A Fast Eigen-based Signal Combining Algorithm by Using CORDIC

Leiou Wang
Institute of Acoustics,
Chinese Academy of Science
Beijing, 100190, China
Email: wangleiou@mail.ioa.ac.cn

Donghui Wang
Institute of Acoustics,
Chinese Academy of Science
Beijing, 100190, China
Email: wangdh@mail.ioa.ac.cn

Chengpeng Hao
Institute of Acoustics,
Chinese Academy of Science
Beijing, 100190, China
Email: haochengp@mail.ioa.ac.cn

***Abstract*—For reliable reception of weak signals, eigen-based signal combining algorithms are very effective. However, the algorithms involve a heavy computational burden. In this paper, a fast eigen-based signal combining algorithm is proposed by using the coordinate rotation digital computer (CORDIC) method. CORDIC can use addition and bitshift operations to replace the multiplications in the eigen-based signal combining algorithms. Simulation results indicate that the proposed algorithm can reduce the computational cost while it provides a good combining performance.**

I. Introduction

The main task of signal combining is to find a set of combining weights, so that the combined signal can obtain a good signal-to-noise ratio (SNR). SUMPLE, one of the traditional signal combining algorithm, is unstable with a weak SNR [1], [2]. Eigen-based algorithms that achieve certain objectives, on the other hand, are very effective in signal combining. In [3], SNR is used as the objective function (SNR EIGEN). It needs to estimate not only the signal but also the noise correlation matrix, which is difficult to directly estimate from the received signals [4], [5]. If the received noises are the additive white Gaussian noise with the same variance (uniform noise power), maximizing the combined output power (COP EIGEN) will be equivalent to maximizing the combined output SNR [6]. Although COP EIGEN can eliminate the estimation of the noise correlation matrix, the combining weights of COP EIGEN are biased when noise variances are different (non-uniform noise power) [6], [7]. In order to overcome the drawback, an autocorrelation coefficient is used as the objective function (AC EIGEN) [7], which has a better performance under a non-uniform noise power condition.

The heavy computational burden is the major problem for all above eigen-based algorithms [2], [7]. Their calculation includes both forming signal or noise correlation matrices and solving an eigenvector problem [7]. A power method (PM) and a proposed matrix-free method (PMFM) are presented in [8], but PM only reduces the computational cost for solving the eigenvector problem [8]. As far as PMFM is concerned, its combining weights are also biased with a non-uniform noise power [7]. As a result, for hardware circuits and systems, the eigen-based algorithms require huge computational cost as the number of sensors N and the length of sample L increase. To alleviate the computational burden, a fast eigen-based algorithm that employs a signum polarization (SP) model and a Chebyshev polynomials (CP) is presented in [9]. However, extra multiplications are incurred by using CP. Therefore, this paper proposes a fast eigen-based signal combining algorithm that combines the SP model with a coordinate rotation digital computer (CORDIC) method to further reduce the extra multiplications.

The remainder of this paper is organized as follows. Section II introduces the related work while Section III proposes the fast algorithm. Section IV provides numerical examples and comparisons. Finally, some concluding remarks are given in Section V.

II. Related work

A. Traditional Eigen-based Signal Combining Algorithms

The received signals of a sensor array can be expressed as [7]

$$x_i\left(k\right) = \alpha_i s\left(k\right) + n_i\left(k\right) \quad i = 1, 2, \cdots, N, \tag{1}$$

where $s\left(k\right)$ is the source signal, k is the sample index, α_i, $n_i\left(k\right)$, and $x_i\left(k\right)$ are the gain, the white Gaussian noise, and the received signal of the ith sensor, respectively.

The combined signal is given by [7]

$$x_c\left(k\right) = \boldsymbol{w}^H \boldsymbol{x}\left(k\right) = \boldsymbol{w}^H \boldsymbol{s}\left(k\right) + \boldsymbol{w}^H \boldsymbol{n}\left(k\right) = s_c\left(k\right) + n_c\left(k\right), \tag{2}$$

$$\boldsymbol{w} = \left[w_1, w_2, \cdots, w_N\right]^T, \tag{3}$$

$$\boldsymbol{\alpha} = \left[\alpha_1, \alpha_2, \cdots, \alpha_N\right]^T, \tag{4}$$

$$\boldsymbol{x}\left(k\right) = \left[x_1\left(k\right), x_2\left(k\right), \cdots x_N\left(k\right)\right]^T, \tag{5}$$

$$\boldsymbol{s}\left(k\right) = \left[\alpha_1 s\left(k\right), \alpha_2 s\left(k\right), \cdots, \alpha_N s\left(k\right)\right]^T, \tag{6}$$

$$\boldsymbol{n}\left(k\right) = \left[n_1\left(k\right), n_2\left(k\right), \cdots n_N\left(k\right)\right]^T, \tag{7}$$

where $\boldsymbol{w}$ is the combining weights, $\boldsymbol{\alpha}$ is the receiving-sensor gain vector, $s_c\left(k\right)$, $n_c\left(k\right)$, and $x_c\left(k\right)$ represent the combined source signal, the combined noise, and the combined signal, the superscript T and H denote the transpose operator and the transpose conjugate operator, respectively.

ISBN 978-90-827970-1-5

According to the definition, the combined signal's SNR is [3]

$$\gamma_{x_c} = \frac{E\left[\left|\boldsymbol{w}^H \boldsymbol{s}(k)\right|^2\right]}{E\left[\left|\boldsymbol{w}^H \boldsymbol{n}(k)\right|^2\right]} = \frac{\boldsymbol{w}^H \boldsymbol{R_s}(0)\,\boldsymbol{w}}{\boldsymbol{w}^H \boldsymbol{R_n}(0)\,\boldsymbol{w}}, \tag{8}$$

$$\boldsymbol{R_x}(0) = \boldsymbol{R_s}(0) + \boldsymbol{R_n}(0), \tag{9}$$

$$\gamma_{x_c} = \frac{\boldsymbol{w}^H \boldsymbol{R_x}(0)\,\boldsymbol{w}}{\boldsymbol{w}^H \boldsymbol{R_n}(0)\,\boldsymbol{w}} - 1, \tag{10}$$

where $E[\cdot]$ is an expected value, $\boldsymbol{R_s}(0)$, $\boldsymbol{R_n}(0)$, and $\boldsymbol{R_x}(0)$ are the source signal correlation matrix, the noise correlation matrix, and the signal correlation matrix, respectively. Taking the partial derivative of (10) with respect to $\boldsymbol{w}$ yields [3]

$$\boldsymbol{R_n^{-1}}(0)\,\boldsymbol{R_x}(0)\,\boldsymbol{w} - \frac{\boldsymbol{w}^H \boldsymbol{R_x}(0)\,\boldsymbol{w}}{\boldsymbol{w}^H \boldsymbol{R_n}(0)\,\boldsymbol{w}}\boldsymbol{w} = 0, \tag{11}$$

$$\boldsymbol{R_x}(\tau) = \begin{bmatrix} R_{x_1x_1}(\tau) & R_{x_1x_2}(\tau) & \cdots & R_{x_1x_N}(\tau) \\ R_{x_2x_1}(\tau) & R_{x_2x_2}(\tau) & \cdots & R_{x_2x_N}(\tau) \\ \vdots & \vdots & \ddots & \vdots \\ R_{x_Nx_1}(\tau) & R_{x_Nx_2}(\tau) & \cdots & R_{x_Nx_N}(\tau) \end{bmatrix}, \tag{12}$$

$$R_{x_ix_j}(\tau) = \frac{1}{L}\sum_{k=1}^{L} x_i(k)x_j(k+\tau) \quad i,j = 1,2,\cdots N, \tag{13}$$

where $\boldsymbol{R_n^{-1}}(0)$ is the inverse matrix of $\boldsymbol{R_n}(0)$, $R_{x_ix_j}(\tau)$ is a cross-correlation function, and τ is a delay. The combining weights of SNR EIGEN are proved to be the eigenvector corresponding to the largest eigenvalue of the $\boldsymbol{R_n^{-1}}(0)\,\boldsymbol{R_x}(0)$ [3]. This eigenvector is regarded as the dominant eigenvector.

Since the $\boldsymbol{R_n}(0)$ can not be easily estimated from received signals, the combining weights are commonly calculated by $\boldsymbol{R_x}(0)$ (COP EIGEN) [6] or $\boldsymbol{R_x^{-1}}(0)\,\boldsymbol{R_x}(\tau)$ (AC EIGEN) [7].

B. Proposed Matrix-Free Method, PMFM

PMFM bypasses the forming of the signal correlation matrix [8]. Starting with an initial vector $\boldsymbol{w}^{(0)}$, the subsequent steps follow [8]

$$\boldsymbol{u}^{(m)}(k) = \sum_{i=1}^{N} w_i^{(m-1)} \boldsymbol{x_i}(k) \quad k = 1,2,\cdots,L, \tag{14}$$

where m is the iterative number, $\boldsymbol{u}^{(m)}(k)$ is the weighted sum of the received signals. The inner product of $\boldsymbol{u}^{(m)}(k)$ with each individual received signal is [8]

$$y_i^{(m)} = E\left[\boldsymbol{x_i}(k)\,\boldsymbol{u}^{(m)}(k)\right] \quad k = 1,2,\cdots,L, \tag{15}$$

$$\boldsymbol{y}^{(m)} = \left[y_1^{(m)}, y_2^{(m)}, \cdots, y_N^{(m)}\right]^T, \tag{16}$$

$$\boldsymbol{w}^{(m)} = \frac{\boldsymbol{y}^{(m)}}{\left\|\boldsymbol{y}^{(m)}\right\|}, \tag{17}$$

where $\boldsymbol{y}^{(m)}$ is a transient vector, and $\boldsymbol{w}^{(m)}$ is normalized at each iteration. Although the computational cost can be reduced by using PMFM, its combining weights are biased with a non-uniform noise power [7]

III. Fast algorithm

Each signal correlation matrix requires an order N^2L multiplication operations [7], [8], which is complicated and impractical in term of hardware systems as N and L increase. Therefore, a fast signal combining algorithm is proposed in this paper. The complicated multiplications can be replaced by much simpler operations.

A. Signum Polarization Model and CORDIC Method

According to [10], if $\boldsymbol{x_i}(t)$ is a Gaussian distribution with zero mean value, the cross-correlation function $R_{x_ix_j}(\tau)$ can be expressed as [10]

$$R_{x_ix_j}(\tau) = \sin\left[\frac{\pi}{2}H_{z_iz_j}(\tau)\right] \quad i,j = 1,2,\cdots N, \tag{18}$$

$$H_{z_iz_j}(\tau) = \lim_{T\to\infty}\frac{1}{T}\int_0^T z_i(t)z_j(t+\tau)dt \quad i,j = 1,2,\cdots,N, \tag{19}$$

$$z_i(t) = \begin{cases} +1, \forall x_i(t) > 0 \\ -1, \forall x_i(t) < 0 \end{cases}, \tag{20}$$

where $H_{z_iz_j}(\tau)$ is the polarization cross-correlation function, and $z_i(t)$ is a polarization signal. Based on above results, we consider using a SP model to calculate an estimation cross-correlation function $\hat{R}_{x_ix_j}(\tau)$ and an estimation of polarization cross-correlation function $\hat{H}_{z_iz_j}(\tau)$.

The mathematical model of the SP can be expressed as [9]

$$Signum(X) = \begin{cases} +1, \forall X > 0 \\ 0, \quad \forall X = 0 \\ -1, \forall X < 0 \end{cases}, \tag{21}$$

$$\boldsymbol{z_i}(k) = Signum(\boldsymbol{x_i}(k)) \quad k = 1,2,\cdots,L, \tag{22}$$

$$\hat{H}_{z_iz_j}(\tau) = \frac{1}{L}\sum_{k=1}^{L} z_i(k)z_j(k+\tau) \quad i,j = 1,2,\cdots N. \tag{23}$$

It can be seen that $\hat{H}_{z_iz_j}(\tau)$ can be calculated by using increase or decrease 1 operations rather than multiplications. Then, $\hat{R}_{x_ix_j}(\tau)$ can be obtained by substituting (23) into (18).

We could use the polynomials methods, such as CP, to solve the sine function in (18) [9]. $\hat{R}_{x_ix_j}(\tau)$ can be calculated by

$$\hat{R}_{x_ix_j}(\tau) = \sin\left[\frac{\pi}{2}\hat{H}_{z_iz_j}(\tau)\right] = \sin\left[\frac{\pi}{2}h\right] = \sum_{p=0}^{\infty}\theta_p T_p(h), \tag{24}$$

$$\theta_p = \begin{cases} \frac{2}{\pi}\int_{-1}^{+1}\frac{\sin\left[\frac{\pi}{2}h\right]T_p(h)}{\sqrt{1-h^2}}dh, p = 1,3,\cdots \\ 0, \qquad p = 0,2,\cdots \end{cases}, \tag{25}$$

where p is the order of Chebyshev polynomials, T_p is the pth Chebyshev polynomials, θ_p is the pth Chebyshev polynomials coefficients. So $\hat{R}_{x_ix_j}(\tau)$ in (24) can be rewritten as [9]

$$\hat{R}_{x_ix_j}(\tau) = \theta_1 h + \theta_3 h^3 + \cdots, p = 1,3,\cdots. \tag{26}$$

From above analysis, we can find that the extra multiplication operations must be involved in calculating $\hat{R}_{x_ix_j}(\tau)$ by using CP [9].

ISBN 978-90-827970-1-5 © EURASIP 2018

TABLE I
THE ACCURACY OF CORDIC METHOD

	$(\pi/2)\,h$	0	$\pi/12$	$\pi/6$	$\pi/4$	$\pi/3$	$5\pi/12$	$\pi/2$
	$R_{x_ix_j}(\tau)$	0	0.2588	0.5000	0.7071	0.8660	0.9659	1.0000
$n=4$	$\hat{R}_{x_ix_j}(\tau)$	0.0476	0.1998	0.4281	0.6374	0.9037	0.9798	0.9989
	$error$	4.76%	22.79%	14.38%	9.85%	4.35%	1.44%	0.11%
	$average\ error$				8.24%			
$n=8$	$\hat{R}_{x_ix_j}(\tau)$	0.0070	0.2530	0.5040	0.7121	0.8637	0.9675	1.0000
	$error$	0.70%	2.24%	0.08%	0.71%	0.27%	0.17%	0.00%
	$average\ error$				0.59%			
$n=12$	$\hat{R}_{x_ix_j}(\tau)$	0.0000	0.2591	0.5002	0.7070	0.8659	0.9659	1.0000
	$error$	0.00%	0.01%	0.01%	0.01%	0.01%	0.00%	0.00%
	$average\ error$				0.01%			

TABLE II
COMPUTATIONAL COMPLEXITY COMPARISON FOR ALL THE CONSIDERED ALGORITHMS

	No. of multiplications	No. of additions	No. of shift	No. of comparisons	No. of increase 1
SNR EIGEN	$2\left(N^2L\right)+N^3$	/	/	/	/
COP EIGEN	N^2L+N^3	/	/	/	/
AC EIGEN	$2\left(N^2L\right)+N^3$	/	/	/	/
PM	$N^2L+N^2m_{PM}$	/	/	/	/
PMFM	$2NLm_{PMFM}$	/	/	/	/
COP+SP+CP	N^3	$128N^2+N^2$	$128N^2$	NL	N^2L
AC+SP+CP	N^3	$256N^2+N^2$	$256N^2$	$2NL$	$2N^2L$
COP+SP+CORDIC	N^3	$24N^2$	$16N^2$	$8N^2+NL$	N^2L
AC+SP+CORDIC	N^3	$48N^2$	$32N^2$	$16N^2+2NL$	$2N^2L$

CORDIC is an iterative method to calculate trigonometric function, and the only operations it requires are addition and bitshift [11]. If a two-dimensional vector (x_I, y_I) is rotated to a vector (x_D, y_D), this computation is given by

$$\begin{bmatrix} x_D \\ y_D \end{bmatrix} = \cos\phi \begin{bmatrix} 1 & -\tan\phi \\ \tan\phi & 1 \end{bmatrix} \begin{bmatrix} x_I \\ y_I \end{bmatrix}, \tag{27}$$

$$\phi = \sum_{l=0}^{n-1} \delta_l \phi_l, \tag{28}$$

$$\phi_l = \arctan\left(2^{-l}\right) \quad l = 0, 1, 2, \cdots, n-1, \tag{29}$$

where n is the iterative number of CORDIC, ϕ is the rotation angle which decomposes into a sequence of micro-rotation angle ϕ_l, δ_l is the direction of ϕ_l. For each iteration, CORDIC can be expressed as

$$\begin{bmatrix} x_{l+1} \\ y_{l+1} \end{bmatrix} = \cos\phi_l \begin{bmatrix} 1 & -\delta_l 2^{-l} \\ \delta_l 2^{-l} & 1 \end{bmatrix} \begin{bmatrix} x_l \\ y_l \end{bmatrix}. \tag{30}$$

From (x_I, y_I) to (x_D, y_D), (27) can be rewritten as

$$\begin{bmatrix} x_D \\ y_D \end{bmatrix} = \mathrm{K} \left(\prod_{l=0}^{n-1} \begin{bmatrix} 1 & -\delta_l 2^{-l} \\ \delta_l 2^{-l} & 1 \end{bmatrix} \right) \begin{bmatrix} x_I \\ y_I \end{bmatrix}, \tag{31}$$

$$K = \frac{1}{P} = \prod_{l=0}^{n-1} K_l = \prod_{l=0}^{n-1} \cos\phi_l = \prod_{l=0}^{n-1} \frac{1}{\sqrt{1+2^{-2l}}}, \tag{32}$$

$$K = \lim_{n\to\infty} \prod_{l=0}^{n-1} K_l \approx 0.607253, \tag{33}$$

where K_l is the scale factor. K_l and ϕ_l in (29) are chosen to certain values according to the iterative number of CORDIC, so they can be calculated in advance and stored in a table.

Therefore, the iterative process of CORDIC method are as follows

$$\begin{array}{c} x_{l+1} = x_l - \delta_l y_l 2^{-l} \\ y_{l+1} = y_l + \delta_l x_l 2^{-l} \\ z_{l+1} = z_l - \delta_l \arctan\left(2^{-l}\right) \\ \delta_l = \begin{cases} +1 & \forall z_l > 0 \\ -1 & \forall z_l < 0 \end{cases} \end{array}, \tag{34}$$

where z_l is the remainder angle from the rotation angle ϕ after l iterations. If z_l is positive, the rotation is clockwise and $\delta_l = +1$, otherwise the rotation is counterclockwise and $\delta_l = -1$. Since $x_l 2^{-l}$ and $y_l 2^{-l}$ can be implemented by a simple bitshift operation for hardware, the only operations CORDIC requires are addition and bitshift.

Let $x_0 = K$, $y_0 = 0$, and $z_0 = \phi = (\pi/2)\,h$ in (24), after n iterations, the final outputs of CORDIC are $x_n = \cos\phi$, $y_n = \sin\phi = \hat{R}_{x_ix_j}(\tau)$, and $z_n = 0$, respectively. As a result, combining the SP model with CORDIC method, $\hat{R}_{x_ix_j}(\tau)$ can be obtained without multiplication operations.

ISBN 978-90-827970-1-5

B. Computational Accuracy and Complexity

The accuracy of CORDIC method is summarized in Table I, $(\pi/2)\,h$ in (24) is equal to different values. It can be seen that CORDIC method with $n=4$ has a poor estimation accuracy. On the other hand, CORDIC method with $n=12$ provides an excellent accuracy but needs more computational operations. CORDIC method with $n=8$ are a good tradeoff between accuracy and computational cost. Therefore, CORDIC method with $n=8$ is used to approximately calculate $\hat{R}_{x_i x_j}(\tau)$. Consequently, the length of table stored K_l and ϕ_l is merely 8.

A third order CP ($p=3$) provides a comparable accuracy 0.52% for the same $(\pi/2)\,h$ [9], but this counterpart needs 4 multiplications and 1 addition for each $\hat{R}_{x_i x_j}(\tau)$ based on (26). For a 32-bit hardware system, each multiplication includes 32 addition and 32 bitshift operations [12]. Therefore, each $\hat{R}_{x_i x_j}(\tau)$ used the third order CP involves 128 addition and 128 bitshift operations. On a contrary, each $\hat{R}_{x_i x_j}(\tau)$ used CORDIC method with $n=8$ only includes 24 addition, 16 bitshift, and 8 comparison operations according to (34).

The computational complexity comparison for all the considered algorithms are summarized in Table II, where / notation represents none. m_{PM} and m_{PMFM} are the iterative number of PM and PMFM, respectively. The computational burden of SNR, COP and AC EIGEN are much heavier than that of the other methods. When N is greater than m_{PM}, PM is an appropriate choice to solve the eigenvector problem [8]. The combining weights of PMFM are biased with a non-uniform noise power [7]. For each signal correlation matrix, the computation cost of SP + CP method still remains $4N^2$ multiplications, which is equivalent to $128N^2$ addition and $128N^2$ bitshift operations based on above discussions. On the other hand, for SP + CORDIC method, these multiplications can be replaced by $24N^2$ addition, $16N^2$ bitshift, and $8N^2$ comparison operations. Therefore, more computational consumption can be saved as N increases [7], [8]. Moreover, the proposed algorithm SP + CORDIC can be flexibly implemented with COP or AC EIGEN for different environments.

IV. Simulation Results

This section contains illustrative simulations aimed at showing the effectiveness of the proposed algorithm SP + CORDIC in terms of the combining performance. A 80KHz sine signal with random phase is the source signal, the sample frequency is 1.4MHz, and $n_i(k)$ is independent white Gaussian noises [9]. $\boldsymbol{\alpha}=[1,1,\cdots,1]^T$, τ of AC EIGEN is selected as 1, and the noise variances are $1:1:1.5:1.5$ for $N=4$ under a non-uniform noise power condition [7]. The iterative number $m_{PMFM}=30$.

The combining loss for different N and L with an uniform noise power is shown in Fig. 1. We notice that COP EIGEN, PMFM, SP + CP, and SP + CORDIC provide a better combining performance compared to AC EIGEN with an uniform noise power. Moreover, it also can be seen that the greater L, the less combining loss for all these methods.

The comparative performance with a non-uniform noise power is displayed in Fig. 2. One can find that the combining loss of AC EIGEN is the minimum with a non-uniform noise power. At the same time, the combining performance of SP + CP and SP + CORDIC is far better than that of COP EIGEN and PMFM under this condition.

V. Conclusion

A fast eigen-based algorithm that combines the SP model with CORDIC method is proposed in this paper. This algorithm can use addition and bitshift operations to replace the multiplications in the signal correlation matrices. Therefore, it can save more computational consumption as N and L increase. In addition, for different noise power environments, this algorithm can be flexibly implemented to COP or AC EIGEN. Mathematical analysis and computational cost for our algorithm are also presented. Simulation results have shown that the proposed algorithm can effectively reduce the computational cost while it provides a good combining performance.

Acknowledgment

The authors are with the Key Laboratory of Information Technology for Autonomous Underwater Vehicles, Chinese Academy of Sciences. This work was supported by the Young Talent Program of Institute of Acoustics, Chinese Academy of Science under Grant QNYC201622 and the National Natural Science Foundation of China under Grant 61571434.

References

[1] Rogstad, D.H. "The SUMPLE algorithm for aligning arrays of receiving radio antennas: coherence achieved with less hardware and lower combining loss." *IPN Progress Report*, vol. 42, no. 162, pp. 1-29, Aug, 2005.
[2] Shang, Y. and Feng, X.T. "MLC-SUMPLE algorithm for aligning antenna arrays in deep space communication." *IEEE Trans. on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2828-2834, Oct, 2013.
[3] Cheung, K.M. "Eigen theory for optimal signal combining: a unified approach." *TDA Progress Report*, vol. 42, no. 126C, pp. 1-9, Aug, 1996.
[4] Tan, H.H. "Optimum combining of residual carrier array signals in correlated noises." *TDA Progress Report*, vol. 42, no. 124, pp. 33-52, Feb, 1996.
[5] Doclo, S, and Moonen, M. "Robust time-delay estimation in highly adverse acoustic environments." *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics 2001*, New York, pp. 59-62, Oct. 2001.
[6] Lee, C.H. Vilnrotter, V, Satorius, E, Ye, Z, Fort, D, Cheung, K.M. "Large-array signal processing for deep space applications." *IPN Progress Report*, vol. 42, no. 150, pp. 1-28, Aug, 2002.
[7] Luo, B, Yu, H, Zhang, X, and Li, Q. "On eigen-based signal combining using the autocorrelation coefficient." *IET Communications*, vol. 6, no. 18, pp. 3091-3097, Sep, 2012.
[8] Lee, C.H, Cheung, K.M, and Vilnrotter, V.A. "Fast eigen-based signal combining algorithms for large antenna arrays." *IEEE Aerospace Conference*, Montana, pp. 1123-1129, Mar. 2003.
[9] Wang, L.O, Wang, D.H, and Hao, C.P. "A signum polarization fast eigen-based signal combining algorithm." *IEEE Trans. on Aerospace and Electronic Systems*, Accept.
[10] Van Vleck, J.H, and Middleton, D. "The spectrum of clipped noise." *Proceedings of The IEEE*, vol. 54, no. 1, pp. 2-19, Jan, 1966.
[11] Hsieh, Q.H, Chang, C.F, and Kao, M.S. "Efficient acquisition algorithm for long pseudorandom sequence." *IEEE Trans. on Aerospace and Electronic Systems*, vol. 50, no. 3, pp. 1786-1797, Jul, 2014.
[12] Tang, G.M, Takagi, K, and Takagi, N. "32 x 32 -bit 4-bit bit-slice integer multiplier for RSFQ microprocessors." *IEEE Trans. on Applied Superconductivity*, vol. 27, no. 3, pp. 1301005, Apr, 2017.

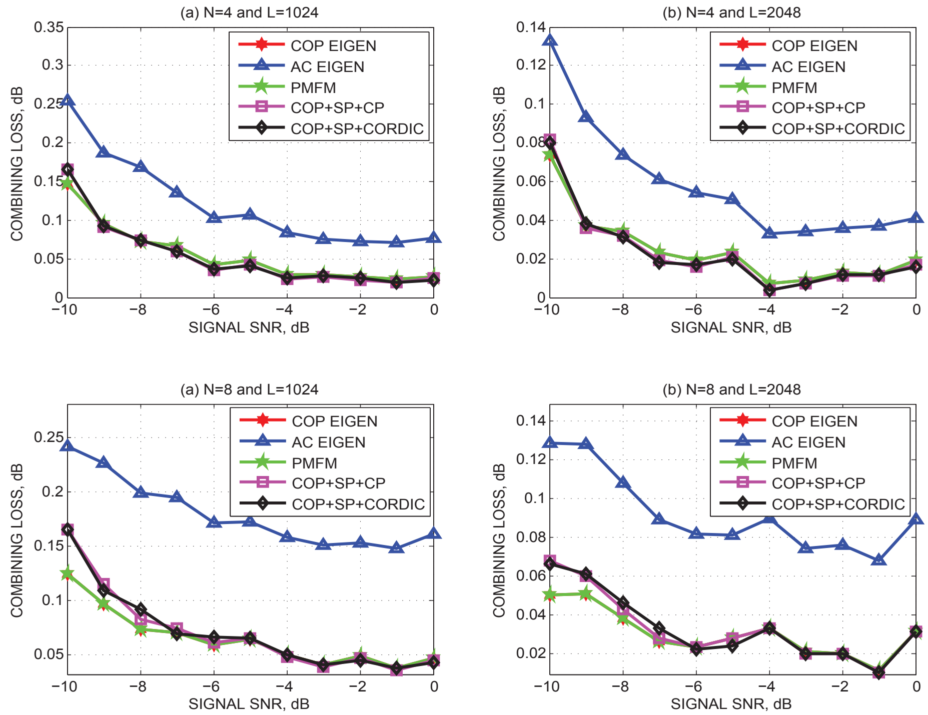

Fig. 1. The combining loss for different N and L with an uniform noise power.

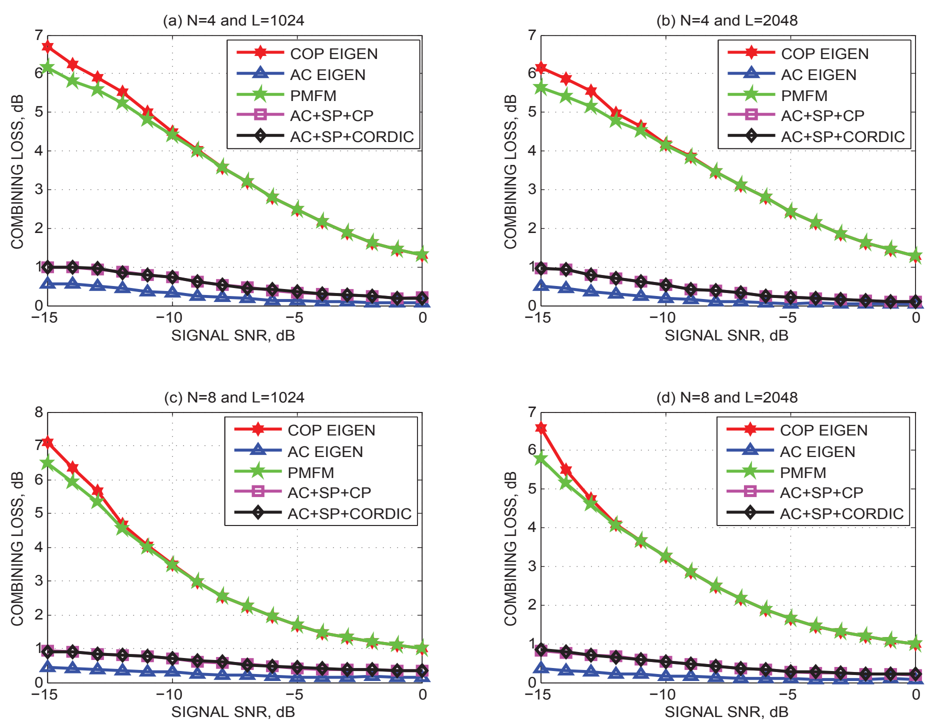

Fig. 2. The combining loss for different N and L with a non-uniform noise power.

ISBN 978-90-827970-1-5 © EURASIP 2018