

Distributed Incremental Interference Avoidance

Jasvinder Singh & Christopher Rose
WINLAB at Rutgers University
Piscataway, New Jersey 08854
Email: jasingh/crose@winlab.rutgers.edu

Abstract—We investigate practical methods of distributed interference avoidance where users iteratively adapt their codewords in response to global feedback from the receiver. In turn, the receiver adaptively tracks user codewords and offers a reasonable alternative to feeding back codewords. We introduce variants of standard interference avoidance procedures which produce more easily tracked incremental codewords and study the response of the system to abrupt changes in the interference background as might be encountered in a practical system. Furthermore, the methods we propose are strongly reminiscent of adaptive equalization for which a large body of knowledge and hardware expertise exist.

I. INTRODUCTION

Interference avoidance (IA) algorithms are motivated by emerging programmable radio technology [1] which will enable transceivers to improve transmission/reception methods on the fly to suit channel conditions. In a typical multiuser environment, iterative and greedy application of IA for each user results in overall improvement in performance. We envision such methods as especially useful in unlicensed bands where central control is absent or difficult to implement.

The basic idea of IA is simple. Each user waveform is represented as a linear combination of orthonormal basis functions which span the signal space. The set of real valued coefficients used to represent the waveform is a *codeword* and each user changes its codeword greedily to improve *SINR* for whatever linear receiver method it is using (i.e., matched or MMSE filter). We find it most useful to think in terms of the inverse *SINR*, given by the Rayleigh quotient [2, pp. 253] as

$$\chi_k = \frac{\mathbf{s}_k^\top \mathbf{R}_k \mathbf{s}_k}{\mathbf{s}_k^\top \mathbf{s}_k} \quad (1)$$

where \mathbf{R}_k is the interference plus noise covariance seen by codeword k at the receiver. Since it is known that the Rayleigh quotient of a matrix is minimized by the “minimum eigenvector” associated with the minimum eigenvalue, we can see that *SINR* for user k will be maximized if codeword \mathbf{s}_k is replaced by the minimum eigenvector of \mathbf{R}_k . Moving toward a state where all codewords are simultaneously minimum eigenvectors of their respective covariances is the overall goal of IA – and one which is achieved by iterative application of a variety of codeword update procedures [3], [4], [5], [6], [7], [8] or in a finite number of steps using centralized procedures [9], [10], [11]. There has also been some recent work on easier-to-implement binary approximations to WBE sets [12].

Strangely enough, it has proven difficult to prove *optimal codeword ensemble* convergence in general of IA algorithms.

Rose has proven convergence for a variant of minimum eigenvector IA [5]. Anigstein and Anantharam have proven convergence for a stochastic version of the MMSE IA algorithm [6]. The arguments and methods used in both papers are not terribly simple. Nonetheless, never has IA been observed to *not* converge to an optimal codeword ensemble numerically [3], [4], [13], [14], [15].

Regardless of which distributed procedure is used, codewords must either be calculated at the receiver and fed back to the transmitters, or the receiver must track codeword changes initiated in a distributed fashion by the transmitters. Since feeding back full codewords to each user can be onerous in terms of the bandwidth required [16], [17], here we consider the second more practical scenario where users know their channels – here, only gain for the time-invariant white noise case – the receiver covariance or the received vector is broadcast by the receiver and users employ some IA procedure to improve their codewords. Given the empirical evidence, we will not worry about detailed optimal codeword ensemble convergence proofs as in [5], [6]. Rather, we will take on faith that such distributed iterative procedures converge and examine systems level issues such as speed of convergence and ease of implementation.

Since the receiver must adapt to codeword changes, and the only information the receiver has are user transmissions, we note that IA algorithms such as the MMSE [3] or eigen algorithm [4] can cause abrupt changes in codewords which might be difficult for receivers to track without disruption of associated data streams. To minimize such disruption we introduce the following two schemes which allow only incremental adjustments to codewords:

- **Lagged IA:** codewords are adjusted in the direction of the optimal codeword.
- **Gradient Descent IA:** codewords are adjusted to most rapidly reduce the inverse *SINR*.

Based on these methods, we propose a simple structure for practical distributed IA. Overall, the method is reminiscent of adaptive equalization and appears robust to reasonably abrupt changes in the interference environment as well as the amount of broadcast feedback provided by the transmitter.

II. SYSTEM MODEL

We consider a system with block diagram as shown in FIGURE 1. The users/transmitters modulate the information to be transmitted using their codewords. The receiver uses a separate filter for each user which acts on the received

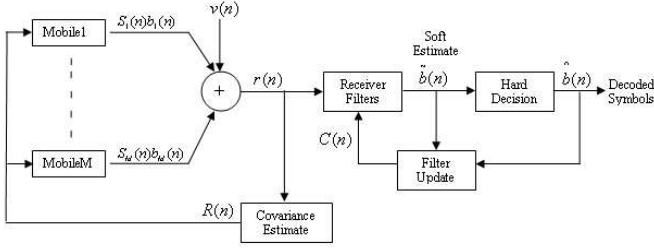


Fig. 1. System Model

signal and produces an estimate of the transmitted symbol for that user. The receiver has no *a priori* information about the transmitter codewords and starts with randomly selected filter coefficients. The transmitters send a training sequence with which the receiver iteratively refines the receiver filters based on a typical error minimization criteria. During the training phase, the system equations are,

$$\mathbf{r}(n) = \mathbf{S}(n)\mathbf{b}(n) + \mathbf{v}(n) \quad (2)$$

$$\mathbf{C}(n+1) = f[\mathbf{C}(n), \mathbf{r}(n), \mathbf{b}(n)] \quad (3)$$

where, n is the time index and $\mathbf{S}(n) = [\mathbf{s}_1(n), \mathbf{s}_2(n), \dots, \mathbf{s}_M(n)]$ is the codeword matrix $[L \times M]$ whose columns are the unit norm user codewords. $\mathbf{C}(n) = [\mathbf{c}_1(n), \mathbf{c}_2(n), \dots, \mathbf{c}_M(n)]$ is the receiver filter matrix $[L \times M]$ whose columns are receiver filters corresponding to the transmitter codewords. $\mathbf{r}(n)$ is the received signal vector at the receiver $[L \times 1]$. $\mathbf{b}(n)$ is the vector containing symbols sent by each user $[1 \times M]$. $\mathbf{v}(n)$ is assumed zero mean and white with covariance $\mathbf{K}_v(n, k) = N_0 \mathbf{I} \delta_{nk}$. $f[\cdot]$ is the receiver filter update scheme used.

After the training phase, the receiver measures the received signal covariance \mathbf{R} and broadcasts it to all users periodically. Alternately, and perhaps more parsimoniously, the receiver could periodically broadcast the received vector $\mathbf{r}(n)$ and let each user construct receiver covariance estimates. We assume that each user knows its channel, so the feed back can be used by the transmitters to steer transmitted codewords toward higher SINR. The receiver decodes the symbols sent by the users and continues updating \mathbf{C} as it did during training but now assuming that the decoded symbols are correct. This is exactly analogous to the operation of a typical adaptive equalizer [18].

The system equations are,

$$\mathbf{r}(n) = \mathbf{S}(n)\mathbf{b}(n) + \mathbf{v}(n) \quad (4)$$

$$\hat{\mathbf{b}}(n) = \text{sgn}[\tilde{\mathbf{b}}(n)] = \text{sgn}[\mathbf{r}(n)^\top \mathbf{C}(n)] \quad (5)$$

$$\mathbf{C}(n+1) = f[\mathbf{C}(n), \mathbf{r}(n), \mathbf{b}(n)] \quad (6)$$

where $\tilde{\mathbf{b}}(n)$ is the soft estimate of the transmitted symbol vector $\mathbf{b}(n)$ and $\hat{\mathbf{b}}(n)$ is the corresponding hard estimate. We define the received covariance as

$$\mathbf{R}(n) = E[\mathbf{r}(n)\mathbf{r}^\top(n)] \quad (7)$$

and the covariance seen by a particular user k as

$$\mathbf{R}_k(n) = \mathbf{R}(n) - \mathbf{s}_k(n)\mathbf{s}_k(n)^\top \quad (8)$$

We use a standard covariance estimate

$$\hat{\mathbf{R}}(n) = \hat{\mathbf{R}}(n-1) + (1-\xi)(\mathbf{r}(n)\mathbf{r}(n)^\top - \hat{\mathbf{R}}(n-1)) \quad (9)$$

where ξ is a "forgetting factor" [19]. Likewise the individual user k covariance estimate is defined as

$$\hat{\mathbf{R}}_k(n) = \hat{\mathbf{R}}(n) - \mathbf{s}_k(n)\mathbf{s}_k(n)^\top \quad (10)$$

Finally, the codeword update equation is

$$\mathbf{s}_k(n+1) = g[\mathbf{s}_k(n), \mathbf{R}_k(n)] \quad k \in 1, 2, \dots, M \quad (11)$$

where $g[\cdot]$ is the codeword steering scheme used at the transmitters.

III. CODEWORD STEERING SCHEMES

A. Lagged IA

As seen previously, the optimal codeword for user k is given by the eigenvector corresponding to the minimum eigenvalue of the channel interference matrix \mathbf{R}_k . Let us denote this eigenvector by \mathbf{s}_k^* . Assuming codewords of other users remain fixed, user k can increase its SINR by steering its codeword toward \mathbf{s}_k^* using the iteration

$$\mathbf{s}_k(n+1) = \frac{\alpha \mathbf{s}_k(n) + m \beta \mathbf{s}_k^*}{|\alpha \mathbf{s}_k(n) + m \beta \mathbf{s}_k^*|} \quad (12)$$

$\alpha, \beta \in \mathfrak{R}^+$. We have defined $m = \text{sgn}[\rho_k(n)]$ with $\rho_k(n) = \mathbf{s}_k^\top(n) \mathbf{s}_k^*$ and note that $|\rho_k(n)| \leq 1$ since $|\mathbf{s}_k^*| = |\mathbf{s}_k(n)| = 1$.

Equation (12) has a simple and intuitive geometric meaning: $\mathbf{s}_k(n+1)$ represents a step toward the closest optimal codeword $m \mathbf{s}_k^*$ along the arc joining $m \mathbf{s}_k^*$ and $\mathbf{s}_k(n)$. That is, $\mathbf{s}_k(n)$ and $m \mathbf{s}_k^*$ share the same half plane. Formally, we have

Theorem 1: User k 's SINR will increase under the iteration of equation (12) where \mathbf{s}_k^* is a minimum eigenvector of \mathbf{R}_k .

Proof: Theorem 1

\mathbf{R}_k is a covariance matrix and therefore positive-semidefinite with orthonormal eigenvectors which span the signal space. Let $[(\lambda_1, x_1), (\lambda_2, x_2), \dots, (\lambda_L, x_L)]$ be the eigenvalues and eigenvectors of \mathbf{R}_k such that $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_L$ ($\mathbf{s}_k^* = x_1$). Thus

$$\mathbf{s}_k^\top(n) \mathbf{R}_k \mathbf{s}_k(n) \geq (\mathbf{s}_k^*)^\top \mathbf{R}_k \mathbf{s}_k^* = \lambda_1 \quad (13)$$

since $|\mathbf{s}_k(n)| = |\mathbf{s}_k^*| = 1$. The change in inverse SINR is

$$\begin{aligned} \Delta \chi_k &= \mathbf{s}_k^\top(n+1) \mathbf{R}_k \mathbf{s}_k(n+1) - \mathbf{s}_k^\top(n) \mathbf{R}_k \mathbf{s}_k(n) \\ &= \frac{(\alpha \mathbf{s}_k(n) + m \beta \mathbf{s}_k^*)^\top}{|\alpha \mathbf{s}_k(n) + m \beta \mathbf{s}_k^*|} \mathbf{R}_k \frac{(\alpha \mathbf{s}_k(n) + m \beta \mathbf{s}_k^*)}{|\alpha \mathbf{s}_k(n) + m \beta \mathbf{s}_k^*|} - \mathbf{s}_k^\top(n) \mathbf{R}_k \mathbf{s}_k(n) \end{aligned} \quad (14)$$

We then note that

$$|\alpha \mathbf{s}_k(n) + m \beta \mathbf{s}_k^*|^2 = \alpha^2 + \beta^2 + 2\alpha\beta|\rho(n)| = \kappa^2 \quad (15)$$

so that

$$\begin{aligned} \kappa^2 \Delta \chi_k &= \alpha^2 \mathbf{s}_k^\top(n) \mathbf{R}_k \mathbf{s}_k(n) - \kappa^2 \mathbf{s}_k^\top(n) \mathbf{R}_k \mathbf{s}_k(n) \\ &+ m \alpha \beta \mathbf{s}_k^\top(n) \mathbf{R}_k \mathbf{s}_k^* + m \alpha \beta \mathbf{s}_k^{*T} \mathbf{R}_k \mathbf{s}_k(n) + \beta^2 \mathbf{s}_k^{*T} \mathbf{R}_k \mathbf{s}_k^* \\ &= (\alpha^2 - \kappa^2) \mathbf{s}_k^\top(n) \mathbf{R}_k \mathbf{s}_k(n) + \lambda_1 (\kappa^2 - \alpha^2) \\ &= (\kappa^2 - \alpha^2) (\lambda_1 - \mathbf{s}_k^\top(n) \mathbf{R}_k \mathbf{s}_k(n)) \end{aligned} \quad (16)$$

which for $\alpha, \beta > 0$ is always less than or equal to zero by equation (15) and equation (13) •

To ensure that codewords change incrementally we require $|\alpha| \gg |\beta|$. However, convergence of iterative IA requires a stronger condition than just SINR increase. Specifically, we must also have a decrease in the total squared correlation (TSC), defined as

$$\text{TSC}(n) = \text{Trace}[\mathbf{R}(n)\mathbf{R}(n)^\top] \quad (17)$$

where

$$\mathbf{R}(n) = \mathbf{R}_k(n) + \mathbf{s}_k(n)\mathbf{s}_k^\top(n) \quad (18)$$

to guarantee convergence [4], [20]. Thus, we require

$$\Delta\text{TSC} = \text{Trace}[\mathbf{R}(n+1)\mathbf{R}^\top(n+1)] - \text{Trace}[\mathbf{R}(n)\mathbf{R}^\top(n)] \leq 0 \quad (19)$$

Theorem 2: If the replacement of the k^{th} user codeword $\mathbf{s}_k(n)$ with $\mathbf{s}_k(n+1)$ decreases inverse SINR, then it also decreases TSC given that $\|\mathbf{s}_k(n)\| = \|\mathbf{s}_k(n+1)\| = 1$

Proof: Theorem 2 We can write TSC in the following manner,

$$\begin{aligned} \text{TSC}(n) &= \sum_{i=1}^M \sum_{j=1}^M (\mathbf{s}_i(n)^\top \mathbf{s}_j(n))^2 \\ &= (\mathbf{s}_k(n)^\top \mathbf{s}_k(n))^2 + \sum_{i \neq k} \sum_{j \neq k} (\mathbf{s}_i(n)^\top \mathbf{s}_j(n))^2 \\ &\quad + 2\mathbf{s}_k(n)^\top \sum_{j \neq k} (\mathbf{s}_j(n)\mathbf{s}_j(n)^\top) \mathbf{s}_k(n) \end{aligned} \quad (20)$$

Now, identifying that $\sum_{j \neq k} (\mathbf{s}_j(n)\mathbf{s}_j(n)^\top) = \mathbf{R}_k(n)$ and using $\|\mathbf{s}_k(n)\| = \|\mathbf{s}_k(n+1)\| = 1$, we can see that, $\Delta\text{TSC} = \text{TSC}(n+1) - \text{TSC}(n) = 2\Delta\chi_k$ and the result follows.

•

B. Gradient Descent

From above, the inverse SINR for the k^{th} user is

$$\chi_k = \frac{\mathbf{s}_k^\top \mathbf{R}_k \mathbf{s}_k}{\mathbf{s}_k^\top \mathbf{s}_k} \quad (21)$$

and its gradient with respect to the codeword components $\{s_{kj}\}$ is

$$\nabla \chi_k = \frac{2[\mathbf{s}_k^\top \mathbf{R}_k \mathbf{s}_k \mathbf{s}_k - (\mathbf{s}_k^\top \mathbf{R}_k \mathbf{s}_k) \mathbf{s}_k]}{(\mathbf{s}_k^\top \mathbf{s}_k)^2} \quad (22)$$

Therefore, the iteration $\mathbf{s}_k(n+1) = \mathbf{s}_k(n) - v \nabla \chi_k$, with v a suitably small constant, increases SINR.

Now, even if we impose the unit power constraint on $\mathbf{s}_k(n+1)$ by normalization, SINR still increases because normalization doesn't change the value of χ_k . So, our iteration becomes,

$$\mathbf{s}_k(n+1) = \frac{\mathbf{s}_k(n) - v \nabla \chi_k}{\|\mathbf{s}_k(n) - v \nabla \chi_k\|} \quad (23)$$

Decreasing inverse SINR implies TSC decrease via Theorem 2, so convergence is guaranteed. We note that the gradient descent iteration does not explicitly require calculation of the minimum eigenvector – a computational advantage.

IV. THE RECEIVER FILTER

A. Matched Filter

For optimal codeword ensembles, the matched filter ($\mathbf{c}_i = \mathbf{s}_i$) is the optimal linear receiver [9]. The filter update equation can be obtained by using gradient descent to minimize $E[\mathbf{e}(n)^2]$, where

$$\mathbf{e}(n) = \mathbf{r}(n) - \mathbf{C}(n)\mathbf{b}(n) \quad (24)$$

so that the filter update equation is

$$\mathbf{C}(n+1) = \mathbf{C}(n) + \mu \mathbf{e}(n)\mathbf{b}(n)^\top \quad (25)$$

for some suitable constant μ . Note that after training, we use $\mathbf{b}(n) = \hat{\mathbf{b}}(n)$ and continue updating \mathbf{C} in the same manner.

B. MMSE Filter

The MMSE filter for the k^{th} user is defined as the vector \mathbf{c}_k which minimizes $E[e_k(n)^2]$ where $e_k(n)$ is now defined as,

$$e_k(n) = b_k(n) - \mathbf{r}(n)^\top \mathbf{c}_k(n) \quad (26)$$

Note that in the above equation, $\mathbf{r}(n)^\top \mathbf{c}_k(n)$ represents a soft estimate for $b_k(n)$ and hence $e_k(n)$ is the symbol estimation error for user k . The filter update equation in this case is

$$\mathbf{C}(n+1) = \mathbf{C}(n) + \mu \mathbf{e}(n)\mathbf{r}(n) \quad (27)$$

where,

$$\mathbf{e}(n) = \mathbf{b}(n) - \mathbf{r}(n)^\top \mathbf{C}(n) \quad (28)$$

After training, we use $\mathbf{b}(n) = \hat{\mathbf{b}}(n) = \text{sgn}[\tilde{\mathbf{b}}(n)]$ and continue updating \mathbf{C} in the same manner.

V. SIMULATIONS

A. Experiment 1

We first consider users with static codewords (no codeword adaptation) and examine filter acquisition (training) and subsequent tracking. FIGURE 2 shows the BER vs SNR curves for underloaded system 4 users in 12 dimensions and also draws a comparison between MMSE and matched filter codeword update schemes at the receiver. FIGURE 3 repeats the same experiment with an overloaded system (14 users in 12 dimensions). Note that the same transmitter codeword sets are used in both MMSE and matched filter schemes. As might be expected for unadapted codewords, MMSE filtering is superior since it mitigates the effects of any randomly high correlation between user codewords. Since codeword tracking and adaptation critically depend on low error rates, we abandoned use of the matched filter in favor of the MMSE filter for subsequent experiments. Of course, it is also worth mentioning that *any* linear filtering technique will eventually succumb to high error rates if enough users are present. For discrete user symbol alphabets (binary) this is easily seen since WBE signal sets can produce fewer signal constellation points than possible user symbol combinations. This issue is carefully treated in [21] where a case for nonlinear decoding (over multiple symbols) is made.

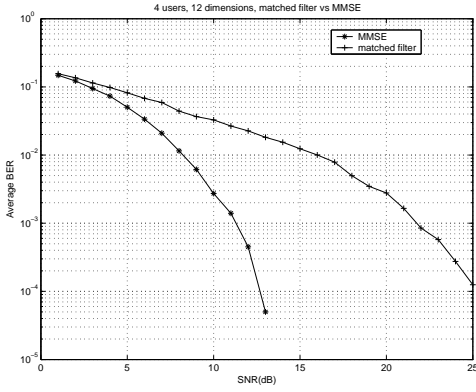


Fig. 2. BER vs SNR plot for 4 *static* users in 12 dimensions. $\mu = 0.001$.

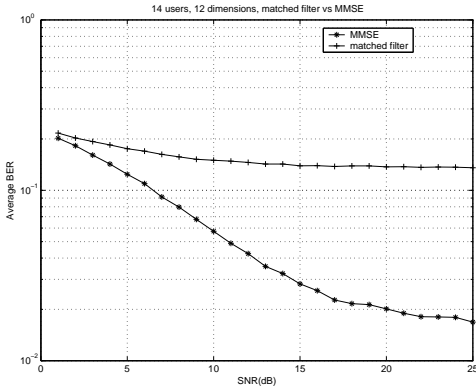


Fig. 3. BER vs SNR plot for 14 *static* users in 12 dimensions. $\mu = 0.001$.

B. Experiment 2

We then consider an overloaded system (14 users in 12 dimensions) and look at the codeword steering performance for both gradient descent as well as lagged IA. The value of ξ in equation (9) is chosen as 0.98. We seek minimum codeword convergence time without incurring high BER at the receiver and chose a nominal value for SNR as 20 dB since we expect such systems to be interference rather than noise limited. The initial codewords are chosen randomly as before. We vary the steering step size control parameters (v for gradient descent, α for lagged IA with β fixed) and measure the BER while the codewords are being adapted. If the steering step size is too large, receiver cannot track the codewords and experiences a high BER. On the other hand, if steering step size is too small, codewords converge very slowly to their optimal values and high correlation leads to high BER during the finite measurement window. By plotting BER versus step size, a suitable range for steering step sizes was found empirically: $10^{-3} < v < 10^{-2}$ and $40 < \alpha < 300$.

C. Experiment 3

We also considered the effect of adding static interference (in the form of a new user codeword) to the system after the user codewords have settled down to a WBE set through application of IA. First, we consider an underloaded system

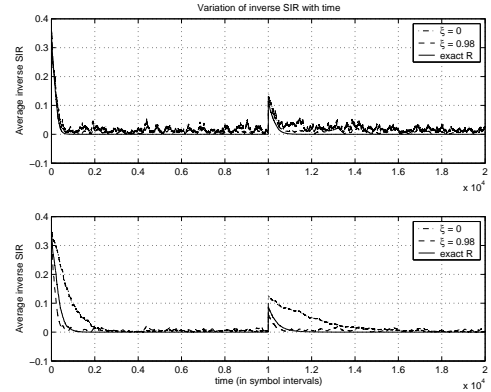


Fig. 4. Average inverse SIR variation for 4 users in 12 dimensions. New user added at 10000th bit. TOP: gradient descent ($v = 0.005$). BOTTOM: lagged IA ($\alpha = 100, \beta = 1$).

with 4 users in 12 dimensions. As before, the SNR for each user is chosen to be 20 dB. We used the following values for steering step control parameters: ($v = 0.005, \alpha = 100, \beta = 1$). FIGURE 4 shows the average inverse SIR variation with time. The first 10000 bit intervals comprise the post training interval where transmitters are adapting their codewords using IA. Since this is an underloaded system, we expect the inverse SIR value to become small (nearly orthogonal user codewords). At $t = 10001$, a user with a random but subsequently fixed codeword is introduced. From $t = 10001$ onward we see that IA reduces the average SIR to near zero within 500 symbol intervals for gradient descent and 1000 symbol intervals for lagged IA. FIGURE 4 also compares the performance for exact covariance feedback and for estimates of different qualities ($\xi = 0.98, 0$). Note that $\xi = 0$ corresponds to an instantaneous – and therefore highly volatile – estimate ie. $\mathbf{R}(n) = \mathbf{r}(n)\mathbf{r}(n)^T$.

FIGURE 5 shows the same plots for an overloaded system (14 users in 12 dimensions). For exactly known and average covariance before the introduction of static interference, the average inverse SIR value is approximately $0.1667 = \frac{14-12}{12}$, the theoretically optimal value associated with a Welch bound equality codeword set (see [4]). The performance of the instantaneous covariance feedback is slightly poorer, but not unreasonable.

Adding sudden interference does not greatly increase the average SIR and more importantly, does not greatly disrupt the data streams of other users – at least in as much as no retraining was required, even with the imprecise instantaneous covariance. Following the interference insult, the system quickly settles down to the theoretically minimum inverse SIR of $\frac{15-12}{12} = 0.25$ in the case of exact and averaged covariance feedback, and once again, performs a bit more poorly for instantaneous covariance feedback. We also plot the variance of inverse SIR among users vs time in FIGURE 6 which suggests that codeword adaptation rapidly equalizes user SIRs and indirectly corroborates convergence to approximately optimal codeword ensembles.

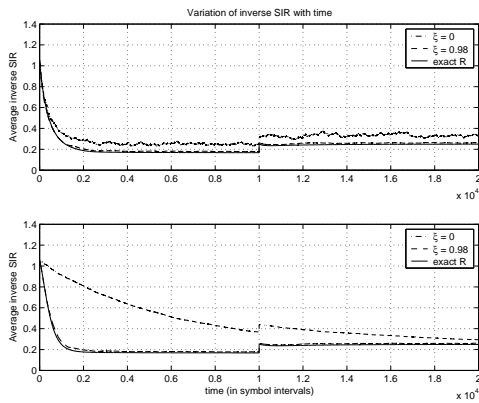


Fig. 5. Average inverse SIR variation for 14 users in 12 dimensions. New user added after 10000 bit intervals. Top plot shows the case when gradient descent ($\nu = 0.005$) was used while the bottom plot corresponds to lagged IA ($\alpha = 100, \beta = 1$).

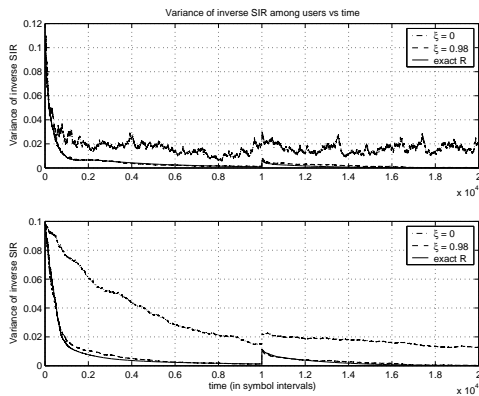


Fig. 6. Variance of inverse SIR for 4 users in 12 dimensions. New user added after 10000 bit intervals. Top plot shows the case when gradient descent ($\nu = 0.005$) was used while the bottom plot corresponds to lagged IA ($\alpha = 100, \beta = 1$).

VI. CONCLUSION

We have analyzed and simulated distributed interference avoidance (IA) based on covariance feedback broadcast from the receiver and incremental codeword changes by each user. The feedback could be a covariance matrix estimate from the receiver, or a sequence of received vectors $\mathbf{r}(n)$ to allow estimates to be constructed by each transmitter. The receiver tracks codeword changes by adapting the associated MMSE filters under a symbol error criterion. With perfect covariance feedback, from the perspective of codeword ensemble performance the distributed method is equivalent to centralized methods where codewords are computed by the receiver and distributed to transmitters. Lagged IA, however, shows a greater sensitivity than gradient descent IA to covariance uncertainty. Of course, no attempt was made here to optimize covariance estimates and in addition nonlinear joint decoding [21] was also not considered since we opted for simplicity. So perhaps impressive performance improvements are possible.

Since codeword feedback is onerous from a beacon channel bandwidth perspective [16], covariance feedback confers some

advantage in a practical setting. Furthermore, the codeword tracking and codeword update machinery employed are both strongly reminiscent of adaptive equalization for which a large body of work and hardware methods exist. Thus, covariance feedback IA could allow practical and inexpensive implementations with associated increases in wireless network capacities.

REFERENCES

- [1] J. Mitola. The software radio architecture. *IEEE Communications Magazine*, 33:26–38, May 1995.
- [2] Gilbert Strang. *Linear Algebra and Its Applications*. Academic Press, second edition, 1992.
- [3] S. Ulukus and R. D. Yates. Iterative construction of optimum signature sequence sets in synchronous CDMA systems. *IEEE Trans. Info. Theory*, 47(5):1989–1998, July 2001.
- [4] C. Rose, S. Ulukus, and R. Yates. Interference Avoidance in Wireless Systems. *IEEE Transactions on Wireless Communications*, 1(3):415–428, July 2002. (also available at www.winlab.rutgers.edu/~crose/papers/avoid17.ps).
- [5] C. Rose. CDMA Codeword Optimization: interference avoidance and convergence via class warfare. *IEEE Transactions on Information Theory*, 47(6):2368–2382, September 2001.
- [6] P. Anigstein and V. Anantharam. Ensuring convergence of the MMSE iteration for interference avoidance to the global optimum. *IEEE Transactions on Information Theory*, 49(4):873–885, April 2003.
- [7] P. Cotae. An Algorithm for Obtaining Welch Bound Equality Sequences for S-CDMA Systems. *AEÜ, International Journal of Electronics and Communications*, 55(2):95–99, 2001.
- [8] P. Cotae. Distributive iterative method for minimizing generalized total square correlation of CDMA systems. In *International Symposium on Information Theory*, 2003. paper no.0187, “La Sorbonne”, Paris, France.
- [9] P. Viswanath, V. Anantharam, and D. Tse. Optimal sequences, power control and capacity of spread-spectrum systems with multiuser receivers. *IEEE Transactions on Information Theory*, 45(6):1968–1983, 1999.
- [10] P. Viswanath and V. Anantharam. Optimal sequences and sum capacity of synchronous cdma systems. *IEEE Transactions on Information Theory*, 45(6):1984–1991, Sept 1999.
- [11] P. Viswanath and V. Anantharam. Optimal Sequences for CDMA with colored noise: A Schur-Saddle Function Property. *IEEE Transactions on Information Theory*, 48(6):1295–1318, June 2002.
- [12] G. N. Karystinos and D. A. Pados. New bounds on the total squared correlation and optimum design of DS-CDMA binary signature sets. *ieeecom*, 49:261–269, January 2003.
- [13] D. C. Popescu and C. Rose. Interference avoidance and multiaccess dispersive channels. In *35th Annual Asilomar Conference on Signals, Systems, and Computers*, Nov 4-7 2001. Pacific Grove, CA.
- [14] D.C Popescu and C. Rose. A new approach to multiple antenna systems. In *CISS'01*, March 2001. Baltimore, MD.
- [15] D. C. Popescu and C. Rose. Fading Channels and Interference Avoidance. In *Proc. 39th Allerton Conf. on Communication, Control, and Computing*, pages 1073–1074, Monticello, IL, September 2001. (available at <http://www.winlab.rutgers.edu/~cripop/papers>).
- [16] D. C. Popescu and C. Rose. Codeword Quantization for Interference Avoidance. In *Proc. 2000 Int. Conf. on Acoustics, Speech, and Signal Processing - ICASSP 2000*, volume 6, pages 3670–3673, June 2000. Istanbul.
- [17] W. Santipach and M. L. Honig. Signature Optimization for DS-CDMA with Limited Feedback. In *Int. Symposium on Spread Spectrum Systems and Applications*, September 2002. Prague, Czech Republic.
- [18] S. Haykin. *Communication Systems*. John Wiley & Sons, 1994.
- [19] S. Orfanidis. *Optimum Signal Processing*. Mc-Graw Hill, Newyork, 1988.
- [20] C. Rose. Cdma codeword optimization: interference avoidance and convergence via class warfare. *IEEE Transactions on Information Theory*, 47(7):2368–2382, September 2001.
- [21] A. Kapur and M. K. Varanasi. On the Limitation of Linear MMSE Detection and (Generalized) Welch Bound Equality Signals. In *Proc. of the 37th. Annual Conf. Inform. Sciences and Systems*, March 2003. Baltimore, MD.