Mechanical Systems and Signal Processing, vol. 7, no. 2, p. 191-192, Mar. 1993

A Note on the Block Sum Transformation

Paulo Jorge S. G. Ferreira*
Dep. de Electrónica e Telecomunicações / INESC
Universidade de Aveiro
3800 Aveiro Portugal

February 1993

In a recent paper, Bourchard *et al.* [1] proposed the block sum transformation, a method intended as an alternative to the discrete Fourier transform (DFT), with application to the analysis of excited vibrations of linear systems. In this note, we mention the existence of a number of alternative algorithms for the computation of the DFT, and show that the block sum transformation is equivalent to one of these algorithms.

Direct implementation of the DFT in $\mathbf{C}^n$ is equivalent to the computation of

$$\hat{x}_i = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} x_k e^{-j2\pi ki/n}, \tag{1}$$

for $i = 0, 1, \ldots, n-1$, and requires about n complex multiplications/additions for each desired sample $\hat{x}_i$ of the DFT of the x_k. In vector notation, $\hat{\mathbf{x}} = \mathbf{F}\mathbf{x}$, where $\mathbf{F}$ is the (unitary) $n \times n$ Fourier matrix, $\mathbf{x}$ is the n-dimensional time-domain vector, and $\hat{\mathbf{x}}$ is the DFT of $\mathbf{x}$.

The block sum transformation proposed in [1] allows the computation of a subset of the samples $\hat{x}_i$ of the DFT, but uses the m-dimensional vector $\mathbf{y}$ defined by

$$y_v = \sum_{u=0}^{a-1} x_{um+v}$$

*Tel. +351-34-20173, fax +351-34-21740, e-mail `pjf@inesca.pt`.

instead of the n-dimensional vector $\mathbf{x}$. Here, v ranges over $0, 1, \ldots, m-1$ and $n = ma$, with $a > 1$.

Since $m < n$, the advantages of the proposed method relatively to direct DFT computation using (1) include reduced computation time and storage requirements.

A number of alternative algorithms for the computation of the DFT do exist, including Goertzel algorithm [2], and Boncelet algorithm, originally proposed by Boncelet [3]. These algorithms allow the computation of the DFT of a n-dimensional data vector more efficiently than possible using (1). However, the number of arithmetic operations required to compute one DFT coefficient using these methods is still roughly proportional to n.

This number can be drastically reduced if the so-called fast Fourier transform (FFT) algorithms are used. For a description of the FFT algorithm see, for example, [2]. The FFT algorithm seems to be much older than originally anticipated, since it may go back to Gauss [4]. Although it cannot be said that the FFT is the unique choice for fast computation of the DFT, it is fair to say that it is the most popular. The total number of arithmetic operations necessary to find $\hat{\mathbf{x}}$ from $\mathbf{x}$ is proportional to $n \log_2 n$, a remarkable improvement over the algorithms mentioned, whose arithmetic complexity grows quadratically.

If only a small number m of samples $\hat{x}_i$ are required, the FFT algorithm may not be the best choice. Although pruning techniques do exist [5, 6, 7], for sufficiently small m the simpler quadratic algorithms start to become attractive alternatives. We have used these algorithms occasionally, under these circumstances, with good results. Also, for narrow-band signals, a direct decimation technique exists [8].

The block sum transformation can be found in a work of Boncelet [3], who described two simple algorithms to cut down the number of arithmetic operations necessary to compute the DFT of a vector $\mathbf{x} \in \mathbf{C}^n$. His first algorithm is based on the following observation. Let $n = ma$, with $a \neq 1$. Setting $i = \ell a$ and $k = um + v$ in (1), one obtains

$$\hat{x}_{\ell a} \quad = \quad \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} x_k e^{-j2\pi k\ell/m}$$

$$
\begin{aligned}
&= \frac{1}{\sqrt{n}} \sum_{v=0}^{m-1} \sum_{u=0}^{a-1} x_{um+v} e^{-j2\pi(um+v)\ell/m} \\
&= \frac{1}{\sqrt{n}} \sum_{v=0}^{m-1} \left[\sum_{u=0}^{a-1} x_{um+v} \right] e^{-j2\pi v\ell/m} \\
&= \frac{1}{\sqrt{n}} \sum_{v=0}^{m-1} y_v e^{-j2\pi v\ell/m}.
\end{aligned}
$$

Clearly, this is equivalent to the block sum transformation proposed by Bourchard *et al.* [1].

Using the second Boncelet algorithm, which can be applied whenever n is a multiple of 4, this transformation could be further improved from the computational point of view. For details, see [3].

References

[1] D. Bourchard, R. Schmidt, and H. Waller. "The block sum transformation — a new special discrete Fourier transformation method for the analysis of vibrations". *Mechanical Systems and Signal Processing*, vol. 6, no. 5, pp. 483–489, September 1992.

[2] A. V. Oppenheim and R. W. Schafer. *Digital Signal Processing.* Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

[3] C. G. Boncelet. "A rearranged DFT algorithm requiring $n^2/6$ multiplications". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 6, pp. 1658–1659, December 1986.

[4] J. W. Cooley. "How the FFT gained acceptance". *IEEE Signal Processing Magazine*, vol. 9, no. 1, pp. 10–13, January 1992.

[5] J. D. Markel. "FFT pruning". *IEEE Transactions on Audio and Electroacoustics*, vol. 19, no. 4, pp. 305–311, December 1971.

[6] D. P. Skinner. "Pruning the decimation in time FFT algorithm". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 2, pp. 193–194, April 1976.

[7] T. V. Sreenivas and P. V. S. Rao. "FFT algorithm for both input and output pruning". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 3, pp. 291–292, June 1979.

[8] B. Liu and F. Mintzer. "Calculation of narrow-band spectra by direct decimation". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 6, pp. 529–534, December 1978.