Journal of Southeast University (English Edition) June 2001 Vol.17 No.1 ISSN 1003—7985

A Reduced Search Soft-Output Detection Algorithm and Its Application to Turbo-Equalization*

Fan Xiangning** Dou Huaiyu Bi Guangguo

(Department of Radio Engineering, Southeast University, Nanjing 210096, China)

Abstract: To decrease the complexity of MAP algorithm, reduced-state or reduced-search techniques can be applied. In this paper we propose a reduced search soft-output detection algorithm fully based on the principle of M-algorithm for turbo-equalization, which is a suboptimum version of the Lee algorithm. This algorithm is called soft-output M-algorithm (denoted as SO-M-algorithm), which applies the M-strategy to both the forward recursion and the extended forward recursion of the Lee algorithm. Computer simulation results show that, by properly selecting and adjusting the breadth parameter and depth parameter during the iteration of turbo-equalization, this algorithm can obtain good performance and complexity trade-off.

Key words: MAP algorithm, Lee algorithm, soft-output M-algorithm, turbo-equalization

For digital communication over channels with intersymbol interference (ISI) and additive white Gaussian noise (AWGN), the maximum likelihood sequence estimation (MLSE) algorithm[1,2] is an optimum sequence detection algorithm providing hard-output, while the maximum *a posteriori* probability (MAP) algorithm[3,4] is an optimum symbol-by-symbol detection algorithm delivering soft-output. Although the complexity of the MAP algorithm is higher than that of the MLSE algorithm, the ability of delivering soft-output makes the MAP algorithm be able to perform turbo-equalization[5] and greatly improve the error rate performance.

Both the MLSE algorithm and the MAP algorithm view the channel filter as a discrete finite state machine (DFSM) and perform recursion through the trellis of the DFSM. When the length of the channel impulse response (CIR) and/or the size of the channel symbol set are large, the number of the states of the channel DFSM trellis will become very large and those algorithms will become too complex to be realized. Because S, the number of states of channel DFSM trellis, increases exponentially with L, the length of channel impulse response, we have $S = M^{L-1}$, where M is the size of channel symbol set.

To decrease the complexity of MAP algorithm, there are two kinds of sequence estimation algorithms, i.e. the reduced-state sequence estimation (RSSE) algorithm[6] and the M-algorithm[7]. The principles of reducing complexity in these sequence estimation algorithms can also be used to reduce the complexity of symbol-by-symbol soft-output detection algorithm. Ref. [8] compared some reduced-state soft-output detection algorithms(denoted as SO-RS-algorithm) based on the principle of RSSE for concatenated detecting and decoding. Ref. [9] proposed a reduced-search soft-output decoding algorithm partially based on the principle of M-algorithm for turbo decoding. In this paper we discuss a reduced complexity soft-output detection algorithm, the soft-output M-algorithm, fully based on the principle of M-algorithm for turbo-equalization.

1 Review of the MAP Algorithm

In general, the MAP algorithm (or *a posteriori* probability (APP) algorithm) is a solution to the problem of calculating the *a posteriori* probability of the input symbols (or the output symbols, the states) of a DFSM on the condition that the output sequence of the DFSM, observed through a discrete memoryless channel (DMC), is given.

The underlying model of DFSM + DMC is depicted in Fig. 1, where s_n is the state sequence of the DFSM

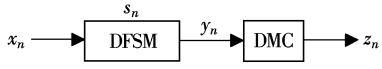

Fig.1 The model of DFSM + DMC

taking values from a finite set S, x_n and y_n are the input and output sequences of the DFSM taking values from finite sets X and Y respectively, and z_n is the

Received 2000 - 08 - 24.

* The project supported by the National Natural Science Foundation of China (69882004).

** Born in 1964, male, doctor, associate professor.

output sequence of the DFSM observed through the DMC. Suppose we have got a segment of observed sequence $\{z_n\}_1^N$, our purpose is to calculate the APP of a particular input symbol $\hat{x}_k$, i.e., $P(\hat{x}_k \mid \{z_n\}_1^N)$.

There are two types of MAP algorithm: the BCJR algorithm[3] and the Lee algorithm[4]. Both algorithms have their equivalent Log version and simplified Max-Log version. They are summarized in Tab.1, where $s_{l-1}\ \dfrac{x_l}{y_l} s_l$ denotes a trellis branch with starting state s_{l-1}, ending state s_l, input symbol x_l and output symbol y_l. $\delta\left(s_{l-1}\ \dfrac{x_l}{y_l} s_l\right) = 1$ if $s_{l-1}\ \dfrac{x_l}{y_l} s_l$ is a valid branch, $\delta\left(s_{l-1}\ \dfrac{x_l}{y_l} s_l\right) = 0$ if $s_{l-1}\ \dfrac{x_l}{y_l} s_l$ is an invalid branch. When N is a constant, the algorithms operate at block mode. When $N = k + D$, and D is a constant, the algorithms operate at continuous mode.

Tab.1 Two types of MAP algorithm

BCJR algorithm	Branch values	$\gamma_l(s_{l-1}, x_l, y_l) = P(x_l \mid s_{l-1})P(z_l \mid y_l), l = 1,2,\cdots,N$
	State values of forward recursion	$\alpha_l(s_l) = P(\{z_n\}_1^l, s_l), l = 1,2,\cdots,k-1$
	Initialization of forward recursion	$\alpha_0(s_0) = P(s_0)$
	Forward recursion	$\alpha_l(s_l) = \sum_{s_{l-1}\ \frac{x_l}{y_l} s_l} \alpha_{l-1}(s_{l-1})\gamma_l(s_{l-1}, x_l, y_l)$
	State values of backward recursion	$\beta_l(s_l) = P(s_l \mid \{z_n\}_{l+1}^N), l = N-1, N-2,\cdots,k$
	Initialization of backward recursion	$\beta_L(S_L) = 1$
	Backward recursion	$\beta_l(s_l) = \sum_{s_1\ \frac{x_{l+1}}{y_{l+1}} s_{l+1}} \gamma_l(s_l, x_{l+1}, y_{l+1})\beta_{l+1}(s_{l+1})$
	Calculation of the APP	$P(\hat{x}_k \mid \{z_n\}_1^N) = \dfrac{\sum_{s_{k-1}\ \frac{\hat{x}_k}{y_k} s_k} \alpha_{k-1}(s_{k-1})\gamma_k(s_{k-1}, \hat{x}_k, y_k)\beta_k(s_k)}{\sum_{s_{k-1}\ \frac{x_k}{y_k} s_k} \alpha_{k-1}(s_{k-1})\gamma_k(s_{k-1}, x_k, y_k)\beta_k(s_k)}$
The Lee algorithm	Branch values	$\gamma_l(s_{l-1}, x_l, y_l) = P(x_l \mid s_{l-1})P(z_l \mid y_l), l = 1,2,\cdots,N$
	State values of forward recursion	$\alpha_l(s_l) = P(\{z_n\}_1^l, s_l), l = 1,2,\cdots,k$
	Initialization of forward recursion	$\alpha_0(s_0) = P(s_0)$
	Forward recursion	$\alpha_l(s_l) = \sum_{s_{l-1}\ \frac{x_l}{y_l} s_l} \alpha_{l-1}(s_{l-1})\gamma_l(s_{l-1}, x_l, y_l)$
	State values of extended forward recursion	$\alpha_l(\hat{x}_k, s_l) = P(\hat{x}_k, \{z_n\}_1^l, s_l), l = k+1,\cdots,N$
	Initialization of extended forward recursion	$\alpha_k(\hat{x}_k, s_k) = \alpha_k(s_k)\delta\left(\cdot\ \dfrac{\hat{x}_k}{\cdot} s_k\right)$
	Extended forward recursion	$\alpha_l(\hat{x}_k, s_l) = \sum_{s_{l-1}\ \frac{x_l}{y_l} s_l} \alpha_{l-1}(\hat{x}_k, s_{l-1})\gamma_l(s_{l-1}, x_l, y_l)$
	Calculation of the APP	$P(\hat{x}_k \mid \{z_n\}_1^N) = \dfrac{\sum_{s_N} \alpha_N(\hat{x}_k, s_N)}{\sum_{x_k}\sum_{s_N} \alpha_N(x_k, s_N)}$

The BCJR algorithm includes two recursions: the forward recursion and the backward recursion. The Lee algorithm also includes two recursions: the forward recursion and the extended forward recursion. A recursion is called a public recursion if its initial state values are independent of the particular input symbol $\hat{x}_k$, and a private recursion if its initial state values depend on the particular input symbol $\hat{x}_k$. The forward recursions in both algorithms and the backward recursion in the BCJR algorithm are public recursions, while the extended forward recursion in the Lee algorithm is a private recursion. Because the Lee algorithm includes a private recursion, the computational complexity of the Lee algorithm is much higher than that of the BCJR algorithm.

2 The Soft-Output M-Algorithm

To reduce the complexity of the MAP algorithm, a natural strategy is that at each step of recursion, only M of the largest state values are retained while the

others are omitted and set to zeros. Next step of recursion will be based on the previously retained state values so that the complexity of the MAP algorithm will be reduced. This M-strategy is similar to that of the M-algorithm reducing complexity for MLSE, and the reduced complexity MAP algorithm based on M-strategy is called soft-output M-algorithm.

When the SO-M-algorithm is applied to soft-output detection, the breadth parameter M is sensitive to the phase response of channel filter. If the channel filter is a minimum phase filter, M may be quite small for forward and extended forward recursion because the significant state values tend to concentrate at a few states, but may be very large for backward recursion because the significant state values tend to spread over a lot of states. Thus the M-strategy seems to be suitable for the Lee algorithm rather than the BCJR algorithm.

Nevertheless, the BCJR algorithm is still attractive due to its lower computational complexity. Ref. [9] proposed a scheme, called M-BCJR algorithm, which applies the M-strategy only to the forward recursion while restricts the backward recursion on the states retained during the forward recursion. In this paper, we discuss the SO-M-algorithm fully using the M-strategy, specifically, an M-Lee algorithm which applies the M-strategy to both the forward recursion and the extended forward recursion and operating at continuous mode. Assume that the breadth parameter of the forward recursion is M_0, the breadth parameter for each step of the extended forward recursion is M_1, $\cdots$, M_D, respectively, where D is the depth parameter of the extended forward recursion, then the computational complexity of the M-Lee algorithm is proportional to $C = M_0 + (M_1 + \cdots + M_D) \cdot |X|$.

Suppose the computational complexity is given, i.e. C is a constant. We should select a proper composition of the parameters M_0, M_1, $\cdots$, M_D to obtain good performance/complexity ratio.

For the M-Lee algorithm under the constraint of complexity, error is introduced and accumulated along the recursion because some state values are omitted during the recursion. To reduce the accumulation of error, the breadth parameter of previous step of recursion should be no smaller than that of next step of recursion, i.e. $M_0 \geqslant M_1 \geqslant, \cdots, \geqslant M_D$.

Because the M-Lee algorithm is operating at continuous mode, more information can be obtained for calculating the APP when the depth parameter D is larger, which will improve the quality of soft-output. However, when the depth parameter is larger, the breadth parameter will be smaller and more error will be introduced at each step of recursion, which will degenerate the quality of soft-output. Thus for the M-Lee algorithm, D should be chosen to keep the balance between the information and error.

The error introduced at each step of recursion also depends on the signal-to-noise ratio (SNR). The lower the SNR is, the larger the error is introduced. Thus, generally speaking, larger breadth and smaller depth should be chosen as SNR is lower, while smaller breadth and larger depth as SNR is higher. Because the equivalent SNR is increased during the iteration of turbo-equalization, better performance/complexity ratio is expected if the depth parameter and breadth parameter could be adjusted for different iterations.

3 Simulation

In this section, we will give an example to show the effect of selecting proper parameters for the M-Lee algorithm and compare the performance of the proposed SO-M-algorithm to that of another kind reduced complexity algorithm, i.e., the reduced-state soft-output detection algorithm (SO-RS-algorithm)[8].

The model of the transmission system is shown in Fig.2. The transmitter consists of a channel encoder, coder, and an interleaver, π. The channel is represented by an equivalent discrete-time white noise filter model and the channel filter is a minimum phase filter[1]. If we include the channel filter h_l into the transmitter, the transmitter is nothing but a generalized serially concatenated turbo encoder with its outer encoder being the channel encoder and inner encoder being the channel filter (it can be viewed as a special encoder with code rate of 1).

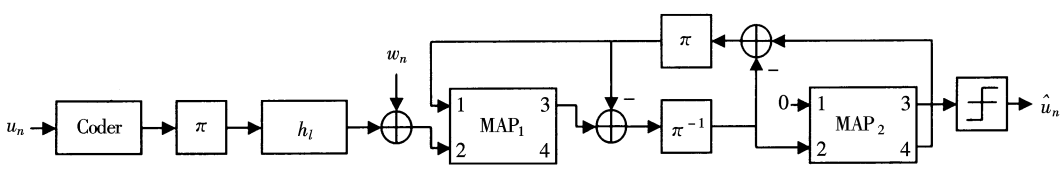

Fig.2 The model of the transmission system

The receiver is a generalized serially concatenated turbo decoder consists of two MAP modules, MAP_1 and MAP_2, and a pair of interleaver/deinterleaver, π/π^{-1}. MAP_1 serves as channel symbol probabilistic detector, while MAP_2 as channel symbol probabilistic filter and decoder. During the iteration of turbo-equalization, only the extrinsic information, permuted by π^{-1} or π, is passed between the two MAP modules.

In this paper, coder is Ungerboeck's 8PSK TCM with 8 states[10], π is a random interleaver with length of 1 024, and h_l is the same as that of the ISI channel given in Ref. [5], i.e. $\{\sqrt{0.45}, \sqrt{0.25}, \sqrt{0.15}, \sqrt{0.10}, \sqrt{0.05}\}$. MAP_1 uses the Max-Log version of SO-M-algorithm while MAP_2 uses the Max-Log version of MAP algorithm. The complexity of the SO-M-algorithm is constrained by $C = 32$.

Under the constraint of $C = 32$, we examine three groups of parameter $\{M_0, M_1, \cdots, M_D\}$: {8,3}, {8,2,1} and {8,1,1,1}. Fig.3 gives the bit error rate performance vs. SNR with these groups of parameter. It can be seen that the performance at lower SNR with smaller depth and larger breadth is better than that with larger depth and smaller breadth, while at higher SNR the case is inverted. If we adjust the parameters at different SNR, the performance curve will reach the lower envelope of Fig.3. However, the equivalent SNR is increased during the iteration of turbo-equalization, so even better performance can be obtained by adjusting the parameters for different iterations. The principle for adjusting the parameters is based on the following fact. The convergence process of turbo-equalization can be represented by curve $\bar{P}_2(i)$, the geometric mean of the second largest symbol probability vs. iteration number, i.e.

$$\bar{P}_2(i) = \left(\prod_{k=1}^{K} P_2^i(x_k)\right)^{\frac{1}{K}}$$

where $P_2^i(x_k)$ is the second largest value of set $\{P(x_k) \mid x_k \in X\}$ at the i-th iteration, K is the number of symbols. If turbo-equalization is convergent, $\bar{P}_2(i)$ will decrease to a lower level as i increases. Otherwise, $\bar{P}_2(i)$ will oscillate and keep at a higher level. In this example, for total 10 iterations, at SNR = 10 dB, six iterations of turbo-equalization are performed with parameters {8,3}. Then we adjust the parameters to {8,2,1} and 4 iterations of the turbo-equalization are performed. At SNR = 10.5 dB, two iterations with {8,3}, one iteration with {8,2,1} and 7 iterations with {8,1,1,1} are performed sequentially, and at SNR = 11 dB, one iteration with {8,3}, one iteration with {8,2,1} followed by 8 iterations with {8,1,1,1} are performed. The bit error rate at 10 dB, 10.5 dB and 11 dB by the 10th iteration (See Fig.4) are lower than those with fixed parameters (See Fig.3). A scheme of adaptively adjusting the parameters is developed and will be reported in another paper. For comparison, the performance of reduced-state soft-output detection algorithm is also shown in Fig. 4. The SO-RS-algorithm totally retains 32 state values (16 for forward recursion and 16 for backward recursion), so its complexity is about the same as that of the SO-M-algorithm. It can be seen that the performance of the SO-M-algorithm is better than that of the SO-RS-algorithm.

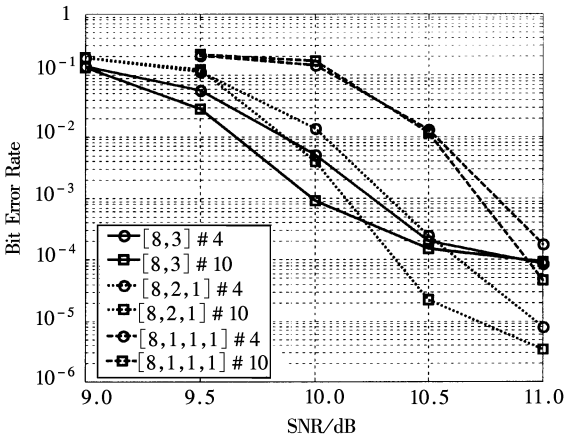

Fig.3 The performance of the SO-M-algorithm with different parameters (4th and 10th iterations)

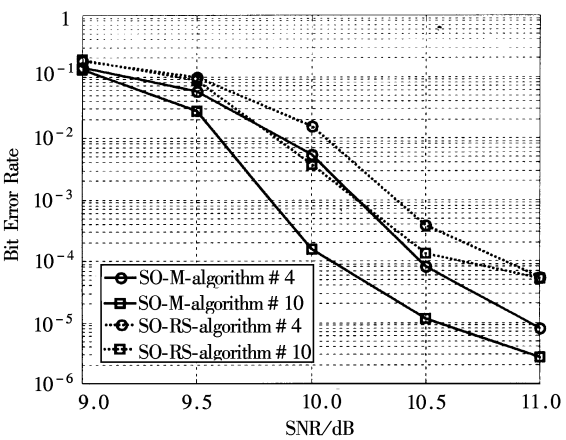

Fig.4 The performance of the SO-M-algorithm and SO-RS-algorithm (4th and 10th iterations)

4 Conclusion

The SO-M-algorithm is a reduced search soft-output detection algorithm which applies the M-strategy to both the forward and the extended forward recursions of the Lee algorithm. Applying the SO-M-algorithm to turbo-equalization and under the constraint of complexity, good performance/complexity ratio can be

obtained by properly selecting and adjusting the breadth parameter and depth parameter during the iteration of turbo-equalization.

References

1 Jr. G. D. Forney, Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference, *IEEE Transactions on Information Theory*, vol. IT-18, no.3, pp.363 – 378, 1972

2 Jr. G. D. Forner, The Viterbi algorithm, In: *Proceeding of the IEEE*, vol.61, no.3, pp.268 – 278, 1973

3 L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, Optimal decoding of linear codes for minimizing symbol error rate, *IEEE Transactions on Information Theory*, vol. IT-20, no.2, pp.284 – 287, 1974

4 L. Lee, Real-time minimal-bit-error probability decoding of convolutional code, *IEEE Transactions on Communications*, vol. COM-22, no.2, pp. 146 – 151, 1974

5 C. Douillard, M. Jézéquel, and C. Berrou, Iterative correction of intersymbol interference: Turbo-equalization, *European Transactions on Telecommunications*, vol.6, no.5, pp.507 – 511, 1995

6 M. V. Eyuboglu, and S. U. H. Qureshi, Reduced-state sequence estimation with set partitioning and decision feedback, *IEEE Transactions on Communications*, vol. COM-36, no.1, pp.13 – 20, 1988

7 J.B. Anderson, Sequential coding algorithms: a survey and cost analysis, *IEEE Transactions on Communications*, vol. COM-32, no.2, pp.169 – 176, 1984

8 P. Hoeher, TCM on frequency-selected fading channels: a comparison of soft-output probabilistic equalizers, In: *Proceeding of the IEEE GLOBLECOM'90*, San Diego, pp. 401.4.1 – 401.4.6, 1990

9 V. Franz, and J. B. Anderson, Concatenated decoding with a reduced-search BCJR algorithm, *IEEE Journal of Selected Areas on Communications*, vol.16, no.2, pp.186 – 195, 1998

10 G. Ungerboeck, Channel coding with multilevel/phase signals, *IEEE Transactions on Information Theory*, vol. IT-28, no.1, pp.55 – 67, 1982

一种减少搜索的软输出检测算法及其在 Turbo 均衡中的应用

樊祥宁　窦怀宇　毕光国

(东南大学无线电工程系，南京 210096)

摘　要　为了减少 MAP 算法的复杂度，可以采用减状态或减搜索技术.本文提出了一种完全基于 M 算法原理、应用于 Turbo 均衡的减少搜索的软输出检测算法，它是一种次最佳的 Lee 算法.该算法称为软输出 M 算法(SO-M-算法)，它同时在 Lee 算法的前向迭代及扩展前向迭代中采用了 M 策略.计算机仿真结果表明，通过适当选择和调整 Turbo 均衡迭代过程中算法的广度参数和深度参数，该算法可获得较好的性能与复杂度的折衷.

关键词　MAP 算法，Lee 算法，软输出 M 算法，Turbo 均衡

中图分类号　TN911.5