要 旨

並列 BCJR アルゴリズムによる
高速 Turbo-Code 復号方式

宮崎 康徳

様々な情報システムに依存している近年、情報の信頼性の確保は、現在のみならず、将来を視野に入れても、大容量のデータを扱う情報化社会において、必須である。このような情報の信頼性を確保する技術として、誤り訂正符号は欠くことができない。しかし、伝送される情報は、大容量となるにつれて、誤る可能性は高くなるのが現状である。そこで理論限界に近く、高 BER 特性を持つ誤り訂正符号が必要とされている。現在実用化されている多くの誤り訂正符号は、理論限界に遠く及ばなかった。そこで、1993 年に提案された Turbo-Code は理論限界に最も近いとされ、その潜在的性能の高さは様々な研究で明らかになりつつある。しかし、復号器側では、事後確率を繰り返しにより近似しているため、復号遅延が大きくなり、ソフトウェアによる高速化が困難とされている。

本研究では、このような復号遅延問題を解決するべく、Turbo-Code の並列性について検討し、並列処理能力に優れたデータ駆動型プロセッサによりソフトウェアで高速実現することにより、様々なシステムに柔軟に対応可能で、かつ高速な Turbo-Code 復号処理を実現する。

性能評価の結果、従来の Turbo-Code 復号処理と比較して、本提案手法は処理遅延を削減でき、高速に処理することが可能であることを確認した。

キーワード　Turbo-Code、データ駆動型プロセッサ、DECODER、BCJR アルゴリズム

Abstract

A High Speed Turbo-Decoding System based on Parallel BCJR Algorithm

Yasunori MIYAZAKI

With spreading influence of information systems, guarantee of information reliability is indispensable to information society where a large amount of data is treated. There are error-correcting codes in order to guarantee the information reliability. However, the error rate of the transmiting information becomes higher as data becomes larger. Furthermore, error-correcting codes require the theoretical limit and high BER performance. Many error-correcting codes in practical use are far inferior to theoretical limit.

Turbo-Code suggested in 1993, which is most near to the theoretical limit, has been studied in various research to show the high level of the potential performance. However, in the decoder side, decoding delay becomes large and speeding up by software is difficult for posterior probability approximates by iteration.

In this research, in order to solve such a decoding delay problem, Turbo-Code decoding processing was implemented to be able to respond flexibly in various systems and high-speed processing by data driven processor which realizes high-parallel processing.

As the result of performance evaluation, this proposed method is proved to reduce processing delay and to realize high-speed processing.

key words Turbo-Code, Data-Driven Processor, DECODER, BCJR Algorithm