Check for updates

Frequency Response and Transfer Functions of Large Self-Similar Networks

Xiangyu Ni
Department of Aerospace and Mechanical Engineering,
University of Notre Dame,
Notre Dame, IN 46556
e-mail: xni@nd.edu

Bill Goodwine[1]
Professor
Department of Aerospace and Mechanical Engineering,
University of Notre Dame,
Notre Dame, IN 46556
e-mail: jgoodwin@nd.edu

Large-scale dynamical systems, no matter whether possessing interconnected appearances, are frequently modeled as networks. For instance, graphs, multi-agent systems, and materials' intricate behaviors are often treated as networked dynamical systems. However, only a few studies have approached the problem in the frequency domain, mostly due to the complexity of evaluating their frequency response. That gap is filled by this paper, which proposes algorithms computing a general class of self-similar networks' frequency response and transfer functions, no matter they are finite or infinite, damaged or undamaged. In addition, this paper shows that for infinite self-similar networks, even when they are damaged, fractional-order and irrational dynamics naturally come into sight. Most importantly, this paper illustrates that for a network under different operating conditions, its frequency response would form a set of neighboring plants, which sets the basis of applying robust control methods to dynamic networks.
[DOI: 10.1115/1.4054645]

1 Introduction

Networked dynamical systems are ubiquitous, the study of which mainly arises from the following three aspects: graphs, multi-agent systems and materials' intricate behaviors. The study about graphs is often linked with graph theory, whose starting point is usually credited to Euler's solution to the Königsberg Bridge Problem [1]. According to a detailed review [2], the primary goal of this field is to understand the structure and function of complex networks, where the empirical studies of the former, the structure of real-life networks, are the outset of this research area. Those networks include social networks [3], information networks like the World Wide Web [4], technological networks like cellular networks [5], and biological networks [6]. The focus of those empirical studies at first was to quantify a number of statistical properties of networks being important for their function, which gradually unveiled that many statistical properties are commonly shared by networks from different domains. That observation of nonrandomness among networks from various origins later led to a number of abstract mathematical models, such as random graphs [7], Markov graphs [8], small-world models [9], and models of network growth [10]. The ongoing focus of this research domain mostly shifts to explanations of processes occurring on networks *via* those mathematical models, for example, search and navigation processes, and network transmission and epidemiology [11].

As for the researchers in robotics, the most heavily investigated topic about networked dynamical systems is multi-agent systems. The objective for this discipline is to design a strategy for two or more agents to achieve a global task in a cooperative manner [12]. One example global task is consensus problems where the states of all robots are asked to converge to the same value [13]. Multi-agent systems have at least two outstanding advantages over single-agent systems. First, multi-agent systems are more robust in the sense that if a few agents fail, the other agents could quickly adapt the situation to continue finishing the global task. Second, multi-agent systems can cover a much larger physical area simultaneously, or equivalently, execute an involved task in a temporally parallel manner. The above two benefits result in a wide spectrum of applications, such as search and rescue [14], distributed map merging [15], collective transport [16], clock synchronization [17], sensor fusion [18], localization [19], distributed support vector machine [20], and distributed air-conditioning optimization [21]. The continuing emphases include incorporation between network topologies and agent dynamics, rigid formations in three-dimensional space, and fully autonomous and distributed multi-agent systems.

Approximating intricate dynamical systems by using networks also attracts significant attention, which, however, can be easily neglected by researchers concentrating on networks since those systems usually do not possess networked appearances. For instance, a long electrical transmission line can be estimated by an electrical network where electrical properties like resistance are lumped at each section [22], which is often used to model railway track circuits with the purpose of detecting whether a certain portion of a track is occupied [23–25]. A closely related equivalent is the representation of a multistory building as a shear-frame structure where a pair of parallel spring and damper connecting two neighboring masses resembles the shearing motion between two adjacent floors, which is often employed in structural vibration control [26–29]. Another class of applications falling into this classification is about reproducing materials' complicated behavior through large-scale networks. For a wide range of composite materials, one prominent phenomenon is that the magnitude of their overall admittance is always in a power-law scaling relation with the frequency under the alternating current field within an intermediate frequency region, which is called the universal dielectric response, proposed by Jonscher [30,31]. To date, there is no agreement on the origins of such response. However, it is possible to reproduce that phenomenon by using random resistor–capacitor networks and show that the ratio of quantities between resistors and capacitors determines the slope of that power-law relation [32–34]. Interestingly, the same fact is also observed from random biphasic mechanical truss networks, which are constituted by springs with two different constants [35].

Some examples of using frequency-domain methods for dynamic networks include [36] which models and simulates power systems in the frequency domain. The paper [37] models multi-agent systems composed of distributed controllers in the frequency domain. The study [38] designs and analyzes a resilient consensus controller for multi-agent systems in the frequency domain. However, in contrast to plentiful works on dynamic networks from various disciplines aforementioned, the number of

[1]Corresponding author.

Contributed by the Dynamic Systems Division of ASME for publication in the Journal of Dynamic Systems, Measurement, and Control. Manuscript received March 4, 2021; final manuscript received May 16, 2022; published online June 15, 2022. Assoc. Editor: Sandipan Mishra.

Copyright © 2022 by ASME

Downloaded from http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/144/8/081007/6888118/ds_144_08_081007.pdf by University Of Notre Dame user on 13 July 2023

studies which apply frequency-domain methods to dynamic networks in literature is low. Moreover, those existing works often focus on a specific network, and none of them has considered modeling a general class of dynamic networks from the perspective of the frequency-domain, mostly due to the complexity of computing their frequency response. Therefore, there is a gap between available frequency-domain tools and the fact that those tools are rarely applied to large-scale networks. That gap is filled by this paper, which shows that problem is tractable once self-similarities are leveraged and that the resultant outcomes have potential to be further employed in dynamic networks' simulation, health monitoring and control. One mathematical novelty of this work is it provides specific examples, that is infinite dynamic networks, where fractional and irrational transfer functions naturally come to light. That offers possibilities for understanding the physical meaning of fractional-order derivatives and implicit operators in the future.

Fractional and irrational dynamics naturally emerge from complex systems and systems with memory. For instance, a previous paper from the second author [39] models unidirectional vibrations as fractional systems through infinite mechanical networks. Fractional-order controllers are implemented on irrational large-scale systems in Ref. [40]. The work [41] proposes general ways of approximating fractional dynamics to model power law type long memory behavior. The paper [42] studies Newton's second law where the external force has a memory effect through fractional integrations. The fact that infinite self-similar networks' transfer functions are very likely to be fractional or irrational as shown by this paper adds another example.

Specifically, this paper proposes four algorithms computing the dynamics of a general class of networked dynamical systems in the following four situations:

(1) Frequency response for finite networks,
(2) Transfer functions for finite networks,
(3) Frequency response for infinite networks, and
(4) Transfer functions for infinite networks.

Here, a *transfer function* refers to the analytical expression of $G(s)$ which describes a system's behavior through the ratio of the output signal to its input signal in the frequency domain. Its corresponding *frequency response* is obtained by sampling $G(s)$ at a sequence of angular frequencies ω, i.e., $G(j\omega)$.

The rest of this paper is structured as follows. Section 2 lists assumptions for that general class of networks to which the modeling methods in this paper are applicable. In addition, it also delineates the preliminaries such as notation used in this paper. Section 3 discusses recurrence formulas, which are the core of all modeling algorithms. Section 4 talks about the two algorithms regarding finite networks. Section 5 briefly introduces how to evaluate infinite networks' dynamics in a special case, which is also a foundation of calculating all infinite networks' frequency response and transfer functions in this paper as showcased by Sec. 6. Section 7 discusses the impact of knowing dynamic networks' frequency response and transfer functions from three aspects. Finally, Sec. 8 suggests future works and concludes this paper.

2 Assumptions and Preliminaries

For a network to be qualified for the approach proposed in this paper, it must satisfy the following assumptions.

(A-1) The network is one-dimensional.
(A-2) The network is self-similar. That is, its topology is invariant throughout all generations.
(A-3) All components within the network are connected either in series or in parallel.
(A-4) All components are linear and time-invariant, such as dampers, capacitors, or transfer function blocks.
(A-5) *For infinite networks only:* The network has a finite number of damaged components.
(A-6) *For infinite networks only:* The network's undamaged transfer function can be obtained.

Note that the first four assumptions from (A-1) to (A-4) are for both finite and infinite networks, whereas the last two assumptions (A-5) and (A-6) are for infinite networks only. The scope of this paper is limited to single-input, single-output, linear, time-invariant dynamical networks, which is the main reason behind the assumptions (A-1) and (A-4). Both assumptions (A-2) and (A-3) are related to whether we can deduce a dynamical network's recurrence formula, which is the key part of our modeling method as introduced in Sec. 3. The assumptions (A-5) and (A-6) characterize the infinite networks to which the proposed method can be applied. As explained in Secs. 5 and 6, if either of those two assumptions fails, the proposed method cannot be employed to compute that infinite network's transfer function or frequency response.

The definition of a damaged network and an undamaged network is established as follows: For instance, there is a network consisting of five springs and five dampers, denoted as k_1 to k_5 and b_1 to b_5. Each type of component has its undamaged constant, indicated by the undamaged spring constant k and the undamaged damper constant b. Then, when that network is undamaged, all components' constants are same as their corresponding undamaged ones, i.e., $\forall i = 1, \ldots, 5$, $k_i = k$, and $b_i = b$. Otherwise, that network is damaged, in which situation a pair of two lists $(\boldsymbol{l}, \boldsymbol{e})$ is employed to represent a specific damage case. The letter $\boldsymbol{l}$ is the list of damaged components, and $\boldsymbol{e}$ is the corresponding damage amounts. For example

$$(\boldsymbol{l}, \boldsymbol{e}) = ([k_1, b_2], [0.3, 0.4])$$

designates the damage case where $k_1 = 0.3k$ and $b_2 = 0.4b$, while all the other springs and dampers have unchanged stiffness and damping constants. Note that, in fact, $(\boldsymbol{l}, \boldsymbol{e})$ can also denote the undamaged case. In all modeling algorithms in this paper, the undamaged case is indicated by empty lists $\boldsymbol{l}$ and $\boldsymbol{e}$.

The nomenclature of frequency response and transfer function used in this paper is listed as follows:

$G(s)$ is a general transfer function in Laplace variable s.
$G(j\omega)$ is the corresponding frequency response sampled at a sequence of angular frequencies ω.
$G_r(s)$ is the recurrence formula for a self-similar network introduced in Sec. 3.
$G_{si}(s)$ is the transfer function for the ith subnetwork inside a self-similar network also introduced in Sec. 3.
$G_1(s)$ is the transfer function when a network only has one generation first used in Sec. 4.
$G_{g,(\boldsymbol{l},\boldsymbol{e})}(\cdot)$ is designated for a specific configuration of a network. The positive integer g denotes the number of generations inside a network. When that network is infinitely large, $g = \infty$. The pair of two lists $(\boldsymbol{l}, \boldsymbol{e})$ indicates a particular damage case. When that network is undamaged, $(\boldsymbol{l}, \boldsymbol{e})$ is simply replaced by $\emptyset$.

Next, the three example networks, which are used throughout this paper, are introduced to showcase that a vast range of networked dynamical systems could leverage the methods proposed in this dissertation. The first example is the *mechanical tree* network as shown in Fig. 1, which has been used to model the relaxation of the aortic valve [43] and materials' viscoelastic behaviors [44] in literature. In any numerical computations in this dissertation, the undamaged constants are assumed to be $k = 2\,\mathrm{N/m}$ and $b = 1\,\mathrm{Ns/m}$. The dynamics of interest is the ratio of its length $X_{1,1}$ to the force exerted at $x_{1,1}$, F, in the frequency domain.

The second example is the *electrical ladder* network as shown in Fig. 2, which is used to approximate electrical transmission lines in literature [22–25]. Its counterpart constructed by mechanical components is also utilized to approximate structures like buildings and bridges [26–29]. The undamaged constants are $r_1 = 10\,\Omega$, $r_2 = 1k\,\Omega$, and $c = 100\,\mu F$. When undamaged, $\forall i \in \mathbb{Z}^+$, $r_{i,1} = r_1$, $r_{i,2} = r_2$, and $c_i = c$. The dynamics of interest

Downloaded from http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/144/8/081007/6888118/ds_144_08_081007.pdf by University Of Notre Dame user on 13 July 2023

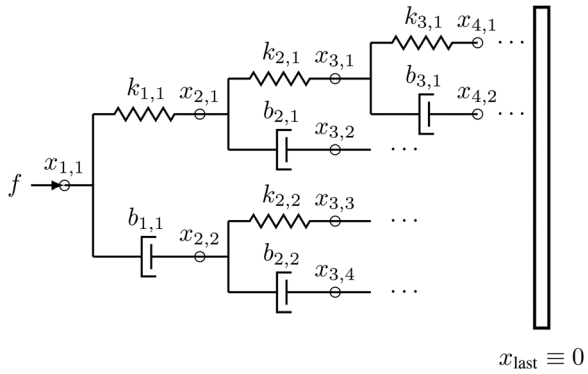

Fig. 1 ***Mechanical tree*** **network**

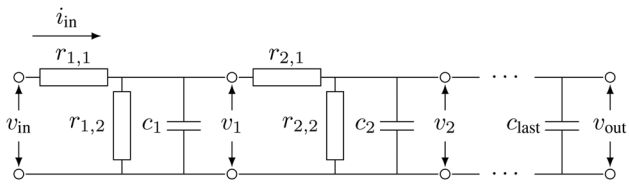

Fig. 2 ***Electrical ladder*** **network**

is the input impedance $V_{\text{in}}/I_{\text{in}}$ in the frequency domain. Note that in order to save some space, we ignore series inductance in this paper. An example with series inductance can be found in the first author's dissertation [45], where it is also found that our result converges to the result given by the telegrapher equations as the length of subnetworks decreases.

The third example is the *mechanical ladder* network as shown in Fig. 3. The significance of this example is that it includes nonzero masses and proportional-integral-derivative controller blocks, which is designed purposefully to exhibit the methods proposed in this paper have potentials to impact physical systems such as multi-agent structures. In fact, this example can be viewed as a line of vehicles moving together in the same direction where the separation between every two neighboring cars is maintained by a proportional-integral-derivative controller. In addition, every vehicle follows the speed of the leading vehicle (m_{last}) through a damper-like controller. All masses are assumed to be 1 kg. Moreover, the undamaged constants are $k_p = 10\,\text{N/m}$, $k_i = 0.5\,\text{N/ms}$, $k_d = 2\,\text{N}\cdot\text{s/m}$, and $b = 1\cdot\text{Ns/m}$. The dynamics of interest is the ratio of the entire line's length X to the disturbance at the m_1, F, in the frequency domain.

It is worth noting that the dynamics of interest are all at the ends in the aforementioned three examples. In fact, frequency response and transfer functions between any two nodes inside a dynamic network obeying the assumptions (A-1) to (A-6) can be obtained by the modeling algorithms illustrated in this paper. Furthermore, the methods proposed in this paper do not depend on the value of those assumed undamaged constants.

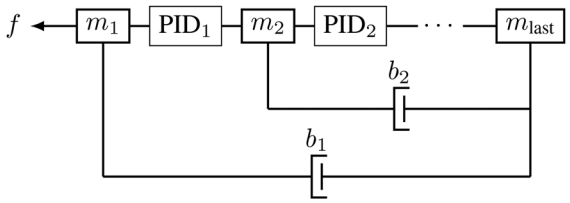

Fig. 3 ***Mechanical ladder*** **network**

3 Recurrence Formula

In this paper, a *recurrence formula* for a self-similar network is defined as a frequency-domain equation relating its overall dynamics to its subnetworks' dynamics. As we shall see later, recurrence formulas form the core of all four modeling algorithms proposed in this paper. The construction of recurrence formulas is exemplified through the aforementioned three examples. The key idea is to derive a network's transfer function assuming that all of its subnetworks' transfer functions are known.

For the mechanical tree network in Fig. 1, the transfer functions of two subnetworks are assumed to be available. That is

$$G_{s1}(s) = \frac{X_{2,1}(s)}{F_1(s)}$$
$$G_{s2}(s) = \frac{X_{2,2}(s)}{F_2(s)}$$

where $F_1(s) + F_2(s) = F(s)$. The corresponding illustration is shown in Fig. 4. When the first generation is taken into account, the forces $F_1(s)$ and $F_2(s)$ have their specific representations, which are

$$F_1(s) = k_{1,1}(X_{1,1}(s) - X_{2,1}(s))$$
$$F_2(s) = b_{1,1}s(X_{1,1}(s) - X_{2,2}(s))$$

The above two equations lead to that

$$\frac{X_{1,1}(s)}{F_1(s)} = \frac{1}{k_{1,1}} + G_{s1}(s)$$
$$\frac{X_{1,1}(s)}{F_2(s)} = \frac{1}{b_{1,1}s} + G_{s2}(s)$$

Then

$$\frac{X_{1,1}(s)}{F(s)} = \frac{X_{1,1}(s)}{F_1(s)+F_2(s)} = \frac{1}{\dfrac{1}{\dfrac{1}{k_{1,1}}+G_{s1}(s)} + \dfrac{1}{\dfrac{1}{b_{1,1}s}+G_{s2}(s)}}$$

Note that $X_{1,1}(s)/F(s)$ represents the dynamics of the entire mechanical tree network. As a result, the recurrence formula for the mechanical tree network is

$$G_r(s) = \frac{1}{\dfrac{1}{\dfrac{1}{k_{1,1}}+G_{s1}(s)} + \dfrac{1}{\dfrac{1}{b_{1,1}s}+G_{s2}(s)}} = \frac{k_{1,1}b_{1,1}sG_{s1}G_{s2} + k_{1,1}G_{s1} + b_{1,1}sG_{s2} + 1}{k_{1,1}b_{1,1}s(G_{s1}+G_{s2}) + k_{1,1} + b_{1,1}s} \quad (1)$$

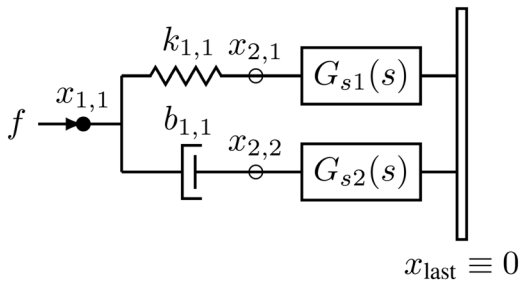

Fig. 4 A simplified version of the mechanical tree network in Fig. 1 used to derive its recurrence formula

Downloaded from http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/144/8/081007/6888118/ds_144_08_081007.pdf by University Of Notre Dame user on 13 July 2023

Note that Eq. (1) actually directly follows the series and parallel connecting rules of idealized mechanical components since no masses are involved.

For the electrical ladder network in Fig. 2, the input impedance of its subnetwork is assumed to be known, which is indicated by

$$G_{s1}(s) = \frac{V_1(s)}{I_1(s)}$$

The illustration of that is shown in Fig. 5. The derivation of its recurrence formula is similar to that of the mechanical tree network's. Therefore, that is omitted here. The result is directly led by the series and parallel connection rules of idealized electrical components (or, by the concept of equivalent impedance), i.e.,

$$G_r(s) = \frac{V_{\text{in}}(s)}{I_{\text{in}}(s)} = r_{1,1} + \frac{1}{\frac{1}{r_{1,2}} + c_1 s + \frac{1}{G_{s1}(s)}}$$

$$= \frac{(r_{1,1}r_{1,2}c_1 s + r_{1,1} + r_{1,2})G_{s1}(s) + r_{1,1}r_{1,2}}{(r_{1,2}c_1 s + 1)G_{s1}(s) + r_{1,2}} \tag{2}$$

For the mechanical ladder network in Fig. 3, the dynamics of the subnetwork is again assumed to be known, where

$$G_{s1}(s) = \frac{X_s(s)}{F_s(s)}$$

as shown in Fig. 6. Similar to the mechanical tree network, when the first generation is taken into account, the force $F_s(s)$ is determined by the controller block PID_1. Hence

$$G_{s1}(s) = \frac{X_s(s)}{\underbrace{\left(k_{p1} + \frac{k_{i1}}{s} + k_{d1}s\right)}_{K_1(s)} X_1(s)} = \frac{X_s(s)}{K_1(s)X_1(s)}$$

Next, assume that m_{last} is always moving at a constant speed. That can be achieved when the disturbance $f(t)$ has negligible impacts on m_{last}, e.g., $f(t)$ is small, or the mechanical ladder network possesses many generations, or m_{last} has an external controller to maintain its constant speed. From Newton's second law of motion, at m_1

$$m_1 s^2(X_1(s) + X_s(s)) = F(s) - K_1(s)X_1(s) - b_1 s(X_1(s) + X_s(s))$$

which leads to

$$(m_1 s^2 + b_1 s + K_1(s))X_1(s) + (m_1 s^2 + b_1 s)X_s(s) = F(s)$$

Because $X_s(s) = G_{s1}(s)K_1(s)X_1(s)$, we then have the following two equations:

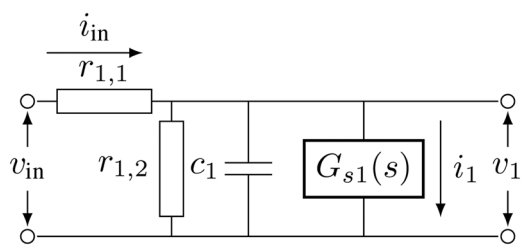

Fig. 5 A simplified version of the electrical ladder network in Fig. 2 used to derive its recurrence formula

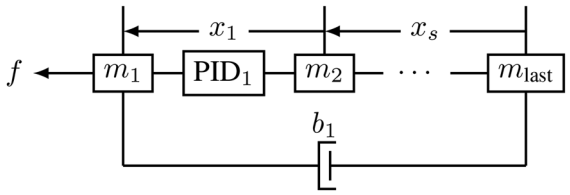

Fig. 6 A simplified version of the mechanical ladder network in Fig. 3 used to derive its recurrence formula

$$\Big((m_1 s^2 + b_1 s)(G_{s1}K_1 + 1) + K_1\Big)X_1(s) = F(s)$$

$$\frac{(m_1 s^2 + b_1 s)(G_{s1}K_1 + 1) + K_1}{G_{s1}K_1} X_s(s) = F(s)$$

Finally, the recurrence formula of the mechanical ladder network is given by

$$G_r(s) = \frac{X(s)}{F(s)} = \frac{X_1(s) + X_s(s)}{F(s)} = \frac{G_{s1}(s)K_1(s) + 1}{(m_1 s^2 + b_1 s)(G_{s1}(s)K_1(s) + 1) + K_1(s)} \tag{3}$$

4 Finite Networks

In this section, the two algorithms of computing frequency response and transfer functions for finite self-similar networked dynamical systems are presented. Then, in Sec. 4.3, the correctness of the results is verified.

The main idea is simply using a network's recurrence formula from the first generation repeatedly until the last one. However, the tricky part is correctly substituting the corresponding components' constants into their respective iterations. Therefore, the algorithms in this paper leverage a recursive process to assure that aspect of correctness.

4.1 Frequency Response. The algorithm to compute the frequency response of a self-similar finite dynamic network is listed in Algorithm 1.

Algorithm 1 Pseudo-code of computing finite networks' frequency response. It computes the frequency response `G` at the angular frequency `w` for a finite network with `nG` number of generations given its damage case (`l,e`) and the undamaged constants `undCst`.

```
1: function G=freqFin(l,e,undCst,w,nG)
2: s=1j*w;
3: [l1,e1,lS,eS]=partition(l,e);
4: g1Cst=getG1Cst(l1,e1,undCst);
5: if nG == 1 then
6:   G=G1(g1Cst,s);
7: else
8:   nG=nG-1;
9:   for idx from 1 to nS do
10:    GS[idx]=freqFin(lS[idx],eS[idx],
11:    undCst,w,nG);
12:  end for
13:  G=Gr(g1Cst,GS,s);
14: end if
```

The algorithm starts with the `partition()` function, which splits the damage case of the entire network (`l,e`) into two groups. The first group is the damage case at the first generation (`l1,e1`). The other group contains all damage cases for subnetworks, where the damage case (`lS[idx],eS[idx]`) is with respect to the `idx`-th subnetwork. As a concrete example, for the mechanical tree network in Fig. 1, suppose its damage case is

Downloaded from http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/144/8/081007/6888118/ds_144_08_081007.pdf by University Of Notre Dame user on 13 July 2023

$$(\texttt{l},\texttt{e}) = ([k_{1,1}, b_{2,1}, k_{3,2}, b_{3,3}], [0.1, 0.2, 0.3, 0.4])$$

Then, the `partition()` function returns the following results:

$$(\texttt{l1},\texttt{e1}) = ([k_{1,1}], [0.1])$$
$$(\texttt{lS[1]},\texttt{eS[1]}) = ([b_{1,1}, k_{2,2}], [0.2, 0.3])$$
$$(\texttt{lS[2]},\texttt{eS[2]}) = ([b_{2,1}], [0.4])$$

Note that the indices of the components in `lS` are with respect to their corresponding subnetworks. That is the reason why $b_{2,1}$ in `l` is converted to $b_{1,1}$ in `lS` [1]. Then, the damage case at the first generation (`l1,e1`) is used to compute the values of constants at the first generation `g1Cst` by the `getG1Cst()` function. For the above example, `g1Cst` includes that

$$k_{1,1} = 0.1\,\text{k} = 0.2\text{N/m} \quad \text{and} \quad b_{1,1} = b = 1\,\text{N}\cdot\text{s/m}$$

Then, the algorithm breaks into two parts determined by the criterion if the network has only one generation, i.e., `nG` $= 1$. If so, the result is directly returned by the `G1()` function given the constants at the first generation. It is worth noting that case is also the base case of the entire recursive procedure. The computations in the `G1()` function can be easily derived. For the mechanical tree network, it is

$$G_1(s) = \frac{1}{k_{1,1} + b_{1,1}s} \tag{4}$$

For the electrical ladder network, it is

$$G_1(s) = r_{1,1} + \frac{1}{\frac{1}{r_{1,2}} + c_1 s} = \frac{r_{1,1}r_{1,2}c_1 s + r_{1,1} + r_{1,2}}{r_{1,2}c_1 s + 1} \tag{5}$$

For the mechanical ladder network, it is

$$G_1(s) = \frac{1}{m_1 s^2 + b_1 s + K_1(s)} \tag{6}$$

If the network has more than one generation, i.e., `nG` > 1, the algorithm recursively calls itself for each of its subnetworks to compute their frequency responses. Those frequency responses are then used to compute the final result, that is the frequency response of the entire network, through those recurrence formulas implemented in the `Gr()` function.

Note that Algorithm 1 is less likely to cause mistakes as opposed to manually inserting the values of components' constants within a large network at every iteration of computation. As long as Algorithm 1 is coded correctly, that type of error would not occur. Another advantage of Algorithm 1 is its adaptability. It can deal with all finite dynamic networks satisfying the assumptions from (A-1) to (A-4) in Sec. 2.

The frequency response of the three examples under some specific situations acquired by Algorithm 1 is shown in Figs. 7–9. Note that Algorithm 1 is also able to compute frequency response of an undamaged finite network, in which case the input arguments `l` and `e` should be two empty lists.

4.2 Transfer Functions. This section proposes an algorithm for computing transfer functions of finite dynamic networks that satisfy the assumptions (A-1) to (A-4) in Sec. 2. The recursive structure of the algorithm in this section is same as that of Algorithm 1. The difference is that the algorithm in this section does not depend on angular frequencies w, and it purely operates on the coefficients of transfer functions. That is guaranteed by the equivalence between polynomial multiplications and tensor convolutions. Therefore, in what follows, that equivalence is reviewed first.

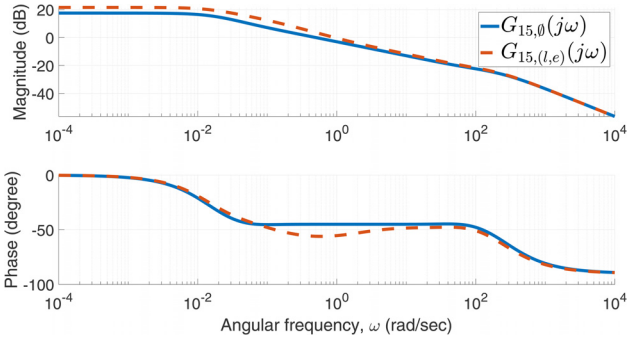

Fig. 7 Frequency response for two 15-generation mechanical tree networks. The blue curve is for the undamaged case, $G_{15,\emptyset}(j\omega)$. The red dashed curve is for a damage case, $G_{15,(l,e)}(j\omega)$, where $(l,e) = ([k_{2,1}, k_{2,2}, b_{3,1}], [0.1, 0.2, 0.3])$.

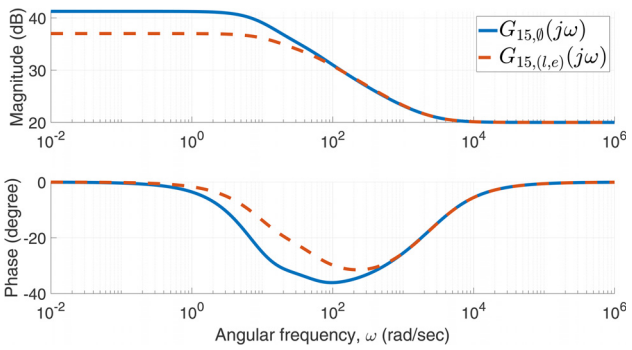

Fig. 8 Input impedance for two 15-generation electrical ladder networks. The solid curve is for the undamaged case, $G_{15,\emptyset}(j\omega)$. The dashed curve is for a damage case, $G_{15,(l,e)}(j\omega)$, where $(l, e) = ([r_{2,2}], [0.1])$.

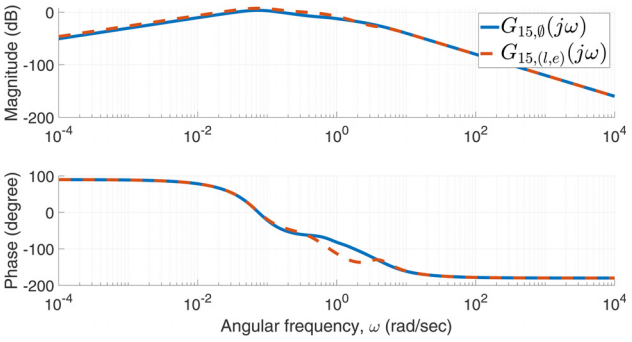

Fig. 9 Frequency response for two 15-generation mechanical ladder networks. The blue curve is for the undamaged case, $G_{15,\emptyset}(j\omega)$. The red dashed curve is for a damage case, $G_{15,(l,e)}(j\omega)$, where $(l, e) = ([k_{p2}, k_{i2}, k_{d2}], [0.1, 0.1, 0.1])$.

4.2.1 Equivalence Between Polynomial Multiplication and Tensor Convolution. Here, only vector convolutions and matrix convolutions are reviewed, since they cover all three examples appearing in this paper. However, note that the equivalence is actually satisfied for all finite dimensional tensors.

- In the case of vector convolution, $C = A * B$ is defined as

$$C(k) = \sum_p A(p)B(k - p + 1)$$

where p runs over all values that lead to legal subscripts of $A(p)$ and $B(k - p + 1)$. For two univariate polynomials

Downloaded from http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/144/8/081007/6888118/ds_144_08_081007.pdf by University Of Notre Dame user on 13 July 2023

$$a(x) = a_1 x^{n_A - 1} + a_2 x^{n_A - 2} + \cdots + a_{n_A}$$
$$b(x) = b_1 x^{n_B - 1} + b_2 x^{n_B - 2} + \cdots + b_{n_B}$$

their coefficient vectors are defined as

$$A = [\, a_1 \quad a_2 \quad \cdots \quad a_{n_A} \,]$$
$$B = [\, b_1 \quad b_2 \quad \cdots \quad b_{n_B} \,]$$

It can be shown that the vector convolution of A and B, $C = A * B$, is the coefficient vector of the corresponding univariate polynomial multiplication

$$c(x) = a(x)b(x)$$

- In the case of matrix convolution, $C = A * B$, is defined as

$$C(j,k) = \sum_p \sum_q A(p,q) B(j - p + 1, k - q + 1)$$

where p and q run over all values that lead to legal subscripts of $A(p,q)$ and $B(j - p + 1, k - q + 1)$. For two bivariate polynomials

$$\begin{aligned} a(x,y) = {} & a_{1,1} x^{n_A - 1} y^0 + \cdots + a_{1,n_A} x^0 y^{n_A - 1} \\ & + a_{2,2} x^{n_A - 2} y^0 + \cdots + a_{2,n_A} x^0 y^{n_A - 2} \\ & + \cdots + a_{n_A,n_A} x^0 y^0 \\ b(x,y) = {} & b_{1,1} x^{n_B - 1} y^0 + \cdots + b_{1,n_B} x^0 y^{n_B - 1} \\ & + b_{2,2} x^{n_B - 2} y^0 + \cdots + b_{2,n_B} x^0 y^{n_B - 2} \\ & + \cdots + b_{n_B,n_B} x^0 y^0 \end{aligned}$$

their coefficient matrices are defined as

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n_A - 1} & a_{1,n_A} \\ 0 & a_{2,2} & \cdots & a_{2,n_A - 1} & a_{2,n_A} \\ 0 & 0 & \cdots & a_{3,n_A - 1} & a_{3,n_A} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & a_{n_A,n_A} \end{bmatrix}$$

$$B = \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,n_B - 1} & b_{1,n_B} \\ 0 & b_{2,2} & \cdots & b_{2,n_B - 1} & b_{2,n_B} \\ 0 & 0 & \cdots & b_{3,n_B - 1} & b_{3,n_B} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & b_{n_B,n_B} \end{bmatrix}$$

It can be shown that the matrix convolution of A and B, $C = A * B$, is the coefficient matrix of the corresponding bivariate polynomial multiplication

$$c(x,y) = a(x,y)b(x,y)$$

Additionally, a related new operator for additions between two coefficient vectors and matrices is defined here. The operator $\oplus$ can be applied between two coefficient vectors or matrices with different dimensions. The result of that should be consistent with the addition between two univariate or bivariate polynomials. Two examples are provided as follows.

- In the case of between two coefficient vectors

$$[\, a_1 \quad a_2 \quad a_3 \,] \oplus [\, b_1 \quad b_2 \,] = [\, a_1 \quad a_2 + b_1 \quad a_3 + b_2 \,] \tag{7}$$

- In the case of between two coefficient matrices

$$\begin{aligned} & \begin{bmatrix} a_{1,1} & a_{1,2} \\ 0 & a_{2,2} \end{bmatrix} \oplus \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ 0 & b_{2,2} & b_{2,3} \\ 0 & 0 & b_{3,3} \end{bmatrix} \\ = {} & \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ 0 & a_{1,1} + b_{2,2} & a_{1,2} + b_{2,3} \\ 0 & 0 & a_{2,2} + b_{3,3} \end{bmatrix} \end{aligned} \tag{8}$$

4.2.2 Algorithm for Transfer Functions. The computation of finite networks' transfer functions is based on the observation as follows. Recall that the computation of a finite network's frequency response repeatedly uses its recurrence formula $G_r(s)$ starting with the transfer function when it only has one generation, $G_1(s)$. Moreover, note that both $G_r(s)$ and $G_1(s)$ are rational expressions where the numerator and denominator of both are polynomials in the Laplace variable s. Therefore, the result of combining them together, that is a finite network's transfer function, is also a rational expression of s. Hence, the coefficient vectors of both its numerator and denominator can be obtained directly, which leads to the analytical expression of a finite network's transfer function.

For the mechanical tree example, its one-generation transfer function $G_1(s)$ in Eq. (4) can be converted to two coefficient vectors for the numerator and denominator, $\boldsymbol{c}_{N1}$ and $\boldsymbol{c}_{D1}$, where

$$\boldsymbol{c}_{N1} = [\, 1 \,]$$
$$\boldsymbol{c}_{D1} = [\, b_{1,1} \quad k_{1,1} \,]$$

Define the transfer functions of two subnetworks as

$$G_{s1}(s) = \frac{N_{s1}(s)}{D_{s1}(s)}, \quad \text{and} \quad G_{s2}(s) = \frac{N_{s2}(s)}{D_{s2}(s)}$$

Substituting them into the recurrence formula in Eq. (1) leads to

$$G_r(s) = \frac{k_{1,1} b_{1,1} s N_{s1} N_{s2} + k_{1,1} N_{s1} D_{s2} + b_{1,1} s N_{s2} D_{s1} + D_{s1} D_{s2}}{k_{1,1} b_{1,1} s (N_{s1} D_{s2} + N_{s2} D_{s1}) + (k_{1,1} + b_{1,1} s) D_{s1} D_{s2}}$$

Therefore, if the coefficient vectors for the two subnetworks are defined as $\boldsymbol{c}_{Ns1}$, $\boldsymbol{c}_{Ds1}$, $\boldsymbol{c}_{Ns2}$, and $\boldsymbol{c}_{Ds2}$, then, its recurrence formula $G_r(s)$ in Eq. (1) can be converted to the following two equations which only operate on the coefficients:

$$\begin{aligned} \boldsymbol{c}_{Nr} = {} & [\, k_{1,1} b_{1,1} \quad 0 \,] * \boldsymbol{c}_{Ns1} * \boldsymbol{c}_{Ns2} \oplus [\, k_{1,1} \,] * \boldsymbol{c}_{Ns1} * \boldsymbol{c}_{Ds2} \\ & \oplus [\, b_{1,1} \quad 0 \,] * \boldsymbol{c}_{Ns2} * \boldsymbol{c}_{Ds1} \oplus \boldsymbol{c}_{Ds1} * \boldsymbol{c}_{Ds2} \end{aligned} \tag{9}$$

$$\begin{aligned} \boldsymbol{c}_{Dr} = {} & [\, k_{1,1} b_{1,1} \quad 0 \,] * (\boldsymbol{c}_{Ns1} * \boldsymbol{c}_{Ds2} \oplus \boldsymbol{c}_{Ns2} * \boldsymbol{c}_{Ds1}) \\ & \oplus [\, b_{1,1} \quad k_{1,1} \,] * \boldsymbol{c}_{Ds1} * \boldsymbol{c}_{Ds2} \end{aligned} \tag{10}$$

where the operator $*$ means vector convolutions, and $\oplus$ is defined as the addition between two coefficient vectors according to Eq. (7).

Similar procedures can be applied to the other two examples. For the electrical ladder network, its one-generation transfer function $G_1(s)$ in Eq. (5) can be converted to

$$\boldsymbol{c}_{N1} = [\, r_{1,1} r_{1,2} c_1 \quad r_{1,1} + r_{1,2} \,]$$
$$\boldsymbol{c}_{D1} = [\, r_{1,2} c_1 \quad 1 \,]$$

and its recurrence formula $G_r(s)$ in Eq. (2) can be converted to

$$\boldsymbol{c}_{Nr} = [\, r_{1,1} r_{1,2} c_1 \quad r_{1,1} + r_{1,2} \,] * \boldsymbol{c}_{Ns1} \oplus r_{1,1} r_{1,2} \boldsymbol{c}_{Ds1} \tag{11}$$

Downloaded from http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/144/8/081007/6888118/ds_144_08_081007.pdf by University Of Notre Dame user on 13 July 2023

$$\boldsymbol{c_{Dr}} = [\, r_{1,2}c_1 \quad 1\,] * \boldsymbol{c_{Ns1}} \oplus r_{1,2}\boldsymbol{c_{Ds1}} \tag{12}$$

For the mechanical ladder network, its one-generation transfer function $G_1(s)$ in Eq. (6) can be converted to

$$\boldsymbol{c_{N1}} = [\, 1 \quad 0\,]$$
$$\boldsymbol{c_{D1}} = [\, m_1 \quad b_1 + k_{d1} \quad k_{p1} \quad k_{i1}\,]$$

and its recurrence formula $G_r(s)$ in Eq. (3) can be converted to

$$\boldsymbol{c_{Nr}} = [\, k_{d1} \quad k_{p1} \quad k_{i1}\,] * \boldsymbol{c_{Ns1}} \oplus [\, 1 \quad 0\,] * \boldsymbol{c_{Ds1}}$$
$$\boldsymbol{c_{Dr}} = [\, m_1 \quad b_1 \quad 0\,] * [\, k_{d1} \quad k_{p1} \quad k_{i1}\,] * \boldsymbol{c_{Ns1}} \oplus [\, m_1 \quad b_1 + k_{d1} \quad k_{p1} \quad k_{i1}\,] * \boldsymbol{c_{Ds1}}$$

Finally, the pseudo-code of computing finite networks' transfer functions is listed in Algorithm 2.

Algorithm 2 Pseudo-code of computing finite networks' transfer function. It computes the coefficient vectors `cN` and `cD` of an `nG`-generation network's transfer function given its damage case `(l,e)` and the undamaged constants `undCst`.

```
function [cN,cD]=tranFin(l,e,undCst,nG)
[l1,e1,lS,eS]=partition(l,e);
g1Cst=getG1Cst(l1,e1,undCst);
if nG == 1 then
   [cN,cD]=C1(g1Cst);
else
   nG=nG-1;
   for idx from 1 to nS do
    [cNS[idx],cDS[idx]]
    =tranFin(lS[idx],eS[idx],undCst,nG);
   end for
   [cN,cD]=Cr(g1Cst,cNS,cDS);
   [cN,cD]=simplify(cN,cD);
end if
```

The structure is same as its counterpart for frequency response in Algorithm 1. The main difference is that Algorithm 2 is independent of angular frequencies `w`, and it returns the coefficient vectors `cN` and `cD` of a transfer function $G(s)$. The `G1()` function in Algorithm 1 is replaced by the `C1()` function in Algorithm 2 which returns $\boldsymbol{c_{N1}}$ and $\boldsymbol{c_{D1}}$. The `Gr()` function in Algorithm 1 is replaced by the `Cr()` function in Algorithm 2 whose implementation is given by the expressions for $\boldsymbol{c_{Nr}}$ and $\boldsymbol{c_{Dr}}$. In addition, there exists a new function called `simplify()` in Algorithm 2, which reduces the resultant coefficient vectors from the `Cr()` function. For example, the order of two coefficient vectors can be lowered by the same amount, like

$$\boldsymbol{c_N} = [\, 1 \quad 2 \quad 0\,] \quad \text{and} \quad \boldsymbol{c_D} = [\, 3 \quad 4 \quad 0 \quad 0\,]$$

being simplified to

$$\boldsymbol{c_N} = [\, 1 \quad 2\,] \quad \text{and} \quad \boldsymbol{c_D} = [\, 3 \quad 4 \quad 0\,]$$

Another possible simplification is that all elements in both $\boldsymbol{c_N}$ and $\boldsymbol{c_D}$ can be divided by the same number, such as

$$\boldsymbol{c_N} = [\, 2 \quad 4\,] \quad \text{and} \quad \boldsymbol{c_D} = [\, 6 \quad 8\,]$$

being simplified to

$$\boldsymbol{c_N} = [\, 1 \quad 2\,] \quad \text{and} \quad \boldsymbol{c_D} = [\, 3 \quad 4\,]$$

The results for the three examples in some specific situations are presented as follows: For a two-generation mechanical tree network whose damage case is $(\boldsymbol{l}, \boldsymbol{e}) = ([k_{2,1}, k_{2,2}], [0.1, 0.2])$, its transfer function is

$$G_{2,(\boldsymbol{l},\boldsymbol{e})}(s) = \frac{2s^2 + 4.8s + 0.88}{s^3 + 6.6s^2 + 2.48s + 0.16} \tag{13}$$

For a four-generation electrical ladder network whose damage case is

$$(\boldsymbol{l}, \boldsymbol{e}) = ([r_{2,2}, r_{3,2}], [0.1, 0.1])$$

the analytical expression of its input impedance is

$$G_{4,(\boldsymbol{l},\boldsymbol{e})}(s) = \frac{s^4 + 7220s^3 + 1.6 \times 10^7 s^2 + 1.2 \times 10^{10} s + 1.6 \times 10^{12}}{0.1s^4 + 622s^3 + 1.1 \times 10^6 s^2 + 5 \times 10^8 s + 2.4 \times 10^{10}} \tag{14}$$

For a two-generation mechanical ladder network whose damage case is

$$(\boldsymbol{l}, \boldsymbol{e}) = ([k_{p2}, k_{i2}, k_{d2}], [0.1, 0.1, 0.1])$$

its transfer function is

$$G_{2,(\boldsymbol{l},\boldsymbol{e})}(s) = \frac{s^4 + 3.2s^3 + 11s^2 + 0.55s}{s^6 + 6.2s^5 + 26.6s^4 + 26.05s^3 + 11.25s^2 + s + 0.025} \tag{15}$$

Same as Algorithm 1, it is worth emphasizing that Algorithm 2 is able to compute transfer function of an undamaged finite network, in which case the input arguments l and e should be two empty lists. Note that Algorithm 2 is always computationally efficient compared to Algorithm 1 no matter the network is damaged or not. However, Algorithm 2 is more likely numerically overflow because the magnitude of coefficients grows fast as the number of generation increases as can be seen in Eqs. (13)–(15).

4.3 Correctness Check. The correctness of the results obtained by the two algorithms in Secs. 4.1 and 4.2 is checked here. That confirmation is twofold. First, a frequency response obtained from Algorithm 1 should be consistent with its transfer function obtained from Algorithm 2. The comparison is cast between a frequency response $G(j\omega)$ from Algorithm 1 and a sampling of its transfer function $G(s)$ from Algorithm 2 at a sequence of frequencies. That consistency is displayed by Figs. 10–12. Note that the frequency ranges in Figs. 10–12 are all chosen so that the corresponding frequency response is within the transient region.

The second confirmation is whether the frequency response and transfer functions obtained by Algorithms 1 and 2 are consistent with their time-domain response. On the one hand, the time-domain response of a network is obtained by the `lsim()` function in `MATLAB` given its transfer function from Algorithm 2 and an input signal $u(t)$. On the other hand, that time-domain response

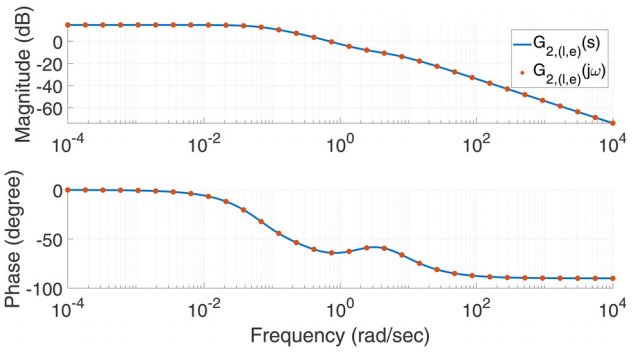

Fig. 10 The consistency between a finite mechanical tree network's transfer function in Eq. (13) and its corresponding frequency response from Algorithm 1

Downloaded from http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/144/8/081007/6888118/ds_144_08_081007.pdf by University Of Notre Dame user on 13 July 2023

can also be obtained by the numerical integration of the differential equations describing that network's dynamics given the same input signal $u(t)$.

For the mechanical tree network in Fig. 1 with two generations, the following equations of force balance can be formulated:

$$f = k_{1,1}(x_{1,1} - x_{2,1}) + b_{1,1}(\dot{x}_{1,1} - \dot{x}_{2,2})$$
$$k_{1,1}(x_{1,1} - x_{2,1}) = k_{2,1}x_{2,1} + b_{2,1}\dot{x}_{2,1}$$
$$b_{1,1}(\dot{x}_{1,1} - \dot{x}_{2,2}) = k_{2,2}x_{2,2} + b_{2,2}\dot{x}_{2,2}$$

which result in a system of equations

$$\begin{bmatrix} b_{1,1} & 0 & -b_{1,1} \\ 0 & b_{2,1} & 0 \\ b_{1,1} & 0 & -b_{1,1} - b_{2,2} \end{bmatrix} \underbrace{\begin{bmatrix} \dot{x}_{1,1} \\ \dot{x}_{2,1} \\ \dot{x}_{2,2} \end{bmatrix}}_{\dot{x}} = \begin{bmatrix} f - k_{1,1}(x_{1,1} - x_{2,1}) \\ k_{1,1}(x_{1,1} - x_{2,1}) - k_{2,1}x_{2,1} \\ k_{2,2}x_{2,2} \end{bmatrix}$$

Then, the time-domain response is obtained by using the `ode45()` function in MATLAB to integrate $\dot{x}$ solved from the above system of equations. The input signal is a logistic function $f(t) = 1/(1 + e^{-50(t-0.2)})$. The consistency between the above result and the one given by the `lsim()` with the transfer function in Eq. (13) for the same input $f(t)$ is shown in Fig. 13, which confirms the correctness from the perspective of time-domain response. Note that we have tested various error tolerances and different integration solvers such as `ode15s()` and `ode23()`, but no qualitative differences can be found.

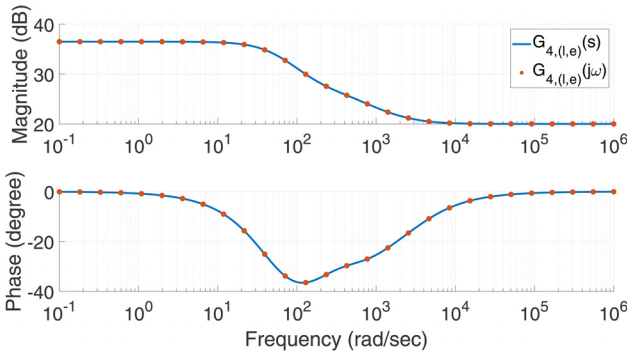

Fig. 11 The consistency between a finite electrical ladder network's transfer function in Eq. (14) and its corresponding frequency response from Algorithm 1

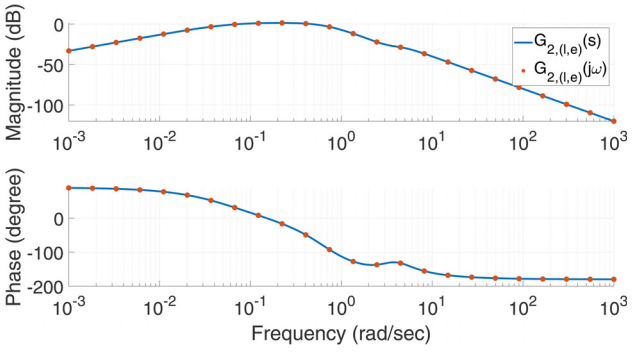

Fig. 12 The consistency between a finite mechanical ladder network's transfer function in Eq. (15) and its corresponding frequency response from Algorithm 1

For the electrical ladder network in Fig. 2 with four generations, the following system of differential equations results is derived from Kirchhoff's circuit laws:

$$\dot{v}_1 = \frac{1}{c_1}\left(i_{\text{in}} - \frac{v_1}{r_{1,2}} - \frac{v_1 - v_2}{r_{2,1}}\right)$$
$$\dot{v}_2 = \frac{1}{c_2}\left(\frac{v_1 - v_2}{r_{2,1}} - \frac{v_2}{r_{2,2}} - \frac{v_2 - v_3}{r_{3,1}}\right)$$
$$\dot{v}_3 = \frac{1}{c_3}\left(\frac{v_2 - v_3}{r_{3,1}} - \frac{v_3}{r_{3,2}} - \frac{v_3 - v_4}{r_{4,1}}\right)$$
$$\dot{v}_4 = \frac{1}{c_4}\left(\frac{v_3 - v_4}{r_{4,1}} - \frac{v_4}{r_{4,2}}\right)$$

The numerical integration of the above system of differential equations is performed by the `ode45()` function given the input signal $i_{\text{in}}(t) = 1/(1 + e^{-50(t-0.2)})$. Once $v_1(t)$ is obtained from the integration, $v_{\text{in}}(t)$ can also be computed from $v_{\text{in}}(t) = v_1(t) + r_{1,1}i_{\text{in}}(t)$. That is compared with the result given by the `lsim()` on the transfer function in Eq. (14), which is plotted in Fig. 14.

For the mechanical ladder network in Fig. 3 with two generations, the following equations of motion can be acquired by Newton's second law:

$$m_1\ddot{x}_1 = f - k_{p1}(x_1 - x_2) - k_{i1}\int_0^t (x_1 - x_2)d\tau - k_{d1}(\dot{x}_1 - \dot{x}_2) - b_1\dot{x}_1$$
$$m_2\ddot{x}_2 = k_{p1}(x_1 - x_2) + k_{i1}\int_0^t (x_1 - x_2)d\tau + k_{d1}(\dot{x}_1 - \dot{x}_2) - k_{p2}x_2 - k_{i2}\int_0^t x_2 d\tau - k_{d2}\dot{x}_2 - b_2\dot{x}_2$$

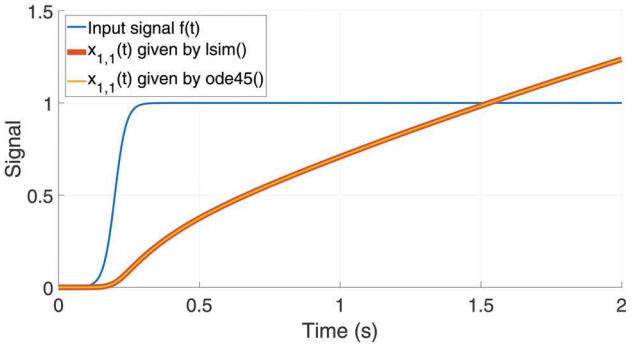

Fig. 13 Two ways of obtaining a finite mechanical tree network's time-domain response $x_{1,1}(t)$ give the same result, which confirms the correctness of the transfer function in Eq. (13)

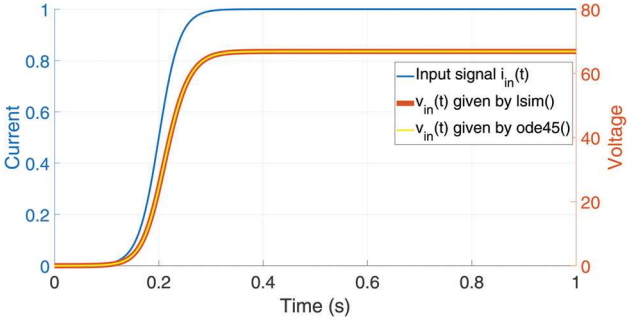

Fig. 14 Two ways of obtaining a finite electrical ladder network's time-domain response $v_{\text{in}}(t)$ give the same result, which confirms the correctness of the transfer function in Eq. (14)

Downloaded from http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/144/8/081007/6888118/ds_144_08_081007.pdf by University Of Notre Dame user on 13 July 2023

Note that x_1 and x_2 are the absolute displacements of m_1 and m_2 which are different from those in Fig. 6. The above two equations of motion can be organized into a matrix form, where

$$\underbrace{\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}}_{\boldsymbol{M}} \underbrace{\begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix}}_{\ddot{\boldsymbol{X}}} + \underbrace{\begin{bmatrix} k_{d1}+b_1 & -k_{d1} \\ -k_{d1} & k_{d1}+k_{d2}+b_2 \end{bmatrix}}_{\boldsymbol{B}} \underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}}_{\dot{\boldsymbol{X}}} + \underbrace{\begin{bmatrix} k_{p1} & -k_{p1} \\ -k_{p1} & k_{p1}+k_{p2} \end{bmatrix}}_{\boldsymbol{K}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{\boldsymbol{X}} = \underbrace{\begin{bmatrix} f - k_{i1}\int_0^t (x_1 - x_2)d\tau \\ k_{i1}\int_0^t (x_1 - x_2)d\tau - k_{i2}\int_0^t x_2 d\tau \end{bmatrix}}_{\boldsymbol{U}}$$

If we define that $\boldsymbol{Q} = [\boldsymbol{X}^\top \quad \dot{\boldsymbol{X}}^\top]^\top$, the corresponding system of first-order differential equations is given by

$$\dot{\boldsymbol{Q}} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{K} & -\boldsymbol{B} \end{bmatrix} \boldsymbol{Q} + \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{U} \end{bmatrix}$$

which can then be numerically integrated given the input signal $f(t) = 1/(1+e^{-50(t-0.2)})$. Because $\boldsymbol{U}$ contains the integrations of state variables, `ode45()` in not employed this time. Instead, a Riemann sum with constant time steps is performed. The resultant length of the entire two-generation mechanical ladder network, $x_1(t)$, is compared with the result given by the `lsim()` on the transfer function in Eq. (15), which is plotted in Fig. 15.

5 Undamaged Infinite Networks' Transfer Functions

For an infinite network that satisfies the assumptions (A-1) to (A-6) in Sec. 2, its transfer function when undamaged is a crucial part of finding other transfer functions in more general cases. In fact, that plays the same role as those one-generation transfer functions $G_1(s)$ do for finite networks in Algorithms 1 and 2. Therefore, how to compute them is reviewed in this section. Note that the method to evaluate them is from existing work in literature, e.g., Refs. [46–48], rather than this work. However, it is still repeated here for the purpose of completeness.

One observation is that when a self-similar infinite network is undamaged, its transfer function is same as its subnetworks', i.e., $G_{si}(s) = G_{\infty,\emptyset}(s)$ for all i. Specifically, for an infinite mechanical tree network, when undamaged, the following relation can be established from its recurrence formula in Eq. (1)

$$G_{\infty,\emptyset}(s) = \frac{kbsG^2_{\infty,\emptyset}(s) + (bs+k)G_{\infty,\emptyset}(s) + 1}{2kbsG_{\infty,\emptyset}(s) + k + bs}$$

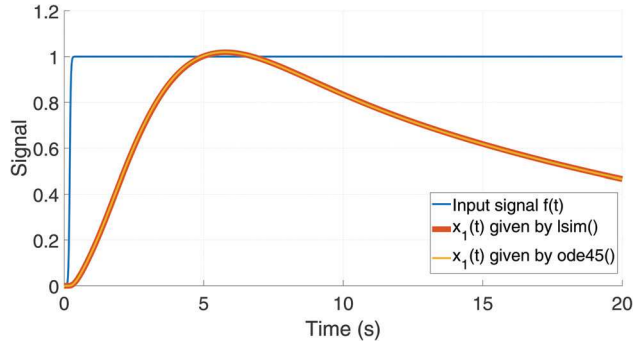

Fig. 15 Two ways of obtaining a finite mechanical ladder network's time-domain response $x_1(t)$ give the same result, which confirms the correctness of the transfer function in Eq. (15)

Note that $k_{1,1} = k$ and $b_{1,1} = b$ in the case of no damages. The above equation leads to

$$kbsG^2_{\infty,\emptyset}(s) = 1$$

which results in that the undamaged transfer function for an infinite mechanical tree network is

$$G_{\infty,\emptyset}(s) = \frac{1}{\sqrt{kbs}} \tag{16}$$

The other candidate result $-1/\sqrt{kbs}$ is not suitable because it is actually a nonminimum-phase system, which is physically impossible for a mechanical tree network in Fig. 1. That can be observed from the fact that, when undamaged, finite mechanical trees' transfer functions converge to $1/\sqrt{kbs}$ instead of the other one as the number of generations grows (see Fig. 16).

Similar procedures can be applied to the other two examples. For an infinite electrical ladder network, when undamaged, its recurrence formula leads to the following equation:

$$G_{\infty,\emptyset}(s) = \frac{(r_1r_2cs + r_1 + r_2)G_{\infty,\emptyset}(s) + r_1r_2}{(r_2cs+1)G_{\infty,\emptyset}(s) + r_2}$$

where $r_{1,1} = r_1$, $r_{1,2} = r_2$, and $c_1 = c$ when undamaged. That further leads to a quadratic equation in $G_{\infty,\emptyset}(s)$

$$(r_2cs+1)G^2_{\infty,\emptyset}(s) - (r_1r_2cs + r_1)G_{\infty,\emptyset}(s) - r_1r_2 = 0$$

Therefore, the undamaged transfer function of infinite electrical ladder network is

$$G_{\infty,\emptyset}(s) = \frac{s + \frac{1}{r_2c} + \sqrt{s^2 + \frac{2r_1+4r_2}{r_1r_2c}s + \frac{r_1+4r_2}{r_1r_2^2c^2}}}{\frac{2}{r_1}s + \frac{2}{r_1r_2c}} \tag{17}$$

The other candidate is also eliminated for the same reason, i.e., it is not the limiting point of convergence. That can be observed in Fig. 17.

For an infinite mechanical ladder network, when undamaged, its recurrence formula results in

$$G_{\infty,\emptyset}(s) = \frac{G_{\infty,\emptyset}(s)K(s) + 1}{(ms^2+bs)(G_{\infty,\emptyset}(s)K(s)+1) + K(s)}$$

where $K(s) = k_p + k_i/s + k_ds$. The equation leads to another quadratic equation in $G_{\infty,\emptyset}(s)$

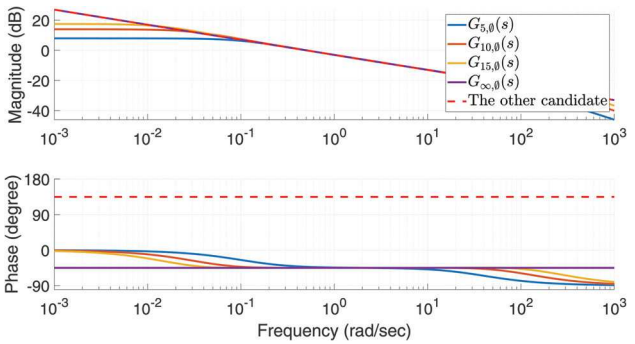

Fig. 16 Finite mechanical tree networks' undamaged transfer functions $G_{g,\emptyset}(s)$ converge to the infinite mechanical tree network's undamaged transfer function $G_{\infty,\emptyset}(s)$ in Eq. (16) as $g \to \infty$. The other candidate $-1/\sqrt{kbs}$ is not the limiting point of convergence, so it is not suitable.

Downloaded from http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/144/8/081007/6888118/ds_144_08_081007.pdf by University Of Notre Dame user on 13 July 2023

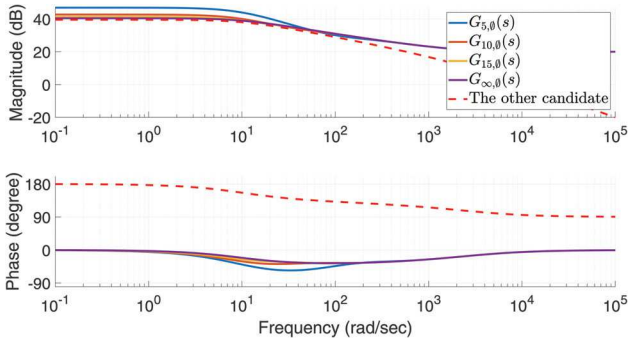

Fig. 17 Finite electrical ladder networks' undamaged transfer functions $G_{g,\emptyset}(s)$ converge to the infinite electrical ladder network's undamaged transfer function $G_{\infty,\emptyset}(s)$ in Eq. (17) as $g \to \infty$. The other candidate is not the limiting point of convergence, so it is not suitable.

$$[mk_d s^3 + (mk_p + bk_d)s^2 + (mk_i + bk_p)s + bk_i]G^2_{\infty,\emptyset}(s) + (ms^2 + bs)G_{\infty,\emptyset}(s) - 1 = 0$$

Therefore, the undamaged transfer function of infinite mechanical ladder network is

$$G_{\infty,\emptyset}(s) = \frac{-ms^2 - bs + A(s)}{2\left[mk_d s^3 + (mk_p + bk_d)s^2 + (mk_i + bk_p)s + bk_i\right]} \tag{18}$$

where

$$A(s) = [m^2 s^4 + (2mb + 4mk_d)s^3 + \left(b^2 + 4mk_p + 4bk_d\right)s^2 + 4(mk_i + bk_p)s + 4bk_i]^{\frac{1}{2}}$$

Again, the other candidate is eliminated for the same reason, and the convergence of finite mechanical ladder networks' undamaged transfer functions to the infinite one is shown in Fig. 18.

Section 6 shows how those undamaged transfer functions are employed to compute other transfer functions of infinite networks in more general cases.

6 General Infinite Networks

In this section, two algorithms to compute frequency response and transfer functions of infinite self-similar dynamic networks are presented, especially when they are damaged. The basic idea is similar to its counterpart for finite networks in Sec. 4. However, some indispensable changes are made to adapt for infinite networks.

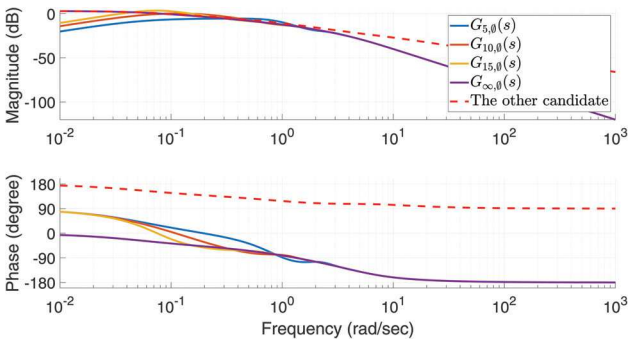

Fig. 18 Finite mechanical ladder networks' undamaged transfer functions $G_{g,\emptyset}(s)$ converge to the infinite mechanical ladder network's undamaged transfer function $G_{\infty,\emptyset}(s)$ in Eq. (18) as $g \to \infty$. The other candidate is not the limiting point of convergence, so it is not suitable.

6.1 Frequency Response. Recall that when evaluating frequency response for a finite network, the entire procedure starts with its one-generation transfer function $G_1(s)$ and employs its recurrence formula $G_r(s)$ in iterations until its total number of generations is achieved. That does not fit the case of infinite networks, as the number of iterations would become infinite. To overcome that difficulty, the assumption (A-5) in Sec. 2 is set which requires that an infinite network has a finite number of damaged components. For an infinite network fulfilling that assumption, it has to contain a deepest generation after which all subnetworks are undamaged. Then, that deepest generation is the starting point of the entire computation; thus, undamaged transfer functions for infinite networks $G_{\infty,\emptyset}(s)$ play the same role as one-generation transfer functions for finite networks $G_1(s)$ in the entire modeling procedure. By doing that, the number of iterations is limited to a finite number for infinite networks.

As a simple example, assume that an infinite mechanical tree network in Fig. 1 has a damage case $(\boldsymbol{l}, \boldsymbol{e}) = ([k_{1,1}], [0.1])$. Then, both subnetworks after the second generation are undamaged. Since the network is infinitely large, both of them have the same transfer functions as in Eq. (16). That would be the starting point of the computation. In fact, the frequency response of that infinite mechanical tree network can be evaluated by only using the recurrence formula in Eq. (1) once. That is

$$G_{\infty,(\boldsymbol{l},\boldsymbol{e})}(j\omega) = \frac{k_{1,1}b_{1,1}j\omega G^2_{\infty,\emptyset}(j\omega) + (k_{1,1} + b_{1,1}j\omega)G_{\infty,\emptyset}(j\omega) + 1}{2k_{1,1}b_{1,1}j\omega G_{\infty,\emptyset}(j\omega) + k_{1,1} + b_{1,1}j\omega}$$

where $k_{1,1} = 0.1k$, $b_{1,1} = b$, and $G_{\infty,\emptyset}(j\omega) = 1/\sqrt{kbj\omega}$.

In general, the pseudo-code for computing frequency response of infinite networks that satisfy the assumptions (A-1) to (A-6) in Sec. 2 is listed in Algorithm 3, which is same as Algorithm 1 except for the base case.

Algorithm 3 Pseudo-code of computing infinite networks' frequency response. It computes the frequency response `G` at the angular frequency `w` for an infinite network given its damage case `(l,e)` and the undamaged constants `undCst`.

```
function G=freqInf(l,e,undCst,w)
s=i*w;
if isEmpty(l) then
  G=GUnd(undCst,s);
else
  [l1,e1,lS,eS]=partition(l,e);
  for idx from 1 to nS do
    GS[idx]=freqInf(lS[idx],eS[idx],undCst,w);
  end for
  glCst=getGlCst(l1,e1,undCst);
  G=Gr(glCst,GS,s);
end if
```

For finite networks in Algorithm 1, the base case is determined by the criterion if the network has only one generation. Here, for infinite networks in Algorithm 3, the base case is determined by the criterion if the network is undamaged, which is characterized by an empty list of damaged components $\boldsymbol{l}$. The computation of the base case is also replaced by the `GUnd()` function, which implements undamaged transfer functions for infinite networks $G_{\infty,\emptyset}(s)$, such as those in Sec. 5. The resultant frequency response for the three examples networks under some specific damage cases are shown in Figs. 19–21, where the convergence of finite networks' frequency response to infinite networks' under the same damage cases is also plotted to verify the correctness of the results.

Downloaded from http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/144/8/081007/6888118/ds_144_08_081007.pdf by University Of Notre Dame user on 13 July 2023

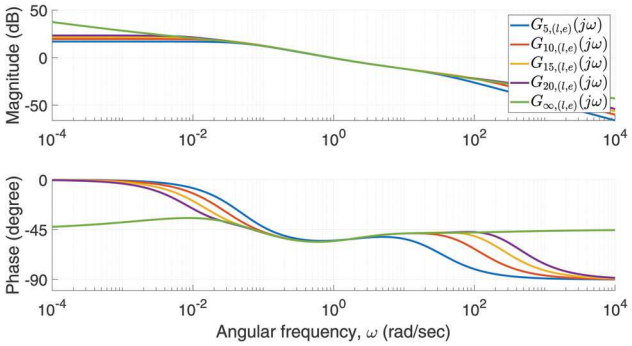

Fig. 19 Frequency response for mechanical tree networks in Fig. 1 when the damage case is $(l,e) = ([k_{2,1}, k_{2,2}, b_{3,1}], [0.1, 0.2, 0.3])$. Finite networks' frequency response is computed by Algorithm 1, whereas the infinite network's is computed by Algorithm 3.

6.2 Transfer Functions. Recall that to compute finite networks' transfer functions, the coefficient vectors of one-generation transfer functions $G_1(s)$ are extracted, and recurrence formulas $G_r(s)$ are also revised accordingly. Since the role of $G_1(s)$ is replaced by undamaged transfer functions of infinite networks $G_{\infty,\emptyset}(s)$ here, it seems straightforward to extract coefficient vectors of $G_{\infty,\emptyset}(s)$ as well. However, some adaptations have to be made because $G_{\infty,\emptyset}(s)$'s are no longer rational expressions as shown in Eqs. (16)–(18). Instead of extracting coefficients of s directly, in the case of infinite networks, those of some expressions in s, denoted by $\phi_i(s)$, are pulled out.

Specifically, for infinite mechanical tree network, the numerator and denominator of its undamaged transfer function

$$G_{\infty,\emptyset}(s) = \frac{1}{\sqrt{kbs}}$$

can be viewed as two polynomials with respect to the variable $\phi_1(s) = \sqrt{s}$. Then, the coefficient vectors of its undamaged transfer functions are

$$\boldsymbol{c}_{N\infty,\emptyset} = [\,1\,]$$
$$\boldsymbol{c}_{D\infty,\emptyset} = [\,\sqrt{kb} \quad 0\,]$$

Moreover, because its recurrence formula $G_r(s)$ is still a rational expression, the numerator and denominator of damaged transfer functions of infinite mechanical tree networks, obtained by repeatedly applying $G_r(s)$ starting with $G_{\infty,\emptyset}(s)$, are also polynomials with respect to $\phi_1(s) = \sqrt{s}$.

The reason for that is the operations in a rational expression are all basic mathematical operations. None of them could dissolve that $\phi_1(s) = \sqrt{s}$ into a univariate polynomial of s unless all exponents are even. Hence, $\phi_1(s) = \sqrt{s}$ must exist in the numerator and denominator of damaged transfer functions of infinite mechanical tree networks. Luckily, in this case, the variable s is the square of $\phi_1(s) = \sqrt{s}$, so the numerator and denominator can be viewed as univariate polynomials with respect to $\sqrt{s}$. General speaking, as demonstrated in the other two examples later, that relation is not obeyed. As a result, those transfer functions' numerator and denominator have to be regarded as multivariate polynomials with respect to both s and $\phi_i(s)$.

Therefore, in this case, the mechanical tree network's recurrence formula $G_r(s)$ in Eq. (1) can be revised as

$$\begin{aligned}\boldsymbol{c}_{Nr} = {} & [\,k_{1,1}b_{1,1} \quad 0 \quad 0\,] * \boldsymbol{c}_{Ns1} * \boldsymbol{c}_{Ns2} \\ & \oplus [\,k_{1,1}\,] * \boldsymbol{c}_{Ns1} * \boldsymbol{c}_{Ds2} \\ & \oplus [\,b_{1,1} \quad 0 \quad 0\,] * \boldsymbol{c}_{Ns2} * \boldsymbol{c}_{Ds1} \oplus \boldsymbol{c}_{Ds1} * \boldsymbol{c}_{Ds2}\end{aligned} \tag{19}$$

$$\begin{aligned}\boldsymbol{c}_{Dr} = {} & [\,k_{1,1}b_{1,1} \quad 0 \quad 0\,] * (\boldsymbol{c}_{Ns1} * \boldsymbol{c}_{Ds2} \oplus \boldsymbol{c}_{Ns2} * \boldsymbol{c}_{Ds1}) \\ & \oplus [\,b_{1,1} \quad 0 \quad k_{1,1}\,] * \boldsymbol{c}_{Ds1} * \boldsymbol{c}_{Ds2}\end{aligned} \tag{20}$$

which only operate on coefficients and are independent of frequency ω. In Eqs. (19) and (20), $\boldsymbol{c}_{Ns1}$, $\boldsymbol{c}_{Ds1}$, $\boldsymbol{c}_{Ns2}$, and $\boldsymbol{c}_{Ds2}$ are the coefficient vectors of both subnetworks with respect to the variable $\phi_1(s) = \sqrt{s}$. In addition, $*$ is the vector convolution operator, and $\oplus$ is a new vector addition operator defined in Sec. 4.2.1. Be careful that Eqs. (19) and (20) are different from their counterparts for finite mechanical tree networks in Eqs. (9) and (10) because the variable of polynomials for finite mechanical tree networks is s, while that for infinite mechanical tree networks is $\phi_1(s) = \sqrt{s}$.

For infinite electrical ladder network, its undamaged transfer function is

$$G_{\infty,\emptyset}(s) = \frac{s + \frac{1}{r_2 c} + \sqrt{s^2 + \frac{2r_1 + 4r_2}{r_1 r_2 c}s + \frac{r_1 + 4r_2}{r_1 r_2^2 c^2}}}{\frac{2}{r_1}s + \frac{2}{r_1 r_2 c}}$$

which is more complicated compared to the mechanical tree network because it contains an irrational expression of s. In this case, the numerator and denominator are regarded as two bivariate polynomials whose two variables are

$$\phi_1(s) = s \text{ and } \phi_2(s) = \sqrt{s^2 + \frac{2r_1 + 4r_2}{r_1 r_2 c}s + \frac{r_1 + 4r_2}{r_1 r_2^2 c^2}}$$

According to the definition of coefficient matrices for bivariate polynomials in Sec. 4.2.1, the coefficient matrices for infinite electrical ladder network's undamaged transfer function are

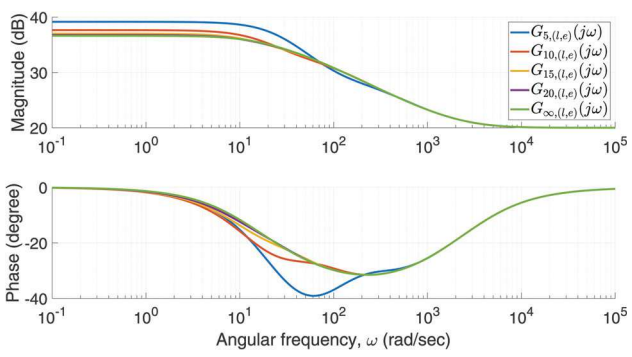

Fig. 20 Frequency response for electrical ladder networks in Fig. 2 when the damage case is $(l,e) = ([r_{2,2}], [0.1])$. Finite networks' frequency response is computed by Algorithm 1, whereas the infinite network's is computed by Algorithm 3.

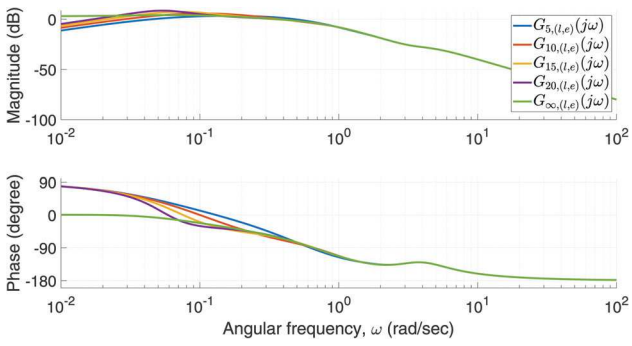

Fig. 21 Frequency response for mechanical ladder networks in Fig. 3 when the damage case is $(l,e) = ([k_{p2}, k_{i2}, k_{d2}], [0.1, 0.1, 0.1])$. Finite networks' frequency response is computed by Algorithm 1, whereas the infinite network's is computed by Algorithm 3.

Downloaded from http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/144/8/081007/6888118/ds_144_08_081007.pdf by University Of Notre Dame user on 13 July 2023

$$\boldsymbol{c_{N\infty,\emptyset}} = \begin{bmatrix} 1 & 1 \\ 0 & \dfrac{1}{r_2 c} \end{bmatrix}$$

$$\boldsymbol{c_{D\infty,\emptyset}} = \begin{bmatrix} \dfrac{2}{r_1} & 0 \\ 0 & \dfrac{2}{r_1 r_2 c} \end{bmatrix}$$

Correspondingly, its recurrence formula $G_r(s)$ in Eq. (2) should be revised to

$$\boldsymbol{c_{Nr}} = \begin{bmatrix} r_{1,1}r_{1,2}c_1 & 0 \\ 0 & r_{1,1}+r_{1,2} \end{bmatrix} * \boldsymbol{c_{Ns1}} \oplus r_{1,1}r_{1,2}\boldsymbol{c_{Ds1}} \tag{21}$$

$$\boldsymbol{c_{Dr}} = \begin{bmatrix} r_{1,2}c_1 & 0 \\ 0 & 1 \end{bmatrix} * \boldsymbol{c_{Ns1}} \oplus r_{1,2}\boldsymbol{c_{Ds1}} \tag{22}$$

where $*$ is the operator of matrix convolution and $\oplus$ is a new operator of matrix addition defined in Sec. 4.2.1. Again, note that Eqs. (21) and (22) are different from their counterparts Eqs. (11) and (12) for finite electrical ladder networks because the variables of polynomials are different.

Same situation happens for infinite mechanical ladder networks whose undamaged transfer function is

$$G_{\infty,\emptyset}(s) = \frac{-ms^2 - bs + A(s)}{2\left[mk_d s^3 + (mk_p + bk_d)s^2 + (mk_i + bk_p)s + bk_i\right]}$$

where $A(s)$ is defined in Eq. (18). In this case, the numerator and denominator of $G_{\infty,\emptyset}(s)$ can also be regarded as two bivariate polynomials whose two variables are

$$\phi_1(s) = s \text{ and } \phi_2(s) = A(s)$$

Then, the coefficient matrices of $G_{\infty,\emptyset}(s)$ are

$$\boldsymbol{c_{N\infty,\emptyset}} = \begin{bmatrix} -m & 0 & 0 \\ 0 & -b & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{c_{D\infty,\emptyset}} = \begin{bmatrix} 2mk_d & 0 & 0 & 0 \\ 0 & 2(mk_p + bk_d) & 0 & 0 \\ 0 & 0 & 2(mk_i + bk_p) & 0 \\ 0 & 0 & 0 & 2bk_i \end{bmatrix}$$

The corresponding recurrence formula $G_r(s)$ in Eq. (3) is revised to

$$\boldsymbol{c_{Nr}} = \begin{bmatrix} k_{d1} & 0 & 0 \\ 0 & k_{p1} & 0 \\ 0 & 0 & k_{i1} \end{bmatrix} * \boldsymbol{c_{Ns1}} \oplus \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} * \boldsymbol{c_{Ds1}}$$

$$\boldsymbol{c_{Dr}} = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & b_1 & 0 \\ 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} k_{d1} & 0 & 0 \\ 0 & k_{p1} & 0 \\ 0 & 0 & k_{i1} \end{bmatrix} * \boldsymbol{c_{Ns1}} \oplus \begin{bmatrix} m_1 & 0 & 0 & 0 \\ 0 & b_1 + k_{d1} & 0 & 0 \\ 0 & 0 & k_{p1} & 0 \\ 0 & 0 & 0 & k_{i1} \end{bmatrix} * \boldsymbol{c_{Ds1}}$$

Once coefficient vectors and matrices are extracted from $G_{\infty,\emptyset}(s)$, and the recurrence formula $G_r(s)$ is revised to the form of $\boldsymbol{c_{Nr}}$ and $\boldsymbol{c_{Dr}}$, which directly operate on the coefficients, Algorithm 4 is ready for computing transfer functions of those infinite networks fulfilling the assumptions (A-1) to (A-6) in Sec. 2.

Algorithm 4 Pseudo-code of computing infinite networks' transfer functions. It computes the coefficient tensors `cN` and `cD` of an infinite network's transfer function given its damage case `(l,e)` and the undamaged constants `undCst`.

```
1: function [cN,cD] = tranInf(l,e,undCst)
2: if isEmpty(l) then
3:   [cN,cD] = CUnd(undCst);
4: else
5:   [l1,e1,lS,eS] = partition(l,e);
6: for idx from 1 to nS do
7:   [cNS[idx],cDS[idx]] = tranInf(lS[idx],eS[idx],undCst);
8:   end for
9:   g1Cst = getG1Cst(l1,e1,undCst);
10:  [cN,cD] = Cr(g1Cst,cNS,cDS);
11:  [cN,cD] = simplify(cN,cD);
12: end if
```

The structure of Algorithm 4 is same as that of Algorithm 3, which returns an infinite network's frequency response. The `GUnd()` function in Algorithm 3 is replaced by `CUnd()` function in Algorithm 4, which outputs the coefficient tensors for an infinite network's undamaged transfer function $\boldsymbol{c_{N\infty,\emptyset}}$ and $\boldsymbol{c_{D\infty,\emptyset}}$. The `Gr()` function in Algorithm 3 is replaced by `Cr()` function in Algorithm 4, which implements the computations of revised recurrence formulas $\boldsymbol{c_{Nr}}$ and $\boldsymbol{c_{Dr}}$. The `simplify()` function in Algorithm 4 is similar to that in Algorithm 2.

It is worth noting that the coefficient tensor is not unique for a fractional or irrational transfer function because transfer functions are always univariate with the only variable s. They are manufactured to multivariate polynomials with respect to variables $\phi_i(s)$ to enable that computations can take place only on the coefficients of those fractional or irrational transfer functions. Take the following irrational function $T(s)$ as an example, where

$$T(s) = s + 1 + \sqrt{s+1} = (\sqrt{s+1})^2 + \sqrt{s+1}$$

That $T(s)$ can be regarded as a bivariate polynomial with the two variables $\phi_1(s) = s$ and $\phi_2(s) = \sqrt{s+1}$. Then, both of the following two coefficient matrices represent that $T(s)$:

$$\boldsymbol{c_1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \text{ and } \boldsymbol{c_2} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Finally, the transfer functions obtained by Algorithm 4 for the three example networks in some specific cases are showcased

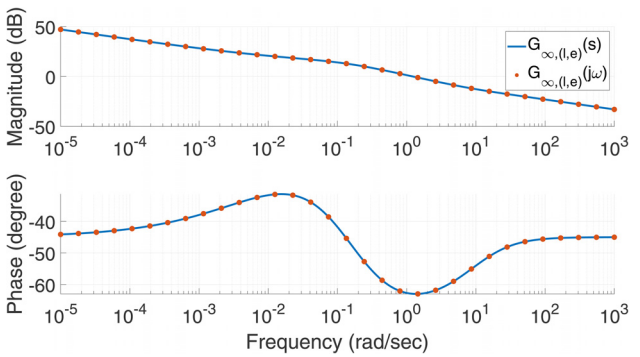

Fig. 22 The consistency between an infinite mechanical tree network's transfer function in Eq. (23) and its corresponding frequency response from Algorithm 3

Downloaded from http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/144/8/081007/6888118/ds_144_08_081007.pdf by University Of Notre Dame user on 13 July 2023

here. For an infinite mechanical tree network in Fig. 1 with the damage case $(\boldsymbol{l},\boldsymbol{e}) = ([k_{2,1}, b_{2,1}], [0.1, 0.2])$, its transfer function is

$$G_{\infty,(\boldsymbol{l},\boldsymbol{e})}(s) = \frac{0.71s^2 + 2.20s^{\frac{3}{2}} + 9.62s + 12.20s^{\frac{1}{2}} + 1.41}{s^{\frac{5}{2}} + 3.11s^2 + 13.60s^{\frac{3}{2}} + 3.40s + 2.00s^{\frac{1}{2}}} \tag{23}$$

For an infinite electrical ladder network in Fig. 2 with the damage case $(\boldsymbol{l},\boldsymbol{e}) = ([r_{2,2}, r_{3,2}], [0.1, 0.1])$, its transfer function is

$$G_{\infty,(\boldsymbol{l},\boldsymbol{e})}(s) = \frac{N_{\infty,(\boldsymbol{l},\boldsymbol{e})}(s)}{D_{\infty,(\boldsymbol{l},\boldsymbol{e})}(s)} \tag{24}$$

where

$$\begin{aligned} N_{\infty,(\boldsymbol{l},\boldsymbol{e})}(s) &= s^4 + s^3\phi_2(s) + 7.2\times10^3 s^3 \\ &\quad +5.2\times10^3 s^2\phi_2(s) + 1.5\times10^7 s^2 \\ &\quad +6.7\times10^6 s\phi_2(s) + 8.1\times10^9 s \\ &\quad +1.5\times10^9\phi_2(s) + 8.0\times10^{10} \\ D_{\infty,(\boldsymbol{l},\boldsymbol{e})}(s) &= 0.1s^4 + 0.1s^3\phi_2(s) + 622s^3 + 421s^2\phi_2(s) \\ &\quad +9.8\times10^5 s^2 + 3.5\times10^5 s\phi_2(s) + 2.6\times10^8 s \\ &\quad +2.2\times10^7\phi_2(s) + 2.5\times10^9 \\ \phi_2(s) &= \sqrt{s^2 + 4020s + 40,100} \end{aligned}$$

For an infinite mechanical ladder network in Fig. 3 with the damage case

$$(\boldsymbol{l},\boldsymbol{e}) = ([k_{p2}, k_{i2}, k_{d2}], [0.1, 0.1, 0.1])$$

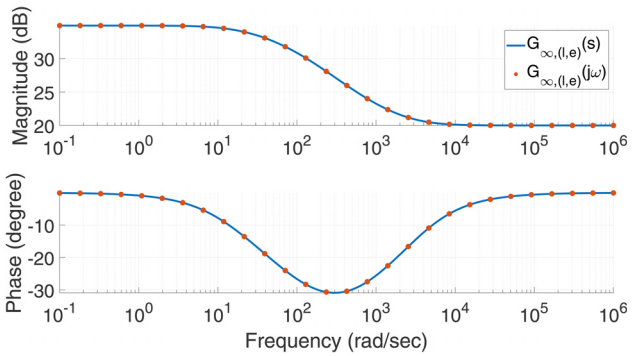

Fig. 23 The consistency between an infinite electrical ladder network's transfer function in Eq. (24) and its corresponding frequency response from Algorithm 3

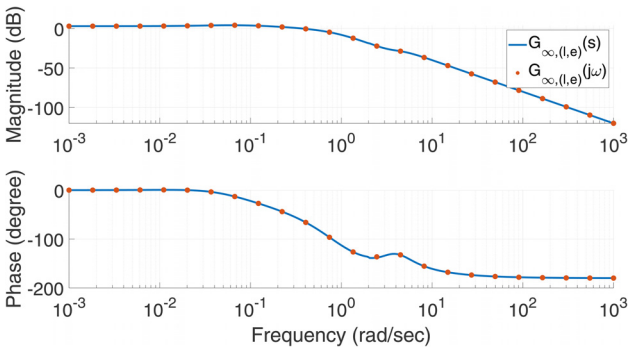

Fig. 24 The consistency between an infinite mechanical ladder network's transfer function in Eq. (25) and its corresponding frequency response from Algorithm 3

the coefficient matrices of its transfer function are

$$c_{N\infty,(\boldsymbol{l},\boldsymbol{e})} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9.21 & 0.05 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 35.6 & 0.42 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 84.0 & 1.33 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 62.1 & 2.70 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5.66 & 0.26 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.14 & 0.01 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$c_{D\infty,(\boldsymbol{l},\boldsymbol{e})} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 12.2 & 0.05 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 69.2 & 0.58 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 219 & 2.91 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 311 & 7.76 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 217 & 5.91 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 74.8 & 0.54 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8.63 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.40 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 \end{bmatrix} \tag{25}$$

with two variables

$$\phi_1(s) = s \text{ and } \phi_2(s) = \sqrt{s^4 + 10s^3 + 49s^2 + 42s + 2}$$

6.3 Correctness Check. The correctness of frequency response and transfer functions returned by Algorithms 3 and 4 for the three example networks when they are infinitely large is demonstrated here. The verification is threefold. First, for the same damage case, a finite network's frequency response should converge to its corresponding infinite network's. That has already been proved previously in Figs. 19–21. Second, for the same infinite network, its frequency response from Algorithm 3 should be consistent with its transfer function from Algorithm 4. That consistency is proved by Figs. 22–24.

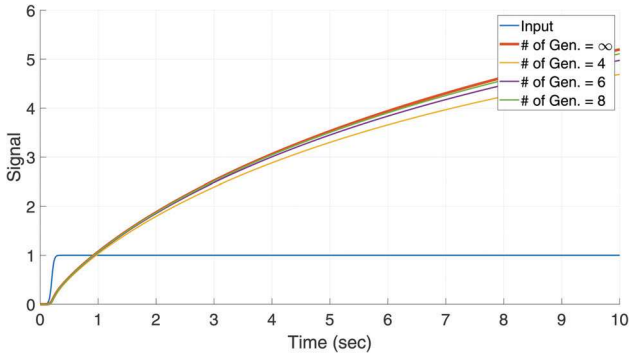

Fig. 25 The blue curve is the input signal $f(t) = 1/(1+e^{-50(t-0.2)})$. The thick red curve is the time-domain response of an infinite mechanical tree network's transfer function in Eq. (23) given by the `lsim()` function in the TOFT toolbox [49]. The other three time-domain response are obtained by using `ode45()` to integrate the differential equations that describe a four, six, and eight-generation mechanical tree network's dynamics

Downloaded from http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/144/8/081007/6888118/ds_144_08_081007.pdf by University Of Notre Dame user on 13 July 2023

Third, the time-domain response of an infinite network's transfer function returned by Algorithm 4 should be consistent with the numerical integration of the corresponding differential equations describing that network's dynamics. For an infinite mechanical tree network with the transfer function in Eq. (23), the time-domain response of that transfer function is obtained by using the `lsim()` function provided by the TOFT toolbox [49]. The input signal is $f(t) = 1/(1 + e^{-50(t-0.2)})$. On the other hand, the numerical integration of differential equations cannot be performed on an infinite mechanical tree network. Therefore, that numerical integration through `ode45()` is carried out on three finite mechanical tree networks, with four, six, and eight generations, respectively. The differential equations describing those three finite mechanical tree networks are similar to those in Sec. 4.3. The convergence of those three finite networks' time-domain response to the infinite network's is shown in Fig. 25, which indirectly verifies that consistency. Similarly, for an infinite electrical ladder and an infinite mechanical ladder with the transfer functions in Eqs. (24) and (25), their correctness is verified in Figs. 26 and 27. Because the transfer functions are irrational in this case, their time-domain responses are computed by a numerical inverse Laplace transform package in Refs. [50] and [51].

7 Discussion

Three discussions yielded by the knowledge about large self-similar networks' frequency response and transfer functions are presented here. Section 7.1 illustrates that approximating the frequency response of a large finite network with shallow damages by using its infinite variant is a reasonable alternative to save computation time. Section 7.2 illustrates the convergence of finite networks' integer-order transfer functions to infinite networks' fractional-order or irrational ones from the perspective of zeros and poles. That convergence can also be leveraged to find rational approximations for some irrational functions. Section 7.3 provides a crucial observation that the frequency response of a dynamic network under different operating conditions could form a set of neighboring plants. That makes robust control methods applicable to dynamic network and, thus, has great potentials to offer a new route of seeking control strategies for dynamic networks.

7.1 Computation Time. Because all four algorithms are recursive, their running time depends on how soon the base case is reached. Therefore, for finite networks, the running time of Algorithms 1 and 2 relies on those networks' size because the base case is their one-generation transfer functions. In contrast, for infinite networks, the running time of Algorithms 3 and 4 depends on the deepest generation where damage resides because the base case is their undamaged transfer functions. As a result, if the frequency response of a large but finite network with shallow damages is needed, approximating that with its infinite variant is possibly a good idea to save the computation time.

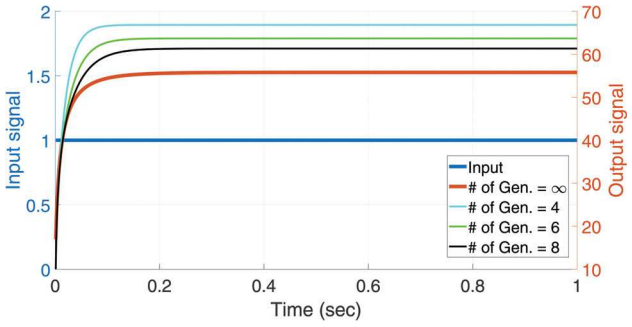

Fig. 26 The blue curve is the input signal, unit-step signal. The thick red curve is the time-domain response of an infinite electrical ladder network's transfer function in Eq. (24) given by the package in Ref. [50]. The other three time-domain response are obtained by using `ode45()` to integrate the differential equations that describe a four, six, and eight-generation electrical ladder network's dynamics

Table 1 Comparisons of computation time between finite and infinite networks for the other two examples

	Finite (20 Generations)	Infinite
Electrical ladder ($[r_{2,2}]$, $[0.1]$)	0.0003 sec	0.00007 sec
Mechanical ladder ($[k_{p2}, k_{i2}, k_{d2}]$, $[0.1, 0.1, 0.1]$)	0.0008 sec	0.00015 sec

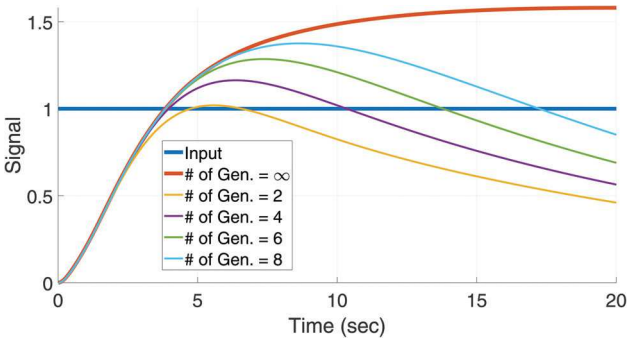

Fig. 27 The blue curve is the input signal, unit-step signal. The thick red curve is the time-domain response of an infinite mechanical ladder network's transfer function in Eq. (25) given by the MATLAB package in Ref. [50]. The other four time-domain response are obtained by using `ode45()` to integrate the differential equations that describe a two, four, six, and eight-generation mechanical ladder network's dynamics

For example, the frequency response of a 20-generation mechanical tree network with a shallow damage case at the first generation, $(\boldsymbol{l}, \boldsymbol{e}) = ([k_{1,1}, b_{1,1}], [0.1, 0.2])$, is required. The computation time of using Algorithm 1 to exactly evaluate that frequency response takes 7.2 s. In contrast, using Algorithm 3 to approximate that with its infinite version only entails 0.005 s. All computations are implemented in MATLAB `R2020b` on a laptop with an Intel `Core i7-10510u` CPU at 1.8 GHz. The comparison between the exact frequency response and the approximated one is shown in Fig. 28, from which we can confirm that the approximation is accurate within the frequency range from 0.2 rad/sec to 100 rad/sec. Comparisons of computation time between finite and infinite networks for the other two examples are listed in Table 1.

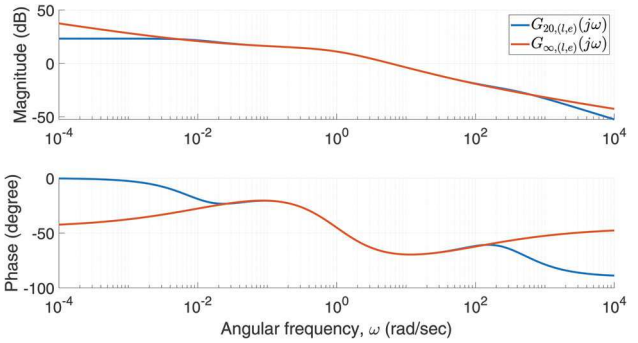

Fig. 28 The blue curve is the exact frequency response of a 20-generation mechanical tree network with the damage case `(l,e)` = $([\boldsymbol{k}_{1,1}, \boldsymbol{b}_{1,1}], [0.1, 0.2])$ obtained by Algorithm 1. The red curve is the approximated one using its infinite variant through Algorithm 3

Downloaded from http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/144/8/081007/6888118/ds_144_08_081007.pdf by University Of Notre Dame user on 13 July 2023

7.2 Fractional or Irrational Nature of Infinite Networks' Dynamics. For an infinite self-similar dynamic network, it is very likely that its transfer function is of noninteger order as illustrated by the three examples in this paper. Some of those transfer functions are fractional, like an infinite mechanical tree network. Others are irrational, like an infinite electrical or mechanical ladder network. However, finite self-similar networks' dynamics are always of integer order as long as they satisfy the assumptions (A-1) to (A-4) in Sec. 2. That can be understood as a convergent behavior that while the size of a finite network grows, its transfer function's order is also increasing. Eventually, when its size grows to infinity, its high-order transfer function converges to a noninteger-order one. As a concrete example, consider a mechanical tree network with the damage case $(\boldsymbol{l},\boldsymbol{e}) = ([k_{1,1}, b_{1,1}], [0.1, 0.2])$, its transfer function as the number of generations grows is listed below. The finite transfer functions are given by Algorithm 2, and the infinite one is given by Algorithm 4

$$G_{1,(\boldsymbol{l},\boldsymbol{e})} = \frac{1}{0.2s + 0.2}$$

$$G_{2,(\boldsymbol{l},\boldsymbol{e})} = \frac{6s^2 + 23.2s + 22}{s^3 + 5.4s^2 + 8.8s + 4}$$

$$\vdots$$

$$G_{\infty,(\boldsymbol{l},\boldsymbol{e})} = \frac{0.7071s + 5.1s^{\frac{1}{2}} + 0.7071}{s^{\frac{3}{2}} + 0.2828s + s^{\frac{1}{2}}}$$

That convergent behavior can also be observed from the perspective of zeros and poles. The effect brought by a zero and a pole on a frequency response's phase is shown in Fig. 29, where a zero increases the phase by 90 deg while a pole decreases it by 90 deg, and there exists a transitional region where that zero and pole locates. As a result, usual integer-order systems' phase always stays at multiples of 90 deg for a wide bandwidth of frequencies. It is not difficult to understand that when lots of zeros and poles concentrate within one region, the dynamical system should behave in a noninteger-order manner in that bandwidth of frequency, as illustrated in Fig. 30.

That concentration of zeros and poles for an integer-order system renders its behavior like a noninteger-order one is exactly what happens to a finite network's dynamics as it grows larger. As a concrete example, using Algorithm 2, we obtain the transfer functions of undamaged finite mechanical trees from the one-generation network to the nine-generation one, that is from $G_{1,\emptyset}(s)$ to $G_{9,\emptyset}(s)$. Figure 31 plots the negative of their zeros' and poles' real parts as the number of generations increases, from which we can observe the aforementioned effect brought by piling up zeros and poles. Note that the region where zeros and poles pile up is consistent with the transitional region in Fig. 16 where phase is around −45 deg.

7.2.1 Rational Approximation. The convergence of finite networks' integer-order transfer functions to infinite networks' fractional-order or irrational ones can be leveraged to approximate some irrational functions with rational expressions. Suppose now the irrational function $f(s) = \sqrt{s^2 + 100s + 100}$ needs to be estimated by rational expressions, which is a part of the infinite electrical ladder network's undamaged transfer function $G_{\infty,\emptyset}(s)$ in Eq. (17). Therefore, we can assume that

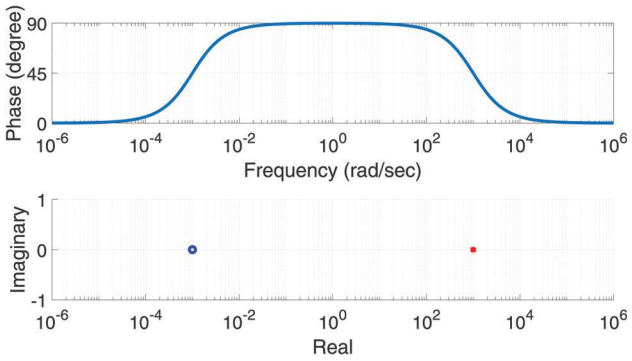

Fig. 29 The effect brought by a zero and a pole on the phase of a dynamical system's frequency response. The transfer function used in this figure is $G(s) = (s+10^{-3})/(s+10^{3})$

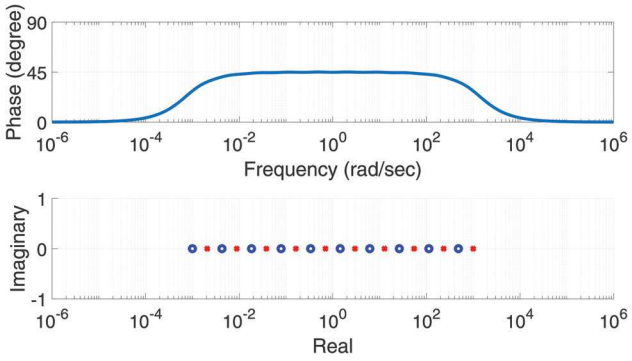

Fig. 30 An integer-order system behaves like a fractional-order system within the region where lots of zeros and poles concentrate. The example dynamical system in this figure would behave like a half-order on within the frequency bandwidth where the phase is 45 deg

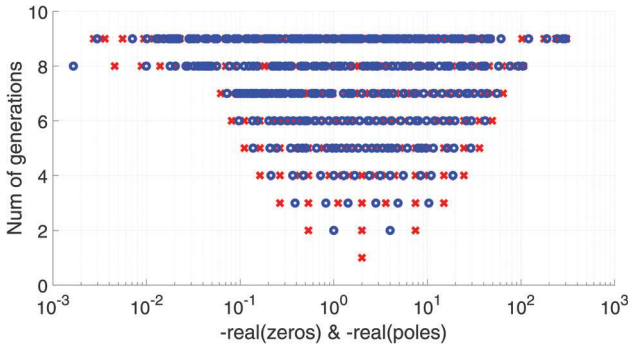

Fig. 31 Negative of the zeros' and poles' real parts for finite undamaged mechanical trees from one-generation to nine-generation. The blue circles are for zeros and the red crosses are for poles

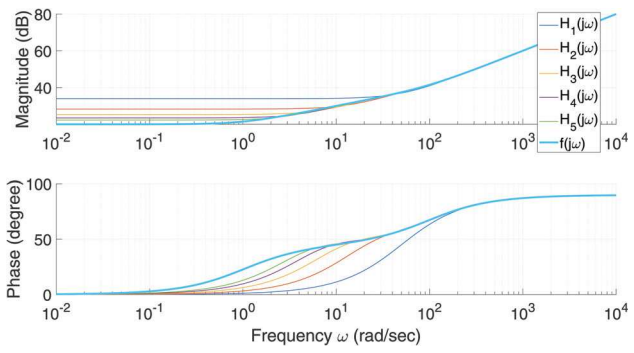

Fig. 32 Rational approximations of $f(s) = \sqrt{s^2+100s+100}$ given by undamaged electrical ladder networks' transfer functions when $s = j\omega$ is an imaginary number

Downloaded from http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/144/8/081007/6888118/ds_144_08_081007.pdf by University Of Notre Dame user on 13 July 2023

$$\frac{2r_1 + 4r_2}{r_1 r_2 c} = 100, \text{ and } \frac{r_1 + 4r_2}{r_1 r_2^2 c^2} = 100$$

which lead to the following three undamaged constants:

$$r_1 = 1,\ r_2 = 5\sqrt{6} + 12, \text{ and } c = \frac{5 + 2\sqrt{6}}{120 + 50\sqrt{6}}$$

When the above three undamaged constants are used to call the algorithm for finite networks' transfer functions in Sec. 4.2, the following convergence of finite undamaged electrical ladder networks' transfer functions can be obtained:

$$G_{1,\emptyset}(s) = \frac{0.9899s + 25.247}{0.9899s + 1}$$
$$G_{2,\emptyset}(s) = \frac{s^2 + 75.505s + 675.26}{s^2 + 51.010s + 50.510}$$
$$G_{3,\emptyset}(s) = \frac{s^3 + 125.51s^2 + 3.9 \times 10^3 s + 1.8 \times 10^4}{s^3 + 101.01s^2 + 2.0 \times 10^3 s + 1.9 \times 10^3}$$
$$G_{4,\emptyset}(s) = \frac{s^4 + 175.51s^3 + 9.5 \times 10^3 s^2 + 1.7 \times 10^5 s + 5.2 \times 10^5}{s^4 + 151.01s^3 + 6.5 \times 10^3 s^2 + 7.1 \times 10^4 s + 6.6 \times 10^4}$$
$$\vdots$$
$$G_{\infty,\emptyset}(s) = \frac{s + 1.0102 + \sqrt{s^2 + 100s + 100}}{2s + 2.0204}$$

As a result, an approximation of $f(s) = \sqrt{s^2 + 100s + 100}$ can be obtained from the above transfer functions, that is

$$H_g(s) = (2s + 2.0204)G_{g,\emptyset}(s) - (s + 1.0102)$$

where g indicates the number of generations in the finite electrical ladder network that is used for approximation. Therefore, the following are the results of rational approximations of the irrational function $f(s) = \sqrt{s^2 + s + 100}$ given by undamaged electrical ladder networks' transfer functions

$$H_1(s) = \frac{0.9899s^2 + 50.495s + 50}{0.9899s + 1}$$
$$H_2(s) = \frac{s^3 + 101.01s^2 + 1.4 \times 10^3 s + 1.3 \times 10^3}{s^2 + 51.010s + 50.510}$$
$$H_3(s) = \frac{s^4 + 151.01s^3 + 5.9 \times 10^3 s^2 + 4.1 \times 10^4 s + 3.5 \times 10^4}{s^3 + 101.01s^2 + 2.0 \times 10^3 s + 1.9 \times 10^3}$$
$$\vdots$$
$$H_\infty(s) = f(s) = \sqrt{s^2 + 100s + 100}$$

The convergence of the above rational expressions $H_g(s)$ to the irrational expression $f(s)$ is plotted in Fig. 32. In addition, that convergence when the independent variable $s = x$ is a real number is shown in Fig. 33.

It is worth emphasizing that there is a drawback of the aforementioned method as compared to Padé approximations. That is the irrational expressions which can be approximated by the proposed method must appear in some infinite networks' transfer functions. That is a very demanding requirement, because it eliminates lots of irrational functions. For example, currently, we have no idea which infinite network's transfer function could include exponential or trigonometric functions, while its finite version is still a rational expression. Nevertheless, that idea of using dynamic networks' transfer functions to find rational approximations is still mentioned here as a side note because that is a viable route and, to our knowledge, there are no similar methods in literature. The proposed method may have potentials when studies regarding dynamic networks' frequency response and transfer functions grow in the future.

7.3 Effects of Varying a Network's Status on Its Dynamics. One crucial discovery after writing down a network's transfer function is that the effect of varying its status on its dynamics can be isolated. That isolation is described as a multiplicative disturbance here. Specifically, a self-similar network's transfer function is always a ratio between two functions of s. Therefore, if at two different statuses a and b, a network's respective transfer functions are

$$G_a(s) = \frac{N_a(s)}{D_a(s)} \text{ and } G_b(s) = \frac{N_b(s)}{D_b(s)}$$

then the effect of changing that network's status from a to b on its transfer function can be expressed in terms of a multiplicative disturbance $\Delta(s)$ where

$$\Delta(s) = \frac{G_b(s)}{G_a(s)} = \frac{N_b(s)D_a(s)}{N_a(s)D_b(s)}$$

There exist at least two meaningful perspectives that can be explored in the context of large networks. The first one is quantifying the approximation error of estimating a finite network's transfer function using the corresponding infinite ones. As an example, for an undamaged mechanical tree network, its transfer functions when it is infinitely large and when it has three generations are

$$G_{\infty,\emptyset}(s) = \frac{1}{1.4142\sqrt{s}}$$
$$G_{3,\emptyset}(s) = \frac{N_{3,\emptyset}(s)}{D_{3,\emptyset}(s)}$$

where

$$N_{3,\emptyset}(s) = 3s^6 + 62s^5 + 428s^4 + 1272s^3 + 1712s^2 + 992s + 192$$
$$D_{3,\emptyset}(s) = s^7 + 30s^6 + 300s^5 + 1288s^4 + 2576s^3 + 2400s^2 + 960s + 128$$

Then, the error of approximating the three-generation network using the infinite network is

$$\Delta(s) = \frac{G_{3,\emptyset}(s)}{G_{\infty,\emptyset}(s)} = \frac{N_\Delta(s)}{D_\Delta(s)}$$

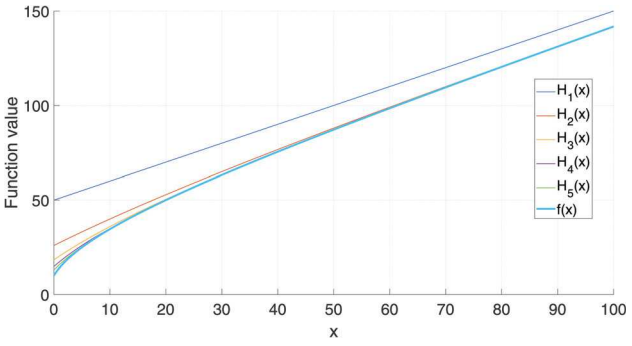

Fig. 33 Rational approximations of $f(s) = \sqrt{s^2+100s+100}$ given by undamaged electrical ladder networks' transfer functions when $s = x$ is a real number

Downloaded from http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/144/8/081007/6888118/ds_144_08_081007.pdf by University Of Notre Dame user on 13 July 2023

where

$$N_\Delta(s) = 4.243s^{\frac{13}{2}} + 87.68s^{\frac{11}{2}} + 605.3s^{\frac{9}{2}} + 1799s^{\frac{7}{2}} + 2421s^{\frac{5}{2}} + 1403s^{\frac{3}{2}} + 271.5s^{\frac{1}{2}}$$

$$D_\Delta(s) = s^7 + 30s^6 + 300s^5 + 1288s^4 + 2576s^3 + 2400s^2 + 960s + 128$$

The second perspective is to study the effect brought by a network's damages on its dynamics. For an infinite mechanical tree network whose damage case is $(\boldsymbol{l}, \boldsymbol{e}) = ([k_{2,1}, b_{2,1}], [0.1, 0.2])$, its transfer function is

$$G_{\infty,(\boldsymbol{l},\boldsymbol{e})}(s) = \frac{0.7071s^2 + 2.2s^{\frac{3}{2}} + 9.6167s + 12.2s^{\frac{1}{2}} + 1.4142}{s^{\frac{5}{2}} + 3.1113s^2 + 13.6s^{\frac{3}{2}} + 3.3941s + 2s^{\frac{1}{2}}}$$

As a result, the effect brought by that damage case is

$$\Delta(s) = \frac{G_{\infty,(\boldsymbol{l},\boldsymbol{e})}(s)}{G_{\infty,\emptyset}(s)} = \frac{s^2 + 3.1113s^{\frac{3}{2}} + 13.6s + 17.2534s^{\frac{1}{2}} + 2}{s^2 + 3.1113s^{\frac{3}{2}} + 13.6s + 3.3941s^{\frac{1}{2}} + 2}$$

The significance here is that while a dynamic network's status is varying within some extents, its frequency response is very likely to also change within some range. That variation can form a set of neighboring plants to which a unified controller can be obtained through robust control methods so that stability and performance are guaranteed for all plants in that set.

8 Future Works and Conclusions

We suggest two future works extended from this paper. The first immediate future work is to test that idea of controlling dynamic networks through robust control methods as presented in Sec. 7.3. We have already started the initial investigation, and applied loop shaping techniques to a network, which estimates the dynamics of a multistory building similar to the one used in Ref. [27]. The resultant unified controller successfully guarantees the stability and enhances the performance of buildings with different number of floors against the impact of the ground movement.

The second future work is brought by the limitation that algorithms computing frequency response and transfer functions in this paper can only be applied to one-dimensional networks. For mechanical networks, that means all nodes can only move back and forth along one direction. For electrical networks, that means the current leaving the input end would eventually come back to the input end, that is no currents could leak to the exterior through other ports except for the input end. That assumption is restrictive and inhibits the idea of promoting frequency-domain methods to more real-life networks. Hence, extending the modeling methods introduced in this paper to high-dimensional networks is important.

When those methods are extended to high-dimensional networks, one challenge is that those systems inevitably become multi-input multi-output, which results in that the number of outputs could grow with the size of those networks. Therefore, a proper definition of system outputs should be carefully determined, so that the number of outputs is invariant with respect to a network's size. For a constant multi-output network, the route of deriving one-dimensional networks' frequency response and transfer functions used in this work is still valuable. The main concept is leveraging recursive algorithms, that is assuming all required results of subnetworks are available, and focusing on how those results would vary if an extra generation is added to those subnetworks. That way of thinking should to a great extent alleviate the burden on coding for high-dimensional networks' frequency response and transfer functions.

In our opinion, extending the modeling methods in this paper to merely two-dimensional networks would have already included a great number of real applications. For example, that could be employed on controlling some multi-agent systems, such as a group of automated guided vehicles moving on the ground. For each topology that this group of agents exhibits, it should have a corresponding frequency response. Then, the next question is whether those frequency responses also form a set of neighboring plants. If that is the case, robust control methods can be applied as well to designing control strategies for those agents when their formation is varying. From a high-level viewpoint, that could be a concrete example of incorporating multi-agent systems' control strategy with their topology, which is one of current research focuses regarding multi-agent systems.

In conclusion, this paper is motivated by the gap between a great number of frequency-domain tools available and the fact that few of them have been applied to dynamic networks in literature. The main reason of that divide is due to the complexity of computing frequency response for a general class of large networks, which is proved to be tractable by this paper once self-similarity is leveraged. In addition, this paper also verifies the correctness of the frequency response and transfer functions of given by the proposed algorithms. Knowledge regarding dynamic networks' frequency response and transfer functions helps us to understand more about their behaviors. Moreover, when they are infinitely large, they become concrete examples of noninteger-order systems, which may assist us to comprehend fractional-order dynamics and irrational systems. Most importantly, this paper builds a bridge connecting large dynamic networks to available frequency-domain tools which hopefully could result in broader impacts within the research field of interconnected systems.

Funding Data

- Division of Civil, Mechanical and Manufacturing Innovation, U.S. National Science Foundation (Grant No. CMMI 1826079; Funder ID: 10.13039/100000147).

References

[1] Biggs, N. L., Lloyd, E. K., and Wilson, R. J., 1976, *Graph Theory 1736-1936*, Clarendon Press, Oxford.
[2] Newman, M. E. J., 2003, "The Structure and Function of Complex Networks," SIAM Rev., **45**(2), pp. 167–256.
[3] Bejan, A., and Merkx, G. W., 2007, *Constructal Theory of Social Dynamics*, Springer, Berlin.
[4] Huberman, B. A., 2001, *The Laws of the Web: Patterns in the Ecology of Information*, MIT Press, Cambridge, MA.
[5] Wang, S., and Ran, C., 2016, "Rethinking Cellular Network Planning and Optimization," IEEE Wireless Commun., **23**(2), pp. 118–125.
[6] Stelling, J., Klamt, S., Bettenbrock, K., Schuster, S., and Gilles, E. D., 2002, "Metabolic Network Structure Determines Key Aspects of Functionality and Regulation," Nature, **420**(6912), pp. 190–193.
[7] Erdős, P., and Rényi, A., 1960, "On the Evolution of Random Graphs," Publ. Math. Inst. Hung. Acad. Sci., **5**(1), pp. 17–60.
[8] Frank, O., and Strauss, D., 1986, "Markov Graphs," J. Am. Stat. Assoc., **81**(395), pp. 832–842.
[9] Watts, D. J., and Strogatz, S. H., 1998, "Collective Dynamics of 'Small-World' Networks," Nature, **393**(6684), pp. 440–442.
[10] Barabaśi, A.-L., and Albert, R., 1999, "Emergence of Scaling in Random Networks," Science, **286**(5439), pp. 509–512.
[11] Pastor-Satorras, R., Castellano, C., Van Mieghem, P., and Vespignani, A., 2015, "Epidemic Processes in Complex Networks," Rev. Mod. Phys., **87**(3), pp. 925–979.
[12] Chen, F., and Ren, W., 2019, "On the Control of Multi-Agent Systems: A Survey," Found. Trends Syst. Control, **6**(4), pp. 339–499.
[13] Ren, W., Beard, R. W., and Atkins, E. M., 2005, "A Survey of Consensus Problems in Multi-Agent Coordination," Proceedings of American Control Conference, Vol. 3, Portland, OR, June 8–10, pp. 1859–1864.
[14] Choi, H.-L., Brunet, L., and How, J. P., 2009, "Consensus-Based Decentralized Auctions for Robust Task Allocation," IEEE Trans. Rob., **25**(4), pp. 912–926.
[15] Aragues, R., Cortes, J., and Sagues, C., 2012, "Distributed Consensus on Robot Networks for Dynamically Merging Feature-Based Maps," IEEE Trans. Rob., **28**(4), pp. 840–854.

Downloaded from http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/144/8/081007/6888118/ds_144_08_081007.pdf by University Of Notre Dame user on 13 July 2023

[16] Rubenstein, M., Cabrera, A., Werfel, J., Habibi, G., McLurkin, J., and Nagpal, R., 2013, "Collective Transport of Complex Objects by Simple Robots: Theory and Experiments," Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems, Saint Paul, MN, May 6–10, pp. 47–54.
[17] Schenato, L., and Fiorentin, F., 2011, "Average Timesynch: A Consensus-Based Protocol for Clock Synchronization in Wireless Sensor Networks," Automatica, **47**(9), pp. 1878–1886.
[18] Spanos, D. P., Olfati-Saber, R., and Murray, R. M., 2005, "Distributed Sensor Fusion Using Dynamic Consensus," IFAC World Congress, Citeseer, Prague, Czech Republic, July 3–8, pp. 1–6.
[19] Simonetto, A., and Leus, G., 2014, "Distributed Maximum Likelihood Sensor Network Localization," IEEE Trans. Signal Process., **62**(6), pp. 1424–1437.
[20] Forero, P. A., Cano, A., and Giannakis, G. B., 2010, "Consensus-Based Distributed Support Vector Machines," J. Mach. Learn. Res., **11**(5), pp. 1663–1707.
[21] Hatanaka, T., Zhang, X., Shi, W., Zhu, M., and Li, N., 2017, "Physics-Integrated Hierarchical/Distributed HVAC Optimization for Multiple Buildings With Robustness Against Time Delays," IEEE 56th Annual Conference on Decision and Control (CDC), Melbourne, Australia, Dec. 12–15, pp. 6573–6579.
[22] Mlynek, P., Misurec, J., Koutny, M., and Silhavy, P., 2012, "Two-Port Network Transfer Function for Power Line Topology Modeling," Radioengineering, **21**(1), pp. 356–363.
[23] Zhao, L., Guo, J., Li, H., and Liu, W., 2009, "The Simulation Analysis of Influence on Jointless Track Circuit Signal Transmission From Compensation Capacitor Based on Transmission-Line Theory," Third IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, Beijing, China, Oct. 27–29, pp. 1113–1118.
[24] Sandidzadeh, M. A., and Dehghani, M., 2013, "Intelligent Condition Monitoring of Railway Signaling in Train Detection Subsystems," J. Intell. Fuzzy Syst., **24**(4), pp. 859–869.
[25] Chen, J., Roberts, C., and Weston, P., 2008, "Fault Detection and Diagnosis for Railway Track Circuits Using Neuro-Fuzzy Systems," Control Eng. Pract., **16**(5), pp. 585–596.
[26] Karimi, H. R., Palacios-Quiñonero, F., Rossell, J. M., and Rubió-Massegú, J., 2013, "Sequential Design of Multioverlapping Controllers for Structural Vibration Control of Tall Buildings Under Seismic Excitation," Proc. Inst. Mech. Eng., Part I J. Syst. Control Eng., **227**(2), pp. 176–183.
[27] Fu, T. S., and Johnson, E. A., 2011, "Distributed Mass Damper System for Integrating Structural and Environmental Controls in Buildings," J. Eng. Mech., **137**(3), pp. 205–213.
[28] Gudarzi, M., 2015, "μ-Synthesis Controller Design for Seismic Alleviation of Structures With Parametric Uncertainties," J. Low Freq. Noise, Vib. Active Control, **34**(4), pp. 491–511.
[29] Hoang, N., Fujino, Y., and Warnitchai, P., 2008, "Optimal Tuned Mass Damper for Seismic Applications and Practical Design Formulas," Eng. Struct., **30**(3), pp. 707–715.
[30] Jonscher, A. K., 1977, "The 'Universal' Dielectric Response," Nature, **267**(5613), pp. 673–679.
[31] Jonscher, A. K., 1999, "Dielectric Relaxation in Solids," J. Phys. D Appl. Phys., **32**(14), pp. R57–R70.
[32] McCullen, N. J., Almond, D. P., Budd, C. J., and Hunt, G. W., 2009, "The Robustness of the Emergent Scaling Property of Random rc Network Models of Complex Materials," J. Phys. D Appl. Phys., **42**(6), p. 064001.
[33] Almond, D. P., Budd, C. J., Freitag, M. A., Hunt, G. W., McCullen, N. J., and Smith, N. D., 2013, "The Origin of Power-Law Emergent Scaling in Large Binary Networks," Phys. A Stat. Mech. Appl., **392**(4), pp. 1004–1027.
[34] Zhai, C., Hanaor, D., and Gan, Y., 2017, "Universality of the Emergent Scaling in Finite Random Binary Percolation Networks," PLoS One, **12**(2), p. e0172298.
[35] Murphy, K. D., Hunt, G. W., and Almond, D. P., 2006, "Evidence of Emergent Scaling in Mechanical Systems," Philos. Mag., **86**(21–22), pp. 3325–3338.
[36] Buła, D., Grabowski, D., Lewandowski, M., Maciążek, M., and Piwowar, A., 2020, "Software Solution for Modeling, Sizing, and Allocation of Active Power Filters in Distribution Networks," Energies, **14**(1), p. 133.
[37] Nasir, M., Hayat, M. F., Jamal, A., and Ahmed, Z., 2021, "Frequency Domain Consensus Control Analysis of the Networked Multi-Agent System With Controller Area Network Bus-Induced Delay," J. Vib. Control, epub.
[38] Ahmed, Z., Saeed, M. A., Jenabzadeh, A., and Weidong, Z., 2021, "Frequency Domain Analysis of Resilient Consensus in Multi-Agent Systems Subject to an Integrity Attack," ISA Trans., **111**, pp. 156–170.
[39] Leyden, K., Sen, M., and Goodwine, B., 2019, "Large and Infinite Mass–Spring–Damper Networks," ASME J. Dyn. Syst., Meas., Control, **141**(6), p. 061005.
[40] Guel-Cortez, A.-J., Méndez-Barrios, C.-F., Kim, E-J., and Sen, M., 2021, "Fractional-Order Controllers for Irrational Systems," IET Control Theory Appl., **15**(7), pp. 965–977.
[41] Sabatier, J., 2020, "Beyond the Particular Case of Circuits With Geometrically Distributed Components for Approximation of Fractional Order Models: Application to a New Class of Model for Power Law Type Long Memory Behaviour Modelling," J. Adv. Res., **25**, pp. 243–255.
[42] Nigmatullin, R. R., and Baleanu, D., 2010, "Is It Possible to Derive Newtonian Equations of Motion With Memory?," Int. J. Theor. Phys., **49**(4), pp. 701–708.
[43] Doehring, T. C., Freed, A. D., Carew, E. O., and Vesely, I., 2005, "Fractional Order Viscoelasticity of the Aortic Valve Cusp: An Alternative to Quasilinear Viscoelasticity," ASME J. Biomech. Eng., **127**(4), pp. 700–708.
[44] Heymans, N., and Bauwens, J.-C., 1994, "Fractal Rheological Models and Fractional Differential Equations for Viscoelastic Behavior," Rheol. Acta, **33**(3), pp. 210–219.
[45] Ni, X., 2021, "Frequency Response of Self-Similar Dynamic Networks With Applications to Health Monitoring and Control," Ph.D. thesis, University of Notre Dame, Notre Dame, IN.
[46] Goodwine, B., 2014, "Modeling a Multi-Robot System With Fractional-Order Differential Equations," IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, May 31–June 7, pp. 1763–1768.
[47] Leyden, K., and Goodwine, B., 2016, "Using Fractional-Order Differential Equations for Health Monitoring of a System of Cooperating Robots," IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, May 16–21, pp. 366–371.
[48] Mayes, J., 2012, "Reduction and Approximation in Large and Infinite Potential-Driven Flow Networks," Ph.D. thesis, University of Notre Dame, Notre Dame, IN.
[49] Chen, Y., Petráš, I., and Xue, D., 2009, "Fractional Order Control - A Tutorial," American Control Conference, St. Louis, MO, June 10–12, pp. 1397–1411.
[50] McClure, T., 2013, "Numerical Inverse Laplace Transform," accessed 27 June, 2021, https://www.mathworks.com/matlabcentral/fileexchange/39035-numerical-inverse-laplace-transform
[51] Joseph Abate, W. W. A., 2006, "Unified Framework for Numerically Inverting Laplace Transforms," INFORMS J. Comput., **18**(4), pp. 408–421.

Downloaded from http://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/144/8/081007/6888118/ds_144_08_081007.pdf by University Of Notre Dame user on 13 July 2023