# Diversity Combining with Novel Fixed Delay M-D Decoder for Fast Time-Varying Fading Channels

*H. Wang and K.J.R. Liu*

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

## ABSTRACT

*This paper presents a simultaneous diversity weights tracking and decoding technique using QRD-RLS based parallel prediction error updating and proposed M-D decoder. This system significantly reduces error propagation in the decision directed array systems with moderate increase in complexity. Computer simulation showed its effectiveness and better performance than that of the existing techniques.*

## 1. INTRODUCTION

Diversity combining has been widely used in wireless communication. It is a powerful technique for combating multipath flat fading. The most commonly used diversity combining techniques include maximal-radio combining, MMSE optimum coherent combining, equal-gain combining and selective combining. Among them, the optimum coherent combining can reduce multipath fading of the desired signal as well as suppress interfering signals [1]. It has attracted lots of attentions recently [1]-[2]. Its effectiveness on a fast time-varying channel, however, depends on the tracking speed of the adaptive algorithms and on the control of a decision error propagation.

The array weights in the optimum coherent combining are chosen to minimize the error between the reference signal and array output. Unfortunately, the reference signals are not always available. In the TDMA system, we have only 14 symbols at the begining of each time slot that can be used as the reference signals, after that, we make symbol by symbol decision, and feedback the decided symbol every time to update the array weights. If we make a decision error, this error will propagate and cause subsequent decision errors.

Instead of making instantaneous premature decision, One solution is to combine with coding and feedback more reliable delayed tentative decision from a decoder. This can be used in slowly time-varying channels. but in fast time-varying channels, a decision delay results in poor tracking performance.

Alternatively, blind equalization techniques such as constant modulus algorithm (CMA) can be used to avoid error propagation in the decision-directed algorithm. Unfortunately, blind equalizing algorithms converge slowly and are not capable of tracking fast time-varying fading channels.

We developed simultaneous diversity weights updating and decoding algorithm. It reduces error propagation in the decision directed system while maintaining the same tracking speed.

## 2. PROPOSED ADAPTIVE DIVERSITY COMBINING SYSTEM

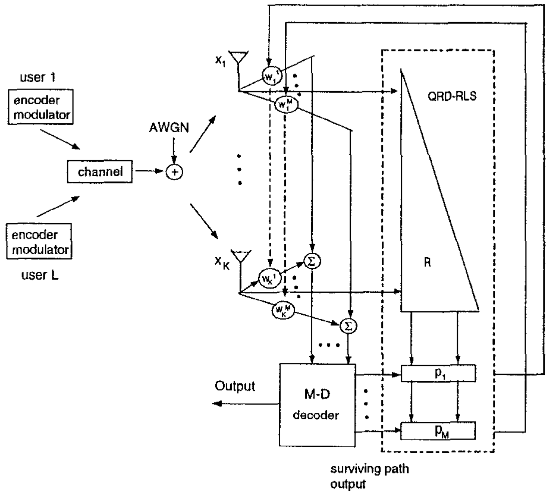

Figure 1. The proposed adaptive diversity combining system

Fig. 1 is a block diagram of a prototype 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.

We use binary convolutional code. In Fig. 2, at each state, two possible QPSK symbols may be transmitted based upon a "0" or a "1" input. Without making premature symbol decisions based upon one symbol, we calculate and save the Euclidean distance between array output and each of two path outputs. We update weights separately along the two possible paths and calculate the accumulated path metrics at the next stage. This process is iterately. Each time, we use path output as the reference signal, instead of feeding back any decided symbol. After D branches, A false path will cause erroneous weights estimation and results in a large accumulated path metric. Along the correct path, weights are updated correctly symbol by symbol and there is no error propagation. The accumulated path metric on the correct path should be small. Therefore, we look for the smallest path metric. Once this path is chosen, we decide the input bit back by $D$ stages. We save all the branches originating from it and discard all the other branches. This way, we make a more reliable symbol decision based upon $D$ symbols. We reduce error propagation but keep the same tracking speed.

There are two new issues associated with the above algorithm.

- When there is strong interference in the received signals, if the path output of a wrong path happens to be the same

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

0-7803-3250-4/96$5.00©1996 IEEE

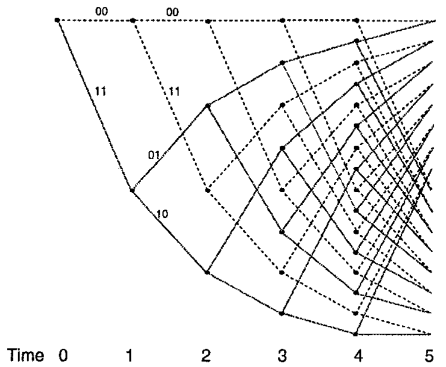

Figure 2. Simultaneous Weights Tracking and Decoding (D=5)

as the transmitted sequences from that strong interference, and if the $D$ is too large, weights may converge along the wrong path.

- The complexity of this algorithm is increased drastically compared to the conventional symbol by symbol decision directed algorithm. Now we need to update $2^D$ weighting vector instead of 1 weighting vector.

To avoid the first problem, $D$ has to be chosen appropriately based upon the window size used in the weights updating algorithm.

To solve the second problem, we find that the correct path should have much smaller accumulated path metric than the rest of the selected paths. So, we can discard most of the selected paths and keep those most likely ones without compromising the performance. This results in the following M-D algorithm.

### 2.1. M-D algorithm

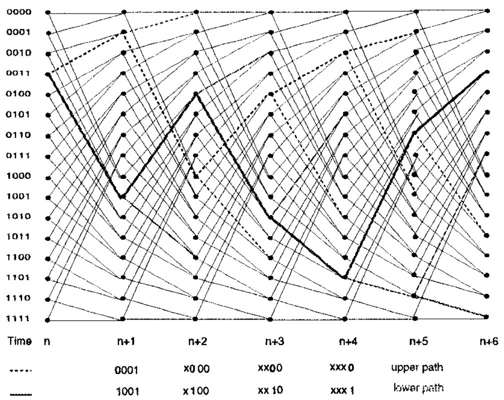
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
Time n n+1 n+2 n+3 n+4 n+5 n+6
0001 x0 00 xx00 xxx 0 upper path
1001 x100 xx 10 xxx 1 lower path

Figure 3. 4-symbol delay decoding trellis diagram

- At each stage, calculate the next branch metric and accumulated path metrics of all the paths that are generated from the surviving states. 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. $m_i$ is a modulated signal of the $i$th 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".

- When all the binary form of the surviving states have a common last bit either "1" or "0", the input bit $D$ stages back is decided to be equal to this common last bit. Select $M$ paths that have the smallest accumulated path metrics from all the surviving paths from the surviving states.

- When all the surviving states have a different last bit, compare all the path metrics from these states. If the smallest accumulated path metric is from an upper path, keep all the upper paths and discard all the lower paths. The input bit $D$ stages back is decided to be "0". Vice versa. The input bit $D$ stages back is decided to be "1". Select $M$ paths that have the smallest accumulated path metrics among surviving paths.

- At the end of each time slot, The last $D$ input bits are decided based upon the final state of the selected path. The last $i$th bit is equal to the first $i$th bit of the final selected state.

In Fig. 3, the two surviving states 0010 and 1101 at stage $n+4$ have different last bits. We calculate the four path metrics from the four paths generated from these two states. The path that enters state 0110 at stage $n+5$ has the smallest accumulated path metric. It is a lower path. The input bit 4 stages back is decided to be "1" which is the bit that generates the path from state 0011 at stage $n$ to the state 1001 at stage $n+1$. 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 4 stages back is decided to be "0" which is the bit that generates the path from 1001 at stage $n+1$ to the state 0100 at stage $n+2$.

### 2.2. Diversity Weights Tracking

To achieve optimum diversity combining on time varying fading channel, recursive algorithms such as LMS or RLS can be applied for weights updating. RLS has been known to have fast convergence rate and good tracking capability [4] compared to the low complexity LMS algorithm or blind adaptive algorithms such as CMA. Among different RLS algorithms, QRD-RLS [5] has excellent numerical stability which provides a much larger dynamic range compared to the conventional RLS algorithm. Moreover, by using QRD-RLS, path metric can be calculated directly without getting $\mathbf{w}$ explicitly. The total complexity is only $O(K^2 + M*K)$. Therefore, QRD-RLS algorithm is adopted in our system.

### 2.3. 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. In our example of TCM, we partition the eight-phase signal constellation shown in Fig. 4 into subsets of increasing minimum Euclidean distance. We use the rate of 1/2 convolutional code

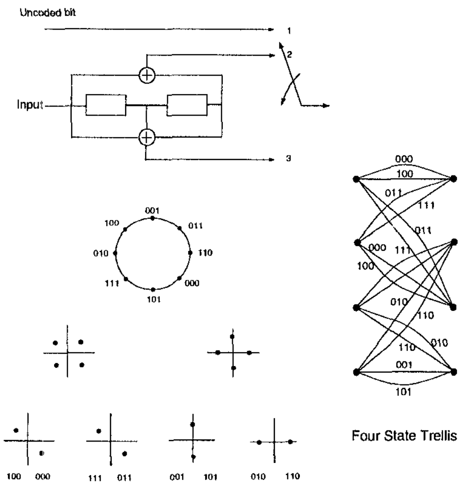

Figure 4. Four-state trellis-coded 8-PSK modulation

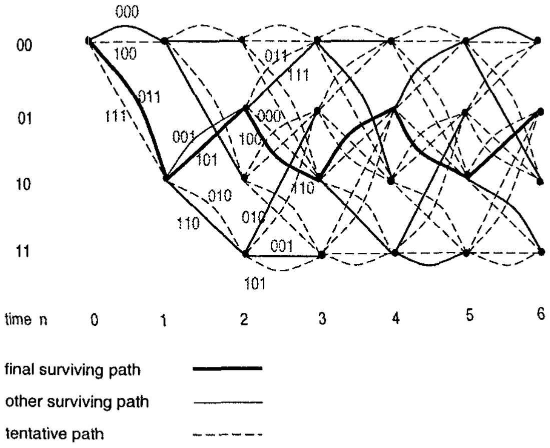

Figure 5. 2-symbol delay 8-PSK TCM decoding trellis diagram

to encode one information bit while the second information bit is left uncoded. The encoder is shown in Fig 4. The coded bits are used to select one of the four subsets that contain two signal points each, while the uncoded bit is used to select one of the two signal points within each subset. The Euclidean distance between parallel paths is $2\sqrt{\epsilon}$, where $\epsilon$ is the energy of the signal.

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.

Fig. 5 shows a 2-symbol delay 8-PSK TCM decoder. At time $n$, one branch is selected between each pair of parallel branches in favor of the one having a smaller branch metric. Then we choose surviving paths among these selected paths following the same procedure that is used in the convolutional decoder having a 2-symbol delay. At time 3, one path is selected from each pair of branches that are generated from surviving states 01 or 11 at stage 2. Within these four selected paths, the path enters state 10 at stage 3 has the smallest accumulated path metrics. All the lower paths in the surviving paths are saved, i.e., the paths enter state 00 and state 10 at stage 3.

## 3. PERFORMANCE ANALYSIS

We calculate the $i$th path accumulated metric $P_i$ at time $n$ as follows

$$P_i = \sum_{d=n-D}^{n} b_{d,i} = \sum_{d=n-D}^{n} |\mathbf{w}_{d-1,i}^{H}\mathbf{x}_d - s_{d,i}|^2 \qquad (2)$$

where $b_{d,i}$ is the $d$th branch metric of the $i$th path, $\mathbf{w}_{d-1,i}$ is the array weighting vector updated based upon the output symbols from the $(d-1)$th branch of the $i$th path, $s_{d,i}$ is the output symbol from the $d$th branch of the $i$th path. Since all the surviving pathes have common branches before time $n-D$, we only need to compare their accumulated path metrics from time $n-D$ to time $n$. In general, if we assume that M-PSK signal modulation is used, the branch metrics can be divided into three categories.

- Case 1: $\mathbf{w}$ is updated based upon $n$ desired symbols, $s_{d,l}$ is a desired symbol. $E(b_{d,l}) = mse$. $\frac{b_{d,l}}{mse} \sim \chi_1^2$. The $mse$ is determined by the adaptive algorithm such as RLS or LMS and also determined by the channel conditions such as fading rate and the number of interferences.
- Case 2: $\mathbf{w}$ is updated based upon $n$ desired symbols, $s_{d,i}$ is an undesired symbol, $\frac{b_{d,i}}{mse}$ is from noncentral $\chi_1^2$ with 1 degree of freedom and noncentral parameter $d_1$, where $d_1$ is the Euclidean distance between two output signals from two branches that stem from the same node. We denote it as the first degree free Euclidean distance.
- Case 3: $\mathbf{w}$ is updated based upon $n-m$ desired symbols and $m$ undesired symbols, $s_{d,i}$ is an undesired symbol, $\frac{b_{d,i}}{MSE_m}$ is approximately $\sim \chi_1^2$. We denote $E[b_{d,l}]$ by $MSE_m$. It can been shown that

$$MSE_D \geq MSE_{D-1} \geq MSE_{D-2} \geq \cdots \geq MSE_1 > mse. \qquad (3)$$

Note that $\mathbf{w}$ is from false training and is not controllable. Neither is $MSE_m$.

Accordingly, there are three classes of path metrics:

- Class 1: $P_c$ is a path metric of a correct path.
- Class 2: The first $m$ branches belong to case 1, the $(m+1)$th branch belongs to case 2, the last $D-m$ branches belong to case 3, $m = 1, \cdots, D$
- Class 3: The first branch metric belongs to case 2, the rest branches belong to case 3.

In a D-delay decoder, we make a choice between two subsets of equal size based upon the smallest $P_i$ at each stage. Notice that all the paths in the undesired subset belong to class 3, and the paths in the desired subset belong to either class 1 or class 2. The average difference between a $P_u$ in the undesired subset and the $P_d$ corresponding to the correct path is given by

$$E(P_u - P_d) = \sum_{i=1}^{D} MSE_m - (D) \cdot mse + d_1^2. \qquad (4)$$

We minimize the $mse$ by using the fast convergent RLS algorithm and by choosing a proper window size. To increase $E(P_u - P_d)$, a convolutional code with given constraint length needs to be chosen to maximize $d_1$. In our selected Q-PSK convolutional code, $d_1$ is maximized to $\sqrt{2\epsilon}$.

The average difference between a $P_u$ and a $P_i$ of a false path in the desired subset is

$$E(P_u - P_i) = \sum_{i=D-m+1}^{D} MSE_i - m \cdot mse \quad m = 1, 2, \cdots, D, \tag{5}$$

The overall average difference between a path in Class 3 and a path in Class 2 for a D-symbol fixed-delay decoder is

$$\begin{aligned} E_D &= \frac{1}{2^D - 1} \sum_{m=1}^{D} 2^{D-m} \sum_{i=D-m+1}^{D} (MSE_i - m \cdot mse) && (6) \\ &= E_{D-1} + [\sum_{i=1}^{D} (MSE_i - D \cdot mse) - E_{D-1}] && (7) \\ &> E_{D-1} && (8) \end{aligned}$$

The increase of the decision delay increases the average Euclidean distance between the correct path and a false path in the other subset and also the average Euclidean distance between paths in the two different subsets. Therefore, The first event error is decreased with the increase of the decoding decision delay. However, if the output symbols of one of the $2^D$ surviving paths happen to be the same as the transmitted symbols from the interference, the further increase of $D$ will also cause the weights to converge along the interference path and will result in an increase of the BER. Our simulation results showed that the increase of $D$ will decrease the BER when the interference signals are not as strong as the desired signal. When an interference signal is stronger than the desired signal, BER decreased as we increase the decision delay from 1 to 4 symbols. The improvement becomes smaller and smaller. As we further increase the decision delay, we observe a slightly increase of BER. A proper delay length needs to be decided based upon I/S ratio and RLS updating window size.

The correct path has the smallest expected accumulated metric. It is not likely that the accumulated metric of the correct path is larger than one half, one third or an even smaller portion of the accumulated metrics in Class 2. Therefore, we can use the M-D decoding algorithm with $M << 2^D$ ($M = 2$, $D = 4$ in our examples) without compromising the performance.

## 4. 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. The carrier frequency 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 (2,1,2) convolutional encoder and the TCM encoder as shown in Fig. 4 are used to generate convolutional coded QPSK signals and 8-PSK TCM signal.

Computer simulation results provided in Fig. 6 and Fig. 7 give a quantitative examination of the BER improvement from using D-symbol delay algorithm and M-D algorithm, respectively. Three sources are all encoded with the convolutional encoder shown in Fig 4. The desired vehicle is moving at 60miles/hr. The other two are moving at 30miles/hr. We observe the improved BER performance as we increase D. A total of 10dB improvement with $D = 5$ over $D = 0$ in both cases. With M-D algorithm, the improvement is slightly less compared to $D$ algorithm, but the complexity of M-D algorithm is greatly reduced.

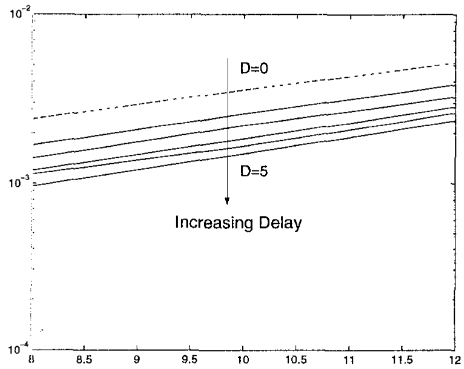

Figure 6. Influence of D on the performance of D-delay algorithm

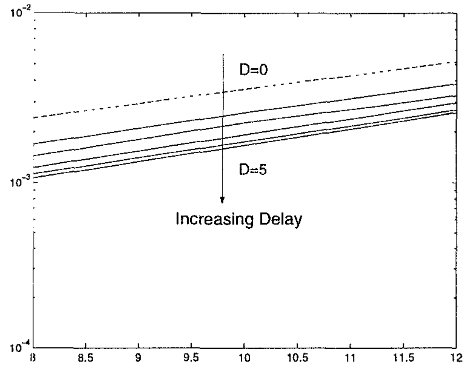

Figure 7. Influence of D on the performance of M-D algorithm (M=2)

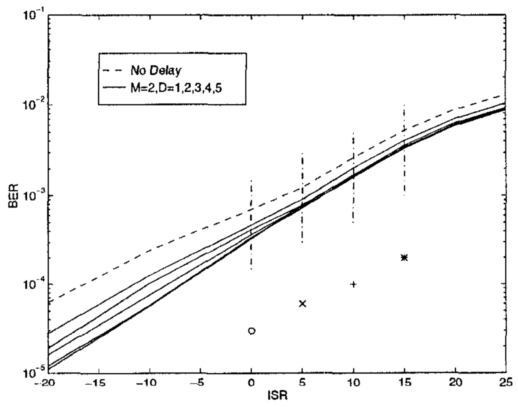

Figure 8. More significant improvement at low ISR (M-D algorithm, M=2)

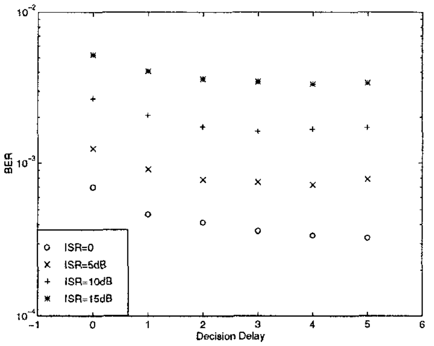

Figure 9. Influence of D on the performance of M-D algorithm at high ISR

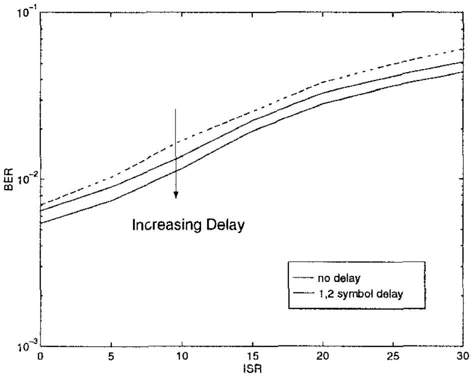

Figure 10. BER performance of D-symbol delay decoding of a TCM code

In Fig. 8, the M-D algorithm is used and M is set to 2. At low ISR, BER performance is gradually improved as $D$ is increased. At high ISR, Fig 9 shows an improved BER performance when $D$ is increased from 1 to 4 and a slightly decreased performance in BER as $D$ is further increased. When the interference is stronger than a desired signal, weights may converge along a wrong surviving path. So the $D$ should be appropriately chosen at high ISR to achieve optimal BER.

An extension of D-delay and M-D algorithms to the TCM code is demonstrated in the following example. A TCM encoder shown in Fig. 4 is used. In Fig. 10, the dashed line represents the BER performance under decision directed QRD-RLS for weights tracking. An immediate symbol by symbol decision is made on an uncoded QPSK signal sequence. The solid lines represent the BER under 1-symbol delay and 2-symbol delay algorithm. Using 2-symbol delay algorithm, 3 to 5 dB improvement in the strong interference suppression over the conventional decision directed algorithm is observed.

The BER performance achieved by using M-D algorithm with $M = 2, D = 4$ are compared to the that achieved by using Viterbi algorithm [3] with 4 states. Infinite memory length is used in the Viterbi algorithm. More BER improvement is achieved by using M-D algorithm at high ISR. The complexity of M-D($M = 2$) algorithm is also lower. This is because the length of two parallel surviving paths in the viterbi decoding algorithm is not fixed and can not be controled. Therefore, using viterbi decoding algorithm, we can not avoid the converging of weights along a wrong path at high ISR.

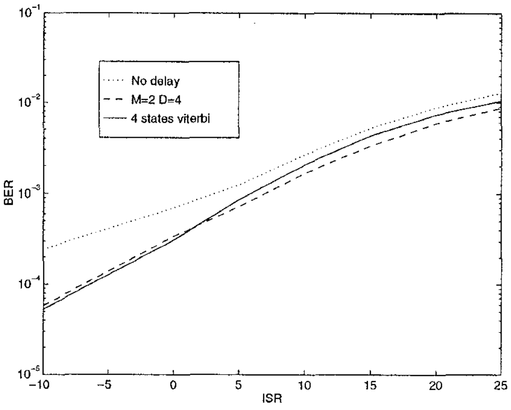

Figure 11. Comparison between D-delay algorithm and VA

## 5. CONCLUSIONS

This paper presents a simultaneous diversity weights updating and decoding technique. M-D decoding algorithm is developed for the binary convolutional codes and TCM codes. It provides instantaneously a set of candidate reference signals for weights tracking and makes a final symbol decision with a $D$ symbol delay based on more reliable accumulated path metrics. It thus significantly 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 $O(M)$ for decoding, which is not much increased from $O(K^2)$ in the conventional decision directed adaptive array system when $M$ is 2.

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

## 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, "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.

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

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