

분산형 PLC 시스템에서의 고장 허용 제어

정영미\* · 정석권\*\*

\*부경대학교 대학원 메카트로닉스, \*\*부경대학교 기계공학부

Fault Tolerant Control for Distributed Programmable Logic Controller System

Young-Mi Jeong\*, Seok-Kwon Jeong\*\*

\* Mechatronics Engineering, Graduate School, Pukyong National University, Busan 608-739, Korea

\*\*School of Mechanical Engineering, Pukyong National University, Busan 608-739, Korea

**Abstract** - Most of distributed PLC system is secured reliability of the system by fully duplicated controller. But it costs too much. In this paper, we proposed fault tolerant control using minimum spare controllers. First, the system is modeled using Galois field. Next, we designed additional spare controllers using the parity code of two dimensions. Finally, the algorithm for producing estimated output in remote I/O when the fault occurs was suggested. This method will be applied to an automatic system in order to increase reliability and improve cost performance.

대의 분산형 PLC 시스템은 Fig. 1의 (a),(b)를 제외한 부분으로서 각 컨트롤러가 리모트와 공통의 네트워크에 접속되어져 있으며, 이산량만을 취하는 것으로 한다.

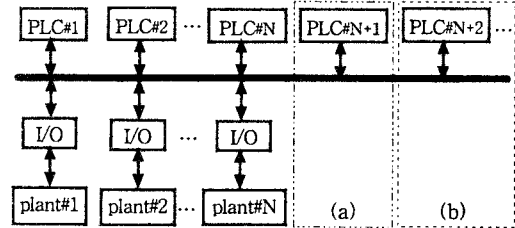


Fig. 1 Model of distributed PLC system

i번째 컨트롤러의 제어칙을 부울대수상의 방정식으로 표현하면 (1),(2)와 같다<sup>2)</sup>.

$$Y_i(t) = h_{Bi}(X_i(t), U_i(t)) \tag{1}$$

$$X_i(t+1) = f_{Bi}(X_i(t), U_i(t)) \tag{2}$$

단,  $X_i$ 는 i번째 컨트롤러의  $n_i$ 차원의 상태벡터,  $U_i$ 는  $m_i$ 차원의 입력벡터,  $Y_i$ 는  $p_i$ 차원의 출력벡터이다. 함수  $f_{Bi}$ ,  $h_{Bi}$ 는 부울대수의 연산으로 정의되어진다. 부울대수는 거리가 정의되지 않으므로 차분방정식의 해석에는 적합하지 않다. 그러므로 사칙연산이 정의되어지는 표수(2)의 가우아제(Galois field) GF(2)의 표현으로 변환한다<sup>3)</sup>.

Table 1 Boolean algebra and Galois field  
(a) Arithmetic in Boolean algebra

		$a \vee b$	
		b=0	b=1
a	b	0	1
	0	0	1
1	1	1	

		$a \wedge b$	
		b=0	b=1
a	b	0	1
	0	0	0
1	0	1	

(b) Arithmetic in Galois field

		$a + b$	
		b=0	b=1
a	b	0	1
	0	0	1
1	1	0	

		$a \times b$	
		b=0	b=1
a	b	0	1
	0	0	0
1	0	1	

		$a - b$	
		b=0	b=1
a	b	0	1
	0	0	1
1	1	0	

		$a \div b$	
		b=0	b=1
a	b	0	1
	0	-	0
1	1	-	1

GF(2)는 일반적인 정수 연산에 대해서 2의 잉여계산을 적용한 결과를 그 값으로 한다.

입의 요소(a, b)에 대해 부울대수 표현을 가우아제 표현으로 변환하면 Table 1에 의해 (3)과 같이 된다.

$$a \vee b \rightarrow a + b + ab, a \wedge b \rightarrow ab, \bar{a} \rightarrow 1 + a \tag{3}$$

시스템함수  $f_{Bi}$ 와 출력함수  $h_{Bi}$ 는 부울대수 연산으로

1. 서 론

현재, 대부분의 분산형 프로그래머블 로직 컨트롤러(programmable logic controller, 이하 PLC) 시스템은 요구되는 신뢰 정도에 따라 제어기인 PLC의 2중화나 3중화를 통해 신뢰성을 확보하고 있다. 그러나, 이 방법은 시스템 구축에 고비용을 초래하며, 제어기의 중복 설치로 인한 설치 장소등의 제약 요인이 발생한다. 이를 해결하기 위해 여분의(spare) 제어기 1대만을 부가적으로 설치하는 방법이 제안되어 있다<sup>1)</sup>. 그러나, 이 방법은 복수의 제어기에 고장이 동시에 발생할 경우에는 그 대안이 마련되어 있지 않으므로 대응이 불가능하다는 단점을 가진다. 그러므로 복수의 제어기에 대한 동시 고장에 대해서도 제어계의 신뢰성을 향상시킬 수 있는 고장 허용 제어법이 절실히 요구되고 있다.

본 논문에서는 이 문제를 해결하기 위한 한 방안으로서 2차원 패리티(parity) 출력을 이용한 고장 허용 제어법을 제안한다. 우선, 부가해야할 spare 제어기의 대수가 전체 제어기의 대수에 따라 어떻게 결정될 수 있는지, 그리고 부가할 제어기를 2차원적으로 배열하는 방법에 대해서 검토하였다. 또한, 리모트 I/O에 대한 설계를 행하고 이를 통해 고장 허용제어가 가능함을 이론적으로 증명하였다.

제안된 방법은 복수의 spare 제어기를 부가함으로써 단일 제어기의 고장뿐만 아니라 복수 제어기의 동시 고장에 대해서도 대응이 가능한 방법이다. 따라서 이 방법은 향후 분산형 PLC 시스템 설계시 복수대의 제어기 고장을 허용하는 신뢰성 높은 고장 허용 제어계 설계에 유용하게 적용될 수 있을 것으로 기대된다.

2. 본 론

2.1 분산형 PLC 시스템의 모델링

분산형 PLC 시스템은 네트워크상에 접속한 독립된 프로그램을 가지는 PLC로 구성되어 있다. 이러한 시스템은 네트워크상에서 실시간적으로 정보를 전달하기 때문에 고장이 발생할 경우에도 즉각적인 대응이 가능하다. N

서 (3)에 의해 GF(2)의 가루아제 연산으로 나타내면 (1)과 (2)는 (4)와 (5)로 각각 표현된다.

$$Y_i(t) = h_i(X_i(t), U_i(t)) \quad (4)$$

$$X_i(t+1) = f_i(X_i(t), U_i(t)) \quad (5)$$

## 2.2 단일 제어기 고장에 대한 고장허용제어기 설계

Fig. 1에서 임의의 PLC 한 대에 고장이 발생할 경우를 상정한다. 이 경우, Fig. 1의 (a)와 같이 한 대의 spare 컨트롤러를 부가하여 이 제어기가 동적인 다른 컨트롤러의 출력에 대해 패리티 출력을 발생하도록 (6), (7)과 같은 spare 제어기의 제어칙을 설계한다.

$$Y_{N+1}(t) = h_{N+1}(X_{N+1}(t), U_{N+1}(t)) \quad (6)$$

$$X_{N+1}(t+1) = f_{N+1}(X_{N+1}(t), U_{N+1}(t)) \quad (7)$$

$X_{N+1}(t), U_{N+1}(t)$ 는 서브시스템 전체의 상태벡터를 갖는 확장된 상태벡터와 입력벡터로서 (8), (9)로 표현된다.

$$X_{N+1}(t) = [X_1(t), X_2(t), \dots, X_N(t)]^T \in GF(2)^{n_{N+1}} \quad (8)$$

$$U_{N+1}(t) = [U_1(t), U_2(t), \dots, U_N(t)]^T \in GF(2)^{m_{N+1}} \quad (9)$$

(7)의 함수  $f_{N+1}(t)$ 는  $f_{N+1}(t) = [f_1(t), f_2(t), \dots, f_N(t)]^T$ 로서 서브시스템 전체의 시스템함수로부터 구할 수 있으며, 출력벡터의 차원은 서브시스템의 출력벡터의 차원 중 가장 큰 차원  $p_{N+1} = \max(p_1, p_2, \dots, p_N)$ 을 취한다. 또한, 출력함수  $h_{N+1}(t)$ 는 (10)과 같이 설계되어진다.

$$h_{N+1} = h_1 + h_2 + \dots + h_N \quad (10)$$

(10)을 이용하면 패리티 조건을 만족시키는 출력이 동적으로 발생가능하며 (4), (10)에 의해 다음 식이 성립함을 알 수 있다.

$$\sum_{i=1}^{N+1} Y_i(t) = 0 \quad (11)$$

$i$ 번째의 컨트롤러가 시간  $t \geq t_i$ 에 있어서 고장이 발생했을 때, 그 출력  $y_i(t)$ 가 잘못된 신호를 내는 경우를 고려한다.  $i$ 번째 컨트롤러의 고장이 I/O에 의해 검출 가능하다고 가정하고, 그 고장 플래그를  $\theta_i(t) = 1$ 로 정의한다. 이때,  $i$ 번째 컨트롤러의 출력 추정치는 (12)를 이용하여 구할 수 있다.

$$\hat{Y}_i(t) = (1 - \theta_i(t)) Y_i(t) + \theta_i(t) \sum_{j=1}^{N+1} Y_j(t) \quad (12)$$

고장이 발생한 경우  $\theta_i(t) = 1$ 로 되므로 (6)에 의해 (13)과 같이 표현되어지며 본래의 출력이 얻어짐을 알 수 있다. 그러므로 각  $i$ 번째 리모트 I/O에 (12)의 출력 추정이 가능한 소규모의 논리회로를 장착하면 본래의 출력을 얻는 것이 가능하게 된다.

$$\hat{Y}_i(t) = \sum_{j=1}^{N+1} Y_j(t) = h_i(X_i(t), U_i(t)) \quad (13)$$

2.3 복수 제어기 고장에 대한 고장허용제어기 설계  
분산형 PLC 시스템에 있어서 고장은 하나의 제어기 뿐만 아니라 복수의 제어기에서 동시에 발생할 수도 있다. 그러므로 이하에서는 복수의 고장에 대해서도 대응이 가능한 spare 제어기의 제어칙 설계법을 검토한다. 이는 2.2절의 1차원 패리티를 2차원 패리티 개념으로

확장함으로써 얻어진다. 이 방법은 Fig. 1의 (b)와 같이 spare 컨트롤러를 다수 대 부가하여 설계되어질 수 있다.

전체 컨트롤러에 최소의 spare 제어기를 부가하기 위해서 다음의 정의를 이용한다.

[정의 1] 전체 컨트롤러 수  $N$ 에서 정수  $c$ 를 뺀  $N-c$ 가 행렬  $a \times b$ 를 구성할 때,  $a + b + c$ 가 최소로 되는  $a, b, c$ 이면  $a + b + c$ 를 최소의 spare 제어기 수라 한다. 단,  $0 \leq c \leq 3, a, b < N$ 인 정수이다.

[정의 2]  $a + b + c$ 는 부가해야할 spare 제어기의 총 수를 나타낸다. 이 가운데  $c$ 는 이중화 형식으로 배치해야할 spare 제어기의 수이다.

[정의 3]  $a$ 와  $b$ 는  $N-c$ 대의 컨트롤러를 Fig. 2와 같이  $a$ 행  $b$ 열의 행렬로 배치되어 있다고 가정하였을 때 각 행과 열에 부가될 spare 제어기의 수이다.

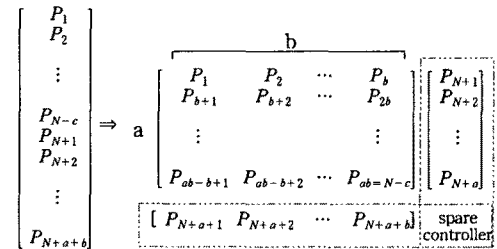


Fig. 2 Array of spare controllers

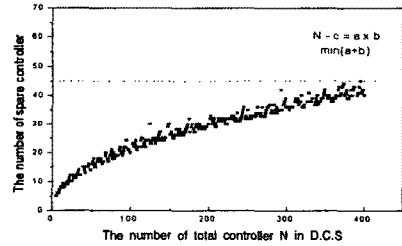


Fig. 3 Estimation number of spare controller

Fig. 3은 시스템의 제어기 총 수  $N$ 을 6부터 400까지로 가정하여 이 때 부가해야할 spare 제어기의 수를 [정의 1]에 의해 추정한 결과이다. 이 결과로부터 시스템의 제어기 수가 400일 경우에도 부가해야 할 spare 제어기의 수는 45대 내의임을 알 수 있다.

spare 제어기의 제어칙은 각 행과 열에 부가된 spare 제어기가 그 행과 열의 동적인 다른 컨트롤러의 출력에 대해 수평, 수직 패리티 출력이 발생하도록 설계된다. 우선, 임의의  $k$ 행 spare 제어기의 제어칙은 수평 패리티 출력이 발생되도록 (14), (15)와 같이 설계된다. 여기서,  $k$ 는  $1 \leq k \leq a$ 인 정수이다.

$$Y_{N+k}(t) = h_{N+k}(X_{N+k}(t), U_{N+k}(t)) \quad (14)$$

$$X_{N+k}(t+1) = f_{N+k}(X_{N+k}(t), U_{N+k}(t)) \quad (15)$$

$$X_{N+k}(t) = [X_{(k-1)b+1}(t), X_{(k-1)b+2}(t), \dots, X_{(k-1)b+a}(t)]^T \in GF(2)^{n_{N+k}} \quad (16)$$

$$U_{N+k}(t) = [U_{(k-1)b+1}(t), U_{(k-1)b+2}(t), \dots, U_{(k-1)b+a}(t)]^T \in GF(2)^{m_{N+k}} \quad (17)$$

(16)은  $k$ 행에 대한 서브시스템 전체의 상태벡터를 갖는 확장된 상태벡터이고, (17)은  $k$ 행에 대한 서브시스템 전체의 입력벡터를 갖는 확장된 입력벡터이다.

$f_{N+k}(t) = [f_{(k-1)b+1}(t), f_{(k-1)b+2}(t), \dots, f_{(k-1)b+b}(t)]^T$  (18)  
 시스템함수 (18)은  $k$ 개의 서브시스템 전체의 시스템함수로 이루어진다. 출력벡터의 차원은 서브시스템의 출력벡터의 차원 가운데 최대의 것을 가지는 것으로  $\hat{p}_{N+k} = \max(\hat{p}_{(k-1)b+1}, \hat{p}_{(k-1)b+2}, \dots, \hat{p}_{(k-1)b+b})$ 이다. 한편, 출력함수  $h_{N+k}$ 는 다음과 같다.

$$h_{N+k} = h_{(k-1)b+1} + h_{(k-1)b+2} + \dots + h_{(k-1)b+b} \quad (19)$$

이와 같은 출력함수의 설계에 의해 행에 대한 동적 패리티 조건을 만족하는 출력이 발생 가능하다.

$$\sum_{i=(k-1)b+1}^{(k-1)b+b} Y_i(t) + Y_{N+k}(t) = 0 \quad (20)$$

다음으로, 임의의 열에 대한 spare 제어기의 제어척은 수직 패리티 출력이 발생되도록 (21), (22)와 같이 설계된다. 여기서,  $l$ 은  $1 \leq l \leq b$ 인 정수이다.

$$Y_{N+a+l}(t) = h_{N+a+l}(X_{N+a+l}(t), U_{N+a+l}(t)) \quad (21)$$

$$X_{N+a+l}(t+1) = f_{N+a+l}(X_{N+a+l}(t), U_{N+a+l}(t)) \quad (22)$$

열에 있어서 서브시스템 전체의 상태벡터와 입력벡터를 갖는 확장된 상태벡터, 입력벡터는 (23), (24)와 같이 정한다.

$$X_{N+a+l}(t) = [X_l(t), X_{b+l}(t), \dots, X_{(a-1)b+l}(t)]^T \in GF(2)^{n \times \dots} \\ n_{N+a+l} = n_l + n_{b+l} + \dots + n_{(a-1)b+l} \quad (23)$$

$$U_{N+a+l}(t) = [U_l(t), U_{b+l}(t), \dots, U_{(a-1)b+l}(t)]^T \in GF(2)^{m \times \dots} \\ m_{N+a+l} = m_l + m_{b+l} + \dots + m_{(a-1)b+l} \quad (24)$$

$$f_{N+a+l}(t) = [f_l(t), f_{b+l}(t), \dots, f_{(a-1)b+l}(t)]^T \quad (25)$$

열에 대한 서브시스템 전체의 시스템함수는 (25)로 표현되어지며, 출력벡터의 차원은 서브시스템 전체의 출력벡터의 차원 가운데 최대의 것을 가지는 것으로  $\hat{p}_{N+a+l} = \max(\hat{p}_l, \hat{p}_{b+l}, \dots, \hat{p}_{(a-1)b+l})$ 이다. 한편, 출력함수  $h_{N+a+l}$ 는 다음과 같다.

$$h_{N+a+l} = h_l + h_{b+l} + \dots + h_{(a-1)b+l} \quad (26)$$

이러한 출력함수의 설계에 의해 열에 대한 동적 패리티 조건을 만족하는 출력이 발생 가능하다.

$$\sum_{j=0}^l Y_{b+j}(t) + Y_{N+a+l}(t) = 0 \quad (27)$$

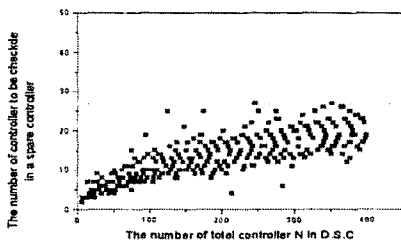


Fig. 4 The number of controller to be checked in a spare controller by column and row

Fig. 4는 시스템 제어기의 총 수에 대하여 각 행과 열에 부가된 spare 제어기 한 대가 감시해야 할 제어기의 수를 나타낸다. 시스템의 제어기의 총 수가 400까지 증가하여도 하나의 spare 제어기가 감시해야 할 제어기는 대략 30대 이내임을 알 수 있다.

$i$ 번째 컨트롤러의 고장이 발생하였을 경우 고장 플래그를  $\theta_i(t) = 1$ 로 정의한다. 이 때,  $i$ 번째의 컨트롤러의 추정치  $\hat{y}_i(t)$ 는 다음에 표시한 알고리즘에 의해 구해질 수 있다.

<알고리즘>

R: for ( $k=1; k \leq a; k++$ ) {

$$r_k = \sum_{p=1}^k \theta_{(k-1)b+p}$$

if  $r_k = 1$  {  
 $\theta_r = 1$

$$\hat{Y}_r = \sum_{p=1, p \neq r}^k \theta_{(k-1)b+p} Y_{(k-1)b+p} + Y_{N+k}$$

$\theta_r = 0$  }

}

C: for ( $l=0; l \leq b; l++$ ) {

$$c_l = \sum_{q=0}^{c-1} \theta_{qb+l}$$

if  $c_l = 1$  {  
 $\theta_c = 1$

$$\hat{Y}_c = \sum_{q=0, q \neq \frac{c-l}{b}}^1 Y_{qb+l} + Y_{N+k+l}$$

$\theta_c = 0$  }

}

go to R

$i$ 번째의 컨트롤러의 고장이 검지 되었다면  $\theta_i(t) = 1$ 로 변환된다. 복수의 제어기에서 동시에 고장이 발생할 수 있으므로 고장이 발생된 제어기의 값을 추정하기 위해 각 행과 열에 대해 고장 개수를 검사한다. 행과 열 가운데 행을 우선으로 하고 고장이 하나만 발생한 행이나 열이 존재할 경우, 그 컨트롤러의 추정치  $\hat{y}_i(t)$ 를 우선적으로 구한다. 추정치가 구해진 컨트롤러의 고장을 제외하고 다시 나머지 행과 열에 대해 축차적으로 추정치를 구한다. 본 논문에서 제안하는 방법은 수직, 수평 패리티 출력을 이용하여 고장이 발생한 컨트롤러의 출력치를 추정하기 때문에 행과 열에 두 개 이상의 고장이 동시에 발생하였을 때는 추정이 어려운 단점이 있다.

### 3. 결 론

본 연구에서는, 분산형 PLC 시스템에서 단일의 제어기 고장뿐만 아니라 복수의 제어기가 동시에 고장이 날 경우까지를 상정하여 2차원 패리티 출력을 이용한 고장 허용 제어법을 제안하였다. 이를 위해 제어기의 총 수에 따라 필요한 최소의 spare 제어기의 수를 결정하고, 2차원 패리티의 동적 출력을 이용할 수 있도록 이들을 효과적으로 배열하는 방법도 정의하였다. 그리고 리모트 I/O의 설계를 통해 고장 허용 제어가 가능함을 이론적으로 증명하였다. 제안된 방법은 분산형 PLC 시스템에 있어서 복수 제어기의 동시 고장에 대해서도 신뢰성이 높은 시스템의 구축이 가능할 뿐만 아니라 설치 장소의 제약도 줄일 수 있다. 따라서 네트워크상에 접속된 분산형 PLC 시스템과 같이 정보가 실시간으로 교환 처리되는 자동화 시스템에 적용하면 효과를 볼 것으로 기대된다.

### [참 고 문 헌]

- [1] Y. Fujimoto, T. Sekiguchi, "Fault Tolerant Configuration of Distributed Discrete Controllers", T.IEE Japan, vol 121-D, No 3, 2001
- [2] 정석권·양주호·유삼상 공역, "시퀀스 제어공학", 인터비전, 2002
- [3] Y. Fujimoto, T. Sekiguchi, "An Algebraic Approach to Design of Discrete system", IEEE IECON, pp 2068~2613, 2000