

우선순위를 고려한 고속 CAN 통신 메시지 버퍼-스케줄링 방법

한 신*, 윤 태웅*, 최 익**
 고려대학교 전기과*, 한국과학기술 연구원**

Priority Considered Fast CAN Message Buffer-scheduling Method

Shin Han*, Tae-wonng Yoon*, Ick Choy**

*Dept. of Electrical Eng. Korea Univ.,

**Intelligent System Control Resech Center, KIST

Abstract - CAN 통신은 네트워크 시스템 제어를 통한 각종 산업설비에서 제어 및 자동화 관련 장비들간에 데이터 교환을 위한 통신망으로 널리 사용되고 있다. 이러한 CAN통신을 응용하여 실시간 메시지 송수신이 가능한 시스템을 구성하려면 메시지의 특성분류 및 메시지의 증가에 따른 트래픽 발생을 고려해야 한다.

CAN의 제한된 속도에 따른 실시간 메시지 전송의 여부는 데드라인을 통한 메시지 설계로 성능을 보장받고 있다. 하지만 이러한 메시지의 송수신에 있어 단순 CAN네트워크 망에 의존한 설계만으로 메시지의 신뢰성을 보장받을 수 없다.

네트워크 상황에 따른 버퍼를 통한 능동적인 메시지 스케줄링 방법을 통하여 성능효과를 보장받을 수 있는 방법을 제안한다.

1. 서 론

CAN(1)(Controller Area Network)은 원래 독일의 Bosch에서 차량용 네트워크 시스템으로 개발되었으나, 추후에 각종 산업 자동화 분야로 적용 범위가 확장된 네트워크 시스템이다.

CAN의 특성 중 CSMA/NBA(Carrier Sense Multiple Access with Non-destructive Bit Wise Arbitration) 라는 메시지 전송 메커니즘을 이용해 네트워크 프로토콜을 디자인 할 수 있다. 프로토콜의 디자인함으로써 네트워크의 상태에 따른 각 메시지의 의미를 부여할 수 있고, 능동적인 네트워크의 구성이 용이해진다.

CAN의 하드 리얼타임(Hard-real time)특성을 고려한 프로토콜은 다양하게 설계되었고, 성능은 어느 정도 보장받고 있다. CAN을 이용한 실시간 메시지 전송방법 중에서 메시지 인식자 필드(identifier)에 메시지의 전송제한 시간을 포함시켜 전송하는 DM(Deadline Monotonic) 방법(2)이 있고, 이를 응용하여 능동적인 데드라인 할당 방법인 ED(Earliest Deadline) (3)도 개발되었다. 또한 최근에 기존에 ED 방법에 스케줄링 방법을 감안한 MTS(Mixed Traffic Scheduling) (4) 방법이 제시되었다.

메시지의 생성에서 도착까지의 시간적 제한을 CAN 네트워크에서 바라본 것이 데드라인 방법이라면, 시스템 내부 발생해서 CAN으로 전송하기 이전의 시간적 제한을 받는 것이 바로 시스템 버퍼이다. 이러한 버퍼를 FIFO 방식으로 처리하면 CAN 통신에 의한 지연효과를 버퍼내부에서 모두 흡수해야 하는데, 이러한 FIFO 방식의 버퍼는 많은 문제점을 가지고 있다. 기존에 약간의 시간적인 지연을 감안한 고정 데드라인 방법(5)을 제시하기도 하였다. 하지만 능동적으로 데드라인을 조정해서 스케줄링하는 동적 스케줄링이 가능한 방법은 고안되지 않았다.

이에 본 논문은 데드라인 CAN통신 프로토콜을 이용하여, 시스템에 필요한 메시지를 우선 순위 및 데드라인

으로 분류하고 네트워크상에 약간의 트래픽이 발생했을 때에도 실시간 메시지의 전송 가능한 동적 스케줄링 방법을 도입하여, 메시지 전송의 신뢰성을 향상시키고 나아가 시스템 전반의 뛰어난 성능향상을 기대할 수 있는 알고리즘을 제시한다.

2. 본 론

2.1 CAN 메시지 Frame 설계.

CAN통신에서의 데드라인을 이용한 전송방법은 '시간 제한' 의미를 갖는 데드라인을 중재(Arbitration) 필드 중 인식자 필드에 포함시켜 전송하는 방법이다(그림 1).

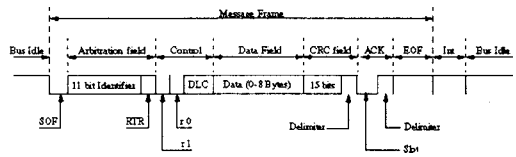


그림 1 CAN 메시지 프레임 (기본형)

CAN 메시지 특성상 인식자 필드는 각각의 메시지의 종류에 따라 고유하게 부여되는 숫자인데 이에는 그 메시지의 버스 접근 우선 순위를 담고 있다. 즉 숫자가 낮을수록 높은 우선 순위를 가지고 버스를 사용할 수 있다. 또한 이 인식자 필드에 의해서 메시지의 충돌도 막을 수 있는데 "비 파괴 bitwise" 중재방식을 통해 충돌한 메시지 중 높은 우선 순위의 메시지가 지연시간 없이 버스를 이용할 수 있다.

이러한 특성을 이용하여 메시지 필드를 그림 2와 같이 설계한다. 최 상위 2비트를 메시지의 최상위 우선순위를 고려하기 위해 설계하고 중간 부분은 스케줄된 데드라인으로 구성한다. 그리고 마지막 부분은 인식자 필드에서 고유위치를 지정 또는 고정순위 데드라인을 이용할 수 있도록 설계한다. 응급메시지는 항상 전송할 수 있도록 최상위 우선순위를 갖는다. 실시간 제어용 메시지 중 주기 또는 비 주기 메시지는 스케줄링된 인식자 값을 중간에 넣어서 전송할 수 있도록 설계한다.

비 실시간으로 전송 가능한 센서 데이터나, 시간적인 제한이 없는 데이터 참조용 메시지 프레임은 고정 우선순위를 가진 "Deadline Monotonic" 방법을 통해 전송한다. 만약 이러한 메시지 중 우선순위의 변화를 가져야 할 필요가 있을 때 빠른 메시지 프레임 또는 응급메시지

00	Deadline	Command
	응급 메시지 용 (실시간)	
01	Scheduled Deadline	Deadline (Priority)
	빠른 PR 메시지 용 (실시간 제어용)	
10	Deadline Monotonic	
	비 실시간 PR 메시지 용 (Data 참조용)	

그림 2 CAN 메시지 프레임과 데드라인

프레임으로 변화 가능한 능동적인 스케줄 방법을 도입한다.

2.2 데드라인 스케줄링 방법.

2.2.1 시스템 구성 및 스케줄링.

주기적인 메시지만 고려한 경우 각 메시지의 주기에 따른 데드라인은 우선순위를 고려해 설계하되 메시지의 주기를 넘지 않는 범위에서 설계해야한다. 또한 전체 메시지의 평균 전송 가능시간을 고려해서 설계해야 한다.

$$T_{max} \geq \sum_{i=0}^n \left(\frac{T_{max}}{T_i} \right) C_i \quad (i=1 \dots n) \quad (1)$$

이는 주기적 메시지들이 가장 메시지가 긴 T_{max} 안에 평균적으로 전송이 가능하도록 설정하는 방법이다. 여기서 C_i 는 메시지의 전송시간을 의미한다.

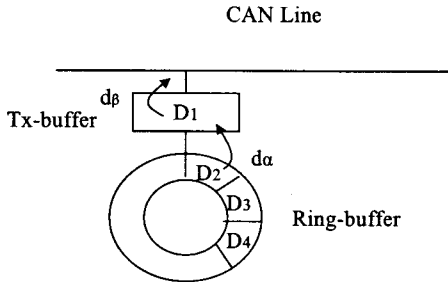


그림 3 버퍼 지연시간

그림3은 한 노드에 전송버퍼(Tx-buffer)와 링 구조 버퍼(Ring-buffer)를 나타낸다. 메시지는 발생하면, 일단 버퍼에 저장한다. CAN의 중재에 의해 전송버퍼의 데이터가 전송되면, 버퍼에 저장된 데이터를 스케줄에 의한 우선순위에 따라 전송버퍼에 넣는다.

2.2.2 선입선출 방법(FIFO).

선입선출 방법은 먼저 발생한 메시지를 전송버퍼에 넣는 스케줄링 방법이다. 선입선출 방법을 통한 CAN 데드라인 메시지 송수신시 메시지가 수식1을 만족해도, 버퍼에서 생기는 지연시간의 누적으로 인한 메시지의 데드라인-어긋 현상은 발생하게 된다. 그림3과 같이 전송버퍼(Tx-buffer) 지연(d_b)는 CAN 네트워크 상태에 따라 좌우된다. 하지만 링 버퍼 지연시간(d_a)은 스케줄링을 통한 능동적으로 변경이 가능하다. 그러므로 버퍼 지연시간의 조정으로 전송버퍼 지연시간의 변동을 가져올 수 있다.

이에 전송 버퍼에 데드라인 우선 스케줄링을 방법을 도입한다.

2.2.3 데드라인 스케줄링 방법.

데드라인기준 우선 전송 스케줄링 방법은 각 데드라인 메시지마다 일정수준의 버퍼지연 시간을 제공하는 장점이 있다. 하지만 이러한 데드라인 우선 전송 스케줄링 방법을 이용하여 CAN통신을 하면 다음과 그림4와 같은 문제점이 발생한다.

현재 스케줄링 시간이 t 라 할 때, 메시지 1의 데드라인은 메시지 2의 데드라인 보다 우선 하게 된다. 하지만 데드라인 우선순위에 의해 메시지 1은 메시지 2보다 먼저 CAN 통신기회를 갖게 된다. 메시지 1이 전송으로 지연이 발생하면, 메시지 2는 데드라인이내에 전송할 수 있는 기회를 잃어버리기 쉽다.

이러한 단순 데드라인 스케줄링 방법이 데드라인까지 남은 시간을 고려하는 방법이[5] 도입되었다. 하지만 기존의 남은 시간을 고려한 방법은 고정된 데드라인을

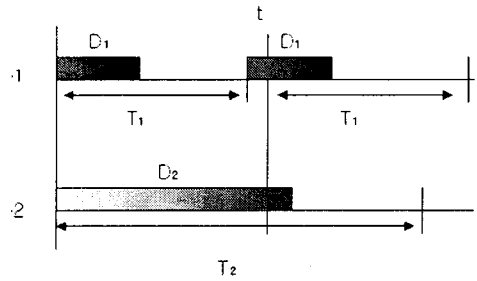


Fig 4 데드라인 스케줄링 방법

기준으로 제시되었으나, 능동적으로 변경 가능한 데드라인을 고려하지는 못한다. 이에 네트워크 상태에 따른 능동적으로 여유시간을 보장할 수 있는 데드라인을 고려한 평균비율 여유시간 스케줄링 방법을 제안한다.

2.3 평균비율 여유시간 스케줄링 방법.

평균 비율 여유시간 스케줄링 방법은 주기적 메시지의 평균 지연시간을 고려한 방법을 도입해 전체 데드라인과의 비율을 능동적으로 스케줄링 한다. 각 메시지의 여유시간은 메시지의 전송 후 결정되는 값으로 보내진 메시지의 데드라인까지의 남은 시간을 일컫는다. 이는 Trade-off 방식으로 여유시간의 비율을 고려 메시지 ID의 증감법을 통해 적정의 여유시간이 보장된 데드라인을 능동적으로 구할 수 있다.

스케줄링 방법은 다음과 같다.

- $T_{(i)}$: i-번째 메시지 주기
- $T_{C(i)}$: i-번째 메시지 전송 시간
- $T_{D(i)}$: i-번째 메시지 데드라인
- $I_{D(i)}$: i-번째 메시지 인식자(ID) 값

$$T_{(i)}, T_{D(i)}, I_{D(i)} \quad (i=0 \dots n: T_{C(i)} \text{ get after Transmit})$$

$$\text{If } (T_{C(i)} < T_{D(i)}) \{$$

$$\text{If } \left(\frac{(T_{D(i)} - T_{C(i)})}{T_{D(i)}} > P(k) \right)$$

$$\{ I_{D(i+1)} = I_{D(i)} \times \frac{P(k)}{1 - P(k)}, P(k+1) = P(k) + 0.01 \}$$

$$\text{Else } \{ I_{D(i+1)} = I_{D(i)} \times \frac{1 - P(k)}{P(k)}, P(k+1) = P(k) - 0.01 \}$$

)Else

$$\{ I_{D(i+1)} = I_{D(i)} \times \frac{1 - P(k)}{P(k)}, P(k+1) = P(k) - 0.1 \}$$

여기서 P는 비율을 의미하며 0.99에서 0.01 값이 된다. 또한 비율이 0.5(50%)에 귀결하도록 스케줄링 한다.

3. 시뮬레이션

본 장에서는 실제 시스템에 도입하기 전에 가상 시뮬레이션을 통한 스케줄링 방법의 성능효과를 관찰한다.

실험조건:

1. 모든 메시지는 주기적으로 발생한다.
2. 모든 메시지의 통신시간(C_i)는 최대 전송시간으로 고정시킨다.
3. 각 메시지는 평균 메시지 전송 조건 및 평균 데드라인을 어기지 않는 조건을 만족하도록 설계한다.

이에 스케줄링이 가능한 버퍼와 스케줄이 되지 않는 버퍼의 데드라인-어긋 현상을 시뮬레이션을 통한 실험을 한다. 메시지 통신 시간은 $C_i = 150 \mu\text{sec}$ 로 정한다

노드 번호	주기		데드라인		노드 번호	주기		데드라인	
	ms	(us)	(us)			ms	(us)	(us)	
0	4	2167	2168		0	4	2169	2170	
0	5	2705	2706		0	10	5404		
0	20	10803			0	50	27001	27002	
1	4	2171	2172		1	5	2713	2714	
1	10	5415							
2	2.5	1366			2	4	2177		
2	10	5418	5419		3	50	27020		
3	2	1101			3	4	2182		
3	10	5423			3	20	10824		
3	50	27025							
4	4	2186	2187		4	4	2188	2189	
4	10	5430	5431		4	10	5432	5433	
4	20	10834			4	50	27035		

표 1 노드별 주기 및 데이터

위 노드 정보를 가지고 평균 주기 어김-조건을 확인 해 보면, $T_{max}(50000\mu\text{sec}) \geq \sum_{i=0}^n \left(\frac{T_{max}}{T_i} \right) C_i = 43875\mu\text{sec}$

를 만족한다. 즉 각 메시지는 평균적으로 데드라인 및 주기를 벗어나지 않는 범위 내에서 통신을 하게 된다.

노드 4의 4ms 주기 중에서 가장 우선순위가 낮은 ID 인 2189 μs 데드라인 메시지에 대한 1초 동안 시뮬레이션 한 결과를 비교한다.

3.1.1 선입선출 스케줄링 방법.

CAN 통신에서 발생하는 전송비퍼 지연(d_B)으로 인한 데드라인-어김 현상이 발생했다.

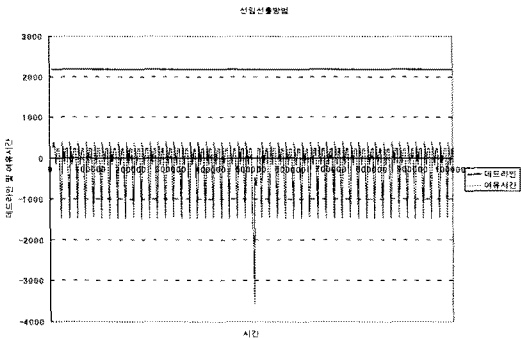


그림 5 선입선출 방법

총 466개의 데드라인-어김 메시지가 발생했다. 데드라인-어김 메시지 종류 및 개수는 표 2와 같다.

노드 번호	주기		데드라인		어김		노드 번호	주기		데드라인		어김	
	ms	(us)	(us)		(개)			ms	(us)	(us)		(개)	
2	2.5	1366			2	2	4	2177			1		
3	4	2182			3	2	1101			99			
4	4	2186			4	4	2187			51			
4	4	2188			4	4	2189			99			

표 2 데드라인-어김 메시지

3.1.2 데드라인 스케줄링 방법.

총 98개의 데드라인-어김 메시지가 발생했다(그림7). 데드라인-어김 메시지 종류 및 개수는 표3과 같다.

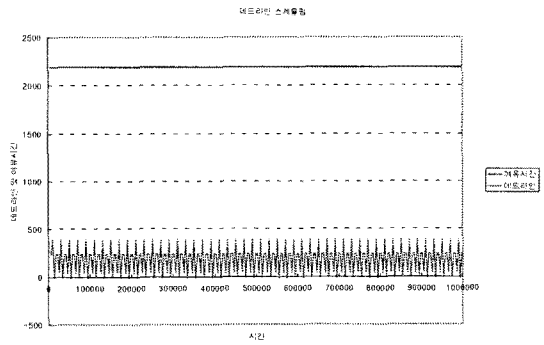


그림 7 데드라인 스케줄링 방법

노드 번호	주기		데드라인		어김		노드 번호	주기		데드라인		어김	
	ms	(us)	(us)		(개)			ms	(us)	(us)		(개)	
1	5	2714			48		2	4	2189			50	

표 3 데드라인-어김 메시지

3.2 평균비율 여유시간 스케줄링 방법

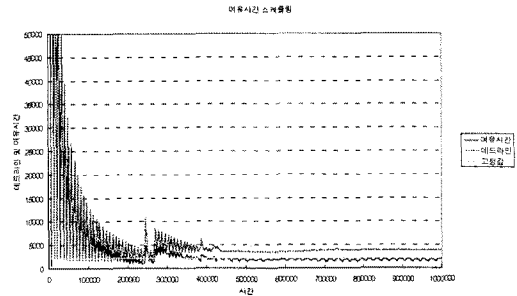


그림 8 여유시간 스케줄링 방법

그림8과 같이 여유시간 및 데드라인이 처음엔 발산하다가 서서히 여유시간을 보장하는 데드라인으로 수렴한다. 스케줄링 도중에 데드라인-어김 메시지(44개)가 발생하였으나, 10초 정도 데드라인 조정이 끝난 후 데드라인-어김 메시지는 더 이상 발생하지 않는다.

4. 결 론

일반적인 방법보다 데드라인 우선 스케줄링 방법은 데드라인-어김 메시지를 상당량 감소시킨다. 이를 기반으로 평균비율 여유시간 스케줄링 방법을 도입하면, 기존의 장점에도 각 메시지의 여유시간을 보장해주기 때문에, 능동적인 노드 상황에 따른 여유시간이 보장된 데드라인을 설정 가능하게 해준다.

차후 이러한 능동적인 여유시간을 메시지의 특정 우선순위를 보장하는 방법에 도입할 예정이다.

[참 고 문 헌]

age on controller Area Network for Real-time CIM Applications," IEEE trans. Robot. Automat., vol 13, pp 310-314 April, 1997.

[5] K. G. Shin, "Real-Time Communication in Multi-hop Networks", IEEE Trans. Parallel and Distributed Systems, Vol5, No10, Oct, 1994

[1] CAN, International Standard 11898, 'Road Vehicles-Interchange of Digital Information-Controller Area Network for High-Speed Communication' (ISO, 1993)

[2] J. Y. T. Leung, and J. Whitehead, "On the Complexity of Fixed-priority Scheduling of Periodic, Real-time Tasks," Performance Evaluation, vol 1, 2, no. 4 pp.237-250, 1982

[3] K. G. Shin, "Real-time Communication in a Computer-Controlled Work-cell," IEEE trans. Automat., vol 13, pp 310-314 April, 1997.

[4] K. M. Zuberi, and K. G. Shin "Scheduling Mess