A Fast Tracking Adaptive Array System For Mobile Communications

H. Wang and K.J.R. Liu

Electrical Engineering Department and Institute for Systems Research
University of Maryland, College Park, MD 20742

ABSTRACT

In slowly time-varying fading channels, adaptive diversity combining can reduce multipath fading of desired signal and suppress interfering signals. However, for fast time-varying fading channels, there are no effective techniques to achieve the same results. The continued use of decision directed adaptive array algorithms on the fast fading channels will cause error propagation. This paper presents a novel adaptive diversity combining technique with proposed M-D decoder and QRD-RLS based parallel weights tracking. This system reduces error propagation in the decision directed array systems while maintaining the same tracking speed. Computer simulation has been performed to show the effectiveness and much better performance than that of the existing techniques.

1. INTRODUCTION

Adaptive diversity combining is a powerful technique in wireless communications[1][2]. It can reduce multipath fading of the desired signal and suppress interfering signals. Its effectiveness on a fast time-varying channel, however, depends on the tracking speed of the adaptive algorithms used and on the control of a decision error propagation.

Adaptive optimum diversity combining weights tracking and adaptive equalization are two closely related problems. Most of the techniques used for equalization can also be applied to adaptive diversity combining. However, the majority of previous studies on channel equalization or adaptive array combining have concentrated mostly on slowly time-varying fading channels.

In a slowly time-varying fading channel, the LMS algorithm is used in most channel estimation and equalization situations. A small change in the equalizer weights is capable of tracking the fading variation. Error propagation is not a severe problem. The Bit Error Rate (BER) performance can be further improved with the combination of an equalizer and a Viterbi decoder. A tentative decision with small delay or no delay from a Viterbi decoder has been used for channel estimation and tracking. In contrary, in a fast time-varying fading channel, a decision delay results in poor tracking performance, and a premature tentative decision will cause erroneous channel estimation and error propagation.

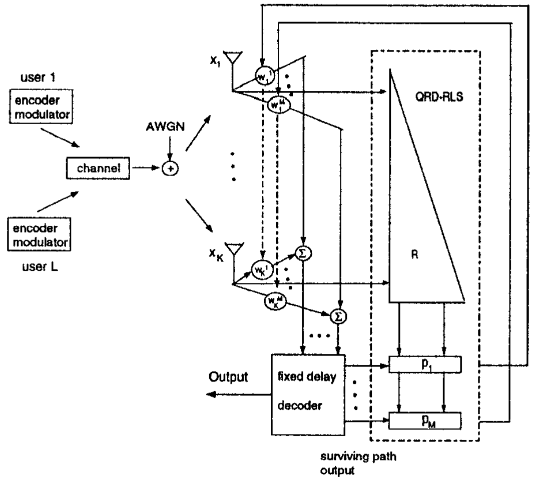


Figure 1. The proposed adaptive diversity combining system

Recently, respective-states channel estimation(RCE) [3] [4] was proposed and better performance over conventional decision-directed channel equalization was reported on fast time-varying fading channels. Nevertheless, the continued use of the Viterbi algorithm (VA) in RCE tends to introduce error. The tracking ability of this approach is also limited by the convergence rate of LMS. Moreover, this method has not been studied on adaptive array systems.

To effectively perform diversity combining on a fast time-varying fading channel, we developed a novel adaptive diversity combining system which incorporates combined array weights tracking and decoding. This system reduces error propagation in the decision directed array system while maintaining the same tracking speed.

2. PROPOSED ADAPTIVE DIVERSITY COMBINING SYSTEM

Fig. 1 is a block diagram of a multi-user communication system with the proposed adaptive array receiver. There are L users at the same time, These L sources are assigned with different training sequences and encoded with a binary convolutional code or 8-PSK Trellis code. The communication channel is a flat fading channel with additive white Gaussian noise.

At each symbol interval, M sets of array weights are

This work was supported in part by the NSF grant MIP9457397 and MIP9309506

0-7803-3192-3/96 $5.0(©1996 IEEE

tracked by using the QRD-RLS algorithm along each of the M surviving paths selected by our proposed M-D decoding algorithm.

A false path will cause erroneous weights estimation in the subsequent D branches and results in a large accumulated path metric. It is most likely that the corresponding path will be discarded D branches later from the surviving paths. A correct path will result in a small accumulated path metric and will most likely be kept as the surviving path. A final symbol decision is made with D or $D+1$ symbol delay.

Our M-D decoding algorithm incorporates (a) the feedback decoding algorithm previously used for a binary symmetric channel (BSC) hard decision decoding and (b) the M-algorithm which is one of the sequential decoding algorithms. It is developed based on our D-symbol delay decoding algorithm, which is presented here first.

2.1. D-symbol delay decoding algorithm

Understanding of the following key properties of the trellis diagram we used is crucial to the development of our algorithms.

- Property 1. The states in the trellis are ordered from 0 to $2^D - 1$.
- Property 2. In the construction of our trellis diagram, for each state, the path generated by a "0" input bit is always placed above that by a "1" input bit.
- Property 3. The ith bit of each state corresponds to the input bit back by i stages.
- Property 4. Every upper path of a pair that merges to each state corresponds to a "0" input bit back by $D+1$ stages. Every lower path of a pair that merges to each state corresponds to a "1" input bit back by $D+1$ stages.

The developed D-delay decoding algorithm is given as follows.

- At each stage, all the accumulated path metrics of surviving paths are compared. If the path with the smallest path metric is an upper path, then every upper path at each state is kept and all the lower paths are discarded. The information bit back by $D+1$ stages is decided to be 0. Otherwise, all the lower paths are kept and all the upper paths are discarded, the information bit back by $D+1$ stages is decided to be 1.
- At the end of each time slot, the path with the smallest path metric is selected. If the path is an upper path, then the bit back by $D+1$ stages is decided to be"0", vice versa. The last D input bits are decided based upon the final state of the selected path. The last ith bit is equal to the first ith bit of the selected state.
- The accumulated path metric is calculated by the following equation:

$$\sum_{t=t_0+1}^{t_0+D} (\mathbf{w}_s(t-1)\mathbf{x}(t) - m_i(t))^2 \qquad (1)$$

where array weights $\mathbf{w}_s(t)$ is estimated along each surviving path. $m_i(t)$ is a modulated signal of the ith path output at time t. In general, at time t_0, a set of array weights $\mathbf{w}_s(t)$ at state "s" is obtained by updating $\mathbf{w}_{pre(s)}(t-1)$, which is a set of array weights obtained at the state prior to "s" along the surviving path, by using array data $\mathbf{x}(t)$ and the modulated output signal of the path from state "pre(s)" to "s".

2.2. M-D algorithm

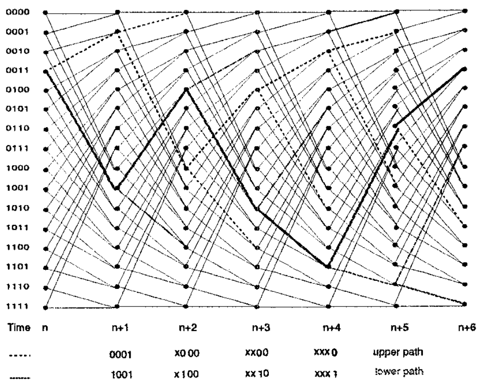


Figure 2. 4-symbol delay decoding trellis diagram

- At each stage, calculate the next path metrics of all the paths that are generated from the surviving states.
- When all the binary form of the surviving states have a common last bit either "1" or "0", select M paths that have the smallest path metrics from all the surviving paths from the surviving states. If the next states of these M surviving paths also have a common last bit, the input bit back by D stages is decided to be equal to this common last bit. Otherwise no symbol decision is made.
- When all the surviving states have a different last bit, compare all the path metrics from these states. If the smallest path metric is from an upper path, keep all the upper paths and discard all the lower paths. The input bit back by $D+1$ stages is decided to be "0", vice versa. The input bit back by $D+1$ stages is decided to be "1". Select M paths that have the smallest path metrics among surviving paths. If the M selected states have a common last bit, the input bit back by D stages is decided to be equal to this common last bit. Otherwise no more symbol decision is made.
- At the end of each time slot, the path with the smallest path metric is selected. If the path is an upper path, then the bit back by $D+1$ stages is decided to be"0", vise versa. The last D input bits are decided based upon the final state of the selected path. The last ith bit is equal to the first ith bit of the final selected state.

An M-D decoding example is shown in Fig. 2. The path that enters state 0110 at stage $n+5$ has the smallest path metric. It is a lower path. The input bit back by 5 stages is

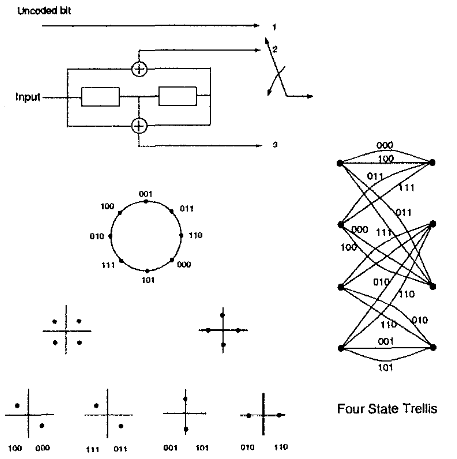


Figure 3. Four-state trellis-coded 8-PSK modulation

decided to be "1". We then discard all the upper paths at stages $n+5$ and select 2 paths from all the lower paths. In this case, both lower paths are selected and kept. We compare the last bit of the two corresponding surviving states 0110 and 1110. They have a common last bit "0". The input bit back by 4 stages is decided to be "0".

2.3. Diversity Weights Tracking

To achieve optimum diversity combining on time varying fading channel, recursive algorithms such as LMS or RLS can be applied. RLS has been known to have fast convergence rate and good tracking capability compared to the low complexity LMS algorithm or blind adaptive algorithms such as CMA. Among different RLS algorithms, QRD-RLS has excellent numerical stability which provides a much larger dynamic range compared to the conventional RLS algorithm. Moreover, it is computational more efficient in our applications. Our system requires the generation of M sets of array weights simultaneously. The array outputs have no special structure. Therefore, none of the fast RLS algorithms such as RLS lattice algorithm is applicable. By using conventional RLS algorithm, the complexity of updating M sets of diversity weights, each of size K, becomes $O((M*K)^2)$. However, the QRD-RLS [6] algorithm can generate M sets of array weights parallelly. The total complexity is only $\mathrm{O}(K^2+M*K)$. Therefore, QRD-RLS algorithm is adopted in our system.

2.4. M-D Decoding of A Trellis Coded 8-PSK code

In some situations such as wireless video transmission, we need to transmit high speed data through some limited frequency bandwidth. A use of 8-PSK signal constellation in conjunction with trellis codes can double the transmitted information bit rate compared to a binary 1/2 convolutional coded QPSK signal. Therefore, we modified the D and M-D decoding algorithm of convolutional code and applied to TCM for these applications. We use the rate 1/2 convolutional code to encode one information bit while the second information bit is left uncoded. The encoder is shown in Fig 3. The decoding algorithm is given as follows:

- Select the branch having the smaller path metric among the parallel branches. If an upper path is selected, the uncoded bit at current time is decided to be "0", otherwise, it is decided to be "1".
- The coded bit is decoded using the D or M-D algorithm presented in the previous sections.

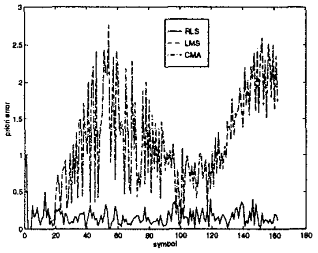


Figure 4. Learning curve of LMS, CMA and RLS

3. SIMULATION RESULTS

In our simulations, four antennas are used. The channel is time-division-multiplexed. There are 162 symbols in each time slot. The first 14 symbols are from the training sequence. Two users (generating desired signal and interference signal) both move at 60 miles/hour. The carrier frequency used is 900 MHz, the modulated data rate is 24.3ksym/s, which is the same as in IS-136 standard. The SNR is 15dB. The TCM encoder shown in Fig. 3 and (2,1,2) convolutional encoder which is used for encoding input bits as shown in Fig. 3 are used to generate 8-PSK TCM signal and convolutional coded QPSK signals.

3.1. Comparison Between LMS, CMA and RLS

It is already known that using either LMS or CMA, it is not possible for weights to converge to optimal values with 14 training symbols (in IS-54) [5]. To see the feasibility of these algorithms on the tracking of fast time-varying fading channels, we did the following simulations. In Fig. 4, ISR(interference to signal ratio) is 0 dB. RLS is used in the training mode. RLS, LMS and CMA are used individually in the tracking mode. We observe a severe performance loss in fast fading channel tracking by using LMS or CMA. Different stepsizes have been tried. Similar or even worse results have been observed when LMS and CMA are used. Therefore the RLS algorithm is adopted in our system for both weights acquisition and tracking.

3.2. Effects of Delayed Temperal Decision

A frequently used approach in a slowly fading channel equalization with convolutional or Trellis coded signal is to use a delayed temporal decision from a viterbi decoder to update the equalizer weights. However, when adopting such techniques to our system, we found a loss in channel tracking and poor BER performance. Especially, for the fast time-varying channels in 2GHz frequency band as shown in Fig. 5, a larger decision delay results in a worse performance.

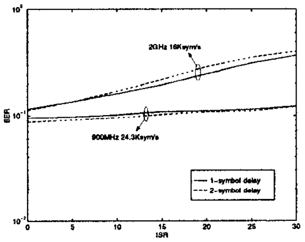


Figure 5. Delayed Decision-directed weights tracking

3.3. Performance Improvement with Proposed Algorithms

Computer simulation results provided in Fig. 6 gave a quantitative examination of the BER improvement of using D-symbol delay algorithm. In comparison to BER achieved by the conventional RLS algorithm without decision delay for uncoded QPSK and coded QPSK signals, 3 ~ 5 dB improvement in interference suppression is observed in both the TCM decoding and the (2,1,2) convolutional decoding. To further reduce the computational cost, we use M-D algorithm. We set D to 4. Fig. 7(a) shows the BER relative to M. We see that an M as small as 4 results in a negligible degradation in the performance relative to the 4-symbol delay algorithm in which $M = 2^4$. The complexity of M-D algorithm is much reduced.

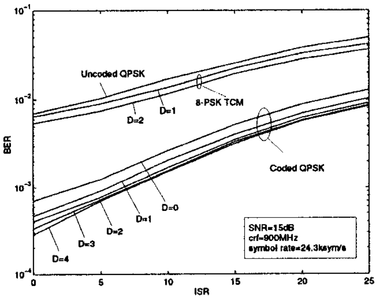


Figure 6. BER performance comparison

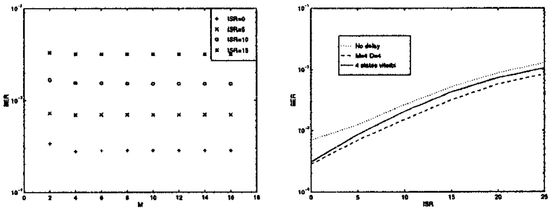
In Fig. 7(b), the BER performance achieved by using M-D algorithm is compared to that achieved by using Viterbi algorithm incorporated in RCE [3] [4]. Infinite memory length is used in the Viterbi decoding. Better BER performance by using M-D algorithm with $D = 4$ and $M = 4$ is observed. The complexity of these two algorithm is compatable. The memory required by M-D algorithm is much less than that of the Viterbi algorithm.

4. CONCLUSIONS

This paper presents a novel adaptive diversity combining technique. D-symbol delay and M-D decoding algorithms are developed for the binary convolutional codes and TCM codes. A decoder with M-D algorithm provides instantaneously a set of candidate reference signals for weights tracking and makes a final symbol decision with a D or $D + 1$ symbol delay based on more reliable accumulated path metrics. It thus reduces error propagation in the decision directed array systems while maintaining the same tracking speed.

The memory required by the M-D algorithm is only O(M). The computational complexity required is $O(K^2 + M \cdot K)$ for weights updating and $O(M)$ comparisons for decoding, which is not much increased compared to $O(K^2)$ in the conventional decision directed adaptive array system.

The effectiveness of this technique and better performance than that of the existing techniques have been shown in computer simulation.



(a)

(b)

Figure 7. (a) Influence of M on BER (b) Comparison between D-delay Algorithm and VA

REFERENCES

[1] J. H. Winters, J. Salz and R. D. Gitlin, "The Impact of Antenna Diversity on the Capacity of Wireless Communication Systems," *IEEE Trans. on Communications*, vol. 42, No. 2/3/4, pp. 1740-1751, February/March/April 1994.

[2] P. Jung, "Performance Evaluation of a Novel M-Detector for Coherent Receiver Antenna Diversity in a GSM-Type Mobile Radio System," *IEEE Journal on Selected Areas in Communications*, vol. 13, No.1, pp. 80-88, January 1995.

[3] H. Kubo, K. Murakami and T. Fujino, "An Adaptive Maximum-Likelihood Sequence Estimation for Fast Time-Varying Intersymbol Interference Channels" *IEEE Journal on Selected Areas in Communications*, vol. 42, No. 2/3/4, pp. 1872-1880, Feb./March/April 1994.

[4] H. Kubo, K. Murakami and T. Fujino, "Adaptive Maximum-Likelihood Sequence Estimation by Means of Combined Equalization and Decoding in Fading Environments," *IEEE Journal on Selected Areas in Communications*, vol. 13, No. 1, pp. 102-109, January 1995.

[5] J. Proakis, "Adaptive Equalization for TDMA Digital Mobile Radio," *IEEE Trans. on Vehicular Technology*, vol. 40, No.2, pp. 333-341, May 1991.

[6] S. Haykin, "Adaptive Filter Theory," Second Edition, Prentice Hall Information and System Sciences Series, 1991.