MATHEMATICAL ANALYSIS OF LOG-MAP ALGORITHMS WITH SPECIAL REFERENCE TO FIXED AND FLOATING MODEL

ABHISHEK

Research Scholar, Dept. of Electronics and Communication Engineering,

Sri Satya Sai University of Technology & Medical Sciences,

Sehore, Bhopal-Indore Road, MadhyaPradesh, India

Dr.M.Vanitha

Research Guide, Dept. of Electronics and Communication Engineering,

Sri Satya Sai University of Technology & Medical Sciences,

Sehore, Bhopal Indore Road,Madhya Pradesh, India.

ABSTRACT

Turbo coding is related with two precise encoders, where the first encoder gets the source data in common request and simultaneously the second encoder gets the interleaved one. The yield is made out of source data and related equality bits in regular and interleaved areas. The equality pieces are normally penetrated so as to raise the code rate to the ideal qualities. Diverse floating-point and fixed-point execution issues of the Log-MAP algorithm for deciphering turbo codes are considered. A hypothetical system is proposed to clarify why the Log-MAP turbo decoders are more open minded toward overestimation of SNR than underestimation. A direct estimation (the Log-Lin algorithm) that takes into consideration proficient ASIC executions and results in negligible execution misfortune in floating-point models is proposed. It is additionally indicated that a Log-Lin algorithm accomplishes a similar exhibition as the Log-MAP algorithm, and that a fixed-point execution of the Log-Lin algorithm with enhanced section and quantization to 6 pieces, brings about lost execution of under 0.05 dB.

***Keywords**: LOG-MAP,* Fixed Model, Floating Model, turbo decoder, etc.

1. INTRODUCTION

As of late, turbo codes have been received as the channel coding plan by some serious correspondence norms. To improve the dependability in remote data transmission, this group of mistake adjustment code likewise ventures into some vitality resource-compelled remote applications [3]. In certain situations, for example, wireless sensor networks (WSNs), the sensor hubs just have restricted load of batteries, practically 80% of the general force scattering in the hubs is represented remote data correspondence, and lifetime of the sensor hubs is overwhelmed by the force dissemination of the correspondence module. To diminish the transmission power and to diminish the quantity of data outline retransmission in the sensor hubs however much as could reasonably be expected, the examination of turbo decoder with low force utilization and close to ideal bit error rate (BER) is a key subject. Notwithstanding, in the designing usage of turbo code decoder, the maximum a posteriori (MAP) deciphering algorithm is iteratively played out; the decoder requires huge limit memory and regular memory getting to, which lead to high power dispersal of the turbo decoder. Consequently, traditional turbo decoder isn't appropriate for power resource restricted WSNs situations. Subsequently, the vitality issue of turbo decoder has become a bottleneck limitation that ought to be truly concerned.

Because of its great exhibition, after the presentation of Turbo Code [1, 2], it has pulled in numerous analysts. There are fundamentally two sorts of algorithms in Turbo iterative disentangling algorithms. One is Log-Map and Max-Log-MAP algorithm, the other is SOVA algorithm with lower multifaceted nature. All the investigates principally focus on the interpreting execution, usage intricacy, deciphering imperfect improve, equal translating usage, fixed point evaluate on the presentation of disentangling.

1.1 Log Map Algorithm

The MAP algorithm which is utilized in the turbo decoders is operated in the logarithmic space so as to diminish the computational multifaceted nature. The calculation of log probability proportion with the estimations of state measurements is more confounded with log exponential whole count. The log exponential whole can be streamlined by Jacobian algorithm by including a remedy term alongside the maximum administrator. The amendment work count is basic as a result of the presentation and unpredictability of the turbo decoder. So a great deal of techniques is utilized to disentangle the calculation to fulfill the exhibition necessities.

The unpredictability in the MAP algorithm can be decreased by Log MAP algorithm. To evade confounded counts in the MAP algorithm, the whole MAP algorithm is determined in Logarithmic space. So the fundamental conditions in the MAP algorithm (2), (3) and (4) are changed over into log space which diminishes the quantity of increases and expansion activities used. So,ᾰk(Sk) = ln αk(Sk).

$$\hat{\alpha}(S_k) = \ln\Big(\sum_{S_{k-1}}\sum_{i=0}^{1} e^{\ln(\gamma_i(y_{sk}, y_{pk})S_{k-1}, S_k) + \ln \alpha_{k-1}(S_{k-1})}\Big) - \sum_{S_k}\sum_{S_{k-1}} e^{\ln(\gamma_i(y_{sk}, y_{pk})S_{k-1}, S_k) + \ln \alpha_{k-1}(S_{k-1})} \quad (1)$$

$$\hat{\beta}(S_k) = \ln\Big(\sum_{S_{k+1}}\sum_{i=0}^{1} e^{\gamma_i(y_{s(k+1)}, y_{p(k+1)}, S_k, S_{k+1})\beta_{k+1}(S_{k+1})}\Big) - \sum_{S_k}\sum_{S_{k+1}}\sum_{i=0}^{1} e^{\ln(\gamma_i(y_{s(k+1)}, y_{p(k+1)}, S_k, S_{k+1})\alpha_k(S_k))} \quad (2)$$

$$\hat{\alpha}(S_k) = \ln\Big(\sum_{S_{k-1}}\sum_{i=0}^{1} e^{\ln(\gamma_i(y_{sk}, y_{pk})S_{k-1}, S_k) + \ln \alpha_{k-1}(S_{k-1})}\Big) - \sum_{S_k}\sum_{S_{k-1}} e^{\ln(\gamma_i(y_{sk}, y_{pk})S_{k-1}, S_k) + \ln \alpha_{k-1}(S_{k-1})} \quad (3)$$

The above condition (1),(2) and (3) can be disentangled into general structure, which is given by the equation (4).

$$F(x_1, x_2, x_3 \ldots x_N) = \ln\Big(\sum_{i=1}^{n} e^{x_i}\Big) \quad (4)$$

The jacobian logarithm for two variable articulations is given by

$$\max{}^*(x_1, x_2) = \ln(e^{x1} + e^{x2}) = \max(x_1, x_2) + \ln(1 + e^{-|x_1 - x_2|}) = \max(x_1, x_2) + C(x) \quad (5)$$

The figuring of the adjustment term C(x) in (9) is more muddled in Log MAP algorithm. A similar type of log exponential total happens in the figuring of in reverse state metrics (βk), and LLR Λ (Dk).So, to improve the presentation of the Log Map Algorithm, a straightforward technique for execution of adjustment work must be required. The least demanding technique for discovering the rectification work is presented.But this requires more memory to store the table groupings.

2. LOG MAP FLOATING-POINT MODEL

The floating-point model of the channel and recipient structure is illustrated in Fig.1. The coded bits ct is balanced into $\pm A = \pm\sqrt{RE_b}$, where R is the code rate and E_b is normal vitality per bit. The sign is increased by a perplexing blurring factor $h_t = a_t e^{j\theta_t}$. For an AWGN channel, $h_t \triangleq 1$ and, hence, at = 1 and θ_t = 0. While for a free Rayleigh fading channel, θ_t is an arbitrary stage turn consistently appropriated in [0, 2π) and the envelope at is a Rayleigh irregular variable with unit second, i.e., the likelihood thickness capacity of a_t is $p_r(a_t) = 2a_t e^{-a_t^2}$, for $a_t \geq 0$.

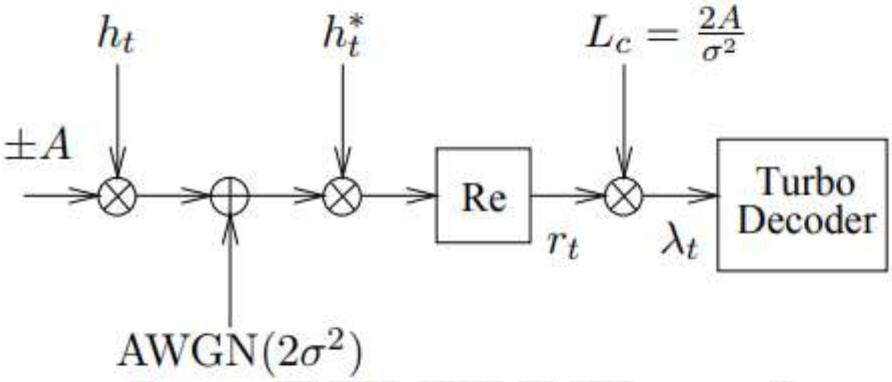

Figure 1: Floating-point receiver structure

A zero-mean complex Gaussian irregular variable with fluctuation $2\sigma^2$ = N0 is added to the blurred sign, where N0/2 is the twofold sided commotion power range thickness. The collector duplicates the sign with the form of the ideal channel gauge h∗ t. The found the middle value of sign to-clamor proportion is accordingly

$$\overline{\text{SNR}} = E[a_t^2]\frac{A^2}{\sigma^2} = \frac{A^2}{\sigma^2} = \frac{2RE_b}{N_0}. \quad \textbf{(6)}$$

The delicate qualities are then scaled by $L_c = \frac{2A}{\sigma^2}$ before taking care of into the turbo decoder.

Naturally, the necessity to scale the delicate qualities originates from the perception that the a posteriori probability of the data bit u_t can be decayed into three terms:

$$\Lambda(u_t) = \log\frac{\Pr\{u_t = +1 \mid \mathbf{r}\}}{\Pr\{u_t = -1 \mid \mathbf{r}\}} = \mathrm{V}(u_t) + L_c r_t + \mathrm{W}(u_t), \quad (7)$$

where V(u_t) and W(u_t) are the from the earlier and the outward data of bit u_t, separately. Scaling subsequently guarantees the similarity of these terms and the rightness of delicate data trade between constituent decoders.

From an implementational perspective, nonetheless, the nonlinear revision work f_{MAP} is the main activity in the Log-MAP algorithm that needs the specific greatness data. The other two activities (max and expansion) are scale-invariant. This suggests the delicate qualities don't should be scaled for a LogMax decoder, since the remedy term isn't utilized in any way. A turbo decoder dependent on the Log-Max decoder is along these lines strong against any SNR confuse issues. This is illustrated in Fig. 2 for a turbo code of square length N = 640 on the AWGN channel.

The Log-MAP turbo translating algorithm, then again, is touchy to scaling errors as revealed. Notwithstanding, the particular conduct of being more open minded toward overestimation of SNR than underestimation has not been clarified As the accompanying analysis shows, this unequal response is chiefly brought about by the nonlinearity of the rectification work.

Realizing that $r_t = \frac{1}{L_c}\lambda_t$, precisely the same Log-MAP and thus the turbo deciphering algorithms can really operate on un-scaled delicate qualities r_t by utilizing an altered delicate combiner

$$\mathrm{MAX}^*(r_1, r_2) \triangleq \frac{1}{L_c}\mathrm{MAX}^*(\lambda_1, \lambda_2) = \max(r_1, r_2) + \frac{1}{L_c} f_{\mathrm{MAP}}(L_c|r_1 - r_2|). \quad \textbf{(8)}$$

Let $z = |r_1 - r_2|$ what's more, assume the assessed scaling factor is $\hat{L}_c = \delta L_c$. Alluding to Fig. 3, we can see how the error in $\hat{L}_c$ spreads through the algorithm:

For the instance of overestimation, we have $\delta > 1$ and therefore $f_{MAP}(\hat{L}_c z) < f_{MAP}(L_c z)$. The previous is additionally downsized by partitioning δL_c. Overscaling the delicate qualities in this way implies de-accentuating the revision term. Thusly, the exhibition of the Log-MAP decoder will join to that of the Log-Max decoder in the constraint of unbounded overscaling. This intermingling can be seen at higher assessment errors towards the right-hand side of Fig. 2. The casing size is N = 640 and 8 emphasess are expected.

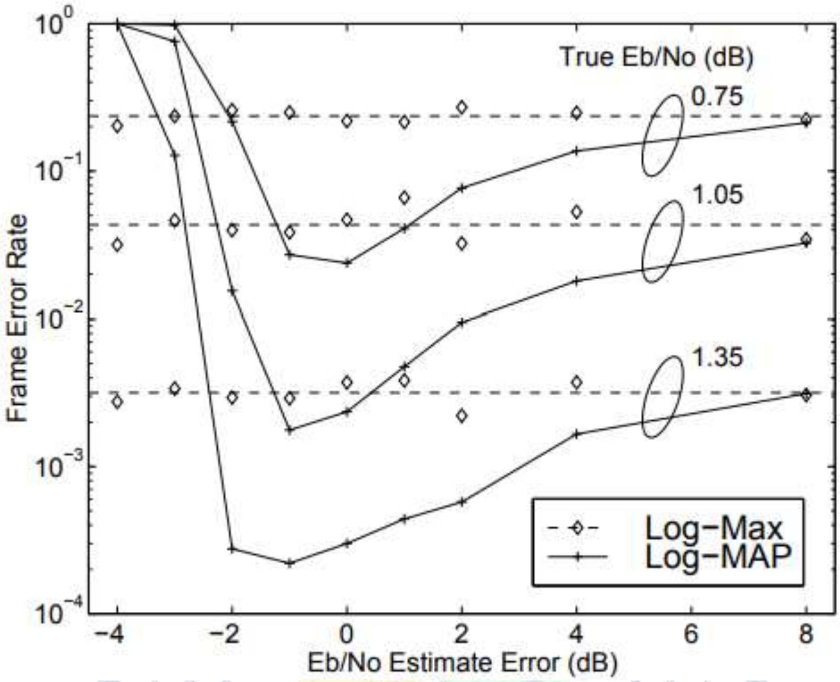

Figure 2: Log-MAP turbo decoder on the AWGN channel

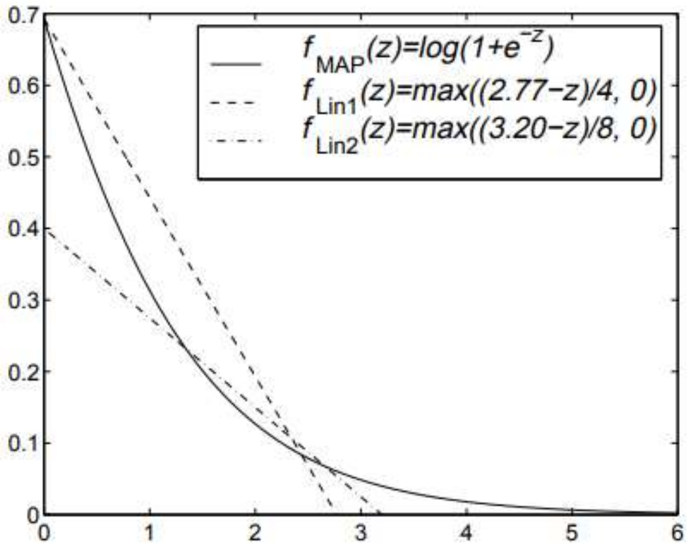

Figure 3: fMAP(z) = log(1+ e^{-z}) and two linear approximations

For the instance of underestimation, we have $\delta < 1$. $f_{MAP}(\hat{L}_c z)$ won't just be bigger than $f_{MAP}(L_c z)$ be that as it may, will likewise be additionally intensified by separating δL_c. since the function $f_{MAP}(\cdot)$ develops quicker when the contention is little, the mistaken amendment term is excessively accentuated. This adds huge measure of commotion to the delicate data and prompts an enormous loss of execution as appeared towards the left-hand side of Fig.2.

3. LOG MAP FIXED-POINT MODEL

We consider a q-bit uniform quantization front end for the fixed-point turbo decoder as illustrated in Fig. 4. The quantizer is modeled as scaling the got signal by $G = \frac{2^{q-1}}{MA}$ and afterward adjusting the outcome to the closest ostensible whole number from the set $[-2^{q-1}, 2^{q-1} - 1]$. . Successfully, the got signals are cut at M times their normal sufficiency and are quantized by the progression size of $\frac{MA}{2^{q-1}}$. For a given number of bits q, the boundary M would thus be able to be utilized to control how these bits are dispensed for dynamic range (cutting) and accuracy (step size).

The quantity of bits q utilized for delicate worth portrayal is a significant plan boundary for fixedpoint turbo decoders. It influences the equipment multifaceted nature for executing the algorithm and is additionally straightforwardly connected to the complete cushion size required for putting away the delicate qualities. Under the limitation of not losing huge execution, this number ought to thusly be limited. As for the situation for convolutional codes, be that as it may, the exhibition of turbo codes is unequivocally influenced by the cut-out level multiplier M in any event, for a fixed q. utilizing the basis of amplifying the channel cutoff rates, discover the ideal M to be a nonlinear capacity of the working SNR.

Utilizing a similar strategy as the last Section, the fixed-point delicate consolidating administrator MAX∗(x_1, x_2) should approximate

$$\max(x_1, x_2) + \frac{G}{L_c} f_{\text{MAP}}\left(\frac{L_c}{G}|x_1 - x_2|\right)$$

(9)

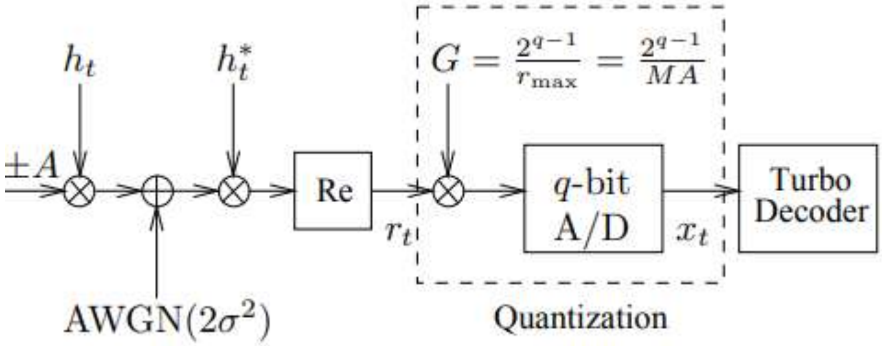

Figure 4: Fixed-point

Note that both G and Lc are elements of the SNR. To help the wide scope of working SNRs in IMT-2000, various look-into tables along these lines should be actualized appropriately. Moreover, these query tables will be gotten to at exceptionally rapid in the turbo decoder. It may thusly be expensive to execute such rapid irregular access memory equipment. Thus, it is standard to actualize just a solitary physical $\text{MAX}^*(x_1, x_2)$ operator. The calculation of $\text{MAX}^*{}_i(x_i)$ would then depend on the successive deterioration of the activity as one appeared in Eq. (2). This methodology makes a long basic way in the framework. Thus, the framework needs to run at a higher clock rate to help a good number of interpreting cycles for high data rate administrations.

4. LOG-LIN ALGORITHMS

To evade the disadvantages referenced in the last area, we propose utilizing basic operational approximations to the remedy work. The objective is to make the MAX∗ module straightforward enough with the end goal that it tends to be cost viably copied to shape a profoundly pipelined usage of $\text{MAX}^*{}_i(x_i)$ utilizing the recursive equal disintegration (Eq. (1)) as one appeared in Fig. 5. For the instance of S contributions to the MAX∗ activity, the equal execution can lessen the length of the basic way down to $\log_2$ S from (S − 1) for the consecutive usage. In mathematical reenactments, the equal disintegration shows marginally preferable execution over the successive methodology, which is by all accounts a consequence of better mathematical properties.

As appeared in Fig. 3, we proposed the accompanying direct estimate to the rectification work:

$$f_{\text{Lin}}(z) = \max\left(\frac{c - z}{D}, 0\right), \quad \textbf{(10)}$$

Where D can be 4 or 8 and c is a boundary that can be streamlined for execution. Following Eq (3), the delicate consolidating activity can be executed in the fixed-point area as

$$\mathrm{MAX}^*_{\mathrm{Lin}}(x_1, x_2)$$
$$= \max(x_1, x_2) + \frac{G}{L_c} \max\left(\frac{c - \frac{L_c}{G}|x_1 - x_2|}{D}, 0\right)$$
$$= \max(x_1, x_2) + \max\left(\frac{C - |x_1 - x_2|}{D}, 0\right), \quad \textbf{(11)}$$

where $C = \lfloor \frac{G}{L_c} c \rfloor$. Since isolating by 4 or 8 is essentially a move of the bit positions by 2 or 3, the MAX∗Lin activity can be actualized with basic logic circuits as one appeared in Fig. 6. The algorithm can be handily adjusted to various working SNRs, on the grounds that the impact of quantization gains and the prerequisite to scale delicate qualities are joined into a solitary control boundary C.

We first think about the exhibition of the proposed algorithm with that of the ideal Log-MAP algorithm under the floating-point model. Four experiments are viewed as based the two edge sizes (N = 640 and N = 2560) and the two channel models (AWGN and free Rayleigh blurring suppositions). For the directly approximated decoder, we set D = 4 and c = 4 ln 2. Fig. 7 plots the edge error rates (FER) of these experiments after 8 translating emphasess utilizing the elective algorithms. The proposed Log-Lin algorithm is found to accomplish indistinguishable execution as the ideal Log-MAP algorithm.

Next we consider a quantization front end that cuts the delicate qualities at M = multiple times the normal sufficiency and utilizations q = 6 bits to speak to the outcomes [12]. For the interior stockpiling of outward data and state measurements, qe bits are utilized by the turbo decoder. In our reproduction, these interior numbers are essentially restricted to inside $[-2^{q_e-1}, 2^{q_e-1} - 1]$. Since the extraneous data will in general develop with the quantity of unraveling emphasis, qe should be bigger than q. In any case, as appeared in Fig. 7, Figure present buoy point Log-Lin and fixed-point Log-Lin decoders on the AWGN and free Rayleigh blurring channels. Eight interpreting emphasess are expected. the FERs of the fixed-point Log-Lin decoder utilizing qe = q + 2 are under 0.05 dB second rate compared to the relating floating-point models for the four experiments considered. This is amazing since the extraneous data could without much of a stretch develop to a significant degree bigger than the info delicate qualities over emphasess. To be sure, the majority of the outward data in our recreation is discovered to be saturated after a couple of cycles. We likewise contrasted the exhibition bends with those utilizing qe = q + 3 and found no recognizable upgrades.

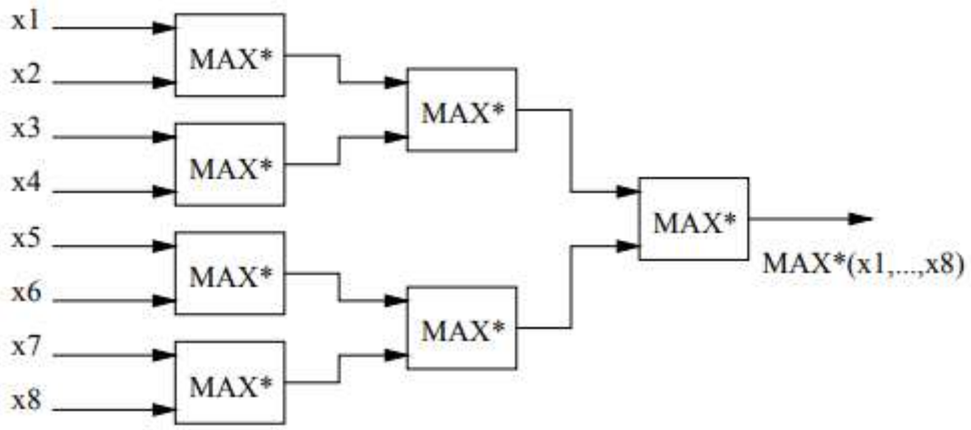

Figure 5: Pipelined implementation of MAX∗(x_1, x_2,..., x8).

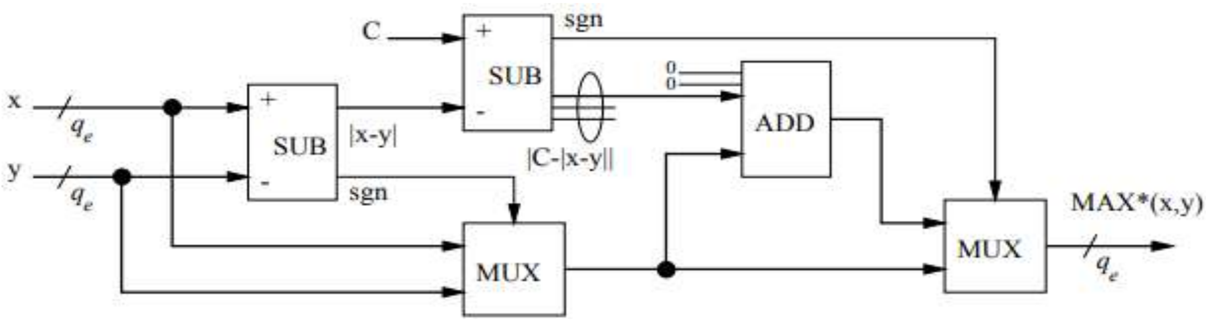

Figure 6: simplified MAX∗ operation with D = 4.

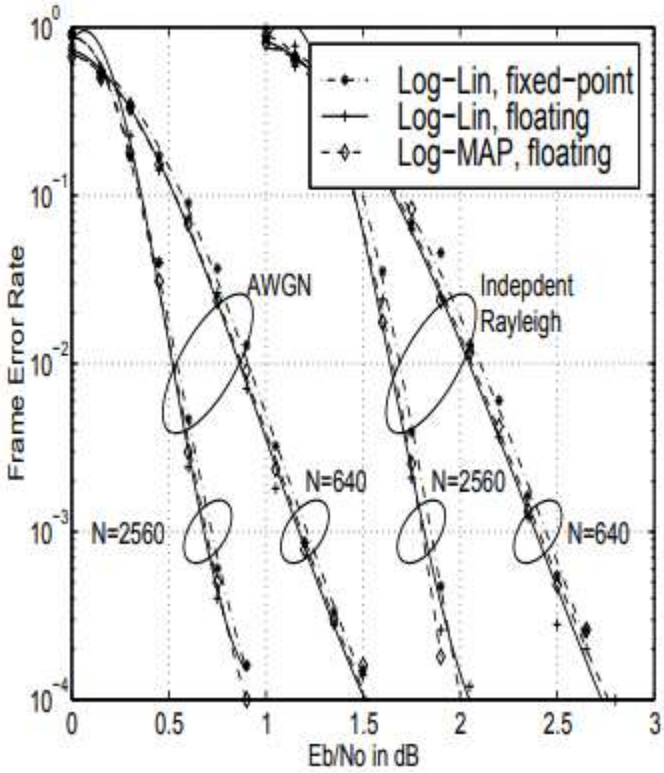

Figure 7: Performance comparison of the floating point Log-MAP

5. CONCLUSIONS

The Log-MAP algorithm is a logarithmic space portrayal of the MAP algorithm that can be communicated as a maximum activity and an added substance revision term. Ignoring the adjustment term, we get the LogMax algorithm. In impedance restricted frameworks like WCDMA, nonetheless, this estimate may bring about a limit loss of as much as 10%, and henceforth we have to incorporate likewise the revision term (or a guess thereof). The disadvantage of including the revision term is that it makes the algorithm touchy towards SNR crisscross. In the analysis we indicated why this is valid and furthermore clarified the motivation behind why, for the Log-MAP algorithm, overestimation of the SNR is more tolearable than underestimation. Since the rectification term is a nonlinear capacity, it is commonly approximated by a lot of look-into tables (one for each SNR area). Be that as it may, for ASIC usage this infers a need of rapid memory. Rather we proposed utilizing a straight estimation (the Log-Lin algorithm) of the revision term. Recreations demonstrate that this Log-Lin algorithm accomplishes a similar exhibition as the Log-MAP algorithm, and that a fixed-point usage of the Log-Lin algorithm with advanced cut-out and quantization to 6 bits, brings about lost execution of under 0.05 dB.

REFERENCES

1. Byoung-Sup Shim, Seok-Jun Choi, Hyoung-Keun Park, Sun-Youb Kim, Yu-Chan Ra. A Study on Performance Evaluation of the Asymmetric Turbo Codes. International Conference on Convergence and Hybrid Information Technology. 2008.

2. Cheng, Jung-Fu & Ottosson, Tony. (2000). Linearly Approximated Log-MAP Algorithms for Turbo Decoding. Proc. 51st IEEE Vehicular Technology Conference (VTC 2000-Spring). 3. 2252 - 2256 vol.3. 10.1109/VETECS.2000.851673.

3. J. Kaza and C. Chakrabarti, “Design and implementation of low-energy turbo decoders,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., Vol. 12, no. 9, pp. 968–977, Sep. 2004.

4. Lin Zhang, Shun-Zheng Yu, "A simplified log-MAP turbo decoder by fitting method", *Advanced Communication Technology 2005 ICACT 2005. The 7th International Conference on*, vol. 2, pp. 854-857, 2005.

5. M.A. Castellon, I.J. Fair, D.G. Elliott, "Fixed-point turbo decoder implementation suitable for embedded applications", *Electrical and Computer Engineering 2005. Canadian Conference on* , pp. 1065-1068, 2005.

6. P.Maniraj Kumar, ET AL (2014) – "Mathematical Implementation of MAX LOG MAP Algorithm for Low Power Applications in Turbo Decoders", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (An ISO 3297: 2007 Certified Organization) Vol. 3, Issue 5, May 2014

7. S.-J. Lee, N. R. Shanbhag, and A. C. Singer, "Area-efficient highthroughput MAP decoder architectures," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 8, pp. 921–933, Aug. 2005

8. Sanjeet Kumar, ET AL (2009) – "Performance of Symmetric and Asymmetric Turbo Code for Distributed Source Coding in a Flat Fading Channel", NCC 2009, January 16-18, IIT Guwahati 123

9. Silvio A.Abrantes, "From BCJR to turbo decoding :MAP algorithms made easier", lecture script, pp.1-30,April 2004.

10. T.-H. T. C.-H. Lin and A.-Y. Wu, "A memory-reduced Log-MAP kernel for turbo decoder," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), 2005, pp. 1032–1035