On the SOVA for Extremely High Code Rates over Partial Response Channels

Ali Ghrayeb

Abstract: **In this paper, we extend the derivation of the iterative soft-output Viterbi algorithm (SOVA) for partial response (PR) channels, and modify its decoding process such that it works consistently for arbitrary high code rates, e.g., rate 64/65. We show that the modified SOVA always outperforms the conventional SOVA that appears in the literature with a significant difference for high code rates. It also offers a significant cut down in the trace-back computations. We further examine its performance for parallel and serial concatenated codes on a precoded Class IV partial response (PR4) channel. Code rates of the form $\frac{k_0}{k_0+1}$ ($k_0 = 4, 8,$ and 64) are considered. Our simulations indicate that the loss suffered by the modified SOVA, relative to the APP algorithm, is consistent for all code rates and is at most 1.2 dB for parallel concatenations and at most 1.6 dB for serial concatenations at $P_b = 10^{-5}$.**

Index Terms: **SOVA, partial response signaling, iterative decoding, parallel and serial concatenation, decoding depth.**

I. INTRODUCTION

Considerable research has been performed recently [1]–[6] on the application of turbo coding to partial response channels. In [1] and [2], a parallel concatenated ("turbo") coded PR4 system is considered where the precoded PR4 channel is treated as an inner code. The PR4 channel is detected using an *a posteriori* probability (APP) detector matched to the precoded PR4 trellis, and the turbo code is decoded using two APP decoders matched to the two recursive systematic convolutional (RSC) encoders. The PR4 detector may or may not share soft information with the turbo decoding, depending on performance/complexity requirements [1], [2]. The turbo code may be replaced by a single convolutional encoder as in [3]–[6], effecting a serial concatenated code system with the precoded PR4 channel acting as the inner code. For simplicity, we refer to the former system as the parallel concatenated code (PCC) system and the latter one as the serial concatenated code (SCC) system. Obviously, three APP decoders are required in the PCC system whereas two APP decoders are required in the SCC system (see Fig. 1). In both cases, simulations revealed that a substantial performance improvement is achieved over the uncoded system [1]–[6].

A study by Robertson *et al.* [7] has shown that little degradation is suffered on the binary antipodal AWGN channel when the APP algorithm is replaced with approximate versions of it. One of those suboptimal algorithms is the *soft-output Viterbi algorithm* (SOVA), which offers an attractive complexity advantage:

Manuscript received September 18, 2001; approved for publication by Joseph Boutros, Division I Editor, November 1, 2002.

A. Ghrayeb is with the Department of Electrical and Computer Engineering, Concordia University, Montreal, Quebec, H3G 1M8, CANADA, e-mail: aghrayeb@ece.concordia.ca.

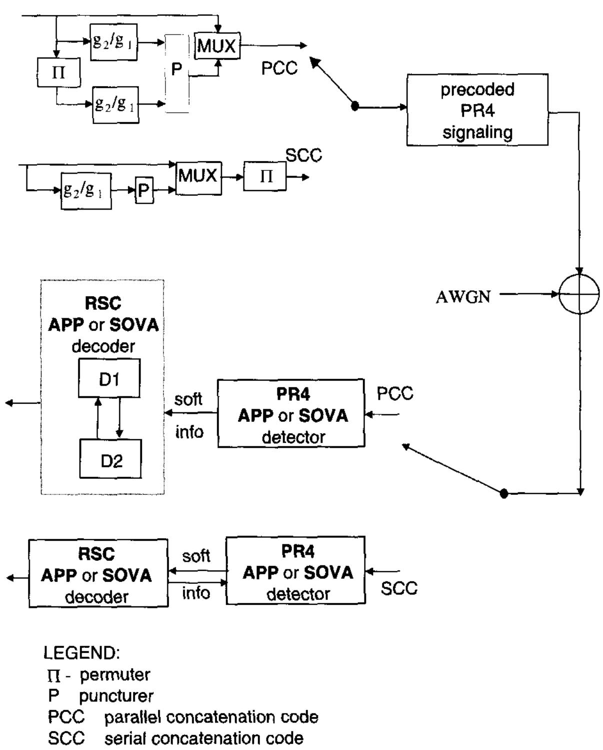

Fig. 1. Model for PCC and SCC encoding/decoding on the precoded PR4 channel.

a factor of less than one half relative to the APP decoder for 16 states, and slightly more than a half for 4 states [7]. In addition to reducing complexity, the SOVA decoder, unlike the APP decoder, does not require knowledge of the channel noise variance [8]. The SOVA algorithm may be implemented via the trace-back algorithm [9] or the register exchange algorithm [10]. We shall focus on the former algorithm.

In this paper, we extend the derivation of the SOVA algorithm for partial response (PR) channels and introduce a slight modification in its decoding process such that it works consistently for extremely high code rates, e.g., rate 64/65. This modification in the decoding process becomes very crucial when considering high code rates. (We mention that in all cases in which the performance of SOVA was examined, only code rates up to $\frac{8}{9}$ were considered.) This stems from the fact that, to achieve reliable estimates of the transmitted sequence at the output of the SOVA algorithm, a relatively large decoding depth should be used. Consequently, according to the nature of the SOVA, unlike the standard Viterbi algorithm, it requires tracing back the nonsurviving paths within a time window of a length equal to the decoding depth *every time* an estimate on a bit is made. This introduces a considerable delay in the system, in addition to the already existing delay, assuming an iterative decoding type of environment. The modified SOVA provides absolutely the most reliable bit estimates with a drastic cut down in the trace-back computations. A comparison in performance between the modified SOVA and the conventional SOVA [7]–[11] is presented.

We then examine the performance of the modified SOVA for parallel and serial concatenated codes of the form $\frac{k_0}{k_0+1}$ for $k_0 = 4, 8$, and 64, with the following decoder configurations. We first replace the APP decoder(s) that correspond to the RSC codes with matching iterative SOVA decoder(s), and leave the APP detector that corresponds to the PR4 channel unchanged (in both the PCC and the SCC systems, see Fig. 1). This is a reasonable thing to do because we would like the inner detector to provide as much reliable information as possible to the outer decoder. In addition, the inner "code" has only 4 states and thus complexity is not big an issue here. We then replace every APP detector in both systems with a matching SOVA detector (i.e., three SOVA's in the PCC system, and two SOVA's in the SCC system). According to the setup shown in Fig. 1, in the PCC system, the outer decoders and inner detector do not exchange extrinsic information back and forth in an iterative manner, as is the case in the SCC system. This reduction in complexity comes at the expense of a degradation in performance by about 0.5 dB [2].

The rest of the paper is organized as follows. In Section II, we derive the SOVA algorithm for PR channels, and outline the modification in its decoding process as compared to the version that exists in the literature. Section III presents the simulation results for all code rates. Further, a comparison in performance between the modified SOVA and the SOVA that appears in the literature is presented. Finally, Section IV concludes the paper.

II. THE MODIFIED SOVA ALGORITHM

We discuss in this section the SOVA algorithm as an approximation to the BCJR-APP algorithm for the SCC system; SOVA decoding for the PCC system is similar. As mentioned earlier, the SCC system requires two SOVA's. One SOVA decoder is matched to the RSC encoder, and the second SOVA is a detector matched to the precoded PR4 trellis. Hagenauer in [9] and [11] derived the SOVA algorithm for binary trellises. In this section, we modify the SOVA algorithm for PR trellises. We will also outline the modification that we introduce in the decoding process of the SOVA, which is very important when considering relatively high code rates.

A. The SOVA for PR Trellises

The branches of the trellis of a PR channel are labeled with the information bit u_k and its corresponding channel symbol c_k. Let $c_1^N = [c_1\ c_2\ \cdots\ c_N]$ be the PR channel output sequence that corresponds to a block of information bits $u_1^N = [u_1\ u_2\ \cdots\ u_N]$, and $y_1^N = [y_1\ y_2\ \cdots\ y_N]$ be the corresponding noisy received sequence. Additive white Gaussian noise (AWGN) of variance $N_0/2$ is assumed.

At any given time k, a block-wise APP detector searches for the state sequence $s_{1,m}^k = [s_{1,m}\ s_{2,m}\ \cdots\ s_{k,m}]$ that corresponds to the information sequence u_1^k by maximizing the *a posteriori* probability

$$M_{k,m} = P\left(s_{1,m}^k \mid y_1^k\right), \tag{1}$$

where m is an index that corresponds to the trellis path corresponding to the state sequence $s_{1,m}^k$. Since y_1^k does not depend on m then maximizing (1) is equivalent to maximizing

$$P\left(y_1^k \mid s_{1,m}^k\right) \cdot P\left(s_{1,m}^k\right). \tag{2}$$

Now, the term $P\left(s_{1,m}^k\right)$can be expressed as

$$\begin{aligned} P\left(s_{1,m}^k\right) &= P\left(s_{1,m}^{k-1}\right) \cdot P\left(s_{1,m}\right) \\ &= P\left(s_{1,m}^{k-1}\right) \cdot P\left(u_{k,m}\right), \end{aligned} \tag{3}$$

where $u_{k,m}$ is the information bit that corresponds to the state transition $s_{k-1,m} \rightarrow s_{k,m}$. We thus have from (1)–(3),

$$\max_m \{M_{k,m}\} = \max_m \begin{Bmatrix} P\left(s_{1,m}^{k-1}\right) \prod_{i=1}^{k-1} p\left(y_i \mid s_{i-1,m}, s_{i,m}\right) \\ \cdot P\left(u_{k,m}\right) p\left(y_k \mid s_{k-1,m}, s_{k,m}\right) \end{Bmatrix}, \tag{4}$$

where $p\left(y_k \mid s_{k-1}, s_{k,m}\right) = p\left(y_k \mid c_{k,m}\right)$. In maximizing (4), it is equivalent to taking its logarithm, and adding two constants that are independent of m. Let the two constants be $K_1 = -\frac{1}{2}\log\left[P\left(u_k = +1\right) \cdot P\left(u_k = -1\right)\right]$ and $K_2 = \log\sqrt{\pi N_0}$. Also, denote the logarithm of the first line in (4) by $M_{k-1,m}/2$ which corresponds to the cumulative metric at time $k-1$ along path m. With this, (4) becomes

$$\begin{aligned} \max_m \{M_{k,m}\} = \max_m \{ & M_{k-1,m} + \left[\log P\left(u_{k,m}\right) + K_1\right] \\ & + \left[\log p\left(y_k \mid s_{k-1,m}, s_{k,m}\right) + K_2\right]\}, \end{aligned} \tag{5}$$

which simplifies to

$$\max_m \{M_{k,m}\} = \max_m \begin{Bmatrix} M_{k-1,m} - \frac{1}{N_0}\left(y_k - c_{k,m}\right)^2 \\ +\frac{1}{2} u_{k,m} L^a\left(u_{k,m}\right) \end{Bmatrix}, \tag{6}$$

where $L^a(u_{k,m}) = \log\frac{P(u_{k,m}=+1)}{P(u_{k,m}=-1)}$ is the *a priori* (extrinsic) information on bit $u_{k,m}$ that is usually obtained from another decoder assuming iterative decoding.

It follows from (6), after multiplying both sides by -1, that the cumulative metric $M_k(s)$ for state s at time k along some arbitrary path m is updated according to

$$M_k(s) = \min\left\{\lambda(s', s) + M_{k-1}(s'), \lambda(s'', s) + M_{k-1}(s'')\right\}, \tag{7}$$

where $\lambda(s', s'')$ is the branch metric for the transition from state s' to state s at time k which is defined as

$$\lambda(s', s) = \frac{1}{N_0}(y_k - c_k)^2 - \frac{1}{2} u_k L^a(u_k), \tag{8}$$

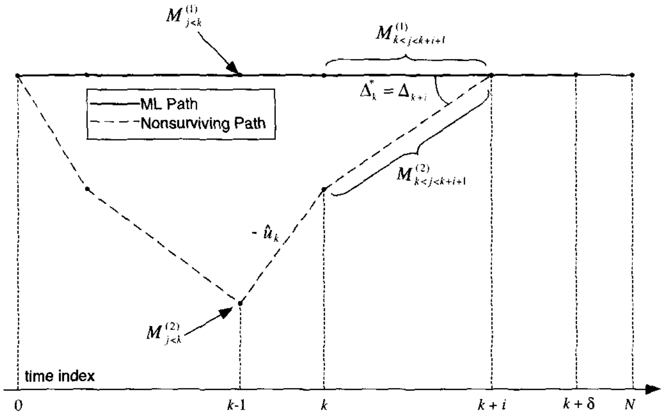

Fig. 2. The VA soft-output for bit $\hat{u}_k$.

and c_k is the PR channel output at time k corresponding to the transition $s' \to s$. The branch metric $\lambda(s'', s)$ is similarly defined. Now define the difference metric for state s at time k as [9]

$$\Delta_k = \left|(M_{k-1}(s') + \lambda(s', s)) - (M_{k-1}(s'') + \lambda(s'', s))\right|. \tag{9}$$

In [9], Δ_k was also shown to be approximated as

$$\Delta_k \approx \log \frac{P(\text{correct})}{1 - P(\text{correct})}, \tag{10}$$

where P(correct) is the probability that the path decision of the survivor at time k was correct. Therefore, Δ_k represents the reliability that the path ending at state s at time k was correct.

In the conventional SOVA, to obtain the soft output for bit u_k, we first obtain the hard decision $\hat{u}_k$ after a delay δ (i.e., at time $k + \delta$) , where δ is the *decoding depth*. At time $k + \delta$, we select the surviving path that ends at the state that has the lowest metric and the selected path is considered to be the *maximum-likelihood* (ML) path. We trace back the ML path to obtain the hard decision $\hat{u}_k$. Along the ML path, there are $\delta + 1$ nonsurviving paths that have been discarded, and each nonsurviving path has a certain difference metric Δ_j where $k \leq j \leq k + \delta$. (Clearly, this is because along the ML path there are $\delta + 1$ states, and each state has a difference metric that was calculated using (9).)

Now define

$$\Delta^*_k = \min\left\{\Delta_k, \Delta_{k+1}, \cdots, \Delta_{k+\delta}\right\}, \tag{11}$$

where the minimum is taken *only* over the nonsurviving paths within the time window $[k\,,\ k+\delta]$ that would have led to a different decision $\hat{u}_k$. It was shown in [9] that Δ^*_k represents the reliability of the hard decision $\hat{u}_k$, and the reliability increases with increasing δ. Given $\hat{u}_k$ and Δ^*_k, the soft output of the Viterbi algorithm for bit u_k is approximated by [9]

$$L_{sova}(u_k) \approx \hat{u}_k \cdot \Delta^*_k. \tag{12}$$

From Fig. 2, and (8) and (9) it can be shown [9] that Δ^*_k has the following structure

$$\begin{aligned}\Delta^*_k \;=\; & \left(M^{(2)}_{j<k} - M^{(1)}_{j<l}\right) + \left(M^{(2)}_{k<j<k+i} - M^{(1)}_{k<j<k+1}\right) \\ & + \frac{1}{N_0}\left((y_k - c_{k2})^2 - (y_k - c_{k1})^2\right). \\ & + \hat{u}_k L^a(u_k),\end{aligned} \tag{13}$$

where c_{k1}, and c_{k2} are the nominal channel outputs to state transitions at time k along paths 1 and 2, respectively, i is the time index at which $\Delta^*_k = \Delta_{k+i}$, and $M^{(1)}_{j<k}$ is the cumulative metric at time $k-1$ along path 1. The other terms are obviously defined (see Fig. 2). Substituting (13) into (12) yields

$$\begin{aligned}L_{sova}(u_k) \;\approx\; & \hat{u}_k\left(M^{(2)}_{j<k} - M^{(1)}_{j<k}\right) \\ & + \hat{u}_k\left(M^{(2)}_{k<j<k+i} - M^{(1)}_{k<j<k+i}\right) \\ & + \frac{1}{N_0}\hat{u}_k\left((y_k - c_{k2})^2 - (y_k - c_{k1})^2\right) \\ & + L^a(u_k).\end{aligned} \tag{14}$$

where the first three terms represent the extrinsic information, which we denote by $L^e(u_k)$, i.e., $L^e(u_k) = L_{sova}(u_k) - L^a(u_k)$. $L^e(u_k)$ is then passed to the subsequent decoder to be used as an *a priori* information in the next iteration. Note that results obtained in [9] cannot be applied directly to PR channels since the channel symbols in this case have unequal energy.

B. Modification in the SOVA Decoding Process

In the conventional SOVA, as mentioned above, to obtain the soft output for bit u_k, we first obtain the hard decision $\hat{u}_k$ after a delay δ (i.e., at time $k + \delta$) , where δ is the *decoding depth.* At time $k + \delta$, we select the ML path to be the surviving one that ends at the state that has the lowest metric. We trace back the ML path to obtain the hard decision $\hat{u}_k$. Along the ML path, there are $k + \delta$ nonsurviving paths that have been discarded, and each nonsurviving path has a certain difference metric Δ_j where $k \leq j \leq k + \delta$. We then obtain the reliability value Δ^*_k with respect to δ. Δ^*_k is obtained by taking the minimum over the nonsurviving paths within the time window $[k\,,\ k+\delta]$, *i.e.*, $\Delta^*_k = \min\left\{\Delta_k, \Delta_{k+1}, \cdots, \Delta_{k+\delta}\right\}$ such that the path corresponding to every element in this set would have led to a different decision $\hat{u}_k$. Note that every time a hard decision $\hat{u}_k$ is made, a new set of $\delta + 1$ nonsurviving paths will have to be traced back. This introduces a considerable amount of delay in the system, especially when δ is large, which is the case when the code rate is high.

In the modified SOVA, first we assume that all encoders start at the zero state, and at least one encoder is terminated at the zero state (assuming SCC and iterative decoding.) The decoding process proceeds the usual way, *i.e.*, choosing a survivor and calculating the metric as well as difference metric for each state, using (7) and (9), respectively, for all $\{k \in 1, 2, \cdots, N\}$. In other words, no hard decisions are made until $k = N$. When $k = N$ (*i.e.*, at the end of the data block), we select the surviving path that ends at *state zero* as the ML path. For decoders whose corresponding encoders are not terminated at state zero, we select the surviving path that ends at the state that has the lowest metric. We then trace back the ML path to obtain the hard decisions $\hat{u}_k$, and the corresponding Δ_k, for $k = 1, ..., N$. To obtain the soft output for bit u_k, we define a parameter, δ', called the *reliability depth*, and find the reliability value $\Delta_k^{*\prime}$ with respect to δ', i.e., $\Delta_k^{*\prime} = \min\{\Delta_k, \Delta_{k+1}, \cdots, \Delta_{k+\delta'}\}$ such that the path corresponding to every element in this set would have led to a different decision $\hat{u}_k$.[1]

Note that, unlike the conventional SOVA, the nonsurviving paths are always the same regardless of the time they are traced back. Thus, once a nonsurviving path has been traced back, and its relevant information (such as the difference metric) has been stored, it can be used for later processing without the need to trace it back over and over. In this case, only one new nonsurviving path is traced back every time a new soft decision is made as opposed to $\delta + 1$ paths in the conventional case. With this, the modified SOVA becomes somewhat comparable in terms of computational complexity to the standard Viterbi algorithm. In addition, these modifications guarantee obtaining the most reliable hard decisions $\hat{u}_k$, which is of significant importance especially in the first decoder iteration as the subsequent iterations highly depend on the first iteration. Moreover, we note that the reliability of the hard decisions $\hat{u}_k$ does not depend on the reliability depth δ' as they are obtained after the whole data block has been decoded. As for the reliability value $\Delta_k^{*\prime}$, as we shall show below, the modified SOVA becomes less sensitive to the reliability depth δ' as the code rate increases. Lastly, we conclude that such modifications result in a reduction in the trace-back computations, with a drastic cut down when considering high code rates.

III. SIMULATION RESULTS

The simulation model for the parallel and serial concatenations systems on the precoded PR4/AWGN channel is shown in Fig. 1. For the parallel concatenation, the outer code comprises two identical RSC codes. Each RSC encoder employs the generator polynomials $(g_1, g_2) = (23, 31)_{oct}$, where g_1 is the feedback polynomial and g_2 is the feedforward polynomial. As for the serial concatenation, the outer code is a single RSC code whose encoder employs the same polynomials mentioned above. The inner code for both systems is the precoder $\frac{1}{1 \oplus D^2}$ followed by the polynomial $1 - D^2$, where $\oplus$ indicates modulo-2 addition.[2] All simulations were done with an S-random interleaver of size $N = 4096$ where the value $S = 31$ was used [14]. Employing such an interleaver results in eliminating the error sequences that dominate the performance in the floor region, and, consequently, lowering the error rate floor of the performance curves [6]. For PCC's (SCC's), code rates of the form $\frac{k_0}{k_0+1}$, where $k_0 = 4$, 8, and 64, were achieved by saving the second bit in every $2k_0$-bit (k_0-bit) parity block of each RSC encoder output and puncturing the rest.

The error rate performance (P_b) of the various PCC's versus E_b/N_0 is plotted in Fig. 3 (for 15 iterations). As a baseline, the solid curves correspond to the "all APPs" configuration

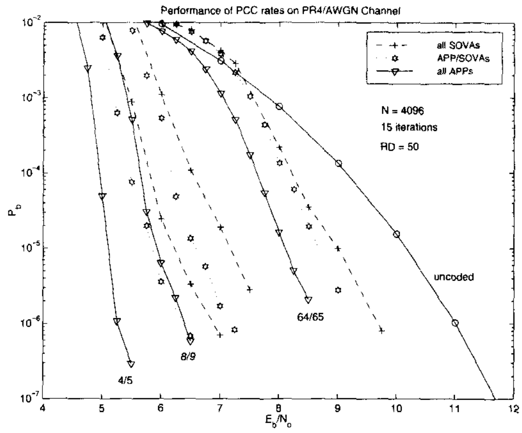

Fig. 3. Performance of the various PCC rates on the PR4/AWGN channel. (RD in the figure refers to the reliability depth, δ'.)

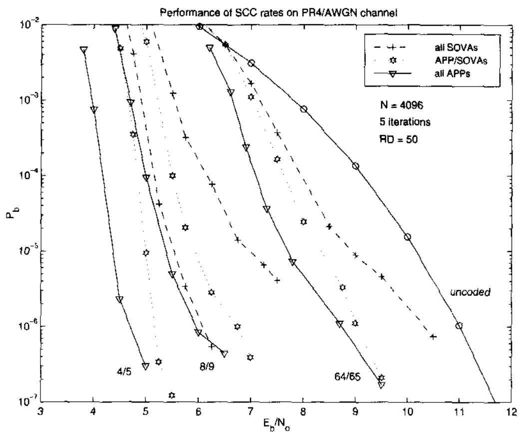

Fig. 4. Performance of the various SCC rates on the PR4/AWGN channel.

in which two APP/RSC decoders are employed for the outer code and one APP/PR4 detector is employed for the inner code. When the two APP/RSC decoders are replaced with two matching iterative SOVA/RSC decoders, the error rate performance degrades by about 0.7 dB at $P_b = 10^{-5}$ (dotted curves). The bit error rate performance degrades by an additional 0.3 to 0.5 dB at $P_b = 10^{-5}$ when the APP/PR4 detector is replaced with a matching SOVA/PR4 detector (dashed curves). For example, when employing SOVA's everywhere, the performance degrades by 0.9 dB for rate 64/65, and 1.2 dB for rate 4/5 at $P_b = 10^{-5}$ relative to the "all APPs" situation. A performance improvement of over 1 dB for rate $64/65$ and over 4 dB for rate $4/5$ is achieved relative to uncoded PR4 signaling (dashed curves).

Fig. 4 presents the bit error rate performance, P_b, of the various SCC's versus E_b/N_0 (for 5 iterations).[3] We observe from the figure that when the APP/RSC decoder is replaced with a

[1] Note that $\Delta_k^{*\prime}$ need not equal Δ_k^*.

[2] We remark that the employed precoder is not the optimal for the PR4 signaling scheme where optimality is in the sense of achieving the lowest error rate floor for a given outer code and PR scheme [13].

[3] We mention that the reason for choosing 15 iterations in PCCs and 5 iterations in SCCs was to obtain the best performance possible in both cases. In general, the maximum number of iterations required to achieve the best performance in SCCs is smaller than that in PCCs. We attribute this to the fact that extrinsic information exchanged iteratively between the inner detector and outer decoder correlates faster in the SCC case. Nonetheless, we have observed in both cases that after 3 iterations the improvement in performance becomes marginal.

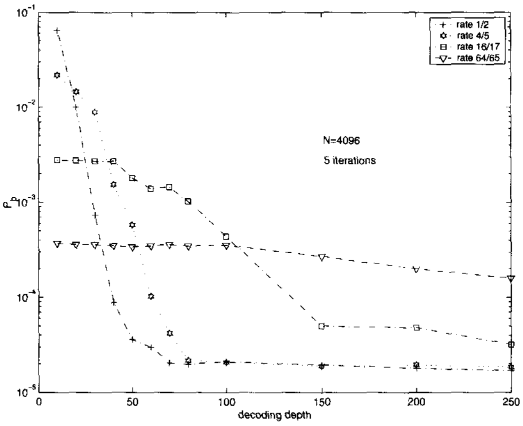

Fig. 5. Performance comparison between the modified SOVA and the conventional SOVA while fixing the reliability depth to $\delta' = 50$ that achieves about $P_b = 10^{-5}$ for all cases (PCC codes are considered here.)

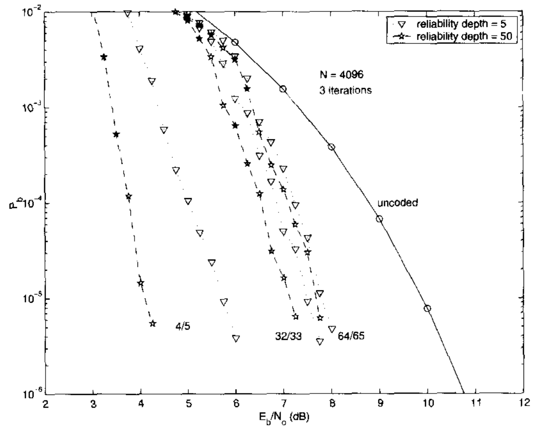

Fig. 6. Performance of the modified SOVA algorithm as a function of the reliability depth, δ'.

Table 1. Information Rates for the PR4/AWGN Channel.

Code Rate	4/5	8/9	64/65
AWGN (dB)	2.07	3.01	5.67
PR4/AWGN (dB)	2.5	3.8	6.3

matching iterative SOVA/RSC decoder, the error rate performance degrades by about 0.6 dB at $P_b = 10^{-5}$ (dotted curves). The error rate further degrades by an additional 0.6 to 1.0 dB at $P_b = 10^{-5}$ when the APP/PR4 detector is replaced with a matching iterative SOVA/PR4 detector (dashed curves).

Before we proceed further in the discussion of simulations results, we believe it is of interest to include the capacity limits and achievable information rates for these codes on the PR4/AWGN channel under consideration. Shamai *et al.* [15] derived upper bounds on the achievable rates on this channel where these upper bounds are essentially the capacity of the binary input AWGN channel (no partial response). Recently, a number of papers [16]–[18] have appeared independently and at about the same time in which tighter bounds were derived for channels with memory, including a number of PR channels. A summary of these results are shown in Table 1. (For comparison, these codes are only about 0.7 dB from their capacity limits on the binary AWGN channel [19].)

In Fig. 5, we present a performance comparison between the modified SOVA and the conventional SOVA. In the figure, we examine P_b as a function of SOVA decoding depth δ for various PCC's. To produce each curve in the figure, we first fixed E_b/N_0 to the value corresponding to $P_b = 10^{-5}$ when the reliability depth was set to $\delta' = 50$, i.e., when adopting the modifications in the decoding process outlined above. We then plotted P_b as function of δ for several code rates.[4] We observe that even when $\delta = 250$, none of the codes have yet reached $P_b = 10^{-5}$. We observe also that the higher rate codes require larger values of δ for a given value of P_b (after some transition region). This is due to the following. The hard decisions $\hat{u}_k$ become more reliable as δ increases because they include more parity bits. But for higher code rates such as the rate 64/65, the improvement is gradual and is still slight even when $\delta = 250$ (when $\delta = 250$, there are only 4 parity bits among the 250 code bits).

In Fig. 6, we examine the sensitivity of the modified SOVA with respect to the reliability depth and code rate (for 3 decoder iterations). In the figure, we present the trade-off between using $\delta = 5$ and $\delta = 50$ for rates 4/5, 32/33 and 64/65. As observed from the figure, when using $\delta = 5$ instead of 50, the performance degrades by about 1.5 dB for rate 4/5, 0.3 dB for rate 32/33, and only about 0.15 dB for rate 64/65. This is a very interesting result because, unlike the conventional SOVA, as the code rate increases the algorithm becomes less sensitive to the reliability depth [20]. This leads to a considerable cut down in the trace-back computations. Furthermore, we observe that rate 64/65 code (3 iterations and $\delta = 5$) achieves a performance gain of 2-plus dB over uncoded antipodal signaling at $P_b = 10^{-5}$.

In conclusion, the modified SOVA has several advantages over the conventional SOVA including:

1. Performance degradation remains almost the same for arbitrary code rates. This is obvious from Fig. 3 and Fig. 4.
2. The modified SOVA always outperforms the conventional SOVA with a significant difference for high code rates. This is obvious from Fig. 5.
3. Unlike the conventional SOVA, the modified SOVA becomes less sensitive to δ as the code rate increases as shown in Fig. 6, which plays a major role in cutting down the trace-back computations.

We finally mention that the aforementioned advantages come at the cost of increasing storage complexity as we need to store u_k and k for all $k \in \{1, 2, \cdots, N\}$. However, this cost is by far negligible as a trade-off for improving the performance and cutting down trace-back computations.

IV. CONCLUSIONS

We have extended the derivation of the SOVA algorithm for PR channels. We have also introduced some modifications in the decoding process of the algorithm so that it now works con-

[4]For example, when $\delta = 200$, it is required to trace back 201 nonsurviving paths to make a hard decision on a bit as well as obtaining the corresponding reliability value. This repeats every time a new soft output is calculated.

sistently for arbitrary high code rates. We have argued that the modified SOVA always outperforms the conventional SOVA with a considerable reduction in the trace-back computations. This argument was also supported by computer simulations. The sensitivity of the modified SOVA has also been studied with respect to the reliability depth and code rate. The conclusion was that it becomes less sensitive to the reliability depth as the code rate increases. Lastly, we mention that the modified SOVA can be extended to arbitrary ISI channels in a straightforward manner.

ACKNOWLEDGEMENT

The Author would like to thank Zheng Zhang of Arizona State University for providing the information rates on the PR4/AWGN channel reported on the third row of Table 1.

REFERENCES

[1] W. Ryan, "Performance of high rate turbo codes on a PR4-equalized magnetic recording channel," in *Proc. IEEE Int. Conf. Commun.*, vol. 2, June 1998, pp. 947–951.

[2] W. Ryan, L. McPheters, and S. McLauphlin, "Combined turbo coding and turbo decoding for PR4-equalized Lorentzian channels," in *Proc. Conf. Information Sciences and Systems*, Mar. 1998, pp. 489–493.

[3] L. McPheters, S. McLaughlin and E. Hirsh,"Turbo Codes for PR4 and EPR4 Magnetic Recording," in *Proc. the 32nd Asilomar Conf. Signals, Systems, and Computers*, vol. 2, Nov. 1999, pp. 1778–1782.

[4] L. McPheters, S. McLaughlin, and K. Narayanan, "Precoded PRML, serial concatenation, and iterative (turbo) decoding for digital magnetic recording," *IEEE Trans. Mag.*, vol. 35, pp. 2325–2327, Sept. 1999.

[5] T. Souvignier *et al.*, "Turbo Decoding for PR4: Parallel Versus Serial Concatenation," in *Proc. 1999 Int. Conf. Commun.*, vol. 3, June 1999, pp. 1638–1642.

[6] A. Ghrayeb and W. Ryan, "Concatenated system design for storage systems," *IEEE J. Select. Areas Commun.*, vol. 19, no. 4, pp. 709–718, Apr. 2001.

[7] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. 1995 Int. Conf. Commun.*, pp. 1009–1013.

[8] L. Gong, W. Xiaofu, and Y. Xiaoxin, "On SOVA for nonbinary codes," *IEEE Commun. Lett.*, vol. 3, issue 12, pp. 335–337, Dec. 1999.

[9] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Infomation Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996.

[10] J. Hagenauer, and P. Hoeher, "A Viterbi Algorithm with soft-decision outputs and its applications," in *Proc. GLOBECOM'89*, pp. 1680–1686.

[11] J. Hagenauer, "Source-controlled channel decoding," *IEEE Trans. Comm.*, vol. 43, issue 9, pp. 2449–2457, Sept. 1995.

[12] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Infom. Theory*, Mar. 1974.

[13] W. E. Ryan, "Concatenated codes for class IV partial response channels," *IEEE Trans. Commun.*, pp. 445–454, Apr. 2001.

[14] D. Divsalar and F. Pollara, "Multiple-turbo codes for deep-space communications," *JPL TDA Progress Report* 42–121, May 1995.

[15] S. Shamai, L. H. Ozarow, and A. D. Wyner, "Information rates for a discrete-time Gaussian channel with intersymbol interference and stationary inputs," *IEEE Trans. Infom. Theory*, vol. 37, pp. 1527–1539, Nov. 1991.

[16] H. D. Pfister, J. B. Soriaga and Paul H. Siegel, "On the achievable information rates of finite state ISI channel," in *Proc. GLOBECOM'2001*, pp. 2992–2996.

[17] D. Arnold and H. Loeliger, "On the information rate of binary input channels with memory," in *Proc. Int. Conf. Commun.*, vol. 9, June 2001, pp. 2692–2695.

[18] V. Sharma and S. K. Singh, "Entropy and channel capacity in the regenerative setup with applications to Markov channel," in *Proc. Int. Symp. Infom. Theory*, p. 283, June 2001.

[19] O. Acikel and W. E. Ryan, "Punctured turbo codes for BPSK/QPSK channels," *IEEE Trans. Commun.*, pp. 1315–1323, Sept. 1999.

[20] A. Ghrayeb and W. Ryan, "Performance of high rate turbo codes employing the soft output Viterbi algorithm (SOVA)," in *Proc. the 33rd Asilomar Conf. Signals, Systems, and Computers*, vol. 2, Nov. 1999, pp. 1665–1669.

Ali Ghrayeb received the B.S. degree from the University of Jordan, Amman, Jordan, in June 1994, the M.S. degree from New Mexico State University, Las Cruces, NM, in December 1996, and the Ph.D. degree from the University of Arizona, Tucson, AZ, in May 2000, all in electrical engineering.

From 2000 to 2002, he was an Assistant Professor in the Electrical Engineering Department at the American University of Sharjah, Sharjah, U.A.E. In Summer 2001, he held a short term appointment as a Faculty Research Associate with the Telecommunications Research Center, Electrical Engineering Department, Arizona State University, Tempe, AZ. Since August 2002, he has been with the Department of Electrical and Computer Engineering, Concordia University, Montreal, Canada, where he is an Assistant Professor. His research interests include digital and wireless communications, channel coding, turbo codes, space-time codes, linear and nonlinear equalization, and coding for data transmission and storage. He has published over 20 refereed technical articles in the above research areas.