Detection of Islands in Power Networks

Mirko Todorovski, *Member, IEEE* and Risto Ačkovski, *Member, IEEE*

***Abstract*—In this letter, we present fast and simple method for island detection in power networks based on the bus admittance matrix. We also give complete Matlab code featuring identification of bus and branch indices in each detected island.**

***Index Terms*—Island formation, bus admittance matrix.**

I. Introduction

THE detection of island formation in power networks is essential for wide range of power system studies including reliability and security analysis, as well as load flow and associated optimization problems. Detection methods are based on numerical procedures [1] or search algorithms [2]. In [1] a sequence of multiplications of the network node-to-node connectivity matrix is used and it requires optimal bus ordering to reduce computation time. A fast method to check whether a network is split is given in [2] where only information about the off-diagonal nonzero elements in the bus admittance matrix is used without any computation. In this letter, a new fast island detection method based on the bus admittance matrix is proposed. The main features include: (i) identification of bus and branch indices in each detected island, (ii) simple implementation and (iii) high speed. The method was successfully tested on large-scale system with 21,880 buses and 22,603 branches.

II. Island Detection Using Network Incidence Matrix

A. Network Representation

Let us consider a simple network whose directed graph is given in Fig. 1. Buses and branches in the network are numbered using consecutive numbering with numbers 1-7 for buses and encircled numbers 1-7 for branches. Branch orientation is marked with an arrow. We denote number of buses and branches with N_B and N_L respectively.

Directed graph can be fully described using two vectors $\boldsymbol{F}$ and $\boldsymbol{T}$ (of length N_L) containing indexes of *from* and *to* ends of a branch respectively. For the network in Fig. 1 these vectors are $\boldsymbol{F} = [1\ 4\ 4\ 7\ 1\ 1\ 5]^\mathrm{T}$ and $\boldsymbol{T} = [2\ 3\ 5\ 6\ 7\ 6\ 3]^\mathrm{T}$.

A graph may also be described in terms of a connection or incidence matrix $\boldsymbol{A}$ which holds information on branch-to-node connections. It has one row for each branch and one column for each node with an entry A_{ij} according to the following rule [3]:

M. Todorovski and R. Ačkovski are with the Faculty of Electrical Engineering and Information Technologies, University Ss. Cyril and Methodius, Skopje, Republic of Macedonia (e-mail: acko@feit.ukim.edu.mk; mirko@feit.ukim.edu.mk).

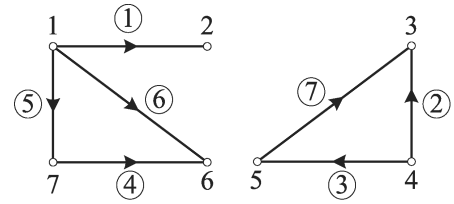

Fig. 1. Directed graph of a simple network

$$A_{ij} = \begin{cases} 0 & \text{if branch } i \text{ is not connected to node } j, \\ 1 & \text{if branch } i \text{ is directed away from node } j, \\ -1 & \text{if branch } i \text{ is directed toward node } j. \end{cases} \tag{1}$$

Using sparse storage and matrix operations, which are readily available in MATLAB™ or GNU Octave, and introducing vector $\boldsymbol{E}$ of length N_L whose elements are all equal to 1 we can write a simple relation between incidence matrix $\boldsymbol{A}$ and vectors $\boldsymbol{F}$ and $\boldsymbol{T}$ in the following way:

```
A = sparse(1:NL,F,E,NL,NB)-sparse(1:NL,T,E,NL,NB);
```

The bus admittance matrix $\boldsymbol{Y}$ is obtained with the incidence matrix $\boldsymbol{A}$ and the primitive admittance matrix $\boldsymbol{Y}_\mathrm{p}$ as

$$\boldsymbol{Y} = \boldsymbol{A} \cdot \boldsymbol{Y}_\mathrm{p} \cdot \boldsymbol{A}^\mathrm{T}, \tag{2}$$

where where the cardinality of $\boldsymbol{Y}$ and $\boldsymbol{Y}_\mathrm{p}$ are $N_\mathrm{B} \times N_\mathrm{B}$ and $N_\mathrm{L} \times N_\mathrm{L}$ [3].

In networks without mutually coupled branches, which is also the case here, the primitive admittance matrix is a diagonal with elements on the diagonal equal to corresponding branch admittances. Let all branches be comprised of serial resistors with resistance of 1 Ω, in which case $\boldsymbol{Y}_\mathrm{p}$ is identity matrix and (2) becomes

$$\boldsymbol{Y} = \boldsymbol{A} \cdot \boldsymbol{A}^\mathrm{T}. \tag{3}$$

B. Island Detection

With the bus admittance matrix we can write a simple matrix equation connecting bus voltages $\boldsymbol{V}$ and bus current injections $\boldsymbol{I}$:

$$\boldsymbol{Y} \cdot \boldsymbol{V} = \boldsymbol{I}, \tag{4}$$

where $\boldsymbol{V}$ and $\boldsymbol{I}$ are vectors of length N_B.

The idea of island detection is based on (4) using the following obvious fact: if there is a single current injection in the whole network at bus i then there will be non-zero voltage at all buses which have at least one direct/indirect connection with bus i.

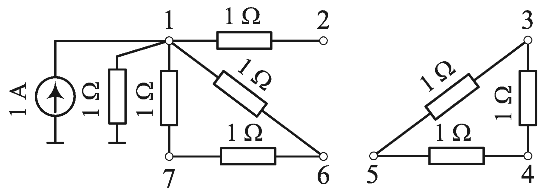

Fig. 2. Circuit equivalent for island detection

The problem arises when one has to solve (4) under these specific current injection conditions. Namely, the bus admittance matrix calculated with (3) is singular since it has no connection with the reference bus, i.e. with the ground. To overcome the singularity problem we have to ground bus i and solve the system (4) afterwards. This is done with a resistor of 1 Ω as shown in Fig. 2 for bus 1. The bus admittance submatrix for buses 3, 4 and 5 is still singular, however there no current injections in these buses and the only viable solution is all zero voltages at these buses. To be on the safe side and be sure that underlying linear algebra algorithms will deal with this case appropriately we suggest that big resistors of 1 MΩ are connected between each bus and the ground. These resistors are included in $\boldsymbol{Y}$ by increasing its diagonal elements by 10^{-6} which will negligibly influence the solution of (4).

When detecting islands in power networks we are not only interested in discovering if there is more than one island. It is useful to have information about the bus and branch indices in each detected island. Therefore, at the beginning we define vector $\boldsymbol{IB}$ which will contain the island index for each bus and similarly vector $\boldsymbol{IL}$ for branches.

The island detection procedure can be summarized in the following steps:

1) Set the number of islands N_{I} to zero. Set all elements of $\boldsymbol{IB}$ to zero (zero island index means that the bus is not classified into any island);
2) Create vector $\boldsymbol{IBX}$ with indices of zero valued elements from vector $\boldsymbol{IB}$, i.e. indices of non-classified buses;
3) Set $k = IBX(1)$, meaning that k is index of the first bus which is still not classified in any island;
4) Ground bus k with a resistance of 1 Ω (increase Y_{kk} by 1), set all elements of $\boldsymbol{I}$ to zero except I_k = 1 A;
5) Solve for voltages in (4) and find the buses with non-zero voltages.
6) Increase the number of islands N_{I} by 1 and put the value of N_{I} into $\boldsymbol{IB}$ at positions corresponding to buses with non-zero voltages from step 5;
7) Create vector $\boldsymbol{IBX}$ with indices of zero valued elements from vector $\boldsymbol{IB}$. If vector $\boldsymbol{IBX}$ is not empty go to step 3, otherwise go to step 8.
8) Create vector $\boldsymbol{IL}$ whose elements are $IL_i = IB(F_i)$, since each branch is a member to the same island as its from bus.

C. Matlab Code

In this section we present Matlab code snippet for islands detection procedure, which also works under GNU Octave without modifications. It requires that vectors $\boldsymbol{F}$ and $\boldsymbol{T}$ are predefined (for example as given in II.A), as well as number of buses and branches (N_{B} = 7 and N_{L} = 7 for the network in Fig. 1). The code strictly follows steps 1-8 of II.B. In addition, two cell arrays are defined denoted with $\boldsymbol{ISB}$ and $\boldsymbol{ISL}$. Number of elements in both of them is N_{I} and the elements are vectors containing bus and branch indices in each detected island, respectively.

```
E = ones(NL,1);
A = sparse(1:NL,F,E,NL,NB)-sparse(1:NL,T,E,NL,NB);
Y = A'*A + 1e-6*speye(NB);
IB = zeros(NB,1);
NI = 0;
IBX = find(IB == 0);
while ~isempty(IBX)
   k = IBX(1);
   Y(k,k) = Y(k,k) + 1;
   I = zeros(NB,1); I(k) = 1;
   V = Y\I;
   ind = find(V > 0.001);
   NI = NI + 1;
   IB(ind) = NI;
   IBX = find(IB == 0);
end
IL = IB(f);
ISB = cell(NI,1); ISL = cell(NI,1);
for i = 1:NI
    ISB{i} = find(IB == i);
    ISL{i} = find(IL == i);
end
```

As output from the above code, for the network from Fig. 1, we obtain the following: N_{I} = 2, $\boldsymbol{IB} = [1\ 1\ 2\ 2\ 2\ 1\ 1]^{\mathrm{T}}$, $\boldsymbol{IL} = [1\ 2\ 2\ 1\ 1\ 1\ 2]^{\mathrm{T}}$, $\boldsymbol{ISB}\{1\} = [1\ 2\ 6\ 7]^{\mathrm{T}}$, $\boldsymbol{ISB}\{2\} = [3\ 4\ 5]^{\mathrm{T}}$, $\boldsymbol{ILB}\{1\} = [1\ 4\ 5\ 6]^{\mathrm{T}}$ and $\boldsymbol{ILB}\{2\} = [2\ 3\ 7]^{\mathrm{T}}$. It is evident that both islands in the network are detected and we have information that the first island consist of buses 1, 2, 6 and 7 ($\boldsymbol{ISB}\{1\}$) and branches 1, 4, 5 and 6 ($\boldsymbol{ISL}\{1\}$). The second island consist of buses 3, 4 and 5 ($\boldsymbol{ISB}\{2\}$) and branches 2, 3 and 7 ($\boldsymbol{ISL}\{2\}$).

The presented procedure is very fast and requires little memory since it works with sparse matrices. We have performed tests on bigger networks using laptop with a 2.5-GHz Intel Core i5 processor running Matlab 7.9. The timing in the smaller cases, with up to 3,000 buses, was a fraction of a millisecond per detected island with total runtime about a millisecond. The biggest system we have analyzed is a distribution network spanning over several cities with N_{B} = 21,880 and N_{L} = 22,603. It has 69 islands and the total runtime was 0.97 seconds, which in average is 14 milliseconds per detected island.

III. Conclusion

A new method for detection of islands in power networks has been developed in this work. The method is based on bus admittance matrix and uses appropriate bus current injections to detect islands and identify its buses and branches. The method is tested on large network and complete Matlab code is given.

References

[1] F. Goderya, A. A. Metwally, O. Mansour, "Fast Detection and Identification of Islands in Power Networks," IEEE Transactions on Power Apparatus and Systems, vol. PAS-99, no. 1, pp. 217-221, Jan. 1980.

[2] D. Rajicic, “A Method for Determination Whether a Network Is Connected or Split,” IEEE Power Engineering Review, vol. 21, no. 7, pp. 65-66, July 2001.

[3] J. J. Grainger, and W. D. Stevenson, Power System Analysis, International Edition, New York: McGraw-Hill, 1994, pp. 257-263.