

## 디지털 데이터 전송에서의 오류 검출 및 수정에 관한 연구

우익제\*, 신두진\*, 박대섭\*, 허욱열\*  
인하대학교 전기공학과\*

### A Study on the Error Detection and Correction for Digital Data Transmission

Ik-Jae Woo\* Doo-Jin Shin\* Dae-Sop Park\* Uk-Youl Huh\*  
Dept. of Electrical Engineering Inha University

**Abstract** - In this paper, for the more accurate data communication, Voting Algorithm is used to detect and correct some error. Voting is a fundamental operation in the realization of ultrareliable systems that are based on multi-channel computations. Finally, the performance of this algorithm is analyzed theoretically and some simulation results are presented to demonstrate the efficiency of the proposed algorithm.

## 2. 본 론

### 2.1 데이터 전송방식 및 오차 분석

#### 2.1.1 데이터 전송방식

데이터 전송 방식에는 폴링(Polling)기법, 인터럽트 방식, 핸드 셰이킹 방식 등이 있다. 오류 발생시 컴퓨터의 레지스터를 포함한 환경을 저장 및 복구에 필요한 시간 때문에 데이터 전송의 가장 주요한 문제인 '타이밍'을 유지하기 어렵다. 핸드 셰이킹 방식은 마스터 컴퓨터와 서브 컴퓨터 사이에 데이터 교환 승인 바이트의 교환으로 인한 전송률 저하를 가져오게 된다[4].

## 1. 서 론

최근 반도체의 집적 기술의 발달로 IC의 소형화와 고성능 반도체가 만들어 지고 있다. 더욱이 고속의 데이터 처리 능력으로 많은 분야에서 디지털 방식의 산업 자동화가 이루어 지고 있다. 이러한 변화는 디지털 데이터 전송이나 정보저장에 있어서 보다 효율적이고 신뢰성 있는 정보의 송수신을 요구하고 있다. 실제로 컴퓨터를 통해 데이터를 취득하고 시스템을 제어하는 이른바 "컴퓨터 제어 시스템(Computer-Controlled System)"이 독자적인 연구 분야가 될 정도로 성장하였다. 이러한 통신 시스템이 대규모화되고 고속화됨에 따라 정보의 신뢰성을 높이기 위해 시스템 설계자의 입장에서 가장 관심을 두어야 할 부분은 전송로상의 잡음으로 인하여 발생하는 오류의 제어 즉, 검출 및 수정(Detection and Correction : EDC)일 것이다. 이러한 오류 제어를 목적으로 데이터 통신 시스템에서 사용하는 기법으로는 ARQ(Automatic Repeat Request) 와 FEC(Forward Error Correction)가 있다[1]. ARQ는 수신 측에서 오류를 검출하게 되면 정보의 재전송을 송신 측에 요구함으로써 오류 없는 정보를 수신하고자 하는 방식으로 오류의 검출과 재전송을 수행하여 정보의 신뢰도를 높이는 방식이다. FEC는 오류 정정 부호(Error-Correction Code)를 사용하여 데이터, 음성, 영상신호 등 정보의 전송, 저장, 처리과정에서 발생한 오류를 수신 측에서 직접 정정함으로써 비트 오류(Bit Error Rate)이 아주 낮은 고 신뢰도의 통신이 가능하게 하는 방식이다. 본 논문에서는 대상 시스템의 동작을 감시하기 위한 감시용 장비-메인 컴퓨터-와 시험 장비의 데이터 통신에서 발생하는 데이터 오류를 검출하고 수정하는데 초점을 맞추었다. 하드웨어에서 발생하는 데이터 오류는 편의상 전송시 발생하는 오류로 취급하여 처리하였다. 오류 검출 및 수정을 위해서는 FEC방식을 따르는 Voting Algorithm을 사용하여 소프트웨어적으로 오류를 검출하고 수정하였다[2,3]. 모의실험은 60(Hz) 사인 파형을 발생시켜 한 주기에 대하여 실험하였으며, 샘플링 값을 메인 컴퓨터에 전송할 때 랜덤 함수를 발생시켜 전송데이터에 임의의 값을 삽입하였다. 원래의 신호에 외란을 15%와 35%를 포함한 두 가지 신호에 대하여 모의실험을 하였다.

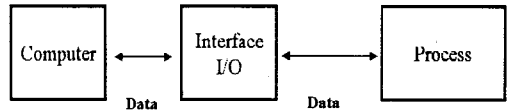


그림 1. 시스템의 기본적인 흐름도

본 논문의 대상 시스템은 폴링 기법에 의한 데이터 교환으로 메인 컴퓨터가 지속적으로 대상 프로세서의 정보를 읽어오도록 한다. 이럴 경우에 메인 컴퓨터는 대상 시스템의 샘플링 시간 이내에 있어서는 동일한 데이터 값을 얻게 되는데 이를 이용하여 오류 검출 및 수정이 가능하게 된다.

#### 2.1.2 오류 분석

데이터의 오류는 하드웨어 소자의 결함이나 노이즈 또는 기능상의 오류에 의해 발생된다. 메인 컴퓨터는 대상 시스템 상태가 정상인지 비정상인지를 알기 위해 프로세서 동작을 검사하여야 한다. 이때 관심이 되는 오류는 대상 시스템에서 메인 컴퓨터로 데이터를 전송할 때 데이터에 이상 신호에 의한 데이터의 변화에 의한 오류이다[5]. 산발적으로 나타나는 노이즈나 외란 등으로 인한 데이터의 변화에 대해서 오류 검출 및 수정을 할 수 있는 방법은 일반적으로 데이터 전송 시스템에 데이터 자체에 페리티 비트를 삽입하도록 회로를 디자인하여서 시스템 자체에서 오류 여부를 검사하여 오류가 발생하면 데이터를 다시 받아들이도록 하는 방식을 사용하거나 지속적으로 입력되는 데이터를 비교하여 오류값을 수정하는 데 감시용 장비 즉, 메인 컴퓨터와 대상 시스템과의 데이터 통신은 감시용 컴퓨터의 속도가 대상시스템 속도에 비해 빠르므로 대상 시스템의 상태가 변하기 전에 반복적으로 같은 데이터를 읽어 들인다. 이러한 데이터를 비교하여 데이터의 오류를 감지 및 수정할 수 있다.

### 2.2 Voting Algorithm

이 알고리즘은 데이터 통신이 다중 채널을 기반으로 하여 구현하는데 시스템의 신뢰도를 높이기 위한 기본적인 동작을 수행하는 알고리즘이다. Voting Algorithm 은 다중 채널로 구성된 하드웨어나 다양한 데이터를 처

리하는데 있어서 기본 구조가 같은 하드웨어에서 실행되는 다양한 프로그램 모듈이 실행될 때도 필요하다(2). 고속으로 동작하는 프로세서의 경우 데이터의 오류 검출 및 수정으로 인한 시간 지연으로 인한 오류가 발생되지 않도록 Voting Algorithm의 구현은 단순화 하는 것이 중요한데 이러한 이유로 Bit-Voting 방식이 사용된다.

### 2.2.1 BIT-VOTING

일반적으로 Voting Algorithm을 적용할 때 사용되는 방식으로 m-out-of-n Bit-Voter는 하드웨어적으로는 2단계 AND-OR 로직 회로로 구성된다. Bit Voter 회로는 n개의 bit 입력  $X_i, i=1,2,\dots,n$ 일 때 한 개의 출력 비트  $y$ 는 1로 값이 정해지기 위한 필요충분 조건으로 n개의 입력 중 m개 이상의 bit가 1이 되어야 한다. 그림2는 3 of 5 Bit Vorter를 로직회로로 구현한 회로이다.

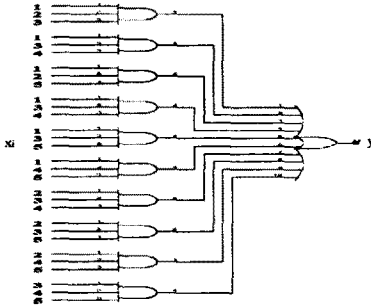


그림 2. 3-out-of-5 Bit Voter 회로도

Bit-Voter를 구현하기 위한 AND gate 수는 입력 단자가 m개인 AND gate일 경우 다음과 같다.

$$nC_m = \frac{n!}{m!(n-m)!} \quad (1)$$

그리고  $nC_m$ 개의 입력을 받는 OR gate 1개로 구성된다. 여기서 gate수의 증가는 소프트웨어 관점에서는 연산속도의 증가를 가져온다. 더욱이 n값과 m값의 변화는 gate수를 증가시키게 된다.

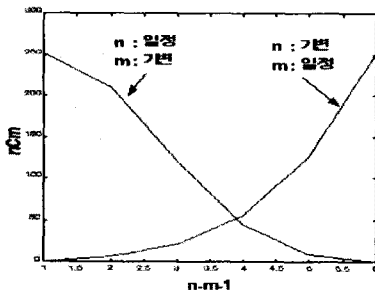


그림 3. Bit-Voter의 n과 m값의 변화에 따른 gate수 비교

그림3은  $m=5, n=5$ 에서 10으로 증가시킨 경우와  $n=10, m=5$ 에서 10으로 증가시킬 경우에 gate 수의 증가를 나타낸 그래프이다. 이 곡선은 가로축이 증감에 따라 지수적인 증가나 감소를 보이게 된다. 즉 BIT-VOTING 방식은 입력 비트 수가 큰 데이터 전송의 오류 검출 및 수정에 있어서는 시간적 지연을 가질 수 있다.

### 2.2.2 WORD-VOTING

m-out-of-n bit-voter는 Bit Voter를 확장한 것으로 회로는 n개의 워드 입력  $X_i, i=1,2,\dots,n$ 일 때 한 개의 출력 워드  $y$ 는 m개 이상 동일 값으로 입력된 값 1개를 가지거나 의사결정 실패를 나타내는 고유의 신호를 출력하게 된다.

### 2.3 오류 검출 및 수정

오류의 검출은 오류를 검출하기 위해 구성된 필터나 관측자를 이용하여 실제값과 추정값의 잔차(Residual)를 구해 이 값이 미리 설정된 문턱값(Threshold)를 넘어서게 되는가의 여부에 의해 이상 발생 유무를 판단함으로써 이루어지는 것이 일반적인 방식이다. 오류를 검출하기 사용된 Voting Algorithm 경우에는 별도의 필터나 관측자를 사용하지 않고 입력되는 데이터를 비교하여 오류 여부를 판단한다. r-bits 신호가 중복해서 n개 입력되었을 때 이러한 데이터 구간에서의 오류를 판정하기 위해 입력채널이 n개인 m-out-of-n Bit-Voter를 구성하여 입력되는 데이터 신호에 대한 오류 검출 수정이 가능하게 된다. 여기서 m값은 문턱값으로 사용될 수 있으며 m값의 증가는 Bit-Voter의 정확도를 높게 된다. 입력된 신호가  $n \times r$  행렬을 V라 할 때 Voting Algorithm의 입력으로

$$X_k(i) = V_{ki} \quad (2)$$

여기서,  $i = 1,2,\dots,n$  : 입력채널  
 $k = 1,2,\dots,r$  : k번째 voter

과 같이 된다.

다음은  $n=5, m=3$ 일 때 Voting Algorithm을 구현한 것이다. 출력  $Y_k$ 는 다음과 같다.

$$Y_k = \sum_{p=1}^3 \sum_{q=2}^4 \sum_{r=3}^5 X_k(p)X_k(q)X_k(r) \quad (3)$$

$k = 1,2,\dots,r$

위 식에서 결정된 출력  $Y_1$ 부터  $Y_r$ 는 원래의 r-bit로 구성된 데이터를 수정하게 된다.

### 2.4 시스템 구성

시험 장비로부터 읽혀진 데이터는 오류 제어기에 적합하도록 데이터를 처리하여야 한다.

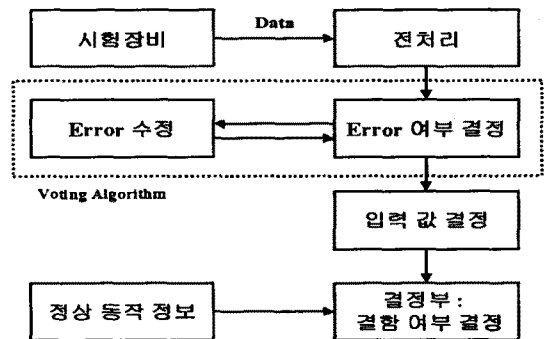


그림 4. 보팅 알고리즘의 software구조

샘플링이 h일때 비트 p로 된 데이터가 h초동안 동일한 데이터가 n개 들어온 경우이다. 이러한 데이터는 같은 열에 위치한 비트는 동일한 값을 가지고 있어야 한다. 그러므로 Bit-Voter에 사용되는 입력 비트는  $V_{1p}, V_{2p}, \dots, V_{np}$ 가 되고 이에 대한 출력값이 입력값을 대표하는 값으로 결정한다.

## 2.5 실험 결과 및 고찰

모의 실험은 다음과 같은 조건하에서 이루어졌다. 입력 값으로는 60[Hz]인 사인 파형의 크기를 7,000부터 13,000사이의 값을 가지게 하였다. 그리고 사인 파형의 출력값을 데이터로 받기 위하여 16비트를 사용하였으며, Voting Algorithm을 적용하기 위하여 Bit-Voter의 구성은 9-out-of-16 방식을 사용하였다. 사인 파형에 섞인 외란은 임의적으로 발생시킨 것이며, 각 데이터가 전송될 때 외란이 삽입될 확률 15%일 때와 35%인 경우 2가지에 대해서 실험하였다. 이러한 파형에 상기 고찰된 Voting Algorithm을 이용하여 모의 실험을 해 본 결과는 그림 6 과 그림 8과 같은 파형을 얻었다.

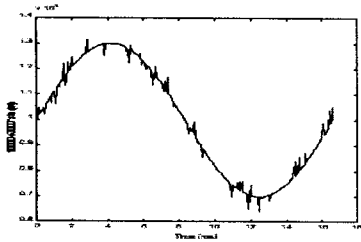


그림 5. 외란 15%일 때의 사인 파형

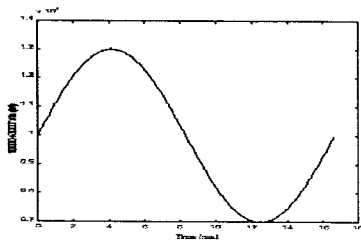


그림 6. 외란 15% 일 때 보팅 알고리즘을 적용한 사인 파형

각 데이터에 외란이 삽입될 확률 15% 일 때의 파형이 그림 5 이며 이에 Voting Algorithm을 적용한 파형이 그림 6이다. 모의 실험 결과에서 보여주듯이 그림 6은 외란이 포함된 파형이 원래의 파형과 같은 파형을 보여주고 있다.

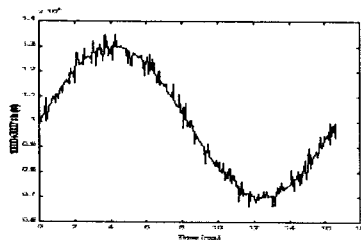


그림 7. 외란 35%일 때의 사인 파형

그림 7은 각 데이터에 외란이 삽입될 확률 35%을 가지는 사인 파형이며 이 파형에 Voting Algorithm을 적용한 파형은 그림 8이다. 이 파형은 그림 6과는 달리 어느 정도 외란성분이 남아 있는 것을 알 수 있다. 그림 6과 그림 8을 비교에서 알 수 있듯이 오류의 수정이 이루어지지 않은 이유는 동일한 데이터를 가져야 되는 구간에서 오차 성분이 원래의 신호 값보다 많아 졌기 때문이다[6].

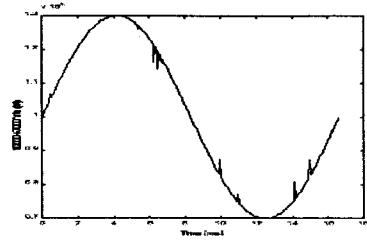


그림 8. 외란 35%일 때 보팅 알고리즘을 적용한 사인 파형

## 3. 결 론

본 논문은 데이터 통신의 신뢰도를 높이기 위해 데이터 오류를 검출하고 수정하기 위하여 Voting Algorithm을 적용하였다. 제안된 방식은 소프트웨어 측면에서 수행되는 것으로 하드웨어 설계시 데이터의 오류 및 수정을 위한 추가 설계 요소가 없으면서도 신뢰성 있는 데이터를 얻을 수 있다. 따라서 본 논문에서 제안한 Voting Algorithm을 이용하여 모의 실험을 한 결과에서 보듯이 데이터의 오류를 수정하는데 우수한 성능의 결과를 얻을 수 있었다.

향후 본 논문에서 데이터의 신뢰도를 더욱 높이기 위한 알고리즘의 연산 속도를 줄이기 위하여 더욱 처리 속도가 빠른 Voting Algorithm에 대한 연구를 진행 중이다.

## (참 고 문 헌)

- [1] 조용석, 이만영, "(255, 223) RS 부호의 직렬부호기", 한국통신학회 논문집 제13권 제5호, pp. 429-436, 1988.
- [2] B. Parhami, "Voting Algorithms", IEEE Trans. Reliability, vol 43, pp617-629, 1994
- [3] Lihao Xu, Jehoshua Bruck, "Deterministic Voting in Distributed Systems Using Error-Correcting Codes", IEEE Trans. Parallel and Distributed System, Vol. 9, pp813-824, 1998
- [4] STUART BENNETT, "Real-Time Computer Control", second edition, pp72-87, 1994
- [5] 김준희, 권오규 "다중 프로세서를 이용한 이상 허용 제어기의 설계 및 실현" 한국 자동 제어 학회 회의 논문집 vol. 1, pp 95-100, 1994
- [6] B. Parhami, "Voting networks", IEEE Trans. Reliability, vol 40, pp380-394, 1991