소프트웨어가 주도하는 차량 안전 기술: Connected Automated Vehicle(CAV) 구현의 열쇠

Software-driven Vehicle Safety: Key Technology Enabling Connected Automated Vehicle (CAV)





홍성수
서울대학교
Seongsoo Hong
Seoul National University



이청화
서울대학교
Cheonghwa Lee
Seoul National University

강환철
서울대학교
Hwancheol Kang
Seoul National University

미래 자동차가 연결기반 자율주행 차량(CAV : Connected Automated Vehicle)의 형태로 구현될 것이라는 합의가 세계적으로 이루어지고 있다.[1] 그 이름이 의미하듯이 CAV 기술의 핵심은 연결성과 기계 지능에 있다. CAV는 연결성을 통해 인프라로부터 실시간 교통환경 정보를 획득하고 인접 차량과 센서 데이터를 공유함으로써 인지거리를 확대하며, 기계 지능을 통해 누적된 주행경험에 근거한 적응적 차량 제어를 수행한다.

최근 CAV가 부각되는 이유는 관련된 기반기술이 빠르게 성숙되고 있기 때문이다. 즉, 큰 대역폭과 제한된 전송지연시간을 지원하는 5G의 상용화가 시작되었으며, 딥러닝에 기반한 AI 기술이 성숙되었고, 빅데이터를 처리할 수 있는 클라우드 서버들이 보편화되고 있다는 점이 중요한 촉매제가 된 것이다.

CAV가 지니고 있는 무한한 잠재력으로 인해 사용자들은 지금까지 보지 못한 새로운 가치와 혁신적인 서비스를 경험하게 될 것이다. CAV는 자동차 산업에서 추구하는 세가지 궁극의 목표인 1) 안전, 2) 편의, 3) 친환경성을 통합적으로 달성하는 완성 기술이기 때문이다. 그러나 이런 CAV를 실질적으로 구현하여 양산의 단계에 이르게 하는 것은 매우 어려운 미션이다. 그 이유는 다음과 같다.

- CAV는 "System-of-Systems Integration via Connectivity"를 추구하기 때문에 Heterogeneity, Interoperability, Concurrency, Distribution 등의 무한한 시스템 복잡도를 가진다.
- CAV는 본질적으로 개방성을 추구하기 때문에 외부의 사이버 공격에 능동적으로 대응해야 한다. 또한 개방된 외부 환경에 적응적으로 대응할 수 있도록 온라인 동작 제어 최적화를 구현해야 한다.
- CAV는 차량의 기계 지능을 구현하기 위해 방대한 연산 성능을 요구하는 컴퓨팅 플랫폼을 장착해야 한다.

이와 같은 CAV의 기술적 난제를 극복하는 데에 있어 핵심적인 것이 소프트웨어 기술력을 확보하는 것이다. 본 고에서는 궁극적 안전 차량을 추구하는 CAV의 소프트웨어적 측면을 살펴보고자 한다. 이를 위해 먼저 차량 소프트웨어가 자동차 안전의 측면에서 진화해온 과정을 구체적 사례를 통해 살펴본다. 이렇게 차량 안전에서 소프트웨어의 역할을 인식한 뒤, CAV 소프트웨어의 기술적 난제들과 그에 대한 해결 방안, 그리고 새로운 개발방법론을 살펴본다.

차량 안전 기술의 진화와 소프트웨어의 역할

차량 안전 기술은 수동 안전과 능동 안전의 순서로 발전해 왔으며, 미래에는 연결기반 자율주행의 단계로 완성될 것이다. 이 장에서는 이 세가지 기술에서 소프트웨어가 어떠한 역할을 수행하는지 알아본다.

초기의 차량 안전은 수동 안전(Passive safety) 기술에 기반을 두고 있었다.[2] 수동 안전의 목적은 사고가 발생하였을 때, 그 피해를 최소화시키는 것이다. 이에 반해 능동 안전(Active safety)은 사고가 예상되는 상황을 감지하여 예방하는데 주안점을 둔다.[3] 특히 소프트웨어 기술이 혁신적으로 발전함에 따라, 능동 안전 시스템은 기계적 요소와 각종 센서들을 융합하여 차량의 주변환경까지 인지하여 사고를 예방하는 단계로 발전하게 되었다. 미래에는 소프트웨어뿐만 아니라 통신과 컴퓨팅 기술이 동반 성장하여 CAV 기반 안전 기술이 보편화 될 것이다.[4] CAV 기반 안전 기술은 시스템 간에 공유된 정보를 통해 자율주행차량이 거시적인 관점에서 상황을 인지하고, 종합적인 해석과 예측을 통해 일어날 가능성이 있던 사고마저 예방할 수 있도록 하는 이상적인 안전 기술이다.

과거로부터 진화해온 각 안전 기술의 특징들을 그 목표, 사례, 구성 요소, 기반 기술 등으로 나누어서 요약한 내용이 〈표 1〉에 제시되어 있다. 이를 통해 차량 안전이 점점 더 통신, 소프트웨어, 컴퓨팅 하드웨어 기술에 의존하는 경향이 있음을 확인할 수 있다.

표 1. 차량 안전 기술과 특징

Category		수동 안전 기술	능동 안전 기술	CAV 안전 기술
목적		피해를 최소화	사고 예방	사고 예측 및 회피
예시		에어백, 안전벨트, 머리지지대	제동거리 단축 시스템, 주차 보조 시스템, 차선 이탈 방지/유지 보조 시스템	Dynamic Online Control Cooperative Powertrain Control
구성 요소		기계 및 전자	전자 및 시스템	시스템 및 통신
소프트웨어 의존도		적음	많음	매우 많음
적용 범위		차량 영역	차량 및 주위환경	차량간 및 광범위 환경
통신	In-vehicle	Low Bandwidth	High Bandwidth	High Bandwidth
	V2X	–	Minimum Capability	Maximum Capability
컴퓨팅 하드웨어		Low Performance MCU Many ECU	Consolidated SoC Domain Control Unit/ECU	Centralized, Heterogeneous, High Performance Computer

● 수동 안전에서 소프트웨어의 역할

수동 안전에서 소프트웨어는 차량 내부의 상태를 모니터링하고, 특정 조건이 되었을 때 정해진 방어기능을 수행시키는 역할을 한다. 이 역할에는 하드웨어에서 Failure가 발생할 경우, 이를 감지하여 안전한 상태로 복구하는 작업을 수행시키는 것이 포함된다. 수동 안전에서 소프트웨어가 수행하는 가장 중요한 역할은 기계나 전자적 요소로부터 파생되는 결함을 마스킹(Masking)하는 것이라고 요약할 수 있다.

"페달 게이트(Pedal-Gate)"라고도 불리는 도요타 리콜 사태〈그림 1〉는 수동 안전에서 이러한 소프트웨어의 역할을 간과한 대표적인 사례이다.[5] 해당 사건은 가속 페달의 오작동이 원인이 되었지만, 수동 안전 소프트웨어 기술을 통해 브레이크를 방어적으로 동작시켰다면 피해를 최소화할 수 있었을 것이다.

이를 이해하기 위해 도요타 리콜 사태를 소프트웨어적 관점에서 다시 살펴보자. 그 구체적인 해결 방법은 브레이크 오버라이드 시스템(BOS, Brake Override System)이라는 간단한 소프트웨어 예외 처리기를 시스템을 추가하는 것이다. 브레이크 오버라이드 시스템은 가속 페달이 동작하고 있는 상황에서 브레이크가 인가되는 경우, 이를 비정상적인 예외 상황으로 인지한다. 이런 예외 상황에서 브레이크 오버라이드 시스템은 소프트웨어적으로 가속페달의 영향을 무시하고 브레이크만을 활성화하여 감속 제어를 실시한다. 도요타 리콜 사태를 되돌아 보면, 매우 간단한 방법으로 심각한 피해를 예방할 수 있었구나 하는 큰 아쉬움이 남게 된다.

● 능동 안전에서 소프트웨어의 역할

사고를 예방하는 목적을 가진 능동 안전은 수동 안전에 비해 소프트웨어에 대한 의존도가 더욱 심화된다. 차량 안전 시스템이 사고를 예방하려면, 차량 내부뿐만 아니라 차량 외부까지 상황을 모니터링 해야 한다. 이때, 위험요소의 특성을 판단하는 과정에서의 정확성과 사고회피를 위한 대응조치를 계산하는 과정에서의 신속성이 요구된다. 이를 위해 센서나 프로세서 등의 하드웨어를 제어하는 소프트웨어의 역할이 매우 중요하다. 이러한 측면에서 보면, 능동 안전에서의

그림 1. 도요타 리콜 사태를 대표하는 그림 (Cameron Cardow)[6]



그림 2. 우버의 자율주행자동차 보행자 사고 당시 사용된 실험 자동차의 장착된 센서 (Electrek)[8]



소프트웨어는 하드웨어 구성요소들과 유기적으로 상호작용하여 인적 오류(Human error)를 마스킹하는 역할을 한다고 요약할 수 있다.

지난 해 발생한 우버 자율주행자동차의 보행자 사고 사례는 능동 안전 시스템에서도 핵심적인 부분인 인지 소프트웨어의 중요성을 보여준다. 〈그림 2〉에서 보는 것과 같이 해당 우버 자율주행차량은 외부 상황을 모니터링하는 3종의 센서를 장착하고 있었다. 사고 당시 이들 센서는 정상적으로 동작하며 보행자를 감지하였다. 그러나 해당 차량의 인지 소프트웨어는 보행자를 주행 회피 장애물로 인지하지 못하고, 풍선이나 종이상자 같이 무시하고 달려도 될 대상으로 분류하였다.[7] 즉, 소프트웨어가 부정 오류(False negative)를 범한

것이 사고의 원인이라고 분석된다. 우버 회사 측은 실험 당시 운행을 원활하게 하기 위해 주변 위험 요소에 대한 민감도를 낮게 설정하였기 때문에 발생한 사고라고 발표하였다. 최적의 민감도를 찾아서 원활한 운행을 하면서도 사고 발생율을 최소화하는 것이 능동 안전에서 소프트웨어의 숙제이다.

● CAV에서 소프트웨어의 역할

CAV에서 소프트웨어는 실시간으로 공유되는 방대한 양의 스트리밍 정보를 관리하고 외부의 부적절한 공격으로부터 시스템을 지키는 역할을 수행한다. 뿐만 아니라 획득한 정보를 기반으로 가상 시뮬레이션을 수행하여 가능성이 있는 모든 경우를 예측하고, 그에 상응하는 동작 방식을 찾아 최적의 차량 제어를 수행한다. 이렇게 함으로써 미래에 발생할 사고를 원천적으로 예방하게 되는 것이다.

CAV는 각각의 개별적인 차량의 개념을 넘어, 5G, WiFi 등의 무선 통신망과 유선 인터넷, 그리고 에지 서버(Edge server)와 클라우드 서버(Cloud server)를 포함하는 포괄적이면서 매우 이질적인 구조를 가진다. 〈그림 3〉은 이와 같은 연결성과 복잡성을 갖는 CAV 아키텍처를 보여준다.

CAV 형태를 갖는 미래의 차량의 가장 큰 유용성은 "연결성에 의한 개방성"으로 창출된다고 볼 수 있다. 전통적으로 매우 폐쇄적이었던 차량 제어 및 주행 시스템이 외부 통신망과의 연결을 통해 개방됨으로써 지금까지 구현할 수 없었던 획기적인 기능을 소프트웨어 앱(Software application)의 형태로 구현해 넣을 수 있다. 그 대표적인 예로, Cooperative powertrain control을 들 수 있다. 또한 차량의 자율주행에 필수적인 기계 지능의 분산적 구현과 동적 연산 오프로딩과 재분배도 들 수 있다.

여기에서는 Cooperative powertrain control에 대해 좀더 자세히 살펴본다. 기존 엔진 제어 시스템은 Crankshaft의 각도에 따라 사전에 정해진 Angular task를 실행시켜 엔진을 관리한다. 이때 RPM에 무관하게 동일한 Angular task를 수행시키면, RPM이 높아짐에 따라 Angular task의 실행 주기가 짧아져서 CPU에 더 많은 부하가 걸리게 된다. 따라서 정적 스위칭을 사용하는 기존 엔진 제어 시스템은 RPM이 높

그림 3. CAV 기반의 스마트시티 개략도 (Yang et al.)[9]

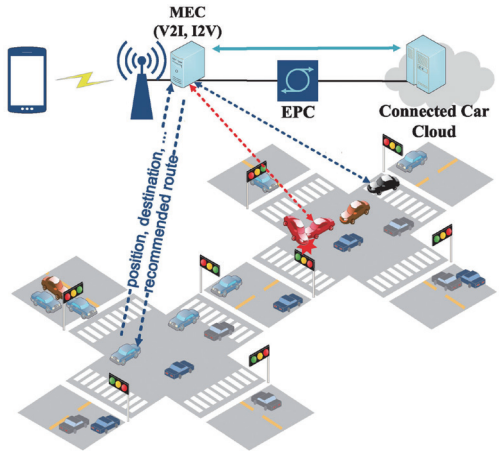
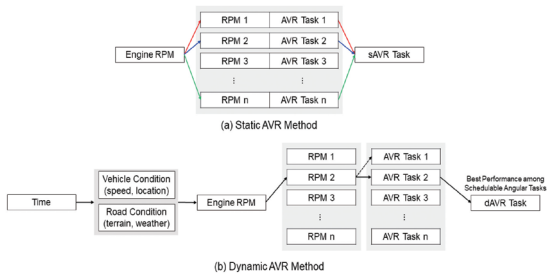


그림 4. Cooperative Powertrain Control



을수록 성능과 복잡도가 낮은 Task를 실행시킨다. 이와 같이 RPM의 구간별로 다르게 실행되는 Task를 AVR(Angular Variable Rate) task라고 한다. 〈그림 4〉의 (a)는 기존 엔진 제어 방식인 정적 스위칭 제어 기법을 보여준다. 이 기법에서는 각 AVR task가 Trigger되는 RPM 값(Switching speed)을 개발 단계에서 미리 정하기 때문에 사전에 고려된 특정 교통 환경에서만 최적화된다는 한계가 있다.

이에 반해 Cooperative powertrain control 시스템은 CAV의 연결성을 통해 제공받은 정보(미래 제어 시점 τ에 지나가게 될 도로 정보와 차량 정보)를 이용하여 제어 대상 차량의 τ에서의 RPM을 예측한다. 이어서 제공받은 통합 정보

와 예측된 RPM을 통해 그에 최적화된 여러 개의 AVR task들을 동적으로 도출한다. 또한 도출된 AVR task들의 스케쥴링 가능성을 검사하여, 스케쥴링이 가능한 것 중 가장 성능이 좋은 task를 선택한다. 해당 시점 τ에 도달하면 미리 선택된 AVR task를 수행시킨다. 〈그림 4〉의 (b)는 CAV에서 사용될 수 있는 동적 스위칭 제어 기법을 보여준다. 이와 같은 Cooperative powertrain control에서는 동적 예측 알고리즘과 동적 스케쥴링 기법이 매우 중요한 요소가 된다.[10,11]

위의 Cooperative powertrain control의 예에서 볼 수 있듯이, CAV 기술은 단순히 차량 탑승자와 인터넷 서버 간의 편의 정보 공유에 그치지 않고, 동적 최적 제어를 통한 연비 개선과 무사고에 도전하는 궁극의 차량 안전을 제공할 수 있는 토대를 제공한다. 그러나 CAV를 실제로 구현하는 과정에는 많은 어려운 기술들, 특히 소프트웨어 기술들이 존재한다. 따라서 미래 자동차에서 소프트웨어의 역할은 더욱 심화될 것이다.

미래 자동차 설계 혁신을 위한 개발방법론

다가오는 미래에 CAV는 새롭고 무한한 모빌리티 서비스를 창출할 것으로 기대된다. 또한 세계적으로 수 많은 혁신적인 사회적 가치를 인류에게 제시하게 될 것이다. 그러나 CAV가 창출하는 이런 혜택을 얻고, CAV를 수익성 높은 산업으로 실체화 시키려면, 많은 기술적인 난제들을 해결해야 한다. 이러한 난제들은 자동차 산업의 범위를 넘어 반도체, 통신, 소프트웨어 산업과 밀접하게 연결되어 있다. 물론 자동차 산업은 그 중심에서 핵심적인 역할을 수행하여야 한다. 본 절에서는 CAV 소프트웨어의 기술적 난제들과 그에 대한 해결방안, 그리고 새로운 개발방법론에 대해 살펴본다.

자동차공학의 관점에서, CAV 구현을 위해 우선적으로 주목해야 할 분야는 차량 제어 및 주행 소프트웨어 기술이다. CAV에서 사용될 제어 소프트웨어나 응용 소프트웨어를 개발할 때, 개발자를 가장 힘들게 하는 점은 개방성이 될 것이다.

지금까지 차량 제어기를 설계할 때, 개발자는 사전에 차량 제어기의 동작 조건을 완벽하게 분석하고, 한정 지으며, 이를 토대로 동작 기능을 구현하였다. 즉 차량 제어기의 폐쇄성에 근거하여, 개발 단계에서 수행되는 정적 최적화를 통해 제어 알고리즘을 구현한 것이다. 또한 차량 제어기 검증을 위해 사전에 분석된 테스트 케이스를 취합하여 테스트를 수행하였다. 이것이 자동차 산업에서 널리 사용되는 전통적인 V모델 방법론의 핵심이다.

그러나 CAV 소프트웨어를 개발할 때는, CAV의 개방성으로 인해 개발자는 더 이상 이런 폐쇄성을 가정할 수 없게 된다. CAV의 제어 시스템은 주행 시 제공받은 다양한 환경 정보와 차량의 상태 정보를 결합하여, 개방된 외부 환경에 적응적으로 대응할 수 있는 온라인 동작 제어 최적화를 구현하기 때문이다.

이와 같은 CAV 소프트웨어의 개방성은 개발자에게 크게 세 가지 어려움을 지우게 될 것이다. 첫째, 동적 최적화를 위해서는 정적 최적화 보다 더욱 복잡도가 높은 제어 알고리즘이 요구된다. 이런 알고리즘은 딥러닝 기술과 결합되어 누적된 주행 경험을 바탕으로 한 적응적 제어를 수행할 수 있도록 하지만, 한편으로 제어 시스템을 더욱 외부 의존적으로 만들 수 있다.

둘째, 동적 최적화에 필요한 소프트웨어 태스크들을 미리 결정할 수 없기 때문에 이런 태스크들이 사용하게 될 컴퓨팅 자원을 사전에 파악하는 것은 거의 불가능하다. 이런 한계를 극복하기 위해 동적 자원 할당을 지원하는 컴퓨팅 자원 관리가 필요하다. 그러나 동적 자원 관리를 통해 태스크의 실시간성을 보장하는 것은 이론적으로나 기술적으로 매우 어려운 일이다.

셋째, 지금까지 자동차 업계에서 널리 사용되던 V모델이 CAV 소프트웨어의 동적 특성과 충돌하게 된다. 한 예로, CAV 소프트웨어는 개발 단계에서 예측하지 못한 시나리오에서도 동작하여야 하기 때문에 V 모델의 정적 테스트가 더 이상 적용되지 못한다.

위에서 언급한 CAV의 개방성으로 초래되는 소프트웨어적 위협에 체계적으로 대처하기 위해 영국 정부와 영국 자동차

공학계는 사이버 복원력(Cyber resilience)으로 명명한 기술을 제시하고 있으며, 이를 포함한 CAV 로드맵 2030을 발표하였다.[1] 이 사이버 복원력은 Monitor–simulate–deploy에 기반한 소프트웨어 아키텍처와 개발방법론을 포함하고 있다. 이 아키텍처는 CAV가 미리 지정된 동작 환경에서 벗어나는 시점을 모니터링하고, 그런 상황이 발생하면 시뮬레이션을 통해 대응 가능한 기능들을 생성하도록 한다. 이어서 그런 기능들 중에 정당성 있는 것을 선택하여 차량에서 동적으로 구동시킨다.

CAV는 위에서 언급한 이슈들 외에도 많은 기술적 어려움들을 포함하고 있다. 우선 CAV는 System–of–systems라는 방대한 특성으로 인해 Heterogeneous distributed computing, Interoperability, Diversity 등의 복잡한 이슈를 포함하고 있다. 구체적인 예로 CAV를 위한 기계 지능의 구현을 보자. CAV의 기계 지능은 차량 내부, 에지 서버, 클라우드 서버가 나누어서 구현하게 되는데, 이들 간의 동작과 타이밍을 조정하는 것은 매우 어려운 기술이다.

이런 문제점을 인식하여 영국 정부는 자동차 업계, 학계와 더불어 CAV 기술의 리더쉽을 확보하기 위한 매우 세밀하고 체계적인 연구를 진행하고 있다. 영국 자동차 업계의 이런 활동은 전세계 자동차 산업에서 한 축을 담당하고 있는 한국의 자동차 업계와 학계에 많은 시사점을 주고 있다.

미래 자동차는 연결성과 기계 지능을 바탕으로 구현된 CAV의 형태가 될 것임에 의심하는 사람은 없다. CAV는 안전, 편의, 친환경성을 추구하는 자동차의 궁극의 형태로서 인류에게 혁신적 가치와 새로운 산업적 기회를 제공할 것이다. 그러나 CAV가 가지고 있는 본질적인 개방성으로 인해, 관련된 시스템 소프트웨어와 응용 소프트웨어를 개발하는 것은 기술적으로 매우 어려운 일이다. 이미 영국과 같은 선도국들은 2030년까지 CAV를 완성하고자 하는 적극적인 연구 개발 활동을 시작하였다. 세계적으로 자동차 산업의 한 축을 담당하는 한국에서도 CAV 소프트웨어의 다양한 기술적 이슈들을 체계적으로 분석하여 이를 해결하고자 하는 노력을 경주하여야 한다. 이를 통해 미래 모빌리티를 선도할 수 있는 기반을 구축하여야 할 것이다.

홍성수 교수 : sshong@redwood.snu.ac.kr

〈참고문헌〉

1. Zenzic, UK CAV Roadmap, https://zenzic.io/ roadmap/, 2019.
2. K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller and H. Winner, Three Decades of Driver Assistance Systems: Review and Future Perspectives. IEEE Intelligent Transportation Systems Magazine, Vol.6, No.4, pp.6–22, 2014.
3. A. Mukhtar, L. Xia and T. B. Tang, Vehicle Detection Techniques for Collision Avoidance Systems: A review. IEEE Transactions on Intelligent Transportation Systems, Vol.16, No.5, pp.2318–2338, 2015.
4. N. Wang, X. Wang, P. Palacharla and T. Ikeuchi, Cooperative Autonomous Driving for Traffic Congestion Avoidance through Vehicle–to–vehicle Communications, In 2017 IEEE Vehicular Networking Conference VNC, pp.327–330, 2017.
5. Wikipedia, 2009–11 Toyota Vehicle Recalls, http://en.wikipedia.org/wiki/2009%E2%80%9311_Toyota_vehicle_recalls, 2019.
6. C. Cardow, Toyota, https://www.cagle.com/cameron–cardow/2010/02/toyota–color–2, 2019.
7. National Transportation Safety Board, Preliminary Report Highway: HWY18MH010, 2018.
8. Electrek, Volvo Cars and Uber Join Forces to Develop Autonomous Driving Cars, https:// electrek.co/2017/11/20/uber–volvo–xc90/#jp–carousel–56567, 2019.
9. S. R. Yang, Y. J. Su, Y. Y. Chang and H. N. Hung, Short–Term Traffic Prediction for Edge Computing–Enhanced Autonomous and Connected Cars. IEEE Transactions on Vehicular Technology, Vol.68, No.4, pp.3140–3153, 2019.
10. C. Peng, Y. Zhao and H. Zeng, Dynamic Switching Speed Reconfiguration for Engine Performance Optimization, 56th ACM/IEEE Design Automation Conference (DAC), p.11, 2019.
11. Y. Shao, M. A. M. Zulkefli, Z. Sun and P. Huang, Evaluating Connected and Autonomous Vehicles Using a Hardware–in–the–loop Testbed and a Living Lab. Transportation Research Part C: Emerging Technologies, Vol.102, pp.121–135, 2019.