Maximin Algorithm With a Step-Length Estimation Technique

Raja D. Balakrishnan, Hyuck M. Kwon, *Senior Member, IEEE*, Dong-Hyeuk Yang, and Yong H. Lee, *Senior Member, IEEE*

***Abstract*—We propose a technique that enhances the performance of Torrieri and Bakhru's maximin algorithm in the presence of frequency offsets and modulated interferences, a common scenario in multiple-access environments, where we observed that the Maximin algorithm suffers performance degradation in the aforesaid scenario. To combat modulated interferences and fading effects, we propose to update the step length during every smart antenna weight vector update. We found that our scheme improves the robustness of the maximin algorithm to a greater extent. Simulation results for a frequency-hopping system confirm the validity of the approach and demonstrate improvements in the performance of the maximin algorithm in the presence of frequency offsets and data modulated interferences in a fading channel environment.**

***Index Terms*—Frequency hopping, maximin, smart antenna, step length.**

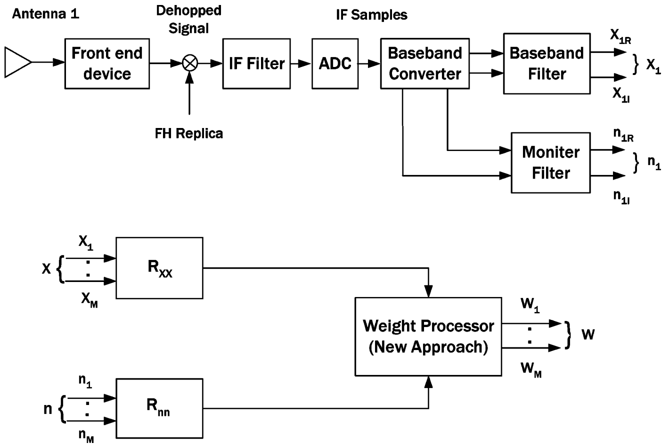


Fig. 1. Block diagram of a typical FH system with array antennas.

I. Introduction

Most adaptive algorithms differentiate between the desired signal and the interfering signal based on the angle of arrival of the signals that impinge the antenna array. Other algorithms make this differentiation based on maximization of the desired signal power at the output of the antenna array.

Shor [1] demonstrated that the method of "steepest ascent" could be employed to increase the desired signal power while differentiating the coherent interferences. Torrieri and Bakhru [2] applied a version of Shor's algorithm to a frequency-hopping (FH) communication system and called it *maximin*. They proposed a receiver structure operating on two frequency bands—one to estimate the power in the desired signal band and the other to estimate the noise and interference powers in a frequency band adjacent to the desired signal band. Maximin is a blind algorithm since it does not require any training sequences or decision-directed adaptation or knowledge of the directions of arrival. It was shown that the maximin algorithm requires only synchronization of the frequency-hop pattern of the transmitted signal with that of the receiver for it to perform. They demonstrated the ability of maximin to maximize the desired signal power while simultaneously suppressing noise and interferences. In addition, we found that the maximin algorithm suffers when the frequency offset $\Delta\mathrm{f}$ is larger than $1/(1.5\mathrm{T_b})$, where $\mathrm{T_b}$ is a bit time interval.

We observed that the performance of the maximin algorithm is drastically affected in the presence of *friendly* interferences, i.e., when the interferences are also users in the same system. We also noted that the maximin algorithm is sensitive to the initial weight vector and is sluggish in a fading channel.

We propose a technique to enhance the performance of the maximin algorithm by updating the step length (also known as the adaptation parameter), based on a maximization criterion instead of retaining it as a constant. We call the proposed scheme maximin algorithm with step-length estimation (MOSTLE). Simulation results for an FH system reveal the improvement in convergence time and robustness in the presence of friendly interferences and frequency offsets in a fading channel.

We proceed to Section II, where we describe the system model and the development of the adaptive algorithm. In Section III, we present our technique and discuss its salient features. We demonstrate the efficiency of our method through simulations in Section IV, where we present comparisons of performances with maximin. Finally, we draw conclusions in Section V.

II. System Model

We consider the FH system described in Fig. 1. The frequency-hopping replica generated by a synchronized local frequency synthesizer removes the time variations in the received signal. The signal is then processed in an intermediate frequency (IF) range, where it is sampled at a rate prescribed in Table I. The sampled values of the complex envelope of the dehopped signal are then applied to the adaptive filter.

Manuscript received June 22, 2005; revised February 18, 2006. This work was supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under Contract/Grant DAAD19-01-10537.

R. D. Balakrishnan, H. M. Kwon, and D.-H. Yang are with the Department of Electrical and Computer Engineering, Wichita State University, Wichita, KS 67260-0044 USA (e-mail: hyuck.kwon@wichita.edu).

Y. H. Lee is with the Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea (e-mail: yohlee@ee.kaist.ac.kr).

Digital Object Identifier 10.1109/TAP.2006.880720

TABLE I
SIMULATION PARAMETERS

Parameters	**Value**
Antenna array	Square with M=4 omni antennas at the vertices, or a uniform linear array (ULA) with M omni antennas
Array edge length (d)	λ
DOA of desired user	0°
DOA of interferences	40°, 70°
Center frequency	3 GHz
Hop dwell time *(T_H)*	1 ms
Data rate	100 *kbps*
Frequency modulation	Minimum Shift Keying (MSK)
Signal-to-noise-ratio	20 dB per antenna per channel
Hopping bandwidth	30 MHz
Number of frequency hopping channels	$\mathrm{N_{FHC}}$ = 300
Monitor filter offset (f_o)	200 KHz
Sampling rate	800 kilo-samples per second
Weight iterations per hop	8
Total interference-to-signal-ratio	10 $\mathrm{N_{FHC}}$
Interference type	Tones and MSK signals in all channels
Number of hops per experiment	50
Fading model	Jakes
Vehicular speed	50 km/h

We consider an arbitrary antenna array of M antennas. The discrete-time vector of complex envelopes at the output of the baseband filters is

$$x(i) = s(i) + n(i) \tag{1}$$

where $s(i)$ is the vector of the desired-signal complex envelopes and $n(i)$ is the vector of noise plus interferences. We note that $x(i)$, $s(i)$, $n(i) \in C^M$, and i denote the sample index.

Treating the complex weight vector $w(i)$ as a constant during the update period, we obtain

$$y(i) = \omega^T x(i) = y_s(i) + y_n(i) \tag{2}$$

where

$$y_s(i) = \omega^T s(i) \tag{3a}$$
$$y_n(i) = \omega^T n(i) \tag{3b}$$

and the superscript ^{T} denotes the transpose.

Following [3], we define the power of the complex signal component as

$$P_s = E\left[|y_s(i)|^2\right] = \omega^\dagger R_{ss}\omega \tag{4}$$

where the signal autocorrelation matrix is defined as

$$R_{ss} = E\left[s^*(i)s^T(i)\right] \tag{5}$$

the superscript † denotes the conjugate transpose, and * is the complex conjugation.

Similarly, the noise-plus-interference power is

$$P_n = \frac{1}{2} \cdot E\left[|y_n(i)|^2\right] = \omega^\dagger R_{nn}\omega \tag{6}$$

with the noise-plus-interference autocorrelation matrix given as

$$R_{nn} = E\left[n^*(i)n^T(i)\right]. \tag{7}$$

The signal-to-(interference plus noise) ratio (SINR) is given as

$$\rho = \frac{P_s}{P_n} = \frac{\omega^\dagger R_{ss}\omega}{\omega^\dagger R_{nn}\omega}. \tag{8}$$

The method of the steepest descent [4] applied to discrete-time systems gives rise to

$$\omega(k+1) = \omega(k) + \mu(k)\nabla_\omega J(k) \tag{9}$$

where $\nabla_\omega J(k)$ represents the gradient of the cost function $J(k)$ with respect to ω and k denotes the update index. If we regard $\rho(k)$ as the cost function, then (9) becomes

$$\omega(k+1) = \omega(k) + \mu(k)\nabla_\omega \rho(k) \tag{10}$$

where

$$\nabla_\omega \rho(k) = \rho(k)\left[\frac{R_{ss}(k)\omega(k)}{P_s(k)} - \frac{R_{nn}(k)\omega(k)}{P_n(k)}\right] \tag{11}$$

and $R_{ss}(k)$, $R_{nn}(k)$, $P_s(k)$, and $P_n(k)$ vary with every update in practice and need to be estimated.

III. PROPOSED TECHNIQUE

We define a new cost function $f(\mu(k))$ such that

$$f(\mu(k)) = \frac{\widetilde{P}_s}{\widetilde{P}_n} \tag{12}$$

with

$$\widetilde{P}_s = \omega^\dagger(k+1)\hat{R}_{ss}(k)\omega(k+1) \quad (13a)$$

$$\widetilde{P}_n = \omega^\dagger(k+1)\hat{R}_{nn}(k)\omega(k+1) \quad (13b)$$

where $\hat{x}$ denotes the estimate of a quantity x.

The process of finding $\mu(k)$ among all $\mu \in \mathbb{R}$ is called the line search. Rewriting (12) with respect to the step-length parameter $\mu(k)$, we obtain

$$f(\mu(k)) = \frac{A_1\mu^2(k) + B_1\mu(k) + C_1}{A_2\mu^2(k) + B_2\mu(k) + C_2} \quad (14)$$

where the coefficients A_q, B_q, C_q, and $q = 1, 2$ can be found in Appendix A, and the update index k is suppressed for brevity.

Observe that $f(\mu(k))$ is scalar valued with critical points that can be obtained by differentiating with respect to $\mu(k)$, yielding

$$\mu(k) = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \quad (15)$$

where A, B, and C are computed in Appendix A. Thus, we obtain two specific values of $\mu(k)$ among the infinite possibilities. We observed through simulations that the sequence $\{\mu(k)\}$ converges to zero as $k \to \infty$ and is shown in Fig. 7.

A. Autocorrelation Updates

We present two methods to estimate the matrices R_{ss} and R_{nn} during every update.

1) Singular Value Decomposition: Define

$$X = [x(1), \ldots, x(m)] \quad (16)$$

where $x(i) \in C^M$ is defined in (1) and $i = 1, \ldots, m$, and m is the number of observations per iteration. A singular value decomposition (SVD) of X gives

$$X = S_X \cdot \sum\nolimits_X \cdot V_X \quad (17)$$

where $S_X \in \mathbb{C}^{\mathrm{M}\times\mathrm{M}}$, $\sum_X \in \mathbb{C}^{M\times m}$, and $V_X \in \mathbb{C}^{m\times m}$, with S_X and V_X being unitary.

Defining the estimate of R_{xx} as

$$\hat{R}_{XX}(k) = \sum_{i=1}^{m} x(i)x^\dagger(i) \quad (18)$$

we readily have

$$\hat{R}_{XX}(k) = XX^\dagger = S_X \cdot \sum\nolimits_X^2 \cdot S_X^\dagger \quad (19)$$

which indicates that we need only to compute the left singular vectors and the corresponding singular values of X, which requires fewer computations compared to a full SVD. For (19), we need to find $\mathrm{S_X}$ and Σ_X because $\mathrm{S_X} = \mathrm{V_X}$ for a Hermitian matrix. However, for (17), we need to compute $\mathrm{S_X}$, Σ_X, and $\mathrm{V_X}$ because X is a nonsquare matrix. Therefore, the computational cost for (19) is smaller than that for a full SVD in (17).

In a similar way, we can estimate R_{nn} as

$$\hat{R}_{nn}(k) = \sum_{i=1}^{m} n(i)n^\dagger(i) = NN^\dagger = S_N \cdot \sum_N^2 \cdot S_N^\dagger \quad (20)$$

where $N = [n(1), \ldots, n(m)] = S_N \cdot \sum_N \cdot V_N$, with $S_N, \sum_N$, and V_N defined in a similar manner.

2) Recursive Updates: We also can define the estimates of R_{xx} and R_{nn} as

$$\begin{aligned}\hat{R}_{xx}(k) &= \sum_{i=1}^{km} \gamma^{k-i} x(i)x^\dagger(i) \\ &= \gamma\hat{R}_{xx}(k-1) + \sum_{i=(k-1)m+1}^{km} x(i)x^\dagger(i) \quad (21)\end{aligned}$$

and

$$\begin{aligned}\hat{R}_{nn}(k) &= \sum_{i=1}^{km} \gamma^{k-i} n(i)n^\dagger(i) \\ &= \gamma\hat{R}_{nn}(k-1) + \sum_{i=(k-1)m+1}^{km} n(i)n^\dagger(i) \quad (22)\end{aligned}$$

where γ is referred to as the forgetting factor.

We observe that (19) and (20) are special cases of (21) and (22) when $\gamma = 0$. We note that the choice of γ plays a crucial role in the success of the latter scheme. For instance, a choice of $\gamma = 0.99$ will produce better estimates of R_{xx} and R_{nn} in a stationary environment, whereas $\gamma = 0$ is the best choice for a nonstationary environment, such as in the presence of multiple-access or partial band-jamming interferences. We invite the reader to [5] for performance results in the presence of partial band-jamming.

We note from (1) that

$$\hat{R}_{ss}(k) = \hat{R}_{xx}(k) - \hat{R}_{nn}(k) \quad (23)$$

where $\hat{R}_{ss}(k)$ is the estimate of the signal autocorrelation matrix during the kth update.

We observe from [6] that the maximin algorithm also employs the same autocorrelation estimates R_{xx} and R_{nn}. Unlike (19) and (20), Torrieri and Bakhru [6] employ an approximation under the assumption of a nonstationary environment. One might say that both the maximin algorithm and the autocorrelation update using SVD are theoretically the same, whereas their implementation is different.

IV. Simulations

We consider an array of four omnidirectional antennas located at the vertices of a square. The symmetry of this structure allows full azimuth coverage. The edge length is equal to one or two times the wavelength corresponding to the center frequency. Each interference source is in the plane of the array, and all the signals are assumed to arrive as plane waves. Refer also to Table I.

The FH signal has a randomly chosen carrier frequency within the hopping band and is modulated by binary minimum shift keying (MSK). The sequence of data bits is randomly generated at the rate of 100 kbps with a hop dwell time of 1 ms. The thermal noise at the output of each IF filter following an antenna is modeled as filtered white Gaussian noise. The ratio of signal power to thermal noise power (SNR) is defined relative to a single-frequency channel and is set at 20 dB. Since the FH signal is assumed to be noncoherent in carrier phase from one hop to another, each FH pulse has an initial phase that is uniformly distributed over the interval $[-\pi, \pi]$. The hopping frequencies are separated by 100 kHz and spread uniformly over the total hopping band, which occupies either 30 or 300 MHz. Thus, there are either 300 or 3000 contiguous frequency channels. The total interference power due to all interference signals is equal to ten times the number of frequency channels, which maintains a constant value of interference power per frequency channel. Thus, ρ_1 is slightly less than -10 dB. Here we consider the cases when the interferences are tone- and MSK-modulated.

Perfect synchronization between the frequency-hopping signal at all antenna outputs and the frequency synthesizer in the receiver is assumed. The baseband and monitor filters are modeled as digital eight-pole Butterworth filters with 3-dB bandwidths equal to 100 kHz, the bandwidth of a frequency channel. The monitor filter has a single passband with center frequency of $f_0 = 200$ kHz and a bandwidth equal to that of the baseband filter. The sampling rate of the analog-to-digital-converter is 800-ksamples per second. There are eight weight iterations per hop.

We consider the initial weight vector $\omega(0) = [1, 0, 0, 0]^T$ to represent the *cold* start of the algorithm. We set $\mu(0)$ to a small positive constant 0.1 for our simulations. We observed that scaling $\mu(k)$ by a factor of one-tenth expedites the convergence, and hence, we divide $\mu(k)$ by ten for every iteration. As noted in the simulations, the initial value $\mu(0)$ plays an important role, and hence, we introduce the scaling by a factor of ten. We also note that $f(\mu(k))$ of (14) is a *suboptimal* cost function to find $\mu(k)$, and hence, a scaling factor proved essential to expedite the convergence.

For fairness in comparison, we compute the SVD of the correlation matrices and use them in conjunction with the maximin algorithm. The results in this section juxtapose our approach with maximin [2] and maximin with SVD computations. We introduce the technique of computing the autocorrelation matrices via SVD for the maximin algorithm. This explicitly calculates the products $R_{ss}(k)\omega(k)$ and $R_{nn}(k)\omega(k)$, while the original maximin algorithm approximates it by time averages.

We note that the two methods used to estimate autocorrelation matrices, i.e., SVD computations and recursive updates, yielded performances that were very close to one another for the scenarios we considered. Hence, we present our scheme with the autocorrelation matrices computed via the SVD.

In the course of simulations, we observed that the choice of

$$\mu^{-}(k) = \frac{-B - \sqrt{B^2 - 4AC}}{2A} \tag{24}$$

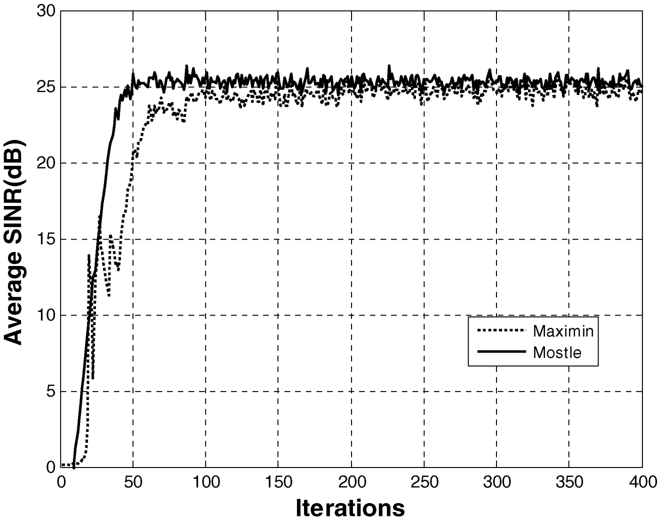


Fig. 2. Average SINR for 20 trials under AWGN channel in the presence of tone interferences, assuming input SNR $= 20$ dB, $\Delta\mathrm{f} = 0$, $\mathrm{M} = 4$ square array, and $\omega(0) = [1, 0, 0, 0]$. The maximin algorithm used a convergence parameter of $\alpha = 0.2$.

yielded consistent performance over all the trials, while

$$\mu^{+}(k) = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \tag{25}$$

resulted in inconsistent performance. Hence, we opted to use $\mu^{-}(k)$ for all our simulations, though theoretically both solutions stand valid. We conjecture that $\mu^{-}(k)$ may correspond to a local maxima, while $\mu^{+}(k)$ relates to the local minima. Since $f(\mu(k))$ attempts to maximize the SINR based on $\mu(k)$, the solution $\mu^{-}(k)$ is an appropriate choice. We comment that a rigorous mathematical analysis of the convergence of the weight vector to the optimum weight vector and the convergence of the step-length parameter are highly complicated. Hence, we present simulation results to validate our claims.

Fig. 2 depicts the performance of maximin and the proposed approach MOSTLE in the presence of tone interferences in an additive white Gaussian noise (AWGN) channel, assuming an input SNR equal to 20 dB, frequency offset $\Delta\mathrm{f} = 0$, $\mathrm{M} = 4$ square array, and $\omega(0) = [1, 0, 0, 0]$. The maximin algorithm used a convergence parameter of $\alpha = 0.2$. We note that the proposed approach converges faster than the maximin algorithm for a cold start. We observe that there is no immediate performance gain for the proposed approach except the faster convergence.

We define the "final SINR" as the average of the last 40 iterations of all the trials, which is used to calculate the corresponding standard deviations. We also define "crossing number" as the number of weight iterations required to exceed a threshold value equal to 3 dB less than the final SINR. This provides a rough measure of the relative time required for convergence to steady state. Then, we make the final SINR and crossing number observations during our extensive simulations.

Tables II and III tabulate the simulation results under various conditions and compare the maximin algorithm and the proposed scheme, respectively, assuming the initial weight vector $\omega(0) = [1, 0, 0, 0]^T$, frequency offset $\Delta\mathrm{f} = 0$, $\mathrm{M} = 4$

TABLE II
SIMULATION RESULTS FOR MAXIMIN ALGORITHM UNDER AWGN CHANNEL ($\mathrm{M} = 4$ SQUARE ARRAY, $\Delta \mathrm{f} = 0, \omega(0) = [1,0,0,0], \alpha = 0.2$)

Input SNR (dB)	DOAs of Interferences	Final SINR (dB)	Standard Deviation (dB)	Crossing Number
20	40°, 70°	24.76	1.42	66
10	40°, 70°	15.51	0.44	53
10	20°, -20°	15.04	0.35	67
2	20°, -20°	7.52	-0.31	33

TABLE III
SIMULATION RESULTS FOR PROPOSED SCHEME UNDER AWGN CHANNEL ($\mathrm{M} = 4$ SQUARE ARRAY, $\Delta \mathrm{f} = 0, \omega(0) = [1,0,0,0]$)

Input SNR (dB)	DOAs of Interferences	Final SINR (dB)	Standard Deviation (dB)	Crossing Number
20	40°, 70°	25.34	1.41	37
10	40°, 70°	15.50	0.35	26
10	20°, -20°	14.00	0.29	29
2	20°, -20°	7.61	- 0.16	24

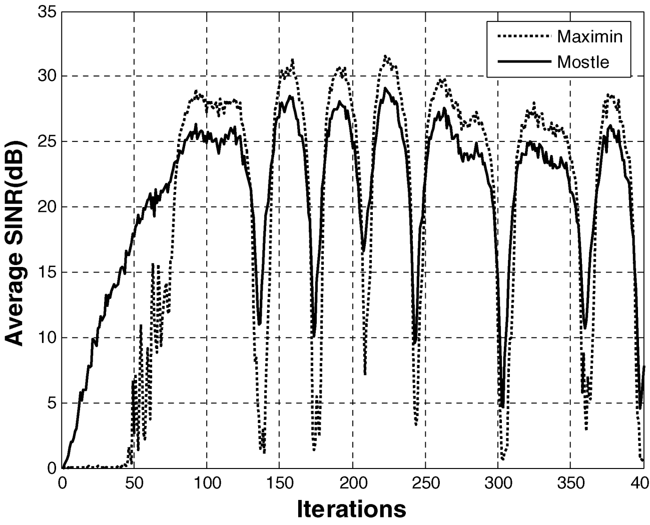


Fig. 3. Average SINR for 20 trials under fading channel in the presence of tone interferences, assuming input SNR $= 20$ dB, $\Delta \mathrm{f} = 0, \mathrm{M} = 4$ square array, and $\omega(0) = [1,0,0,0]$. The maximin algorithm used a convergence parameter of $\alpha = 0.2$.

square array, an AWGN channel, and input SNR $=$ 20 dB. The maximin algorithm used a convergence parameter of $\alpha = 0.2$. Tables II and III confirm that the convergence speed of MOSTLE is faster than that of maximin because the crossing numbers for the proposed algorithm are half those for the maximin algorithm.

Fig. 3 shows that the proposed approach converges faster than the maximin algorithm and follows the signal fluctuations in the fading channel better. This indicates that our method is robust even under a fading channel. The same simulation parameters as in Fig. 2 were used, but with a different channel, i.e., a fading channel instead of an AWGN channel.

Fig. 4 shows the normalized fading amplitude used in the simulations of Fig. 3. A Jakes' fading model was employed with a vehicular speed of 50 kmph.

In Fig. 5, we employed the same simulation parameters as in Fig. 2, except for a different interference environment, i.e., *friendly* interference, or data-modulated signals from other users instead of tone interference. And we modified the maximin algorithm to include autocorrelation computations, via SVD, so

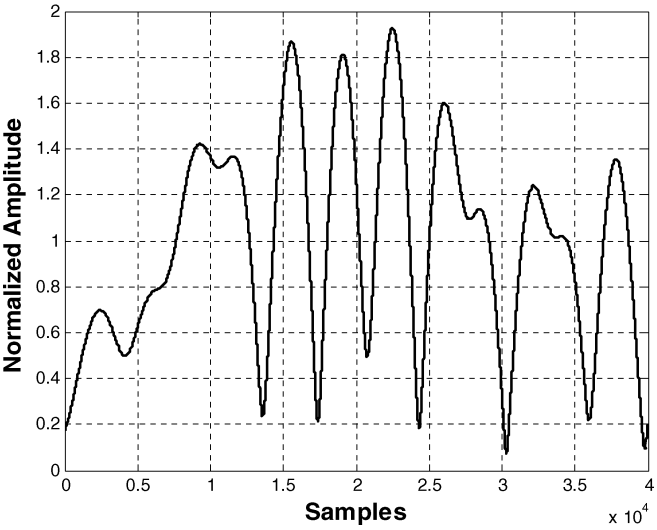


Fig. 4. Fading pattern for Fig. 3.

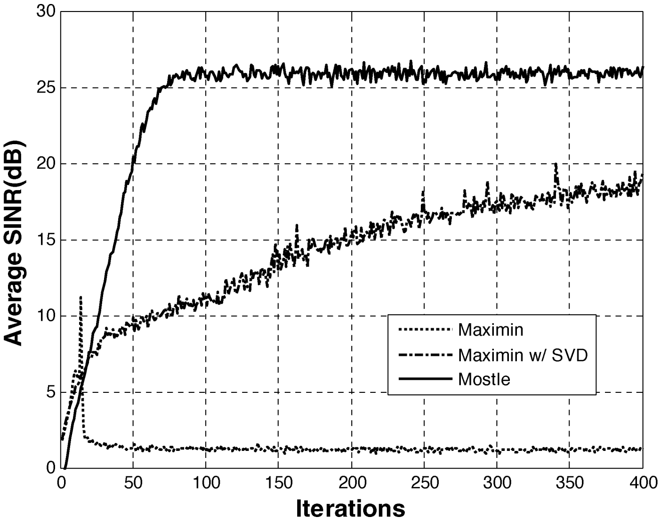


Fig. 5. Average SINR for 20 trials under AWGN channel in the presence of friendly interferences, assuming input SNR $= 20$ dB, $\Delta \mathrm{f} = 0, \mathrm{M} = 4$ square array, and $\omega(0) = [1,0,0,0]$. The maximin algorithm used a convergence parameter of $\alpha = 0.2$.

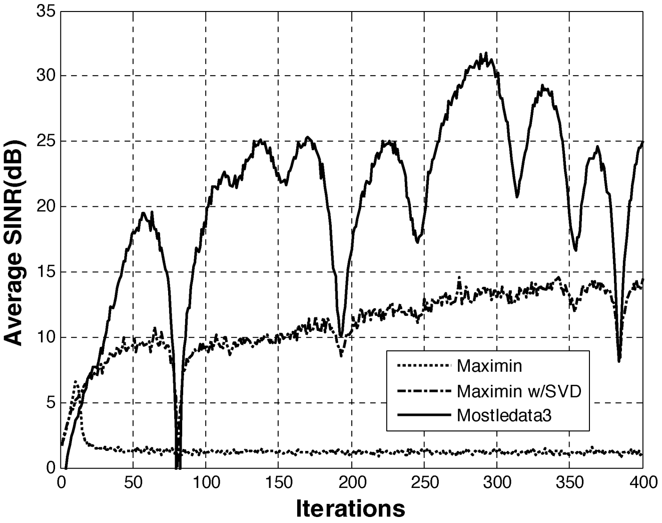


Fig. 6. Average SINR for 20 trials under fading channel in the presence of friendly interferences, assuming input $\mathrm{SNR} = 20$ dB, $\Delta \mathrm{f} = 0, \mathrm{M} = 4$ square array, and $\omega(0) = [1, 0, 0, 0]$. The maximin algorithm used a convergence parameter of $\alpha = 0.2$.

that it could be compared with the proposed scheme. For clarity, we also provided the original maximin algorithm without the autocorrelation computations. We observed that our method is robust in the presence of *friendly* interferences, while the maximin algorithm suffers severely, and the maximin algorithm with SVD computations fares better but still falls behind the proposed technique. This scenario is typically encountered in a multiple-access FH system where users in the system try to communicate.

Fig. 6 shows the performance of the proposed technique under a fading environment in the presence of friendly interferences. It is evident that our method performs better than both the maximin algorithm and the maximin with SVD computations. This scenario is also encountered in practice when users move at a reasonable speed. The same simulation parameters as in Fig. 5 were used, except the different channel, i.e., a fading channel instead of an AWGN channel. For simulations, we assume that the desired user and the interferences undergo independent fading channels of the same characteristics and that they all travel with a vehicular speed of 50 km/h. We note that the performance of our technique is unaffected when the interferences have independent fades or different speeds.

Fig. 7 shows the behavior of the adaptation parameters $\mu(k)$ for the scenario described in Fig. 6. We find that the adaptation parameter used in our approach converges faster to zero, while that of the maximin algorithm does not. Also shown is the convergence of the adaptation parameter of the maximin algorithm with autocorrelation estimations via the SVD. Even though one notes the convergence of the adaptation parameter for the latter scheme, it does not guarantee SINR performance of the algorithm. This demonstrates the need for computing the adaptation parameter for the aforesaid scenarios.

Now, we introduce a nonzero frequency offset $\Delta \mathrm{f}$ for the desired user MSK signal for more realistic applications and observe the effects of this frequency offset on performance. Fig. 8 shows the average SINR for 20 trials under an AWGN channel in the presence of tone interferences and frequency offset $\Delta \mathrm{f} = 1/(1.5\mathrm{T_b})$, assuming input $\mathrm{SNR} = 20$ dB, $\mathrm{M} = 4$ square array, and $\omega(0) = [1, 0, 0, 0]$. The maximin algorithm used a convergence parameter of $\alpha = 0.2$. Comparing Figs. 2 and 8, we observe that the frequency offset can cause degradation in the final SINR by 2.5 dB for both algorithms. However, we observe that the frequency offset does not degrade the convergence speed of the proposed MOSTLE, but it does that of maximin noticeably, compared to the zero frequency offset case.

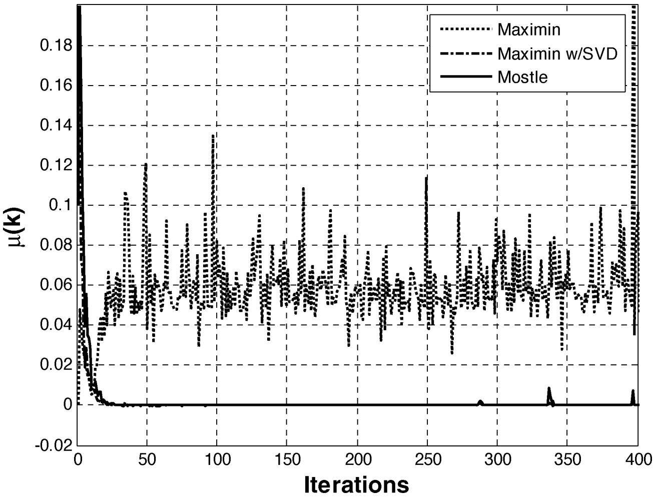


Fig. 7. Comparison of adaptation parameters for Fig. 6, assuming input $\mathrm{SNR} = 20$ dB, $\Delta \mathrm{f} = 0, \mathrm{M} = 4$ square array, and $\omega(0) = [1, 0, 0, 0]$. The maximin algorithm used a convergence parameter of $\alpha = 0.2$.

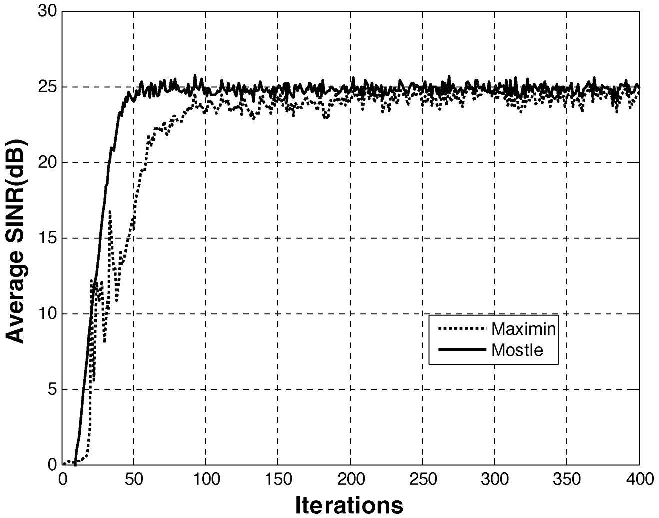


Fig. 8. Average SINR for 20 trials under AWGN channel in the presence of tone interferences and frequency offset $\Delta \mathrm{f} = 1/(1.5\mathrm{T_b})$, assuming input $\mathrm{SNR} = 20$ dB, $\mathrm{M} = 4$ square array, and $\omega(0) = [1, 0, 0, 0]$. The maximin algorithm used a convergence parameter of $\alpha = 0.2$.

Table IV(a) and (b) lists the final SINR in decibels under tone and friendly user interference, respectively, for various frequency offsets, assuming a uniform linear array (ULA) of $\mathrm{M} = 4$ elements and antenna element spacing equal to a half-wavelength of the desired signal, an AWGN channel, input $\mathrm{SNR} = 20$ dB, and $\omega(0) = [1, 0, 0, 0]$. The maximin algorithm used a convergence parameter of $\alpha = 0.2$. We observe from Table IV(a) that the frequency offset does not degrade all three schemes under tone interference when $\Delta \mathrm{f}$ is smaller than $1/(1.5\mathrm{T_b})$. However, the final SINRs become unacceptably

TABLE IV
(a) Final SINR in dB Under Frequency Offset and Tone Interference ($\mathrm{M} = 4$ ULA, $\omega(0) = [1, 0, 0, 0]$). The Maximin Algorithm Used a Convergence Parameter of $\alpha = 0.2$. (b) Final SINR in dB Under Frequency Offset and Friendly Interference ($\mathrm{M} = 4$ ULA, $\omega(0) = [1, 0, 0, 0]$). The Maximin Algorithm Used a Convergence Parameter of $\alpha = 0.2$

Δf	0	$1/(5\mathrm{T_b})$	$1/(2\mathrm{T_b})$	$1/(1.5\mathrm{T_b})$	$1/\mathrm{T_b}$	$1/(0.5\mathrm{T_b})$
Maximin	24.71 dB	23.97 dB	23.95 dB	22.15 dB	11.10 dB	0.0024 dB
Maximin SVD	24.72 dB	24.13 dB	24.11 dB	22.30 dB	14.55 dB	7.29 dB
Mostle	24.84 dB	24.29 dB	24.25 dB	22.50 dB	15.17 dB	7.98 dB

(a)

Δf	0	$1/(5\mathrm{T_b})$	$1/(2\mathrm{T_b})$	$1/(1.5\mathrm{T_b})$	$1/\mathrm{T_b}$	$1/(0.5\mathrm{T_b})$
Maximin	1.02 dB	0.93 dB	0.96 dB	1.00 dB	0.97 dB	0.96 dB
Maximin SVD	13.50dB	13.20 dB	13.05 dB	12.85 dB	11.38 dB	7.25 dB
Mostle	25.24dB	24.70 dB	24.46 dB	22.78 dB	16.41 dB	9.50 dB

(b)

TABLE V
(a) Number of Iterations to Reach SINR Within ±0.5 dB From Final SINR Under Frequency Offset and Tone Interference ($\mathrm{M} = 4$ Square, $\Delta\mathrm{f} = 1/(5\mathrm{T_b})$). The Maximin Algorithm Used a Convergence Parameter of $\alpha = 0.2$. (b) Number of Iterations to Reach SINR Within ±0.5 dB From Final SINR Under Frequency Offset and Friendly Interference ($\mathrm{M} = 4$ Square, $\Delta\mathrm{f} = 1/(5\mathrm{T_b})$). The Maximin Algorithm Used a Convergence Parameter of $\alpha = 0.2$

ω(0)	[1,0,0,0]	[1,1,1,1]
Maximin	149	7
Maximin SVD	44	34
Mostle	44	31

(a)

ω(0)	[1,0,0,0]	[1,1,1,1]
Maximin	20	35
Maximin SVD	1503 (2800 Iterations/Trial)	176
Mostle	80	41

(b)

TABLE VI
(a) Final SINR in dB Versus M When Input SNR $= 20$ dB Under Tone Interference (ULA, $\Delta\mathrm{f} = 0$). The Maximin Algorithm Used a Convergence Parameter of $\alpha = 0.2$. (b) Final SINR in dB Versus M When Input SNR $= 20$ dB Under Friendly Interference (ULA, $\Delta\mathrm{f} = 0$). The Maximin Algorithm Used a Convergence Parameter of $\alpha = 0.2$

M	2	4	6	8
Maximin	3.78	24.71	26.04	27.69
Maximin SVD	13.39	24.72	26.74	26.41
Mostle	17.38	24.84	26.49	26.19

(a)

M	2	4	6	8
Maximin	0.69	1.02	0.89	1.0249
Maximin SVD	3.73	13.50	25.97	26.048
Mostle	1.30	25.24	26.51	26.602

(b)

low when $\Delta\mathrm{f}$ is larger than $1/(1.5\mathrm{T_b})$. The maximin algorithm is unusable, and MOSTLE is the most robust among three schemes. Also, we observe from Table IV(b) that both the maximin and maximin with SVD algorithms fail when friendly interference and frequency offset are simultaneously active. However, MOSTLE shows robust performance even under these worse environments.

Table V(a) and (b) lists the average number of iterations to reach an SINR the first time within ±0.5 dB from the final SINR, under tone and friendly interference, respectively. We assumed two initial weights $\omega(0) = [1, 0, 0, 0]$ and $\omega(0) = [1, 1, 1, 1]$, a square array of $\mathrm{M} = 4$ elements, an AWGN channel, input SNR $= 20$ dB, and $\Delta\mathrm{f} = 1/(5\mathrm{T_b})$. The maximin algorithm used a convergence parameter of $\alpha = 0.2$. We observe from Table V(a) and (b) that the convergence speed of maximin can be the fastest among three schemes under tone interference when $\omega(0)$ is [1,1,1,1]. However, its final SINR values are unacceptable under a friendly interference environment, even if $\omega(0) = [1, 1, 1, 1]$ is used. In other words, the final SINRs of the maximin algorithm are around 1 dB and close to those shown in Fig. 5 that were obtained with $\omega(0) = [1, 0, 0, 0]$. Note from Table V(b) that the maximin with SVD algorithm shows the worst convergence speed among three schemes under a friendly interference environment, and its corresponding average number of iterations to reach a SINR within ±0.5 dB from the final SINR is 1503, when $\omega(0) = [1, 0, 0, 0]$ was used.

Table VI(a) and (b) lists the final SINR in decibels for various antenna elements M when input SNR is 20 dB under tone and friendly interference, respectively, assuming a ULA of M elements, an AWGN channel, and $\Delta\mathrm{f} = 0$. The maximin algorithm used a convergence parameter of $\alpha = 0.2$. We observe from Table VI(a) that performance of all three schemes improves as

TABLE VII
APPROXIMATE OPERATION COUNT-REAL FLOPs

CRITERION	PROPOSED SCHEME	MAXIMIN W/ SVD	MAXIMIN
SVD($\boldsymbol{X}$)	12M^3	12M^3	
SVD($\boldsymbol{N}$)	12M^3	12M^3	
$R_{ss}(k)\omega(k)$	2M^2	2M^2	mM
$R_{nn}(k)\omega(k)$	2M^2	2M^2	mM
$\hat{P}_s(k)$	M	M	m
$\hat{P}_n(k)$	M	M	m
$\mu(k)$	2M^2+4M		
TOTAL	**24M^3+6M^2+6M**	**24M^3+4M^2+2M**	**2mM+2m**

M increases under tone interference. In addition, we observe from Table VI(b) that the maximin algorithm fails under friendly interference, even if M increases. However, performance of the MOSTLE and maximin with SVD algorithms improves as M increases, even under friendly interference.

The proposed scheme, the maximin with SVD, and the maximin require real floating-point operations (FLOPs) proportional to $24\mathrm{M}^3 + 6\mathrm{M}^2 + 6\mathrm{M}$, $24\mathrm{M}^3 + 4\mathrm{M}^2 + 2\mathrm{M}$, and $2\mathrm{mM} + 2\mathrm{m}$ per iteration, respectively. Table VI in Appendix B presents comparisons of the computational complexity in detail. If $M = 4$ and $m = 100$, then the proposed scheme, maximin with SVD, and maximin require real FLOPs of 1656, 1608, and 1000 in total per iteration, respectively. Therefore, the proposed scheme requires real FLOPs computation comparable to that of maximin when $\mathrm{M} \leq 4$. Though computationally intensive per iteration for $\mathrm{M} \gg 4$, our technique ensures that communication is possible, irrespective of the scenario of operation. In the course of simulations, we also noted that the proposed technique is robust even when the interferences are spaced very close to the desired user and are friendly. We refer the interested reader to [5] for further results.

For a tradeoff analysis between performance and complexity, first, the authors found the number of iterations to reach a SINR the first time within ±0.5 dB from the final SINR for each of three algorithms. They are listed in Table V(a) and (b). This is used for counting the total number of FLOPs to reach a steady-state SINR value, which is the number of iterations multiplied by the number of FLOPs per iteration. The number of FLOPs per iteration can be calculated from Table VII. Then, the authors tried to plot the total number of FLOPs versus the number of antenna elements M. However, they realized that this comparative analysis does not fairly indicate a tradeoff between performance and complexity because each algorithm yields different SINR values, i.e., different performance. For example, as indicated from Table VI(b), when $\mathrm{M} = 4$, the final SINR values are 1.02, 13.50, and 25.24 dB for the maximin, maximin with SVD, and MOSTLE algorithm, respectively. The maximin algorithm cannot be used in this situation because its final SINR is so low. Even with an infinite number of FLOPs, the maximin algorithm will not reach the MOSTLE's final SINR under this environment. Therefore, it is difficult or impossible to compare the total FLOPs of maximin with that of MOSTLE at the same performance.

V. CONCLUSIONS

We presented a technique that enhances the performance of the maximin algorithm in the presence of *friendly* interferences and frequency offsets. We proposed a method to evaluate the adaptation parameter, also referred to as step length, using a maximization criterion. We observed in the simulation results that our scheme performs as well as the maximin algorithm in the presence of tone interferences. We also observed that the proposed technique enhances the performance of the maximin in the presence of friendly interferences and frequency offsets, both in AWGN and fading channels. We found that our technique is useful for multiple-access FH systems, even though it incurs higher computational costs.

APPENDIX A

We list the values of A_q, B_q, and C_q for $q = 1, 2$ of (14) as

$$\begin{aligned} A_q &= \hat{\rho}^2(k) z^\dagger(k) R_q z(k) \\ B_q &= \hat{\rho}(k) \left(z^\dagger(k) R_q \omega(k) + \omega^\dagger(k) R_q z(k) \right) \\ C_q &= \omega^\dagger(k) R_q \omega(k) \end{aligned} \tag{26}$$

where $R_1 = \hat{R}_{ss}$ and $R_2 = \hat{R}_{nn}$, the signal and noise autocovariance matrices, and

$$z(k) = \nabla_\omega \hat{\rho}(\mathrm{k}) = \left[\frac{\hat{R}_{ss}(k)\omega(k)}{\hat{P}_s(k)} - \frac{\hat{R}_{nn}(k)\omega(k)}{\hat{P}_n(k)} \right]. \tag{27}$$

Setting the derivative of (14) with respect to $\mu(k)$ to zero, we obtain

$$A\mu^2(k) + B\mu(k) + C = 0 \tag{28}$$

where

$$\begin{aligned} A &= A_1 B_2 - A_2 B_1, \quad B = A_1 C_2 - A_2 C_1 \\ C &= B_1 C_2 - B_2 C_1. \end{aligned} \tag{29}$$

Difficulties With Shor-Type Array Algorithms

We agree with Compton [7, pp. 63–64] regarding the difficulties with Shor-type array algorithms. Though Torrieri and Bakru [2] had successfully implemented the signal separator box for an FH system, the mathematical analysis of the behavior of such a scheme is still an open question, largely due to the nature of cost function.

Compton points out that the differential equation governing the weight vector is highly nonlinear, and hence, it is difficult to deduce general results about the solutions. However, he points out that the final weight vector is within a constant multiplier of the optimal solution. We present the results below.

We first show that the cost function ρ of (8) attains a maximum. We note that the frequency-hopping signal consists of pulses transmitted at many different frequencies, whereas the frequency channel associated with a single pulse is usually narrow. For a narrow-band desired signal, the vector of sampled complex envelopes is well approximated by

$$s(i) = s_1(i)a_0 \tag{30}$$

where $s_1(i)$ is the sample of complex envelope at antenna 1, which serves as the reference, and a_o is the array response vector (also referred to as the steering vector). Substitution of (30) in (5) yields

$$R_{ss} = E\left[|s_1(i)|^2\right] a_0^* \cdot a_0^T. \tag{31}$$

Observe that $R_{nn} = R_{nn}^{1/2} R_{nn}^{1/2}$, since it is a symmetric positive definite matrix. Using this observation and (31), we rewrite (8) as

$$\rho = E\left[|s_1(i)|^2\right] \cdot \frac{\omega^\dagger a_0^* \cdot a_0^T \omega}{\omega^\dagger R_{nn}^{1/2} R_{nn}^{1/2} \omega}. \tag{32}$$

Letting

$$y = R_{nn}^{1/2}\omega \tag{33a}$$

and

$$u = R_{nn}^{-1/2} a_0^* \tag{33b}$$

we have

$$\rho = E\left[|s_1(i)|^2\right] \cdot \frac{\|y \cdot u\|}{y^\dagger y} \leq \Lambda \|y\|_2 \cdot \|u\|_2 \tag{34}$$

by Schwarz's inequality, and Λ is the proportionality constant.

Hence, the maximum of ρ is reached only when $y = \kappa u$, where κ is a constant of proportionality. Thus, the optimum weight vector is

$$\omega_{opt} = \kappa R_{nn}^{-1} a_0^*. \tag{35}$$

Appendix B

We now present comparisons of the computational operations for the proposed algorithm, the maximin, and the maximin with SVD. We present the dominant operation of matrix calculations, namely, the multiplication count, since addition is relatively less expensive. In this discussion, we calculate the real floating operations (FLOPs). The complex FLOPs can be obtained by dividing these numbers by four. In all these calculations, we consider a typical weight vector update (iteration), say, the kth. Table I gives an array of $M = 4$ elements and the number of samples per update $m = 100$.

Several versions of the SVD algorithm exploit the specific structure of the matrices to reduce the computation load. We follow [8] to provide an estimate of the operation count based on a standard SVD algorithm. The approximate cost for computing the left singular vectors and singular values of X and $R_{xx}(k)$ is $12M^3$ and $\hat{R}_{XX}(k)\omega(k) = XX^\dagger\omega(k) = S_X \cdot \sum_X^2 \cdot S_X^\dagger \omega(k)$ is $2M^2$, noting that $\sum_X^2$ is a diagonal matrix. On the other hand, $\hat{P}_s(k) = \omega^\dagger(k) R_{ss}(k)\omega(k)$ needs M operations.

The evaluation of $\mu(k)$ involves the quadratic products A_q, B_q, and C_q, $q = 1, 2$ (see Appendix A). The cost for computing A_q and B_q is $M^2 + M$ and M, respectively. Hence, the total cost is $2M^2 + 4M$. By definition, $C_1 = \hat{P}_s(k)$ and $C_2 = \hat{P}_n(k)$ were computed earlier. The formula for $\mu(k)$ does not involve any matrix-vector products. Hence, the operation count is negligible and shall not be included.

The proposed scheme is numerically inexpensive when the number of antenna array of elements is small, such as $\mathrm{M} \leq 4$, but expensive when $\mathrm{M} \gg 4$, compared to maximin per iteration. Operation costs can be greatly reduced by implementing the SVD algorithm in an efficient way. Nevertheless, with a tradeoff in computational complexity, we enhance the robustness of the communication link in a multiple-access scenario.

References

[1] S. W. W. Shor, "Adaptive technique to discriminate against coherent noise in a narrow-band system," *J. Acoust. Soc. Amer.*, vol. 39, no. 1, pp. 74–78, Jan. 1966.

[2] D. Torrieri and K. Bakhru, The maximin algorithm for adaptive arrays and frequency hopping systems U. S. Army Research Laboratory, Adelphi, MD, Tech. Rep. TR-2026, Dec. 1999.

[3] J. G. Proakis*, Digital Communications*, 3rd ed. New York: McGraw-Hill, 1995.

[4] S. S. Rao*, Optimization Theory and Applications*, 2nd ed. New Delhi: Wiley Eastern, 1991.

[5] R. D. Balakrishnan, B. S. Nugroho, and H. M. Kwon, "Modified maximin adaptive array algorithm for FH systems under fading situations," in *MILCOM 2002*, Anaheim, CA, Oct. 2002.

[6] K. Bakhru and D. Torrieri, "The maximin algorithm for adaptive arrays and frequency-hopping communications," *IEEE Trans. Antennas Propag.*, vol. AP-32, pp. 919–928, Sep. 1984.

[7] R. T. Compton*, Adaptive Antennas: Concepts and Performance*. New York: Prentice-Hall, 1988.

[8] G. H. Golub and C. F. Van Loan*, Matrix Computations*, 3rd ed. New York: Johns Hopkins Univ. Press, 1996.

Raja D. Balakrishnan received the B.E. degree in electronics and communication engineering (with first-class honors) from Alagappa Chettiar College of Engineering and Technology, Karaikudi, India, in 1995 and the M.S. degree in electrical and computer engineering from Wichita State University, Wichita, KS, in 1999. He is currently pursuing the doctoral degree in mathematics.

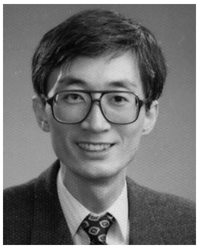
His doctoral work is focusing on ergodic theory and flows for dynamical systems. His areas of interest include array signal processing, information theory, numerical algorithms, differential geometry, algebraic geometry, and category theory.

Hyuck M. Kwon (S'82–M'84–SM'96) was born in Korea on May 9, 1953. He received the B.S. and M.S. degrees in electrical engineering from Seoul National University, Seoul, Korea, in 1978 and 1980, respectively, and the Ph.D. degree in computer, information, and control engineering from the University of Michigan at Ann Arbor, in 1984.

From 1985 to 1989, he was with the University of Wisconsin, Milwaukee, as an Assistant Professor in Electrical Engineering and Computer Science Department. From 1989 to 1993, he was with the Lockheed Engineering and Sciences Company, Houston, TX, as a Principal Engineer, working for NASA Space Shuttle and Space Station satellite communication systems. Since 1993, he has been with the Electrical and Computer Engineering Department, Wichita State University, Wichita, KS, where he is now a full Professor. In addition, he held several visiting and consulting positions at communication system industries, was a Visiting Associate Professor at Texas A&M University, College Station, in 1997, and a Visiting Professor at KAIST, Daejeon, Korea, in 2005. His current research interests are in wireless, CDMA/FH spread spectrum, smart antenna, MIMO, and OFDMA communication systems.

Dong-Hyeuk Yang received the B.S.E.E. degree in electrical engineering from Korean Air Force Academy, Chung-Ju, Korea, in 1997. He received the M.S. degree from the Department of Electrical and Computer Engineering, Wichita State University, Wichita, KS, in 2006.

From 1997 to 2001, he was with the 30th Air Defense Wing. From 2001 to 2002, he was with the Republic of Korea Air Force (ROKAF) APG as an Action Officer in charge of an international business. Since 2002, he has been with ROKAF Headquarters, Daejeon, Korea, as an Action Officer of EW. His current interests include smart antenna techniques, weapon simulations, military satellite, and data link communications.



Yong H. Lee (S'81–M'84–SM'98) was born in Seoul, Korea, on July 12, 1955. He received the B.S. and M.S. degrees from Seoul National University, Seoul, Korea, in 1978 and 1980, respectively, and the Ph.D. degree from the University of Pennsylvania, Philadelphia, in 1984, all in electrical engineering.

From 1984 to 1988, he was an Assistant Professor with the Department of Electrical and Computer Engineering, State University of New York, Buffalo. Since 1989, he has been with the Department of Electrical Engineering, KAIST, where he is currently a Professor and the Dean of the Engineering College. His research activities are in the area of communication signal processing, which includes synchronization, estimation, detection, interference cancellation, and resource allocation for CDMA, TDMA, and OFDM systems. He is also interested in design and implementation of transceivers.