

실시간 고장허용 제어시스템을 위한 새로운 보팅 알고리즘

김민기, 황영호, 양해원
한양대학교 전자전기제어계측공학과

A Novel Voting Algorithm for Real Time Fault Tolerant Control Systems

Min-Gi Kim, Young-Ho Hwang, Hai-Won Yang
Dept. of Elec. Elec. Con. & Inst. Eng. Hanyang Univ.

Abstract - In this paper, a novel algorithm for adaptive majority voting using additional information is described. The novel algorithm increases the reliability of adaptive majority's history. The novel algorithm assists adaptive majority voter to select the result of the most reliable module. The performance of the proposed voter is evaluated through a series of fault injection simulations. The results are compared with those of adaptive majority voter. The empirical results show that the proposed voter has higher safety and availability levels than the adaptive majority voter.

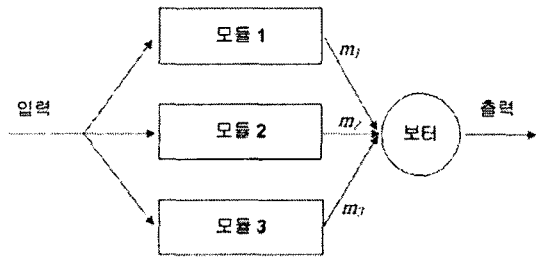


그림 1. TMR 시스템의 기본모델

1. 서 론

하드웨어 중복은 고장을 허용하기 위해서 널리 쓰이는 기법이다. 특히 하드웨어 중복기법 중 대표적인 것이 triple modular redundant(TMR) 시스템이다.

TMR 시스템은 중복 모듈과 보터를 사용하여 fault masking 기법을 수행하며, fault masking의 성능은 보팅 알고리즘에 기인한다. 그러므로 고신뢰성을 갖는 보팅 알고리즘 개발이 필요하다[1]. 기본적인 보팅 알고리즘은 다음과 같다.

만약 모듈의 결과값으로 과반수이상의 원소를 가진 집합을 만들 수 있다면 majority voter(MV)를 사용하여 집합 내의 원소 중 하나를 임의적으로 선택할 수 있다. 여기에 모듈의 결과값들 간의 일치정도를 나타내는 history를 추가한다면 adaptive majority voter(AMV)를 사용할 수 있다. AMV는 MV에서 선택한 집합 내의 원소 중 history가 가장 큰 원소를 선택함으로써 MV의 성능을 한층 증가시켰다[2][3].

본 논문에서 제안된 새로운 보팅 알고리즘은 AMV에 새로운 정보를 추가적으로 사용하여 AMV의 취약한 점을 개선하고 보다 신뢰성있는 시스템을 구현하고자 한다.

2. 본 론

2.1. TMR 시스템의 구조

본 논문에서는 하드웨어 중복 기법 중 TMR 시스템을 고려한다. TMR 시스템은 그림 1과 같다[4].

그림 1에서 보는 바와 같이 입력은 세 모듈에 동일하게 들어간다. 모듈 1, 2, 3은 동일한 모듈이다. 이 세 모듈들의 결과값은 보터의 입력으로 들어간다. 보터는 보팅 알고리즘에 따라 이들 결과값들 중 하나를 선택하여 출력한다.

2.2. History 정보의 구성

AMV는 history를 이용하여 보팅을 하고 NAMV는 history 뿐만 아니라 추가정보를 이용하여 보팅을 한다. history와 추가정보는 보터가 모듈의 결과값 중 정확한 값을 선택하는데 중요한 역할을 한다. N개의 모듈을 가진 N-modular redundant(NMR) 시스템은 history와 추가정보를 아래와 같이 구성할 수 있다.

단계 1-1 : 서로 다른 두 모듈의 결과값 m_i 와 m_j 에 대해 아래 식의 관계를 확인한다.

$$|m_i - m_j| \leq \alpha \tag{1}$$

여기서 $i, j = 1, 2, 3, \dots, N$ 이고 $i \neq j$ 이다. N은 모듈의 총 개수이다. α 는 보터 임계값을 나타낸다. 보터 임계값은 두 모듈의 결과값이 동일하다고 간주할 수 있는 최대 거리이다.

만약 위의 식 (1)을 만족한다면 파라메타 a_{ij} 를 1로 설정하고 그렇지 않으면 a_{ij} 를 0으로 설정한다.

단계 1-2 : 각 모듈에 대해 (N-1)개의 부울 대수로 이루어진 집합을 아래와 같이 만든다.

$$A_i = a_{ij}, a_{i(j+1)}, \dots, a_{i(N-1)}, a_{iN} \tag{2}$$

여기서 $i, j = 1, 2, 3, \dots, N$ 이고 $i \neq j$ 이다. 모듈 i에 대해 식 (2)에서 생성한 집합의 원소를 가지고 다음과 같이 R_i 를 구한다.

$$R_i = \sum_{j=1}^N a_{ij} \tag{3}$$

여기서 $i = 1, 2, \dots, N$ 이고 $j \neq i$ 이다. 식 (3)에서 얻은 R_i 를 가지고 다음의 식에서 S_i 를 구한다.

$$\begin{cases} \text{If } R_i \geq (N-1)/2 \text{ then } S_i = 1 \\ \text{else } R_i < (N-1)/2 \text{ then } S_i = 0. \end{cases} \tag{4}$$

단계 1-3 : 모듈 i에 대해 q번 실행동안 S_i 의 누적된 값을 history h_i라고 정의하고 다음과 같다.

$$h_i(q) = \sum_{l=1}^q S_i(l). \quad (5)$$

식 (5)에서 얻은 h_i 는 모듈 i 가 q 번 실행하는 동안 다른 모듈과 과반수이상 보터 임계값 내에 존재하는데 기여한 총 횟수를 나타낸다.

단계 1-4 : h_i 를 정규화한 값을 상태지시자 P_i 로 정의하고 다음과 같이 나타낸다.

$$P_i(q) = \frac{1}{q} \sum_{l=1}^q S_i(l) \quad (6)$$

P_i 는 모듈 i 에 대한 신뢰척도가 되며, 가장 높은 P_i 값을 가진 모듈이 옳은 결과값을 출력할 확률이 가장 높다.

2.3. New adaptive majority voter(NAMV) 설계

NAMV를 아래와 같은 순서로 설계할 수 있다.

단계 2-1 : 임의의 q 번째에서 각 모듈의 결과값을 m_i 라고 정의하자. 여기서 $i = 1, 2, 3, \dots, N$ 이다. m_i 를 다음과 같이 집합 M 으로 표시한다.

$$M = \{m_1, m_2, m_3, \dots, m_N\} \quad (7)$$

이 때, $(q-1)$ 번째까지의 각 모듈의 history는 다음과 같다.

$$H(q-1) = \{h_1(q-1), h_2(q-1), \dots, h_N(q-1)\}. \quad (8)$$

집합 M 의 원소 중 식 (1)을 만족하고 m_i 와 m_j 를 최대한 많이 가지도록 하는, M 의 부분집합 $V_1, V_2, V_3, \dots, V_k$ 을 만든다.

단계 2-2 : 부분집합 $V_1, V_2, V_3, \dots, V_k$ 중 원소수가 가장 많은 집합을 V 로 설정한다.

단계 2-3 : $n(V) \geq (N+1)/2$ 이면, 부분집합 V 로부터 보터의 출력을 정한다. 이 때, 모듈은 가장 큰 h_i 를 가지고 있다. $n(V)$ 는 V 의 원소 개수이다.

V 에 속한 모듈의 h_i 는 1씩 증가한다. 반면에 부분집합 V 에 속하지 못한 모듈의 h_i 는 변하지 않는다.

단계 2-4 : $n(V) < (N+1)/2$ 이면, 보터는 '결과없음'을 출력하고 h_i 는 변하지 않는다.

단계 2-5 : 다음의 식을 만족하는 모듈이 있는지 확인한다.

$$R_i \geq (N+1)/2 \quad (9)$$

여기서 $i = 1, 2, 3, \dots, N$ 이다.

식 (9)을 만족하는 모듈의 개수를 z 라고 정의하고 다음 식을 만족하는지 확인한다.

$$z \geq (N-1)/2. \quad (10)$$

만약 식 (10)를 만족한다면 단계 2-3에서 얻은 보터의 출력을 무시하고 식 (9)에서 얻은 모듈 중 하나를 보터의 출력으로 정한다. 이 때, h_i 는 단계 2-3, 2-4를 따른다.

Example : NAMV의 입력이 다음과 같다.

$$M = \{9.6, 10.4, 10.6, 11.1, 10.9\} \quad (11)$$

이 때, $(q-1)$ 번째까지의 각 모듈의 history는 다음과 같다.

$$H(q-1) = \{70, 79, 94, 96, 82\}. \quad (12)$$

우선 보터 임계값 0.5라고 가정하고 다음과 같이 집합 M 의 부분집합을 구한다.

$$V_1 = \{10.4, 10.6, 10.9\}, n(V_1) = 3$$

$$V_2 = \{10.6, 10.9, 11.1\}, n(V_2) = 3 \quad (13)$$

$n(V_1) = n(V_2) \geq (5+1)/2$ 이기 때문에 V_1, V_2 는 모두 V 가 될 수 있다. AMV는 임의로 V 를 택해 V 의 원소 중 가장 높은 h_i 를 가지고 있는, V_1 의 10.6 또는 V_2 의 11.1를 출력으로 선택한다. 그러나 NAMV는 앞서 정의한 단계 2-5에 의해서 $R_3 = R_5 \geq (5+1)/2$ 이고 $z = 2 \leq (5-1)/2$ 이기 때문에 세 번째 모듈의 결과값인 10.6 또는 다섯 번째 모듈의 결과값인 10.9를 출력으로 선택한다.

$$A_1 = \{a_{12}, a_{13}, a_{14}, a_{15}\} = \{0, 0, 0, 0\} \rightarrow S_1 = 0, R_1 = 0$$

$$A_2 = \{a_{21}, a_{23}, a_{24}, a_{25}\} = \{0, 1, 0, 1\} \rightarrow S_2 = 1, R_2 = 2$$

$$A_3 = \{a_{31}, a_{32}, a_{34}, a_{35}\} = \{0, 1, 1, 1\} \rightarrow S_3 = 1, R_3 = 3 \quad (14)$$

$$A_4 = \{a_{41}, a_{42}, a_{43}, a_{45}\} = \{0, 0, 1, 1\} \rightarrow S_4 = 1, R_4 = 2$$

$$A_5 = \{a_{51}, a_{52}, a_{53}, a_{54}\} = \{0, 1, 1, 1\} \rightarrow S_5 = 1, R_5 = 3$$

그러므로 q 번째까지 각 모듈의 history는 다음과 같다.

$$H(q) = \{70, 80, 95, 97, 83\}.$$

3. 모의실험 및 결과

3.1. 실험 방법

보터의 성능을 평가하기 위해 아래 그림과 같은 실험 모델을 이용한다.

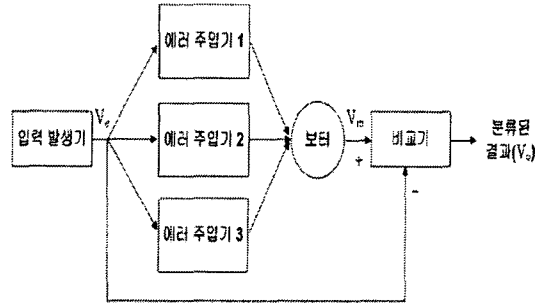


그림 2. 모의실험의 보터 모델

그림 2에서 보는 바와 같이 입력 발생기는 각 주기가 다 값을 발생시킨다. 에러 주입기는 입력 발생기로부터 받은 값에 정해진 간격 $[-\theta, +\theta]$ 내에서 무작위로 에러를 주입한다. 보터는 보팅 알고리즘에 따라 하나의 결과값을 출력한다.

비교기는 고정 성분이 없는 값 V_e 와 보터 출력값 V_m 의 차를 이용하여 V_e 를 얻을 수 있다. V_e 는 다음과 같이 세 가지 결과로 나눌 수 있다.

$$i) n(V) \geq (N+1)/2 \text{ 일 때} \quad \begin{cases} \text{If } |V_m - V_e| \leq \beta \text{ then } V_e \text{ is CO} \\ \text{else } |V_m - V_e| > \beta \text{ then } V_e \text{ is IC} \end{cases} \quad (15)$$

$$ii) n(V) < (N+1)/2 \text{ 일 때} \quad V_e \text{ is DIS} \quad (16)$$

여기서 CO는 보터가 옳은 값을 선택한 경우이고, IC는 보터가 옳지 않은 값을 선택한 경우이며, DIS는 보터가 어떤 값도 선택을 하지 못한 경우이다. β 는 정확도 임계값을 나타낸다. 정확도 임계값은 V_m 을 CO라고 수용할 수 있는 한계거리이다.

V_e 가 CO인 개수를 n_c 로 정의하고, IC인 개수를 n_{ic}

로 정의하며, DIS인 개수를 n_{dis} 로 정의하자. 총 시행횟수를 n 이라고 했을 때 다음의 관계가 성립함을 알 수 있다.

$$n = n_c + n_{ic} + n_{dis} \quad (17)$$

본 논문에서 n 은 10000번이고, V_m 은 $100+100\sin(t)$ 이다. 샘플링주기는 0.1이다. 보터 임계값과 정확도 임계값은 모두 0.5이다. 그리고 각 에러 주입기는 입력 발생기로부터의 값에 에러를 주입시키되 하나의 에러주입기는 다른 두 개의 에러주입기보다 에러진폭을 더 크게 주입시킨다.

보터의 성능평가 기준은 이용도(A)과 안정도(S)이다. 이용도는 보터가 옳은 결과값을 많이 뽑을 확률로써 다음과 같이 정의한다.

$$A = \frac{n_c}{n} \quad (18)$$

안정도는 보터가 옳지 않을 결과값을 적게 뽑을 확률로써 다음과 같이 정의한다.

$$S = 1 - \frac{n_{ic}}{n} \quad (19)$$

보터가 이상적인 경우에는 A와 S가 모두 1이다.

3.2. 실험 결과

그림 3은 세 보터 MV, AMV, NAMV의 이용도 결과를 보여준다. 가로축은 에러의 최대진폭을 나타낸다. 세 보터는 10000번 시행해서 얻은 값을 정규화 시켜서 나타낸 수치이다. NAMV가 아래 그림에서 보는 바와 같이 가장 좋은 성능을 보여 준다.

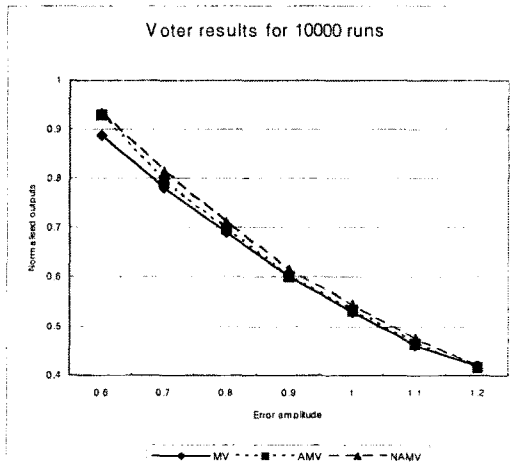


그림 3. 이용도 결과

그림 4는 세 보터 MV, AMV, NAMV의 안정도 결과를 보여준다. 그림 3에서의 이용도와 마찬가지로 안정도에서도 NAMV가 가장 좋은 성능을 보여준다.

4. 결 론

본 논문에서 제안한 NAMV는 AMV에 추가정보를 사용하여 AMV의 성능을 개선시킨 알고리즘으로서, 그림 3과 4에서 보는 바와 같이 이용도와 안정도가 모두 증가하였다. 여러 보터들은 각기 장단점을 가지고 있어서 실제 시스템에 보터를 적용할 때에는 적용되는 시스템의 성질이나 요구조건에 따라 적절한 보터를 선택해야 한다. 이에 NAMV를 사용하면 AMV 보다 좋은 성능을

얻을 수 있다.

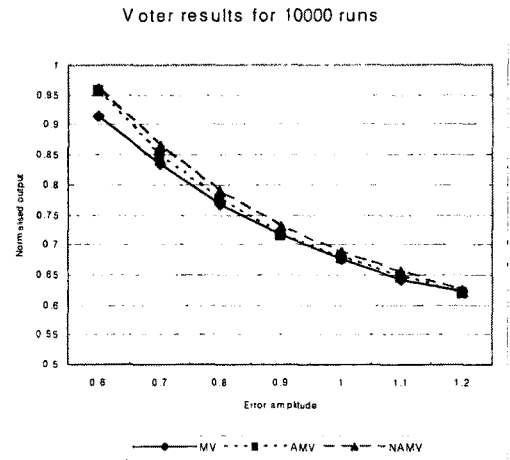


그림 4. 안정도 결과

[참 고 문 헌]

- [1] J. M. Bass, G. Latif-Shabgahi and S. Bennett, "Experimental Comparison of Voting Algorithms in Cases of Disagreement", Proc. of the 23rd EUROMICRO Conference, pp. 516-523, 1997
- [2] P. R. Lorczak, A. K. Caglayan, and D. E. Eckhardt, "A Theoretical Investigation of Generalised Voters", Digest of papers FTCS'19: IEEE 19th Int. Ann. Symp. on Fault-Tolerant Computing Systems, pp. 444-451, June 1989
- [3] G. Latif-Shabgahi, and S. Bennett, "Adaptive Majority Voter: A Novel Voting Algorithm for Real-Time Fault-Tolerant Control Systems", Proc. of the EUROMICRO Second Workshop on Dependable Computing Systems, Vol. II, pp. 113-119, 8-10 Sept. 1999
- [4] B. W. Johnson, "Design and Analysis of Fault-Tolerant Digital Systems", Addison-Wesley Publishing Company, 1989