

Interference Cancellation in Coherent CDMA Systems Using Parallel Iterative Algorithms

Kamal Shahtalebi

Department of Information Technology
University of Isfahan, Isfahan
Iran, Postal Code: 81746-73441
Email: shahtalebi@eng.ui.ac.ir

Gholamreza Bakhshi

Communication Research Lab.
Department of Electrical Engineering
Yazd University, Yazd, Iran
Email: bakhshi@stu.yazduni.ac.ir

Hamidreza Saligheh Rad

School of Engineering and Applied
Sciences, Harvard University
Cambridge MA, 02138, USA
Email: hamid@seas.harvard.edu

Abstract—Least mean square-partial parallel interference cancellation (LMS-PPIC) is a partial interference cancellation using adaptive multistage structure in which the normalized least mean square (NLMS) adaptive algorithm is engaged to obtain the cancellation weights. The performance of the NLMS algorithm is mostly dependent to its step-size. A fixed and non-optimized step-size causes the propagation of error from one stage to the next one. When all user channels are balanced, the unit magnitude is the principal property of the cancellation weight elements. Based on this fact and using a set of NLMS algorithms with different step-sizes, the parallel LMS-PPIC (PLMS-PPIC) method is proposed. In each iteration of the algorithm, the parameter estimate of the NLMS algorithm is chosen to match the elements' magnitudes of the cancellation weight estimate with unity. Simulation results are given to compare the performance of our method with the LMS-PPIC algorithm in three cases: balanced channel, unbalanced channel and time varying channel.

I. INTRODUCTION

Multisuser detectors for code division multiple access (CDMA) receivers are effective techniques to eliminate the multiple access interference (MAI). In CDMA systems, all users receive the whole transmitted signals concurrently that are recognized by their specific pseudo noise (PN) sequences. In such a system, there exists a limit for the number of users that are able to simultaneously communicate. This limitation is because of the MAI generated by other users (see e.g. [1], [2]). High quality detectors improve the capacity of these systems [1] – [6]. However their computational complexities grow exponentially with increasing the number of users and the length of the transmitted sequence [7].

Multiple stage subtractive interference cancellation is a suboptimal solution with reduced computational complexity. In this method and before making data decisions, the estimated interferences from other users are removed from the specific user's received signal. The cancellation can be carried out either in a serial way (successively) (see e.g. [8], [9]) or in a parallel manner (see e.g. [2], [3], [10]). The parallel interference cancellation (PIC) is a low computational complex method that causes less decision delay compared to the successive detection and is much simpler in implementation.

Usually at the first stage of interference cancellation in a multiple stage system, the interfering data for each user which is made by other users is unknown. PIC is implemented

to estimate this data, stage by stage. In fact when MAI is estimated for each user, the bits decisions at the $(s-1)^{\text{th}}$ stage of cancellation are used for bits detection at the s^{th} stage. Apparently, the more accurate the estimates are, the better performance of the detector is. However, in the conventional multistage PIC [3], a wrong decision in one stage can increase the interference. Based on minimizing the mean square error between the received signal and its estimate from the previous stage, G. Xue *et al.* proposed the least mean square-partial parallel interference cancellation (LMS-PPIC) method [10], [11]. In LMS-PPIC, a weighted value of MAI of other users is subtracted before making the decision of a specific user. The least mean square (LMS) optimization and the normalized least mean square (NLMS) algorithm [12] shape the structure of the LMS-PPIC method of the weight estimation of each cancellation stage. However, the performance of the NLMS algorithm is mostly dependent on its step-size. Although a large step-size results in a faster convergence rate, but it causes a large misadjustment. On the other hand, with a very small step-size, the algorithm almost keeps its initial values and can not estimate the true cancellation weights. In the LMS-PPIC method, both of these cases cause propagation of error from one stage to another. In LMS-PPIC, the m^{th} element of the weight vector in each stage is the true transmitted binary value of the m^{th} user divided by its hard estimate value from the previous stage. Hence the magnitudes of all weight elements in all stages are equal to unity. This is a valuable information that can be used to improve the performance of the LMS-PPIC method. In this paper, we propose parallel LMS-PPIC (PLMS-PPIC) method by using a set of NLMS algorithms with different step-sizes. The step-size of each algorithm is chosen from a sharp range [13] that guarantees stable operation. While in this paper we assume coherent transmission, the non-coherent scenario is investigated in [5].

The rest of this paper is organized as follows: In section II, the LMS-PPIC [10], an important example of multistage parallel interference cancellation methods, is reviewed. In section III, the PLMS-PPIC method is explained. In section IV some simulation examples are given to compare the results of PLMS-PPIC with those of LMS-PPIC. Finally, the paper is concluded in section V.

II. MULTISTAGE PARALLEL INTERFERENCE CANCELLATION: LMS-PPIC METHOD

We assume M users synchronously send their symbols $\alpha_1, \alpha_2, \dots, \alpha_M$ via a base-band CDMA transmission system where $\alpha_m \in \{-1, 1\}$. The m^{th} user has its own code $p_m(\cdot)$ of length N , where $p_m(n) \in \{-1, 1\}$, for all n . It means that for each symbol N bits are transmitted by each user and the processing gain is equal to N . At the receiver we assume that perfect power control scheme is applied. Without loss of generality, we also assume that the power gains of all channels are equal to unity and users' channels do not change during each symbol transmission (it can change from one symbol transmission to the next one) and the channel phase ϕ_m of m^{th} user is known for all $m = 1, 2, \dots, M$ (see [5] for non-coherent transmission). We define

$$c_m(n) = e^{j\phi_m} p_m(n). \quad (1)$$

According to the above assumptions, the received signal is

$$r(n) = \sum_{m=1}^M \alpha_m c_m(n) + v(n), \quad n = 1, 2, \dots, N, \quad (2)$$

where $v(n)$ is additive white Gaussian noise with zero mean and variance σ^2 . In order to make a new variable set $\alpha_1^s, \alpha_2^s, \dots, \alpha_M^s$ for the current stage s , multistage parallel interference cancellation method uses $\alpha_1^{s-1}, \alpha_2^{s-1}, \dots, \alpha_M^{s-1}$ (the bit estimate outputs of the previous stage $s-1$) to estimate the related MAI of each user, to subtract it from the received signal $r(n)$ and to make a new decision on each user variable individually. The output of the last stage is considered as the final estimate of the transmitted bits. In the following we explain the structure of the LMS-PIC method. Assume $\alpha_m^{(s-1)} \in \{-1, 1\}$ is a given estimate of α_m from stage $s-1$. Let us define

$$w_m^s = \frac{\alpha_m}{\alpha_m^{(s-1)}}. \quad (3)$$

From (2) and (3) we have

$$r(n) = \sum_{m=1}^M w_m^s \alpha_m^{(s-1)} c_m(n) + v(n). \quad (4)$$

Define

$$W^s = [w_1^s, w_2^s, \dots, w_M^s]^T, \quad (5a)$$

$$X^s(n) = [\alpha_1^{(s-1)} c_1(n), \alpha_2^{(s-1)} c_2(n), \dots, \alpha_M^{(s-1)} c_M(n)]^T \quad (5b)$$

where T stands for transposition. From equations (4), (5a) and (5b), we have

$$r(n) = W^{sT} X^s(n) + v(n). \quad (6)$$

Given the observations $\{r(n), X^s(n)\}_{n=1}^N$, an adaptive algorithm can be used to compute

$$W^s(N) = [w_1^s(N), w_2^s(N), \dots, w_M^s(N)]^T, \quad (7)$$

which is an estimate of W^s after N iterations. Then α_m^s , the estimate of α_m at stage s , is given by

$$\alpha_m^s = \text{sign} \left(\text{Re} \left\{ \sum_{n=1}^N q_m^s(n) c_m^*(n) \right\} \right), \quad (8)$$

where $(\cdot)^*$ stands for complex conjugation and

$$q_m^s(n) = r(n) - \sum_{m'=1, m' \neq m}^M w_{m'}^s(N) \alpha_{m'}^{(s-1)} c_{m'}(n). \quad (9)$$

The inputs of the first stage $\{\alpha_m^0\}_{m=1}^M$ (needed for computing $X^1(n)$) is given by the conventional bit detection

$$\alpha_m^0 = \text{sign} \left(\text{Re} \left\{ \sum_{n=1}^N r(n) c_m^*(n) \right\} \right), \quad m = 1, 2, \dots, M. \quad (10)$$

Given the available information $\{r(n), X^s(n)\}_{n=1}^N$ and using equation (6), there are a variety of choices for parameter estimation. In LMS-PPIC, the NLMS algorithm is used to compute $W^s(N)$. Table I shows the full structure of the LMS-PPIC method.

To improve the performance of the LMS-PPIC method, in the next section we propose a modified version of it. In our method, a set of individual NLMS algorithms with different step-sizes is used.

III. MULTISTAGE PARALLEL INTERFERENCE CANCELLATION: PLMS-PPIC METHOD

The NLMS (with fixed step-size) converges only in the mean sense. In the literature, $\mu \in (0, 2)$ guarantees the mean convergence of the NLMS algorithm [12], [14]. Based on Cramér-Rao bound, a sharper range was given in [13] as follows

$$\mu \in \Psi = \left(0, 1 - \sqrt{\frac{M-1}{M}} \right], \quad (11)$$

where M is the length of the parameter under estimate. Here M is the number of users or equivalently the system load. As equation (11) shows, the range of the step-size is a decreasing function of the system load. It means that as the number of users increases, the step-size must be decreased and vice versa. In the proposed PLMS-PPIC method, Ψ has a critical role.

From (3), we have

$$|w_m^s| = 1 \quad m = 1, 2, \dots, M, \quad (12)$$

which is equivalent to

$$\sum_{m=1}^M ||w_m^s| - 1| = 0. \quad (13)$$

To improve the performance of the NLMS algorithm, at time iteration n , we can determine the step size $\mu(n)$ from Ψ , in such a way that $\sum_{m=1}^M ||w_m^s(n)| - 1|$ is minimized, i.e.

$$\mu(n) = \arg \min_{\mu \in \Psi} \left\{ \sum_{m=1}^M ||w_m^s(n)| - 1| \right\}, \quad (14)$$

where $w_m^s(n)$, the m^{th} element of $W^s(n)$, is given by (see Table I)

$$w_m^s(n) = w_m^s(n-1) + \mu \frac{\alpha^{(s-1)} c_m^*(n)}{\|X^s(n)\|^2} e(n). \quad (15)$$

The complexity to determine $\mu(n)$ from (14) is high, especially for large values of M . Instead we propose the following method.

We divide Ψ into L subintervals and consider L individual step-sizes $\Theta = \{\mu_1, \mu_2, \dots, \mu_L\}$, where $\mu_1 = \frac{1-\sqrt{\frac{M-1}{M}}}{L}$, $\mu_2 = 2\mu_1, \dots$, and $\mu_L = L\mu_1$. In each stage, L individual NLMS algorithms are executed (μ_l is the step-size of the l^{th} algorithm). In stage s and at iteration n , if $W_k^s(n) = [w_{1,k}^s, \dots, w_{M,k}^s]^T$, the parameter estimate of the k^{th} algorithm minimized our criteria, i.e.

$$W_k^s(n) = \arg \min_{W_l^s(n) \in I_{W^s}} \left\{ \sum_{m=1}^M |w_{m,l}^s(n)| - 1 \right\}, \quad (16)$$

where $W_l^s(n) = W^s(n-1) + \mu_l \frac{X^s(n)}{\|X^s(n)\|^2} e(n)$, $l = 1, 2, \dots, k, \dots, L-1, L$ and $I_{W^s} = \{W_1^s(n), \dots, W_L^s(n)\}$, then it is considered as the parameter estimate at time iteration n , i.e. $W^s(n) = W_k^s(n)$ and all other algorithms replace their weight estimates by $W_k^s(n)$. Table II shows the details of the PLMS-PPIC method. As Table II shows, in stage s and at time iteration N where $W^s(N)$ is computed, the PLMS-PPIC method computes α_m^s from equation (8). This is similar to the LMS-PPIC method. Here the PLMS-PPIC and the LMS-PPIC methods are compared with each other.

- Computing $\mu_l Z(n) = \mu_l \frac{X^s(n)}{\|X^s(n)\|^2}$, L times more than LMS-PPIC, and computing $\sum_{m=1}^M |w_{m,l}^s(n)| - 1$ in each iteration of each stage of PLMS-PPIC, are the differences between it and the LMS-PPIC method.
- Because the step-sizes of all individual NLMS algorithms of the proposed method are given from a stable operation range, all of them converge (fast or slowly). Hence the PLMS-PPIC is a stable method.
- As we expected and our simulations show, choosing the step-size as a decreasing function of system loads (based on relation (11)) improves the performance of both NLMS algorithm in LMS-PPIC and parallel NLMS algorithms in PLMS-PPIC methods in such a way that there is no need for the third stage, i.e. both the LMS-PPIC and PLMS-PPIC methods get the optimum weights in the second stage. However only when the channel is time varying, the third stage is needed (e.g. example 3).
- Increasing the number of parallel NLMS algorithms L in PLMS-PPIC method increases the complexity, while it improves the performance as well.
- As our simulations show, the LMS-PPIC method is more sensitive to the Channel loss, near-far problem or unbalanced channel gain compared to the PLMS-PPIC.

In the following section, some examples are given to illustrate the effectiveness of our proposed method.

IV. SIMULATIONS

In this section we have considered some simulation examples. Examples 1-3 compare the conventional, the LMS-PPIC and the PLMS-PPIC methods in three cases: balanced channels, unbalanced channels and time varying channels. Example 1 is given to compare LMS-PPIC and PLMS-PPIC in the case of balanced channels.

Example 1: Balanced channels: Consider the system model (4) in which M users, each having their own codes of length N , send their own bits synchronously to the receiver and through their channels. The signal to noise ratio (SNR) is 0dB. In this example we assume that there is no power-unbalanced or channel loss. The step-size of the NLMS algorithm in LMS-PPIC method is $\mu = 0.1(1 - \sqrt{\frac{M-1}{M}})$ and the set of step-sizes of the parallel NLMS algorithms in PLMS-PPIC method is $\Theta = \{0.01, 0.05, 0.1, 0.2, \dots, 1\}(1 - \sqrt{\frac{M-1}{M}})$, i.e. $\mu_1 = 0.01(1 - \sqrt{\frac{M-1}{M}}), \dots, \mu_4 = 0.2(1 - \sqrt{\frac{M-1}{M}}), \dots, \mu_{12} = (1 - \sqrt{\frac{M-1}{M}})$. Figure 1 shows the average bit error rate (BER) over all users versus M , using two stages when $N = 64$ and $N = 256$. As it is shown, while there is no remarkable performance difference between all three methods for $N = 64$, the PLMS-PPIC outperforms the conventional and the LMS-PPIC methods for $N = 256$. We also run the program for three stages and noticed that there is no remarkable difference between results in two and three stages scenarios.

Although LMS-PPIC and PLMS-PPIC are structured based on the assumption of no near-far problem, these methods (especially the second one) have remarkable performance in the cases of unbalanced and/or time varying channels. These facts are shown in the two upcoming examples.

Example 2: Unbalanced channels: Consider example 1 with power unbalance and/or channel loss in transmission system, i.e. the true model at stage s is

$$r(n) = \sum_{m=1}^M \beta_m w_m^s \alpha_m^{(s-1)} c_m(n) + v(n), \quad (17)$$

where $0 < \beta_m \leq 1$ for all $1 \leq m \leq M$. Both the LMS-PPIC and the PLMS-PPIC methods assume the model (4), and their estimations are based on observations $\{r(n), X^s(n)\}$, instead of $\{r(n), \mathbf{G}X^s(n)\}$, where the channel gain matrix is $\mathbf{G} = \text{diag}(\beta_1, \beta_2, \dots, \beta_m)$. In this case we repeat example 1. We randomly get each diagonal element of G from $(0, 0.3]$. Results are given in Figure 2. As it is shown, in all cases the PLMS-PPIC method outperforms both the conventional and the LMS-PPIC methods.

Example 3: Time varying channels: Consider example 1 with time varying Rayleigh fading channels. In this case we assume the maximum Doppler shift of 40HZ, the three-tap frequency-selective channel with delay vector of $\{2 \times 10^{-6}, 2.5 \times 10^{-6}, 3 \times 10^{-6}\}$ sec and gain vector of $\{-5, -3, -10\}$ dB. Results are given in Figure 3. As it is seen while the PLMS-PPIC outperforms the conventional and the LMS-PPIC methods when the number of users is less than

30, all three methods have the same performance when the number of users is greater than 30.

V. CONCLUSION

In this paper, parallel interference cancellation using adaptive multistage structure and employing a set of NLMS algorithms with different step-sizes is proposed. According to the proposed method, in each iteration the parameter estimate is chosen in a way that its corresponding algorithm has the best compatibility with the true parameter. Because the step-sizes of all algorithms are chosen from a stable range, the total system is therefore stable. Simulation results show that the new method has a remarkable performance for different scenarios including Rayleigh fading channels even if the channel is unbalanced.

REFERENCES

- [1] S. Verdú, *Multuser Detection*, Cambridge University Press, 1998.
- [2] D. Divsalar, M. K. Simon and D. Raphaeli, "Improved parallel interference cancellation for CDMA," *IEEE Trans. Commun.*, vol. COM-46, no. 2, pp. 258-268, Feb. 1998.
- [3] M. K. Varanasi and B. Aazhang, "Multistage detection in asynchronous code division multiple-access communication," *IEEE trans. Commun.*, vol. COM-38, no 4, pp.509-519, April 1990.
- [4] S. Verdú, "Minimum probability of error for asynchronous Gaussian multiple-access channels," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 85-96, Jan. 1986.
- [5] K. Shahtalebi, G. R. Bakhshi and H. S. Rad, "Interference Cancellation in Noncoherent CDMA Systems Using Parallel Iterative Algorithms", *IEEE Wireless Communications and Networking Conference, WCNC'08*, Las Vegas, USA, April 2008.
- [6] S. Verdú, "Optimum multiuser asymptotic efficiency," *IEEE Transactions on Communications*, vol. COM-34, pp. 890-897, Sept. 1986.
- [7] S. Verdú and H. V. Poor, "Abstract dynamic programming models under commutativity conditions," *SIAM J. Control and Optimization*, vol. 25, pp. 990-1006, July 1987.
- [8] A.J. Viterbi, "Very low rate convolutional codes for maximum theoretical performance of spread-spectrum multiple-access channels," *IEEE Journal on Selected Areas in Communications*, vol. 8, pp. 641-649, May. 1989.
- [9] P. Patel and J. Holtzman, "Analysis of a simple successive interference cancellation scheme in DS/CDMA system," *IEEE Journal on Selected Areas in Communications*, vol. 12, pp. 509-519, June. 1994.
- [10] G. Xue, J. Weng, Tho Le-Ngoc, and S. Tahar, "Adaptive multistage parallel interference cancellation for CDMA," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 10, pp. 1815-1827, Oct. 1999.
- [11] G. Xue, J. Weng, Tho Le-Ngoc, and S. Tahar, "Another Approach for partial parallel interference cancellation," *Wireless Personal Communications Journal*, vol. 42, no. 4, pp. 587-636, Sep. 2007.
- [12] S. Haykin, *Adaptive Filter Theory, 3rd ed.* Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [13] K. Shahtalebi and S. Gazor "On the adaptive linear estimators, using biased Cramer Rao bound" *Elsevier Signal Processing*, vol. 87, pp. 1288-1300, June 2007.
- [14] D. T. M. Slock, "On the convergence behavior of the LMS and the Normalized LMS algorithms," *IEEE Trans. on Signal Processing*, vol. 41, No. 9, pp. 2811-2825, Sep. 1993.

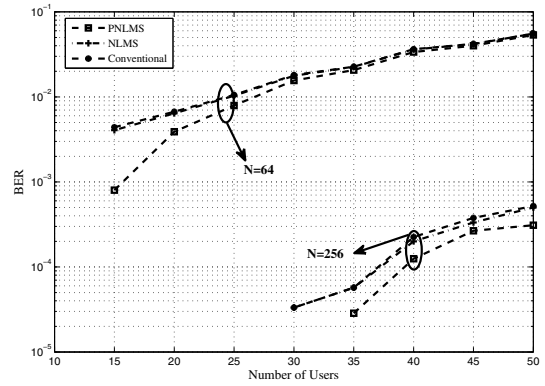


Fig. 1. The BER of the conventional, the LMS-PPIC, and the PLMS-PPIC methods versus the system load in balanced channel, using two stages for $N = 64$ and $N = 256$.

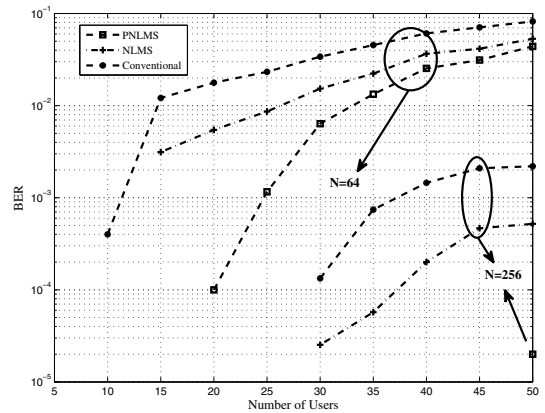


Fig. 2. The BER of the conventional, the LMS-PPIC, and the PLMS-PPIC methods versus the system load in unbalanced channel, using two stages for $N = 64$ and $N = 256$.

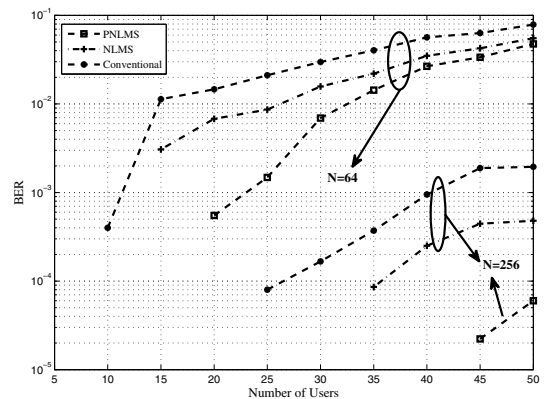


Fig. 3. The BER of the conventional, the LMS-PPIC, and the PLMS-PPIC methods versus the system load in time varying Rayleigh fading channel using two stages for $N = 64$ and $N = 256$.

TABLE I
THE PROCEDURE OF THE LMS-PPIC METHOD

Initial Values	for $m = 1, 2, \dots, M$	$\alpha_m^0 = \text{sign} \left\{ \text{real} \left\{ \sum_{n=1}^N r(n)c_m^*(n) \right\} \right\}$
for $s = 1, 2, \dots, S$	$W^s(0) = [w_1^s(0), \dots, w_M^s(0)]^T = [0, \dots, 0]^T$	
	NLMS algorithm	for $n = 1, 2, \dots, N$ $X^s(n) = [\alpha_1^{(s-1)}c_1(n), \alpha_2^{(s-1)}c_2(n), \dots, \alpha_M^{(s-1)}c_M(n)]^T$ $e(n) = r(n) - W^{sT}(n-1)X^s(n)$ $Z(n) = \frac{X^{s*}(n)}{\ X^s(n)\ ^2}e(n)$ $W^s(n) = W^s(n-1) + \mu Z(n)$
	for $m = 1, 2, \dots, M$	$q_m^s(n) = r(n) - \sum_{m'=1, m' \neq m}^M w_{m'}^s(N)\alpha_{m'}^{(s-1)}c_{m'}(n)$ $\alpha_m^s = \text{sign} \left\{ \text{real} \left\{ \sum_{n=1}^N q_m^s(n)c_m^*(n) \right\} \right\}$

TABLE II
THE PROCEDURE OF THE PLMS-PPIC METHOD

Initial Values	for $m = 1, 2, \dots, M$	$\alpha_m^0 = \text{sign} \left\{ \text{real} \left\{ \sum_{n=1}^N r(n)c_m^*(n) \right\} \right\}$
for $s = 1, 2, \dots, S$	$W^s(0) = [w_1^s(0), \dots, w_M^s(0)]^T = [0, \dots, 0]^T$	
	PNLMS algorithm	for $n = 1, 2, \dots, N$ $X^s(n) = [\alpha_1^{(s-1)}c_1(n), \alpha_2^{(s-1)}c_2(n), \dots, \alpha_M^{(s-1)}c_M(n)]^T$ $e(n) = r(n) - W^{sT}(n-1)X^s(n)$ $Z(n) = \frac{X^{s*}(n)}{\ X^s(n)\ ^2}e(n)$ $\min = \infty, l = 1$
		for $k = 1, 2, \dots, L$
	$W^s(n) = W_l^s(n)$	
for $m = 1, 2, \dots, M$	$q_m^s(n) = r(n) - \sum_{m'=1, m' \neq m}^M w_{m'}^s(N)\alpha_{m'}^{(s-1)}c_{m'}(n)$ $\alpha_m^s = \text{sign} \left\{ \text{real} \left\{ \sum_{n=1}^N q_m^s(n)c_m^*(n) \right\} \right\}$	