23rd International Symposium on Mathematical Theory of Networks and Systems
Hong Kong University of Science and Technology, Hong Kong, July 16-20, 2018

A Scheme for Computing the Transfer Function of Power Grids with Grounded Capacitors*

Mark Jeeninga[1] Claudio De Persis[2] Arjan J. van der Schaft[3]

Abstract— **We consider a simple model for power grids with grounded unitary capacitors at the loads, and look at the computation of the transfer function from generator potentials to generator currents. We propose an alternative scheme for computing this transfer function.**

I. INTRODUCTION

Electrical networks such as power grids can be modelled by considering dynamics over a weighted graph. The nodes in such a graph correspond with the buses in the network, whereas the edges represent the lines. The conductances of the lines correspond to the weights of the edges.

In this work we are interested in exploiting the underlying graph structure in the study of such models. For a detailed analysis of the interconnection between such dynamics and (algebraic) graph theory we refer to [1]. See also [6] for relevant results on resistive circuits and graphs.

We consider two types of buses in such models: *load buses*, which drain power from the grid, and *generator buses*, which provide power to the grid. The associated graph typically tends to be sparse, as seen in most benchmark models.

In this paper we propose a scheme for computing the transfer function from the potentials to the currents at generator buses in a power grid with grounded unitary capacitors. Our approach makes use of computing a resolvent in an element-wise fashion, which might perform faster compared to traditional approaches, especially if the degree of generator loads is low. However, we will not present any computational comparison in this paper.

The proposed scheme makes use of the weighted Laplacian matrices of the graph induced by load nodes, along with its connected subgraphs. In particular, it makes use of their relation to spanning tree numbers and the characteristic polynomials of closely related matrices.

*This work is supported by the NWO (Netherlands Organisation for Scientific Research) project 'Energy management strategies for interconnected smart microgrids' within the DST-NWO Joint Research Program on Smart Grids.

[1]Mark Jeeninga is with Engineering and Technology Institute Groningen, and Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, 9747AG Groningen, The Netherlands m.jeeninga@rug.nl

[2]Claudio De Persis is with Engineering and Technology Institute Groningen, University of Groningen, 9747AG Groningen, The Netherlands c.de.persis@rug.nl

[3]Arjan J. van der Schaft is with Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, 9747AG Groningen, The Netherlands a.j.van.der.schaft@rug.nl

II. PRELIMINARIES

Before being able to formulate the approach, we introduce the dynamics and introduce some notions from graph theory. We first describe a simple toy model for a power grid. Next we define some standard graph operations with respect to the node set of the graph.

A. Dynamics and Problem

Let V_L and V_G be the potentials at the load and generator buses respectively, I_G the outgoing current at the generator buses, and $\begin{pmatrix} \mathcal{L}_{LL} & \mathcal{L}_{LG} \\ \mathcal{L}_{GL} & \mathcal{L}_{GG} \end{pmatrix}$ the weighted Laplacian matrix representing the interconnection between buses over the lines. We consider the simple dynamical system described by

$$\dot{V}_L = \mathcal{L}_{LL}V_L + \mathcal{L}_{LG}V_G, \quad I_G = \mathcal{L}_{GL}V_L + \mathcal{L}_{GG}V_G,$$

which models a power grid where grounded capacitors of unit capacitance are connected to the loads.

The transfer function from the potentials at the loads V_G to the outgoing currents at the loads I_G is given by

$$H_G(s) := \mathcal{L}_{GG} - \mathcal{L}_{GL}(\mathcal{L}_{LL} - sI)^{-1}\mathcal{L}_{LG}.$$

The main issue is the computation of $(\mathcal{L}_{LL} - sI)^{-1}$, which is known as the *resolvent* of the linear operator $\mathcal{L}_{LL}$. A general method for computing this rational matrix in the Laplace variable s is by computing the spectral decomposition of the symmetric matrix $\mathcal{L}_{LL}$ [10].

Since the number of load nodes might be large, it might be desirable to avoid computing the spectral decomposition in favor of another approach. The scheme suggested in this paper might fill this gap.

B. Graph Theory

Let Γ denote a weighted undirected graph with node-set $N = \{1, \ldots, n\}$. For any graph Γ let L_Γ denote its corresponding Laplacian matrix. For any $S \subset N$, the complement of S in N is denoted by S^c.

A spanning tree of a graph Γ is a subgraph which is a tree and contains all nodes (buses) of the original graph. To each spanning tree we assign a weight, which is the product of the weight of the edges (lines) in the tree. The sum over the weights of all spanning trees is known as the *weighted spanning tree number* of the graph, here denoted by $\mathrm{t}(\Gamma)$.

We let $\Gamma \backslash S$ denote the subgraph of Γ obtained by removing the nodes indexed by $S \subset N$, along with their incident edges. In the literature $\Gamma \backslash S$ is better known as the subgraph induced by the vertex set S^c.

III. THE APPROACH

Let Γ be the weighted graph which describes the interconnection between the load nodes in our problem, and let L_Γ be the Laplacian corresponding to Γ. The weights again correspond to the conductances of the lines. This means that $\mathcal{L}_{LL} = L_\Gamma + D$ for some nonzero diagonal matrix $D = \operatorname{diag}(d_1, \ldots, d_n)$. In fact, we have $D = \operatorname{diag}(\mathcal{L}_{LL}\mathbb{1})$.

To obtain an expression for the inverse of $\mathcal{L}_{LL}$ we use the following formula, which relates a diagonal update of a Laplacian matrix to the weighted spanning tree numbers of the subgraphs of Γ:

$$\begin{aligned}\mathcal{L}_{LL}^{-1} &= (L_\Gamma + D)^{-1} = \frac{\operatorname{adj}(L_\Gamma + D)}{\det(L_\Gamma + D)} \\ &= \sum_{Q \subset N} \frac{\det(L_{\Gamma\backslash Q} + P_Q^{\mathrm{T}} D P_Q)}{\det(L_\Gamma + D)} \cdot \mathrm{t}(\Gamma\backslash Q^c) \cdot P_{Q^c}\mathbb{1}\mathbb{1}^{\mathrm{T}} P_{Q^c}^T. \end{aligned} \tag{1}$$

where P_S is the matrix obtained by removing the columns indexed by $S \subset N$ from I_n, the identity matrix. The matrix $P_Q^{\mathrm{T}} D P_Q$ is a principal submatrix of D.

The formula (1) was derived by the authors in a separate paper [5] which is currently being prepared for submission. As mentioned, the described method avoids computing the spectral decomposition of the Laplacian matrix of the graph. We observe that $(\mathcal{L}_{LL} - sI)^{-1}$ is obtained by performing the substitution $D \mapsto D - sI$ in (1).

Note that $(\mathcal{L}_{LL} - sI)^{-1}$ may be computed element-wise. We investigate the minimal amount of information we need to compute a single element of this resolvent. More specifically, we will investigate when such terms vanish.

We will proceed by looking at the evaluation of the factors

$$\mathrm{t}(\Gamma\backslash Q^c) \cdot P_{Q^c}\mathbb{1}\mathbb{1}^{\mathrm{T}} P_{Q^c}^T. \tag{2}$$

The factor $\mathrm{t}(\Gamma\backslash Q^c)$ is nonzero if and only if the graph induced by Q is connected. Since the graph is sparse, the number of spanning trees and connected subgraphs is relatively low. Note that the weighted spanning tree numbers may be computed via Kirchhoff's theorem, i.e. by computing a determinant. Such determinants can be computed using sparse LU decomposition methods. A conservative approximate of their complexity is $\mathcal{O}(n^3)$ [9], but is hard to analyze for general cases.

The (i, j)-th element of the matrix $P_{Q^c}\mathbb{1}\mathbb{1}^{\mathrm{T}} P_{Q^c}^T$ is nonzero if and only if $i, j \in Q$. Since the graph induced by Q is connected if (2) is non-zero, this means that nodes i and j should be path-connected, which is trivially satisfied if $i = j$. Furthermore, the intersection over the nodes of all paths between i and j should be contained in Q.

The denominator of (1) corresponds, after the substitution, to the characteristic polynomial of $\mathcal{L}_{LL}$. Similarly, the numerator of the same fraction corresponds to the characteristic polynomial of a diagonal update of a principal submatrix of $\mathcal{L}_{LL}$. Methods for computing the characteristic polynomial, such as Danilevsky's method [7], [8], can be used to compute these characteristic polynomials.

IV. THE SCHEME

The advantage of (1) is that is can be computed element-wise. The element-wise computation of $(\mathcal{L}_{LL} - sI)^{-1}$ is, to the authors' knowledge, not possible for the spectral decomposition. When considering the product with matrices $\mathcal{L}_{LG}$ and $\mathcal{L}_{GL}$, element-wise computation can be used to accelerate the computation of $H_G(s)$.

When assuming that the degree of the generator nodes is low, most loads are not directly connected to the generators. This means that most of the rows of $\mathcal{L}_{GL}$ are zero, and so that we only need to know a limited number of elements of $(\mathcal{L}_{LL} - sI)^{-1}$.

Indeed, if we the define the *boundary nodes* $B \subset N$ as the set of load nodes which are connected to a generator node, then $P_{B^c} P_{B^c}^{\mathrm{T}} \mathcal{L}_{LG} = \mathcal{L}_{LG}$, by definition of B, and it is sufficient to compute $P_{B^c} P_{B^c}^{\mathrm{T}} (\mathcal{L}_{LL} - sI)^{-1} P_{B^c} P_{B^c}^{\mathrm{T}}$.

Let $\mathcal{C}$ be the collection of sets of nodes for which the graph induced by this set is connected, and which contain at least one node in B. We propose the following scheme for computing $P_{B^c} P_{B^c}^{\mathrm{T}} (\mathcal{L}_{LL} - sI)^{-1} P_{B^c} P_{B^c}^{\mathrm{T}}$.

Algorithm 1 Computing $P_{B^c} P_{B^c}^{\mathrm{T}} (\mathcal{L}_{LL} - sI)^{-1} P_{B^c} P_{B^c}^{\mathrm{T}}$

INPUT: $\mathcal{L}_{LL} \in \mathbb{R}^{n \times n}$, boundary nodes $B \subset N$, sets of nodes $\mathcal{C}$ inducing a connected subgraph and containing at least one node in B
OUTPUT: $A = P_{B^c} P_{B^c}^{\mathrm{T}} (\mathcal{L}_{LL} - sI)^{-1} P_{B^c} P_{B^c}^{\mathrm{T}}$

```
Let A = 0 ∈ ℂ^{n×n}(s)
for all Q ∈ 𝒞 do
    Compute p_Q(s), the char. poly. of L_{Γ\Q} + P_Q^T D P_Q
    Compute t(Γ\Q^c)
    for all (i, j) ∈ B × B such that i, j ∈ Q do
        A_ij ← A_ij + p_Q(s) · t(Γ\Q^c)
    end for
end for
Compute p_∅(s), the characteristic polynomial of 𝓛_LL
A ← A / p_∅(s)
return A
```

For every pair of boundary nodes (i, j) the algorithm computes the polynomial

$$\sum_{\substack{Q \subset N \\ i,j \in Q}} \det(L_{\Gamma\backslash Q} + P_Q^{\mathrm{T}} D P_Q - sI_{|Q^c|}) \cdot \mathrm{t}(\Gamma\backslash Q^c),$$

which, after dividing by the characteristic polynomial of $\mathcal{L}_{LL}$, corresponds to $(\mathcal{L}_{LL} - sI)_{ij}^{-1}$. We may then proceed to compute $\mathcal{L}_{GL}(\mathcal{L}_{LL} - sI)^{-1}\mathcal{L}_{LG}$ and eventually $H_G(s)$.

The collection of sets $\mathcal{C}$ can be obtained by observing the following. If $S \in \mathcal{C}$ is a singleton, then S only consists of a boundary node, since all sets in $\mathcal{C}$ contain a boundary node. Let $S \in \mathcal{C}$ be not a singleton. For any spanning tree T of the graph induced by S, take i a leaf of T, which has at least two leaves since S is not a singleton. Then $T\backslash i$ is a spanning tree of $S\backslash i$, and $S\backslash i$ is connected. If $|S| = 2$ we can pick i such that $S\backslash i$ is a singleton of a boundary node, hence in $\mathcal{C}$. If $|S| > 2$ we pick i such that $B \cap S \backslash i \neq \emptyset$, and therefore also $S \backslash i \in \mathcal{C}$. We may repeat removing nodes in this fashion until we are left with a singleton.

This observation tells us that we can always remove a node i in the graph induced by $S \in \mathcal{C}$ such that $S \backslash i \in \mathcal{C}$, until we are left with a singleton of a boundary node. Inverting this process leads to Algorithm 2, where we iteratively add nodes to the singletons of boundary nodes to obtain all sets in $\mathcal{C}$. The algorithm ensures that every such set is appended to $\mathcal{C}$ exactly once, by keeping track of certain nodes to ignore in subsequent iterations. We will not elaborate on this here.

Algorithm 2 Computing $\mathcal{C}$

INPUT: $\Gamma = (N, E)$ with $E \subset N \times N$ the edge set, boundary nodes $B \subset N$
OUTPUT: $\mathcal{C}$

$\mathcal{C} \leftarrow \emptyset$
function ITERATE(S, T) ▷ selected and ignored nodes
    $U \leftarrow \{l \in N \backslash (S \cup T) \mid (i, l) \in E \text{ for some } i \in S\}$
    $W \leftarrow \emptyset$
    for all $u \in U$ **do**
        Append $S \cup \{u\}$ to $\mathcal{C}$
        ITERATE($S \cup \{u\}, T \cup W$)
        $W \leftarrow W \cup u$
    end for
end function
$V \leftarrow \emptyset$
for all $b \in B$ **do**
    Append $\{b\}$ to $\mathcal{C}$
    ITERATE($\{b\}, V$)
    $V \leftarrow V \cup b$
end for
return $\mathcal{C}$

V. CONCLUSION

We have introduced an alternative scheme for computing the transfer function $H_G(s)$ which exploits the sparse nature of power grids, under the assumption that the degree of generators is low. For this an element-wise formula for the resolvent $(\mathcal{L}_{LL} - sI)^{-1}$ was used. The authors are currently working on extending this research by performing a quantitative computational comparison between the proposed scheme and traditional approaches.

REFERENCES

[1] F. Dörfler, J. W. Simpson-Porco, F. Bullo, “Electrical Networks and Algebraic Graph Theory: Models, Properties, and Applications,” Proceedings of the IEEE, submitted.
[2] S. Sanchez, R. Ortega, G. Bergna, M. Molinas and R. Grio, ”Conditions for existence of equilibrium points of systems with constant power loads,” 52nd IEEE Conference on Decision and Control, Firenze, 2013, pp. 3641-3646.
[3] J. W. Simpson-Porco, “A Theory of Solvability for Lossless Power Flow Equations - Part I: Fixed-Point Power Flow,” in IEEE Transactions on Control of Network Systems, to appear.
[4] J. W. Simpson-Porco, “A Theory of Solvability for Lossless Power Flow Equations - Part II: Conditions for Radial Networks,” in IEEE Transactions on Control of Network Systems, vol. PP, to appear.
[5] M. Jeeninga, C. De Persis, A.J. van der Schaft, “A formula for the adjugate and determinant of the sum of two matrices – Resolvents, M-matrices and graph-theoretic consequences,” to be submitted.
[6] A.J. van der Schaft, ”Characterization and partial synthesis of the behavior of resistive circuits at their terminals”, Systems & Control Letters, vol. 59, pp. 423–428, 2010.
[7] M. Bujnak, Z. Kukelova and T. Pajdla, “Making minimal solvers fast,” 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012, pp. 1506-1513.
[8] A. M. Danilevskii, “The numerical solution of the secular equation” (Russian), Matem. Sbornik, 44(2), 1937.
[9] A. J. Schneider, “Estimate of the number of arithmetic operations required in the LU decomposition of a sparse matrix,” IEEE Trans. Circuit Theory, 17(2), 1970, pp. 269-270.
[10] C. Godsil, G. Royle, “Algebraic graph theory,” Springer, New York, 2001.
[11] B. Bollobás, “Modern graph theory,” Springer, New York, 1998.