논문 13-38A-08-04

한국통신학회논문지 '13-08 Vol.38A No.08
http://dx.doi.org/10.7840/kics.2013.38A.8.644

DC-억압 변조를 위한 GS 코딩의 최악 성능 평가 MaxMin 모형

박 태 형•, 이 재 진°

A MaxMin Model for the Worst Case Performance Evaluation of GS Coding for DC-free Modulation

Taehyung Park•, Jaejin Lee°

요 약

광기록 정보저장장치에서 인코딩된 시퀀스의 DC-억압을 위해 Guided Scrambling 기법이 널리 사용된다. 후보 코드시퀀스 중 최적의 DC-억압 코드를 선택하기 위해 digital sum value (DSV)의 함수로 정의된 기준을 사용한다. 이 중 minimum DSV (MDSV), minimum squared weight (MSW), minimum threshold overrun (MTO) 등이 널리 사용된다. 본 연구에서는 MDSV, MSW, MTO 기준을 채택하는 GS 코딩 알고리즘과 동등한 정수계획법 모형을 제안한다. 개발된 MDSV 정수계획법 모형을 MaxMin 형태의 모형으로 확장하여 스크램블링 다항식과 제어비트에 따른 MDSV GS 코딩의 최악 성능을 평가할 수 있는 모형을 개발하였다. 모의실험에서는 다수의 스크램블링 다항식 및 제어비트 조합에 대하여 MDSV 최악 성능을 계산하였다.

Key Words : DC-free Coding, GS Coding, Digital Sum Value, Integer Programming, Worst Case Bound

ABSTRACT

For effective DC-free coding in the optical storage systems, the Guided Scrambling algorithm is widely used. To reduce digital discrepancy of the coded sequence, functions of digital sum value (DSV) are used as criteria to choose the best candidate. Among these criteria, the minimum digital sum value (MDSV), minium squared weight (MSW), and minimum threshold overrun (MTO) are popular methods for effective DC-suppression. In this paper, we formulate integer programming models that are equivalent to MDSV, MSW, and MTO GS coding. Incorporating the MDSV integer programming model in MaxMin setting, we develop an integer programming model that computes the worst case MDSV bound given scrambling polynomial and control bit size. In the simulation, we compared the worst case MDSV bound for different scrambling polynomial and control bit sizes. We find that careful selection of scrambling polynomial and control bit size are important factor to guarantee the worst case MDSV performance.

Ⅰ. 서 론

광기록 정보저장장치에서는 입력 데이터를 변조부호를 사용하여 직류-제약(DC-free)을 갖는 시퀀스로 변환한다. 광기록 정보저장장치에서는 서보제어 신호와 데이터 신호간의 상호간섭이 발생하지 않도록 데이터 신호는 특정한 주파수 대역에 있어야 한다. 이렇게 변조하는 코드를 DC-억압 (DC-free) 코드라

• 주저자 겸 교신저자 : 숭실대학교 산업정보시스템공학과, tpark@ssu.ac.kr, 정회원
° 숭실대학교 정보통신전자공학부 정보저장및통신 연구실, zlee@ssu.ac.kr, 종신회원
논문번호 : KICS2013-07-296, 접수일자 : 2013년 7월 18일, 최종논문접수일자 : 2013년 7월 23일

한다. DC-억압을 위해서 제어비트를 소스데이터에 추가한다. 멀티모드 (multi-mode) 코딩에서는 각 소스워드에 L개의 후보 코드워드를 1-to-L 함수로 정의하고, 후보 코드워드 중에서 성능이 가장 우수한 코드를 선택하여 전송(기록)한다. 가이디드 스크램블링 (Guided Scrambling, GS) 코딩은 소스워드에 p 개의 제어비트를 붙인 후, 지정된 다항식의 시프트 레지스터를 이용하여 $L=2^p$ 개의 후보코드를 생성하고, 후보코드 시퀀스의 저주파수 영역이 최소가 되는 코드를 선택한다. 멀티모드 코딩은 GS 코딩 이외에 dc-free coset 코딩, Hadamard 변환 코딩이 있다[1-3].

인코딩된 시퀀스의 DC 기여도는 시퀀스의 시작부터 마지막까지 누적된 신호값을 나타내는 digtal sum value (DSV) 혹은 running digital sum (RDS) 계산을 통해 측정한다. DSV에 기반한 코드 선택 기준으로는 DSV의 절대값의 최소를 찾는 Minimum DSV (MDSV), DSV의 분산을 최소화하는 Minimum Squared Weight (MSW), 주어진 역치 (threshold) τ를 초과하는 회수를 최소화하는 Minimum Threshold Overrun (MTO)가 있다[4,5].

MDVS 기준을 적용할 때, GS 코딩의 성능평가에는 Markov Chain을 이용한 분석이 소개되었다. Markov Chain을 이용한 분석에서는 DSV 값들 사이의 전환확률을 정의하고 각 DSV 상태별로 수렴확률을 계산하였다. 하지만 이 분석에서는 스크램블링 다항식이나 제어비트에 따른 GS 코딩의 성능은 분석할 수 없었다[6]. 본 논문에서는 스크램블링 다항식과 제어비트가 정해진 경우, MDSV 값의 최악성능을 평가하는 모형을 개발한다.

이 논문의 구성은 다음과 같다. 2장에서는 DC-억압을 위한 MDSV, MDW, MDT 선택기준을 사용하는 GS 코딩절차를 최적화 모형으로 수식화한다. 3장에서는 개발된 0-1 정수계획법 모형을 Max-Min 문제로 확장한 모형을 이용하여 GS 코딩의 최악 성능을 평가하는 모형을 소개한다. 4장에서는 각 스크램블 다항식과 제어비트 값에 따라 최악 성능이 어떻게 변화하는지 모의실험을 통하여 비교한다.

Ⅱ. 정수계획법 모형

GS코딩은 n비트 소스 시퀀스에 p비트 제어비트를 앞에 붙여서 확장된 $N:=n+p$ 비트 시퀀스를 self-synchronizing 다항식

$$c_k = b_k \oplus \sum_{i \in A} a_i c_{k-i} \tag{1}$$

를 이용하여 스크램블한다. 여기에서 $a_i \in \{0,1\}$은 계수이고, 모든 연산은 modulo-2 덧셈이다. 이렇게 전체 $L=2^p$ 후보 시퀀스를 생성한다. DSV s_k는 다음과 같이 정의한다.

$$s_k = s_{k-1} + 2c_k - 1,\ k = 1, \ldots, N \tag{2}$$

MDSV 기준을 적용하는 경우, 시퀀스 $s=(s_1,\ldots,s_N)$의 l_∞-norm 인 $max\{|s_k|\}$가 최소가 되는 후보를 선택한다. MSW의 경우는 l_2-norm 인 $\|s\|_2^2 = \sum_{k=1}^{N} s_k^2$가 최소가 되는 후보를 선택한다. MTO의 경우는 $s_k > \tau$인 회수가 최소가 되는 후보를 선택한다. 이 외에 s_k가 $+$에서 $-$로 천이가 가장 많은 후보를 선택하기도 한다. 아래에서는 다항식 (1)을 이용하여 2^p 후보시퀀스를 생성한 후 MDSV, MSW, MTO 기준으로 후보코드를 선택하는 문제를 정수계획법 모형으로 수식화한다. 아래에서는 스크램블러 다항식 $c_k = b_k \oplus c_{k-2}$을 예제로 사용하여 모형을 개발한다.

MDSV를 최소화는 정수계획법 모형은 다음과 같다. p 제어비트와 n 소스비트를 갖는 경우, 스크램블러를 통과한 코드시퀀스 c_i와 DSV s_i는 다음 식을 만족한다. 아래에서 $d_1,\ldots,d_p$는 제어비트, $\overline{b_1},\ldots,\overline{b_n}$는 소스비트, c_i는 스크램블된 코드시퀀스, s_i는 DSV를 표시한다. 스크램블러의 초기값은 $c_0=0, c_{-1}=0, s_0=0$로 가정한다.

$$\begin{aligned} & c_i \le 2 - d_i - c_{i-2},\ \forall i \\ & c_i \le d_i + c_{i-2},\ \forall i \\ & c_i \ge c_{i-2} - d_i,\ \forall i \\ & c_i \ge d_i - c_{i-2},\ \forall i \\ & s_i = s_{i-1} + 2c_i - 1,\ \forall i \\ & d_{p+1} = \overline{b_1}, \ldots, d_{p+n} = \overline{b_n}, \\ & d_1, \ldots, d_p \in \{0,1\} \end{aligned} \tag{3}$$

식 (3)에서 처음 4개의 부등식은 modulo-2 스크램블링 연산 $c_k = b_k \oplus c_{k-2}$을 나타낸다.

목적함수로 $\|s\|_\infty = \max\{|s_i|\}$를 최소화하는 MDSV 문제의 경우 식 (3)의 제약식에 다음과 같은 목적함수와 제약식을 추가한다.

$$
\begin{aligned}
f_{MDSV}(\boldsymbol{d},\overline{\boldsymbol{b}}) = Min \quad & \gamma \\
s.t. \quad & s_i \le \gamma, \ \forall i \\
& -s_i \le \gamma, \ \forall i
\end{aligned} \tag{4}
$$

목적함수로 DSV의 분산, $\|s\|_2^2 = \sum_{i=1}^{N} s_i^2$를 최소화하는 MSW의 목적함수는 다음과 같다.

$$
f_{MSW}(\boldsymbol{d},\overline{\boldsymbol{b}}) = Min \sum_i s_i^2 \tag{5}
$$

벡터 norm의 성질에 의해 다음 식이 성립한다.

$$
\|s\|_\infty \le \|s\|_2 \le \|s\|_1 \tag{6}
$$

따라서 $f_{MDSV} \le \sqrt{f_{MSW}}$가 성립한다. 역치 τ를 초과하는 DSV $|s_i|$의 회수를 최소화하는 MTO 목적식을 갖는 경우, 목적함수와 제약식을 다음과 같이 수정한다. 아래에서 M은 큰 값을 갖는 양의 정수이다.

$$
\begin{aligned}
f_{MTO}(\boldsymbol{d},\overline{\boldsymbol{b}}) = Min \quad & \sum_i x_i \\
s.t. \quad & s_i \le Mx_i + \tau, \ \forall i \\
& -s_i \le Mx_i + \tau, \ \forall i \\
& x_i \in \{0,1\}, \ \forall \mathrm{i}
\end{aligned} \tag{7}
$$

Ⅲ. MaxMin 모형

2장에서 개발한 모형을 사용하여 최악 MDSV z_{worst}를 다음과 같이 계산할 수 있다.

$$
z_{worst} = \max_{b_1,\dots,b_n}\{\min_{d_1,\dots,d_p} f(\boldsymbol{d},\boldsymbol{b})\} \tag{8}
$$

식 (8)에서 괄호안의 최소화 문제는 다음과 같이 식으로 대체할 수 있다. $[0,1]^p$의 z번째 꼭지점 (vertex)의 좌표를 p-vector $\boldsymbol{a_z} = (a_i^z)$로 표시하면, 제어비트 벡터 $\boldsymbol{d} = (d_1, \dots, d_p)$는 $\boldsymbol{a_z}$중의 한 값을 갖는다. 제어비트 $\boldsymbol{d} = \boldsymbol{a_z}$ 일 때, DSV의 최대값을 θ_z, DSV의 값을 s_i^z로 표시하면,

$$
\begin{aligned}
\theta_z &\ge s_i^z, \ \forall i \\
\theta_z &\ge -s_i^z, \ \forall i
\end{aligned} \tag{9}
$$

가 되고, 이 때

$$
\min_{d_1,\dots,d_p} f(\boldsymbol{d},\boldsymbol{b}) = \min_z \theta_z \tag{10}
$$

가 된다. 변수 $\gamma = \min_z \theta_z$를 다음 식을 만족한다.

$$
\begin{aligned}
& \gamma \le \theta_z, \ \forall z \\
& \theta_z - \gamma \le M(1-y_z), \ \forall z \\
& -\theta_z + \gamma \le M(1-y_z), \ \forall z \\
& \sum_z y_z = 1 \\
& y_z \in \{0,1\}, \ \forall \mathrm{z}
\end{aligned} \tag{11}
$$

식 (9)도 마찬가지로 수식으로 대체할 수 있다. 이와 같이 (8)의 내부 최소화문제를 정수변수들을 추가로 도입하여 식으로 표현하면 식 (8)은 일반적인 혼합정수계획법 모형으로 변환된다.

$$
\begin{aligned}
Max \quad & \gamma \\
s.t. \quad & \sum_z y_z = 1 \\
& d_i^z = a_i^z, \ i = 1,\dots,p, \ \forall z \\
& b_i = d_{i+p}^z, \ i = 1,\dots,n, \ \forall z \\
& c_i^z = d_i^z \diamond c_{i-2}^z, \ \forall i,z \\
& s_i^z = s_{i-1}^z + 2c_i^z - 1, \ \forall i,z \\
& \theta_z \ge s_i^z, \ \theta_z \ge -s_i^z, \ \forall i,z \\
& \pm s_i^z \mp \theta_z \le M(1-x_i^z), \ \forall i,z \\
& \mp s_i^z \pm \theta_z \le M(1-x_{i+n}^z), \ \forall i,z \\
& \sum_i x_i^z = 1, \ \forall z \\
& \pm \theta_z \mp \gamma \le M(1-y_z), \ \forall z \\
& \theta_z \le \gamma, \ \forall z \\
& x_i^z, y_z, b_i \in 0,1, \ \forall \mathrm{i,z}
\end{aligned} \tag{12}
$$

여기에서 초기값 $c_0^z = c_{-1}^z = 0$이다. 식 (12)에

서 $c_i^z = d_i^z \diamond c_{i-2}^z$는 식 (3)의 처음 4가지 부등식을 의미한다. (12)를 풀 때는 4개의 부등식으로 대체해야 한다. 식 (12)에서 변수 x_i^z는 $|s_i^z|$의 최소값의 위치를 표시하는 0-1 변수이고, y_z는 $\max_z \min_i |s_i^z|$를 표시하는 변수이다. 문제 (12)의 크기는 제약식이 3,892 개, 변수가 791 개이고, 이 중 278 개가 0-1 변수인 대규모 정수계획법 문제이다.

Ⅳ. 계산 결과

모형 (4), (5), (7), (12)는 Algebraic Modeling Language인 AMPL[7]과 ILOG의 CPLEX[8]를 사용하여 프로그래밍 하였다. 길이 3의 제어비트를 사용하고 소스데이터 길이가 30인 경우, 후보코드워드는 8개이고 코드율은 (30x8)/(30x8+3) 으로 1.24%의 redundancy를 갖는다. 표 1은 MDSV 알고리즘으로 100개의 임의의 문제를 계산하였을 때 제어비트에 따른 목적값과 $\|s\|_2^2$ 값의 평균을 기록하였다.

표 2는 MTO의 역치 τ 값을 1에서 5까지 변화시키면서 MTO 알고리즘으로 100개의 임의의 문제에 대한 목적함수값의 평균 및 최적해에서의 $\|s\|_2^2$, $\|s\|_\infty$ 값의 평균을 기록하였다. 표 2에서 제어비트는 3으로 고정하였다.

표 1과 표 2를 비교할 때, 제어비트가 4 이상인 경우, MDSV로 계산한 경우, 분산 $\|s\|_2^2$이 작게 나왔다. 또한 MTO의 경우, 역치 τ 값이 1과 5인 경우, $\|s\|_2^2$를 줄이는 선택을 하지 않았음을 알 수 있다. MTO의 $\|s\|_\infty$를 보면, MDSV 기준으로 적용할 때 보다 큰 값을 가짐을 알 수 있다.

표 1. 제어비트에 따른 MDSV 평균
Table 1. MDSV averages for different control bit sizes

# control bit	MDSV	$\|s\|_2^2$
3	4.35	174.48
4	3.89	138.24
5	3.79	133.64
6	3.75	137.19

표 2. 역치값에 따른 MTO의 평균
Table 2. MTO averages for different threshold values.

τ	MTO	$\|s\|_2^2$	$\|s\|_\infty$
1	14.4	183.04	4.71
2	7.6	169.76	4.55
3	3.51	166.72	4.42
4	1.47	168	4.42
5	0.53	187.04	4.57

표 3은 스크램블러 다항식 $c_k = b_k \oplus c_{k-2}$ 와 $c_k = b_k \oplus c_{k-2} \oplus c_{k-11}$의 MDSV 최악 성능을 비교하였다. 표 3에 의하면, 두 스크램블러 다항식간에 MDSV 최악 성능에 큰 차이가 있음을 알 수 있다. 제어비트가 3인 경우, 두 다항식간에 최악 MDSV 값은 각각 15와 9이다. 표 3에서 'z_{worst}' 열에는 모형 (12)를 이용하여 계산한 최악 MDSV 값이 기록되어 있다. 표 3의 'min', 'max', 'average', 'avg var'는 1,000개의 임의의 문제를 생성하여 MDSV를 계산한 후, 1,000 문제 중

표 3. 스크램블러 다항식에 따른 MDSV 최악 성능의 비교.
Table 3. Comparison of MDSV worst case bound for different scrambling polynomial.

	$c_k = b_k \oplus c_{k-2}$					$c_k = b_k \oplus c_{k-2} \oplus c_{k-11}$				
#control bit	min	z_{worst}	max	average	avg var	min	z_{worst}	max	average	avg var
3	2	15	11	4.379	182.099	2	9	7	3.508	107.58
4	2	14	10	3.964	148.806	2	6*	5	3.137	85.572
5	2	13	9	3.852	139.403	1	5*	5	2.85	71.956
6	1	12	8	3.75	135.176	1	4*	4	2.654	66.222

최소 MDSV, 최대 MDSV, 평균, 그리고 $\|s\|_2^2$의 평균을 각각 기록하였다.

제어비트가 3일 때 첫 번째 다항식의 경우, 1,000개중 MDSV의 최대값은 11이었고, 두 번째 다항식의 경우는 최대값이 7이었다. 또한 $\|s\|_2^2$의 값을 보면, 두 번째 다항식을 사용할 때 $\|s\|_2^2$의 값이 작아짐을 알 수 있다. 표 3에서 두 번째 다항식의 'z_{worst}' 중 *가 붙어있는 항은 연산 CPU 시간한계인 2시간을 초과한 경우에 해당하며, 최적값의 근사치가 기록되었다. 두 다항식간에 11단계 다항식을 사용한 경우에 MDSV 최악 성능이 55% 감소되었고, DSV의 분산은 평균 45.2% 감소하였다. 따라서 GS 코딩에서 스크램블 다항식의 선택이 MDSV 최악 성능에 중요한 요소임을 확인할 수 있다.

V. 결 론

DC-억압을 위한 멀티모드 코딩에서는 생성된 후보코드시퀀스 중 최적의 DC-억압 성능을 가진 코드시퀀스를 선택한다. 본 연구에서는 DC-억압을 위한 MDSV, MSW, MTO 기준의 GS 코딩을 정수계획법 모형으로 수식화하였다. 개발된 정수계획법 모형을 내부의 Min모형으로 할 때, GS 코딩에서 사용한 스크램블링 다항식의 최악 성능을 평가할 수 있는 MaxMin 모형을 개발하였다. 모의실험에서는 임의로 생성된 문제들에 대하여 제어비트 및 역치 값에 따른 MDSV, MTO 값과 최적해에서의 DSV 분산값을 비교하였다. 또한 제안한 MaxMin 모형을 이용하여 제어비트와 스크램블러 다항식에 따른 MDSV 최악 성능을 계산하였다. 2차 다항식과 11차 다항식을 비교한 결과, 11차 스크렘블 다항식이 MDSV 최악 성능이 55% 감소함을 확인했다. 또한 두 다항식을 이용하여 계산한 MDSV 최적해의 분산에서도 11차 다항식을 사용한 경우, 평균 45.2% 감소하였다. 본 연구에서 개발한 모형은 GS코딩에서 사용하는 스크램블러 다항식의 최악 성능 평가에 기여한다.

References

[1] K. A. S. Immink, *Codes for Mass Data Storage Systems*, Shannon Foundation Publishers, 1999.

[2] I. J. Fair, W. D. Grover, W. A. Krzymien, and R. I. MacDonald, "Guided scrambling: a new line coding technique for high bit rate fiber optic transmission systems," *IEEE Trans. Commun.*, vol. 39, no. 2, pp. 289-297, Feb. 1991.

[3] S.-W. Suh, S.-G. Ahn, J.-Y. Kim, and K. A. S. Immink, "Performance of GS913 (guided scrambling nine to thirteen) modulation code for high density rewritable optical disc," in *Proc. Int. Conf. Optical Memory Optical Data Storage Topical Meeting*, pp. 153-155, Hawaii, U.S.A., July 2002.

[4] J. Lee and J. Lee, "New DC-suppression method of modulation codes for high density optical recording systems," *J. KICS*, vol. 27, no. 1A, pp. 13-17, Jan. 2002.

[5] M. Lee, J. Lee, and J. Lee, "DC-suppression selection criteria of multimode modulation code for optical recording," *J. KICS*, vol. 28, no. 3C, pp. 209-214, Mar. 2003.

[6] K. A. S. Immink and L. Pátrovics, "Performance assessment of DC-free multimode codes," *IEEE Trans. Commun.*, vol. 45, no. 3, pp. 293-299, Mar. 1997.

[7] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL*, Scientific Press, 1993.

[8] IBM, *IBM ILOG CPLEX 12.4 User's Manual*, 2012.

박 태 형 (Taehyung Park)

1986년 2월 고려대학교 산업공학과 학사

1989년 2월 고려대학교 산업공학과 석사

1998년 6월 Virginia Tech 산업시스템공학과 박사

2001년~현재 숭실대학교 산업정보시스템공학과 부교수

<관심분야> 정수계획법, 통신네트워크

이 재 진 (Jaejin Lee)

1983년 2월 연세대학교 전자공학과 학사
1984년 12월 U. of Michigan, Dept. of EECS 석사
1994년 12월 Georgia Tech. Sch. of ECE 박사
1995년 1월~1995년 12월 Georgia Tech. 연구원
1996년 1월~1997년 2월 현대전자 정보통신 연구소 책임 연구원
1997년 3월~2005년 8월 동국대학교 전자공학과 부교수
2005년 9월~숭실대학교 정보통신전자공학부 교수
<관심분야> 통신이론, 채널코딩, 기록저장 시스템