

CAN 네트워크상 시스템 분석을 위한 시물레이션 알고리즘 개발 및 Fault Test 방법

강호석*, 최경희*, 정기현**, 김상중***

*아주대학교 정보통신전문대학원

**아주대학교 전자공학부

***계명문화대학

e-mail : inspector@freechal.comr

Development of a Simulation algorithm AND Fault Test Methodology on CAN

Ho-Suk Kang*, Kyung-Hee Choi*, Gi-Hyun Jung**, Sang-jung Kim***

*Graduate School of Information and Communication, Ajou University

**Dept. of Electronics Engineering, Ajou University

*** Keimyung University

요 약

오늘날 기술의 발전으로 차량 시스템에 대한 고객의 요구가 다양해져 차량내의 시스템이 복잡해지고 있다. 이에 따라, 배선양을 줄이기 위해 Controller Area Network(CAN)이 개발되었다. 그러나 CAN 을 사용한 시스템이 제대로 동작하는지 또는 안전한지에 대한 안전성 문제가 중요시되면서 시스템을 실제로 어떠한 CAN 네트워크에 연결할 경우 그 시스템이 제대로 동작하는지는 자체 테스트와 효과적인 테스트가 필요하다. 본 논문은 CAN 네트워크 통신상태를 자동 테스트 할 수 있는 시물레이션 알고리즘을 소개하며 또한 fault packet 을 생성하여 CAN 네트워크를 테스트하였다.

키워드:자동차 네트워크(CAN)

1. 서론

최근 자동차에 대한 연비 및 배기에 대한 규제의 강화와 운전자 편의성 증대 욕구에 따라 자동차를 구성하는 전자 시스템 적용이 증가하고 있다. 따라서 차량에 장착된 각종 전자 시스템을 연계시키는 와이어 하네스(wire harness)도 필연적으로 증가하고 있다. 복잡하고 상대적으로 부피가 큰 와이어 하네스는 유지, 보수가 어려울 뿐 아니라 신뢰성 확보에도 문제가 있으며 진단 기능의 구현이 어렵다. 이와 같은 문제점을 근본적으로 해결하기 위해서 사용되기 시작한 것이 CAN 이다. 이는 종래의 1 대 1 네트워크가 아니라 버스형 네트워크 시스템으로 데이터 공유가 가능해질 뿐만 아니라 배선이 깔끔해져 중량을 대폭 삭감할 수 있게 되었다. 그러나 네트워크를 적용하게 되면 자료 전송에 있어서 자료의 손실이 야기될 수 있으며 부하로 인하여 제어시스템의 성능이 저하를 가져온다. 또한 운용미숙으로 예기치 않은 조작이 일어날 위험에

많이 노출되어 있다. 이러한 이유로 제어시스템의 신뢰성 및 오작동에 대한 테스트가 필요하다. 이를 위하여, 실제 시스템을 구성하기 전에 분석 및 평가할 수 있는 CAN 의 자료 전송 시물레이션 알고리즘 및 이를 이용한 시물레이터를 개발과 fault packet 을 생성하고 이를 이용하여 CAN 네트워크를 테스트 하였다.

2. CAN 테스트 시물레이션을 위한 알고리즘 개발

2.1 CAN 및 자동차 네트워크의 개요

CAN 은 분산 전자제어 시스템 상에서 자료를 전달하기 위한 직렬 통신 채널의 하나로 독일의 Bosch 에서 처음 제안하였다.[1] 신호 전달을 위해 두 신호 라인의 전압차를 이용하기 때문에 각종 전기적 노이즈에 강한 내성을 가진다. 또한 강력한 오류 검출 능력과, 최대 1MSbps 의 빠른 전송 속도를 갖고 있기 때문

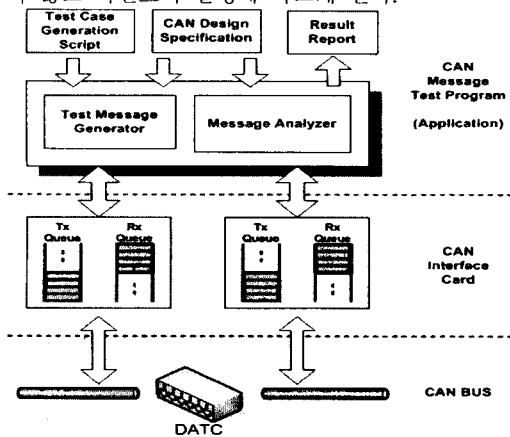
에 자동차에의 적용이 늘어나고 있다. CAN message 형식은 데이터 요청을 위한 remote frame, 그리고 오류 발생을 알리기 위한 error frame 등이 있으며 이는 다시 표준형과 확장형의 두 가지로 나뉜다. 표준형과 확장형의 차이점은 arbitration field 를 사용한다. Arbitration field 는 또한 각 message 의 ID 로도 사용된다. ID 는 전체 네트워크에서 고유하게 할당된 것으로 숫자가 작을수록 높은 우선순위를 갖는다. Message 의 DATA field 는 Signal 을 가지고 있으며 Signal Range 를 가지고 있다. 이 Range 는 0 에서 $2^x - 1$ 까지 integer number 을 가진다. 그리고 사용되는 범위를 valid signal range 라고 부른다. (x 는 각각의 Message 마다 Spec 에 정의된 Number of bit 이다.)

네트워크에서 자료 전송 개시, 혹은 오류 발생등과 같이 사건 처리를 요구하게 되는 것을 이벤트(event)라 정의한다.[2] 자동차 네트워크 이벤트는 주기가 일정한 것(주기적 이벤트)과 그렇지 않은 것(간헐적 이벤트)으로 구분할 수 있다.

자동차 네트워크 구성은 차내온도, 스로틀 위치, 각 조작 스위치의 상태 등과 같은 자료들을 샘플주기 등을 기준으로 정리하고 각각에 고유한 ID 를 부여되며 MM CAN(멀티미디어 부분)과 BODY CAN(자동차 부품속품 부분) 사이의 Gateway 기능으로 DATC (듀얼-존 전자동 온도 제어기)가 구성된다. 이는 각 CAN 네트워크로부터 받은 message 들의 signal 을 새로 조합하여 다른 쪽 CAN 네트워크로 전송하는 역할을 한다.

2.2 시뮬레이션 알고리즘

시뮬레이션 알고리즘은 기본적으로 양방향 송수신 구조인 queue 의 모델과 시간순서에 의해 message 정보가 저장된 Log 파일을 바탕으로 하는 이산 시간 시뮬레이션 이론을 사용하였다. 시간의 흐름은 연속적이지만 알고 이벤트의 발생에 따라게 된다.

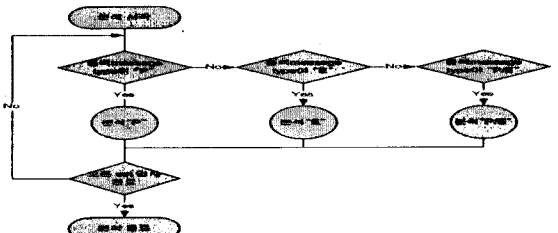


(그림 1) 송수신 Queueing 이론

(그림 1) 에서 알 수 있듯이 두 가지의 queue 를 사용하였다. Tx Queue 는 각 컨트롤러 내부의 버퍼로써 송신될 메시지를 저장하며 Rx Queue 는 각 컨트롤러 내부의 버퍼로써 수신될 메시지를 저장된다. 따라서 각

컨트롤러 모두에 message 을 송수신이 가능하게 되어 양방향 인터페이스 테스트 구조로 모델링 된다. 여기서 Queue 의 크기를 임의로 설정할 수 있도록 하여 사용하는 컨트롤러의 사양에 맞추어 적절히 조절할 수 있다.

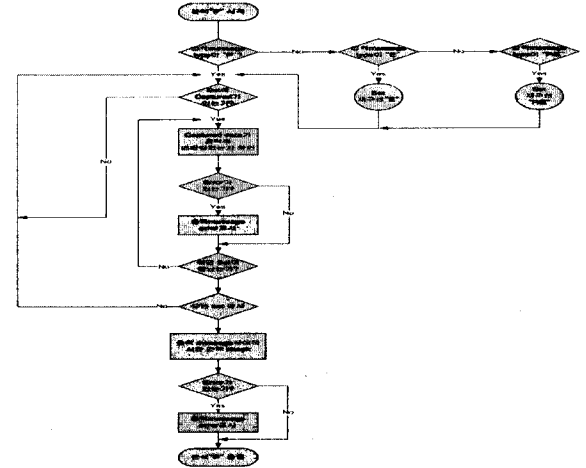
그리고 message 을 분석하기 위해서 송수신된 message 정보가 저장된 Log 파일을 사용하였다. message 분석 방법은 Log 파일을 읽어 들여 ID 별로 message 들을 추출하여 ID 별 Message-Set 파일들을 구성하며 이렇게 구성된 파일을 이용하여 타입별 message 분석 알고리즘을 통해 DATC 가 올바르게 반응하였는지에 대한 분석을 하게 된다. 여기서 ID 별 Message-Set 파일은 Log 파일에 있는 송수신 message 와 그것의 결과에 해당하는 수신 message 집합들이다.



(그림 2) 메시지 분석 알고리즘

타입별 분석방법은 크게 주기성 이벤트, 간헐적 이벤트 타입으로 구분된다.

주기성 이벤트일 경우 200ms 간격으로 message 가 발생하기 때문에 전송 message 사이의 간격이 200ms 인지를 확인된다 즉 전송 message 의 Data 가 수신 message 의 Data 에 반영이 되었는지 확인후 수신 message 간에 200ms 확인한다. 이 중 하나의 message 에 error 가 있을 경우 error 가 표시된다.



(그림 3) 전송타입 분석 알고리즘

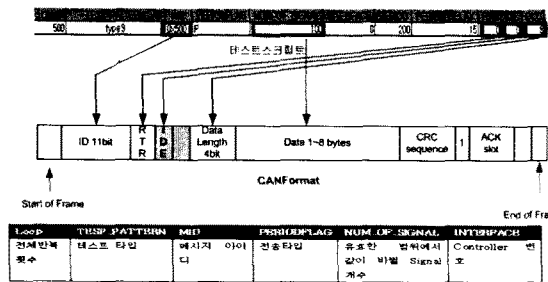
간헐적 이벤트일 경우는 전송 message 의 Data 가 수신 message 의 Data 에 반영이 되었는지 확인 후 수신 message 간에 40ms 간격으로 3 회 message 가 발생했는지 확인된다.

3. 시뮬레이터 구성

이 툴은 CANTester 라고 불리어지며, 시스템의 CAN 인터페이스 동작을 테스트할 수 있는 목적으로 제작되었다. 이 툴의 구성은 CAN Message 생성 및 전송, 설정한 message 스케줄링, 송수신 정보저장 및 분석 부분으로 되어진다.

3.1 CAN message 생성 및 전송

CANTester 는 CAN Format 이 정의된 테스트 스크립트를 통해 single-packet instance 들을 생성시킨다. 이것은 다음과 같은 방법으로 진행 된다. 첫 번째로 메시지 공간이 할당된다. 그리고 나서 할당된 공간에 테스트 스크립트에 의한 조건들이 CAN Format 헤더에 적용된다. 다음은 시그널 값의 범위가 Specification 에 의해 인정되는 경우 그 값은 유효한 값의 범위로부터 랜덤하게 선택되어 Packet Payload 에 복사 되어진다. 마지막으로 Checksums 계산되고 전송타입에 맞게 패킷이 IXXAT library 를 통해 전송된다. 예를 들어 다음과 같은 테스트 스크립트가 있다



(그림 4) 메시지 생성과정

500 개의 0x500 message 가 주기적으로 시그널 값이 바뀌면서 생성되며 컨트롤로 0 번을 통해 버스에 전송된다.

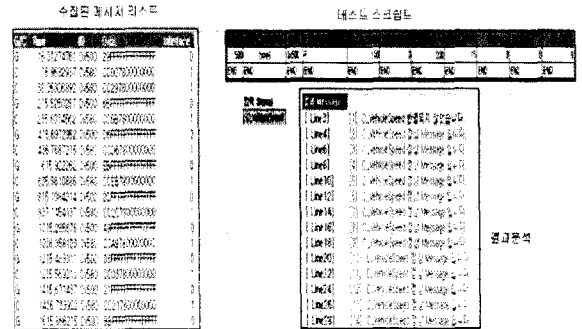
3.2 Message 스케줄링

스케줄링은 메시지 리스트의 각 메시지의 전송타입, 시작시간 주기를 바탕으로 동작한다. 메시지 전송타입이 주기적 이벤트이면 이벤트 발생시점의 message 가 주기시간으로 순차적으로 더해진다. 즉 주기적으로 message 을 전송되다가 이벤트가 발생하면 Signal 값이 바뀐 message 을 주기적으로 전송하게 된다. 만약 간헐적 이벤트이면 이벤트 발생시점에 시간에 랜덤값만 더해지게 된다. 이벤트 시간이 계산된 메시지들은 시간순서로 정렬되어 파일에 저장된다. 파일에 저장된 시간은 타이머에 셋팅되면 타이머가 파일에 저장된 시간순서로 전송합수를 인터럽트 시킨다.

3.3 송수신한 message 분석

CANTester 의 기본기능은 DATC 에 송수신된 각각의 패킷을 분석하고 설정된 규칙에 맞게 송수신 되었는지 찾기 위한 것이다. 이 규칙은 CAN Design Spec 에 의해 송수신된 메시지의 시그널 위치변화와 전송타입을 명시하고 있다. 즉 주어진 조건을 모두 만족하면 메시지가 정상 처리되었음을 기록한다. 그러므로 메

시지 분석은 메시지 필드별 parsing 을 통해 송수신된 메시지의 각 필드가 조건에 만족하는지 판단하며 전송타입의 구분을 포함한다. (그림 5)은 CANTester 에 의해 출력된 결과분석 파일이다.



(그림 5) 결과분석파일

4. 실험 방법

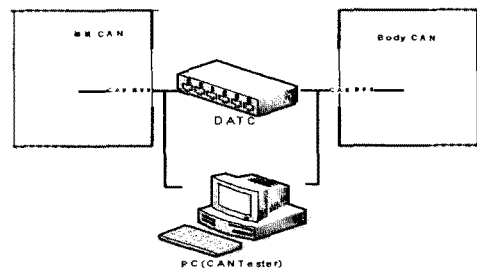
이 장은 시뮬레이션 결과검증, DATC 오작동 테스트 위하여 실제 CAN Network 를 구성하여 DATC 장치를 측정하여 보았다.

4.1 Fault Injection 방법

장비에 오작동으로 인하여 전달되는 메시지에는 다양한 형태의 fault 가 주입될 수 있다. 본 논문에서는 syntactic fault 를 적용하여 장비를 테스트 하였다.[4] Syntactic fault 는 메시지의 구조나 기술 방법, 값의 형태나 범위상에 fault 가 발생된 것을 가리킨다. 즉 specification 에서 허용되지 않는 값의 사용으로 기인한 fault 를 생성한다.

4.1.1 테스트 환경

테스트는 다음 같이 3 가지 환경으로 구성된다. 첫 번째는 DATC 장치 없이 CAN message 가 버스에 전송되는 시스템 환경에서 측정되었으며 두 번째는 정상적인 message 가 DATC 에 전송되는 시스템 환경에 측정되었으며 마지막으로 Fault message 가 DATC 에 전송되는 시스템 환경에서 측정되었다.



(그림 6) 테스트 수행 환경

여기서 CAN Tester 는 Window XP 하에 CAN Specification 에 명시된 제약(constraint)에 만족하는

CAN Message 를 발생시키며 Fault message 는 CAN Specification 과 다르게 CAN Frame 필드에 Fault 를 주입하였다. 이 실험에서 CANTester 에 의해 생성된 message 가 제대로 전송되는지 확인하기 위해 DATC 장치 없이 버스에 전송되는 message 가 검사 되었으며, 이 실험의 결과는 CAN FRAME 필드에 의해 분류 되어진다.

이 실험은 CAN ID 가 0x83,0x103,0x502 인 메시지를 입력 메시지로 사용하였으며 [표 2]의 조건으로 Fault injection 실험을 하였다

<표 2> Syntactic fault 타입

항목	정상	Fault
IDE	ID field의 크기를 0(11bit)으로 고정	ID field의 크기를 1(29bit)으로 고정
Signal	Spec에 정의된 범위	Spec에 정의되지 않은 범위
ID	Spec에 정의된 ID들	Spec에 정의되지 않은 ID들
P./E type	200ms 간격의 전송	600ms 또는 1s이상 전송 없음
E type	3회 40ms간격의 전송	4회 이상 40ms 간격의 전송
전송 Time	같은 ID Message는 20ms 이상 범위에서 전송	같은 ID Message는 20ms 이하 범위에서 전송

5. 실험결과

이 장은 CANTester 틀에 의해 테스트 되었던 DATC 의 반응 결과를 보여준다.

5.1 정상 Packet 에 대한 반응실험

이 실험은 Specification 에 맞는 CAN Message 를 DATC 에 전송하여 DATC 가 Specification 에 맞게 CAN message 을 처리하는지에 대한 반응을 확인하기 위해서이다. <표 4>와 같이 다음과 같은 입력 메시지를 DATC 에 전송한 결과 장비는 CAN specification 에 준하는 응답을 전송하였다

<표 4>시뮬레이션 결과

입력 Message ID	출력 Message	Response	Result
0x83	0x5	CAN specification에 준하여 출력 Message가 나왔다.	good
0x103	0x81		
0x502	0x82		

5.2 Fault Injection 반응실험

메시지의 각 Field 마다 Syntactic fault 를 발생시킨 fault packet 들로 테스트를 수행하였을 때, 장비의 오작동 여부를 테스트하였다.<표 5>는 DATC 장비에 대하여 syntactic fault 를 포함하는 fault packet 으로 장비를 테스트한 결과이다. <표 5>에 있는 각 Fault type 별로 DATC 장비를 테스트한 결과를 정리 한 것이다. 'Warning'이라 표시된 부분은 비록 CAN 프로토콜에 어긋나지만 장비가 지속적으로 정상적인 서비스를 제공할 수 있었다는 것을 가리킨다. 반면 'Failure'는 테스트 장비가 CAN 프로토콜을 사용할 수 없는 심각한 오류를 말한다

<표 5> fault 주입 결과

Fault Type	입력 Message ID	출력 Message ID	Response	Result
F1	0x83	없음	응답없음	Failure
	0x103	없음		
	0x502	없음		
F2	0x83	0x5	Invalid로 표시된 signal 값을 가진 Message 발생	
	0x103	0x81		
	0x502	0x82		
F3	0x83	없음	Fault ID를 가진 Message를 Drop으로 응답이 없음	
	0x103	없음		
	0x502	없음		
F4	0x83	0x5	포수에 상관없이 0x5 Message를 40ms 간격으로 3회 출력	
F5	0x103	0x81	비주기적인 입력 Message와 상관이 정상적으로 작동	
	0x502	0x82		
F6	0x83	0x5	입력 메시지에 반응하여 출력되는 메시지가 나오지 않는 경우도 있음	Warning
	0x103	0x81		
	0x502	0x82		

테스트 결과, fault packet 을 사용하여 테스트를 하였을 에도 장비가 이를 인지거나 Specification 에 준하게 처리하는 등의 동작을 보이는 경우<표 5>의 Result 부분에 표시하지 않았다. DATC 장비에 대하여 Syntactic fault 테스트를 수행한 결과 테스트에 사용된 fault packet 에 대하여 테스트 장비는 타당하게 처리하지 못하는 상황이 존재하였다. Fault type F1 의 경우처럼 Extended Format(확장형) 모드로 CAN 네트워크를 설정하여 DATC 를 연결하였을 때 DATC 는 자체 생성하여 주기적으로 발생시키는 message 가 나오지 않았으며, 또한 입력 message 에 대한 bridge 기능도 상실 되었다.

5. 결론

자동차 네트워크로 CAN 을 사용하는 경우에 대하여 시뮬레이션 알고리즘과 시뮬레이터를 개발하였다. 위 알고리즘은 주기적인 이벤트는 물론 간헐적인 이벤트도 처리할 수 있으며, 네트워크의 노드간의 메시지 전송상태를 알 수 있었다. 따라서 네트워크를 설계할 때, 시스템 구축에 앞서 손쉽게 네트워크의 구조 변경 테스트 및 최적화가 가능하다.또한 Syntactic fault 를 테스트함에 있어 장비가 주고 받는 메시지 중 어떤 메시지가 취약점을 지니고,어떤 field 가 취약점을 지니며, 어떤 형태의 fault 에 취약점을 가지는지 파악 할 수 있었다..

6.참고문헌

[1]Wolfhard Lawrenz," CAN System Engineering",Springer,1997
 [2]Bosh,CAN Specification Version 2.0", Bosch 1991
 [3]K.Estchber " Controller Area Network Basics,Protocols,Chips and Application IXXAT Automation GmbH,Weingarten,Germany,2001
 [4]Lijun Shan, Hong Zhu, " Test are generation: Testing software modeling tool using data mutation",AST,2006
 [5]http://www.can-cia.org,2006.organization CAN in Automation
 [6]http://www.ixxat.de.2005 IXXAT Automation GmbH
 [7]양성모,"자동차 네트워크 시스템 분석을 위한 시뮬레이션 알고리즘 개발에 관한 연구",1999