Manuscript received May 18, 2007; revised Aug. 20, 2007

Design of a Modified Bi-Directional SOVA

KINGSLEY OTENG-AMOAKO
School of Elec. Eng. and Telec.,
University of New South Wales,
Kensington, Sydney, NSW 2052
Australia
k.oteng@ieee.org

SAEID NOOSHABADI
Gwangju Institute of Science and Tech.,
Department of Information and Communication,
261 Cheomdan-gwagrio (Oryong-dong),
Buk gu, Gwangju, 500-712,
Republic of Korea
saeid@gist.ac.kr

Abstract: In this paper, a directional bi-directional Soft-Output Viterbi algorithm (SOVA) based Turbo decoder architecture is proposed. By performing an extra pass through the decoder trellis, effectively bi-directional decoding, decoder performance can be improved. The bi-directional decoding results in a 60% increase in operations over the conventional SOVA decoder, while incurring no significant increase in architecture size. The paper also proposes the use of extrinsic information in selecting the SOVA algorithm, resulting in improved decoding performance. The use of the extrinsic information as a reliability measure does not introduce any inherent latency and requires a negligible increase in hardware. Following both architecture proposals, the paper discusses practical implementation using finite precise arithmetic. Simulation results show that by combining both proposed techniques, an extra coding gain of 0.4dB can be obtained over the conventional SOVA decoder in an Additive White Gaussian Noise (AWGN) channel.

Key–Words: Information rate, adaptive modulation, adaptive coding, fading channels, concatenated coding

1 Introduction

Turbo codes have been shown to encode data using a concatenation of Recursive Systematic Convolutional (RSC) codes [1]. Turbo codes perform iterative Soft-Input Soft-Output (SISO) decoding, based on either the Maximum A Posteriori (MAP) algorithm or the Soft Output Viterbi Algorithm (SOVA). Iterative turbo decoding consists of two decoders separated by an interleaver, where both decoders are capable of exchanging extrinsic information. It has been detailed that the decoder arrangement effectively employs an exchange of information, termed extrinsic information between the two parallel SISO decoders. [2].

SISO decoders generate an extrinsic information output from which an a-posteriori Log-Likelihood Ratio (LLR) and an *a-priori* ratio are obtained. The concatenated structure of the SISO decoder can be observed in Figure 1, where $L_e^{(j)}(d_i)$, $L_\mathrm{a}^{(j)}(d_i)$ and $L_\mathrm{r}^{(j)}(d_i)$ denote the extrinsic, a-priori and a-posteriori LLR outputs of the d_i-th bit of the j-th decoder.

1.1 MAP decoding algorithm

The MAP algorithm is an optimal SISO decoder that employs symbol-by-symbol detection algorithm based on the Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm [3], [4]. In practice the sub-optimal Max-Log-MAP and Log-MAP decoders are usually used as they provide more feasible implementations [1], [5, 6].

1.2 SOVA decoding algorithm

Alternatively, SOVA can be used for iterative suboptimal decoding. The SOVA performs 0.3dB less than the MAP algorithm [7]. However, the decoding complexity in terms of the number of computation operations, of the SOVA can be less than half that of the MAP algorithm. Subsequently SOVA is the preferred algorithm for low-complexity and low-power implementations thus the preferred architecture for mobile applications.

The contributions of the paper are as follows:

1.2.1 Adaptive SOVA decoding

The paper presents a reliability based SOVA turbo decoding architecture. In addition an adaptive directional and bi-directional SOVA decoder is also explored.

1.2.2 Adptive directional, bi-directional SOVA decoding

In [8], a decoder structure was presented termed bi-directional that improved the SOVA by selecting the branch metric at time i corresponding to an information bit $d_i \in \mathbf{d} = (d_1, ..., d_K))$ from the forward pass or backward pass, $\Delta_f(d_i)$ and $\Delta_\text{b}(d_i)$ respectively. At low Signal to Noise Ratios (SNR), the bi-directional SOVA results in a coding gain of 0.35 dB over the conventional SOVA. However, the bi-directional SOVA results in twice the operations of SOVA as shown in Table 1. In addition, the bi-directional SOVA offers no performance improvement over the conventional SOVA at high SNR.

Given that the SOVA and bi-directional SOVA are based on identical trellis structures, the bi-directional decoding can be achieved based on a minor modification of the conventional SOVA architecture. In this work SOVA is employed with an additional backward pass in the architecture in order to achieve bi-directional SOVA. The synergies in the two structures suggests an “aggressive” and “standard ” decoding approach from a single core depending on a given reliability indicator.

The paper suggests a threshold strategy to exploit the similarity between the SOVA and bi-directional SOVA based on the observed SNR. Given a required Bit Error Rate (BER) employed as a threshold, the error rate is initially mapped to a corresponding reliability. The reliability, as defined in this work, is the a-posteriori average LLR of the received information bits. The architecture performs either a directional or bi-directional pass based on whether the observed code reliability is above or below the mapped threshold. In assessing the codeword reliability for the SOVA turbo decoder core, the reliability is considered generated during the 8-th iteration of the SOVA turbo decoder. The paper also considers the effects of quantization on the performance of the SOVA decoder. In addition, the effects of quantization on the extrinsic information transfer are considered and the minimum number of bits required for extrinsic information output is determined.

1.2.3 Hardware considerations for SOVA decoding

In the practical implementation of iterative decoders, the effects of finite precision are particularly relevant in examining the manner in which the combined codewords act to increase maximum likelihood decoding. The finite precision factors that determine the performance of turbo decoding algorithms include the length of received codewords and the length of extrinsic information output.

1.2.4 Extrinsic information scaling

Various schemes have been proposed to improve the performance of SOVA [8, 10, 11]. It has been shown that by scaling the extrinsic information magnitude, the overall performance of the SOVA can be improved [12, 13]. Thus a technique is proposed to improve the throughput performance of the SOVA by scaling the extrinsic information results based on the channel reliability observations.

1.3 Organization of this paper

This paper is organized as follows. In Section 2, a brief introduction is given on the Viterbi algorithm, SISO decoding, SOVA based turbo decoding and the bi-directional SOVA. In addition, some practical implementation issues are discussed including approaches to improving the performance of the conventional SOVA turbo decoder. In Section 3, the selection of a directional or bi-directional architecture based on an observed reliability estimate is discussed. In Section 4, an architecture for the efficient high-throughput implementation based on parallel decoding of bi-directional SOVA is presented. Section 4 also presents the effect of finite word quantization and fixed point arithmetic on error performance. It also presents the work on the extrinsic information correction to remove the performance gap between SOVA and MAP decoding. Section 5 presents the simulation results and provides a discussion on the results of the work. Section 6 concludes the work.

2 SISO SOVA Decoding

2.1 Viterbi decoding

The decoding of convolutional codes has most frequently been achieved through the use of the Viterbi algorithm. The Viterbi algorithm provides a Maximum Likelihood (ML) solution of the the Maximum A Posteriori (MAP) probability estimation of state sequences in discrete time finite-state Markov process [14]. The Viterbi algorithm results in the mathematical optimum estimate of the received sequence. The finite-state Markov process employed during the encoding stage can be represented as a trellis employed in decoding by the Viterbi algorithm. The algorithm works by assigning at each stage in decoding, denoted by an index i corresponding to information bit d_i, a metric to each branch in the trellis.

The Viterbi algorithm performs a forward pass to determine the weights of a trellis, termed path metrics. The branch metrics of each node in the trellis are obtained by determining the Hamming weight of received data. Subsequently at each decoding stage

Table 1: Number of computation operations per bit for various SISO decoding algorithms (v is the Code memory.)

Operation	Max-Log MAP	Log-MAP	SOVA	bi-directional SOVA
max opers.	$5 \times 2^v - 2$	$5 \times 2^v - 2$	$3(v+1) + 2^v$	$5(v+1) + 2^v$
additions	$10 \times 2^v + 9$	$15 \times 2^v + 9$	$2 \times 2^v + 8$	$3 \times 2^v + 8$
mult. by ± 1	8	8	8	16
bit comps			$6(v+1)$	$10(v+1)$
look-ups		$5 \times 2^v - 2$		
total opers.	$15 \times 2^v + 17$	$20 \times 2^v + 17$	$3 \times 2^v + 9v + 25$	$4 \times 2^v + 15v + 39$
opers. for v=2	87	97	55	85
opers. for v=3	137	177	76	116

in the trellis, a branch is selected from a set of two branches depending on the received sequence. The path with the best ensemble selection of branch metrics is chosen as the likely transmitted sequence during the traceback operation. Thus a received sequence of bits, corresponding to a transmitted codeword **c**, results in a unique path through the trellis.

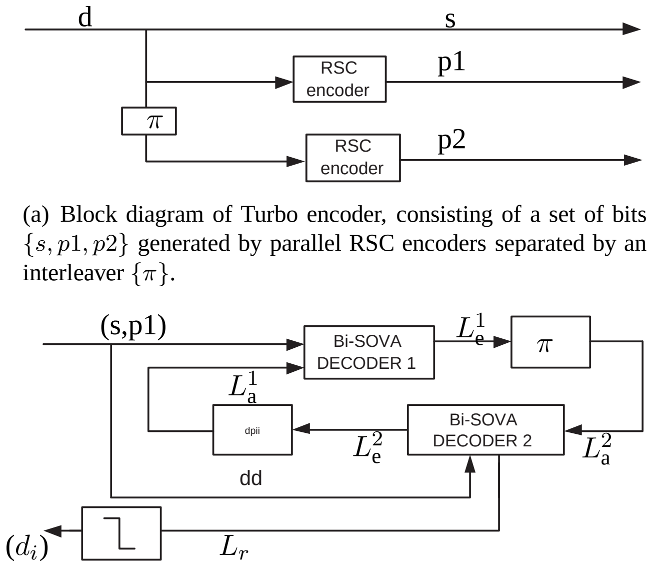
The Viterbi algorithm generates a trellis based on the estimation of the state of code during a successive time instants and consequently selects the most likely decoding path [15]. The selection of the most likely path through the trellis of the Viterbi algorithm is termed ”hard” decision decoding of a received sequence.

2.2 SISO decoding

The hard decision represents a loss of information as subsequent decoding steps are unable to utilize the sequence estimations made by the previous decoder. The use of an algorithm that generates estimations on a bit-by-bit basis and accepts soft-decision inputs from the previous iteration, termed SISO decoding, results in significantly improved performance [1].

2.3 SOVA decoding

The Viterbi algorithm has been modified to generate a low-complexity soft-decision estimate of a received sequence, in a process termed SOVA [10]. SOVA is a modified version of the Viterbi algorithm, similarly employing a trellis-based selection of the most likely path at each stage i corresponding to information bit d_i in the trellis. The SOVA effectively generates a soft-output, by determining the difference between the survivor path and the competing path as observed in Figure 2, effectively estimating a reliability of the received codewords. Given the encoded bit sequence and the bit sequence corrupted by additive white gaussian noise (AWGN), denoted as **c** and **r** respectively, the reliability of the codeword is given by the square Euclidean distance between the bits of $\mathbf{c} = (c_1, ..., c_N)$ and $\mathbf{r} = (r_1, ..., r_N)$ as,



(a) Block diagram of Turbo encoder, consisting of a set of bits $\{s, p1, p2\}$ generated by parallel RSC encoders separated by an interleaver $\{\pi\}$.

(b) Block diagram of an iterative decoder consisting of parallel SISO decoders, deinterleaver $\{\pi^{-1}\}$ and the corresponding hard estimate, $\{d_i\}$

Figure 1: Model of the Turbo Code System

$$\mu = E||\mathbf{r} - \mathbf{c}||^2 \tag{1}$$

Denoting the sequence generated in the parallel encoders of the rate 1/3 turbo code, as $\mathbf{c_1}$ and $\mathbf{c_2}$ respectively, the reliability divergence Δ is given by the expectation,

$$\Delta = E\big| ||\mathbf{r} - \mathbf{c_1}||^2 - ||\mathbf{r} - \mathbf{c_2}||^2 \big| \tag{2}$$

In SOVA, given path-1 and path-2 through a trellis at step i (corresponding to information bit d_i), corresponds to the transition between states s_k and s_{k+1}

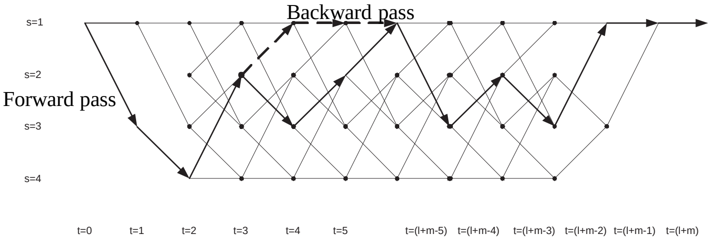


Figure 2: Trellis diagram with competing paths of forward and backward passes of SOVA decoding

thus denoted $\Gamma^1(s_k, s_{k+1})$ and $\Gamma^2(s_k, s_{k+1})$, the reliability difference between the two metrics can be given as,

$$\Delta(i) = \max\{\Gamma^1(s_k^i, s_{k+1}^i\} - \min\{\Gamma^2(s_k^i, s_{k+1}^i)\} \tag{3}$$

Thus the reliability, $\mu(i)$ is given as

$$\mu(i) = \min\Bigg(\mu'(i), \Delta(i)\Bigg) \tag{4}$$

where $\mu'(i)$ is the reliability value stored based on the previous SOVA pass. It is observed from Equation (4), that the reliability estimate generated by a SOVA is optimized by minimizing the difference between the survivor path and the competitor path.

2.4 Bi-directional SOVA

The SOVA does not always select the best branch. In Figure 2, the divergence of segments in the forward pass and backward pass through the trellis are observed. The observed divergence can lead to the correct bit sequence being discarded during the selection process, prior to the path re-merging with the ML-path. Thus the bi-directional SOVA achieves a performance improvement over SOVA by minimizing the likelihood of the wrong path being selected, through a trellis at each stage, as given by,

$$\mu(i) = \min\Bigg(\mu'(i), \Delta_f(i), \Delta_{\mathrm{b}}(i)\Bigg) \tag{5}$$

Although the best path can be discarded by the forward pass, the discarded path may however survive during the backward pass of SOVA. Thus by selecting the pass with the minimum difference between the survivor path and the competition path i, the overall performance of the SOVA algorithm is improved [8].

2.5 SOVA Turbo codes

Turbo codes utilize multiple SISO decoders arranged in serial or parallel, separated by an interleaver and deinterleaver. Close to capacity performance is achieved by Turbo codes by passing soft-information, termed extrinsic information, between a parallel concatenation of two RSC codes [10, 1].

Considering the structure of a Turbo encoder of rate 1/3 in Figure 1(a), where $\mathbf{d} = \{d_1, ..., d_N\}$ are the information bits, and $\mathbf{p1} = \{p1_1, ..., p1_N\}$ and $\mathbf{p2} = \{p2_1, p2_2, ..., p2_N\}$ are the parity bits generated by the first and second RSC encoders, respectively.

At the receiver, decoding is performed by a SISO decoder, either based on the MAP algorithm or the SOVA [10, 9]. The arrangement relies on the exchange of extrinsic information between successive SISO decoders. The *extrinsic information for decoder 2* , L_e^2, is obtained as,

$$L_{\mathrm{e}}^{2\prime} = L_{\mathrm{r}}^{2\prime} - \mu_{\mathrm{s}} \frac{2}{\sigma_{\mathrm{d}}^2} \gamma_{\mathrm{d}} - L_{\mathrm{a}}^{2\prime} \tag{6}$$

where μ_{d}, σ_{d}^2, and γ_{d} are the mean, variance and SNR, respectively of the received information bits $\mathbf{d}$. $L_{\mathrm{r}}^{2\prime}$ is the *a-posteriori* (LLR) soft output in the current decoding pass. $L_{\mathrm{a}}^{2\prime}$ is the *a-priori* soft external input to decoder 2 in the current decoding pass. $L_{\mathrm{a}}^{2\prime}$ is generated as external soft output (L_{e}^1) by decoder 1 during the previous decoding pass and interleaved.

3 Directional And Bi-Directional SOVA Turbo Decoding

Let μ denote the reliability of a bit sequence as a function of the *a-posteriori* LLRs of the information bits, averaged over realizations of noise, sets of transmitted bits and channel taps at a fixed SNR. Following the l-th received bit sequence, the *a-posteriori* extrinsic LLR of the i-th information bit of the l-th received bit sequence can be defined as

$$L_e(d_i|r_l) = \log \frac{P[d_i = 0|r_l]}{P[d_i = 1|r_l]} \tag{7}$$

Given the extrinsic *a-posteriori* LLR values for all transmitted information bits at the output of a SISO decoder, the codeword reliability μ is given as

$$\mu = E_{n,\overline{\Pi}}\left[\left|\log \frac{P[d_i = 0|r_l]}{P[d_i = 1|r_l]}\right|\right] \tag{8}$$

where $E_{n,\overline{\Pi}}$ denotes the expectation taken over the noise realizations n and the cumulative set of transmitted bits, $\overline{\Pi}$. The codeword reliability μ can be considered as a measure of the likelihood of correctly decoding the received sequence. For a given sequence

Table 2: Mapping of target BER to target L_e on the 8-th iteration of a Turbo code for n=3072

BER	0.1	0.01	0.001	0.0001	0.00001
L_e	8.0	15.0	23.0	32.0	42.0

of information bits, an estimate of the codeword reliability, μ is obtained at the receiver by computation of an empirical average

$$\mu = \overline{|L_e|} \tag{9}$$

consequently the codeword reliability μ can be denoted as,

$$\mu = \frac{1}{N}\sum_{i=1}^{k}\left|L_e(d_i|r_l)\right| \tag{10}$$

where k is the number of information bits of a received sequence. In order to explore the relation between the SOVA turbo decoder and the reliability of the received codewords, a Monte Carlo simulation is performed on large numbers of transmitted bit sequences. The BER and codeword reliability, μ, are plotted for the conventional SOVA as a function of SNR in a (2-D) graph, for the 8-th iteration of a turbo decoder, as shown in Figure 3. From the graph a mapping of the BER to μ is obtained. Thus given the target BER, the target $\underline{\mu}$ is obtained. ¿From μ^1, the reliability determined from the first decoder pass, the SOVA turbo decoder operation can be described in Algorithm 1:

Algorithm 1 Determine $\min \Delta(d_i)$

```
Ensure: $\Delta_f(d_i) = \infty$
  if $\mu^1 < \underline{\mu}$ then
    determine $\Delta_f(d_i)$
  else
    determine $\Delta_f(d_i)$
    determine $\Delta_\mathrm{b}(d_i)$
  end if
  $\Delta(d_i) \Leftarrow \min\Big(\Delta_f(d_i), \Delta_\mathrm{b}(d_i)\Big)$
```

where Δ_f and Δ_b are the difference between the ML state metric and the competitor paths of the forward and backward passes respectively. The scheme utilizes the reliability information as a metric.

For a given L_e threshold, the architecture performs the SOVA in backward pass addition to a forward pass in order to achieve the bi-directional SOVA. The mappings of BER to L_e is obtained from Figure 3. A corresponding set of target BER to target L_e mappings are presented in Table 2.

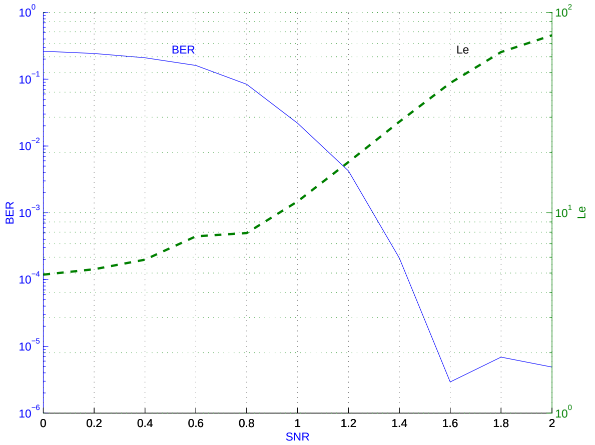


Figure 3: Mean magnitude of the extrinsic information $\overline{|L_e|}$ as a function of SNR in an AWGN channel, for eight-iterations of the SOVA based decoder employing a $\mathbf{g}$ =$(15/17)$ and seven-bit quantization (7:3:3).

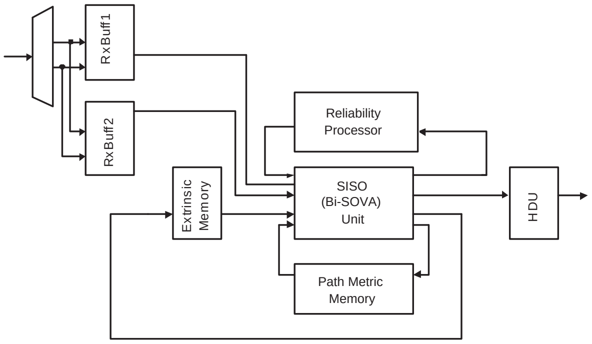


Figure 4: Single receiver architecture for a bi-directional SOVA decoder

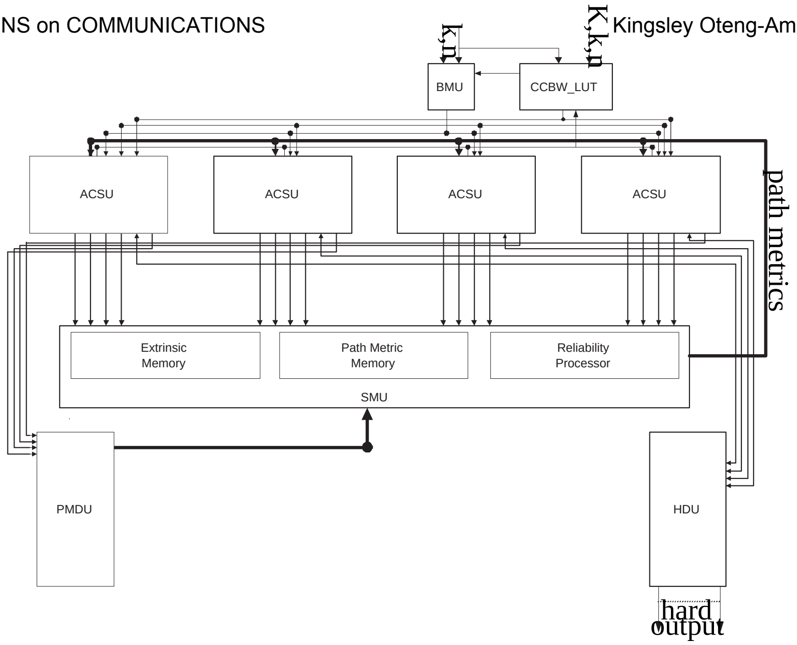


Figure 5: SOVA decoder architecture employed in singular receiver architecture.

4 SISO Decoder Hardware Architecture Overview

The schematic block of the proposed bi-directional SOVA architecture is shown in Figure 4. Soft output from the SISO (bi-SOVA) decoder unit is passed to the Hard Decision Unit (HDU) that generates a hard decision (binary) output from a given soft-output.

The architecture of the bi-directional SOVA hardware unit is described in Figure 5. The *HDU* unit, and the *Extrinsic Memory Path Metric Memory* and *Reliability Processor* units in the feedback path of Figure 4 are embedded in the architecture of Figure 5.

The bi-directional architecture consists of the following components, a *Branch Metric Unit* (*BMU*), a *Component Code Branch Weight LUT* (*CCBW-LUT*), an *Add Compare Select Unit* (ACSU), *Survivor Memory Unit* (*SMU*), *Path Metric Decision Unit* (*PMDU*) and *Decision Branching Unit* (*DBU*).

- The BMU computes the branch metrics at each branch of the successive states of the trellis. The branch metric is obtained by comparing the received data with the branch weights. For a rate k/n constituent encoder, the set of ensemble branch weights has 2^n possible pairs. In our architecture, the set of branch weights are precomputed for all component codes and stored in CCBW-LUT.

- The component CCBW-LUT holds precomputed branch weights of various component codes; given a particular constraint length K and rate k/n.

- The ACSU shown in Figure 6, determines the survivor branch metric from two pairs of branch metric. At a time i, corresponding to information bit d_i the ACSU obtains from memory the bits of the accumulated path metric and adds the least of the branch metrics in the current state, before storing the result back to memory.

- The PMDU generates the corresponding soft-outputs L_e, of the SISO decoding iteration.

- The SMU performs the trace back operation after the accumulated path metric has been determined and stores the result in memory. The SMU contains both the Extrinsic Memory and Path Metric Memory units. The SMU is also responsible for the reliability preprocessing.

- The HDU generates the post-decoder bit d_i, based on a hard decision of accumulated path metric.

4.1 Effect of finite word quantization on error performance

In the hardware implementation of SOVA, the quantization of the codewords causes degradation in the BER performance of the proposed system. However by quantizing the data, the size of the memory required to buffer received codewords and store L_e is minimized. In addition, the overall throughput of the decoder in the receiver architecture is also increased when the arithmetic precision of hardware is reduced.

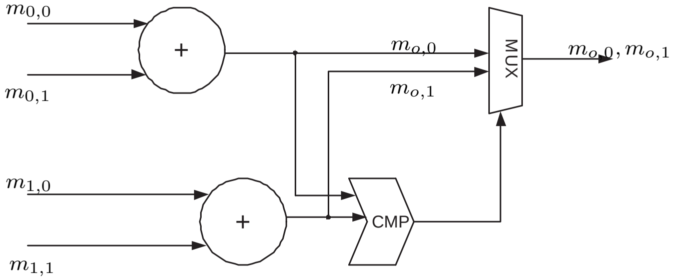


Figure 6: ACSU, as employed in decoder architecture; consisting of input branch metrics of $\{m_{0,0}, m_{0,1}$ $m_{1,0}, m_{1,1}\}$ and corresponding selected output branch metrics of $\{m_{o,0}, m_{o,1}\}$

The number of bits used to represent a real number is denoted as (F:R:M), where F is the total number of bits used to represent the quantized real number, R bits are used to represent the integer part of the real number and M bits to represent the fractional part; such that $(F = R + M + 1)$.

4.2 Effect of upper bounding on extrinsic information

In this subsection, the effects of upper bounding the extrinsic information are examined. It is observed from Figure 3 that $|\overline{L_e}|$ increases unbounded as SNR. The reduced number of bits used to represent L_e are limited and correspondingly there exists a risk of the bit overflowing during the calculation of L_e. However by lowering the upper bound $|L_e|$, a reduction in complexity can is achieved in the SOVA as a smaller bit representation of (F:R:M) is required. It is observed from Figure 7, that for an upper bound of 8-bits per word, both the waterfall region and error floor achieve good performance. For SOVA a compromise of 8-bits per word quantization offers an 0.20dB increase in performance over the 2 and 4 bit per word quantizations. The pseudocode detailed in segment 2 of Figure 8 is used to upper bound the extrinsic information to a value $L_{e,max} = 8$.

4.3 Effect of Finite Word Quantization on Extrinsic Information

During SOVA decoding, the a-priori information is generated as a difference between the received sequence path metric and the path metric of the competing path [10]. In a channel where the SNR is high, the large difference between the path metric and the competing path results in a correspondingly large variation in L_e. Thus, in fading channels where the SNR varies rapidly, the corresponding variation in SNR within a received codeword can be significant. From Figure 3, the variation in the mean magnitude of L_e, $|\overline{L_e}|$,

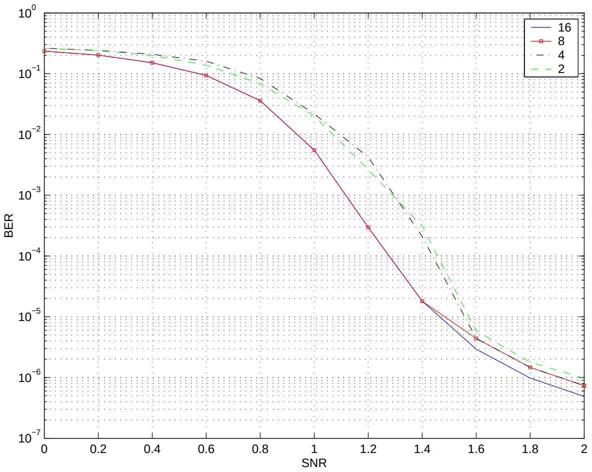


Figure 7: Performance of turbo decoder based on upper bounds of {2, 4, 8, 16} bits for extrinsic information quantization, L_e, for **g** = (15/17) in a fading channel

```
//Determine Correction Factor
if received bit sequence
use SNR to:
determine $K_\gamma$
end if

//Upper Bound Correction Factor
for iteration of SISO decoder loop
use $L_r^2$, $L_e^1$and $K_\gamma$ to:
calculate $L_e^2$
for corresponding $L_e^2$ do
//Lower Bound Correction Factor
if $L_e^2 < -L_{e,max}$ then
set $L_e^2 = -L_{e,max}$
//Upper Bound Correction Factor
else if $L_e^2 > L_{e,max}$ then
set $L_e^2 = L_{e,max}$
end if
end do
end loop
```

Figure 8: Pseudocode to perform extrinsic information correction in the SISO Unit

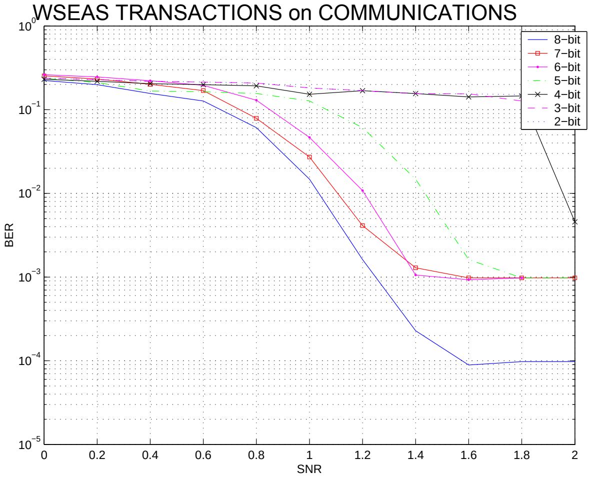


Figure 9: Performance of turbo decoder based on { 2,3,4,5,6,7,8 } levels of extrinsic information quantization, L_e, for **g** = (15/17) in an AWGN channel

as a function of SNR can be large. The variation in L_e can be minimized by taking a log of the value, although L_e variation still remains significant [7]. Thus the large variation in L_e requires an increased word length in order to maintain the dynamic range necessary for computing reliability information.

Decreasing the number of bits employed to represent L_e, minimizes the storage requirements and increases the speed of the SOVA decoding core. However given that L_e is used as a measure of reliability in the decoder, limiting the dynamic range of L_e impairs decoding performance, an effect termed quantization loss [16, 17]. In Figure 9, the performance of the singular receiver architecture is examined for varying bit quantization lengths of L_e. In Figure 9, it is observed that using less than seven-bits (one sign-bit and four magnitude-bits) to represent extrinsic information results in a performance degradation. Thus based on the results, a (7:3:3) numbering system is employed.

4.4 Extrinsic Information Correction

The 0.3 dB performance gap between the MAP algorithm and SOVA has been shown to be due over estimation of the channel reliability and correspondingly an over estimation in the extrinsic information [12, 13]. An over estimation occurs due to the assumption that that noise is Gaussian distributed, being inaccurate in practical circumstances [12].

It has been shown that by scaling the magnitude of the extrinsic information magnitude the overall SOVA performance is improved [18]. The scaling requires a correction factor to be determined based on the distribution of received bits and the subsequently scaling of the extrinsic information bit accordingly. Based on Bayes Rule, the scaling factor as a function of the distribution of received bits given as [18]:

$$L_e' = \mu\frac{2}{\sigma^2}L_e \qquad (11)$$

where L_e' is the scaled extrinsic information bit, μ is the mean of the bits in the received sequence and σ^2 is the standard deviation of the bits in the received sequence. The calculation of μ and σ^2 in Equation (11) is computationally complex.

Based on the above observations of extrinsic information magnitude, a technique is proposed to improve the performance of the SOVA by subtracting a constant value from Equation (6), in order to scale the extrinsic information result. The proposal subtracts a normalization factor, as opposed to scaling (multiplying) the extrinsic information estimate, resulting in a less complex operation than previous proposals.

A precomputed normalization factor, K_γ, is chosen to limit the magnitude of the channel reliability factor. The normalization factor is computed as:

$$K_\gamma = \mu\frac{2}{\sigma^2}\overline{|L_e|} \qquad (12)$$

where $\overline{|L_e|}$ is obtained from Figure 3. In the SOVA proposal, the sequence of received bits, r, is initially summed and K_γ is chosen based on the sum of the received bits r. Thus for BPSK, given a received bit sequence of 3072 bits of all ones (BPSK symbol $= +1$), $\sum_{i=0}^{i=k-1} r_i = +3072$. Correspondingly for a BPSK received bit sequence of all zeros (BPSK symbol $= -1$), $\sum_{i=0}^{i=k-1} r_i = -3072$.

The set of precomputed K_γ for varying values of $\left(\sum_{i=0}^{i=k-1} r_i\right)n$ are shown in Table 3 for $n = 3072$. Given K_γ, the extrinsic information is obtained as follows:

$$L_e^{2\prime} = L_r^{2\prime} \pm \mu\frac{2}{\sigma^2}.\gamma_s \mp K_\gamma - L_e^1 \qquad (13)$$

The pseudocode employed for extrinsic information correction in the SISO unit is given in sections 1 and 3 of Figure 8.

4.5 Initialization

Given a component code, the corresponding trellis and an assigned set of branch weights, are precomputed and stored in a Look-Up-Table (LUT). In addition, in order to handle various bit sequence lengths, interleaver mappings are alsoprecomputed and stored in Look-Up. Prior to the initial decoding attempt, the *a-priori* reliability values are initialized to $\mu_{\mathrm{r}}'(i) \gg 0$ for all bits in the sequence.

Table 3: Scaling factor, Correction factors and bit-sums BPSK modulated data, for n = 3072

$\mu\frac{2}{\sigma^2}$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
K_γ	0	0.2000	0.4062	0.6258	0.8685	1.1489	1.4925	1.9506	2.6533	4.108
$\left(\sum_{i=0}^{i=k-1} r_i\right)k$	0	307	614	922	1229	1536	1821	2150	2457	2765

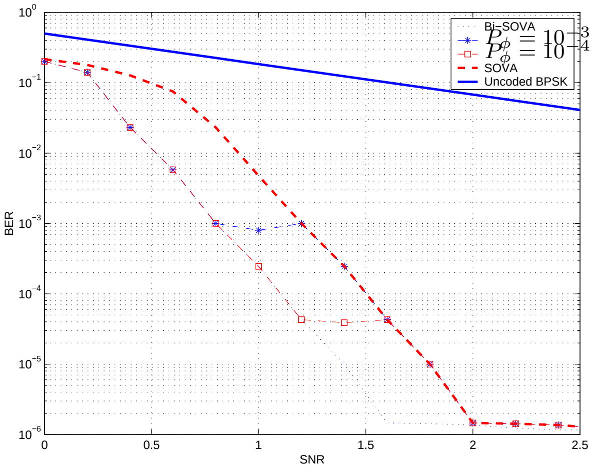


Figure 10: Comparative performance of proposed adaptive reliability based SOVA decoder ($P_\phi = 10^{-3}$ and $P_\phi = 10^{-4}$) for varying μ_r constraints and uncoded BPSK in an AWGN channel

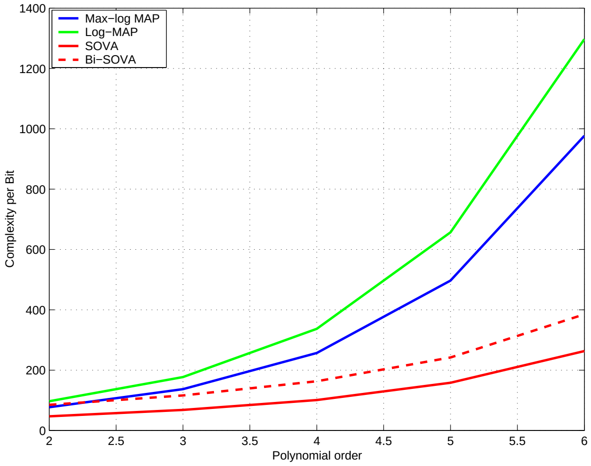


Figure 11: The number of computation operations per bit analysis for various SISO decoding algorithms for memory orders v = (2,..,6)

5 Performance of Bi-directional SOVA Architecture

In this Section the performance of the proposed SOVA architecture is examined. A component code of **g** = $(15/17)$ and an interleaver size of $n = 3072$, is employed for the simulation. The number of computation operations per bit of various schemes is examined at low SNR values for Figure 11. It is observed that the MAP algorithm exhibits the highest complexity across the entire SNR range examined.

The performance of the proposed architecture is examined in Figure 10 and compared to the performance of an uncoded BPSK system. The results indicate that an 0.4dB gain is achieved by the bi-directional SOVA over the SOVA. In addition at low SNR, the bi-directional SOVa is also shown to outperform the MAP algorithm. Further, it is observed that by selecting the target BER based on a reliability metric as detailed in Table 2, good thresholding is obtained between the decoder architectures.

6 Conclusion

In this paper, an architecture that selects between the SOVA and bi-directional SOVA based on an observed reliability of the received codeword was developed. The architecture employs a conventional SOVA decoder and performs an additional backward pass based on an observation of the reliability metric. The architecture in bi-directional SOVA mode results in a performance and complexity trade-off where an 0.4dB coding gain is achievable at the cost of a modest increase in hardware complexity. The performance of the proposed bi-directional scheme is compared to an uncoded system and shown to provide considerable performance gain.

References:

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

[2] D. Gasbert, M. Shafi, D. Shiu, P.Smith, and A. Naquib, "From Theory To Practice: An

Overview of MIMO Space-Time Coded Wireless Systems," *IEEE J. Select. Areas Commun.*, vol. 21, no. 3, April 2003.

[3] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes For Minimizing Symbol Error Rate," *IEEE Trans. Inform. Theory*, vol. 20, no. 2, pp. 284–287, March 1974.

[4] J. P. Woodward and L. Hanzo, "Comparative Study of Turbo Decoding Techniques: An Overview," *IEEE Trans. Veh. Technol.*, vol. 49, no. 6, 2000.

[5] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm: The viterbi algorithm," *IEEE Trans. Inform. Theory*, vol. 13, pp. 260–269, 1967.

[6] G. D. Forney Jr., "Maximum Likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inform. Theory*, vol. 20, May 1972.

[7] B. Vucetic and J. S. Yuan, *Turbo Codes*, 1st ed. Boston, MA: Kluwer Academic Publishers, January 2000.

[8] J. Chen, M. P. Fossorier, S. Lin, and C. Xu, "Bi-Directional SOVA Decoding for Turbo codes," *IEEE Commun. Lett.*, vol. 4, no. 12, pp. 405–408, 2000.

[9] Drago Žagar, Nenad Falamic and Snježana Rimac-Drlje, "Comparison of Decoding Algorithms for Concatenated Turbo Codes, *WSEAS Journal. of Commun.*, vol. 1, no. 4, pp. 650–662, October 2004.

[10] J. Hagenauer and P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications," in *Proc. IEEE Global Commun. Conference (Globecom'89)*, Nov. 1989, pp. 47.1.1–47.1.7.

[11] M. A. U. Bhuiyan, M. S. Hosain and S. Rahman, "Computation of Distance Spectrum and Performance Analysis of Third Generation Turbo Codes," *WSEAS Journal. of Commun.*, vol. 4, no. 11, pp. 1244–1260, April 2005.

[12] L. Lin and R. S. Cheng, "Improvements in SOVA-Based Decoding for Turbo Codes," in *IEEE Int. Conf. Commun. (ICC '97)*, 1997, pp. 1473 – 1478.

[13] L. Papke and P. Robertson, "Improved Decoding with the SOVA in a Parallel Concatenated (Turbo-Code) Scheme," in *IEEE Int. Conf. Commun. (ICC '96)*, 1996, pp. 102–106.

[14] G. D. Forney Jr., "The Viterbi Algorithm," *IEEE Proc.*, vol. 61, no. 3, pp. 268–278, March 1973.

[15] S. Lin and J. Costello Jr., *Error Control Coding*, 2nd ed. Prentice Hall, 2003.

[16] Z. Wang, H. Suzuki, and K. K. Parhi, "VLSI Implementations Issues of Turbo Decoder Design for Wireless Applications," in *IEEE Workshop on Signal Processing Systems:Design and Implementation*, Taipei, Taiwan, October 1999, pp. 503 – 512.

[17] S. Haykin, *Communication Systems*, 3rd ed. New Jersey, NJ: John Wiley and Sons, Inc., 1994.

[18] Z. Wang and K. K. Parhi, "High Performance, High Throughput Turbo/SOVA Decoder Design," *IEEE Trans. Commun.*, vol. 51, no. 4, pp. 570–579, April 2003.