Regular paper http://dx.doi.org/10.13067/JKIECS.2015.10.11.1239

R=1/2 Self-Doubly 조직 직교 길쌈부호를 찾는 효율적인 최적 스팬 알고리듬

아타뱁 도너* · 서희종**

An Efficient Algorithm for finding Optimal Spans to determine R=1/2 Rate Systematic Convolutional Self-Doubly Orthogonal Codes

Atabaev Doniyor* · Hee-Jong Suh**

요 약

본 논문에서는 길쌈 Doubly 직교 부호의 최적 스팬을 찾기 위한 새로운 방법(Convolutional Self-Doubly Orthogonal, CDO)을 제안한다. 이 새로운 방법은 병렬 Implicitly-Exhaustive 탐색방법을 사용하는데, 이 방법으로 R =1/2 CDO 코드에 대한 최적의 스팬을 찾기 위해서 계산시간을 감소시키는 방법으로 동적 검색 공간 감소 방법을 적용했다. 제안된 알고리듬을 모의실험한 결과 기존의 방법에 비해서 계산시간이 감소되었고, 오류 정정 성능이 향상되었음을 확인하였다.

ABSTRACT

In this paper, a new method for finding optimal and short span in Convolutional Self-Doubly Orthogonal(CDO) codes are proposed. This new algorithm based on Parallel Implicitly-Exhaustive search, where we applied dynamic search space reduction methods in order to reduce computational time for finding Optimal Span for R=1/2 rate CDO codes. The simulation results shows that speedup and error correction performance of the new algorithm is better.

키워드

CDO Codes, Implicitly-Exhaustive Search Algorithm, Dynamic Search Space Reduction
CDO 코드, Implicitly-Exhaustive 알고리듬, 동적 검색 공간 감소

Ⅰ. Introduction

In recent years, the rise of mobile devices has been accompanied by an ever increasing need for reliable high-bandwidth wireless communications. Convolutional Self-Doubly Orthogonal(: CDO) codes used to increase efficiency and build high-performance, low latency codes[1] for data transmission in communication channels. The performance of CDO Codes can be improved by optimizing search for optimal/short span(constraint length) using several methods.

Finding optimal span of the code is very important to search over rate $R=1/2$ systematic

* 전남대학교 전자통신공학과(hj-suh@daum.net)
** 교신저자 : 전남대학교 전자통신공학과
· 접 수 일 : 2015. 10. 02
· 수정완료일 : 2015. 11. 13
· 게재확정일 : 2015. 11. 23
· Received : Oct. 02, 2015, Revised : Nov. 13, 2015, Accepted : Nov. 23, 2015
· Corresponding Author : Hee-jong Suh
Dept. of Electronic Communication Engineering, Chonnam National University,
Email : hj-suh@daum.net

CDO codes as the error-correcting capability of these codes depends on the span and dimension J of the vector generator [2]. Pseudo-random and exhaustive search algorithms have been used to find codes of shorter span. The main problem is computational time required to find shorter-span codes is become very high as J increases. This problem may be viewed as second-order Golomb rulers which is believed to have an NP-hard complexity[3]. Implementing a strong parallel exhaustive-search algorithm with enabling dynamic search space reduction methods can efficiently leverage hundreds of processor cores. Speedups are achieved through enhancements in the deterministic search-space reduction and use a novel data structure for enabling data reuse(rejection) and finding optimal path in our search tree.

By using these methods we can obtain new optimal-span CDO codes with significantly shorter spans for a given number of J connections. Preliminary work on reducing the computational time required by the reference exhaustive search algorithm consisted in adapting with several efficient methods to perform a basic parallel search [4].

In this paper, we present a dynamic search-space reduction methods, that result in a very efficient parallel and implicitly exhaustive implementation that greatly reduces the computational time required for finding optimal-span CDO codes. The paper is organized as follows: in Section II, Efficient methods is defined to minimize computational time for search optimal span of CDO codes. Efficient parallel implicitly-exhaustive search algorithm is described in Section III. The Experiment and Results shown in Section IV. Finally, Conclusion described In Section V.

II. Efficient Methods for Finding Optimal Span of CDO Code and Determing Algorithm for These Codes.

Recent code-searching algorithms can be divided into two categories: pseudo-random and exhaustive. Exhaustive search algorithms guarantee that the optimal span for a given J number of connections is found, provided that a sufficiently large initial the smallest known span is used. This method testing all the potentially valid codes in the search space. The main disadvantage of this method is when J becomes larger computational time required for obtain optimal span increases rapidly. Pseudo-random search algorithms are based on the use of a pseudo-random rejection criterion that can be adjusted in order to shorten the spans of the codes obtained[5-7].

There are several efficient methods with which we can improve exhaustive search algorithm to achieve speedup for finding optimal Span Paths. Using parallel dynamic search-space reduction technique with enabling data reuse (rejection) method are substantially reduces the size of the search space without compromising the exhaustive nature of the search. Finding an Optimal Span in valid codes can be calculated with first-order and second-order differences using these equations [7].

$$N_S^J = \frac{J(J-1)}{2} \tag{1}$$

$$N_D^J = \frac{J(J^3-2J^2+3J-2)}{8} \tag{2}$$

where N_s and N_D are first order and second order differences respectively.

Lower bound for the span of CDO can be expressed as a function of J and using equations (1) and (2) as follows

$$\alpha_J \geq a_J^* = \left\lceil \frac{N_S^J + N_D^J}{2} \right\rceil \tag{3}$$

where α_J The Span of CDO

This observation led to the development of the efficient parallel and implicitly-exhaustive search algorithm used for finding the new CDO codes.

By partitioning the search space into independent sub-trees that are only accessed by one thread at a time, threads can do most of the processing within their private workspace, foregoing the need of complex resource-sharing mechanisms and thus reducing the overall synchronization overhead. Sharing this information instantly benefits all thread, thus significantly decreasing the overall computation time by allowing the search space to converge to a smaller search-tree in less time. When a thread has finished processing a job, another job is assigned to it until no more sub-trees are available, at which point the exploration of the search-tree is complete. Data reuse will reduces the computational time through rejection of paths which is not considered. It would compute all the first and second order differences of a code candidate in order to evaluate whether it is a valid CDO code or not.

III. Determing Algorithm for Finding Efficient Codes

In order to find new optimal-span codes for larger values of J, an attempt at improving the reference exhaustive-search algorithm is described in [5]: the computation time is reduced by means of a very basic simultaneous exploration of different regions of the search space.

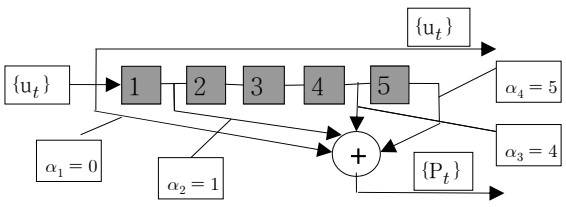

Fig. 1 Example of systematic CDO code encoder: $R=1/2,\ J=4,\ M=5,\ \Omega=\{0,1,4,5\}$.

Systematic CDO codes means that given encoder are in systematic form and since we proposed algorithm with 1/2 rate codes we need only one encoder generator. Figure 1 represent Systematic 1/2 Rate encoder. It does have one input $\{u_t\}$ and two outputs $\{u_t\}$ and $\{P_t\}$.Encoder with one input and two outputs called half Rate ($R=1/2$). One of the output sequence is equal to input sequence where we confirm that this encoder is in systematic form. Also α represents an integer and the number J of elements in Ω is equal to the number of generator connections of the code and is called of the "order" of the CDO code. An "optimal" CDO code of a give order J is defined as a CDO code whose span M is the smallest span that exists for that order.

The algorithm which we proposed performs the search for CDO codes using a tree-like structure (see Fig. 2 for J=4). The root node always has a value of "0". The rest of the tree is composed of nodes which must have a value greater than their parent node and their sibling nodes on the left. Sibling nodes are nodes with same parent. The tree depth represent the number connections J, while the values of these nodes represent $\Omega=\{\alpha_1,\alpha_2,\alpha_J\}$ the set of positive integers forming the code. All nodes at depth J are considered leaf nodes. A valid path in this search-tree starts at the root node and ends at a leaf-node that has a value not larger than M_{curr}, the smallest known span value.

The search tree depends on current M_{curr}, and on J. The total numbers of possible paths N_L is

$$N_L=\left(\frac{M_{curr}}{J-1}\right)=\frac{(M_{curr})!}{(J-1)!(M_{curr}-J+1)!} \quad (4)$$

Since the size of the search-space is a combination of "M_{curr} choose $J-1$ (see above N_L), linear improvements in performance are not sufficient to provide new results. A node whose value is larger than M_{curr} minus the search-tree depth remaining from that node to a leaf-node, must

have descendants such that their leaf-nodes have a value larger than M_{curr}, the current best known span. Thus, these nodes can be safely discarded since they cannot lead to codes with a shorter span than the shortest known span for that order J. Please See table 1 where [0,1,4,5] path with the best for encoder in Figure 1 for $J=4$, $M=5$.

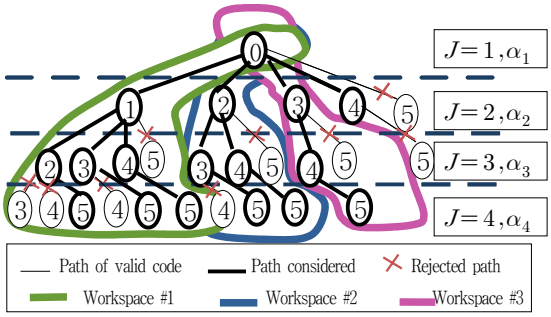

Fig. 2 CDO search tree with parallel implicitly exhaustive search algorithm divedes into set of sub-trees (workspace) in the case of order J=4.

Table 1. Paths in subtree which is considered in Figure 2 for $J=4$ and $M=5$

Subtree	Path considered
Workspace #1	[0,1,2,5];[0,1,3,5]; **[0,1,4,5]**
Workspace #2	[0,2,3,5];[0,2,4,5]
Workspace #3	[0,3,4,5]

The process of adding, testing for validity and discarding a node is repeated for all sibling nodes on valid path until either the added node forms a valid CDO codes or no more such siblings exist(in which case the next parent on a valid path would be evaluated).See table 1 where we can find all path which are considered in the case of M=5 and J=4. If current valid code has J connections and it's span value is below the best known span,α_{max} is updated and thus some paths are invalidated. The optimal CDO codes will be the codes with a Span equal to α_{max}.

Ⅳ. Experiment and Results

We employ a novel data structure for enabling data reuse and incremental computations and a parallel dynamic search-space reduction technique that substantially reduces the size of the search-space without compromising the exhaustive nature of the search. The efficient parallel implicitly-exhaustive search algorithm reduces the computational time required for searching optimal CDO codes is by using more aggressive search-tree pruning techniques to further reduce the number of branches that are explored. Indeed, children nodes of nodes that have been discarded do not need to be traversed or validated, thus allowing for substantial computational savings. Sub-tree base-nodes may be mapped to any node having depth and increase in this depth reduces the size of their corresponding sub-tree, at the expense of increasing the total number of tasks to compute. Therefore Data reuse method reject the paths which is not considered in our sub tree. In result threads can be computed faster and operate concurrently for a longer time.

Matlab simulation graphs shows that with our new method, best results are obtained. Figure 3 shows speed up with all methods which is used for finding optimal span in this paper. Implicitly-Parallel search new dynamic search space reduction and enabling data reuse are shows the best result.

If we compare BER rate this paper CDO algorithm and reference CDO algorithm for CDO Codes we can see that new method has better performance and efficiency over reference algorithm (Figure 4). New CDO codes algorithm from an engineering point of view offer a much reduced latency and thus can be far more advantageous.

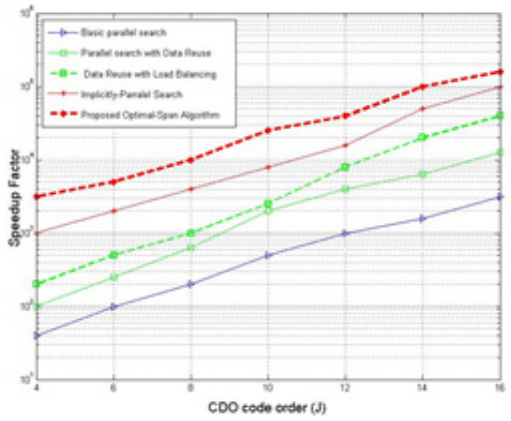

Fig. 3 Scaling of CDO code for finding optimal Span speedup as a function J, For $J \in [4;16]$.

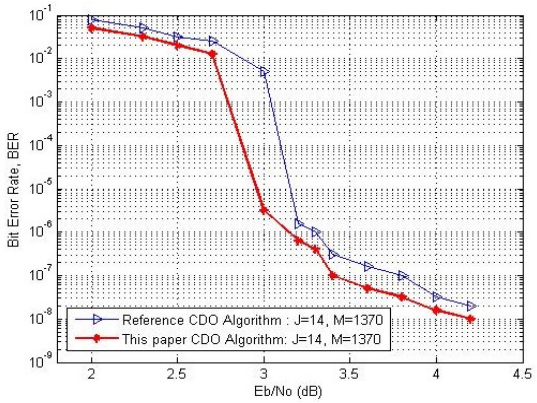

Fig. 4 Rate 1/2 systematic CDO code error correction performance at $E_b/N_0 \in [2.0; 4.3]$ (dB)

V. Conclusions

We presented a parallel, high-performance, efficient implicitly-exhaustive search algorithm implementation for finding optimal span paths and determining the CDO codes with short spans. Efficiently parallel implicitly-exhaustive search achieved by using a novel data structure for enabling data-reuse and incremental computation of differences, and a parallel dynamic search-space reduction technique that substantially reduces the size of the search-space without compromising the exhaustive nature of the search. The novel algorithm computes the new differences generated by the next code candidate and reuses the differences that were already computed for the previous code validation. Furthermore, the algorithm scales well as the number of microprocessors used increases. We have characterized the dramatic speedup achieved with the novel search algorithm over previously published algorithms. As a result, we were able to find and verify optimal codes for $J \in [4;16]$..

References

[1] C. Cardinal, D. Haccoun, and F. Gagnon, "Iterative threshold decoding without interleaving for convolutional self-doubly orthogonal codes," *IEEE Trans. Communications*, vol. 51, no. 8, 2003, pp. 1274-1282.

[2] H. Shin, "Development of constant current SMPS for LED Lighting," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 10, no. 1, 2015, pp. 111-116.

[3] Y. Jeong, "A study on control of generators based on SMPS," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 7, no. 1, 2012, pp. 107-115.

[4] G. Kowarzyk, N. Belanger, D. Haccoun, and Y. Savaria, "Efficient Parallel Search Algorithm for Determining Optimal $R=1/2$ Systematic Convolutional Self-Doubly Orthogonal Codes," *IEEE Trans. Comm.*, vol. 61, no. 3, 2013, pp. 865 - 876.

[5] H. Shin, "Design of LED Driving SMPS for Large Traffic Signal Lamp," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 4, no. 2, 2009, pp. 123-129.

[6] W. Wu, "New Convolutional Codes-Part I," *IEEE Trans. Communications*, vol. 23, no. 9, Sept. 1975, pp. 942-956.

[7] C. Cardinal, E. Roy, and D. Haccoun, "Simplified Convolutional Self-Doubly Orthogonal Codes: Search Algorithms and

Codes Determination," *IEEE Trans. Communications*, vol. 57, no. 6, 2009, pp. 1674-1682.
[8] C. Amin, N. Menezes, K. Killpack, and F. Dartu, "Statistical static timing analysis: how simple can we get?," *Design Automation Conf. 2005. Proc. 42nd*, Anaheim, CA, USA, May 2005, pp.652-657.
[9] X. Bai, P. Patel, and X. Zhang, "A New Statistical Setup and Hold Time Definition,"*IC Design & Technology (ICICDT), 2012 IEEE Int. Conf.* Austin, TX, May 2012, pp.1-4.
[10] B. Rebaud, M. Belleville, and C. Bernard, "Setup and Hold Timing Violations Induced by Process Variations, in a Digital Multiplier," *Symp. on VLSI, 2008. ISVLSI '08. IEEE Computer Society Annual,* Montpellier, France, 2008, pp. 316-321.

저자 소개

아타뱁 도녀
(Atabaev Doniyor)

2010년 타슈켄트 주립 기술 대학 (Radioengineering, 전자 및 자동화 교수)
2015년 전남대학교 대학원 전자통신공학과 석사 졸업
2015년 전남대학교 대학원 전자통신공학과 박사과정 입학
※ 관심분야 : 코딩, 컴퓨터 네트워크

서희종(Hee-Jong Suh)

1975년 한국항공대학 학사
1996년 중앙대학교 대학원 전자공학과 박사
2006년3월~전남대학교 전자통신공학과 교수
※ 관심분야 : 그래프이론, 컴퓨터네트워크, 컴퓨터통신