Channel Equalization for STBC-Encoded Cooperative Transmissions with Asynchronous Transmitters

Xiaohua(Edward) Li, Fan Ng, Jui-Te Hwu, and Mo Chen
Department of Electrical and Computer Engineering
State University of New York at Binghamton
Binghamton, NY 13902
{xli,fngnone1,jhwu1,mchen0}@binghamton.edu

***Abstract*— STBC-encoded cooperative transmissions can potentially enhance the efficiency of distributed wireless networks. In this paper, we study their performance when the cooperative transmitters are not synchronized so that flat-fading channels become dispersive. Two channel equalization methods are proposed: the complex but optimal Viterbi equalizer and a more efficient linear-prediction-based equalizer. The latter is a blind equalizer with much improved robustness thanks to a special property of the distributed channels. The performance of the two methods are analyzed and simulated, which shows the usefulness of STBC-encoded cooperative transmissions in asynchronous environment.**

I. INTRODUCTION

Space-time coding and processing are powerful techniques for enhancing transmission efficiency of wireless networks. Among various widely investigated space-time techniques, space-time block codes (STBC) [1], [2] are especially promising because of their low computational complexity. Since diversity enhances transmission energy efficiency in fading environment, STBC would be desirable for mobile users in wireless networks such as ad hoc and sensor networks.

In order to take the benefits of STBC for distributed mobile networks, recently there have been great interests to investigate cooperative communications encoded by STBC. By exploiting the cooperation capability of multiple mobile users, STBC can still be used based on virtual instead of physical antenna array [3], [4].

So far, most existing researches on cooperative transmission assume perfect synchronization among cooperative users, which means that the users' timing, carrier frequency and propagation delay are identical [3]. Unfortunately, it is difficult, and in most cases impossible, to achieve perfect synchronization among distributed transmitters. This is even more a reality when low-cost, small-sized transmitters are used, such as tiny sensors [7].

Without perfect synchronization, channels become dispersive even in flat fading environment. Due to the transmitting/receiving pulse shaping filters, if the sampling time instants are not ideal, intersymbol interference (ISI) is introduced. This certainly brings performance degradation or performance loss in cooperative STBC. More important, asynchronism among the transmitters may break the orthogonal STBC signal structure, which makes most of the existing STBC decoders fail. Therefore, for the cooperative STBC, one of the major challenges is the synchronization among the distributed nodes.

The synchronization requirement among the transmitters is different from that between the transmitter and the receiver. For the former, usually hand-shaking has to be conducted among the transmitters. Therefore, the synchronization task becomes a cross-layer design problem.

We have recently addressed partially the problem of imperfect synchronization in [6]. An alternative method is using OFDM transmission [8]. One of the major problems for these schemes is that the guard interval greatly reduced bandwidth efficiency, especially when the channels are time-varying. As a matter of fact, channel is very likely time-varying with asynchronous transmitters due to the different carrier frequency drifting and Doppler shifting among the transmitters.

In this paper, we consider the complete asynchronous transmitters with STBC-encoded transmissions. However, we consider only the asynchronism caused by different transmission time instants and propagation delays among them. In this case, equalization techniques can be developed to detect symbols from the received asynchronous signals. The advantage is the reduced synchronization and cooperation overhead.

This paper is organized as follows. In Section II, we give the asynchronous transmission system model and discuss the difficulty of synchronization. Then in Section III, we develop the optimal Viterbi equalizer. A blind linear equalizer is developed in Section IV. Then simulations are given in Section V. Finally, conclusions are presented in Section VI.

II. SYSTEM MODEL AND THE SYNCHRONIZATION PROBLEM

Consider an ad hoc wireless network where a source node needs to transmit a data packet to a destination node through multi-hop relaying as shown in Fig. 1.

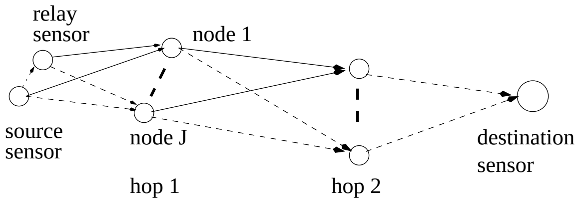


Fig. 1. Multi-hop ad hoc network model with cooperative transmissions.

A. Synchronization of transmitters is difficult

We consider the cooperative transmission among nodes 1 to J in hop 1. Consider first the passband signal. The passband signal to be transmitted by each transmitter has a general form $\mathrm{Re}[\sum_{n=-\infty}^{\infty} s_i(n)p_b(t-nT)e^{j2\pi f_c t}]$, where $s_i(n)$ is the complex signal transmitted within symbol interval $[nT,(n+1)T)$, $p_b(t)$ is the baseband pulse shaping filter, and f_c is the carrier frequency. After delaying with δ_i, the passband signal transmitted from the transmitter i, $1 \le i \le J$, is $\mathrm{Re}[\sum_{n=-\infty}^{\infty} s_i(n)p_b(t-nT-\delta_i)e^{j2\pi f_c(t-\delta_i)}]$.

We assume flat-fading propagation in this paper. The received passband signal at a receiver is

$$r_p(t) = \mathrm{Re}[\sum_{i=1}^{J} \alpha_i \sum_{n=-\infty}^{\infty} s_i(n) \\ p_b(t-nT-\delta_i-\tau_i)e^{j2\pi f_c(t-\delta_i-\tau_i)} + v_p(t)], \quad (1)$$

where α_i and τ_i are (complex) gains and delays of the propagation. We can adjust δ_i to compensate τ_i for carrier phase synchronization. But in general, such synchronization is impossible.

All transmitters should have identical clock, carrier frequency and symbol timing in order to directly use existing STBC-encoded transmission and decoding techniques. The challenge comes from the fact that the propagation delays, i.e., time required for the signal from the transmitter to reach the receiver, may be different among the transmitters. So do the carrier frequencies due to Doppler shifting when the transmitters and receivers are moving independently. As a result, synchronizing all transmitters to one receiver may increase asynchronism toward another receiver [6].

For example, one may synchronize cooperating nodes with an identical reference signal such as their common received signal from the previous transmitters. However, such synchronization can hardly be accurate due propagation delays, Doppler shifting as well as noise. In addition, different circuitry designs introduce different frequency/clock drifting and processing delay. As another example, one may use GPS for timing synchronization. However, there are still problems such as shadowed environment where GPS signals are not available.

One of the fundamental difficulties, however, comes from the traditional separate-layer network design methodology. The MAC-layer clock is usually limited to micro-second accuracy, not accurate enough for symbol-level transmission synchronization. More severely, it is not tightly coupled with the physical-layer transmission clock. MAC-layer can not obtain physical-layer timing, nor does physical-layer have MAC-layer timing, because their clocks are generated by different oscillators.

Perfect synchronization, even if available, is usually achieved at a fixed time instant only. Cooperating nodes still increase mismatch in frequency/timing on the long run because of the drifting of electronic parameters. Therefore, perfect synchronization requires at least tight coordination for frequent re-synchronization. Tight coordination requires frequent handshaking, which surely increases overhead. The increased overhead may overweigh cooperation benefits, which is especially critical for energy-limited sensor networks or bandwidth-limited networks.

If considering the multi-hop networks, then such a point-wise perfect synchronization bring a new issue. Even some cooperating nodes are synchronized at one time, if due to contention they can not be scheduled to transmit immediately, or due to packet collision they have to be scheduled for retransmission at some later time, the cooperating nodes may become asynchronous due to parameter drifting. Then some ways have to be found to re-synchronize them. This certainly causes more overhead, and may even causes scalability problem if not carefully designed. Therefore, instead of considering the more complex and resource consuming cross-layer scheduling, it may be more advantageous to consider directly the asynchronous transmissions in the physical-layer instead.

B. Signal models with asynchronous transmitters

To simplify the problem, we consider only the asynchronism in transmission time and propagation delays, which means that δ_i and τ_i are different for different transmitters.

Without loss of generality, we can demodulate (1) by $e^{-j2\pi f_c t}$ to obtain the continuous-time complex baseband signal

$$r_b(t) = \sum_{i=1}^{J} \alpha_i e^{-j\theta_i} \sum_{n=-\infty}^{\infty} s_i(n)p_b(t-nT-\delta_i-\tau_i) + v_b(t), \quad (2)$$

where $v_b(t)$ is the equivalent baseband noise, and the phase $\theta_i = 2\pi f_c(\delta_i+\tau_i)$.

If δ_i and τ_i are different among the transmitters, it is impossible to achieve timing synchronization. Then without loss of generality, we perform baseband sampling at time instant nT, which gives $x(n) \triangleq r_b(nT)$. The samples $x(n)$ can be written as

$$x(n) = \sum_{i=1}^{J} [\ h_i^*(0) \quad \cdots \quad h_i^*(L)\ ] \begin{bmatrix} s_i(n-d_i) \\ \vdots \\ s_i(n-d_i-L) \end{bmatrix} + v(n), \quad (3)$$

where $v(n)$ is the noise sample, and the channel coefficients are

$$h_i^*(m) = \alpha_i e^{-j\theta_i} p_b(mT+d_iT-\delta_i-\tau_i), \quad d_i \ge 0, \ \ 0 \le m \le L. \quad (4)$$

The baseband channel length L and coefficients $h_i^*(m)$ are determined by both fading and asynchronism.

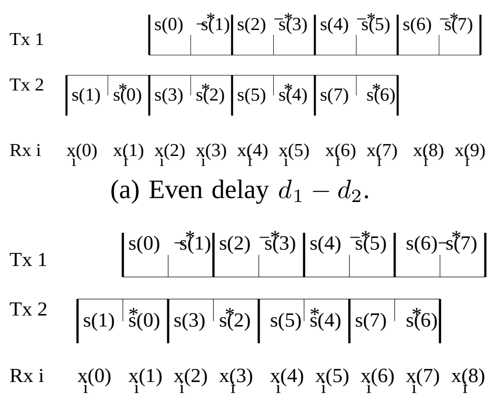
On the other hand, the dispersive channel model in this case is different from that due to multipath propagations. One of the major differences is that we can make the channel of one of the transmitters to be single tap by adjusting the sampling timing of the receiver.

III. Viterbi equalizer for asynchronous STBC

For simplicity, we consider the Alamouti's STBC with 2 transmitters and one receiver. Extension to more general STBC can be conducted similarly.

A. STBC with asynchronous transmitters

An illustration of the transmitted/received symbol sequences is shown in Fig. 2, depending on the delay difference between the signals from the two transmitters.



(a) Even delay $d_1 - d_2$.

(b) Odd delay $d_1 - d_2$.

Fig. 2. Illustration of Alamouti STBC encoded transmission with asynchronous delays among two transmitters.

Without loss of generality, assume that the first transmitter (Tx 1) advances the second one by d during transmission. The receiver can perform sampling with timing synchronized to that of Tx 2. Therefore, the signal model (3) becomes

$$x(m) = \mathbf{h}_1^T \begin{bmatrix} s_1(m) \\ \vdots \\ s_1(m-L) \end{bmatrix} + h_2(0)s_2(m-d) + v(m). \quad (5)$$

where $\mathbf{h}_1 = [h_1(0), \cdots, h_1(L)]^T$.

Consider the case with even delay difference $d = d_1 - d_2$ (odd delay difference can be similarly dealt with). We let $d_1 = d$ and $d_2 = 0$. The received even numbered samples are

$$\begin{aligned} x(2n) &= \mathbf{h}_1^T[s_1(2n), \cdots, s_1(2n-L)]^T \\ &\quad + h_2(0)s(2n-d+1) + v(2n). \end{aligned} \quad (6)$$

In general, we have the following rule to change $s_1(\cdot)$ into $s(\cdot)$ according to the STBC structure

$$s_1(m) = \begin{cases} s(m), & \text{for even } m \\ -s^*(m), & \text{for odd } m \end{cases} \quad (7)$$

On the other hand, the received odd numbered samples are

$$\begin{aligned} x(2n+1) &= \mathbf{h}_1^T[s_1(2n+1), \cdots, s_1(2n+1-L)]^T \\ &\quad + h_2(0)s^*(2n-d) + v(2n+1). \end{aligned} \quad (8)$$

B. Viterbi equalizer

It is well known that one of the major advantages of STBC is that computationally efficient maximum likelihood detection is available with flat fading propagation. However, in case of asynchronous transmissions, since the channels become dispersive, we have to use the Viterbi equalizer for the maximum likelihood performance.

Consider the cases listed in the above section, we find that the maximum number of symbols contained in a sample can be $L_s = \max\{L+1, d+2\}$. Therefore, with signaling alphabet size K, we need a trellis with K^{L_s-1} states. If each transmitter transmits N symbols, then we can perform $N + d$ rounds of trellis updating. The complexity of the Viterbi equalizer depends on the maximum value between the channel length and the delay $(d + 2)$. Because the channels are determined completely by the pulse-shaping in flat fading environment, the channel length can be as small as $L = 1$. So the complexity usually depends on the delay difference d.

C. Viterbi equalizer with decision feedback

When d is very large, we may use decision feedback to remove the only one symbol that depends on d, i.e., $s(2n - d - 1)$, $s(2n - d)$, $s(2n - d + 1)$, or $s(2n - d + 2)$. Note that although there are four possible symbols depending on d, each time there is only one of them involved in trellis updating. This decision feedback is especially reliable in case d is very large. With decision feedback, the complexity of the Viterbi equalizer is then effectively reduced to K^L, where L can be as low as 1. Therefore, the Viterbi equalizer is still very useful for the optimal symbol detection with either very small d or very large d, i.e., either closed synchronized transmissions, or completely asynchronous transmissions.

IV. Linear prediction-based blind equalization

A. Vector system model and MMSE equalizer

Consider the signal model (3) with $J = 2$. We can rewrite (3) as

$$x(m) = h_1(0)s_1(m) + \sum_{\ell=0}^{L} h_2(\ell)s_2(m-d-\ell) + v(m). \quad (9)$$

For the Alamouti STBC mapping scheme, we have (7) for the symbols transmitted from the first transmitter and

$$s_1(m) = \begin{cases} s(m+1), & \text{for even } m \\ s^*(m-1), & \text{for odd } m \end{cases} \quad (10)$$

for the symbol sequence transmitted from the second transmitter.

Under the even d assumption, the even-numbered samples $x(m)$ (i.e., m is even) equals

$$x(m) = h_1(0)s(m) + \sum_{\ell=0}^{L} h_2(\ell)\tilde{s}_e(m, d, \ell) + v(m), \quad (11)$$

where

$$\tilde{s}_e(m,d,\ell) = \begin{cases} s(m-d+1-2\ell), & \text{for } \ell = 0,2,\cdots \\ s^*(m-d-2\ell), & \text{for } \ell = 1,3,\cdots \end{cases} \quad (12)$$

Similarly, the odd-numbered samples $x(m)$ equals

$$x(m) = -h_1(0)s^*(m) + \sum_{\ell=0}^{L} h_2(\ell)\tilde{s}_o(m,d,\ell) + v(m), \quad (13)$$

where

$$\tilde{s}_o(m,d,\ell) = \begin{cases} s^*(m-d-1-2\ell), & \text{for } \ell = 0,2,\cdots \\ s(m-d+2-2\ell), & \text{for } \ell = 1,3,\cdots \end{cases}. \quad (14)$$

When we stack the received samples $x(m)$ into vectors, the corresponding channel usually becomes a sparse matrix. In the following, we give a way to construct such a signal model with a more regular channel matrix. Let us begin with $x(2n)$. There are at most $L+2$ symbols contained in $x(2n)$, i.e., $s(2n)$, $s(2n-d+1-2\ell)$, $s^*(2n-d-2\ell)$, for all $\ell = 0,\cdots,L$. The symbol $s(2n)$ is corresponding to the channel coefficient $h_1(0)$. Next, we find the sample that has the symbol $s(2n-d+1)$ corresponding to $h_1(0)$. According to the rule (11)-(14), this sample can be found as $x(2n-d+1)$, which contains symbols $-s^*(2n-d+1)$, $s^*(2n-2d-2\ell)$ and $s(2n-2d+3-2\ell)$ for $\ell = 0,\cdots,L$. In this case, we can use $x^*(2n-d+1)$ so that both $x(2n)$ and $x^*(2n-d+1)$ contains the same symbol $s(2n-d+1)$.

The result is that we construct a vector signal model

$$\mathbf{x}(m) = \mathbf{H}\mathbf{s}(m) + \mathbf{v}(m), \quad (15)$$

where $\mathbf{x}(m)$ is a vector with N samples, whose first sample is determined as $x(2n)$ or $x(2n+1)$. All other samples depends on the delay and the channel length. The noise vector $\mathbf{v}(m)$ has similar structure. The symbol vector $\mathbf{s}(m)$ contains all the corresponding symbols, whose dimension various with respect to N, d and L. But the first symbol is always either $s(2n)$ or $s(2n+1)$.

Although having a complex structure, the channel matrix has some useful property. The major one is that it is diagonal (but it is a wide matrix). The elements of the main diagonal are either $h_1(0)$ or $-h_1^*(0)$.

A popularly used equalizer might be the MMSE equalizer in this case. Let the $(K+1)$th the column of the channel matrix be $\mathbf{h}_K$, then the MMSE equalizer $\mathbf{f}_{MMSE}$ is an N dimensional vector equals

$$\mathbf{f}_{MMSE} = (\sigma_s^2\mathbf{H}\mathbf{H}^H + \sigma_v^2\mathbf{I}_N)^{-1}\mathbf{h}_K, \quad (16)$$

where $\mathbf{I}_N$ is an $N \times N$ identity matrix.

One of the major difference for this case from traditional MMSE equalization case is that the channel matrix $\mathbf{H}$ now is sparse. For such a sparse channel matrix, the MMSE equalizer usually does not have satisfactory performance, as demonstrated by the simulations.

B. Linear prediction-based blind equalization

In order to obtain linear equalizers with better performance, it would desirable to exploit the special property of the sparse channel matrix $\mathbf{H}$. Such special properties include that $\mathbf{H}$ is upper triangular and the diagonal elements are either $h_1(0)$ and $-h_1^*(0)$. In addition, above this main diagonal there are only a few non-zero elements.

Furthermore, we can assume that $|h_1(0)| \geq |h_2(\ell)|$ for all $\ell = 0,\cdots,L$. This is because of the special property that the receiver can perform sampling with respect to the optimal timing of each individual transmitter, although there are no optimal timing for both of them simultaneously. As a result, the sampling can always be performed with the optimal timing of the stronger component of the received signal.

From the signal model (15), we define the linear prediction problem

$$y(m) = \begin{bmatrix} 1 & -\mathbf{p}^H \end{bmatrix} \mathbf{x}(m). \quad (17)$$

Then we optimize $\mathbf{p}$ to obtain

$$\mathbf{p} = \arg\min_{\mathbf{p}} E[|y(m)|^2]. \quad (18)$$

Note that such a linear prediction are performed with either all even-numbered samples $y(2n)$ or all odd-numbered samples $y(2n+1)$. The even or odd numbered samples are not used together.

Consider the even case $m = 2n$ for example. The optimization (18) gives

$$J = \begin{bmatrix} 1 & -\mathbf{p}^H \end{bmatrix} E[\mathbf{x}(2n)\mathbf{x}^H(2n)] \begin{bmatrix} 1 \\ \mathbf{p} \end{bmatrix}. \quad (19)$$

Define the $N \times N$ correlation matrix

$$\mathbf{R}_e = E[\mathbf{x}(2n)\mathbf{x}^H(2n)] = \begin{bmatrix} r_0 & \mathbf{r}^H \\ \mathbf{r} & \mathbf{R}_1 \end{bmatrix}, \quad (20)$$

where $r_0 = E[x(2n)x^*(2n)]$ is a scalar, $\mathbf{r}$ is an $(N-1)\times 1$ vector, whereas $\mathbf{R}_1$ is an $(N-1)\times(N-1)$ square submatrix. Then the optimal solution can be obtained by letting

$$\frac{\partial J}{\partial \mathbf{p}} = \mathbf{0}, \quad (21)$$

which gives solution

$$\mathbf{p} = \mathbf{R}_1^{-1}\mathbf{r}. \quad (22)$$

In practice, the linear predictions can be implemented by the efficient LMS adaptive algorithm. Therefore, the complexity of this blind equalizer is $O(N)$, where N is the equalizer length.

V. SIMULATIONS

We compare the Viterbi equalizer with the classical STBC decoder when the latter works with perfect synchronization. In addition, we compare this algorithm when working with asynchronous transmissions. All these algorithms are compared with the case without diversity. They are labelled as, respectively, “Asyn”, “SynSTBC”, “w/o Equ” and “noDvst”.

We compare the LP-base linear equalizer (“LP”) with the asynchronous transmission scheme in [6] (“SynSTBC”). The

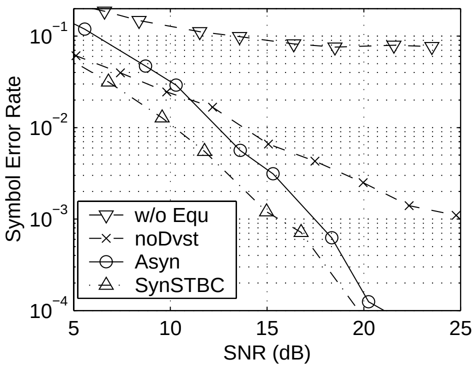


Fig. 3. Symbol-error-rate (SER) as function of SNR. ∇(w/o Equ): using standard STBC receiver in asynchronous transmission. ×(noDvst): no diversity case with flat-fading single-tap channels. ◯(Asyn): Viterbi equalizer proposed for STBC with asynchronous transmissions. △(SynSTBC): standard STBC for synchronous STBC transmissions.

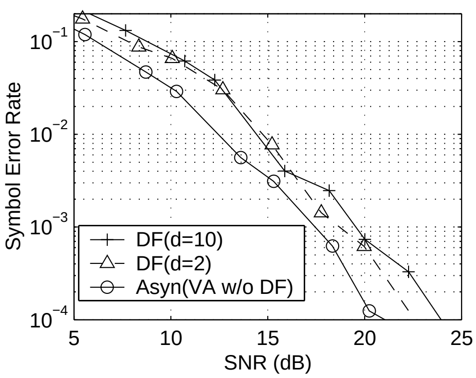


Fig. 4. The Viterbi-equalizer with decision feedback has comparably near-optimal performance. SER vs. SNR with $d = 2$ and $d = 10$. $+$: VA with decision feedback and delay $d = 10$. △: VA with decision feedback and delay $d = 2$. ∘: VA without decision feedback and $d = 2$.

classical STBC decoder ("w/o equ") is again compared, same as the case without diversity ("noDvst"). In addition, the MMSE equalizer is also compared ("MMSE").

For the Viterbi equalizer experiments, the delay $d = 1$, and a trellis with 128 states is used. 1000 randomly generated symbols are transmitted and detected during each run. Simulation results are shown in Fig. 3. The proposed Viterbi equalizer can achieve almost the optimal performance of the synchronous STBC. Both the two cases provide diversity advantage beyond the case transmissions without diversity.

Then we compare the performance of the Viterbi-equalizer with decision feedback. We use only a 4-state trellis in Viterbi equalizer. The simulation results are shown in Fig. 4. Decision feedback can greatly reduce the complexity of the Viterbi equalizer while keeping most of the diversity benefits. Viterbi equalizer with decision feedback is thus a potential candidate for asynchronous STBC.

For the proposed LP-based linear receiver, an equalizer with length $N = 20$ is used. The delay is $d = 10$. The simulation results are shown in Fig. 5. The proposed LP-blind equalizer has about 3dB gap from the linear equalizer with

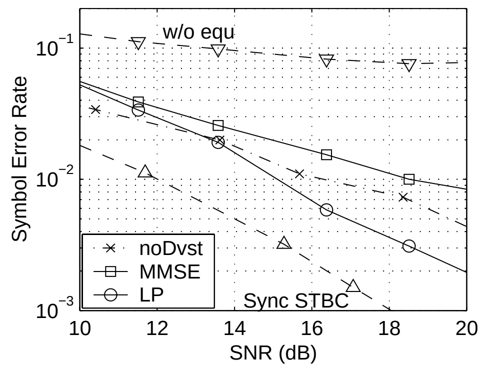


Fig. 5. Symbol-error-rate (SER) as function of SNR for linear equalizers. ∇(w/o equ):standard STBC receiver is used in asynchronous transmissions. ×(noDvst):the transmission without diversity but with dispersive channels. □(MMSE): the MMSE equalizer for asynchronous transmissions. ◯(LP): the proposed LP-based blind equalizer for asynchronous transmissions. △(SyncSTBC): STBC with quasi-synchronous transmissions [5].

quasi-synchronous transmissions. We see that the proposed LP method can successfully equalize the channel even blindly, and has better performance than the transmission without diversity. In contrast, the MMSE equalizer can not outperform and transmissions without diversity, whereas if no equalizer is used, the receiver completely fails.

VI. Conclusions

In this paper, the STBC-encoded transmissions when the transmitters are asynchronous in time are considered. We show that the classical STBC receivers fail in this case. We proposed two equalization techniques to recover the diversity benefits: the optimal Viterbi equalizer and the linear-prediction-based blind linear equalizer. The former has performance near optimal, the latter has lower complexity but has some performance (diversity) loss.

References

[1] S. M. Alamouti, "A simple transmitter diversity scheme for wireless communications", *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1451-1458, Oct. 1998.

[2] V. Tarokh, H. Jafarkhani and A. R. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Trans. Inform. Theory*, vol. 45, no. 7, pp. 1456-1467, July 1999.

[3] A. Sendonaris, E. Erkip and B. Aazhang, "User cooperation diversity, Part I, II," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1927-1948, Nov. 2003.

[4] J. N. Laneman and G. W. Wornell, "Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks," *IEEE Trans. Inform. Theory*, vol. 49. no. 10, pp. 2415-2425, Oct. 2003.

[5] P. Stoica and E. Lindskog, "Space-time block coding for channels with intersymbol interference," *Proc. 35th Asil. Conf. Signals, Syst., Comput.*, vol. 1, pp. 252-256, Pacific Grove, CA, Nov. 2001.

[6] X. Li, "Space-time coded multi-transmission among distributed transmitters without perfect synchronization," *IEEE Signal Process. Lett.*, vol. 11, no. 12, pp. 948-951, Dec. 2004.

[7] X. Li, M. Chen and W. Liu, "Application of STBC-encoded cooperative transmissions in wireless sensor networks," to appear in *IEEE Signal Processing Lett.*, Feb. 2005. Available online at: *http://ucesp.ws.binghamton.edu/∼xli*.

[8] S. Barbarossa and G. Scutari, "Distributed space-time coding strategies for wideband multihop networks: regenerative vs. non-regenerative relays," *Proc. ICASSP 2004*, Montreal, Canada, June 2004.