Proceedings Letters

This section is intended primarily for rapid dissemination of brief reports on new research results within the scope of the IEEE members. Contributions are reviewed immediately, and acceptance is determined by timeliness and importance of the subject, and brevity and clarity of the presentation. Research letters must contain a clear concise statement of the problem studied, identify new results, and make evident their utility, importance, or relevance to electrical engineering and science. Key references to related literature must be given.

Contributions should be submitted in triplicate to the Editor, PROCEEDINGS OF THE IEEE, *345 East 47 Street, New York, NY 10017. The length should be limited to five double-spaced typewritten pages, counting each illustration as a half page. An abstract of 50 words or less and the original figures should be included. Instructions covering abbreviations, the form for references, general style, and preparation of figures are found in "Information for IEEE Authors," available on request from the IEEE Publishing Services Department. Authors are invited to suggest the categories in the table of contents under which their letters best fit.*

After a letter has been accepted, the author's company or institution will be requested to pay a voluntary charge of $95 per printed page, calculated to the nearest whole page and with a $95 minimum, to cover part of the cost of publication.

On Determining a Computable Ordering of a Digital Network

MARC COMEAU AND K. THULASIRAMAN

Abstract—**The problem of determining a computable ordering (whenever it exists) of a digital network is shown to be the same as a well-known problem in Computer Science and Graph Theory literature, namely, the problem of topologically sorting the nodes of an acyclic directed graph.**

I. Introduction

A linear digital network, under certain restrictions, can be completely described by the associated n node equations in matrix form

$$Y(z) = X(z) + H_c(z)Y(z) + z^{-1}H_d(z)Y(z)$$

where $Y(z)$ is a vector of node signals, $X(z)$ is a vector of inputs, and $H_c(z)$ and $H_d(z)$ are appropriately sized coefficient matrices.

To be able to compute the node variables in a sequential manner, the nodes need be renumbered so that the kth node variable depends only on the $k-1$ node variables computed before it. Such a numbering of a digital network is called a *computable ordering* of the network. Clearly, as mentioned in [1], after a computable ordering, the resulting matrix $H_c(z)$ will be lower triangular with a zero diagonal. In this letter, we draw attention to the fact that the problem of determining a computable ordering of a digital network is the same as a well-known problem in Computer Science and Graph Theory literature, namely, the problem of topologically sorting the nodes of an acyclic directed graph [2], [3]. We also point out that the algorithm presented in [4] and [5] for determining a computable ordering is also a familiar algorithm in Computer Science literature. For graph theory terms not defined here, the reader may refer to [2].

II. Computable Node Ordering and Topological Sorting

Let N be a digital network. Since the nodes representing the input and the output do not play any role in determining a computable ordering, we may assume, without any loss of generality, that these nodes are not included in N. Let G be the directed graph of N. Let N_c be obtained from N by deleting the delay elements, and let G_c be the graph of N_c.

Manuscript received April 5, 1983; revised May 23, 1983.
The authors are with the Faculty of Engineering, Concordia University, Montreal, P.Q. H3G 1M8, Canada.

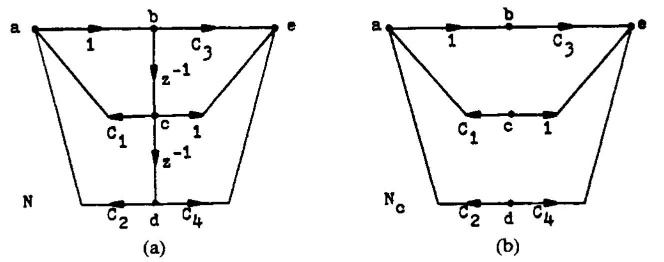

Fig. 1. Digital networks (a) N and (b) N_c, which are obtained from N by deleting the delay elements.

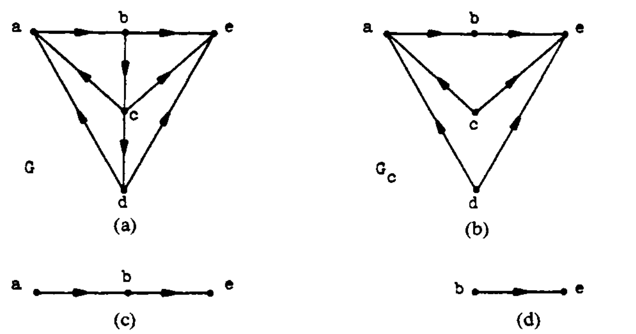

Fig. 2. (a) G and (b) G_c—the graphs obtained from N and N_c, respectively; (c) results from the removal of source nodes c and d, while (d) results from the further removal of a.

For example, consider the digital network N shown in Fig. 1(a). This is the same as the network used in [1] and [4] with the input and output nodes removed. N_c is shown in Fig. 1(b) and the graphs G and G_c are shown in Fig. 2.

An element (i, j) of the matrix $H_c(z)$ is nonzero if and only if in G_c there is an edge directed from the node labeled j to the node labeled i. Thus the matrix $H_c(z)$ contains all the information required to construct the network N_c and hence the graph G_c. In fact, if we replace each nonzero element of $H_c(z)$ by 1, then the resulting matrix will be the transpose of the adjacency matrix of G_c. Note that for purposes of obtaining a computable ordering of N, we need only be concerned with the subnetwork N_c and hence G_c.

Suppose a computable ordering of the nodes of N is possible. Then the variable corresponding to the node labeled k (in the computable ordering) should depend only on the variables which correspond to the nodes labeled $1, 2, \cdots, k-1$. In other words, for every directed edge (i, k) incident into the node k of G_c, i should be less than k. Thus the problem of determining a computable ordering of N_c is equivalent to the problem of assigning integers $1, 2, \cdots, n$ to the nodes of G_c such that for every directed (i, j) in G_c, $i < j$. This problem is known as the problem of *topologically sorting* the nodes of a directed graph [2]. It can be shown [2] that topological sorting of the nodes of a directed graph is possible if and only if the graph is acyclic. Thus summarizing we have the following result.

Theorem 1: A computable node ordering of a digital network exists if and only if the graph G_c is acyclic.

For any acyclic directed graph the following is true [2].

Theorem 2: Every acyclic directed graph has a node with zero in-degree.

The well-known algorithm [2], [3] for topologically sorting the nodes of an n-node directed acyclic graph G_c proceeds as follows:

Select any node with zero in-degree. By Theorem 2, G_c has at least one such node. Label this node with the integer 1. Now remove from G_c this node and the edges incident out of it. Let G_c' be the resulting graph. Since

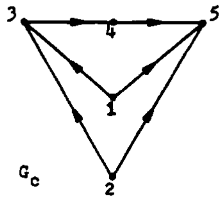

Fig. 3. G_c with its nodes renumbered.

G_c is acyclic, G_c' is also acyclic. So, we can now select a node whose in-degree in G_c' is zero. Label this node with the integer 2. Repeat the above procedure until all the nodes are labeled. This procedure results in a topological sorting of the nodes of G_c.

Noting that a row of zeros in $H_c(z)$ corresponds to a node with zero in-degree in G_c, we can see that the algorithm presented in [4] for determining a computable ordering is the same as the above algorithm for topological sorting.

We shall now illustrate the application of topological sorting on the graph G_c of Fig. 2(b). In this graph, the nodes c and d are source nodes. So we may assign 1 to any one of them, and 2 to the other. Let us assign 1 to node c and 2 to node d. Removing these nodes from G_c, we get the graph in Fig. 2(c). In this graph, node a is the source. So, we should assign 3 to this node. Finally, the graph which results after removing node a is shown in Fig. 2(d). In this graph, node b is the only source. So, it receives 4 in the topological sorting. The remaining node e gets 5. The graph G_c with its nodes renumbered as above is shown in Fig. 3.

III. Conclusions

In this letter we have shown the equivalence of the problem of determining the computable node ordering of a digital network and the problem of topologically sorting the nodes of an acyclic directed graph. It is also shown that the algorithm in [4], [5] is the same as the well-known algorithm for topological sorting. We conclude by noting that topological sorting can be very easily implemented using depth-first search [6].

Acknowledgment

We would like to thank the referee for bringing to our attention the work by Crochiere and Oppenheim [5] where a technique which is essentially the same as topological sorting is described for determining a computable ordering.

References

[1] S. Shiva, "A numerical procedure for renumbering the nodes of a digital network," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-26, pp. 110–111, Feb. 1978.

[2] M. N. S. Swamy and K. Thulasiraman, *Graphs, Networks, and Algorithms*. New York: Wiley-Interscience, 1981.

[3] D. E. Knuth, *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*. Reading, MA: Addison-Wesley, 1968.

[4] S. E. Reyer, J. A. Heinen, and R. J. Niederjohn, "A method for determining a computable ordering of a digital network," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-29, pp. 1226–1227, Dec. 1981.

[5] R. E. Crochiere and A. V. Oppenheim, "Analysis of linear digital networks," *Proc. IEEE*, vol. 63, pp. 581–595, Apr. 1975.

[6] E. M. Reingold, J. Neivergelt, and N. Deo, *Combinatorial Algorithms: Theory and Practice*. Englewood Cliffs, NJ: Prentice-Hall, 1977.

Correction to "Spectrum Analysis: On the Application of Data Windowing to Kay and Marple's Results"

In the above titled letter[1] several errors were inadvertantly introduced. In the Comments by S. Lawrence Marple, Jr., the reference to Fig. 1 was incorrect. Also, the Reply by C. van Schooneveld and M. Sondag was omitted. The entire exchange that should have followed the original letter by C. van Schooneveld and M. Sondag in the June issue[1] of the PROCEEDINGS is given below.

Comments[2] by S. Lawrence Marple, Jr.[3]

In the above correspondence, methods of improving the spectral estimates over those shown in [1] are presented. The author would like to examine the claim of superior spectral estimates and the claim that windowing is necessary to improve least squares estimates.

The author would like to remind readers that the intent in [1] was not to demonstrate relative superiority of various spectral estimation techniques, but to illustrate various features of each technique. The superiority claimed for Fig. 1(f) was due to the fortunate selection of just three sinusoid components in areas A and B of Fig. 1(d). The same argument could be used to select two more sinusoid components near a fraction of sampling frequency of 0.3 since that area of the spectrum has the hint of two "bumps," similar to the "bumps" in area A. The claim of superiority has been used frequently in the literature based on only a few examples. Many different signal classes should be tried and a large ensemble of data sets should be processed before claims of superiority are cited.

A statement is made in the above correspondence that the time series must be windowed to prevent "leakage" effects in the least squares estimates. This is false. Leakage is a frequency-domain phenomenon of Fourier transforms of finite data sets; least squares estimates are time-domain techniques and have no "leakage." A noniterative least squares technique with no windowing (weighting) of the data set was performed to obtain the amplitude, phase, and frequency estimates shown in [1, Table V, p. 1411]. Note that, in at least this particular case, these are closer to the true values than those obtained in Table I, which used an iterative, weighted least squares technique.

The relative strengths of the three sinusoid components to the noise strength, as shown in Fig. 1(b), (f), and (j) are somewhat misleading. Approximately 98 percent of the total energy in the 64 data samples is due to the sinusoids.

Reply[4] by C. van Schooneveld and M. Sondag

1) Our letter addressed two points. First, data weighting is useful in its own right for spectra with large dynamic range. Second, an additional benefit can be obtained by subtracting estimated line components, provided that prior knowledge allows us to do so and that the lines are strong with respect to the local background in the periodogram.

2) Time and frequency domains are linked by a reversible operation, the *FT*. It can make no difference for the errors of an estimate whether the algorithm is executed in one domain or the other. In particular, leakage remains the same. Our example was a least squares fit of a sinusoid to the data. In this case, we obtain the estimated amplitude as a linear operation (a filter) on the data. The spectral sensitivity of this filter is governed by the *FT* of the data window. Thus we can use the window to control the leakage (or bias) in the estimated amplitude.

Closure[5] by S. Lawrence Marple, Jr.

The suggestion to window data prior to a least squares fit is based on the assertion that leakage in the frequency domain implies time-domain data must also exhibit a leakage effect. Leakage *is not* an intrinsic property of a finite time series; leakage *is* a manifestation of the effect of a discrete Fourier transform (DFT) when applied to a finite data set. Many other transforms could have been arbitrarily selected such as the Haar, Slant, or Mellin transforms, just to name three. Each transform has its own unique invertible mathematical relationship with the finite data set on which it operates. The choice of a particular transform will imply an assumption on the nature of the time data outside the window of observed samples. In the case of the DFT, a periodic extension of the data set is implied, whether or not this is a true characterization of the unobserved data. The "leakage" seen in the DFT frequency domain is nothing more than a manifestation of this assumed periodicity. Techniques are available that use the Fourier transform while making a nonperiodic assumption, thereby avoiding leakage in the frequency domain. The autoregressive PSD is one such technique, as in [1, eq. (2.60)] clearly illustrates.

The proposed windowing of data also places more weight on some data samples than others. Why should some data samples be favored more

[1] C. van Schoonevald and M. Sondag, *Proc. IEEE*, vol. 71, no. 6, pp. 776–778, June 1983.

[2] Manuscript received July 29, 1982.

[3] The author was with TASC, McLean Operations, McLean, VA 22102. He is now with Schlumberger Well Services, Houston, TX 77210.

[4] Manuscript received October 29, 1982.

[5] Manuscript received November 15, 1982.

0018-9219/83/1100-1324$01.00 ©1983 IEEE