TABLE IV
CHARACTERISTIC FEATURES OF WEIGHTED ARRAYS MADE UP OF 25 ELEMENTS. NOTATIONS: D = SPATIAL APERTURE, S = SYMMETRIC, E = EQUALLY SPACED, NE = UNEQUALLY SPACED, NS = ASYMMETRIC

Method of synthesis	Side lobes	*CTR*	*D*	$u_{\text{-3dB}}$	B_p	Charact.
Simulated annealing	-13.51 dB	4.1	50λ	0.0143	0.5 dB	NE, NS
Proposed in [5]	-12.20 dB	5.8	50λ	0.0189	1.8 dB	NE, S
Dolph-Chebychev	-7 dB	11.4	12λ	0.0268	7.0 dB	E, S

Fig. 4(b). To the best of our knowledge, the result reported in [5] is the best in the literature for an array of 25 elements over 50λ. The side lobes were limited to -12.20 dB, the width of the main lobe was $u_{-3\text{dB}} = 0.0189$ and the CTR was equal to 5.8. As mentioned earlier, SA produced an array with limited side lobes (1.3 dB lower), a narrower main lobe, and a lower CTR. Moreover, for the array synthesized by SA, the value of the parameter B_p (for $n = 50$) is equal to 0.5 dB. This value is close to the lowest value reported in [10] for an unequally spaced array obtained for $n \cong 15$, i.e., under simpler conditions than those used in our case. Finally, we evaluated the performances of an array made up of 25 elements spaced at $\lambda/2$ intervals, with the Dolph–Chebychev weight coefficients. Such coefficients allowed main lobe to be contained in the area allowed ($u_{\text{start}} = 0.04$), but caused the side lobes to rise up to -7 dB. The size of the array was only equal to 12λ, but the width of the main lobe was considerable ($u_{-3\text{dB}} = 0.0268$), and so was the CTR value, which was equal to 11.4. Table IV summarizes the characteristics of the arrays compared in this sub-section.

VI. CONCLUSION

The synthesis of unequally spaced arrays, with an average spacing many times as large as $\lambda/2$, exhibits some difficulties related to the height of side lobes. The application of SA to solve this problem yields better results than those obtained by other methods proposed in the literature. Satisfactory results can be achieved thanks to the possibility of synthesizing asymmetric arrays (with more degrees of freedom) and to the simultaneous optimizations of positions and weight coefficients. Future work will be aimed at the synthesis of planar arrays, or at using SA to minimize the number of elements necessary to obtain a specific BP. Moreover, a deeper study of the BP behavior in terms of number of elements and spatial aperture may allow a more precise definition of the optimality concept for an antenna. This will lead us to the formulation of adequate energy functions for obtaining optimal array-sensor configurations.

REFERENCES

[1] M. Zhang, S. Wu, and L. Li, "Improving resolution for array processing by new array geometry and spatial filter," *IEEE Trans. Signal Processing*, vol. 42, pp. 680–683, Mar. 1994.
[2] M. I. Skolnik, G. Nemhauser, and J. W. Sherman III, "Dynamic programming applied to unequally spaced arrays," *IEEE Trans. Antennas Propagat.*, vol. AP-12, pp. 35–43, Jan. 1964.
[3] Y. T. Lo and S. W. Lee, "A study of space-tapered arrays," *IEEE Trans. Antennas Propagat.*, vol. AP-14, pp. 22–30, Jan. 1966.
[4] R. K. Arora and N. C. V. Krishnamacharyulu, "Synthesis of unequally spaced arrays using dynamic programming," *IEEE Trans. Antennas Propagat.*, vol. AP-16, pp. 593–595, July 1968.
[5] P. Jarske, T. Sramaki, S. K. Mitra, and Y. Neuvo, "On the properties and design of nonuniformly spaced linear arrays," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 372–380, Mar. 1988.
[6] S. Kirkpatrick, C. D. Gellatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Sci.*, vol. 220, no. 4598, pp. 671–680, 1983.
[7] W. A. Kuperman, M. D. Collins, J. S. Perkins, and N. R. Davis, "Optimal time-domain beamforming with simulated annealing including application of a priori information," *J. Acoust. Soc. Amer.*, vol. 88, pp. 1802–1810, Oct. 1990.
[8] C. S. Ruf, "Numerical annealing of low-redundancy linear arrays," *IEEE Trans. Antennas Propagat.*, vol. 41, pp. 85–90, Jan. 1993.
[9] T. J. Hayward, "Optimization of the hydrophone placement for acoustic arrays using simulated annealing," *J. Acoust. Soc. Amer.*, vol. 95, pp. 201–212, Jan. 1994.
[10] B. D. Steinberg, *Principles of Aperture and Array System Design*. New York: Wiley, 1976, pp. 132–136, ch. 7.

Design of Nonuniformly Spaced Linear-Phase FIR Filters Using Mixed Integer Linear Programming

Joon Tae Kim, Woo Jin Oh, and Yong Hoon Lee

Abstract— **An optimization problem for designing a nonuniformly spaced, linear-phase FIR filter with minimal complexity is formulated and solved by mixed integer linear programming (MILP). Examples illustrate that the proposed method is useful for designing a wide range of filter types and can outperform subset selection-based design methods.**

I. INTRODUCTION

One approach to the design of efficient FIR filters requiring fewer arithmetic operations than conventional ones is to design nonuniformly spaced FIR filters [1]–[5] by using techniques such as the subset selection method [3]–[5]. This approach results in a filter requiring fewer multiplications and additions at the expense of increased delays. In contrast to most of the other efficient FIR filter design techniques [6]–[9], which are mainly useful for narrowband filter design, the subset selection method is effective in designing a broad range of filter types, including wideband and nonlinear phase FIR filters. Filter design using the subset selection algorithm, however, is computationally inefficient and cannot guarantee a desired filter with minimal complexity. This is because the "subset size" of the subset selection method is determined by trial and error.

In this correspondence, we formulate an optimization problem for designing a nonuniformly spaced, linear-phase FIR filter with minimal complexity. This problem, then, is solved by mixed integer linear programming (MILP) [10]. Through some examples, we shall show that our technique is applicable over a wide range of filter types and can outperform the subset selection-based design methods.

II. THE FILTER DESIGN METHOD

The technique proposed in this section can be applied to linear phase FIR filter design. We shall illustrate our method for the case where impulse response is symmetric and filter length is even.

Manuscript received January 1, 1995; revised July 3, 1995. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Victor E. DeBrunner.
The authors are with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Taejon, Korea.
Publisher Item Identifier S 1053-587X(96)00681-0.

Let $h(n), n = 0, 1, \ldots, 2N-1$ denote the impulse response of a nonuniformly spaced linear-phase FIR filter. It is assumed that $2N$ is greater than $2N_c$, which is the minimum length of a conventional filter needed to meet the filter specifications. If $h(n) = h(2N-1-n)$ for $n = 0, 1, \ldots, N-1$, its frequency response is given by (omitting the linear phase term $\exp(-j(N-1)\omega/2)$)

$$H(\omega) = \sum_{n=0}^{N-1} d(n) \cos\left(n + \frac{1}{2}\right)\omega \tag{1}$$

where $d(n) = 2h(N-1-n)$. We define sequences $I_d(n)$ and $P_d(n)$ indicating the existence and location of the nonzero values of $d(n)$ as follows:

$$I_d(n) = \begin{cases} 0 & \text{when } d(n) = 0 \\ 1 & \text{when } d(n) \neq 0 \end{cases} \tag{2}$$

and

$$P_d(n) = \begin{cases} 0 & \text{when } d(n) = 0 \\ n & \text{when } d(n) \neq 0. \end{cases} \tag{3}$$

Note that $\sum_{n=0}^{N-1} I_d(n)$ is the number of nonzero values of $d(n)$ and that $\max_n[P_d(n)]$ is proportional to the memory required in implementing $h(n)$. In order to design a filter with minimal complexity, we formulate the following optimization problem:

$$\underset{d(n)}{\text{Minimize}}\, J(d(n)) = J_A(d(n)) + J_D(d(n)) \tag{4}$$

subject to

$$\begin{array}{ll} |H(\omega) - 1| \leq \delta(\omega) & \text{when } \omega \in \text{passband} \\ |H(\omega)| \leq \delta(\omega) & \text{when } \omega \in \text{stopband} \end{array}$$

where $J_A(\cdot)$ and $J_D(\cdot)$ are the costs for arithmetic operations (multiplications and additions) and delay, respectively, and $\delta(\omega)$ is the ripple size given in the filter specifications. Our objective is to find $d(n)$ minimizing the cost $J(d(n))$ under the constraints of filter specifications. Before defining the costs $J_A(\cdot)$ and $J_D(\cdot)$, we describe some desirable properties that need to be satisfied by these costs.

Desirable Properties of the Costs

Consider two sequences $d_1(n)$ and $d_2(n)$, $n = 0, 1, \ldots, N-1$, having M_1 and M_2 nonzero values, respectively.

- When $M_2 > M_1$, it is desired that

$$J(d_2(n)) > J(d_1(n)). \tag{5}$$

- When $M_1 = M_2$, it is desired that

$$J_A(d_1(n)) = J_A(d_2(n)) \tag{6}$$

and

$$\begin{array}{l} J_D(d_1(n)) > J_D(d_2(n)) \quad \text{whenever} \\ \max_n[P_{d_1}(n)] > \max_n[P_{d_2}(n)]. \end{array} \tag{7}$$

These properties indicate that between two sequences, the one that needs more computations should have the larger cost and that when the required computational load is the same, the one with more delay has the larger cost. Now, we define

$$J_A(d(n)) = c_A \sum_{n=0}^{N-1} I_d(n) \tag{8}$$

with c_A a constant, since the number of arithmetic operations is proportional to $\sum_{n=0}^{N-1} I_d(n)$. Note that the desired property in (6) is satisfied by this $J_A(\cdot)$. The following delay cost $J_D(\cdot)$ meets the property in (7)

$$J_D(d(n)) = \sum_{n=0}^{N-1} q(n) I_d(n) \tag{9}$$

where

$$q(n) = \begin{cases} 1 & \text{when } n = 0 \\ 2 & \text{when } n = 1. \\ \sum_{i=0}^{n-1} q(i) & \text{when } n \geq 2 \end{cases} \tag{10}$$

Here, $q(n) = \sum_{i=0}^{n-1} q(i) = 3 \cdot 2^{n-2}$ for $n \geq 2$.

Observation 1: The delay cost in (9) satisfies the desired property in (7).

Proof: Let $\max_n[P_{d_i}(n)] = m_i, i = 1, 2$, and assume $m_1 > m_2$. Then $\sum_{n=0}^{N-1} q(n)I_{d_1}(n) = \sum_{n=0}^{m_1} q(n)I_{d_1}(n) > \sum_{n=0}^{m_1-1} q(n) \geq \sum_{n=0}^{m_1-1} q(n)I_{d_2}(n) \geq \sum_{n=0}^{m_2} q(n)I_{d_2}(n) = \sum_{n=0}^{N-1} I_{d_2}(n)$, where the first inequality comes from the fact that $q(m_1) \geq \sum_{i=0}^{n-1} q(i)$ for $n > 0$. □

Now the overall cost

$$J(d(n)) = c_A \sum_{n=0}^{N-1} I_d(n) + \sum_{n=0}^{N-1} q(n) I_d(n) \tag{11}$$

that meets the desired property in (5) is found by adjusting c_A. The observation below addresses this issue.

Observation 2: The property in (5) is satisfied by $J(\cdot)$ in (11) if

$$c_A \geq \sum_{n=0}^{N-1} q(n) = 3 \cdot 2^{N-2}. \tag{12}$$

Proof: Assume that $c_A \geq \sum_{n=0}^{N-1} q(n)$. Then $J_A(d_2(n)) = \sum_{n=0}^{N-1} c_A I_{d_2}(n) = M_2 c_A \geq (M_1+1)c_A \geq M_1 c_A + \sum_{n=0}^{N-1} q(n) \geq M_1 c_A + \sum_{n=0}^{N-1} q(n) I_{d_1}(n) = J(d_1(n))$. Since $J_D(d_2(n)) > 0$ for nonzero M_2, $J(d_2(n)) = J_A(d_2(n)) + J_D(d_2(n)) > J(d_1(n))$. □

The overall cost in (11) is a linear function of $I_d(n)$. Next we shall show that $I_d(n)$ can be generated from $d(n)$ using linear inequalities.

Observation 3: Suppose that for a given $d(n)$, $n = 0, 1, \ldots, N-1$, u and v are positive constants satisfying $u < |d(n)| < v$ for all nonzero values of $d(n)$. If we define a sequence $I(n) \in \{0, 1\}, n = 0, 1, \ldots, N-1$, such that

$$\frac{1}{v}|d(n)| \leq I(n) \leq \frac{1}{u}|d(n)| \tag{13}$$

then, $I(n) = I_d(n)$.

The proof for this observation is simple and is not shown here. Summarizing the above results, the optimization problem in (4) is rewritten as

$$\underset{d(n)}{\text{Minimize}}\, J(d(n)) = \sum_{n=0}^{N-1} [c_A + q(n)] I_d(n) \tag{14}$$

subject to

$$\begin{array}{ll} |H(\omega) - 1| \leq \delta(\omega) & \text{when } \omega \in \text{passband,} \\ |H(\omega)| \leq \delta(\omega) & \text{when } \omega \in \text{stopband,} \\ |d(n)|/v \leq I_d(n) \leq |d(n)|/u & \text{for } n = 0, 1, \ldots, N-1, \\ 0 \leq I_d(n) \leq 1 & I_d(n) \text{ is an integer.} \end{array}$$

Here, $c_A = 3 \cdot 2^{N-2}$, $N > N_c$, and u and v, respectively, are set at sufficiently small and large values. (In our design examples, which are presented in the next section, N is set at $1.5N_c$, u and v are set at 10^{-4} and 10^2, respectively.) This problem can be solved by MILP, treating $d(n)$ and $I_d(n)$ as variables. Note that the length of the resultant impulse response $h(n)$ is not $2N$ but $2 \cdot \max_n[P_d(n)]$.

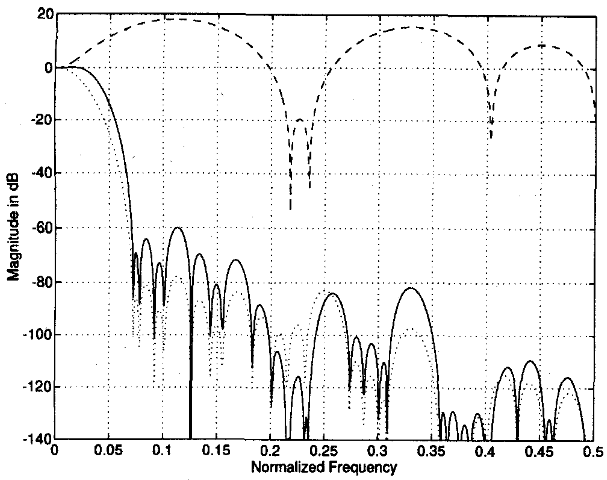

Fig. 1. Frequency responses of the prefilter in [5] (dotted line), the equalizer designed using the proposed algorithm (dashed line), and the overall cascaded filter (solid line).

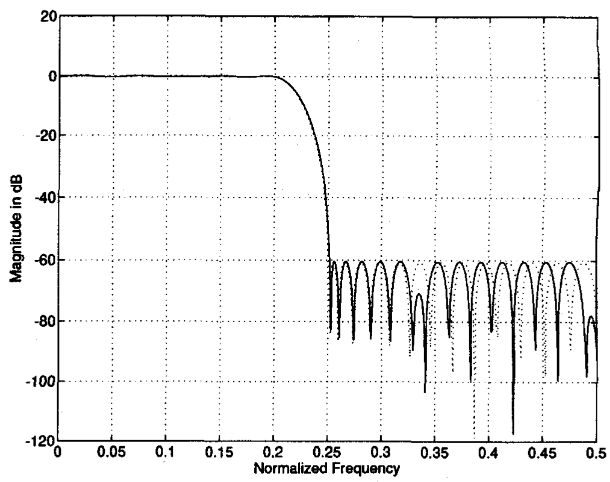

Fig. 2. Frequency responses of conventional filter (dotted line), and filter designed in example 2 (solid line).

III. Design Examples

The MILP problems considered in this section were solved by using the commercial package noted in [11]. The required computation time for each problem was less than five minutes in a Sparc 2 workstation.

Example 1 (Prefilter Equalizer Design): This example compares our method with the subset selection-based method by considering the design example 4 in [5]. The desired specifications in normalized frequency are as follows:

passband: $F \in [0, 0.021]$, stopband: $F \in [0.07, 0.5]$,
$\mathrm{dB}_p = 0.2$ dB maximum passband ripple,
$\mathrm{dB}_s = 60$ dB minimum stopband attenuation.

This filter was designed as a cascade of a multiplierless prefilter and an equalizer. Given the prefilter in [5], we designed an equalizer such that the cascade meets the above specifications. The frequency responses of the prefilter, the equalizer designed by the proposed method, and the overall filter are shown in Fig. 1. The impulse response of the equalizer is $h(0) = h(9) = -1.8020355$, $h(4) = h(5) = 2.2936090$, and $h(i) = 0$ for $i = 1, 2, 3, 6, 7, 8$. This equalizer requiring two multiplications and nine delays is considerably simpler than the equalizer in [5]—designed by the subset selection method—which requires four multiplications and 32 delays.

Example 2 (Wideband Filter Design): This example illustrates that our method is useful in designing a wideband filter. The desired specifications in normalized frequency are as follows:

passband: $F \in [0, 0.2]$, stopband: $F \in [0.25, 0.5]$,
$\mathrm{dB}_p = 0.2$ dB maximum passband ripple,
$\mathrm{dB}_s = 60$ dB minimum stopband attenuation.

Note that we have purposely chosen a "difficult" wideband problem where the passband and stopband have slightly different widths. The Parks–McClellan algorithm determined that a length $2N_c = 48$ linear-phase FIR filter meets this specification. Using our method, we were able to find a length 50, nonuniformly spaced FIR filter satisfying the given specification. The frequency response of this filter is shown in Fig. 2. The impulse response has 40 nonzero coefficients. Thus, at least four multiplications and eight additions are reduced at the expense of only two delays, as compared with the conventional filter.

Example 3 (Interpolated FIR Filter Design): This example shows that our method is also useful for designing the interpolated FIR (IFIR) filter [7], [8]. We consider an example in [7]. The desired filter specifications in normalized frequency are as follows:

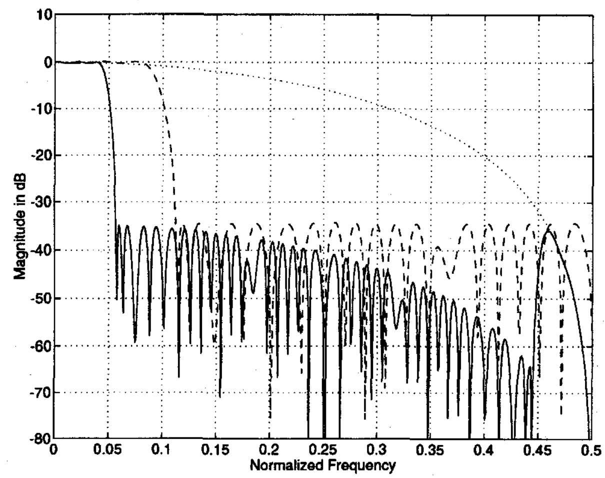

Fig. 3. Frequency responses of interpolator (dotted line), model filter designed using the proposed algorithm (dashed line), and overall interpolated FIR filter (solid line).

passband: $F \in [0, 0.0404]$, stopband: $F \in [0.0556, 0.5]$,
$\mathrm{dB}_p = 0.1612$ dB maximum passband ripple,
$\mathrm{dB}_s = 34.548$ dB minimum stopband attenuation.

In [7], the filter was designed as a cascade of twofold expanded model filter and a second-order interpolator; the model filter, whose passband and transition widths are twice the corresponding widths of the desired filter, was designed using the Parks–McClellan algorithm. The resulting model filter has the length $2N_c = 56$. We applied our method to the design of this model filter and obtained a length 58, nonuniformly spaced filter. The frequency responses of this model filter, interpolator, and the overall IFIR filter are shown in Fig. 3. The impulse response of this filter has 44 nonzero coefficients. Thus, at least six multiplications and 12 additions are reduced at the expense of only two delays as compared with the conventional filter.

References

[1] N. Maeda, "Transversal filters with nonuniform tap spacing," *IEEE Trans. Circuit Syst.*, vol. CAS-27, pp. 1–11, Jan. 1980.

[2] N. M. Mitrou, "Results on nonrecursive digital filters with nonequidistant taps," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-

33, no. 6, pp. 1621–1624, Dec. 1985.

[3] J. R. Khorrami, "Study of digital filter design and frequency estimation by complex subset selection," Ph.D. dissertation, Univ. Rhode Island, Kingston, USA, 1979.

[4] R. J. Hartnett and G. F. Boudreaux-Bartels, "Design of efficient parallel hybrid FIR filters using dynamic programming and subset selection methods," in *Proc. 1990 Int. Conf. Acoust., Speech, Signal Processing*, pp. 1337–1340.

[5] ——, "On the use of cyclotomic polynomial prefilters for efficient FIR filter design," *IEEE Trans. Signal Processing*, vol. 41, no. 5, pp. 1766–1779, May 1993.

[6] J. Adams and A. Wilson, Jr., "Some efficient digital prefilter structures," *IEEE Trans. Circuit Syst.*, vol. CAS-31, no. 3, pp. 260–265, Mar. 1984.

[7] Y. Neuvo, D. Cheng-Yu, and S. K. Mitra, "Interpolated finite impulse response filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 563–570, June 1984.

[8] T. Saramaki, Y. Neuvo, and S. K. Mitra, "Design of computationally efficient interpolated FIR filters," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 70–87, Jan. 1988.

[9] Z. Jing and A. T. Fam, "A new structure for narrow transition band, lowpass digital filter design," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 362–370, Apr. 1984.

[10] Y. C. Lim and S. R. Parker, "FIR filter design over a discrete powers-of-two coefficient space," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 583–591, June 1983.

[11] *Using the CPLEX Callable Library and CPLEX Mixed Integer Library.* Incline Village, NV: CPLEX Optimization, 1994.

Fast Algorithm for FIR Filtering in the Transform Domain

Hwang-Cheng Chiang and Jie-Cherng Liu

Abstract— **In this correspondence, a more efficient algorithm than the overlap-and-add (or overlap-and-save) method is developed for quickly computing the convolution involving very long impulse response. By dividing the input and the impulse response into segments of proper length, the algorithm requires fewer multiplications than the existing methods and therefore reduces the complexity of the filtering process.**

I. Introduction

Finite impulse response (FIR) filters are often used in the design of data communication and speech processing systems for the following characteristics [1], [2]:

1) Linear phase designs are easily achieved.
2) Arbitrary frequency responses can be approximated arbitrarily closely for sufficiently long impulse response.
3) Adaptive equalizers used in data communication have simple tap coefficient adjustment algorithms.

Manuscript received March 18, 1994; revised June 23, 1995. The associate editor coordinating review of this paper and approving it for publication was A. N. Akansu.

The authors are with the Department of Electrical Engineering, Tatung Institute of Technology, Taipei, Taiwan, R.O.C. (e-mail: jcliu@ee.ttit.edu.tw).

Publisher Item Identifier S 1053-587X(96)00687-1.

For filters with low or moderate sample size (less than 128), time domain convolution is usually used to obtain the system output. If systems with larger impulse response are required, it is inefficient to directly compute the convolution. An alternative to obtain the system output is to use the transform domain algorithms, the so-called overlap-and-add (OLA) or overlap-and-save (OLS) methods. However, in some applications we are faced with a problem of feeding an input function to filters with sharp-cutoff. On implementing the convolution with the OLA or OLS method, such long filter impulse may be too large for the discrete Fourier transform (DFT) to compute. To perform a convolution in the transform domain while using the DFT efficiently, an improved algorithm is derived by dividing both the long input and the impulse sequences into proper length of segments. The input segments are convolved with each impulse section and obtain a corresponding output. After properly combining the output sequence corresponding to each impulse section, we may obtain the linear convolution of the original sequence. From the analysis of computational complexity, the modified algorithm behaves better than the OLA and OLS methods.

II. Modified OLA and OLS Algorithm

The OLA and OLS algorithms are reviewed below for the convenience of developing the improved method. Assume the system impulse $h(n)$ is M points of length; for the OLA method, the input sequence $x(n)$ is divided into segments of length $L1$, with the kth section $x_k(n)$ defined as

$$x_k(n) = \begin{cases} x(n), & kL1 \le n \le (k+1)L1 - 1 \\ 0, & \text{otherwise} \end{cases}. \tag{1}$$

Then

$$x(n) = \sum_{k=-\infty}^{\infty} x_k(n). \tag{2}$$

The overall system output $y(n)$ is equal to the sum of $x_k(n)$ convolved with $h(n)$

$$\begin{aligned} y(n) &= x(n) * h(n) \\ &= \sum_{k=-\infty}^{\infty} x_k(n) * h(n). \end{aligned} \tag{3}$$

Since $x_k(n)$ has $L1$ nonzero points and $h(n)$ has M points, to obtain a linear convolution the DFT must have at least $(L1 + M - 1)$ points. Thus, the output segments will overlap each other by $(M-1)$ points. The overlapping sequences are added up to produce the entire convolution.

For the OLS method, we section the input $x(n)$ into segments of length N so that each segment overlaps the preceding section by $(M - 1)$ points.

$$x_k(n) = x(n + k(N - M + 1)) \qquad 0 \le n \le N - 1. \tag{4}$$

By performing a circular N-point convolution for each N-point segment with the M-point impulse response, the first $M - 1$ aliased points are discarded and the remaining $N - M + 1$ points are combined with the nonaliased parts of other segments to form the linear convolution.