CAN 기반 FCHEV 시뮬레이터의 시간 해석 연구

A Study on Timing Analysis of a CAN-Based Simulator for FCHEVs

안봉주*, 이남수*, 양승호**, 손재영**, 박영환**, 안현식***, 정구민****, 김도현***
Bong-Ju Ahn, Nam-Su Lee, Seung-Ho Yang, Jae-Young Son, Young-Hwan Park,
Hyun-Sik Ahn, Gu-Min Jeong and Do-Hyun Kim

Abstract - In this paper, a timing analysis is performed for the CAN-based simulator system for a fuel cell hybrid electric vehicles. The CAN protocol is recently being used for conventional vehicles, however, the network-induced delay can make the in-vehicle network system unstable. This problem may be occurred in the future vehicles because more ECUs are being required than recent vehicles. In order to develop a stable network-based control system, timing analysis is required at the design process. Throughout this analysis, timing parameters that affect transmission delay are examined and an effective method of predicting a sampling time for a stable communication via CAN protocol. In order to show the validity of suggested timing analysis, some experiments are performed using DSPs with CAN module.

Key Words : FCHEV, Fuel Cell, CAN, WCET, WCRT

1. 서론

연비, 출력, 안전성 및 승차감을 향상시키기 위해 점차 많은 수의 ECU(Electronic Control Unit)들이 차량에 장착되고 있다. 최근에는 ECU들이 다양한 센서신호를 공유할 수 있고, ECU간의 정보교환이 효율적으로 이루어지도록 ECU 및 센서들을 CAN(Controller Area Network)을 중심으로 연결하여 사용하는 경우가 일반적이다.

일반적인 실시간 분산제어 시스템에서와 마찬가지로 차량내 네트워크 시스템에서도 가장 중요한 것 중 하나는 실시간 처리가 요구되는 태스크들의 마감시간에 대한 만족이다. 차량이외에 산업 전반에 걸쳐 CAN을 비롯한 필드버스의 사용이 확장됨에 따라 데이터 전송의 실시간성을 보장하기 위한 연구와 한정된 네트워크의 대역폭을 효율적으로 사용하기 위한 방안들이 여러 분야에서 연구되고 있다. [1-4]

차량에서 생성되는 데이터는 크게 산발적 또는 주기적으로 생성되는 실시간 데이터와, 제어 루프의 센서와 제어기에서 주기적으로 생성되는 제어 데이터 및 데이터 전송 지연시간에 크게 구애를 받지 않는 비실시간 데이터로 구분되며, 이러한 데이터들은 통신망이 제공하는 하나의 네트워크 미디어를 공유한다. 실시간 처리를 요구하는 차량의 분산 제어 시스템에서 네트워킹으로 인한 실시간 및 제어 데이터의 지연시간이 허용 한계치를 초과하는 경우에는 차량 제어시스템의 기능과 성능에 치명적인 악영향을 미칠 수 있다.

본 연구에서는 우선, 무공해 차세대 차량인 FCHEV(Fuel Cell Hybrid Electric Vehicles)에 대한 CAN 기반 시뮬레이터를 DSP 보드를 중심으로 구성하고, 홀리스틱(holistic) 스케줄링 해석 기법을 이용하여 네트워크시스템의 시간해석을 수행한다. 기존 연구에서는 간단한 선행관계를 가지고 차체 네트워크 시스템에 한정하여 시간 해석을 하였으나, 본 논문에서는 FCHEV 시뮬레이터를 이용하여 차량 전체 시스템에 응답 시간해석을 수행하고, 특히 실시간 처리가 요구되는 데이터를 고려하여 이를 마감시간 내에 처리할 수 있는 방안을 제시하고자 한다.

2. CAN 기반 시스템의 시간해석

2.1 최악 수행시간(WCET)과 최악 응답시간(WCRT)

여러 개의 노드가 분산되어있는 시스템에서는 각 노드간의 통신이 태스크의 응답시간에 영향을 미치게 된다. 이러한 문제점을 고려한 스케줄링 해석 방법을 홀리스틱 스케줄링 해석이라 한다. 홀리tm틱 스케줄링 해석을 하기 위해 최악 수행시간(Worst Case Execution Time : WCET)과 최악응답시간(Worst Case Response Time : WCRT)에 대한 조사가 필요하다[4, 7-8].

최악 수행시간은 태스크가 다른 태스크의 방해 없이 한번 수행하는데 걸리는 가장 긴 시간을 의미한다. 최악 응답시간은 태스크의 실행이 요구된 후 동작을 완료하는데 필요한 가장 긴 시간이며, 식 (1)과 같이 나타낼 수 있다.

$$R_i = w_i + J_i \quad (1)$$

저자 소개
* 國民大學校 電子情報通信工學部 碩士課程
** 國民大學校 電子情報通信工學部 學士課程
*** 國民大學校 電子情報通信工學部 教授 • 工博
****國民大學校 電子情報通信工學部 助教授 • 交信著者

식 (1)에서 R_i는 태스크 i에 대한 WCRT이고, w_i는 식(2)로 나타낼 수 있으며,

$$w_i = C_i + \sum_{\forall j \in hp(i)} \left[\frac{w_i + J_j}{T_j} \right] C_j \quad (2)$$

여기서, J_i는 태스크 i의 release jitter이다.

2.2 단일 프로세서에서의 스케줄링 해석

본 절에서는 단일 프로세서 시스템에서 고정 우선순위의 태스크들이 수행되는 경우에 스케줄링 해석을 소개한다.

$$R_i = C_i + B_i + I_i \quad (3)$$

$$I_i = \sum_{\forall j \in hp(i)} \frac{R_i}{T_j} C_j \quad (4)$$

$$R_i = C_i + B_i \sum_{\forall j \in hp(i)} \frac{R_i}{T_j} C_j \quad (5)$$

$$R_i = C_i + B_i + J_i \sum_{\forall j \in hp(i)} \left[\frac{R_i + J_j}{T_j} \right] C_j \quad (6)$$

여기서 R_i는 태스크 i에 대한 WCRT 이고, C_j 는 태스크 j 의 WCET 이다. B_i 는 낮은 우선순위의 태스크로부터 지연될 수 있는 가장 긴 시간이며 I_i 는 Worse Case interference 이고 $hp(i)$는 같은 프로세서 내에서 태스크 i 보다 높은 우선순위의 모든 태스크들의 집합이다. 또한, T_j 는 태스크 j 가 연속적으로 도착할 때 도착시간 간격의 최소값이다.

2.3 CAN기반 경우의 시간 해석

단일 프로세서 스케줄링 해석은 태스크의 실행과 메시지 전송이 유사하므로, CAN의 timing analysis에도 적용 가능하다. 네트워크의 송신단의 태스크가 완전히 전송되어 수신단에 도착한 것을 측정한 WCRT는 식(7)과 같다.

$$R_m = J_m + w_m + C_m \quad (7)$$

여기서 R_m은 WCRT이고, J_m은 queuing jitter, C_m은 주어진 메시지 m의 WCET이다. w_m 항은 우선순위가 명령되어진 대기열 중에서 메시지 m이 전송되기 시작한 시각까지의 가장 긴 시간(최악 queuing delay)을 의미하므로, w_m은 다음과 같은 식으로 결정된다.

$$w_m = B_m + \sum_{\forall j \in hp(m)} \left[\frac{w_m + J_j + \tau_{bit}}{T_j} \right] C_j \quad (8)$$

여기서 B_m은 자신보다 낮은 우선순위를 같는 메시지로 인해 대기하게 되는 가장 긴 시간을 의미하고, $h_p(m)$은 동일 프로세서에서 태스크 m 보다 더 높은 우선순위를 갖는 모든 태스크 집합을 의미하며, $1\tau_{bit}$는 bus의 bit time(1Mbps일 때 1us) 이다. B_m은 메시지 m의 최악 blocking time 이고 다음과 같이 표현된다.

$$B_m = \max_{\forall k \in lp(m)} (C_k) \quad (9)$$

여기서 $lp(m)$은 낮은 우선순위 메시지들의 집합이다. 만약 불확실한 크기의 soft real-time 메시지들의 정해지지 않은 수라면 B_m은 약 $130\,\tau_{bit}$ 정도 된다.

C_m항은 메시지 m이 물리적으로 CAN 버스를 점유한 가장 긴 시간을 의미한다. 이 시간 안에는 frame overheads, data contents, extra stuff bits가 차지한 시간도 포함된다.

$$C_m = \left(\left[\frac{34 + 8s_m}{5} \right] + 47 + 8s_m \right) \tau_{bit} \quad (10)$$

여기서 s_m은 메시지 m의 byte 수이다.

2.4 홀리스틱 해석 모델

다중 노드로 구성된 분산제어 시스템에 대한 시간 해석을 하기 위해서, 먼저 그 시스템의 구조와, 시간해석을 위한 파라미터에 대한 정의 및 이해가 필요하다.

분산제어 시스템의 각 노드 간 전송메시지의 지터는, 그 메시지의 전송 태스크가 원인이 된다. 따라서 노드 간 전송 메시지의 지터는 식 (11)에서처럼 전송 태스크의 WCRT와 같다.

$$J_i = R_{send(i)} \quad (11)$$

식 (12)와 같이 송신측 노드에서 발생한 메시지 전송이벤트의 발생부터 수신측 노드의 실행 완료까지 걸리는 전체 시간은 전송 태스크의 WCRT, 메시지가 전송되는 시간의 WCRT, 수신태스크의 WCRT를 모두 더한 값이 됨을 알 수 있다. 여기서 w_i와 $w_{dest(i)}$는 각각 메시지 i의 대기시간과 수신 태스크의 시간을 의미한다. 다음 식(12)의 $R_{end-to-end}$는 그림 1의 송신 노드에서 메시지 전송 시작까지의 응답시간과 메시지 전송에 걸리는 시간(Communication)과 메시지를 이용해 실제 출력을 내보내는데 걸리는 시간(Computation)을 포함한다.[4],[9]

$$R_{end-to-end} = R_{send(i)} + w_i + C_i + w_{dest(i)} \quad (12)$$

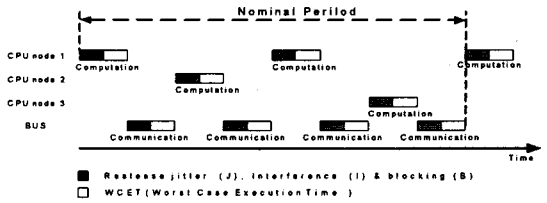

그림1. CAN system 의 기본 모델

3. 실험구성 및 결과

3.1 시뮬레이터 시스템의 구성

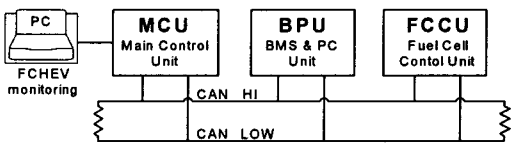

그림2. FCHEV 시뮬레이터의 기본 모델

본 논문의 실험은 그림 2와 같이 3개의 TMS320LF2407로 구성된 FCHEV Simulator를 이용하며, 구현된 시뮬레이터의 각 CPU노드들은 FCHEV power train의 각 서브시스템에 대한 시뮬레이션 코드가 이식되어 있다. Main Control Unit (MCU)에는 차량 동역학 및 표준 드라이브 사이클이 코딩되어 있고, BMS & PC Unit(BPU)에는 배터리 운용 시스템과 전력 배분제어 알고리즘이 코딩되어 있으며, Fuel Cell Controller Unit(FCCU)에는 연료전지의 수학적 모델링이 코딩되어 있다.

또한, 다음과 같은 순서에 따라 메시지가 각 노드에 전송된다. 먼저 MCU에서는 차량 구동에 필요한 전력 요구량을 계산(MCU(1))하여 BMS & PC Unit(BPU)로 메시지를 송신한다(M_to_B(2)). MCU로부터 메시지를 수신한 BPU에서는 배터리 SOC값과 Fuel Cell 전류요구량을 계산(BPU(3))하여 MCU로 메시지를 송신한다(B_to_M(4)). BPU에서 송신된 메시지를 수신한 MCU에서는 Fuel Cell 전력 요구량을 계산(Main(5))하여 FCU로 메시지를 송신한다(M_to_F(6)). FCU는 Fuel Cell이 출력 할 수 있는 전력을 계산(FCU(7))하여 MCU로 메시지를 송신(F_to_M(8))한다. FCU에서 받은 메시지를 수신한 MCU는 마지막으로 배터리와 Fuel Cell이 현재 출력할 수 있는 전력을 계산(MCU(9))한다.

3.2 실험 수행 및 결과 해석

실험 1에서는 앞 절에서 설명한 시뮬레이터의 (1)~(9) 각 단계에 대한 수행 시간(C_i, C_m)과 시뮬레이터의 한 주기 수행 시간(C_{total})을 측정한다.

[표 1] FCHEV 시뮬레이터의 computation 수행 시간

C_i	MCU(1)	BPU(3)	MCU(5)	FCU(7)	MCU(9)
(ms)	1.192	0.035	1.066	0.037	2.139

[표 2] FCHEV 시뮬레이터의 communication 수행 시간

C_m	M_to_B(2)	B_to_M(4)	M_to_F(6)	F_to_M(8)
(ms)	0.128	0.126	0.130	0.126

단계 (2),(4),(6),(8)의 communication 수행 시간(C_m)의 오차는 frame overheads, data contents, extra stuff bits 그리고 수행 시간 측정 카운터의 오차(1.6us sampling)에 기인한다.

실험 2에서는 시뮬레이터의 지터를 고려하지 않은 시뮬레이터 sampling 시간을 실험 1에서 측정한 한 주기 수행 시간($C_{m_{network}}$)보다 크게 결정했다. 실험 2에서는 시뮬레이터의 샘플링 시간을 4.5ms부터 0.5ms단위로 증가시켜 시뮬레이터를 구동하였다. 샘플링 시간을 6.5ms로 설정하여 수행하였을 때부터, 시뮬레이터가 정상 동작함을 확인할 수 있었다.

실험 2를 통하여 시뮬레이터의 전체 지터(J_{total})는 1.581ms로 도출되었다. 홀리스틱 시간 해석을 통해 FCHEV 시뮬레이터 전체의 WCRT($R_{end-to-end}$)는 식 (13)과 같다.

$$R_{end-to-end} = C_{i_{total}} + w_{i_{total}} + C_{m_{total}} + w_{m_{total}} + J_{total} \quad (13)$$

식 (13)에서의 $w_{i_{total}}$는 각 CPU 노드의 computation 대기 시간들의 합이고, $w_{m_{total}}$은 각 CPU 노드간의 communication 대기 시간들의 합이다. 실험에서 구현한 시뮬레이터 시스템은 한 노드안의 수행 태스크들이 순차적으로 수행되고, 태스크 시작은 CPU의 동작 주파수 단위이므로 $w_{i_{total}}$는 0으로 가정하였다. 또한 시뮬레이터의 각 노드들은 고정된 우선순위를 가지고 있고, 순차적으로 수행되므로 전송되는 메시지가 다른 ID를 가진 메시지의 의해 대기 시간이 없으므로 $w_{m_{total}}$은 0으로 가정하였다. 따라서 식 (13)을 다시 정리하면 식 (14)와 같다.

$$R_{end} = C_{i_{total}} + C_{m_{total}} + J_{total} \quad (14)$$

실험 1에 의하여 시뮬레이터의 computation 전체 수행 시간 $C_{i_{total}}$는 4.469ms이다. 실험 1에서 각각의 C_m값들은 실험을 통해서 구한 값이므로, 식 (10)을 통해 $C_{m_{total}}$을 다시 구하면 0.524ms (0.131x4)가 된다. 여기서 J_{total}은 앞의 실험을 통하여 가정하였으며, 이들 값을 통하여 구한 시뮬레이터 시스템 전체의 WCRT ($R_{end-to-end}$)는 6.574ms 이다.

4. 결론

본 연구에서는 TI DSP 보드 3장을 CAN으로 연결시켜 FCEHV 시뮬레이터 시스템을 구현하였으며 각 서브시스템에서 코드수행시간 및 통신시간을 조사하였다. 실제 실험 결과로부터, CAN을 기반으로 연결된 차량내 네트워크 제어 시스템에서 홀리스틱 해석기법으로 구한 WCRT를 고려하여 실제로 구현했을 때에도 예상대로 동작된다는 것을 알 수 있었다. 향후 이 연구 결과는 우선순위가 높은 다른 노드를 연결했을 때도 마감시간을 만족하는 FCHEV 시뮬레이션 시스템 개발에 확장 적용될 예정이다.

참 고 문 헌

[1] K. Y. Yi, "An implementation of the position controller for multiple motors using CAN," *Trans. KIEE*, Vol. 51, No. 2, pp.55-60, 2002.

[2] W. H. Kim and S. H. Hong, "A study on the implementation of CAN in the distributed control system of power plant, "*Trans. KIEE*, Vol. 48, No. 6, pp.760-772, 1999.

[3] H. S. Choi and J. M. Lee, "A distributed precedence queue mechanism to assign efficient bandwidth in CAN networks," *Journal of Control, Automation and Systems Engineering*, Vol. 10, No. 11, pp.1058-1064, 2004.

[4] 신민석, 이우택, 선우 명호, "CAN을 이용한 차체 네트워크 시스템에 대한 Holistic 스케줄링 해석" 한국자동차공학회논문집 제10권 제5호.

[5] M. Joseph and P. K. Pandya, "Finding response times in a real time system, "*The Computer Journal*, Vol. 29, No. 5, 1986.

[6] K. W. Tindell and J. Clark, "Holistic schedulability analysis for distributed hard real-time systems." *Microprocessing and Microprogramming*, Vol. 40, pp.117-134, 1994.

[7] K. W. Tindell, H. Hansson, and A. J. Wellings, "Analysing real-time communications: controller area network (CAN)," *Proc. of the Real-Time Systems Symposium*, pp.259-263, 1994

[8] K. W. Tindell, H. Hansson, and A. J. Wellings, "Calculating controller area network(CAN)message response times," *Control Engineering Practice*, Vol. 3, No. 8, pp.1163-1169, 1995.

[9] H. Lonn and J. Axelsson, "A comparison of fixed - priority and static cyclic scheduling for distributed automotive control applications," *Proc. of the 11th Euromicro Conf. on Real-Time Systems*, pp. 142-149, 1999.

[10] N. S. Lee, S. Y. Shim, H. S. Ahn, J. Y. Choi, I. Choy, and D. H. Kim, "Modeling and an efficient combined control strategy for fuel cell electric vehicles," *Proc. of ICCAS*, Bangkok, 2004.