# NEW SPLIT ALGORITHMS FOR LINEAR LEAST SQUARES PREDICTION FILTERS WITH LINEAR PHASE

*Wen-Hsien Fang and Yang-Lung Hwang*

Department of Electronic Engineering
National Taiwan Institute of Technology, Taiwan, R.O.C.
Email: whf@et.ntit.edu.tw

## ABSTRACT

This paper is concerned with the development of new split algorithms for the design of linear least squares prediction filters with linear phase. The proposed fast algorithm, which fully expresses the inherent symmetry of the problem, requires lower computational complexity than other existing ones. Moreover, unlike other existing ones, the new recurrences involve only the order updates, which lend themselves to more efficient hardware implementations. For parallelization consideration, a new split Schur-like algorithm is also proposed to overcome the nonparallelizable inner product. Some numerical simulation results are provided to verify the proposed fast algorithms and highlight possible applications.

## 1. INTRODUCTION

The design of linear prediction filter (LPF) has been an active research area in signal processing [1]. However, in many practical applications, the direct application of LPF without any constraint will introduce phase distortion which is an undesirable characteristic, especially in situations where the timing of the filter response is crucial. In this paper, we present some efficient algorithms to implement the optimal linear phase LPF in the least squares sense.

Marple [2] was the first to consider this problem and develop some efficient algorithms which are imbedded in the recurrences for the modified covariance (or forward-backward linear prediction) algorithm considered in [3]. These algorithms were later improved by [4] and [5] via some matrix inversion lemmas for partitioned matrix which has some specific structure.

Since linear phase constraint is equivalent to imposing the impulse response of the filter to be symmetric, the same problem has also been addressed from the aspect of two-sided linear prediction with symmetric weighting [6, 7], in which the problem was reformulated as a system of equations with a close-to-block-Toeplitz structure and then solved by using the multichannel Levinson algorithm [1]. The applications of these algorithms to high-resolution spectrum estimation and blind deconvolution of symmetric noncausal impulse responses were also considered in [6] and [8], respectively.

This work is supported by National Science Council of R.O.C. under contract NSC82-0404-E011-197.

Recently, Delsarte and Genin [9] have noted the inherent symmetry in the traditional fast algorithms for the linear prediction problems and proposed the three-term "split algorithm" to reduce the number of multiplications that would have been required. Motivated from the success of the split algorithms, in this paper, we develop a "split" version of the algorithms considered in [2] by further exploiting the centrosymmetric property of the system matrix as that of [9]. Since the inherent symmetry is fully expressed, the proposed fast algorithms require lower computational complexity than other existing ones. For parallelization consideration, a corresponding highly parallelizable Schur algorithm is also addressed.

## 2. NEW THREE-TERM RECURRENCE FOR LINEAR PHASE PREDICTION FILTERS

The problem considered is as follows: given a sequence of data $\{x(M), x(M+1), \ldots, x(N)\}$, our objective is to find a set of linear prediction coefficients $c_p(1), \ldots, c_p(p)$ so that the unwindowed summation of the linear prediction errors,

$$\sum_{n=M+p}^{N-p} \left[x(n) - \sum_{q=1}^{p} c_p(q) \cdot [x(n+q) + x(n-q)]\right]^2 \quad (1)$$

is minimized, where we have used the fact that a linear phase filter has a symmetric impulse response. Using the orthogonality principle yields the following normal equation:

$$\Phi_{2p+1}\Gamma_{2p+1} = \begin{bmatrix} 0 \\ \sigma_p^2 \\ 0 \end{bmatrix} \quad (2)$$

where

$$\Phi_{2p+1} = \sum_{n=M+2p}^{N} [\mathbf{x}_{2p+1}(n)\mathbf{x}_{2p+1}^T(n) + J\mathbf{x}_{2p+1}(n)\mathbf{x}_{2p+1}^T(n)J]$$

with $\mathbf{x}_k(n) = [x(n), \ldots, x(n-k+1)]^T$ and $J$ being an exchange matrix of an appropriate size with one's along the diagonal and zeros' elsewhere, the superscript $^T$ denotes matrix transposition, $\Gamma_{2p+1} = [\mathbf{c}_p^T J, 1, \mathbf{c}_p^T]$ with $\mathbf{c}_p = [c_p(1), \ldots, c_p(p)]^T$, and $\sigma_p^2$ is the corresponding minimum prediction error.

The development of the proposed fast algorithm relies on the centrosymmetric and close-to-Toeplitz-plus-Hankel



0-7803-2431-5/95 $4.00 © 1995 IEEE

structures of the system matrix $\Phi_m$ which admits the following decompositions:

$$\Phi_{m+2} = \begin{bmatrix} \Phi'_{m+1} & \mathbf{s}_{m+1} \\ \mathbf{s}^T_{m+1} & s^o_{m+1} \end{bmatrix} = \begin{bmatrix} q^o_m & \mathbf{q}^T_m J & t_{m+1} \\ J\mathbf{q}_m & \Phi''_m & \mathbf{q}_m \\ t_{m+1} & \mathbf{q}^T_m & q^o_m \end{bmatrix} \quad (3)$$

where

$$\Phi'_{m+1} = \Phi_{m+1} - H_{m+1}H^T_{m+1} \quad (4)$$

$$\Phi''_m = \Phi_m - H_m H^T_m - JH_m H^T_m J \quad (5)$$

with $H_m = [\mathbf{x}_m(M+m-1), J\mathbf{x}_m(N)]$

$$\mathbf{s}_{m+1} = \sum_{n=M}^{N-m-1} \mathbf{x}_{m+1}(m+n+1)x(n) + \sum_{n=M}^{N-m-1} J\mathbf{x}_{m+1}(m+n)x(m+n+1) \quad (6)$$

$$q^o_m = \mathbf{x}^T_{N-M-m}(N-m-1)\mathbf{x}_{N-m-1}(N-m-1) + \mathbf{x}^T_{N-M-m}(N)\mathbf{x}_{N-M-m}(N) = s^o_{m+1} \quad (7)$$

$$t_{m+1} = \mathbf{x}^T_{N-M-m}(N)\mathbf{x}_{N-M-m}(N-m-1) + \mathbf{x}^T_{N-M-m}(N-m-1)\mathbf{x}_{N-M-m}(N) \quad (8)$$

$$\mathbf{q}_m = \sum_{n=M}^{N-m-1} \mathbf{x}_m(m+n)x(n) + \sum_{n=M}^{N-m-1} J\mathbf{x}_m(m+n)x(m+n+1) \quad (9)$$

Note that $H_m$ can be interpreted as the boundary effect due to the unwindowed assumption of the problem. Here, we introduce a symmetric auxiliary matrix $U_m$ to account for this effect. $U_m$ is defined as

$$\Phi_m U_m = H_m + JH_m \quad (10)$$

It can be readily verified that $H_m$ renders the following decompositions:

$$H_{m+2} = \begin{bmatrix} \mathbf{h}^T_{m+1} \\ H_{m+1} \end{bmatrix} = \begin{bmatrix} H''_{m+1} \\ \mathbf{h}^T_0 \end{bmatrix} = \begin{bmatrix} \mathbf{h}^T_{m+1} \\ H''_m \\ \mathbf{h}^T_0 \end{bmatrix} \quad (11)$$

where

$\mathbf{h}^T_k = [x(M+k), x(N-k)]$ and $H''_k = [\mathbf{x}_k(M+k), J\mathbf{x}_k(N-1)]$

The developed algorithm comprises the following three main recurrences:

*(a) Update the normalized symmetric forward-backward prediction filter* $\mathbf{g}_m$ :

Similar to [2], the algorithm developed is imbedded in that for the modified covariance problem [3]. However, in order to yield more efficient recurrences, we consider the normalized symmetric forward-backward prediction filter $\mathbf{g}_m$, which is defined as follows:

$$\mathbf{g}_{m+1} = \frac{1}{1+a_m(m)} \left( \begin{bmatrix} 1 \\ \mathbf{a}_m \end{bmatrix} + \begin{bmatrix} J\mathbf{a}_m \\ 1 \end{bmatrix} \right)$$

where $\mathbf{a}_m = [a_m(1), \cdots, a_m(m)]^T$ is the $m^{th}$ order prediction filter for the modified covariance algorithm [3], i.e. $\mathbf{a}_m$ minimizes $\sum_{n=M+m}^{N}[(x(n)-\sum_{k=1}^{m} a_m(k)x(n-k))^2+(x(n-m)-\sum_{k=1}^{m} a_m(k)x(n-m+k))^2]$. It can be shown that $\mathbf{g}_m$ obeys the following recurrence:

$$\mathbf{g}_{m+2} = \begin{bmatrix} \mathbf{g}_{m+1} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{g}_{m+1} \end{bmatrix} - \begin{bmatrix} 0 \\ \mathbf{g}_m \\ 0 \end{bmatrix} \mu_{m+1} + \begin{bmatrix} 0 \\ U_m \\ 0 \end{bmatrix} \mathbf{b}_{m+1} \quad (12)$$

where $\mu_{m+1}$ and $\mathbf{b}_{m+1}$ can be derived by premultiplying both sides of (12) by $\Phi_{m+2}$ and taking the appropriate partitions in (3) to yield

$$\mu_{m+1} = \tau_{m+1}/\tau_m \quad (13)$$

$$\mathbf{b}_{m+1} = (I_2 - B_m)^{-1}(\mathbf{c}_{m+1} - \mu_{m+1}\mathbf{c}_m) \quad (14)$$

where

$$\tau_m = [q^o_{m-2}, \mathbf{q}^T_{m-2}J, t_{m-1}]\mathbf{g}_m \quad (15)$$

$$B_m = H^T_m U_m, \quad \text{and} \quad \mathbf{c}_m = H^T_m \mathbf{g}_m \quad (16)$$

*(b) Update the symmetric auxiliary matrix $U_m$:*

$$U_{m+2} = \begin{bmatrix} U_{m+1} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ U_{m+1} \end{bmatrix} - \begin{bmatrix} 0 \\ U_m \\ 0 \end{bmatrix} A_{m+1} + \mathbf{g}_{m+2}\mathbf{d}^T_{m+1} \quad (17)$$

where $A_{m+1}$ and $\mathbf{d}_{m+1}$ can also be derived by premultiplying both sides of (17) and taking appropriate partitions in (3) to yield

$$A_{m+1} = (I_2 - B_m)^{-1}(I_2 - B_{m+1}) \quad (18)$$

$$\mathbf{d}^T_{m+1} = \tau^{-1}_{m+2}\{\mathbf{h}^T_{m+1} - \mathbf{h}^T_m(I_2 - B_{m+1}) - \mathbf{s}^T_{m+1}U_{m+1} + \mathbf{r}^T_m A_{m+1}\} \quad (19)$$

where

$$\mathbf{r}^T_m = \mathbf{s}^T_m U_m - \mathbf{h}^T_m B_m \quad (20)$$

*(c) Update the linear phase prediction filter* $\Gamma_{2p+1}$

$$\Gamma_{2p+1} = \zeta_p \left\{ \begin{bmatrix} 0 \\ \Gamma_{2p-1} \\ 0 \end{bmatrix} - \alpha_p \mathbf{g}_{2p+1} + \begin{bmatrix} 0 \\ U_{2p-1} \\ 0 \end{bmatrix} \mathbf{f}_p \right\} \quad (21)$$

where $\zeta_p, \alpha_p$, and $\mathbf{f}_p$ are

$$\mathbf{f}_p = (I_2 - B_{2p-1})^{-1}\mathbf{e}_{2p-1} \quad (22)$$

$$\alpha_p = \tau^{-1}_{2p}(\epsilon_{2p-2} + \mathbf{r}^T_{2p-2}\mathbf{f}_p) \quad (23)$$

$$\zeta_p = (1 - \alpha_{p+1}g_{2p+1}(p+1) + U_{2p-1}(p)\mathbf{f}_p)^{-1} \quad (24)$$

with

$$\mathbf{e}_{2p} = H^T_{2p}\Gamma_{2p}, \quad \epsilon_{2p} = \mathbf{q}^T_{2p}\Gamma_{2p}, \quad \text{and} \quad \sigma_{p-1} = \zeta_p\sigma_{p-2} \quad (25)$$

Note that while $\mathbf{g}_m$ and $U_m$ need to run in every recursion ($m = 2p, 2p+1$), $\Gamma_{2p+1}$ only runs at alternate time steps. Also, since all three quantities are centrosymmetric, i.e. $J(\cdot) = (\cdot)$, only half of the elements need to be computed.

Compared with other existing fast algorithms, the proposed one possesses some attractive features. First, unlike other fast algorithms which require *both* the time and order updates in the recurrences, e.g. [2, 4, 5], the proposed one only involves the order update. More specifically, let $G_m(z), U_m(z)$, and $\Gamma_{2p+1}(z)$ denote the $Z$-transform for $\mathbf{g}_m, U_m$ and $\Gamma_{2p+1}$, respectively, then we have the block diagram as shown in Fig. 1 to illustrate the simpler and more regular update structure for the proposed fast algorithm, thus making the proposed one more suitable for VLSI hardware implementation.

Second, from the comparison of computational complexity for several existing fast algorithms as shown in Table 1, we can observe that the proposed one requires the least number of arithmetic operations. These computational savings can also be verified to be true for more general finite-impulse response (FIR) filters.

## 3. NEW SCHUR-LIKE PARALLELIZABLE THREE-TERM RECURRENCE

The computation of the weighting coefficients in (12), (17), and (21) involve some inner product operations, which are undesirable characteristics in a parallel processing environment. This difficulty can be overcome by introducing the Schur algorithm. The basic concept of the Schur algorithm is to replace the variables involving inner products by some simple recurrences. To accomplish this, we add a new index $n$ into the variables encountered in the previous section. More specifically, assume

$$H_m(n) = [\mathbf{x}_m(M+n-1), J\mathbf{x}_m(N+m-n)] \tag{26}$$

$$\mathbf{h}_m^T(n) = [x(M+n-1) \quad x(N-n+1)] \tag{27}$$

$$\mathbf{q}_m(n) = \sum_{k=M+n}^{N-1} \mathbf{x}_m(k)x(k-n) + J\sum_{k=M+n}^{N-1} \mathbf{x}_m(k+m-n)x(k+1) \tag{28}$$

$$B_m(n) = H_m^T(n)U_m, \mathbf{c}_m(n) = H_m^T(n)\mathbf{g}_m \tag{29}$$

$$\mathbf{r}_m(n) = \mathbf{q}_m^T(n)U_m \tag{30}$$

$$\mathbf{e}_m(n) = H_m^T(n)\Gamma_m \text{ and } \epsilon_m(n) = \mathbf{q}_m^T(n)U_m \tag{31}$$

If $n$(or $k$)$=m$, these variables will be the same as their corresponding counterparts discussed in Section 2. It can be shown that these new variables render the following recurrences:

$$\begin{aligned} B_m(n) &= B_{m-1}(n) + B_{m-1}(n-1) \\ &- B_{m-2}(n-1)A_{m-1} + \mathbf{c}_m(n)\mathbf{d}_{m-1}^T \end{aligned} \tag{32}$$

$$\begin{aligned} \mathbf{c}_m(n) &= \mathbf{c}_{m-1}(n) + \mathbf{c}_{m-1}(n-1) \\ &- \mathbf{c}_{m-2}(n-1)\mu_{m-1} + B_{m-2}(n-1)\mathbf{b}_{m-1} \end{aligned} \tag{33}$$

$$\tau_m = \beta_m(m-1) + \mathbf{h}_{m-1}^T(m-1)\mathbf{c}_m(m) \tag{34}$$

where $\beta_m(n) = \mathbf{q}_m^T(n)\mathbf{g}_m$ and satisfies

$$\begin{aligned} \beta_m(n) &= \eta_{m-1}(n) + \beta_{m-1}(n-1) \\ &- \gamma_{m-2}(n-1)\mu_{m-1} + \mathbf{w}_{m-2}(n-1)\mathbf{b}_{m-1} \end{aligned} \tag{35}$$

with

$$\eta_m(n) = \beta_{m+1}(n) - \mathbf{h}_{m-1}^T(n)\mathbf{c}_m(m+1) \tag{36}$$

$$\mathbf{w}_m(n) = \mathbf{r}_m(n) - \mathbf{h}_m^T(n+1)B_m(m+1) \tag{37}$$

$$\gamma_m(n) = \beta_m(n) - \mathbf{h}_m^T(n+1)\mathbf{c}_m(m+1) \tag{38}$$

Also,

$$\begin{aligned} \mathbf{d}_{m+1}^T &= \tau_{m+2}^{-1}[(\mathbf{h}_{m+1}^T(m+1) - \mathbf{h}_m^T(m))(I_2 - B_{m+1}(m+1)) \\ &- \mathbf{r}_{m+1}^T(m+1) + \mathbf{r}_m^T(m)A_{m+1}] \end{aligned} \tag{39}$$

$$\begin{aligned} \mathbf{r}_m(n) &= \mathbf{v}_{m-1}(n) + \mathbf{r}_{m-1}(n-1) \\ &- \mathbf{w}_{m-2}(n-1)A_{m-1} + \beta_m(n)\mathbf{d}_{m-1}^T \end{aligned} \tag{40}$$

where $\mathbf{v}_m(n)$ admits the following recurrence

$$\mathbf{v}_m(n) = \mathbf{r}_m(n) - \mathbf{h}_{m-1}^T(n)B_m(m+1) \tag{41}$$

$\epsilon_{2p+1}(n)$ and $\mathbf{e}_{2p+1}(n)$ have the following recurrence:

$$\begin{aligned} \epsilon_{2p+1}(n) &= \zeta_{p+1}\{\lambda_{2p-1}(n-1) - \alpha_{p+1}\beta_{2p+1}(n) \\ &+ \mathbf{w}_{2p-1}(n-1)\mathbf{f}_p\} \end{aligned} \tag{42}$$

$$\begin{aligned} \mathbf{e}_{2p+1}(n) &= \zeta_{p+1}\{\mathbf{e}_{2p-1}(n-1) - \alpha_{p+1}\mathbf{c}_{2p+1}(n) \\ &+ B_{2p-1}(n-1)\mathbf{f}_p\} \end{aligned} \tag{43}$$

where $\lambda_{2p+1}(n)$ satisfies

$$\lambda_{2p+1}(n) = \epsilon_{2p+1}(n) - \mathbf{h}_{m+1}^T(n+1)\mathbf{e}_{2p+1}(2p+2) \tag{44}$$

For these new variables, if we substitute $n$ by the same value as their subscript, we can get their corresponding counterparts considered in the previous section. Therefore, we can use the above simple recurrences in lieu of the inner product operations discussed before to get a highly parallelizable algorithm.

## 4. SIMULATION RESULTS

In this section, we provide some simulation results, which not only justify the validity of the proposed algorithm but they also highlight some potential applications. In both examples, the observation data $\{x(n)\}$ is of the type $x(n) = s(n)+w(n)$, where $\{s(n)\}$ is the desired signal and $\{w(n)\}$ is the additive zero mean white Gaussian noise with variance $\sigma^2$.

**Example 1:** Assume $s(n) = \cos(2\pi(0.1)n)+\cos(2\pi(0.2)n)$, $N_0 = 45$, SNR=20 dB. We can employ the Prony's method to estimate the imbedded signal frequencies with the proposed algorithm using $p = 2$ [1]. 100 Monte Carlo simulations have been carried out via the proposed algorithm. The estimated frequencies are 0.1063 and 0.1963, respectively, which are very close to the true ones.

**Example 2:** Suppose $s(n) = 1.352s(n-1) - 1.338s(n-2) + 0.662s(n-3) - 0.24s(n-4)$, $N_0 = 200$, $\sigma^2 = 1$. The average prediction errors from the classical one-sided linear prediction and the two-sided prediction with the proposed algorithm are shown in Fig. 2, in which we can observe that the two-sided linear prediction method yields lower prediction errors. Indeed, from various simulations performed, the same fact always remains although we still have not been able to prove this. This implies that this new schemes might be an attractive alternative to reducing the prediction errors.

## 5. CONCLUSION

This paper develops a new fast algorithm for the design of linear least squares prediction filter with a linear phase characteristic. This new algorithm not only has lower computational complexity than other existing ones, but it lso possesses more regular update structures, which make it more amenable for hardware implementation. For parallelizable consideration, a corresponding Schur-like algorithm is also addressed. Some simulations are provided to justify the validity of these new algorithms.

## 6. REFERENCES


[1] S.L. Marple, Jr., *Digital Spectral Analysis with Applications,* Englewood Cliffs, NJ:Prentice-Hall, 1987

[2] S. L. Marple, "Fast algorithms for linear prediction and system identification filters with linear phase," *IEEE Trans. Acoust., Speech, Signal Processing,*Vol. ASSP-30, pp.942-953, Dec.1982.

[3] S. L. Marple, "A new autoregressive spectrum analysis algoorithm," *IEEE Trans. Acoust., Speech, Signal Processing,* Vol. ASSP-28, pp.441-454, Aug. 1980.

[4] N. Kalouptsidis and G. D. Koyas, "Efficient block LS design of FIR filters with linear phase," *IEEE Trans. Acoust., Speech, Sighal Processing,* Vol. ASSP-33, PP.1435-1444, Dec.1985.

[5] K. Berberidis and S. Theodoridis, "New Levinson, Schur, and lattice type algorithms for linear phase filtering," *IEEE Trans. Acoust., Speech, Signal Processing,* Vol. ASSP-38, pp.1879-1892, Nov. 1990.

[6] A. C. Lee, "A new autoregressive method for high-performance spectrum analysis," *J. Acoust. Soc. Amer.*, vol. 86, pp. 150-157, 1989.

[7] J.-J. Hsue and A. E. Yagle, "Fast algorithms for close-to-Toeplitz-plus-Hankel systems and two-sided linear prediction," *IEEE Trans. Signal Processing,* Vol. ASSP-41, pp.2349-2361, July. 1993.

[8] J.-J. Hsue and A. E. Yagle, "Blind deconvolution of symmetric noncausal impulse responses using two-sided linear prediction," *IEEE Trans. Signal Processing,* Vol. ASSP-42, pp.1509-1518, June. 1994.

[9] P. Delsarte and Y.V. Genin, "On the Splitting of classical algorithms in linear Prediction theory," *IEEE Trans. Acoust., Speech, Signal Processing,* Vol. ASSP-35, pp.645-643, May 1987.


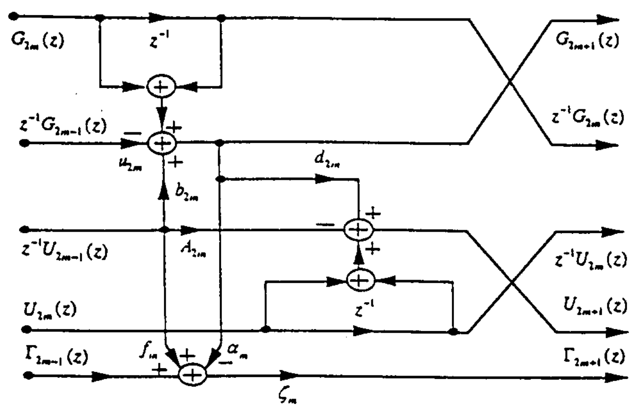

Figure 1: The block diagram of the recurrences for linear-phase prediction filters

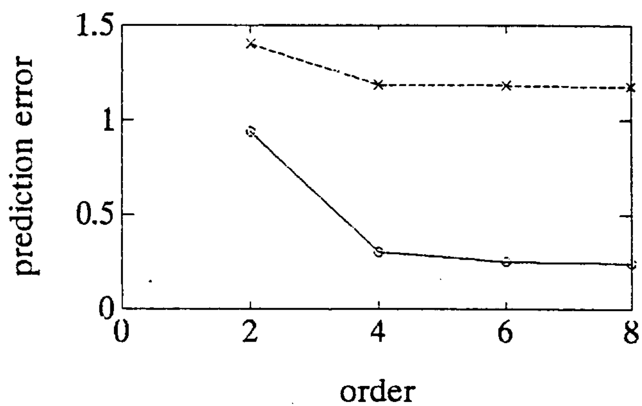


Figure 2: The average prediction errors by using the classical one-sided linear prediction (×) and the proposed one (∘) for different prediction orders.

| Algorithms | Add/Sub | Mul/Div |
|---|---|---|
| [2]& [7] | $1.5N_0 + 19.5m$ | $1.5N_0 + 20m$ |
| [4] | $N_0 + 18.5m$ | $N_0 + 18.5m$ |
| [5] | $N_0 + 17.5m$ | $N_0 + 13.5m$ |
| Proposed | $N_0 + 15m$ | $N_0 + 11.75m$ |

Table 1: Comparison of the computational complexity per update for several fast algorithms, where $m$ is the order of the prediction filter and $N_0 = (N - M + 1)$ is the total number of the data points.