

COMPARISON OF MULTI-CHANNEL ADAPTIVE MLSE EQUALIZERS USING DIFFERENT CHANNEL TRACKING STRATEGIES

Jiunn-Tsair Chen

Joonsuk Kim

John Cioffi

Information Systems Laboratory
Dept. of Electrical Engineering
Stanford University
Stanford, CA 94305

ABSTRACT

In order to gain insights on equalization design in the wireless mobile communication systems, we compare the performance of several multi-channel MLSE equalizers which adaptively track the fast-fading channels. Commonly-used channel tracking schemes, Decision-Directed Recursive Least Square (DD/RLS), Per-Survivor Processing Recursive Least Square (PSP/RLS) and other reduced-complexity MLSE algorithms are considered. Simulation results that illustrate the performance of the equalizers working with various channel tracking schemes are presented.

1. INTRODUCTION

In the mobile communication systems, radio channels vary rapidly with time because of the high mobility of the subscribers. Signals from different paths experience different speeds of phase rotation caused by the Doppler spread. The combined effects of these phase rotations result in rapid fluctuation of the radio channels. We assume a TDMA (Time Division Multiple Access) system, e.g., GSM [1], and we focus on the algorithms of the optimal MLSE equalization. In the fast-fading channels, channels could be quite different at both ends of a TDMA data burst. Therefore, joint channel-data estimation (JCDE) is required. The most straightforward way to track the channel is decision-directed (DD). The decision-directed least mean square (DD/LMS) channel tracking algorithm is proposed in [2]. If the radio channels vary very fast, tentative decision is required to keep up with the channel variation. However, the MLSE equalizer requires a long decision delay to make sure the trellis paths converge. Before trellis paths converge, the tentative decision in the DD may introduce a severe error propagation problem because of the incorrect decision. In an extreme case, the tentative decisions are retrieved with no delay from the survival trellis path with the minimum node metric, and used to update the channel. This algorithm is called the minimum-survivor method in [3].

In addition to the DD, the per-survivor processing (PSP) channel tracking schemes have attracted a lot of attention in recent years, e.g., [4]. In the PSP, channels are updated independently for each survival trellis path and the decision feedback is retrieved from each individual trellis path with no decision feedback delay. The MLSE equalizer is considered more complicated than the MMSE (minimum mean

square estimation) or the ZF (zero-forcing) linear equalizers. The PSP makes the MLSE even more complicated. Several ways to reduce the complexity have been proposed in the literature. One is to reduce the complexity of the MLSE itself by reducing the number of states or the number of trellis paths, and then use the PSP for channel tracking. The DDFSE [6], the RSSE [7] and the (M,L) algorithms [8] have been proposed to reduce the number of states. The M-algorithm [9], the T-algorithm [10] and the (M,L) algorithms [8] can be used to constrain the number of survival paths. The algorithms using the PSP together with the various complexity-reduced MLSEs mentioned above to track the fast-fading channels are discussed in [5, 11]. The other way to reduce the PSP/MLSE complexity is to constrain the number of trellis paths where channel tracking is applied [12], and the PSP is then applied to only these part of the survival paths to update the channel estimates.

This paper is organized as follows. In section 2, we describe a fast-fading specular multi-path channel model. In section 3, we describe several commonly-used adaptive algorithms to be used in a multiple-antenna scenario. In section 4, we compare the BER simulation results of several algorithms described in section 3.

2. FAST-FADING CHANNEL MODEL

Let us first describe the specular multi-path channel model. The radio channel in a wireless communication system is often characterized by a multi-path propagation model. In large cells with high base station antenna platforms, the propagation environment is aptly modeled by a few dominant specular paths, typically 2 to 6. In such a case, the baseband signals received at the antenna array can be expressed as follows

$$\mathbf{x}(t) = \sum_{i=1}^p \mathbf{a}(\theta_i) \beta_i(t) \tilde{s}(t - \tau_i) + \mathbf{n}(t), \quad (1)$$

where $\mathbf{x}(t)$ is the vector of the received baseband signals; $\mathbf{n}(t)$ is the additive Gaussian noise; $\mathbf{a}(\theta_i)$ is the steering vector of a signal arriving from direction θ_i ; β_i is the time-varying path amplitude which is a complex Gaussian random process including both the propagation loss and the signal fading caused by the Doppler spread; $\tilde{s}$ is the transmitted complex baseband signal; τ_i is the propagation delay of the i^{th} path; and p is the total number of paths present in the system.

In a linear system, the transmitted signal s can be represented as the convolution of the data symbols and the pulse-shaping function. Therefore, after sampling, we can express the received signal $\tilde{\mathbf{x}}$ as

$$\begin{aligned}\tilde{\mathbf{x}}_{m \times 1}(k_1) &= \mathbf{H}_{m \times r_l}(k_1)\tilde{\mathbf{s}}_{r_l \times 1}(k_1) + \tilde{\mathbf{n}} & (2) \\ &= \mathbf{A}_{m \times p}\mathbf{B}_{p \times p}(k_1)\mathbf{G}_{p \times r_l}\tilde{\mathbf{s}}_{r_l \times 1}(k_1) + \tilde{\mathbf{n}}(k_1|\beta)\end{aligned}$$

where the subscript denotes the dimension of each matrix or vector; k_1 is the time index; m is the number of antennas; l is the maximum length of the channel impulse response divided by the symbol period T ; r is the oversampling factor; $\mathbf{A} = [\mathbf{a}(\theta_1) \mathbf{a}(\theta_2) \dots \mathbf{a}(\theta_p)]$; $\mathbf{B}(k_1) = \text{diag}\{\boldsymbol{\beta}(k_1)\}$ with $\boldsymbol{\beta}(k_1) = [\beta_1(k_1), \beta_2(k_1) \dots \beta_p(k_1)]^T$ and $\beta_i(k_1)$ being the complex fading amplitudes of the i^{th} path at the k_1^{th} sample; $\mathbf{G}_{ij} = g(t_0 + \frac{i-1}{r}T - \tau_i)$ with $g(\cdot)$ denoting the pulse-shaping function; and $\tilde{\mathbf{n}}$ is the additive Gaussian noise. Thus, we decompose the FIR equivalent channel $\mathbf{H}(k_1)$ into $\mathbf{H}(k_1) = \mathbf{A}\mathbf{B}(k_1)\mathbf{G}$. The column data vector $\tilde{\mathbf{s}}(k_1)$ is the $(k_1 - hr + 1)^{\text{th}}$ column of a Toeplitz matrix $\mathbf{S}(h)$. In $\mathbf{S}(h)$, we have $[s_h, \mathbf{0}_{1 \times (r-1)}, s_{h-1}, \mathbf{0}_{1 \times (r-1)}, \dots, s_{h-l+1}, \mathbf{0}_{1 \times (r-1)}]^T$ as its first column and $[s_h, \mathbf{0}_{1 \times (r-1)}]$ as its first row, where s_h is the transmitted data symbols, and h denotes the maximum possible integer which is not greater than $\frac{k_1}{r}$. The possible values of the data symbol s_h depend on the modulation scheme.

In order to take into account the time correlation of each of the complex path amplitudes, it is more convenient to consider the received data within a time period instead of a single snapshot. Thus after stacking, (3) can be rewritten as

$$\text{vec}\{\mathbf{X}\}_{mrn \times 1} = \mathcal{H}_{mrn \times (m+n)r-1} \mathbf{s}_{(m+n)r-1 \times 1} + \text{vec}\{\mathbf{N}\}, \quad (4)$$

where $\mathbf{X}_{m \times rn} = [\tilde{\mathbf{x}}(k), \tilde{\mathbf{x}}(k+1), \dots, \tilde{\mathbf{x}}(k+rn-1)]$, $\mathbf{N}_{m \times rn} = [\tilde{\mathbf{n}}(k), \tilde{\mathbf{n}}(k+1), \dots, \tilde{\mathbf{n}}(k+rn-1)]$, $\mathbf{s}_{r(n+l)-1 \times 1} = [\mathbf{0}_{1 \times (r-1)}, \mathbf{s}_{h+n-1}, \mathbf{0}_{1 \times (r-1)}, \dots, \mathbf{s}_{h-l+1}, \mathbf{0}_{1 \times (r-1)}]^T$,

$$\mathcal{H}_{mrn \times r(n+l)-1} = \begin{bmatrix} \mathbf{0} & & & & \mathbf{H}(k) \\ & & & \mathbf{H}(k+1) & \\ & & & \mathbf{H}(k+2) & \\ & & \ddots & & \\ & \mathbf{H}(k+r-1) & & & \mathbf{0} \end{bmatrix}, \quad (5)$$

and $\mathbf{H}_{m \times rl}(k) = \mathbf{A}_{m \times p} \mathbf{B}_{p \times p}(k) \mathbf{G}_{p \times rl}$.

3. THE ALGORITHMS UNDER CONSIDERATION

In a TDMA system, data is transmitted in bursts. A training sequence $\mathbf{s}_t$ is embedded in each burst, and data bits are on both sides of the training sequence. The training sequence $\mathbf{s}_t$ is usually used to facilitate identification of the wireless channel. Since the channel varies with time, the channel estimate during the training period turns inaccurate toward either end of the data burst in the MLSE equalizer. Therefore, updating the channel estimate is required for the fast-fading channels.

We summarize the various algorithms considered in this paper as follows:

1. Estimate the FIR channel $\mathbf{H}$ during the training period by solving a system of linear equations

$$\mathbf{F}\text{vec}(\mathbf{H}\mathbf{S}_t) = \mathbf{F}\text{vec}(\mathbf{X}_t), \quad (6)$$

where $\mathbf{S}_t$ is a known Toeplitz data matrix $\mathbf{S}$ consisting of the training symbols; $\mathbf{X}_t$ is the received signals $\mathbf{X}$ during the training period; $\mathbf{F} = \text{diag}(\alpha^{l_t} \mathbf{I}_{mr \times mr}, \dots, \alpha \mathbf{I}_{mr \times mr}, \mathbf{I}_{mr \times mr})$ if tracking forward; $\mathbf{F} = \text{diag}(\mathbf{I}_{mr \times mr}, \alpha \mathbf{I}_{mr \times mr}, \dots, \alpha^{l_t} \mathbf{I}_{mr \times mr})$ if tracking backward; α is the forgetting factor of the RLS algorithm; and l_t is the length of the training sequence. The recursive way to solve (6) is as follows: Initialize the FIR channel estimate of the i^{th} antenna $\mathbf{H}^i$ and the inverse of the covariance matrix of the i^{th} antenna $\mathbf{P}^i$ by setting $\hat{\mathbf{H}}^i(0) = \mathbf{0}_{1 \times r_l}$ and $\mathbf{P}^i(0) = \mathbf{I}_{r_l \times r_l}$, respectively. Then update the Kalman gain vector $\mathbf{g}^i$ and the matrix $\mathbf{P}^i$ recursively by

$$\mathbf{g}^i(h) = \alpha^{-1} \mathbf{P}^i(h) \mathbf{S}_t(h) (\mathbf{I}_{r \times r} + \alpha^{-1} \mathbf{S}_t^*(h) \mathbf{P}^i(h) \mathbf{S}_t(h))^{-1}, (7)$$

$$\mathbf{e}(h) = \mathbf{X}_t^i(h) - \hat{\mathbf{H}}^i(h)\mathbf{S}_t(h), \quad (8)$$

$$\hat{\mathbf{H}}_i(h+1) = \hat{\mathbf{H}}^i(h) + \mathbf{e}(h)(\mathbf{g}^i(h))^*, \quad (9)$$

$$\mathbf{P}^i(h+1) = \alpha^{-1}(\mathbf{I}_{r_l \times r_l} - \mathbf{g}^i(h)\mathbf{S}_t^i(h))\mathbf{P}^i(h), (10)$$

where $\mathbf{e}$ is the error signal; $\mathbf{X}_t^i$ and $\hat{\mathbf{H}}^i$, respectively, represent the i^{th} row of $\mathbf{X}_t$ and $\hat{\mathbf{H}}$; $\mathbf{X}_t(h) = [\tilde{\mathbf{x}}_t(rh), \tilde{\mathbf{x}}_t(rh + 1), \dots, \tilde{\mathbf{x}}_t(rh + r - 1)]$; $\mathbf{S}_t(h) = [\tilde{\mathbf{s}}_t(rh), \tilde{\mathbf{s}}_t(rh + 1), \dots, \tilde{\mathbf{s}}_t(rh + r - 1)]$; and $\tilde{\mathbf{x}}_t(k_1)$ and $\tilde{\mathbf{s}}_t(k_1)$ are defined as $\tilde{\mathbf{x}}(k_1)$ and $\tilde{\mathbf{s}}(k_1)$, respectively, during the training period.

2. Start to update the channel estimates $\hat{\mathbf{H}}$ through each trellis path of the Viterbi algorithm by using the RLS algorithm, when the next transmitted data symbol s_{h-h_d} is outside the training period and is unknown to the receiver, where h_d is the feedback delay. At epoch h , the extra m linear constraints in the RLS algorithm can be written as

$$\hat{\mathbf{H}}_b \mathbf{S}_{\bar{b}}(h - h_d) = \mathbf{X}(h - h_d), \quad (11)$$

where $\hat{\mathbf{H}}_b$ is the FIR channel update of the b^{th} branch; and $\mathbf{S}_{\bar{b}}$ is the data matrix consisting of the feedback data symbols used for the $\bar{b}^{th}$ branch. If the data sequence associated with “the survival path including the $\bar{b}^{th}$ branch” can be written as $(s_{\bar{b},h}, s_{\bar{b},h-1}, \dots)$, then $\mathbf{S}_{\bar{b}}(h - h_d)$ is a Toeplitz matrix with $[\bar{s}_{\bar{b},h-h_d}, \mathbf{0}_{1 \times (r-1)}, \bar{s}_{\bar{b},h-h_d-1}, \mathbf{0}_{1 \times (r-1)}, \dots, \bar{s}_{\bar{b},h-h_d-l+1}, \mathbf{0}_{1 \times (r-1)}]^T$ as its first column and $[\bar{s}_{\bar{b},h-h_d}, \mathbf{0}_{1 \times (r-1)}]$ as its first row. We update $\hat{\mathbf{H}}_b$ as (7) - (10) do, except that $\mathbf{S}_t(h)$ is replaced by $\mathbf{S}_{\bar{b}}(h - h_d)$. If $\bar{b} = b_{min}(h)$, the algorithm is DD and there is only one RLS running to update the channel estimate, where $b_{min}(h)$ is the branch connected to the node with the minimum node metric at epoch h , $\Gamma_{b_{min}}(h)$. If $\bar{b} = b$, the algorithm is PSP and there are n_d RLSs running at the same time along each survival path, where $n_d \geq 1$ is the number of the survival paths. If the decision feedback delay h_d increases, the survival trellis paths converge to the common trellis roots; thus n_d decreases.

3. Reconstruct the received signal $\mathbf{X}_b(h)$ of the b^{th} branch by $\mathbf{X}_b(h) = \hat{\mathbf{H}}_b(h - h_d)\mathbf{S}_b(h)$.
4. Calculate the incremental metrics for the b^{th} branch $\eta_b(h) = \|\mathbf{X}(h) - \mathbf{X}_b(h)\|_F^2$, where $\|\cdot\|_F$ denotes the Frobenius norm.
5. Calculate the node metric for the node connected to the b^{th} branch $\Gamma_b(h) = \eta_b(h) + \Gamma_b(h - 1)$, where $\Gamma_b(h - 1)$ is the node metric of the node on the other side of the b^{th} branch just before epoch h .
6. Assuming each node is connected to n_i branches, pick one branch with the minimum node metric out of n_i branches
7. Repeat Steps 2-6 for all the branches b in epoch h .
8. Make decision of s_{h-h_f} by choosing the trellis path with the minimum node metric $\Gamma_b(h)$ among all the nodes at epoch h . The decision delay h_f is usually no less than $5l$. If the data sequence of the chosen survival trellis path can be expressed as $(\hat{s}_h, \hat{s}_{h-1}, \dots)$, then the final decision of the $(h - h_f)^{th}$ symbol is $\hat{s}_{h-h_f}$.
9. Repeat Steps 2-8 for the next symbol forward starting from the training period until the end of the burst is reached.
10. Repeat Steps 1-8, but backward, starting from the training period until the beginning of the burst is reached.
11. Return to Step 1 for the next burst.

If $\bar{b}(h) = b_{min}(h)$, and $h_d \geq 5l$, the algorithm is DD. If $\bar{b}(h) = b_{min}(h)$, and $h_d = 1$, the algorithm is called the minimum-survivor channel tracking method in [3]. If $\bar{b}(h) = b_{min}(h)$, and $1 < h_d < 5l$, the algorithm is DD with tentative decision. If $\bar{b}(h) = b(h)$, and $h_d = 1$, the algorithm is PSP. If $h_d = 1$ and $\bar{b}(h)$ is defined as

$$\bar{b}(h) = \begin{cases} b(h) & \text{if } \Gamma_b(h - 1) \text{ is among } n_p \text{ smallest metrics} \\ b_{min}(h) & \text{otherwise,} \end{cases} \quad (12)$$

where n_p is a fixed number chosen to be less than the number of nodes, this algorithm is proposed in [12] as a method between the PSP and the DD with tentative decision. We call this algorithm the Raheli's method in the rest of this paper. The DDFSE [6] can also be used to simplify MLSE, and the PSP tracking is then applied to thus simplified MLSE.

4. SIMULATION RESULTS

Simulations were conducted to compare the performance of the adaptive MLSE algorithms. For the MLSE algorithms, we tested several channel tracking schemes, including DD/RLS, DD/RLS with tentative decision, DD/RLS with the minimum-survivor method, PSP/RLS, DDFSE/PSP/RLS, the Raheli's method/RLS and then compared their simulation results with those of the train-and-freeze approach. In the simulations, the GMSK modulation scheme used in the GSM [1] is tested. We approximated the GMSK as a linear modulation scheme, and we tried the simulation on a 3-tap channel model which is made similar to the Typical Urban(TU) channel models in [1]. The path delays are 0, $0.4T$ and $1.3T$, respectively, where

T is the symbol period. The path DOAs are -30° , 0° , and 30° , respectively; and the path weights are $0dB$, $-6dB$, and $-10dB$, respectively. The normalized Doppler spread are assumed to be $f_D T = 6 \times 10^{-3}$. We used a uniform linear array with 4 isotropic elements, spaced half a wavelength apart. In the Viterbi equalizer, we used 16 states.

The BER performances of the MLSE algorithms with various channel tracking schemes are shown in Figs. 1. Fig. 2 compares the BER performance of the DD/RLS with various decision feedback delays. Finally, the performances of the DDFSE/PSP/RLS and the Raheli's method/RLS are presented in Figs. 3 and 4, respectively. The DDFSE/PSP/RLS approach first uses DDFSE to simplify the MLSE by reducing the number of states in the Viterbi algorithm, then applies the PSP/RLS channel tracking. The Raheli's method/RLS is described in Section 3.1.

According to Fig. 1, as expected, the PSP/RLS performs better than the DD/RLS, and both the PSP/RLS and the DD/RLS perform better than the train-and-freeze approach. Since channels are fast time-varying, severe channel mismatches cause the BER of almost all the algorithms to encounter error flooring at high SNR. Better performance is associated with higher complexity of the algorithms. The high complexity of the PSP makes it impractical to implement at the current DSP processing speed. Decision-directed tracking is simple but it may suffer two problems: 1) The decision feedback delay in the fast-fading channel will cause channel mismatch. Therefore, we have to reduce decision feedback delay by making tentative decision. 2) The DD with tentative decision will cause error propagation if the decision feedback delay is too short. This is because we try to make decision of a specific symbol before most energy of that symbol is received, and thus the tentative decision is likely to be incorrect. There is an optimal decision feedback delay working to minimize the channel estimation lag and the error propagation at the same time. Fig. 2 is meant to search for this optimal decision delay. In the scenario we set up, we found both the case with long decision feedback delay and the case with no decision feedback delay (the minimum-survivor method) to perform much worse than the PSP/RLS. As shown in Fig. 2, the optimal decision feedback delay is two symbols.

Other algorithms with higher complexity than the DD/RLS are also possible for simplifying the PSP/RLS. Fig. 3 shows that the DDFSE/PSP/RLS can perform as well as the PSP/RLS if the number of states is reduced by half, and that yet its performance starts to degrade dramatically if the complexity is further reduced. Similarly, Fig. 4 shows that the Raheli's method/RLS can perform as well as the PSP/RLS if the channel-updating is only restricted to one half or one quarter of the survival paths, and that yet its performance starts to degrade if we further reduce the number of survival paths employing channel updating.

REFERENCES

- [1] European Telecommunications Standard Institute (ETSI), Rec. ETSI/GSM 05.02, 1990.
- [2] Magee, F.R.; Proakis, J.G., "Adaptive maximum-likelihood sequence estimation for digital signaling in the presence

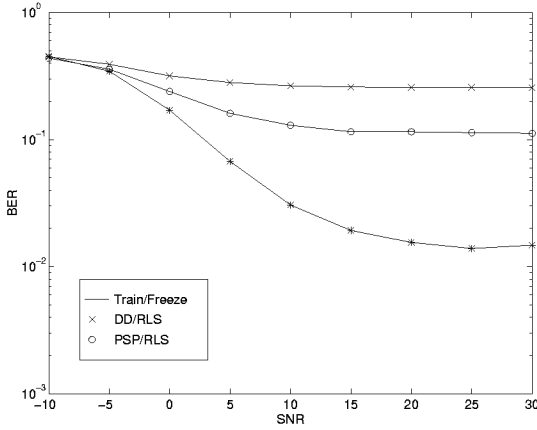


Figure 1. Comparison of the MLSE algorithms with various channel tracking schemes.

of intersymbol interference,” *IEEE Trans. on Information Theory*, Vol. IT-19, no.1, Jan. 1973, pp. 120-24.

- [3] Castellini, G.; Conti, F.; Del Re, E.; Pierucci, L., “A continuously adaptive MLSE receiver for mobile communications: algorithm and performance,” *IEEE Trans. on Communications*, Vol. 45, no.1, Jan. 1997, pp. 80-89.
- [4] Polydoros, A.; Chugg, K.M., “Per-survivor processing (PSP),” *Proc. PIMRC*, Helsinki, Finland, 1997.
- [5] Raheli, R.; Polydoros, A.; Tzou, Ching-Kae, “Per-Survivor Processing: a general approach to MLSE in uncertain environments,” *IEEE Trans. on Communications*, Vol. 43, no.2-4, Feb.-Apr. 1995, pp. 354-64.
- [6] Duel-Hallen, A.; Heegard, C., “Delayed decision-feedback sequence estimation,” *IEEE Trans. on Communications*, Vol. 37, no.5, May. 1989, pp. 428-36.
- [7] Eyuboglu, M.V.; Qureshi, S.U.H., “Reduced-state sequence estimation for coded modulation of intersymbol interference channels,” *IEEE Journal on Selected Areas in Communications*, Vol. 7, no.6, Aug. 1989, pp. 989-95.
- [8] Seshadri, N.; Anderson, J. B., “Decoding of severely filtered modulation codes using the (M, L) algorithm,” *IEEE Journal on Selected Areas in Communications*, Vol. 7, no.6, Aug. 1989, pp. 1006-16.
- [9] Benelli, G.; Garzelli, A.; Salvi, F., “Simplified Viterbi processors for the GSM Pan-European cellular communication system,” *IEEE Trans. Vehicular Tech.*, Vol. VT-43, no.4, Nov. 1994, pp. 870-8.
- [10] Simmons, S.J., “Breadth-first trellis decoding with adaptive effort,” *IEEE Trans. on Communications*, Vol. 38, no.1, Aug. 1990, pp. 3-12.
- [11] Kubo, H.; Murakami, K.; Fujino, T., “An adaptive maximum-likelihood sequence estimator for fast time-varying intersymbol interference channels,” *IEEE Trans. on Communications*, Vol. 42, no.2-4, Feb.-Apr. 1994, pp. 1872-80.
- [12] Raheli, R.; Marino, G.; Castoldi, P., “Per-survivor processing and tentative decisions: what is in between,” *IEEE Trans. on Communications*, Vol. 44, no.2, Feb. 1996, pp. 127-29.

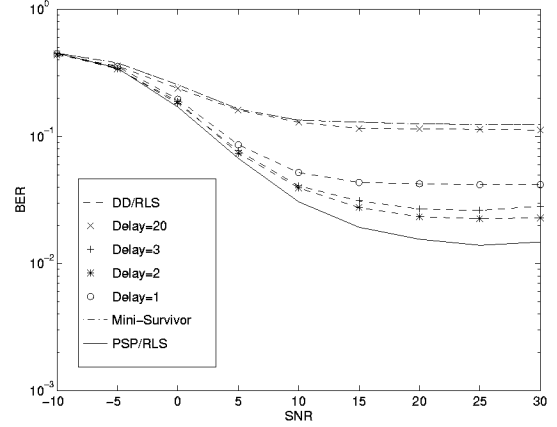


Figure 2. The BER performance of the DD/RLS with various tentative delays.

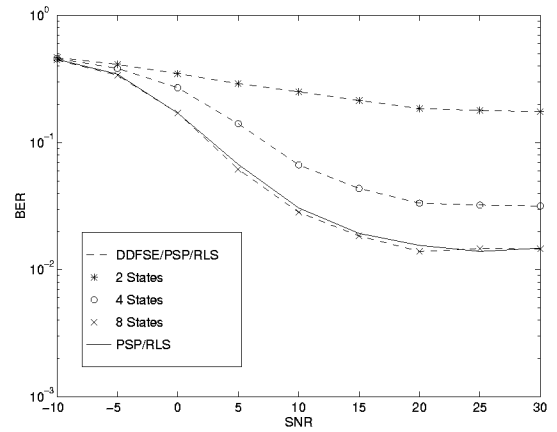


Figure 3. The BER performance of the DDFSE/PSP/RLS with various numbers of states.

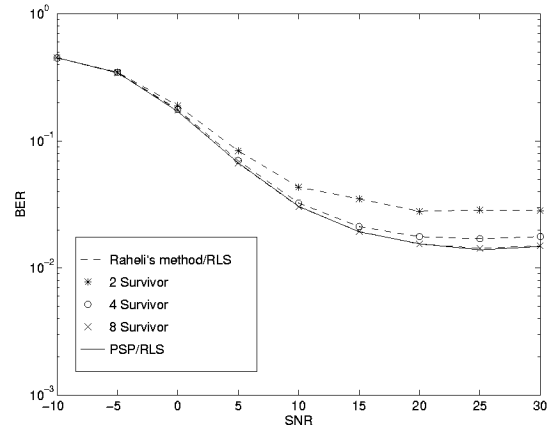


Figure 4. The BER performance of the Raheli's method/RLS with various numbers of survival paths.