특집 05

자동차 내장형 시스템을 위한 소프트웨어 신뢰성

고요한 · 이경우 (연세대학교)

목 차 »

1. 서 론

한·미 FTA를 추진하였던 원동력 중 하나는 한국의 건실한 자동차 산업이었다. 세계자동차공업협회 통계에 따르면 한국의 자동차 생산량은 450만대의 규모를 자랑하며 전 세계 완성차 자동차 시장에서 5위를 기록하고 있다[1]. 또한, 자동차로 인해 생산되는 금액이 총 GDP 대비 3.3%를 기록하며 단일 산업에서 가장 큰 비중을 차지할 정도로 한국에 있어서는 가장 견실한 산업 중 하나로 평가받고 있다[35].

그러나 최근 이러한 한국의 자동차 산업이 위기를 맞고 있다. 2013년을 기준으로 한국의 생산량이 0.9% 줄어드는 동안 브라질과 멕시코 등 신흥 자동차 강국들이 생산량을 오히려 늘리며 추격하고 있다. 또한, 중국과 미국 등 강대국에 의한 생산 쏠림 현상이 발생하면서 한국에 대한 위협은 날로 증가하는 추세이다.

과거 자동차 시장을 호령하였던 한국의 가격 경쟁력이 더 이상 날카로운 무기가 되지 않으며, 이제는 한국의 자동차 산업이 빠른 추종자(fast follower)를 넘어서서 업계 기술 선도자(first mover)로 진입해야만 하는 시기가 도래하였다. 이제는 '기술'로 대표되는 독일의 자동차 업계나, '성능'으로 대표되는 일본의 자동차 업계와 같이 한국의 자동차 업계 역시 그를 대표할 만한 키워드가 필요한 시대를 의미하는 것이다. 이러한 새로운 시대에 한국 자동차 업계의 해법은 무엇일까?

완전한 해법은 아니겠지만 하나의 가능성으로 제안된 것이 자동차 내장형 시스템의 소프트웨어이다. 여전히 자동차에서는 기계공학의 설계 등이 큰 비중을 차지하는 것은 사실이지만, 동시에 적지 않은 부분의 자동차의 문제가 소프트웨어로 넘어오게 되었다[5]. 현대의 자동차는 기계적인 메커니즘에 의해 제어되던 많은 부분이 이제는 엔

진뿐만 아니라 변속기, 에어백, 공기압 조절 등 많은 부분이 전자 제어 장치(electronic control unit)을 통해서 제어되고 있다. 뿐만 아니라 자동차가 제2의 집처럼 받아들여짐에 따라 인포테인먼트(infotainment) 소프트웨어 역시 날로 증가하고 있다.

초기 자동차 내장형 시스템의 소프트웨어가 가지는 숙제는 성능이었다. 즉, 어떻게 자동차 기계장치를 제어할지, 혹은 효율적인 주행을 도와주기 위한 소프트웨어의 기여 등을 고려하였다. 그리고 이러한 내장형 시스템의 소프트웨어는 기술이 발달함에 따라 자동차 한 대가 하나의 컴퓨터 혹은 그 이상의 컴퓨터와 맞먹을 정도의 놀라운 컴퓨팅 속도를 가지는 등 새로운 기술의 시대가 열리고 있다[5].

그러나 자동차 내장형 시스템의 소프트웨어는 이러한 장밋빛 미래만을 그리지는 않는다. 모든 소프트웨어는 버그를 가지기 마련인데, 자동차에서는 이러한 에러가 단순한 버그 정도가 아닌 사람의 목숨과 관련되었기 때문이다. 이에 따라 소프트웨어의 신뢰성이 중요한 문제로 대두되기 시작하였다. 또한, 토요타의 급발진 사태가 최근 한 내장형 시스템 자문회사에 의해 잘못된 소프트웨어가 원인으로 밝혀지면서 이러한 소프트웨어의 신뢰성은 점점 더 중요해지고 있다. 2003년 미국에서 리콜 된 자동차 중 1,950만대의 자동차는 자동차 내장형 시스템의 문제로 밝혀졌다. 또한, 토요타 3세대 프리우스 190만대가 소프트웨어 결함으로 인하여 리콜되는 일이 벌어지기도 하였다[2].

이처럼 자동차 기술 선진국인 미국과 일본의 자동차 시장 역시 소프트웨어 결함으로 인한 리콜을 피하지 못하였다는 것은 시사하는 바가 크다. 보편적인 시각에서는 소프트웨어의 신뢰성이 기술 선진국에서조차 해결하기 어려운 문제라는 것으로 생각할 수 있으나, 다른 시각에서는 리콜과 같은 재정 위기를 막을 수 있는 새로운 시장이기도 하다. 그렇기 때문에 자동차 선진국에 비해서 자동차 내장형 시스템의 소프트웨어에 대한 연구가 다소 늦었던 한국은 이러한 소프트웨어 결함을 극복할 수 있는 고신뢰 소프트웨어 기술이 필요할 것이며, 이러한 기술이 새로운 시장을 열 수 있을 것이다.

물론 이러한 자동차 내장형 시스템의 소프트웨어 신뢰성을 위한 연구가 전무했던 것은 아니다. 예를 들어, 브레이크를 전자화 기술로 제어하는 시스템의 신뢰성을 위한 소프트웨어 방법론이 제안되기도 하였고[3], 자동차 내장형 시스템의 소프트웨어를 통합 및 관리하기 위한 다양한 표준이 제정되기도 하였다[4].

본 논문에서는 왜 자동차 내장형 시스템의 소프트웨어 신뢰성이 중요한지, 그리고 왜 이러한 신뢰성의 위협 요소가 증가하는지, 이러한 신뢰성의 문제로 발생한 토요타 급발진 사태를 2장에서 다룬다. 또한, 3장에서는 자동차 내장형 시스템의 신뢰성을 높이기 위한 기존 연구를 기술한다. 미래 기술의 발전과 더불어 예상되는 자동차의 신뢰성 문제를 4장에서 정리하고, 마지막으로 5장에서 논문을 마무리한다.

2. 자동차의 신뢰성 위협 요소 증가

2.1 왜 자동차의 신뢰성인가?

흔히 자동차의 신뢰성 문제는 기계 결함으로 인해서 발생하는 사고로 생각하기 쉽다. 그러나 최신의 고급 자동차에서 1억 줄이 넘는 소프트웨어가 쓰인지는 이미 오래이며, CPU에 해당하는 전자 제어 장치 역시 10개 이상이 사용되고 있다[5]. 즉, 자동차 내장형 시스템의 소프트웨어는 멀티코어와 같은 새로운 환경을 고려해야 하는 정

도로 점점 더 복잡해지고 있고, 게다가 다양한 기능을 수행해야 하면서 점점 더 규모조차 거대해지고 있다. 또한, 이러한 복잡한 전자 제어 장치 간을 연결하는 CAN(controller area network), LIN(local interconnect network), MOST(media oriented systems transport), FlexRay, Ethernet과 같은 네트워크 역시 아직 안정적인 모습을 보여준다고 단정하기 어렵다[6].

그렇다면 이러한 자동차 내장형 시스템의 소프트웨어 신뢰성이 중요한 이유는 무엇일까? 먼저, 자동차는 안전에 관련해서 치명적인 내장형 시스템(safety-critical systems)이기 때문이다[7]. 예를 들어, 스마트폰에서 멀티미디어 자료를 보는 도중에 에러가 발생하여 화질이 조금 떨어진다고 해도 사용자는 큰 불편을 느끼지 못하거나 심지어 육안으로는 눈치를 채지 못할 수도 있다. 즉, 이러한 스마트폰과 같은 내장형 시스템은 사용자의 안전에 영향을 주는 치명적인 내장형 시스템이 아니다.

그러나 자동차의 경우는 내장형 시스템 소프트웨어의 많은 부분이 치명적이다. 예를 들어, 잠김방지제동장치(ABS: anti-lock braking system)를 제어하는 소프트웨어에 에러가 발생하였다고 가정하자. 이러한 경우 사용자가 급작스럽게 브레이크를 조작할 경우 차체 쏠림 현상이 발생하여 운전자의 목숨이 위협받는 사고가 발생할 수 있다. 다시 말해 빠른 성능보다는 언제나 신뢰성을 유지해주는 보호 방법 등이 요구된다. 물론 자동차 역시 엔터테인먼트적인 요소가 가미된 인포테인먼트 서비스는 이러한 치명적인 시스템이라고 볼 수는 없다. 그렇기 때문에 다양한 자동차 내장형 시스템의 소프트웨어에 대해서도 천편일률적인 보호 방법이 아니라 각 소프트웨어의 특성을 반영한 최적화가 요구된다.

이러한 자동차 내장형 시스템의 소프트웨어 신뢰성의 문제를 수치적으로 고려해보자[8]. 2013년 국도교통부 통계에 따르면 한국에 총 1,940만대에 달하는 자동차가 등록되어 있다. 편의를 위해 모든 차량은 단 1개의 내장형 프로세서를 사용하여 ABS를 구현했다고 가정하자. 또한, 이러한 ABS는 에러가 거의 발생하지 않아 1,000년에 한 번 꼴로 에러를 발생시키는 장치라고 한다면 어떨까? 일견 생각하기에 1,000년에 한번 에러를 발생시키는 시스템은 굉장히 안정적으로 느껴진다. 그러나 2천만대에 가까운 등록 차량을 놓친 안일한 분석이다. 이는 이러한 보수적인 가정에도 하루에 5.3대 차량의 ABS에서 에러가 발생한다는 의미이다. 게다가 이러한 5.3대 역시 끝이 아니다. 한국에 등록된 자동차가 2천만대 일뿐 세계 시장에서 보면 더 많은 자동차가 있음은 물론, ABS 역시 단 하나의 프로세서의 구현되지 않는다. 또한, 1,000년에 한 번 정도 발생한다는 에러의 발생 확률 역시 기술의 소형화, 경량화에 따라 그 빈도가 늘어가고 있다[17].

자동차 내장형 시스템 소프트웨어의 신뢰성이 중요한 두 번째 이유는 자동차 내의 환경과 관계가 깊다. 일반적으로 내장형 시스템이 개발되면 모든 검증은 극단적이지 않은 일반적인 실험실 환경에서 이루어지는 경우가 많다. 그러나 자동차 내부에서 동작하는 시스템은 실험실과 같은 일반적인 환경과는 전혀 다른 다양하고 또한 극단적인 형태를 띠고 있는 경우가 많기 때문에 실험실에서 제대로 동작하던 소프트웨어가 자동차 내부에서는 신뢰성을 보장하지 못하는 경우가 많다[6].

예를 들어, 자동차의 내부 엔진의 경우 극단적인 경우 100℃를 상회하는 경우가 존재한다. 이러한 경우 자동차 내장형 시스템의 하드웨어에 일시적인 에러(soft error)가 발생하는 확률이 급증하게 된다. S. Jagannathan[9]의 연구에 따르면 25℃와 120℃에서 플립플롭의 일시적인 에러의

비율을 비교해보면 최대 3배에 가까운 차이가 발생한다. 즉, 상온에서 제대로 동작하는 내장형 시스템 소프트웨어라고 하더라도 자동차 내부의 극심한 열이 발생하는 환경에서는 제대로 동작하는지 여부를 확신할 수 없는 없다.

또한, 자동차 장치의 특성상 전원을 쉽사리 켜고 끌 수 없다는 점도 신뢰성과 연관이 깊다. 스마트폰과 같은 보통의 치명적이지 않은 내장형 시스템에 대해서 생각해보자. 이러한 경우 에러가 발생한다고 해도 재부팅 등을 통해서 에러를 제거할 수 있는 경우가 많다. 그러나 자동차의 경우 운행하는 도중에 재부팅을 하는 행동 등은 심각한 신뢰성 문제를 야기할 수 있다. 뿐만 아니라 비단 주행 중이 아니더라도 배터리를 관리하는 시스템 등은 자동차의 수명인 15년 이상 동안 꺼지지 않고 동작을 해야 하기 때문에 에러의 발생 확률이 필연적으로 높을 수밖에 없다.

마지막으로 자동차라는 장치 특성상 에러가 발생하고, 그 후 처리에 고비용이 발생한다. 2012년 포드 사의 자동차 제어 시스템인 MyFord Touch에서 소비자의 불만이 발생하자 시스템을 업데이트하기 위해 무려 30만 개에 이르는 USB를 각 운전자의 집에 배송하는 일이 있었다[10]. 즉, 차량용 소프트웨어의 경우 신뢰성 문제가 발생하고, 그 발견이 늦는다면 큰 경제적 피해를 일으킬 수 있다. 또한, 대부분의 프로그래밍에서 에러의 발생은 코딩 단계와 같은 초기 단계에서 발생하므로[11], 신뢰할 수 있는 시스템이 구축되어 있다면 사후 처리에 드는 고비용을 최소화할 수 있을 것이다.

2.2 전자화 기술은 자동차에 어떤 영향을 미치는가?

자동자의 시스템이 점점 전자화(X-by-wire) 되는 것은 에러의 발생 확률을 점차 증가시키고 있으며, 이는 자동차의 새로운 신뢰성 문제를 불러일으켰다. 전자화란 기존의 기계적인 메커니즘을 통해 유압 혹은 공기압 등으로 조절하던 자동차를 전선을 통해서 제어하는 기술을 의미한다[34]. 전자화 기술은 하나의 기술을 의미하는 것이 아니라 자동차 컨트롤을 위한 스로틀 전자화(throttle-by-wire)나 브레이크 전자화(brake-by-wire) 등을 총칭한 기술을 의미하며, 메르세데스 벤츠나 토요타의 회사 등에서는 이미 수년전부터 널리 쓰이고 있는 기술이다.

왜 이러한 전자화 기술이 신뢰성에 악영향을 끼칠 수 있는 것일까? 이는 자동차의 특성이 일반적인 내장형 시스템과 점점 더 같아지는 점에서 그 이유를 찾을 수 있다[6]. 현재의 자동차 집적회로는 대부분 65nm 공정을 사용하고 있다[12,13]. 그러나 기술의 빠른 발전과 더불어 공정은 이미 40nm를, 혹은 심지어 그 보다 더욱 발전된 소형화 공정을 향하고 있다[14,15]. 이러한 기술의 소형화와 첨단화는 전력 소비와 직결된다. 예를 들어, 이미 소형화된 스마트폰의 경우는 이러한 전력의 소비를 줄이기 위해서 동적 전력 조절이나 주파수 조절 등 전력 최적화 기법을 사용하고 있다[16]. 그리고 이러한 동적 전력/주파수 조절은 예기치 못했던 일시적인 에러를 증가시키는 등 신뢰성 문제를 야기하고 있다[17]. 현재까지 자동차에서는 안전상의 이유를 들어서 이러한 실시간 전력 관리 기법을 사용하지는 않고 있다. 그러나 자동차 집적 회로 기술의 소형화와 더불어 전력 관리 기법이 사용될 것은 자명하므로, 이에 따른 신뢰성 문제를 해결할 방법이 요구될 것이다.

또한, 이러한 치명적인 자동차의 기능에도 전자화 기술이 적용된다는 점에서도 그 이유를 찾을 수 있다. 현재 전자화 기술은 페달이나 레버 등과 같은 물리적인 조작을 통해서가 아니라 버

튼 등을 통한 주차 브레이크를 동작시키는 주차 전자화 기술(park-by-wire 혹은 shift-by-wire) 등에 고급차를 대상으로 조금씩 적용되고 있다[18]. 이러한 주차 브레이크를 동작시키는 전자화 기술의 경우 에러가 발생한다고 해도 치명적인 인명사고와 같은 큰 사고를 일으킨다고 말할 수는 없다.

그러나 스티어링 전자화(steering-by-wire) 혹은 브레이크 전자화 기술의 경우는 어떨까? 즉, 의도치 않은 스티어링이 발생하거나 의도한 경우에 브레이크가 동작하지 않는다면 어떨까? 그뿐만 아니라 향후 전자화 기술은 안전과 관련한 위급상황 회피 기술 등이 적용될 예정이다. 회피 기술은 자동차의 각종 센서를 이용하여 자동차의 상태를 체크한 후 긴급 정지를 하는 방식으로 구현이 예상된다. 이러한 전자화 기술의 경우 이를 제어하는 소프트웨어의 신뢰성이 확보되지 않는다면 구현 자체의 의미가 사라지는데, 본 기능의 신뢰성 문제는 전통적 자동차에서의 기계적인 결함보다는 전자공학적인 혹은 컴퓨터과학적인 문제가 예상된다.

2.3 토요타 급발진의 원인은 무엇인가?

본 장에서는 자동차 내장형 시스템 소프트웨어의 신뢰성이 문제가 된 것을 실제 사례를 통해서 알아보자. 2009년부터 2010년까지 세계에서 가장 안전한 자동차중 하나로 손꼽히는 토요타 프리우스에서 급발진 사고가 보고되었다. 이른바 토요타 페달 게이트로 불리는 이 사건으로 토요타는 전 세계 1천만대 이상에 이르는 차량을 리콜 조치하였고, 이로 인해 신용뿐만 아니라 실제 기업 이득에서도 엄청난 손해를 입었다.

본 사태가 심각해지자 미국 항공우주국(NASA: national aeronautics and space administration), 고속도로교통안전국(NHTSA: national highway traffic safety administration)에서는 10개월에 걸친 리콜 차량 검사를 실시하였다[19]. NASA와 NHTSA는 일각에서 제기된 전자제어장치의 문제로 인한 급발진 가능성이 전혀 없는 것은 아니나, 가속페달과 느슨한 바닥매트를 원인으로 발표하였다. 이는 전자제어장치의 문제가 아니라는 토요타의 주장이 힘을 얻는 계기가 되었으며, 이를 계기로 토요타의 주식이 4% 오르는 해프닝이 일어나기도 하였다. 즉, 해당 보고서에 따르면 자동차의 결함이 일어나는 이유는 여전히 기계적인 결함에 의존하고 있는 것처럼 보인다.

그러나 2014년 내장형 시스템 전문가인 바(Barr) 그룹은 토요타 급발진 사태의 원인이 소프트웨어 결함임을 밝혀내고, 실험으로 증명을 하며 사태는 새로운 국면을 맞았다[20]. 해당 보고서를 통해서 자동차의 신뢰성에 영향을 주는 요소에 소프트웨어가 지목되었음은 물론 토요타가 12억 달러에 이르는 벌금에 합의함에 따라 그러한 소프트웨어 에러가 단순한 버그 수준이 아닌 치명적인 경제적 피해를 끼칠 수 있음이 다시 한 번 증명되었다. 그렇다면 도대체 소프트웨어의 어떤 부분이 급발진을 일으킨 것일까? 어떤 부분이 차량의 가속을 부추기는 스로틀을 동작시켰으며, 왜 시스템은 그 부분을 감지하지 못한 것일까?

먼저, 차량 내 하드웨어의 메모리 부분에 에러를 검출 수정할 수 있는 코드가 없었다. 치명적인 내장 시스템의 경우 메모리는 에러 정정 코드(error correction code)란 중복기법을 사용해서 메모리 내에서 발생하는 메모리에 저장된 값이 0에서 1로 바뀌거나 반대의 현상을 탐색 및 검출한다. 자동차의 경우는 대표적인 치명적인 내장형 시스템 중 하나이므로 만에 하나를 대비한 중복기법은 필수불가결하다[7]. NASA의 보고서에 따르면 토요타의 차량은 에러 정정 코드를 사용하였다고 되어있으나[19], 실제 조사 결과 잘못된 위

치에 적용되거나 심지어 사용되지 않은 부분도 발견되었다.

두 번째로 안전장치(fail-safe)가 부족하거나, 잘못된 방향으로 설정되었다. 일례로 토요타의 내장형 시스템에서 중요한 변수는 미러링(mirroring)을 통해서 다양한 곳에 저장되어, 설령 에러가 발생하더라도 복제된 값을 이용하여 복원할 수 있다고 믿어져왔다. 그러나 스로틀의 정도를 저장하는 전역변수도 미러링이 되지 않은 등 문제를 보였다. 또한, 시스템의 오동작을 감시하는 감시 하드웨어(watchdog)도 제대로 동작하지 않았음이 실험을 통하여 확인되었다.

세 번째로 스택의 오버플로우 현상을 들 수 있다. NASA의 보고서에 따르면 스택 오버플로우의 현상이 발생 될 가능성이 없는 것은 아니나 현재 프로그램에서는 절반 정도 스택이 사용되고 있으므로 이를 급발진의 원인이라고 말할 수 없다고 결론지었다. 그러나 추가 조사 결과 재귀 함수, 라이브러리 함수의 스택 사용량의 잘못된 추정, 운영체제의 문맥 전환을 고려하지 않은 점 등의 문제가 드러났다. 즉, 실시간으로 스택을 감시하는 체계에서 허점을 보였으며, 메모리의 잘못된 값이 저장되게 하고, 이로 인해 스로틀의 오동작을 야기하였다.

마지막으로 소프트웨어 자체에도 결함이 많았다. 먼저 토요타의 코드의 경우 데이터 흐름에서 지나치게 복잡하였다. 예를 들어, 토요타의 코드에서는 자그마치 11,000개 이상의 전역 변수를 선언하여 사용하고 있는데, 이는 예상치 못한 작용을 하거나, 무분별한 접근, 정확한 선언 범위의 혼란을 불러일으키는 등의 문제가 있어 적절치 못한 프로그래밍 방법이다[21]. 또한, 제어 흐름에서 있어서도 지나치게 복잡하였다. 소프트웨어의 순환 복잡도를 측정하는 단위인 시험 가능성(testability)에 따르면[22], 토요타의 함수는 '테스트가 불가능한' 수준이며, 스로틀의 정도를 조절하는 함수의 경우에는 '유지가 불가능한' 수준의 높은 복잡도를 보이기도 하였다.

이렇듯 토요타의 급발진 사례는 소프트웨어 결함이 자동차의 신뢰성에 영향을 줄 수 있음이 밝혀지고, 그 사실을 자동차 업체에서 인정한 첫 번째 사례였다. 그렇다면 자동차의 신뢰성을 소프트웨어 혹은 컴퓨터과학에서 본 시각은 어떠하였을지, 그리고 어떠한 연구들이 진행되고 있는지에 대해 다음 장에서 살펴보도록 하겠다.

3. 자동차 내장형 시스템 소프트웨어의 신뢰성 향상을 위한 현행 연구

3.1 각 구성요소의 소프트웨어를 어떻게 보호할까?

먼저 이번 장에서는 자동차 전자화 기술 중 브레이크 전자화 기술의 신뢰성을 다룬 논문을 살펴보겠다[3]. 해당 논문은 몇 가지 질문을 던지고 있다. 자동차 내부에서, 특히 브레이크 전자화 모듈에서 발생하는 일시적인 에러는 심각한 영향을 초래할 것인가? 만약 심각한 문제를 초래한다면 그러한 심각한 영향은 에러가 발생할 때마다 나타날 것인가? 이러한 심각한 영향을 하드웨어의 변경 없이 소프트웨어만으로 줄일 수 있을 것인가?

먼저, 자동차 내의 시스템에 있어서의 에러의 영향을 조사하기 위해 CPU 레지스터와 메인 메모리에 에러를 삽입시켰다[3]. 그 결과 47%에 이르는 에러의 경우 아무런 오동작 없이 시스템이 정상적으로 동작하였고, 23%에 이르는 에러의 경우 현재의 하드웨어 구조상으로도 검출이 가능하였다. 또한, 나머지 30% 중에서도 26%에 이르는 에러는 큰 영향을 끼치지 않았고, 4%의 에러

만 브레이크가 동작하지 않거나, 핸들이 잠기는 등 브레이크 전자화 기술에 심각한 영향을 끼쳤다. 이러한 실험 결과는 시사하는 바가 크다. 지금까지의 가정에서 우리는 자동차는 치명적인 내장형 시스템이므로 완전한 보호가 필요하다고 언급하였다. 그러나 에러가 발생하더라도 4%만이 치명적인 영향을 끼쳤다. 즉, 완전한 보호란 것은 하드웨어 전체를 모든 시간동안 하는 방식이 아닌, 필요한 시간에 필요한 부분에 부분적이고 효율적인 보호를 의미하는 것이다.

해당 논문에서는 치명적 영향이 발생하였을 때 어떤 부분에 에러가 삽입되었는지를 추적하는 비교적 단순한 방법을 사용하였다. 그 결과 ABS 제어기구의 적분기(integrator) 부분이나, 스택 포인터에 에러가 삽입될 경우 브레이크나 핸들에 영향을 주는 것을 찾을 수 있었다. 바꿔 말하면, 위의 4%의 확률을 줄이기 위해서 필요한 것은 하드웨어 전체를 전체 시간 동안 보호하는 것이 아니라 적분기 부분과 스택 포인터를 보호하는 것이 효율적이다.

적분기의 에러를 줄이기 위해서는 비율의 한계를 확인하는 방식(rate limit check)를 사용하였다. 적분기란 어떠한 변수의 값을 입력값으로 받아 적분값을 출력해주는 연산 회로로, ABS에서는 브레이크 요청과 바퀴의 미끄러짐을 입력받아 ABS의 동작 여부 등을 출력한다. 이러한 브레이크 요청이나 바퀴의 미끄러짐의 경우는 분명히 한계가 있는 입력값이므로, 적분 출력값 역시 한계를 지니다. 비율 한계 확인법에서는 이러한 출력값의 한계를 미리 계산한 후 한계점으로 산정하였다. 그 후 계산되는 출력값이 이러한 한계점보다 높거나 숫자값이 아닌 잘못된 값이 출력되는 경우에는 이전의 값으로 돌아가는 간단한 소프트웨어 기법을 사용하였다[3].

그렇다면 스택 포인터의 에러 처리는 어떠하였을까? 브레이크 제어기의 다른 함수를 부르기 전에 CPU 레지스터 안의 스택 포인터의 복사값을 저장하는 스케줄러를 사용하여서 해결하였다. 이를 통해 함수가 리턴 값을 반환할 때 스택 포인터의 값을 비교하게 되고, 다를 때에는 에러를 발견하는 방식이다. 다를 경우에는 에러가 발생하였다고 판단하여 브레이크 제어기를 재동작시켜 에러가 영향을 끼치기 전에 해결하는 방식이었다. 두 개의 간단한 소프트웨어 혹은 하이브리드 방법을 통해서 치명적인 영향을 0.4%로 줄일 수 있었다[3].

위 논문을 통해서 우리는 자동차의 신뢰성 문제가 분명히 중요한 문제이고 그에 따른 완전한 보호가 가장 큰 숙제이지만 동시에 어떻게 효율적인 보호를 해야 할지에 대한 고민이 필요함을 알 수 있었다. 앞서 언급한바와 같이 점점 더 전자기기 혹은 거대한 컴퓨터의 모습을 취하고 있는 자동차의 소프트웨어를 보호하기 위해서는 종래의 기기와 같은 에너지 효율성, 성능 최적화에 대한 문제가 끊임없이 제기되는 것이다.

3.2 표준화가 필요한 이유는 무엇일까?

다시 한 번 토요타 급발진 사태에 대해서 살펴보자. 토요타 급발진 사례에서 문제가 되었던 부분 중 하나는 토요타가 자동차 업계에서 지켜야 할 표준을 지키지 않았다는 점이었다. 토요타에서는 2002년부터 2004년에 이르기까지 지속적으로 표준을 준수하고 있다고 얘기하였지만, 실질적으로 NASA의 보고서에 보고된 사항보다 훨씬 많은 표준 위반 사항이 발견되었다. 프로그래밍의 표준화는 도대체 무엇일까?

자동차, 항공, 우주 산업, 원전 사업 등 치명적인 시스템에서도 프로그래밍이 갈수록 중요해지자, 영국 자동차 산업 신뢰성 협회(MISRA: motor

industry software reliability association)에서는 MISRA 기준을 제정하고 이에 따른 C 프로그램의 기준을 정립한 MISRA-C를 발표하였다[23]. 2004년에 발간된 MISRA-C:2004는 121개의 필수 규칙과 권고 규칙 20개 총 141개로, 함수 사용 방법부터 포인터 사용 방법에 이르기까지 총 21개의 영역에 걸쳐 자세하게 정의되어 있다.

C 언어는 좋은 언어 확장성, 하드웨어 전반에서 사용될 수 있는 범용성을 무기로 자동차뿐만 아니라 많은 내장형 시스템의 소프트웨어에서 사용되고 있다. 그러나 C 언어는 프로그래머의 잦은 실수나 언어에 대한 잘못된 이해, 컴파일러의 오작동이나 내재된 에러로 인한 불안성도 동시에 가지고 있다. 이로 인해, C 언어의 경우 1,000 줄의 코드에서 최대 500개 이상의 버그가, 평균적으로도 50개 이상의 버그가 발생하는 불완전한 모습을 보인다[24].

MISRA-C:2004 규칙 16.2에 따르면 '함수는 직접 혹은 간접적으로 그 함수를 부르는 것을 금지한다'고 말하고 필수 규칙으로 제정되어 어떤 형태로든 재귀 호출을 금하고 있다. 이는 재귀 호출이 간결한 모습을 취하는 강점을 지니나, 스택의 사용 허용량을 초과할 수 있는 잠재적인 문제점을 가지고 있기 때문이다. 그러나 토요타의 프로그래밍 코드에서는 재귀 호출이 쓰이는 등 MISRA-C 위반 사항이 8만 건 이상 발견되었을 뿐만 아니라 100개가 넘는 규칙 중 겨우 11개의 규칙만을 준수하고 있었다는 것이 추가적으로 발견되었다.

이러한 MISRA-C에는 프로그래머 관점에서 인간이 직접 살펴보아야 하는 부분이 존재하나, 100% 프로그래머에게 의존하는 접근은 현실적으로 어렵기 때문에 정적 분석 도구(static code analysis)를 사용하는 방법을 추천한다.

두 번째로 필요한 것은 자동차 운영체제의 표준화이다. 자동차가 거대한 내장형 시스템의 모습을 갖춰감에 따라 운영체제의 중요성이 대두되었다. 이에 따라, BMW, 크라이슬러, 폭스바겐 등 독일의 주요 자동차 회사와 칼스루에 대학교에서는 자동차 내 전기장치를 위한 운영 체제(OSEK: offene systeme und deren schnittstellen für die elektronik in kraftfahrzeugen)에 대한 표준을 정하여 제정하였다[25].

자동차는 자동차 회사의 이름을 걸고 출시되는 하나의 상품이지만 내부의 각 구성 요소는 여러 하청업체에서 제작되어 공급받는다. 당연히 다른 회사에서 제작된 하드웨어의 경우 호환성에 문제가 발생하였고, 게다가 호환성이 보장되지 않다 보니 중구난방 식으로 소프트웨어의 복잡도나 그 수가 증가하였다.

이러한 비효율성을 해결하기 위해서 응용 소프트웨어의 이식성과 재활용성을 높이는 것을 목표로 OSEK이 제안되었다. 당연히 공급자에 의존적인 인터페이스를 최대한 지양해야 하므로, 통신, 운영체제, 네트워크 관리 등 전 분야에 걸친 표준이 정립되었다. 토요타의 경우는 이러한 OSEK를 수정한 Rx-OSEK850를 사용하였다고 주장하였으나, 이는 OSEK의 성질을 따르지 않다는 것이 밝혀졌다. 그로 인해서 주요 데이터가 의도치 않은 행동을 보이는 것이 발견되었다.

마지막으로 살펴볼 사안은 자동차 기능 안전성 국제 표준(ISO 26262)이다. 자동차의 소프트웨어의 비중이 커지자 국제표준기구(ISO: international organization for standardization)에서는 자동차 내 소프트웨어의 거의 모든 사항을 다룬 ISO 26262를 제정하고, 기준을 통과한 업체에 대한 인증을 진행하였다[26,27]. ISO 26262는 출시된 소프트웨어에 대한 검사뿐만 아니라, 명세(specification), 설계, 구현, 통합, 검증, 생산 및 운영에 이르기까지 588개 이상의 주요 요구 사항을 지닌 매우 거

대한 표준이다.

지금까지 살펴본 기술은 현재의 기술에 적용된 내장형 시스템의 신뢰성을 향상시키는 방법이다. 그렇다면 다음 장에서는 미래 자동차 기술에서 예상되는 문제에 대해서 정리한다.

4. 자동차 미래기술과 신뢰성의 상관관계

먼저 생각해볼 수 있는 문제는 무인 자동차이다[28]. 흔히 생각하기에 무인 자동차는 먼 미래의 기술처럼 생각하기 쉽다. 그러나 미국 캘리포니아주에서는 2014년 9월부터 무인자동차에 대한 면허를 교부한다고 밝혔다. 물론 현재는 우수한 운전 실력을 보장받은 제조사 종업원이나 테스트 드라이브에 한해 500만 달러 이상의 보험이 보장된 제한된 환경에서만 교부가 가능하다[36]. 또한, 2014년 5월 구글은 무인 자동차에 일반인 시승을 성공적으로 마친바 있다[29]. 그러나 산적한 문제를 해결하기에는 아직 갈 길이 멀다.

가장 큰 문제는 신뢰성이다. 지속적으로 강조되는 얘기지만 자동차는 대표적인 치명적인 시스템 중 하나로 하나의 에러가 끔찍한 결과를 초래할 수 있는 내장형 시스템이다. 먼저 사용자의 거부감도 하나의 이유로 작용한다. 자동차제조사연합(alliance of automobile manufacturers)의 2012년 설문에 따르면 72%의 운전자가 무인자동차 기술에 부정적인 인식을 가지고 있으며, 그 중 절반이상은 무인자동차의 기술과 신뢰성은 기술 최소 몇 년 이상의 기술 간극을 갖고 있다고 생각하였다[30].

또한, 전기자동차의 증가세 역시 신뢰성 문제를 야기한다[3]. 최근 전기자동차 업계를 선도하는 테슬라(TESLA) 사가 자사의 고유 기술을 조건없이 공개하는, 이른바 오픈소스운동정신을 계승하기로 결정하였다[31]. 또한, 국내에서는 아직까지는 하이브리드 방식이 조금 더 보편적이나 생산을 조금씩 늘려가고 있다. 이 뿐만 아니라, 산업통상자원부 등에서 역시 관련 규정을 하나하나씩 만들어가면서 다가오는 전기자동차 시대를 대비하고 있다[32].

현재 전기 자동차의 안전성은 주로 하드웨어 자체가 다뤄져 왔다. 예를 들어, 전기자동차는 기존의 자동차에 비해 질량이 많이 나가는 것을 이유로 최소한 교통사고에 있어서만큼은 무거운 차량이 가지는 이점을 무기로 안전하다거나, 혹은 테슬라 사 전기자동차의 연이은 사고를 이유로 들어 배터리 기술의 불안정성을 언급하였다[33]. 그러나 미래에는 현재의 문제인 무게를 줄이기 위한 기술이 적용되면서 새로운 문제가 예상된다. 예를 들어 자동차의 무게를 줄이기 위해 컨트롤러를 전기적인 신호를 사용하는 전자화 기술이 조금 더 공격적으로 사용될 가능성이 높고, 그렇다면 앞서 언급한 문제가 발생할 것이다. 혹은, 이러한 전기장치의 최적화를 위해서 멀티코어 환경이 사용되는 등 현재의 스마트폰에서 발생하는 문제가 발생할 가능성도 간과할 수 없다.

5. 결 론

지금까지 자동차의 신뢰성 혹은 안정성을 고려할 때는 기계적인 문제로 접근하였다. 그러나 여전히 기계적인 문제가 중요한다는 점은 간과할 수 없지만, 또한 자동차 내부의 환경, 전자화 기술의 증가 등의 문제로 컴퓨터과학의 문제가 꾸준히 등장하였다. 또한, 가장 안전한 자동차로 인식되었던 토요타 프리우스 급발진 사태는 소프트웨어로 인한 급발진 사태로 밝혀져 이러한 자동차의 소프트웨어가 가지는 의미가 점점 더 중요

해지고 있다. 이러한 자동차 내장형 시스템 소프트웨어의 신뢰성을 향상시키기 위하여 대표적인 치명적 내장형 시스템인 브레이크 전자화 기술의 신뢰성 분석 및 향상 방법을 서술하였다. 또한, 이러한 컴포넌트 하나하나의 접근뿐만 아니라 시스템 전체의 신뢰성을 확보하기 위한 표준화 제정 노력도 필요하다.

한국은 대표적인 자동차 생산국이자 수출국이지만, 그러한 세계적인 위상에 비해서는 자동차 내장형 시스템 소프트웨어의 발전 단계는 아직 완전한 고도에 이르지는 모못한 상태이다. 이러한 내장형 시스템 자체 기술의 고도화를 위해서 선행되어야 할 조건은 신뢰성임을 깨닫고, 하루빨리 소프트웨어의 신뢰성 향상을 위한 연구에 박차를 가해야 할 것이다. 그렇게 된다면 한국의 자동차 산업은 고신뢰 컴퓨팅으로 대표될 수 있는 새로운 마케팅 전략을 얻을 수 있을 뿐만 아니라, 그를 토대로 새로운 시장을 개척할 수 있으리라 사료된다.

Acknowledgement

이 논문은 2012년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2012R1A1A1015421).

참 고 문 헌

[1] OICA, Worldwide automobile production statistics [Internet], http://www.oica.net/category/production-statistics/

[2] 조선일보, “도요타 3세대 프리우스 190만대 리콜…"소프트웨어 결함 때문”, 2013.

[3] D. Skarin and J. Karlsson, “Software implemented detection and recovery of soft errors in a brake-by-wire system”, IEEE Dependable Computing Conference, 2008.

[4] 한태만, 조진희, “자동차 전자제어 장치용 소프트웨어 기술 및 표준화 동향”, ETRI 전자통신동향분석, 2010.

[5] New Electronics, “Growing number of ECUs forces new approach to cars electrical architecture”, 2012.

[6] G. Georgakos, S. Schlichtmann, R. Schneider, and S. Chakraborty, “Reliability challenges for electric vehicles: From devices to architecture and systems software”, ACM Annual Design Automation Conference, 2013.

[7] M. Baleani, A. Ferrari, L. Mangeruca, A. Sangiovanni-Vincentelli, M. Peri, and S. Pezzini, “Fault-tolerant platforms for automotive safety-critical applications”, ACM International Conference on Compilers, Architecture and Synthesis for Embedded Systems, 2003.

[8] J. Blome, S. Mahlke, D. Bradley, and K. Flautner, “A microarchitectural analysis of soft error propagation in a production-level embedded microprocessor”, International Symposium on Microarchitecture, 2005.

[9] S. Jagannathan, Z. Diggins, N. Mahatme, T. Loveless, B. Bhuva, S. Wen, R. Wong, and L. Massengill, “Temperature dependence of soft error rate in flip-flop designs”, IEEE International Reliability Physics Symposium, 2012.

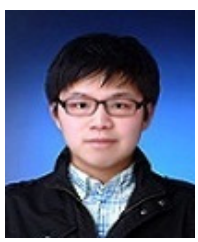
[10] MIT Technology Review, “A software update for your car?”, 2012.

[11] C. Jones, “Measuring defect potentials and defect removal efficiency”, CrossTalk The Journal of Defense Software Engineering, Vol.21, No.6, pp.11-13, 2008.

[12] M. Jefremow, T. Kern, U. Backhausen, C. Peters, C. Parzinger, C. Roll, S. Kassenetter, and S. Thierold, D. Schmitt-Landsiedel, “Bitline-capacitance-cancelation sensing scheme with 11ns read latency and maximum

read throughput of 2.9 GB/s in 65nm embedded flash for automotive", IEEE International Solid-State Circuits Conference Digest of Technical Papers, 2012.
[13] P. Mayr, C. Weyers, and U. Langmann, "A 90GHz 65nm CMOS injection-locked frequency divider", IEEE International Solid-State Circuits Conference Digest of Technical Papers, 2007.
[14] J. Axmacher, "World's first 40nm MONOS flash technology", Electronics World, Vol.118, No.1913, pp.36-38, 2012.
[15] Y. Yokoyama, Y. Ishii, H. Kojima, A. Miyanishi, Y. Tsujihashi, S. Asayama, K. Shiba, K. Tanaka, T. Fukuda, and K. Nii, "40nm Ultra-low leakage SRAM at 170 deg. coperation for embedded flash MCU", IEEE International Symposium on Quality Electronic Design, 2014.
[16] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone", USENIX annual technical conference, 2010.
[17] V. Chandra and R. Aitken, "Impact of technology and voltage scaling on the soft error susceptibility in nanoscale CMOS", IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems, 2008.
[18] S. Edelen and R. Jones, "Electrically operated parking brake system", US Patent, 1994.
[19] NASA and NHTSA, "Technical Support to the national highway traffic safety administration (NHTSA) on the reported Toyota motor corporation (TMC) unintended acceleration (UA) Investigation", 2011.
[20] M. Barr, "2005 Camry L4 software analysis", 2014.
[21] W. Wulf and M. Shaw, "Global variable considered harmful", ACM Sigplan Notices, Vol.8, No.2, pp.28-34, 1973.
[22] T. McCabe, "A complexity measure", IEEE Transactions on Software Engineering, No.4, pp.308-320, 1976.
[23] MISRA, "Guidelines for the use of the C language in vehicle based software, Guidelines for the use of the C language in critical systems", 2004.
[24] J. Ganssle [Internet], "Project management embedded failures", http://archive.oredev.org/download/18.5bd7fa0510edb4a8ce4800019898/1385353982337/Jack_Ganssle_-_Embedded_Failures_2.pdf.
[25] OSEK Group, "SEK/VDX Operating System Specification", 2009.
[26] T. Dittel and H. Aryus, "How to "Survive" a safety case according to ISO 26262", Springer Computer Safety, Reliability, and Security Lecture Notes in Computer Scienc, Vol.6351, pp.97-111, 2010.
[27] 채승엽, "ISO 26262로 달라지는 자동차 시장의 변화", 정보과학회지, Vol.29, No.9, pp.42-47, 2011.
[28] 공영일, "자동운전 자동차(self-driving car), 어떻게 볼 것인가", 정보통신방송정책, Vol.25, No.7, 2013.
[29] Google, "A first drive" [Internet], http://youtu.be/CqSDWoAhvLU
[30] Wall Street Journal, "The big worry about driverless cars? Losing Privacy", 2013.
[31] 아시아투데이, "테슬라, 전기자동차관련 기술 공개 결정", 2014.
[32] 에너지관리공단, "환경친화적 자동차의 요건 등에 관한 규정", 2011.
[33] 동아일보, "테슬라 모델S 또 불, 연이은 화재 사고로 곤혹", 2013.
[34] 권문식, "차세대 자동차의 신기술 동향 및 전망", 산업연구원 한국물리학회지, 2004.
[35] 김현철, "한국 자동차산업의 현재와 미래", 한국자동차공학회 오토저널, Vol.35, No.6, pp.66-69, 2013.
[36] Cars That Think, "California to issue driver's licenses to robots", 2014.

저 자 약 력



고 요 한

이메일 : yohan.ko@yonsei.ac.kr

- 2012년 연세대학교 컴퓨터과학과 졸업(학사).
- 2012년~현재 연세대학교 컴퓨터과학과 석박통합과정.
- 관심분야 : 내장형 시스템, 고신뢰 컴퓨팅, CGRA 등 특수한 내장형 시스템 고성능/고신뢰 컴퓨팅 방법론

이 경 우

이메일 : kyoungwoo.lee@yonsei.ac.kr

- 1995년 연세대학교 컴퓨터과학과 졸업(학사).
- 1997년 연세대학교 컴퓨터과학과 졸업(석사).
- 2008년 University of California at Irvine, Information and Computer Science(박사).
- 2011년~현재 연세대학교 컴퓨터과학과 조교수.
- 관심분야 : 내장형 시스템, 고신뢰성/저전력 컴퓨팅, 모바일 멀티미디어