| + 1\) with a single component making it a spanning tree on \(G\), where no cycles can be formed. ■

B. Proof of Theorem 1

In order to prove Theorem 1, we first introduce a representation of a spanning tree in terms of an arbitrary cycle \(c \in C(G)\).

1) Per Cycle Representation of any Spanning Trees: Every spanning tree \(T\) is defined by the set of edges in the the tree, \(E(T)\) or the set of cuts \(\Delta = \{\delta_1, \ldots, \delta_m\}\). We propose a representation of a spanning tree from the perspective of an arbitrary cycle \(c \in C(G)\) and the set of cuts \(\delta_1, \delta_N\) on the cycle. Since we can use any cycle, we can put any spanning tree \(T\) and \(c\) in this form.

A general representation is shown in Figure [(a)](a). There are \(N\) incoming flows from the upstream subgraph \(G_U\) and \(M\) outgoing flows from the downstream subgraph \(G_D\). There are cuts \(\delta_1, \ldots, \delta_N\) on \(c\) and \(\delta_{N+1}, \ldots, \delta_{N+M}\), \(N + M = \mu(G)\). Note that the vertices are omitted in the graph so to have an uncluttered illustration.

**Proposition 3:** This representation imposes the following requirements in order to maintain the graph structure:

1) Subgraph \(G_U\) has \(\mu(G_U) = N - 1\). Since we consider only planar graphs, these are faces \(f_1 \ldots f_N\). The addition of \(c\) leads to \(\mu(G_U \cup c) = N\).

2) All cuts \(\delta_1, \ldots, \delta_N\) are on faces \(f_1 \ldots f_N\) such that \(\delta_1 \subseteq f_i \cap c\), therefore they 'sandwiched' between upstream flows.

Some comments:

1) If face \(f_i\) has no cut, then a cycle exists, violating the spanning tree assumption.

2) If face \(f_i\) has a cut, but else where \(\delta_i \notin f_i \cap c\) we can redefine the network so that (1) \(G_U\) to have one less face (and upstream flow) (2) downstream subgraph \(G_D\) will have one additional downstream tree.

Using this representation we can prove Theorem 1.
2) Necessity: To show necessity, we must show that if \( \mathcal{M} \) does not cover \( C(G) \) we can always construct two set of cuts \( \Delta \) and \( \Delta' \) where \( s = s' \). Given an unobserved cycle, we only need to show we can always construct two trees \( T_1 \neq T_2 \) where \( s_1 = s_1 \).

Consider the following cycle representation in Figure 8(a).

As long as there is no flow measured on \( c \), any of the cuts on this cycle will map to the same observed flow. So, \( \Delta \neq \Delta' \) but \( s = s' \), can then construct a number of spanning trees on \( G \), since the measured flows no longer depend on cuts \( \delta, \delta' \).

3) Sufficiency: The sufficiency we must show that if \( \mathcal{M} \) covers \( C(G) \) there can never exists a set of cuts \( \Delta, \Delta' \) where \( s = s' \).

This is done inductively, where Theorem 1 is shown for \( G_U \) when \( \mu(\Gamma_U) = 1 \).

- Inductive Hypothesis: If \( \mathcal{M}_U \) covers \( C(G_U) \) and \( \mathcal{T}(G_U) \) is identifiable then \( \mathcal{M}_U \cup s_N \) which by construction covers \( C(G) \) leads to \( \mathcal{T}(G_U \cup c) \) being identifiable.

Here the inductive hypothesis is for graphs were \( \mu(G_U) = N - 1 \) and showing the addition of cycle leads to the property holding for \( \mu(G_U \cup c) = N \).

Consider the algebraic definition for identifiability in \( \mathcal{T}(G) \) as in eq. 1 and with only positive consumption, we have that \( \forall T, T' \) where \( T \neq T' \), \( \Gamma(T, M) - \Gamma(T', M)x \neq 0 \). We will abuse this notation by suppressing the parameters and use \( (\Gamma - \Gamma')x \neq 0 \).

We can then partition the vertices of the graph into the following sets/vectors.

1) Vertex set \( V_A = V(G_U \setminus c) \), consumption vector \( x_A \).
2) Vertex set \( V_B = V(G_U \cap c) \), consumption vector \( x_B \).
3) Vertex set \( V_C = V(c \setminus G_U) \), consumption vector \( x_C \).

First consider only the observations in \( G_U \):

\[
s_U = \Gamma_U x_U = \begin{bmatrix} \Gamma_A & \Gamma_B \end{bmatrix} \begin{bmatrix} x_A \\ x_B \end{bmatrix} \tag{16}
\]

Matrix \( \Gamma_A \) represents the contribution of vertices in \( V_A \) to the measurements in \( G_U \). Similarly, \( \Gamma_B \) represents the contribution of vertices in \( V_B \) to the measurements in \( G_U \).

From inductive hypothesis, we can algebraically state the identifiability of \( \mathcal{T}(G_U) \) as:

\[
(\Gamma_A - \Gamma_A')x_A + (\Gamma_B - \Gamma_B')x_B \neq 0. \tag{18}
\]

From Proposition 3(1), any \( \Delta, \Delta' \) which generates different spanning trees involve modification of the cuts along \( c \). This means that \( (\Gamma_A - \Gamma_A') = 0 \) for all the spanning trees we consider, leaving us with \( (\Gamma_B - \Gamma_B')x_B \neq 0 \).

Now consider the addition of cycle \( c \) to form \( G_U \cup c \). If \( c \) is not observed we showed in the counterexample in Section B.2 that \( \exists \mathcal{T}, \mathcal{T}' \) where \( s = s' \).

Algebraically, this is equivalent to:

\[
(\Gamma_A - \Gamma_A')x_A + (\Gamma_B - \Gamma_B')x_B + (\Gamma_C - \Gamma_C')x_C = 0 \tag{19}
\]

or

\[
(\Gamma_C - \Gamma_C')x_C = -(\Gamma_B - \Gamma_B')x_B \neq 0. \tag{20}
\]

The finally inequality comes from eq. 18. So if we only rely on the sensors \( s = \{s_1 \ldots s_{N-1}\} \) we can have multiple spanning trees that are indistinguishable if they satisfy the relation in eq. 20. Consider now the addition of a single measurement such that \( \mathcal{M} \cup s_N \) covers \( G_U \cup c \). We want to show that the addition of this measurement ensures that there cannot exist two spanning trees which map to identical measurements.

Extending the algebraic model for any \( \mathcal{T} \in \mathcal{T}(G_U \cup c) \) where \( \exists \Gamma_A, \Gamma_B, \Gamma_C, \gamma_A, \gamma_B, \gamma_C \) s.t.

\[
\begin{bmatrix} s \\ s_N \end{bmatrix} = \begin{bmatrix} \Gamma_A & \Gamma_B & \Gamma_C \\ \gamma_A & \gamma_B & \gamma_C \end{bmatrix} \begin{bmatrix} x_A \\ x_B \\ x_C \end{bmatrix}
\]

**Proposition 4:** Under this definition however, the following must be true.

1) \( \gamma_A = 0 \forall \mathcal{T} \);
2) \( \mathcal{M}_N \in V_C, \Delta_B = 0 \Delta_C \neq 0, \forall \delta_N \);
3) \( \mathcal{M}_N \in V_B, \Delta_B \neq 0 \Delta_C = 0, \forall \delta_N \)

We can finally prove the inductive step separately for \( \mathcal{M}_N \in V_C \) and \( \mathcal{M}_N \in V_B \).

\[
\mathcal{M}_N \in V_C: \text{ For a particular } \Delta, \Delta' \text{ we have that:}
\]

\[
(\Gamma_C - \Gamma_C')k = \begin{cases} 
\gamma_C - \gamma_C', & s_k \text{ in path from } s_N \text{ to root} \\
0, & \text{else} 
\end{cases}
\tag{21}
\]

for \( k \)th row of matrix \( (\Gamma_C - \Gamma_C') \). This is true, since the \( k \)th sensor measurement will always measure the value at \( s_N \) and some constant load, i.e. the nodes along the path from \( s_k \) and \( s_N \).

1) Assume \( \Delta, \Delta' \) with at least 1 sensor along the path from \( s_N \) to the root node. If we have \( \mathcal{T}, \mathcal{T}' \) where \( (\Gamma_C - \Gamma_C') \).
\[\Gamma_C'x_C = -(\Gamma_B - \Gamma_B')x_B \text{ like in eq. 20} \]

We are left with:

\[
\begin{bmatrix}
  s - s' \\ s_N - s'_N
\end{bmatrix} =
\begin{bmatrix}
  0 & \Gamma_B - \Gamma_B' & \Gamma_C - \Gamma_C' \\
  0 & 0 & \gamma_C - \gamma_C'
\end{bmatrix}
\begin{bmatrix}
  x_A \\ x_B \\ x_C
\end{bmatrix}
\]

Which leaves \( s - s' = (\Gamma_B - \Gamma_B')x_B + (\Gamma_C - \Gamma_C')x_C = 0 \).

So we fooled sensors \( s_N \). But this means that \((\Gamma_C - \Gamma_C')x_C \neq 0\), specifically \((\Gamma_C - \Gamma_C')x_C \neq 0\) for some \( k \). As a result, we have that \((\gamma_C - \gamma_C')x_C \neq 0\) or \( s_N - s'_N \neq 0 \).

(2) Assume no \( s_k \) along the path from \( s_N \) to the root.

Then \((\Gamma_B - \Gamma_B')x_B = 0\), therefore the only modification can happen on \( V_C \). However since \( e \) is covered, \((\gamma_C - \gamma_C')x_C \neq 0\).

\[ M_N \in V_B \] This follows an identically with the previous case and is included for completeness.

Again, we can show that for the \( k \)-th row of matrix \((\Gamma_B - \Gamma_B')\) we have:

\[ (\Gamma_B - \Gamma_B')k = \begin{cases} 
\gamma_B - \gamma_B', & s_k \text{ in path from } s_N \text{ to root } \\
0, & \text{else}
\end{cases} \]

Under \( \Delta, \Delta' \) if there is 1 sensor in the path from \( s_N \) to the root node. We again have the case \( s - s = (\Gamma_B - \Gamma_B')x_B + (\Gamma_C - \Gamma_C')x_C = 0 \). This implies that if \((\Gamma_B - \Gamma_B')x_B \neq 0 \) then \( \gamma_B - \gamma_B' \neq 0 \).

As before, if there is no other sensor in the path from \( s_N \) to the root node, this means that \((\Gamma_B - \Gamma_B')x_B = 0 \). This implies that \((\Gamma_C - \Gamma_C')x_C = 0 \). If the two trees are in fact different we are only left with \( \gamma_B - \gamma_B' \neq 0 \), which means that \( s_N - s'_N \neq 0 \).

\[ \mathcal{M}' = \begin{bmatrix} s \\ \mathcal{H} = \mathcal{T}_i \sim N\left( \begin{bmatrix} \mu_i \\ \Sigma_{ii} \end{bmatrix}, \begin{bmatrix} \sigma_{ii} & \sigma_{i+} \\ \sigma_{+i} & \sigma_{++} \end{bmatrix} \right) \end{bmatrix} \]

(23)

Conditioning on \( s_+ \) leads to \( \mathcal{H} = \mathcal{T}_i, s_+ \sim N(\mu, \Sigma_{i|s'}) \) where \( \Sigma_{i|s'} \leq \Sigma_i \).

This results in the following inequality, for any region \( R \)

\[ E_{s_+} \left[ I_{s \in R} \right] \leq E_{s} \left[ I_{s \in R} \right] \]

(24)

From Neyman-Pearson Lemma, for a fixed false positive \( \alpha \), the minimum false negative is given by the acceptance region \( R_{ij} \).

\[ Pr(T = T_j | T_i; \mathcal{M}') = E_{s_+} \left[ I_{s \in R_{ij}} \right] \]

(25)

\[ \leq E_{s} \left[ I_{s \in R_{ij}} \right] \forall R_{ij} \]

(26)

\[ = Pr(T = T_j | T_i; \mathcal{M}) \text{ for some } R_{ij} \]

(27)

(28)

Here, \( R_{ij} \) is the region which guarantees a false positive rate \( \beta \) for the detection problem \( |\mathcal{M}'| \) observations.

E. Proof of Theorem 3

Proof: We simply use the inequality in Lemma 1.

\[ g_2(\mathcal{M}') = \sum_{T_i \in \mathcal{T}(G)} \Pr(T_i) \Pr(\hat{T} \neq T_i | T; \mathcal{M}') \]

(29)

\[ = \sum_{T_i \in \mathcal{T}(G)} \Pr(T_i) \sum_{T_j \in \mathcal{T}(G) \setminus T_i} \Pr(\hat{T} = T_j | T; \mathcal{M}') \]

(30)

\[ \leq \sum_{T_i \in \mathcal{T}(G)} \Pr(T_i) \sum_{T_j \in \mathcal{T}(G) \setminus T_i} \Pr(\hat{T} = T_j | T; \mathcal{M}) \]

(31)

\[ = g_2(\mathcal{M}) \]

(32)

An identical proof is requires for \( g_2(\mathcal{M}) \) and but is omitted.

D. Proof of Lemma 7

Proof: Given measurement set \( \mathcal{M} \) we have that \( s_1 | \{ H = T_i \} \sim N(\mu_i, \Sigma_i) \), the addition of a measurement will yield