Turk J Elec Eng & Comp Sci, Vol.20, No.5, 2012, © TÜBİTAK
doi:10.3906/elk-1012-941

A new multipath channel transformation technique

Tuğrul ÇAVDAR
Department of Computer Engineering, Karadeniz Technical University, Trabzon-TURKEY
e-mail: ulduz@ktu.edu.tr

Received: 26.12.2010

Abstract

A new channel transformation technique is introduced in this paper The technique is employed on the receiver side and has 2 filters. The first filter is the inverse complex conjugate of the estimated channel. The profile of this filter's output is triangular. The taps of the second filter are calculated according to the output of the first filter to produce an exponentially decreasing profile. This technique transforms most channel profiles to exponentially decreasing channel profiles. Thus equalizers adopting the proposed technique may have better bit error rates and lower computational complexity on frequency-selective channels. This paper evaluates the efficiency of the proposed technique on a sequential channel equalizer in different channel conditions and modulations to compare the results with others reported in the literature. Computational complexity and bit error rates under different circumstances are provided. The channel transformation technique exhibits performance levels close to the optimum results of the Viterbi algorithm and proves to be a useful tool for equalizers.

Key Words: *Channel transformation, sequential decoding, frequency-selective channel, equalization*

1. Introduction

It is not easy to decide which symbol has been received if there has been interference from delayed copies (echoes) of the previous symbols, especially in low signal-to-noise ratios (SNRs). Wireless transmission over multipath channels spreads the unit impulse over time, which in turn causes intersymbol interference (ISI). Triangular channel profiles lead to worst-case scenarios for mobile wireless communication systems. In such channels, most equalizers suffer from ISI at low SNRs and are unable to deal with it. Viterbi equalizers set an upper limit on the bit error rate (BER) performance since they are the optimum algorithm. However, the Viterbi algorithm [1] brings a computational burden intolerable for today's systems with high data rates. Sequential decoding algorithms are much faster than the Viterbi algorithm as equalizers, but their BER performances are worse than that of the Viterbi algorithm. In the event of exponentially decreasing channel profiles, channel equalization techniques provide better performance. However, if the channel profile is not exponentially decreasing (i.e. the worst channel profile of Proakis is triangular [2]), namely if the energy of the previous symbols is higher than that of the current symbol, it is difficult to prevent wrong decisions. Equalization techniques produce better performance in cases where the first tap of a finite-state machine (FSM) [3] is larger than the other taps corresponding to path gains of multipath channels. Several techniques have been developed to improve channel equalizers by transforming the channel. One of them is the channel matched filter (CMF) [4-6], which leads to dense multipath energy on one tap. Qureshi and Newhall developed an adaptive receiver by limiting the time spread of the channel and mitigating time variation [7]. Lee and Cioffi optimized the equalizer and the target response, and they added a degree of freedom for choosing the position of the reference tap in the target response [8]. Beare chose a desired impulse response for a given channel rather than a complete optimization [9]. Falconer and Magee used a fixed-energy constraint for better impulse response by minimizing the mean square error between the equalizer output and the target channel response to a unit-energy constraint on the target response [10]. Messerschmitt held the first nonzero term in the desired impulse response fixed and minimized the noise variance [11]. Some time-domain equalizer designs, placed in cascade with the channel to produce an effective impulse response, were reinvestigated to mitigate the ISI produced due to an inadequate cyclic prefix in multicarrier modulation systems [12-15]. Each of these designs adopts a particular cost function, which may be general or system-specific, to perform efficient channel shortening. Husain et al. showed that channel shortening can help to mitigate the ISI caused by the long operational window of the rake receiver [16,17]. Another channel shortening design that directly minimizes the BER of cyclic prefix-based systems was proposed in [18]. A prefilter was used to concentrate the effective channel power in a small number of taps followed by a reduced-complexity maximum a posteriori probability equalizer/decoder to produce soft decisions [19]. Samanta et al. employed a space-time equalizer before a Viterbi equalizer to joint coantenna/cochannel-interference suppression and shortening of the effective channel [20]. Toker and Altın recently proposed a blind channel shortening equalizer depending on the minimization of a cost function defined as the sum-squared difference of the autocorrelations of the shortened channel impulse response and a target impulse response [21]. Karaboğa and Çetinkaya recently defined an adaptive filter to remove noise from the channel [22].

In the present paper, a new technique transforming most channel profiles to exponentially decreasing profiles is developed. The performance of the proposed technique is evaluated on a sequential channel equalizer with the sequential estimation algorithm based on branch metric (SEABM) [23-25]. Sequential decoding algorithms have less computational complexity than Viterbi decoding [1], turbo decoding [26], and other decoding algorithms. However, sequential equalizers possess a BER performance close to that of the Viterbi equalizer, which is the optimum algorithm. The proposed channel transformation technique (CTT) helps low-complexity equalizers like sequential decoding to produce BER performances closer to optimum.

2. The channel transformation technique

The communication media will be considered as a discrete-time channel model, defined as a FSM by Forney [3], with additive white Gaussian noise. The channel is assumed to be frequency-selective and multipath. The block diagram of the communication system and the FSM are given in Figures 1 and 2a, respectively.

The received signal over frequency-selective channels is:

$$y_k = \sum_{j=0}^{W-1} c_j x_{k-j} + n_k, \tag{1}$$

where y is the output of the channel, c is the channel tap, x is the transmitted signal, W is the number of taps, and n is the noise. The CMF has the same number of taps as the channel, and the taps of the CMF are

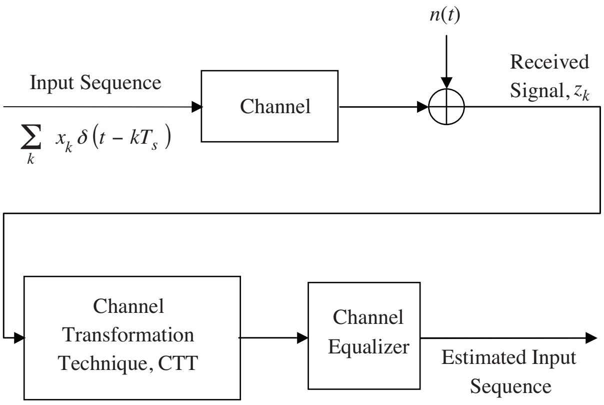

Figure 1. Transmission model.

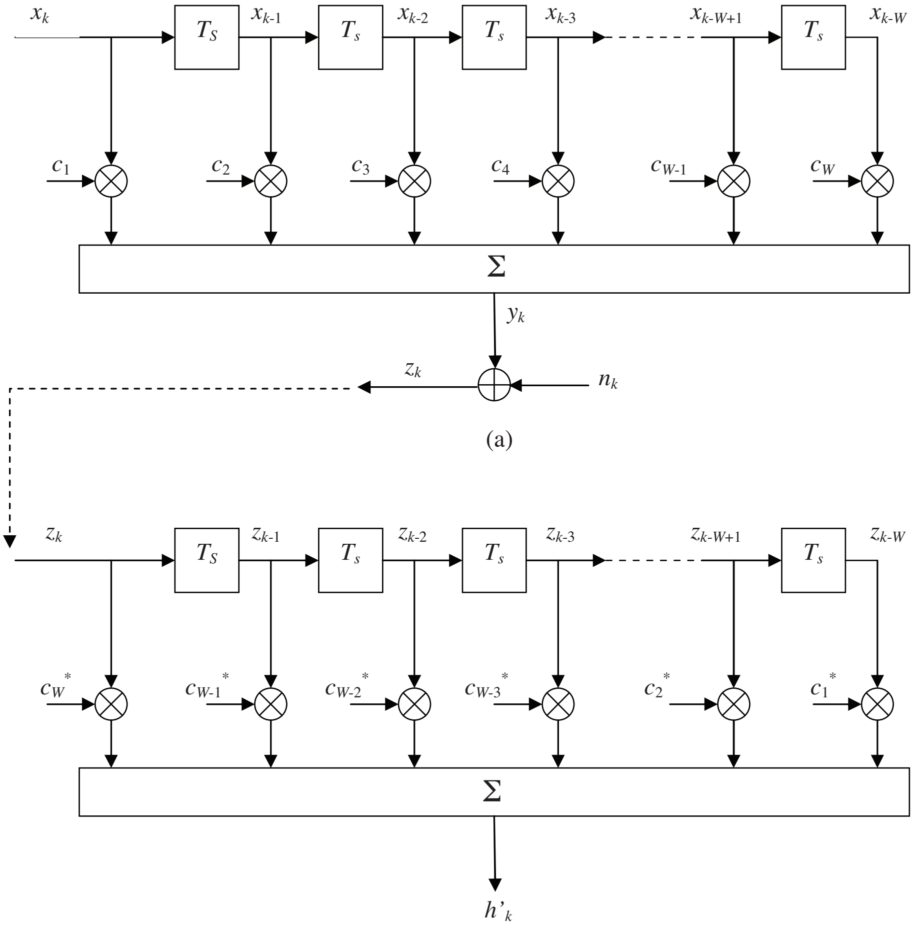

Figure 2. a) Multipath channel modeled as finite-state machine by Forney [3] and b) the CMF.

complex conjugates of those of the channel. The CMF is given in Figure 2b. The output of the CMF is:

$$h'_k = \sum_{i=0}^{W-1} c*_{W-1-i}\, y_{k-i} = \sum_{i=0}^{W-1} c*_{W-1-i} \left(\sum_{j=0}^{W-1} c_j x_{k-i-j} + n_{k-i} \right), \tag{2}$$

or

$$h'_k = \sum_{i=0}^{W-1} \sum_{j=0}^{W-1} c*_{W-1-i}\, c_j x_{k-i-j} + \sum_{i=0}^{W-1} c*_{W-1-i}\, n_{k-i}, \tag{3}$$

where $c*$ is the tap of the CMF and h' is the output of the CMF. The convolution of the channel with the CMF can be seen from the first part of the addition. If the convolution is investigated in the z-domain, the channel without noise is:

$$Y(z) = \sum_{i=0}^{W-1} c_i z^i, \tag{4}$$

and the output of the CMF is:

$$H'(z) = \sum_{i=0}^{W-1} c*_{W-1-i}\, z^i. \tag{5}$$

The revealing filter, g, which is the convolution of the channel and the CMF, is:

$$G(z) = Y(z)H'(z) = \sum_{i=0}^{W-1} \sum_{j=0}^{W-1} c_i c*_{W-1-j}\, z^{i+j}. \tag{6}$$

For a 5-tap channel ($W = 5$), the 9 tap values of filter g, formed by convoluting the 5-tap channel with the 5-tap CMF, are:

$$\begin{aligned}
g_0 &= c_0 c_4^* \\
g_1 &= c_0 c_3^* + c_1 c_4^* \\
g_2 &= c_0 c_2^* + c_1 c_3^* + c_2 c_4^* \\
g_3 &= c_0 c_1^* + c_1 c_2^* + c_2 c_3^* + c_3 c_4^* \\
g_4 &= c_0 c_0^* + c_1 c_1^* + c_2 c_2^* + c_3 c_3^* + c_4 c_4^* \\
g_5 &= c_1 c_0^* + c_2 c_1^* + c_3 c_2^* + c_4 c_3^* \\
g_6 &= c_2 c_0^* + c_3 c_1^* + c_4 c_2^* \\
g_7 &= c_3 c_0^* + c_4 c_1^* \\
g_8 &= c_4 c_0^*
\end{aligned} \tag{7}$$

The length of the g filter is $2W-1$ (i.e. 5 for a 3-tap channel) because 2 W-tap filters (channel and CMF) are convoluted. The center point of this convolution is not only the largest component but also a real value. This center combines all of the multipath energy of the channel. The ISI components at both sides of the center are also symmetrical. The error probability of estimation of the center tap symbol decreases if the received signal is passed through the CMF and the equalization algorithm uses the g filter to produce a hypothesis symbol. This convolution is shown in Figure 3a with the channel profile and the indicated symbol to be estimated. Figure 3b shows the CMF, which is the inverse complex conjugate of the channel. Figure 3c shows the g filter. It is clear from Eq. (7) that the center tap is the largest of all and makes it easier to estimate the symbol of the center tap. However, the problem is that the taps on the right side of the center in Figure 3a correspond to the symbols sent before the center tap symbol, and the equalization technique has already estimated these symbols. Nevertheless, the technique will not have estimated the symbols of the taps on the left side of the center. It thus becomes impossible for the technique to find hypothesis channel input by using the g filter only.

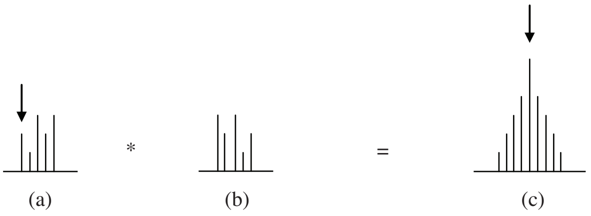

Figure 3. a) The channel profile, b) the CMF's profile, and c) the g filter resulting from the convolution of the channel and the CMF.

In the event that the taps on the left side of the center are zero, the equalization algorithm does not need to know the symbols of these taps. Since the symbols of the taps on the right side of the center are already estimated, the symbol of the central tap can be estimated by using the symbols on the right side. The g filter is multiplied by a new d filter, called the final filter, which has the same number of taps as the channel yielding the taps shown below.

$$\begin{array}{ll}
1. & g_0d_0 \\
2. & g_0d_1+g_1d_0 \\
3. & g_0d_2+g_1d_1+g_2d_0 \\
4. & g_0d_3+g_1d_2+g_2d_1+g_3d_0 \\
5. & g_0d_4+g_1d_3+g_2d_2+g_3d_1+g_4d_0 \\
6. & g_1d_4+g_2d_3+g_3d_2+g_4d_1+g_5d_0 \\
7. & g_2d_4+g_3d_3+g_4d_2+g_5d_1+g_6d_0 \\
8. & g_3d_4+g_4d_3+g_5d_2+g_6d_1+g_7d_0 \\
9. & g_4d_4+g_5d_3+g_6d_2+g_7d_1+g_8d_0 \\
10. & g_5d_4+g_6d_3+g_7d_2+g_8d_1 \\
11. & g_6d_4+g_7d_3+g_8d_2 \\
12. & g_7d_4+g_8d_3 \\
13. & g_8d_4
\end{array} \tag{8}$$

The length of the final filter is $3W-2$ (i.e. 7 for a 3-tap channel) because the $(2W-1)$-tap$g$ filter is convoluted with the W-tap d filter. The $(2W-1)$th tap becomes the largest if the taps of the d filter are selected to make taps# $[W\sim(2W-2)]$ zero in this filter. As the rank of the matrix, resulting from the W-parameter $(W-1)$ equations, is $(W-1)$, the $(W-1)$ taps of the d filter are calculated in terms of the other taps. In order to make taps# $[W\sim(2W-2)]$ zero,

$$\begin{bmatrix} g_3 & g_4 & g_5 & g_6 \\ g_2 & g_3 & g_4 & g_5 \\ g_1 & g_2 & g_3 & g_4 \\ g_0 & g_1 & g_2 & g_3 \end{bmatrix} \begin{bmatrix} d_4 \\ d_3 \\ d_2 \\ d_1 \end{bmatrix} = -\begin{bmatrix} g_7 \\ g_6 \\ g_5 \\ g_4 \end{bmatrix} d_0. \tag{9}$$

By using the Gaussian elimination method, the taps of d filter are computed as:

$$\begin{bmatrix} d_4 \\ d_3 \\ d_2 \\ d_1 \end{bmatrix} = \begin{bmatrix} e_{15} - e_{13}e_{35} - (e_{14} - e_{13}e_{34})\frac{e_{45}}{e_{44}} \\ e_{25} - e_{24}\frac{e_{45}}{e_{44}} \\ e_{35} - e_{34}\frac{e_{45}}{e_{44}} \\ \frac{e_{45}}{e_{44}} \end{bmatrix} d_0, \tag{10}$$

where e_{ij} is defined in the Appendix.

The new filter resulting from the convolution of g and d is shown in Figures 4a-4c. The symbol of the tap marked with an arrow in Figure 4c is the symbol to be estimated by the equalization algorithm.

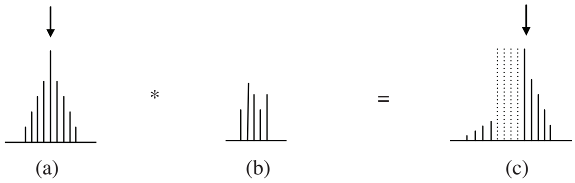

Figure 4. a) The profile of the g filter, b) the d filter, and c) the convolution of the g and d filters.

Taps# [1 ∼ ($W-$ 1)] would be so small that they can be neglected in the event that taps# [W ∼ (2$W-$ 2)] of the final filter are zero. This creates a new profile corresponding to the transformed channel with exponentially decreasing taps# [(2$W-$ 1) ∼ (3$W-$ 2)]. The equalization technique does not need to know channel inputs after the symbol of the (2$W-$ 1)th tap. By assuming that taps# [1 ∼ (2$W-$ 2)] are all zero, the channel equalizer can easily estimate the symbol corresponding to tap# (2$W-$ 1). Nevertheless, it does not estimate the current channel input, but rather the past (2$W-$ 2)th symbol before the actual one for W = 5. Because the transformed channel has an exponentially decreasing good profile from taps# (2$W-$ 1) to (3$W-$ 2), the performance of the equalization algorithm increases.

Figure 5 illustrates the receiver structure with the channel transformation filters. The estimated channel taps in the output of the matched filter are convoluted with the CMF and d, and the tentative decisions are applied to the input of [estimated channel]*CMF*d. After the output of this filter is compared to the output of [real channel]*CMF*d, the difference between the 2 outputs is applied to the channel equalization technique to calculate the metric of the equalization.

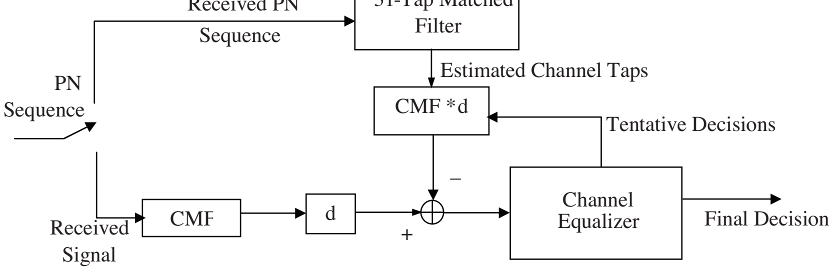

Figure 5. The receiver structure with the CMF and the d filter.

3. Simulations

Simulations are carried out to show that the CTT transforms most channel profiles to an exponentially decreasing good channel profile. The channel is estimated before the equalization process, as shown for the receiver structure in Figure 5. In order to estimate the channel, a 31-bit pseudorandom noise (PN) sequence, the first 31 bits of the data packet of the HiperLAN/1 standard, is sent as a training packet. This sequence is [0 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1]. The received training sequence is passed over a 31-tap matched filter whose taps are inverse complex conjugates of the PN sequence. The channel tap is computed by windowing the output of the matched filter. The taps of the CMF and d are then determined. Computer simulations for different modulations (BPSK and 8-PSK) are performed using 10,000 modulated blocks of 146 symbols each. The symbols are then transmitted over Rayleigh channels with additive white Gaussian noise. The channel is changed for every data packet.

An 8-ray Rayleigh channel model [23-25] is used for computer simulations. This model is mathematically shown below for both in-phase and quadrature components.

$$I = \sum_{j=1}^{8} A \cos\left[\alpha_j + \frac{2\pi v l T_s}{\lambda}\cos\theta_j\right] \tag{11}$$

$$Q = \sum_{j=1}^{8} A \sin\left[\alpha_j + \frac{2\pi v l T_s}{\lambda}\cos\theta_j\right] \tag{12}$$

Here $\theta_j = 2\pi(j-1)/8$ is the arrival angle for the$j$th ray, l represents the simulation iteration number, T_s is the sampling period, v is the user speed, α_j is the initial arrival phase that is uniformly distributed between 0 and 2π rad, λ is the carrier wavelength, and A is the amplitude.

Two 5-tap Rayleigh channel profiles are chosen for comparisons: 0.29, 0.50, 0.58, 0.50, and 0.29 for the worst channel scenario mentioned in [2] and 1.0, 0.39, 0.15, 0.062, and 0.024 in exponentially decreasing form for the good channel profile. By keeping tap amplitudes fixed and varying their phases randomly, 1 million channels are produced; the taps of the g, d, and $g*d$ filters of these 1 million channels are calculated and the average values of these amplitudes are given in the Table. Both the left and right sides of tap g_5 are symmetrical, as seen from the Table, and $(g*d)_{9-13}$ has an exponentially decreasing profile. The taps of filter d are calculated according to filter g. $(g*d)_5$, $(g*d)_6$, $(g*d)_7$, and $(g*d)_8$ become almost zero, and $(g*d)_9$, $(g*d)_{10}$, $(g*d)_{11}$, $(g*d)_{12}$, and $(g*d)_{13}$ exhibit an exponentially decreasing channel profile. $(g*d)_1$, $(g*d)_2$, $(g*d)_3$, and $(g*d)_4$ are so small that they can be neglected.

The computations in Eqs. (8) and (9) and the Appendix are performed for every channel estimation. They are performed once for each channel until the next channel estimation. The computational burden brought about by the CTT is not greater than that of Viterbi decoding, Turbo decoding, or the other algorithms. The computational complexity of the CTT is shown in Figures 6a-6d. The SEABM with the CTT performs about 6000 computations per packet, while the Viterbi algorithm performs about 80 million computations per packet in the circumstances of Figure 6d. In 8-PSK systems, the number of computations due to the CTT is almost the same as those without the CTT. In high M-ary systems, the CTT does not bring an extra computation burden to sequential decoding algorithms. However, it improves the BER performance, as seen in Figures 7a-7d and 8a-8d, while the computational complexity of other decoding algorithms increases exponentially.

Table. The average amplitudes of tap values of g, d and g*d filters after 1 million Rayleigh channels are produced from the worst channel profile.

c_1	0.29		Average channel profile
c_2	0.5		
c_3	0.58		
c_4	0.5		
c_5	0.29		
g_1	0.538751	$g_0 = c_0c_4^*$	Average CMF output profile (triangular profile)
g_2	1.385884	$g_1 = c_0c_3^* + c_1c_4^*$	
g_3	2.325746	$g_2 = c_0c_2^* + c_1c_3^* + c_2c_4^*$	
g_4	3.001188	$g_3 = c_0c_1^* + c_1c_2^* + c_2c_3^* + c_3c_4^*$	
g_5	8.050619	$g_4 = c_0c_0^* + c_1c_1^* + c_2c_2^* + c_3c_3^* + c_4c_4^*$	
g_6	3.001188	$g_5 = c_1c_0^* + c_2c_1^* + c_3c_2^* + c_4c_3^*$	
g_7	2.325746	$g_6 = c_2c_0^* + c_3c_1^* + c_4c_2^*$	
g_8	1.385884	$g_7 = c_3c_0^* + c_4c_1^*$	
g_9	0.538751	$g_8 = c_4c_0^*$	
$(g^*d)_1$	0.538751		Total output profile ($(g^*d)_{9\text{-}13}$ has an exponentially decreasing profile)
$(g^*d)_2$	1.866981		
$(g^*d)_3$	3.880042		
$(g^*d)_4$	6.371522		
$(g^*d)_5$	0.000001		
$(g^*d)_6$	0.000002		
$(g^*d)_7$	0.000003		
$(g^*d)_8$	0.000003		
$(g^*d)_9$	62.556370	$g_4d_4 + g_5d_3 + g_6d_2 + g_7d_1 + g_8d_0$	
$(g^*d)_{10}$	23.676817	$g_5d_4 + g_6d_3 + g_7d_2 + g_8d_1$	
$(g^*d)_{11}$	19.107201	$g_6d_4 + g_7d_3 + g_8d_2$	
$(g^*d)_{12}$	12.735394	$g_7d_4 + g_8d_3$	
$(g^*d)_{13}$	5.180188	g_8d_4	

The BER performance of the SEABM with and without the CTT is compared to that of the optimum Viterbi algorithm in Figures 7-10. Figure 7 indicates the performance over the worst channel profile given in [2] without Doppler effect. The sequential decoding gives better results with the CTT, but these results are still worse than those of the Viterbi algorithm. As the ISI increases, the CTT improves the BER by about 1 dB for ISI = 3 and by 2-3 dB for ISI = 5. Since the Viterbi algorithm takes much longer to simulate under 8-PSK modulation and 5-symbol ISI, Figure 7d does not show the results of the Viterbi algorithm. Figure 8 presents performances over the worst channel profile with Doppler effect. Here, the receiver moves at 120 km/h, such that v = 120 km/h in Eqs. (11) and (12). The results in Figure 8 are worse than those of Figure 7 due to the Doppler effect. The CTT improves the BER by about 6-7 dB in BPSK modulation and by 3 dB in

8-PSK modulation. Figures 9 and 10 show performances over exponentially decreasing channel profiles without Doppler effect and with Doppler effect ($v = 120$ km/h), respectively. The Doppler effect impairs the BER by about 6-8 dB in BPSK and by 4 dB in 8-PSK modulations. The results with and without the CTT are almost the same because the channel already has an exponentially decreasing good profile before application of the CTT.

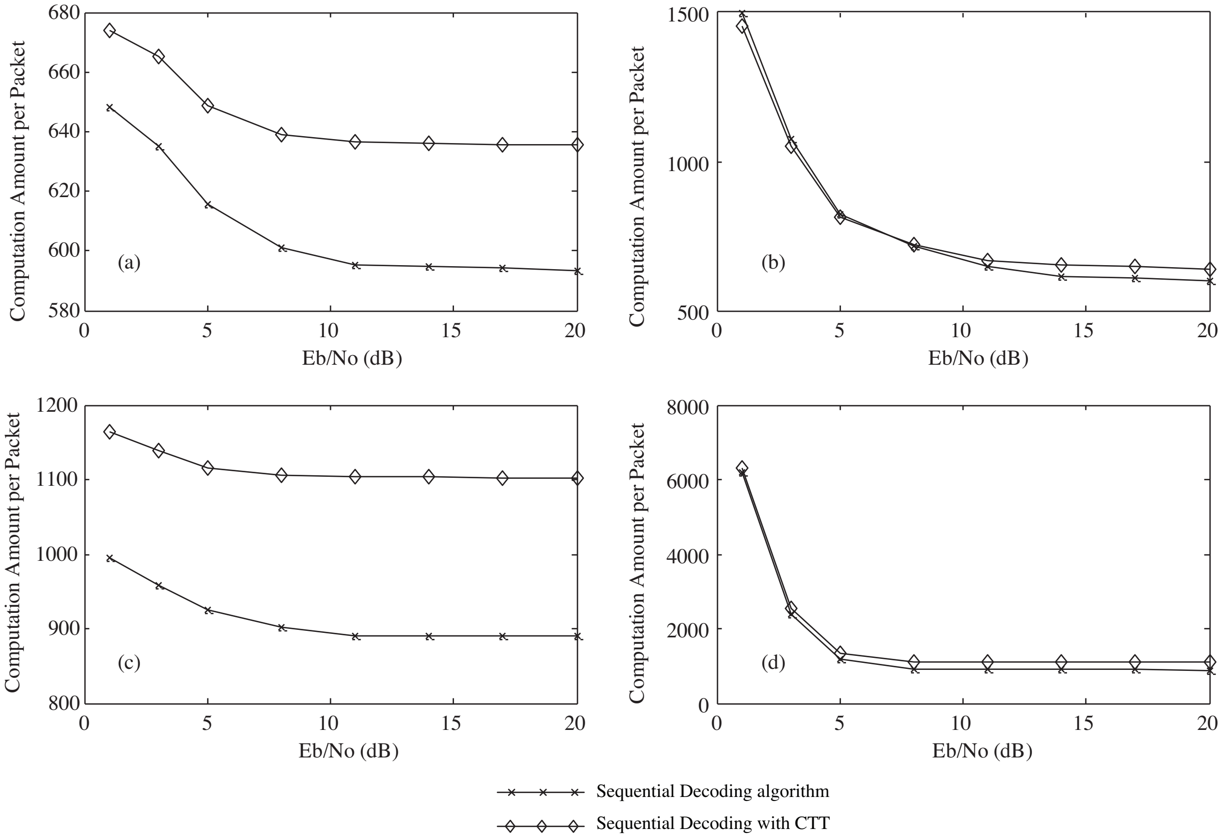

Figure 6. The computation amounts of the SEABM in the worst channel profile per packet both with and without CTT (every addition and multiplication is assumed to be one computation): a) ISI = 3, BPSK, Viterbi (444 + 4800) computations; b) ISI = 3, 8-PSK, Viterbi (444 + 1,228,800) computations; c) ISI = 5, BPSK, Viterbi (740 + 19,456) computations; and d) ISI = 5, 8-PSK, Viterbi (740 + 79,691,776) computations.

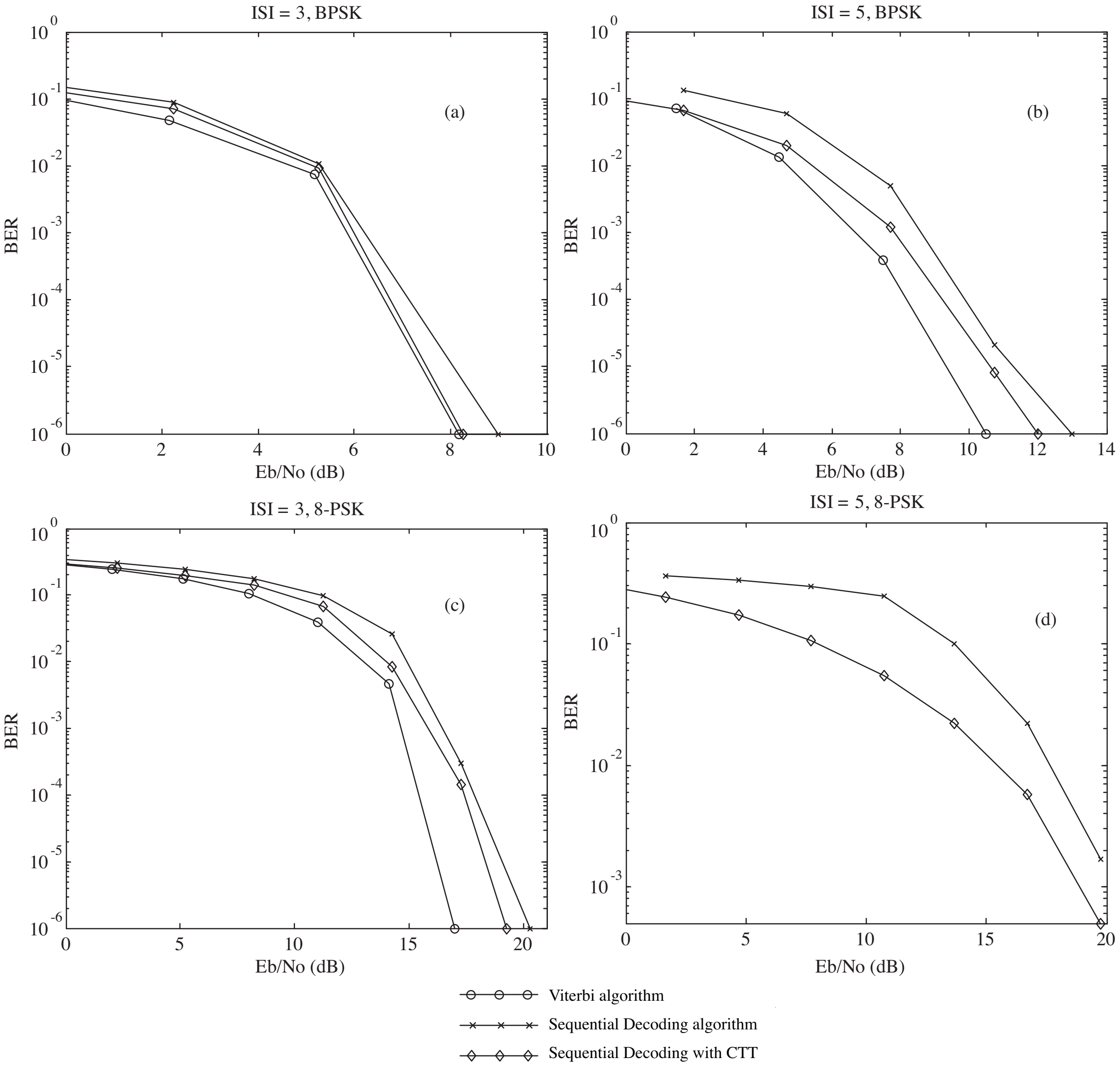

Figure 7. BER performances (the worst channel profile of Proakis, no Doppler effect): a) ISI = 3, BPSK, Viterbi (444 + 4800) computations; b) ISI = 3, 8-PSK, Viterbi (444 + 1,228,800) computations; c) ISI = 5, BPSK, Viterbi (740 + 19,456) computations; and d) ISI = 5, 8-PSK, Viterbi (740 + 79,691,776) computations.

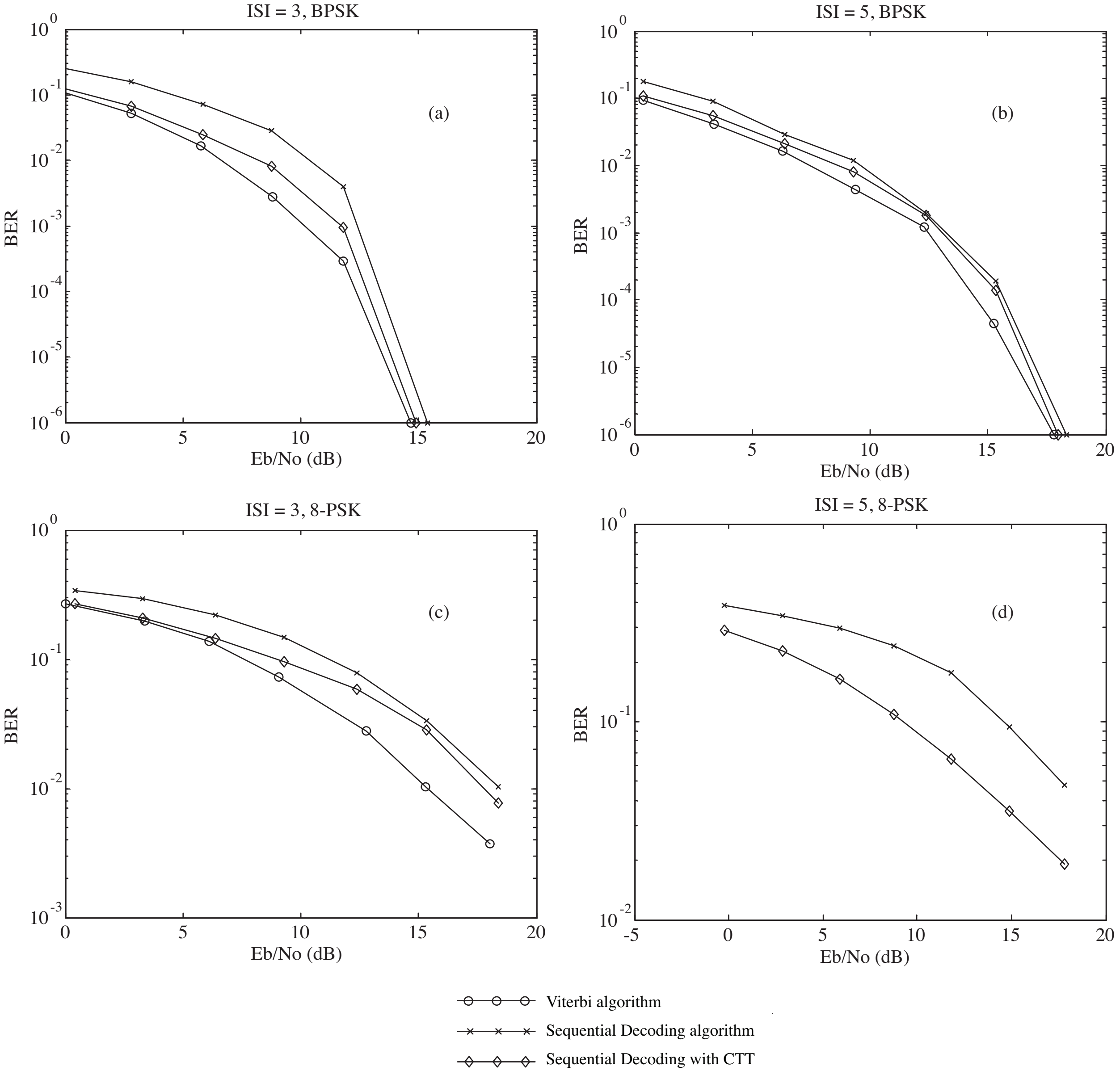

Figure 8. BER performances (the worst channel profile of Proakis with Doppler effect): a) ISI = 3, BPSK, Viterbi (444 + 4800) computations; b) ISI = 3, 8-PSK, Viterbi (444 + 1,228,800) computations; c) ISI = 5, BPSK, Viterbi (740 + 19,456) computations; and d) ISI = 5, 8-PSK, Viterbi (740 + 79,691,776) computations.

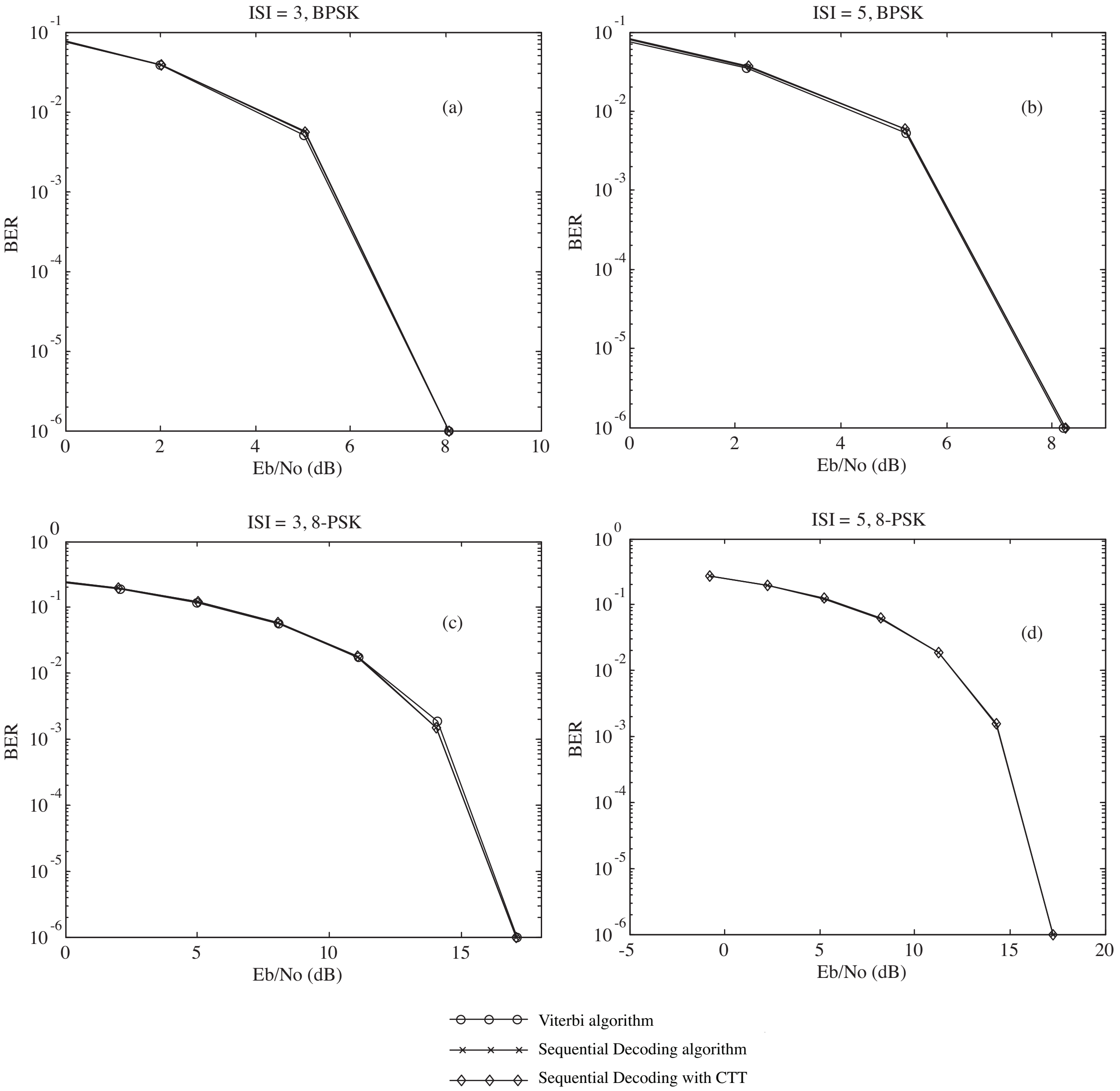

Figure 9. BER performances (exponentially decreasing good channel profile, no Doppler effect): a) ISI = 3, BPSK, Viterbi (444 + 4800) computations; b) ISI = 3, 8-PSK, Viterbi (444 + 1,228,800) computations; c) ISI = 5, BPSK, Viterbi (740 + 19,456) computations; and d) ISI = 5, 8-PSK, Viterbi (740 + 79,691,776) computations.

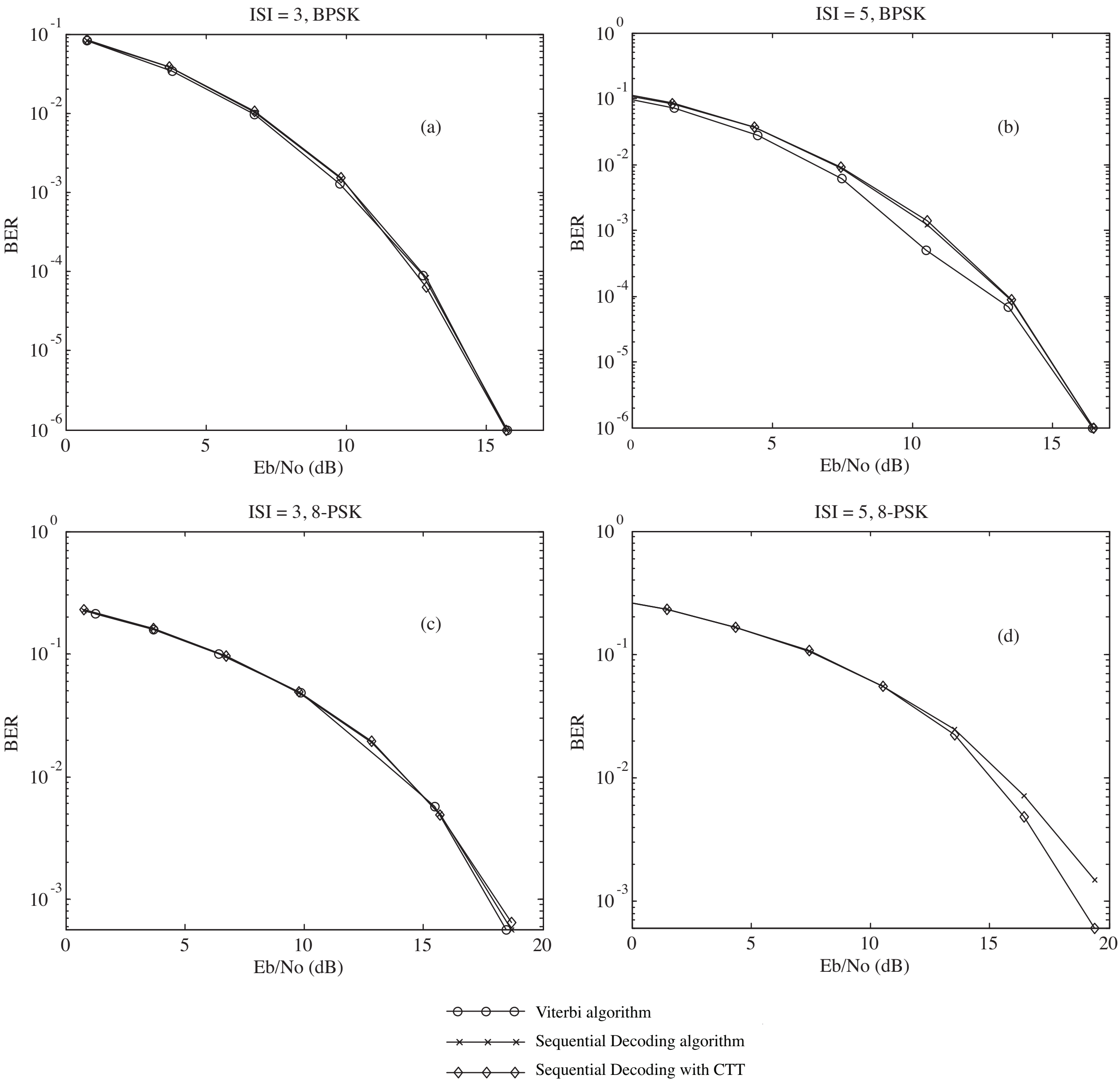

Figure 10. BER performances (exponentially decreasing good channel profile with Doppler effect): a) ISI = 3, BPSK, Viterbi (444 + 4800) computations; b) ISI = 3, 8-PSK, Viterbi (444 + 1,228,800) computations; c) ISI = 5, BPSK, Viterbi (740 + 19,456) computations; and d) ISI = 5, 8-PSK, Viterbi (740 + 79,691,776) computations.

4. Conclusion

The proposed channel transformation technique helps equalizers like sequential decoding to produce a BER performance close to that of the Viterbi algorithm, which is the optimum. The CTT allows the worst channel profiles to be transformed to good channels by boosting up the first tap of channel and suppressing the rest of the channel taps. The technique was evaluated by applying it to systems employing the sequential decoding equalizer. In good channels, all algorithms work properly and do not need the CTT, as seen from Figures 8-10.

However, in the worst channels, the CTT helps equalizers to improve the BER performance. Because most wireless channels have bad profiles, the CTT is a good candidate for use in high M-ary systems to improve BER performance.

References

[1] A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", IEEE Transactions on Information Theory, Vol. 13, pp. 260-269, 1967.

[2] J.G. Proakis, Digital Communications, 4th ed., Singapore, McGraw-Hill, 2000.

[3] G.D. Forney Jr, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference", IEEE Transactions on Information Theory, Vol. 18, pp. 363-378, 1972.

[4] Y. Baltaci, I. Kaya, A.R. Nix, "Implementation of a HIPERLAN/1 compatible CMF-DFE equalizer", 51st IEEE Vehicular Technology Conference, Vol. 3, pp. 1884-1888, 2000.

[5] R. Benjamin, I. Kaya, A.R. Nix, "Smart base stations for 'dumb' time-division duplex terminals", IEEE Communications Magazine, Vol. 37, pp. 124-131, 1999.

[6] I. Kaya, A.R. Nix, R. Benjamin, "Exploiting multipath activity using low complexity equalization techniques for high speed wireless LANs", 48th Vehicular Technology Conference Proceedings, Vol. 2, pp. 18-21, 1998.

[7] S.H. Qureshi, E.E. Newhall, "An adaptive receiver for data transmission over time-dispersive channels", IEEE Transactions on Information Theory, Vol. 19, pp. 448-457, 1973.

[8] I. Lee, J.M. Cioffi, "Design of equalized maximum-likelihood receiver", IEEE Communications Letters, Vol. 2, pp. 14-16, 1998.

[9] C.T. Beare, "The choice of the desired impulse response in combined linear-Viterbi algorithm equalizers", IEEE Transactions on Communications, Vol. 26, pp. 1301-1307, 1978.

[10] D.D. Falconer, J.F.R. Magee, "Adaptive channel memory truncation for maximum-likelihood sequence estimation", Bell Systems Technical Journal, Vol. 52, pp. 1541-1562, 1973.

[11] D.G. Messerschmitt, "Design of a finite impulse response for the Viterbi algorithm and decision feedback equalizer", IEEE International Conference on Communications, 1974.

[12] T. Miyajima, Z. Ding, "Second-order statistical approaches to channel shortening in multicarrier systems", IEEE Transactions on Signal Processing, Vol. 52, pp. 3253-3264, 2004.

[13] J. Balakrishnan, R.K. Martin, C.R. Johnson Jr, "Blind adaptive channel shortening by sum-squared auto-correlation minimization (SAM)", IEEE Transactions on Signal Processing, Vol. 51, pp. 3086-3093, 2003.

[14] R.K. Martin, J. Balakrishnan, W.A. Sethares, C.R. Johnson, Jr, "A blind adaptive TEQ for multicarrier systems", IEEE Signal Processing Letters, Vol. 9, pp. 341-343, 2002.

[15] R.K. Martin, "Fast-converging blind adaptive channel-shortening and frequency-domain equalization", IEEE Transactions on Signal Processing, Vol. 55, pp. 102-110, 2007.

[16] S.I. Husain, J. Yuan, J. Zhang, "Modified channel shortening receiver based on MSSNR algorithm for UWB channels", Electronics Letters, Vol. 43, pp. 535-537, 2007.

[17] S.I. Husain, J. Yuan, J. Zhang, R.K. Martin, “Time domain equalizer design using bit error rate minimization for UWB systems”, EURASIP Journal on Wireless Communications and Networking, Vol. 2009, doi: 10.1155/2009/786291, 2009.

[18] R.K. Martin, G. Ysebaert, K. Vanbleu, “Bit error rate minimizing channel shortening equalizers for cyclic prefixed systems”, IEEE Transactions on Signal Processing, Vol. 55, pp. 2605-2616, 2007.

[19] C. Fragouli, N. Al-Dhahir, N. Diggavi, W. Turin, “Prefiltered space-time M-BCJR equalizer for frequency-selective channels”, IEEE Transactions on Communications, Vol. 50, pp. 742-753, 2002.

[20] R. Samanta, R.W. Heath Jr, B.L. Evans, “Joint interference cancellation and channel shortening in multiuser-MIMO systems”, IEEE Transactions on Vehicular Technology, Vol. 56, pp. 652-660, 2007.

[21] C. Toker, G. Altın, “Blind, adaptive channel shortening equalizer algorithm which can provide shortened channel state information (BACS-SI)”, IEEE Transactions on Signal Processing, Vol. 57, pp. 1483-1493, 2009.

[22] N. Karaboğa, M.B. Çetinkaya, “A novel and efficient algorithm for adaptive filtering: artificial bee colony algorithm”, Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 19, pp. 175-190, 2011.

[23] T. Çavdar, A. Gangal, “A new sequential decoding algorithm based on branch metric”, Wireless Personal Communications, Vol. 43, pp. 1093-1100, 2007.

[24] T. Çavdar, A. Gangal, “Application of sequential estimation algorithm based on branch metric to frequency-selective channel equalization”, Wireless Personal Communications, Vol. 55, pp. 655-664, 2010.

[25] T. Çavdar, A New Sequential Block Data Estimation Algorithm and Its Performance Analysis over Frequency Selective Channels, PhD dissertation, Department of Electronics Engineering, Karadeniz Technical University, Turkey, 2003.

[26] C. Berrou, A. Glavieux, P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: turbo codes”, IEEE International Conference on Communications, Vol. 2, pp. 1064-1070, 1993.

Appendix

By dividing the first line of Eq. (7) by g_3, the second line by g_2, the third line by g_1, and the last line by g_0:

$$\begin{bmatrix} 1 & \frac{g_4}{g_3} & \frac{g_5}{g_3} & \frac{g_6}{g_3} & -\frac{g_7}{g_3} \\ 1 & \frac{g_3}{g_2} & \frac{g_4}{g_2} & \frac{g_5}{g_2} & -\frac{g_6}{g_2} \\ 1 & \frac{g_2}{g_1} & \frac{g_3}{g_1} & \frac{g_4}{g_1} & -\frac{g_5}{g_1} \\ 1 & \frac{g_1}{g_0} & \frac{g_2}{g_0} & \frac{g_3}{g_0} & -\frac{g_4}{g_0} \end{bmatrix}.$$

The first line is then subtracted from the second, the third, and the last line, and the results are placed into the same lines. We can represent the yielded matrix with the following A matrix for abbreviation.

$$\begin{bmatrix} 1 & \frac{g_4}{g_3} & \frac{g_5}{g_3} & \frac{g_6}{g_3} & -\frac{g_7}{g_3} \\ 0 & \frac{g_3}{g_2}-\frac{g_4}{g_3} & \frac{g_4}{g_2}-\frac{g_5}{g_3} & \frac{g_5}{g_2}-\frac{g_6}{g_3} & -\frac{g_6}{g_2}-\frac{g_7}{g_3} \\ 0 & \frac{g_2}{g_1}-\frac{g_4}{g_3} & \frac{g_3}{g_1}-\frac{g_5}{g_3} & \frac{g_4}{g_1}-\frac{g_6}{g_3} & -\frac{g_5}{g_1}-\frac{g_7}{g_3} \\ 0 & \frac{g_1}{g_0}-\frac{g_4}{g_3} & \frac{g_2}{g_0}-\frac{g_5}{g_3} & \frac{g_3}{g_0}-\frac{g_6}{g_3} & -\frac{g_4}{g_0}-\frac{g_7}{g_3} \end{bmatrix} = \begin{bmatrix} 1 & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22} & a_{23} & a_{24} & a_{25} \\ 0 & a_{32} & a_{33} & a_{34} & a_{35} \\ 0 & a_{42} & a_{43} & a_{44} & a_{45} \end{bmatrix} = A$$

If the previous elimination step is repeated for the A matrix, the following B matrix is yielded.

$$\begin{bmatrix} 1 & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & 1 & \dfrac{a_{23}}{a_{22}} & \dfrac{a_{24}}{a_{22}} & \dfrac{a_{25}}{a_{22}} \\ 0 & 0 & \dfrac{a_{33}}{a_{32}} - \dfrac{a_{23}}{a_{22}} & \dfrac{a_{34}}{a_{32}} - \dfrac{a_{24}}{a_{22}} & \dfrac{a_{35}}{a_{32}} - \dfrac{a_{25}}{a_{22}} \\ 0 & 0 & \dfrac{a_{43}}{a_{42}} - \dfrac{a_{23}}{a_{22}} & \dfrac{a_{44}}{a_{42}} - \dfrac{a_{24}}{a_{22}} & \dfrac{a_{45}}{a_{42}} - \dfrac{a_{25}}{a_{22}} \end{bmatrix} = \begin{bmatrix} 1 & b_{12} & b_{13} & b_{14} & b_{15} \\ 0 & 1 & b_{23} & b_{24} & b_{25} \\ 0 & 0 & b_{33} & b_{34} & b_{35} \\ 0 & 0 & b_{43} & b_{44} & b_{45} \end{bmatrix} = B$$

The second line of B is multiplied by b_{12} and then subtracted from the first line. The third line is then divided by b_{33}, the last line is divided by b_{43}, and the elimination step is applied again. This gives the following E matrix.

$$\begin{bmatrix} 1 & 0 & b_{13} - b_{12}b_{23} & b_{14} - b_{12}b_{24} & b_{15} - b_{12}b_{25} \\ 0 & 1 & 0 & b_{24} - b_{b23}\dfrac{b_{34}}{b_{33}} & b_{25} - b_{b23}\dfrac{b_{35}}{b_{33}} \\ 0 & 0 & 1 & \dfrac{b_{34}}{b_{33}} & \dfrac{b_{35}}{b_{33}} \\ 0 & 0 & 0 & \dfrac{b_{44}}{b_{43}} - \dfrac{b_{34}}{b_{33}} & \dfrac{b_{45}}{b_{43}} - \dfrac{b_{35}}{b_{33}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & e_{13} & e_{14} & e_{15} \\ 0 & 1 & 0 & e_{24} & e_{25} \\ 0 & 0 & 1 & e_{34} & e_{35} \\ 0 & 0 & 0 & e_{44} & e_{45} \end{bmatrix} = E$$

The third line of E is multiplied by e_{13} and subtracted from the first line. The last line of E is divided by e_{44}. This gives:

$$\begin{bmatrix} 1 & 0 & 0 & e_{14} - e_{13}e_{34} & e_{15} - e_{13}e_{35} \\ 0 & 1 & 0 & e_{24} & e_{25} \\ 0 & 0 & 1 & e_{34} & e_{35} \\ 0 & 0 & 0 & 1 & \dfrac{e_{45}}{e_{44}} \end{bmatrix}.$$

The last line is multiplied by $e_{14} - e_{13}e_{34}$ and subtracted from the first line. The last line is again multiplied by e_{24} and subtracted from the second line. Finally, it is multiplied by e_{34} and subtracted from the third line. We then get the unit matrix on the left side:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & e_{15} - e_{13}e_{35} - (e_{14} - e_{13}e_{34})\dfrac{e_{45}}{e_{44}} \\ 0 & 1 & 0 & 0 & e_{25} - e_{24}\dfrac{e_{45}}{e_{44}} \\ 0 & 0 & 1 & 0 & e_{35} - e_{34}\dfrac{e_{45}}{e_{44}} \\ 0 & 0 & 0 & 1 & \dfrac{e_{45}}{e_{44}} \end{bmatrix}.$$

The left 4 × 4 unit matrix is now the multiplicand of $[d_4 \;\; d_3d_2d_1]^T$ in Eq. (7), and the right column matrix is the multiplicand of d_0. Therefore, the following equation is yielded:

$$\begin{bmatrix} d_4 \\ d_3 \\ d_2 \\ d_1 \end{bmatrix} = \begin{bmatrix} e_{15} - e_{13}e_{35} - (e_{14} - e_{13}e_{34})\frac{e_{45}}{e_{44}} \\ e_{25} - e_{24}\frac{e_{45}}{e_{44}} \\ e_{35} - e_{34}\frac{e_{45}}{e_{44}} \\ \frac{e_{45}}{e_{44}} \end{bmatrix} d_0.$$