Transactions of KSAE, Vol. 33, No. 11, pp.933-945 (November, 2025)

pISSN 1225-6382 / eISSN 2234-0149
DOI http://dx.doi.org/10.7467/KSAE.2025.33.11.933

소프트웨어 정의 조향 제어기를 위한 자동차 안전공학 전략: 시스템 이론적 관점을 적용한 모델 기반 접근 방식

정 대 희[*1,2] · 권 기 현[2]

HL만도 SW Campus SW Engineering Lab SW 1[1] · 경기대학교 컴퓨터과학과[2]

Automotive Safety Engineering Strategy for Software-defined Steering Controller: A Model-based Approach with System Theoretic Perspective

Daehui Jeong[*1,2] · Gwhon Kwon[2]

[1]*SW Campus SW Engineering Lab SW 1, HL Mando, Bundang-gu, Seongnam-si, Gyeonggi 13486, Korea*
[2]*Department of Computer Science, Kyonggi University, Gyeonggi 16227, Korea*
(Received 1 July 2025 / Revised 7 July 2025 / Accepted 9 July 2025)

Abstract : This paper proposed a model-based safety engineering (MBSE) strategy to prevent accidents and ensure the safety of software-defined vehicles (SDVs). The strategy focused on system theory, especially the control structure diagram (CSD), designed as function-behavior-structure perspectives, and aimed to achieve functional safety and Safety of the Intended Functionality (SOTIF) simultaneously by integrating system theoretic hazard analysis and hierarchical reliability safety analysis in each perspective. To validate the proposed strategy, it was applied to an electric power steering (EPS) controller, which is one of the key systems in SDVs. The analysis results demonstrated that the proposed strategy can systematically identify hazards in SDVs and ensure safe control actions even in hazardous scenarios. This research is thus expected to provide an MBSE framework that addresses the limitations of existing reliability-based methodologies in SDV development by linking system design with analysis and verification for functional safety and SOTIF.

Key words : Model-based safety engineering(모델 기반 안전공학), System theoretic approach(시스템 이론적 접근), Software-defined vehicles(소프트웨어 정의 차량), Functional safety(기능안전), Safety of the intended functionality(운용안전), Function-Behavior-Structure Modeling(기능-행동-구조 모델링), Electric power steering(전동식 동력 조향)

1. 서 론

최근 자동차 분야에서는 고성능 및 고효율의 기능을 제공하기 위해 인공지능을 비롯한 당대 최고의 정보기술을 시스템에 융합하고 있다. 과거와 비교할 수 없을 정도로 시스템이 복잡해지면서 기능의 제어는 기계 중심 방식에서 소프트웨어 중심의 제어, 즉 SDV(Software-Defined Vehicles)로 전환되고 있다.

소프트웨어가 제어의 핵심인 SDV의 등장은 사용자에게 보다 다양하고 새로운 부가가치를 경험할 수 있게 하였지만 한편으로는 인명과 재산, 그리고 환경에 피해를 입히는 사고의 원인으로 부상하였다. 이에 국제 표준 ISO 26262[1] 및 ISO 21448[2]에서는 개발하고자 하는 시스템을 대상으로 안전 활동을 전개하여, SDV의 안전성을 확보하기 위한 기능안전(Functional safety) 및 운용안전(Safety of the intended functionality)의 달성을 요구하고 있다.

안전 활동은 ALARP(As Low As Reasonably Practicable) 원칙[3]에 따라 시스템이 초래할 수 있는 위험으로부터 사용자를 보호하기 위한 안전 기능을 구현하고 이를 검증하는 데 목적이 있다. 특히 Fig. 1에서 볼 수 있듯이 안전 활동의 처음과 끝에 해당하는 위험원 분석(Hazard

[*]A part of this paper was presented at the KSAE 2024 Fall Conference and Exhibition
[*]Corresponding author, E-mail: daehui.jeong@kyonggi.ac.kr

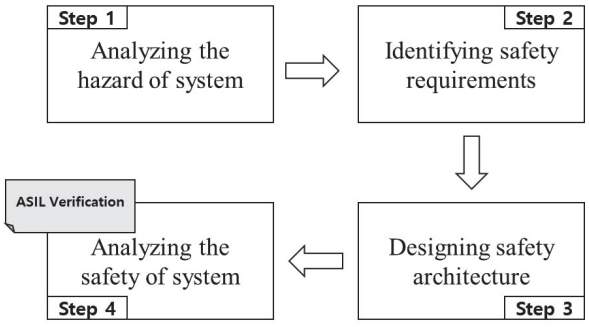

Fig. 1 Concept of developing safety capabilities for automotive

analysis)과 안전 분석(Safety analysis)[4]은 미처 발견되지 못하거나 충분히 검증되지 못한 시스템의 위험이 남지 않도록 시스템을 면밀히 분석한다. 이를 통해 SDV에는 허용할 수 있는 수준의 위험(Reasonable risk)만 남게 되고, 결과적으로 안전성이 확보된다.

안전 활동의 수행에는 시스템 특성과 목적에 따라 알맞은 안전공학 기술이 선택적으로 적용된다. 그 중에서도 자동차 분야에서 널리 활용되는 기술은 FMEA(Failure Modes and Effects Analyss) 또는 FTA(Fault Tree Analyss) 등의 신뢰성 기반 분석 방법론[5]이다. 이 방법론은 시스템을 구성하는 최하위 요소인 하드웨어 결함(Fault) 또는 소프트웨어 결점(Defect)가 어떻게 전체 시스템의 고장(Failure)으로 확산되는지 그 전파 과정을 연쇄적 사건(Chain of events)으로 구성 및 분석하여 차량 시스템의 안전성을 확보할 수 있도록 돕는다.

그러나 신뢰성 기반 방법론으로는 수많은 상호작용으로 기능이 수행되는 SDV의 특성을 충분히 반영하기에 한계가 있다. SDV는 차량에 장착된 감지기(Sensor)로부터 계측된 값을 받아 다음 주행 제어를 위한 명령어를 판단하거나, 차량을 운행하기 위한 전원(Battery)의 전력을 효율적으로 배분하여 전동기의 출력을 극대화하는 등 시스템을 구성하는 구성요소 간의 다양한 상호작용을 통해 차량 운행에 필요한 기능을 제공한다. 이러한 상호작용 구조는 사고를 발생시키는 인과관계에도 영향을 미쳐, 시스템의 고장만으로는 설명하기 어려운 복합적인 원인에 의한 잘못된 제어(Unsafe control)가 차량 사고의 주요 요인[6]으로 지목되기 시작했다.

이러한 문제를 해결하기 위해, 자동차 분야의 많은 연구자들은 MIT의 N. Levenson이 발표한 시스템 이론적 분석 방법론[7]을 활용하여 SDV의 안전성을 확보하는 방안을 제안하고 있다. Martínz[8]는 시스템 이론적 분석 방법론의 STPA(System Theoretic Process Analysis)를 EPS(Electric Power Steering)에 적용하여, 신뢰성 기반 분석 방법론의 FMEA를 사용하였을 때 보다 면밀하게 위험원을 파악할 수 있음을 소개하였다. Do[9,10]는 STPA로 SMK(Smart Key System)의 잘못된 제어로부터 위험원을 분석해, ISO 26262에서 요구하는 안전 활동을 준수하는 과정을 소개하였다.

Abdulkhaleq[11]는 신뢰성 기반 분석 방법론을 극복하기 위한 방안으로 STPA의 분석 결과로 도출된 안전 요구사항을 LTL(Linear Temporal Logic)로 명세하여 시스템의 안전성을 정형 검증하는 방안을 제안하였다. 또한, Abdulkhaleq 등[12]은 완전 자율주행 차량의 안전 설계를 위해 ISO 26262의 안전 활동에 따라 STPA를 적용한 방안을 설명하였다. Jeong 등[13]은 STPA와 FMEA를 각각 위험원 분석과 안전 분석에 나누어 적용하여, AK2 초음파 감지기의 안전 활동을 수행하는 방안을 설명하였다.

R.S. Martínz, S.R. Do, A. Abdulhaleq, 그리고 D.H. Jeong까지 이들의 연구는 소프트웨어 중심으로 기능의 제어를 수행하는 SDV에서 안전성을 확보하기 위해, 시스템 이론적 분석 방법론의 STPA를 활용하는 방안을 소개하고 있다. 이와 같은 새로운 접근법은, SDV의 안전성이라고 하는 자동차 분야의 도전과제를 해결하기 위한 이론적 기반을 선보이고 있다.

본 논문에서는 앞선 선진연구를 바탕으로 소프트웨어 정의 조향 제어기를 위한 자동차 안전공학 전략을 제안한다. 특히 STPA를 수행하기 위해 핵심이 되는 CSD(Control Structure Diagram)를 기능(Function), 행동(Behavior), 그리고 구조(Structure)의 세 가지 관점으로 모델링하고, 각 관점 별로 안전 활동을 위한 분석 기법을 적용하여 기능안전과 운용안전을 달성한다. 이를 통해 시스템 이론적 분석 방법론과 그 기법인 STPA를 소개하거나 차량 제어기를 대상으로 사례 적용하는데 그치지 않고, 안전을 고려한 시스템의 설계부터 시작하여 위험원 분석 및 안전 분석을 통해 SDV의 안전성을 검증할 수 있는 MBSE(Model-based Safety Engineering) 체계를 구성하고자 한다.

2. 배경지식

2장에서는 본 논문에서 제안하고자 하는 자동차 안전공학 전략을 소개하기에 앞서 필요한 배경지식을 설명한다. 2.1절에서는 시스템 이론적 분석 방법론의 기법 중 하나인 STPA의 개념과 절차에 대해 기술한다. 이어서 2.2절에서는 ISO 26262와 ISO 21448에서 요구하는 안전 활동의 개요를 정리하여 설명한다.

2.1 STPA: 제어 구조와 UCA 분석

STPA는 시스템 이론적 분석 방법론에 기반을 둔 대표적 위험원 분석 방법으로, 시스템 구성요소 간의 상호작

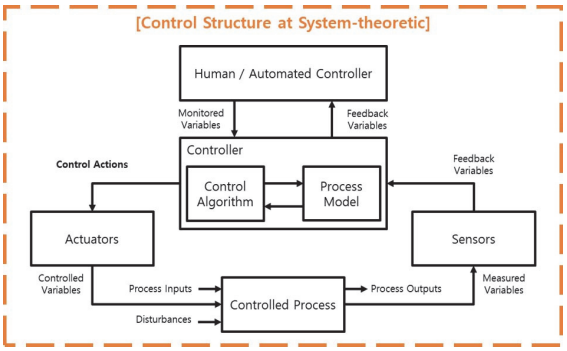

Fig. 2 Concept of control structure at system-theoretic

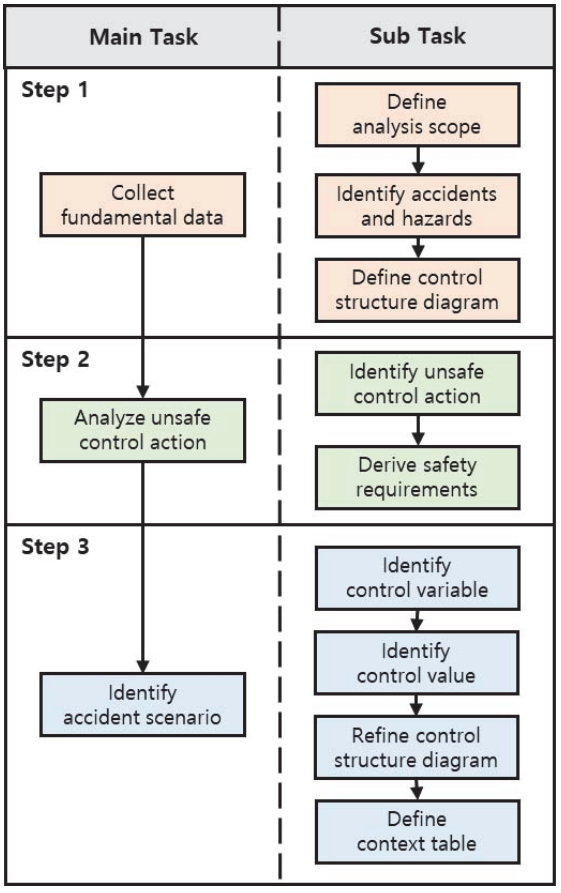

Fig. 3 Process of system theoretic process analysis

용을 중심으로 발생가능한 잘못된 제어로부터 사고를 유발할 수 있는 위험원을 분석한다. Fig. 2의 제어 구조(Control structure)는 STPA를 수행하기 위한 필수적인 요소[14,15]로, 시스템의 구성요소를 감지기, 제어기(Controller), 작동기(Actuator), 그리고 제어되는 대상(Controlled process)으로 폐쇄 시스템(Closed-loop system)을 구성하고 이들 사이에서 시스템을 구동하는 제어 명령(Control action) 및 피드백(Feedback)을 표현한다. 특히 제어기 내 작성되는 제어 모델은, 시스템의 현재 상태와 그 조합으로부터 제어를 위한 명령이 실행되는 과정까지 고려하도록 돕는다. 이를 통해 분석가는 상호작용과 제어, 두 가지 관점에서 시스템이 가질 수 있는 위험원을 면밀하게 분석하여 방지하기 위한 안전 요구사항을 개발할 수 있다.

Fig. 3과 같이 STPA를 수행하기 위한 수행 절차는 크게 기초 자료 수집, UCA(Unsafe Control Action) 분석, 그리고 사고 시나리오 식별의 세 단계[14,15]로 구성된다. 기초 자료 수집에서는 본격적인 분석을 수행하기에 앞서 분석가의 시스템 이해도를 높이게 된다. 분석을 수행하기 위한 범위를 선정하여 시스템으로부터 발생할 수 있는 사고와 위험원을 식별하고, 시스템의 초기 제어 구조를 CSD로 모델링한다. 이어지는 UCA 분석에서는 기초 자료 수집에서 파악한 내용을 바탕으로, 잘못된 제어 명령을 방지하기 위한 안전 요구사항을 도출한다. 이때 제어 명령 별로 'Wrongly Provided', 'Not Provided', 'Provided Wrong Timing or Order', 그리고 'Stopped Too Soon or Applied Too Long'의 네 가지 안내어(Guide-words)를 적용하여 UCA가 식별된다. 마지막 사고 시나리오 식별에서는 UCA에서 도출한 초기 안전 요구사항을 구체화한다. 제어 변수(Process variable)와 제어 값(Process value)를 포함하는 제어 모델을 식별하여 CSD를 재 작성하고, 상황표(Context table)를 작성해 초기 수준의 안전 요구사항을 재 작성하거나 추가한다.

2.2 자동차 안전 활동: 기능안전 및 운용안전

ISO 26262와 ISO 21448은 자동차 분야의 기능안전 및 운용안전 표준으로, 사용자에게 다양한 기능을 제공하는 SDV에서 소프트웨어 오류로 인한 사고를 예방하고 안전성을 확보하도록 돕는다. ISO 26262를 통해 시스템 내부의 고장에 대한 안전을 다루고, ISO 21448을 통해 기능의 한계점이나 사용자의 잘못된 제어에 대한 안전을 다룸으로써, SDV의 기능에 대한 의도하지 않은 부분 뿐만 아니라 의도한 부분의 안전까지 고려할 수 있게 한다.

ISO 26262와 ISO 21448에서 안전성 확보를 위해 공통적으로 수행을 요구하는 것은 Fig. 4의 안전 활동이다. ISO 26262의 경우 안전 활동을 통하여 정해진 목표치까지 시스템의 기능안전이 보증되었는지 정성적/정량적으로 검증한다. 마찬가지로 ISO 21448에서는 안전 활동을 통해, 사고를 발생시키는 수많은 시스템의 시나리오를 대상을 지속적/반복적으로 운용안전이 보증되고 있는지 확인한다. 이러한 안전 활동의 결과로 개발자는 SDV의 위험원을 찾아 그 해결 방안을 시스템에 구현하고 검증하게 된다.

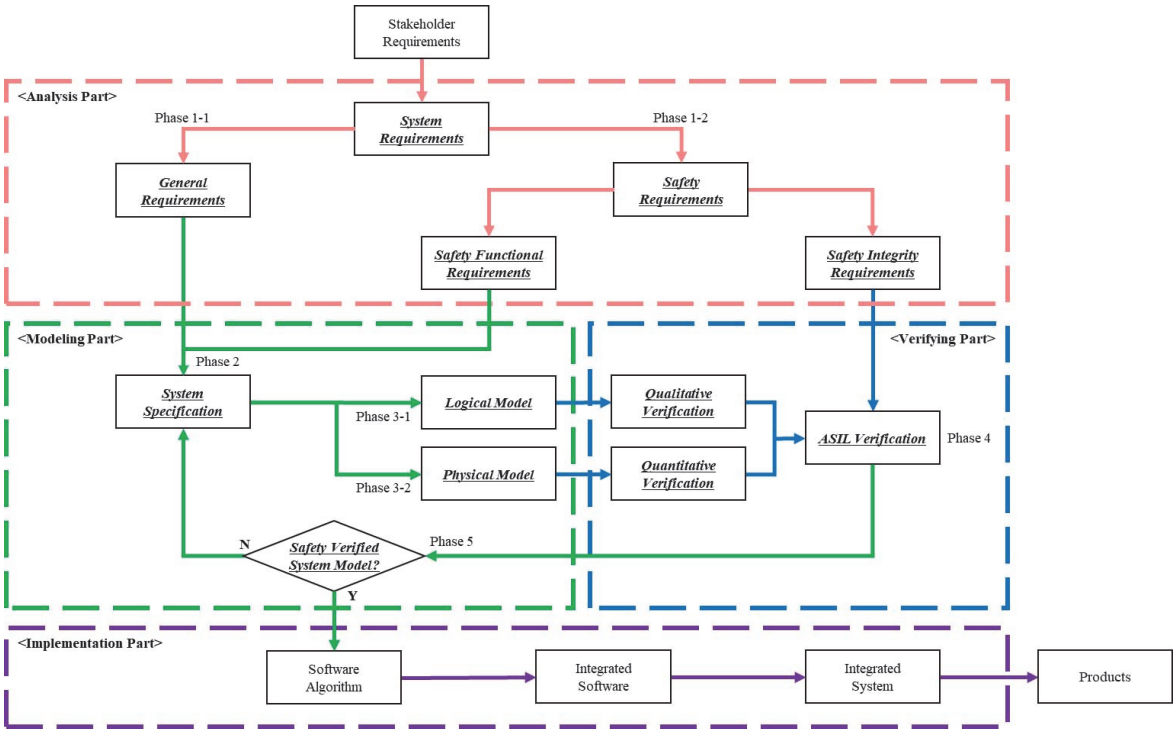

Fig. 4 Safety activities for developing of SDVs

예를 들어 SDV의 조향 시스템 개발에 ISO 26262와 ISO 21448을 적용한다고 해보자. ISO 26262 측면에서는 시스템 내부의 감지기 고장이나 소프트웨어 오류로 인한 문제를 해결하기 위해 필요한 안전 기능을 구현하고, 이 안전 기능의 안전 무결성을 달성하기 위하여 고장 전파(Failure propagation) 관계, 종속 고장(Dependent failure) 관계, HAM(Hardware Architecture Metric), PMHF (Probabilistic Metric for Random Hardware Failure) 등의 사항을 검증한다. 그리고 ISO 21448 측면에서는 사용자가 차량의 조향(Steering) 기능을 잘못 조작할 가능성이나 감지기의 감지 가능 범위 등 기능적 한계점으로 인한 문제를 제어 또는 제거하기 위해 필요한 안전 기능을 구현하고, 정말로 위험하다고 분석된 상황에 대하여 시스템이 명확하게 대응하고 있는지 MIL(Model in the Loop), SIL(Software in the Loop), PIL(Processor in the Loop), HIL(Hardware in the Loop)을 비롯한 다양한 시뮬레이션으로 확인한다. 이 모든 과정이 완료되었을 때 사용자가 믿고 사용할 수 있는 안전한 SDV가 개발된다.

앞서도 소개하였듯이, 2.1절에서 소개한 STPA는 ISO 26262 및 ISO 21448에서 요구하는 안전 활동을 지원하는 분석 방법 중 하나이다. 특히 CSD라고 하는 모델을 중심으로 분석을 전개하는 STPA는 일원화된 개발 절차 내에서 서로 다른 목표를 가진 SDV의 두 가지 안전성, 기능안전과 운용안전을 달성할 수 있게 돕는다. 이어지는 3장에서는 이러한 STPA의 특성을 활용하여 본 논문에서 제안하는 시스템 이론적 관점을 적용한 모델 기반 접근 방식의 자동차 안전공학 전략을 자세히 기술한다.

3. 시스템 이론적 모델 기반 자동차 안전공학 전략

본 연구에서는 Fig. 5와 같이 자동차 안전공학을 위한 MBSE 체계를 제안한다. 제안하는 MBSE 체계는 크게 모델링 활동과 안전 활동의 두 가지로 구분된다. 특히 개발하고자 하는 시스템의 제어 구조를 표현하기 위한 STPA의 CSD를 시스템이 제공하는 기능, 시스템이 동작하는 행동, 그리고 시스템이 구성되는 구조의 세 가지 관점으로 구체화 모델링하고, 각 단계의 목적과 성격에 부합하는 안전공학 기술을 적용하여 안전 활동을 수행한다. 여기서 기능과 행동, 그리고 구조의 세 가지 관점은 SDV와 같이 복잡한 시스템을 논리적으로 표현하기에 알맞은 모델링 방법[16]으로, 품질 높은 SDV를 개발하기 위한 사실상 표준(De facto standard) A-SPICE[17]에서 요구하는 시스템 설계 과정에도 적합하다. 제안하는 전략은 3.1절의 예비 제어 구조 설계, 3.2절의 시스템 이론적 위험원 분석, 3.3절의 상세 제어 구조 설계, 그리고 3.4절의 신뢰성 기반 계층적 안전 분석까지 총 네 단계가 순서대로 수행된다.

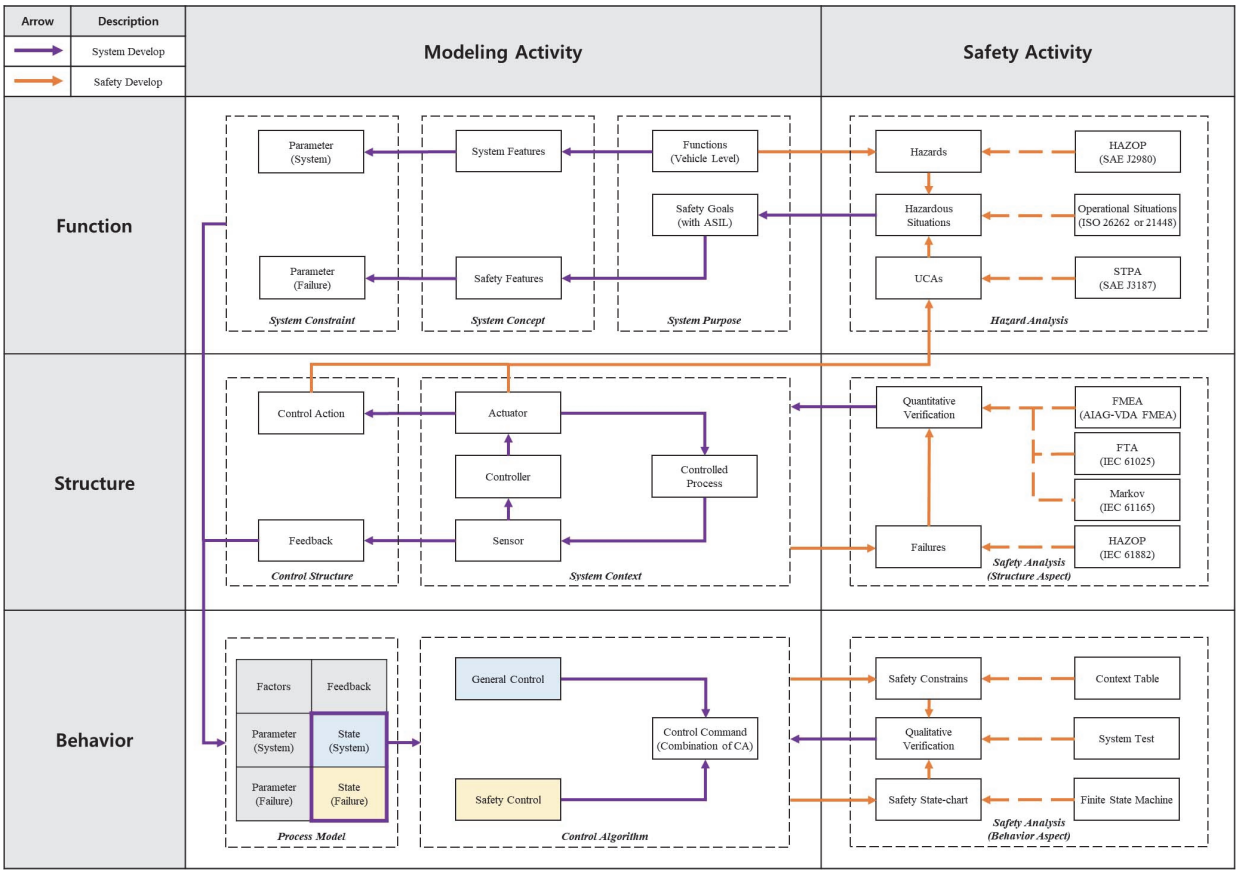

Fig. 5 Proposed automotive safety engineering strategy: Model-based safety engineering framework

3.1 예비 제어 구조 설계

제안하는 MBSE 체계의 첫 단계는, 기능과 구조의 두 가지 관점에서 시스템을 분석해 예비 제어 구조를 설계하는 것에서부터 시작한다. 먼저 기능 관점에서는, 차량 수준에서 시스템이 제공하는 기능을 파악한다. 이를 통해 차량을 구성하는 수많은 구성요소 중에서 대상 시스템이 수행해야 할 주요 목적과 역할이 식별된다. 예를 들어 SDV를 구성하는데 필요한 필수 시스템 중 하나인 EPS의 목적은 조향으로, 운전자의 의도에 따라 EPS가 차량에 제공하는 기능이 된다.

두번째 구조 관점에서는, 전원 또는 다른 제어기 등의 주변 환경을 고려하여 시스템의 전반적인 구조를 구성한다. ISO 26262 및 ISO 21448에서 시스템의 구조로 정의하고 있는 최소 1개 이상의 감지기와 최소 1개 이상의 제어기, 그리고 최소 1개 이상의 작동기와 함께 이들로 실제 제어되는 대상까지, 네 가지 요소를 기준으로 작성한다. 이때 감지기와 작동기는 외부의 타 시스템과 통신을 수행하는 인터페이스(Interface)를 포함하고, 제어기는 시스템의 상태를 보여주는 제어 모델과 이를 조합하여 제어를 수행하는 제어 알고리즘으로 구성된다.

두 가지 관점을 통해 차량 수준의 기능과 전반적인 시스템의 구조가 분석되고 나면, 이들 사이의 상호작용 관계로부터 제어 구조를 명세할 수 있다. 제어기로부터 작동기로 전달되는 제어 명령과 감지기를 통해 제어기로 입력되는 피드백은, 분석된 시스템의 구성요소들 사이에서 만들어지는 폐쇄(Closed-loop)된 형태의 제어 구조를 통해 차량 수준의 기능이 사용자에게 제공되도록 한다.

제어 구조까지 식별하고 나면 예비 제어 구조 설계 단계가 종료된다. 이때 두 가지 관점을 통해 설계된 예비 제어 구조의 세 가지 요소는, 다음과 같은 집합 구성을 가지는 CSD로 표현할 수 있다.

1) Vehicle Func. = $\{func_{[vehicle,1]},\cdots,func_{[vehicle,n]}\}$
2) System Structure = (S, C, A, CP)
   ① S(Sensor)
   $= \{s_1,\cdots,s_n\} \cup$ Comm. Interface
   ② C(Controller)
   = Control Algorithm ∪ Process Model
   ③ A(Actuator)
   $= \{a_1,\cdots,a_n\} \cup$ Comm. Interface

④ CP(Controlled Process)
= {cp_1,⋯,cp_n}
3) Control Structure = (CA, F)
① CA(Control Action)
= {ca_1,⋯,ca_n} ∪ {Comm. Tx}
② F(Feedback)
= {f_1,⋯,f_n} ∪ {Comm. Rx}
4) External Interface
(i.e., Battery, Other E/E Systems, etc.)

3.2 시스템 이론적 위험원 분석

개발하고자 하는 시스템에 대한 예비 제어 구조가 작성된 후에는, MBSE 체계의 두번째 단계인 시스템 이론적 위험원 분석 단계가 수행된다. 제어 구조를 기반으로 위험원과 UCA, 그리고 위험한 사건(Hazardous event)까지 사고를 유발할 수 있는 잠재적 요소를 식별 및 분석하여, 예방 또는 제어하기 위한 초기 안전 요구사항을 도출한다.

잠재적 요소의 첫번째인 위험원의 식별에는, 예비 제어 구조를 통해 파악한 차량 수준의 기능을 대상으로 HAZOP(HAZard and OPerablility study) 기법을 적용한다. HAZOP은 ISO 26262 및 ISO 21448에서 위험원 식별을 위해 권장하고 있는 분석 기법으로, 분석 대상과 안내어를 조합하여 분석을 수행한다. 특히 본 연구에서는 특별히 차량 시스템에 적합하도록 SAE J2980[18]에서 정의한 안내어를 사용하였다. 예를 들어 안내어 중 하나인 'Loss of Function'과, 차량이 제공하는 기능인 조향을 조합하면 '조향 기능의 상실'이라는 위험원을 식별할 수 있다. 그 과정을 정리하면 Table 1과 같이 표현할 수 있다.

두번째로 식별되는 잠재적 요소는 UCA이다. UCA 식별에는 예비 제어 구조 설계 단계에서 구성 및 명세하였던 시스템 구조와 제어 구조 두 가지를 활용한다. 서로 연관된 작동기와 제어 구조를 하나의 묶음으로 목록화하고, 각각에 STPA의 네 가지 안내어[14,15]를 적용한다. 예를 들어 EPS의 작동기인 모터(Motor)와 그 제어 명령 토크(Torque) 출력을 하나의 묶음이라고 할 때, 안내어 중 하나인 'Providing'을 적용하면 '전제 조건이 만족되지 않았지만 토크 출력이 모터에 제공된다'의 UCA를 식별할 수 있다. Table 2는 UCA를 식별하는 과정을 보여준다.

마지막 위험한 사건은, 식별된 두 가지 잠재적 요소로부터 조합하여 분석한다. 각 UCA 별로 ISO 26262의 운행 조건(Operational situations) 또는 ISO 21448의 경계 사례(Edge cases) 조건을 적용하여 하나의 위험한 운행 상황을 생성하고, 이 상황이 어떠한 위험원에 영향을 미치는지 분석한다. 예를 들어 '겨울의 고속도로에서 100 km/h로 주행 중'의 운행 조건 아래에서, '전제 조건이 만족되지 않았지만 토크 출력이 모터에 제공된다'는 UCA가 발생하게 되면, '조향 기능의 상실'이라는 위험원이 발생하게 됨을 분석할 수 있다. 위험한 사건의 과정을 정리하면 Table 3과 같다.

Table 1 Hazards identification at system-theoretic hazard analysis

Hazards Identification	Guide-words (SAE J2980)					
	Loss of function	More than requested	Less than requested	Unintended activation	Activation in opposite direction	Output stuck at a value
Vehicle Func.	[Hazard[= Loss of [Vehicle Func.] function	[Hazard] = [Vehicle Func.] more than requested	[Vehicle Func.] less than requested	Unintended [Vehicle Func.] activation	[Vehicle Func.] activation in opposite direction	Output of [Vehicle Func.] stuck at a value

Table 2 UCAs identification at system-theoretic hazard analysis

UCAs identification		Guide-words (SAE J3187 or STPA Handbook)						
		Not providing	Providing	Too soon	Too late	Out of order	Stopped too soon	Applied too long
CA (Control action)	A (Actuator)	[UCA] = Trigger conditions of [CA] are fulfilled, but it's not provided to the [A]	[UCA] = The situation is difference with trigger conditions of [CA], but it's provided to the [A]	[UCA] = Trigger conditions of [CA] are not fulfilled, but it's provided to the [A] too soon more than defined time	[UCA] = Trigger conditions of [CA] are fulfilled, but it's provided to the [A] too late more than defined time	[UCA] = [CA] is provided to the [A] differently from the order in which it was defined	[UCA] = [CA] provided to [A] was stopped too soon more than the defined time	[UCA] = [CA] provided to [A] was applied too long more than the defined time

Table 3 Analyzing hazardous events at system-theoretic hazard analysis

Hazardous events analyzing	Operational situations or edge cases		Which hazard is affected?
	Conditions about operational situations (i.e., Driving conditions and driving situations)	Conditions about edge cases (i.e., Performance limitations and misuse scenarios)	
UCA	[Hazardous Event] = Combination of operational situations or edge cases with of each [UCA]		Choose the one from [Hazard] what [Hazardous Event] affect to it

위험원, UCA, 그리고 위험한 사건이 모두 파악되면 이를 제어하기 위한 초기 안전 요구사항(Safety goals)을 도출한다. 이때 안전 요구사항은 위험원으로부터 도출되는 안전 기능 요구사항(Safety functional requirements)과 ISO 26262에 명시된 기준에 따라 결정되는 안전 무결 요구사항(Safety integrity requirements)으로 구분된다.

3.3 상세 제어 구조 설계

상세 제어 구조 설계 단계에서는, 시스템 이론적 위험원 분석 단계에서 도출된 초기 안전 요구사항을 고려하여 안전 시스템이 설계될 수 있도록 기능과 행동의 두 가지 관점에서 살펴본다. 우선 기능 관점을 통해 일반 기능과 안전 기능의 두 가지를 시스템 수준에서 설계한다. 일반 기능의 경우 제어기가 제어되는 대상을 통해 시스템 수준에서 제공하는 기능을 의미한다. 예를 들어 EPS의 작동기인 모터가 토크 출력이라고 하는 제어 명령을 제공한다고 할 때, 제어기는 토크 보조라고 하는 시스템 수준 기능을 제공하여 모터가 토크 출력에 따라 작동할 수 있도록 한다.

일반 기능이 예비 제어 구조 설계에서 작성한 사항을 구체화하는 것이라면, 안전 기능은 시스템 이론적 위험원 분석 단계의 결과인 초기 안전 요구사항을 실현이 가능한 수준으로 상세화 한 것이다. 예를 들어 토크 오류 모니터링(Error monitoring)은 EPS의 초기 안전 요구사항인 '조향 기능의 상실을 방지한다'로부터 설계한 것으로, EPS의 제어기가 작동기 중 하나인 모터에 제공하는 제어 명령을 토크 출력이라고 할 때, '전제 조건이 만족되었지만 토크 출력이 모터에 제공되지 않는다'라는 UCA를 안전하게 제어하기 위한 시스템 수준의 기능이다.

기능 관점을 통해 두 가지의 시스템 수준 기능이 파악되고 나면, 다음은 행동 관점에서 상세 제어 구조를 설계하기 위해 제어 모델의 상세화가 필요하다. 제어 알고리즘(Algorithm)에 의해 제어 모델의 조합으로부터 제어 명령을 도출하는 과정이 시스템의 행동을 나타내기 때문이다. 특히 제안하는 MBSE 전략에서는 제어 모델에 필수적으로 정의되어야 하는 요소를 매개변수(Parameter)와 상태(State)로 설정하였다. 여기서 매개변수는 시스템 튜닝(Tuning) 작업 시 개발자가 변경할 수 있는 데이터(Data)로, 시스템 수준의 일반 기능 및 안전 기능으로부터 정의된다. 다른 하나인 상태는 시스템에 의해 결정되는 데이터로, 감지기로부터 유입되는 피드백과 매개변수 값의 변화에 따라 매번 변경된다.

예비 제어 구조와 마찬가지로, 상세 제어 구조 또한 기능 관점을 통해 차량 기능을 구체화한 일반 기능 및 안전 기능과, 행동 관점으로부터 제어 모델을 구체화한 매개변수와 상태의 두 가지 요소는, 다음의 집합 구성을 가지는 CSD로 표현된다. 그리고 Fig. 6.은 예비 제어 구조와 상세 제어 구조에서 도출된 두 개의 집합 구성으로부터 최종적으로 작성된 CSD를 나타낸다.

1) System Func. = (SyF, SaF)
   ① SyF = $\{func_{[system,1]},\cdots,func_{[system,n]}\}$
   ② SaF = $\{func_{[safety,1]},\cdots,func_{[safety,n]}\}$
2) Process Model = (P, S)
   ① P = $\{p_1,\cdots,p_n\}$
   ② S = $\{q_1,\cdots,q_n\}$
3) System Behavior = (SyC, SaC)
   ① SyC = $\{cmd_{[system,1]},\cdots,cmd_{[system,n]}\}$
   ② SaC = $\{cmd_{[safety,1]},\cdots,cmd_{[safety,n]}\}$

3.4 신뢰성 기반 계층적 안전 분석

자동차 안전공학을 위한 MBSE 체계의 마지막 단계는 신뢰성 기반 계층적 안전 분석이다. 안전 분석은 위험원 분석 결과에 따라 시스템이 안전하게 개발되었는지 검증하는 단계로, 안전성의 확보가 불완전할 경우 추가적인 안전 기능을 시스템에 구현하여 불안전 요소를 제거하도록 돕는다. 안전 분석 단계는, 두 개의 설계 단계를 거쳐 작성된 시스템 제어 구조를 기반으로, 정량적 측면과 정성적 측면의 두 가지 유형으로 수행한다.

정량적 측면의 안전 분석은, 예비 제어 구조 단계에서 구조 관점을 통해 파악하였던 시스템 전반적 구조를 기반으로 한다. 그 중에서도 감지기와 제어기, 그리고 작동기와 같이 전체 가용시간 내에서 고장이 날 확률이라는 특성을 가지는 하드웨어적 구성요소를 분석 대상으로 한다.

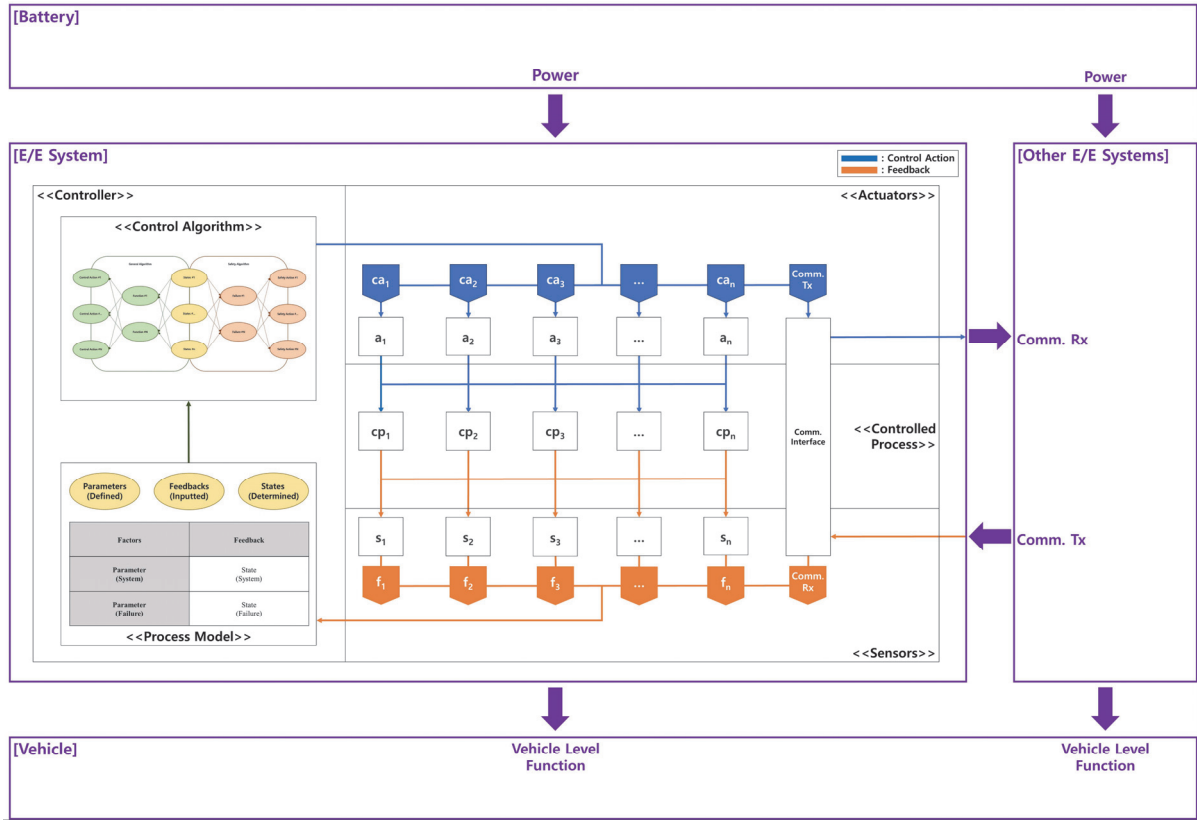

Fig. 6 Control structure diagram for model-based safety engineering framework

Table 4 Failure identification at reliability-based hierarchical safety analysis

Failures identification	Guide-words (IEC 61882)										
	No or Not	More	Less	As Well As	Part of	Reverse	Other than	Early	Late	Before	After
System structure = (S, C, A, CP)	Trigger conditions are met, but [System structure] does not provide the function	[System structure] provides excessive functions beyond the specified performance	[System structure] provides functions that fall short of the specified performance	[System structure] provides functions other than those specified	[System structure] provides only some of the specified functions	Although trigger conditions are not met, the function is provided in [System structure]	[System structure] provides a different function than specified	[System structure] provides functions faster than the designated time	[System structure] provides functions slower than the designated time	[System structure] provides functions faster than the designated order	[System structure] provides functions slower than the specified order

특히 시간이 지남에 따라 마모 등으로 인해 그 수명이 짧아지고 고장률은 높아지는 하드웨어적 구성요소는 FMEDA(Failure Modes, Effect, and Diagnostic Analysis)와 FTA, 그리고 Markov와 같이 전통적으로 수행해왔던 신뢰성 기반 분석 방법론의 기법들을 통해 안전 무결 요구사항이 충족되는지 검증[1,3] 한다. 이를 위해 Table 4와 같이 IEC 61882[19] 기반으로 제어 구조 설계를 통해 파악한 구조 관점의 시스템 구성요소로부터 고장을 식별하는 과정이 선행되어야만 한다.

정성적 측면에서의 안전 분석은, 상세 제어 구조 단계에서 행동 관점에서 파악하였던 제어 모델을 기반으로 한다. 시스템 이론적 위험원 분석 단계에서 식별하였던 모든 위험한 사건에 대하여 관련된 고장 상태가 발견되더라도, 시스템의 행동에 따라 제어 명령이 수행되어 안전 상태가 유지되는지 확인한다.

ISO 21448[2]에서 요구하고 있는 바와 같이, 운용안전의 달성을 위해 네 단계를 거쳐 분석해 온 위험한 상황들에 대하여 시스템이 명확하게 인지하고 안전하게 제어를 하고 있는지 확인하는 것이다. 이를 위해 본 연구에서는 고장 상태 q_n이 정상을 나타내는 V_{normal}과 비 정상을 나타내는 $V_{abnormal}$ 두 개 값을 갖고, 비 정상 상태에 대한 행동 시 준수해야 하는 ISO 26262의 네 가지 시간 제약사항[1] FTTI(Fault Tolerant Time Interval), FDTI(Fault Detection Time Interval), FRTI(Fault Reaction Time Interval), EOTI(Emergency Operation Time Interval)를 각각 t_{FTTI}, t_{FDTI}, t_{FRTI}, t_{EOTI}으로 표현하였을 때, q_n의 값이 $V_{abnormal}$로 변경되는 순간의 행동에 대한 5가지 안전 제약사항을 정의해 정성적 측면의 안전 분석을 수행한다.

1) t_{FDTI}, t_{FRTI}, 그리고 t_{EOTI}까지 세 가지 시간 제약사항의 합은 t_{FTTI} 보다 작아야만 한다.
2) 피드백이 정의된 매개변수 범위 내에 있을 경우, 제어기는 t_{FDTI}의 시간 이내에서 q_n의 값을 $v_{abnormal}$로 변경해야 한다.
3) q_n의 값이 $v_{abnomal}$로 변경되는 순간, 제어기는 t_{FRTI} 시간 이내에서 시스템 행동에 따른 제어 명령을 실행해야 한다.
4) 시스템 행동에 따라 제어 명령이 실행된 순간부터 최소 t_{EOTI} 동안은 안전 기능이 수행되어야만 한다.
5) q_n의 값이 $v_{abnomal}$로 변경되는 순간부터 t_{FDTI}, t_{FRTI}, 그리고 t_{EOTI}까지 세 가지 시간 제약사항의 합 만큼 시간이 지난 후에는, 반드시 시스템 상태가 안전 상태로 전환되어 있어야만 한다.

정량적 측면 및 정성적 측면 두 가지 유형의 안전 분석까지 모두 마치고 나면, 제안하고자 하는 자동차 안전공학을 위한 MBSE 체계의 모든 활동이 종료된다. 다음 장에서는 제안하는 방법의 실효성 및 유효성을 검토하기 위하여 각 단계를 실제 사례에 적용해 본다.

4. 소프트웨어 정의 조향 제어기

본 장에서는 실제 차량에서 사용하고 있는 HL만도의 EPS, 그 중에서도 ECU(Electronic Control Unit)를 대상으로 선정하여 3장에서 기술한 MBSE 체계를 적용한 사례를 다룬다. EPS ECU는 SDV를 구성하는 필수적인 구성요소의 하나로, 운전자의 조작 명령 또는 자율주행의 제어 명령에 따라 SDV의 조향을 제공하는 소프트웨어 정의 조향 제어기이다. 특히 HL만도에서는 시스템의 위험원으로 인해 ECU가 작동하지 않거나 성능이 저하되는 등의 다양한 위험한 사건 속에서도 SDV의 안전성이 확보될 수 있도록 이중화 설계를 적용하여 EPS ECU를 개발하고 있다.

4.1 EPS ECU의 예비 제어 구조

HL만도의 EPS ECU는 운전자의 조작 명령 또는 자율주행의 제어 명령에 따라, 동력 조향 보조 및 자율주행 보조라고 하는 두 가지의 차량 수준 기능을 제공한다.

EPS ECU는 다양한 구성요소가 감지기, 제어기, 그리고 작동기로써 구성되어 있다. 감지기로는 1개의 온도 감지기 인터페이스, 4개의 토크 감지기 인터페이스, 2개의 전류 감지기 인터페이스, 그리고 3개의 모터 위치 감지기 인터페이스가 사용된다. 제어기로는 1개의 MCU(Micro Control Unit)가 사용되며, 작동기로는 1개의 게이트 드라이버(Gate driver)와 1개의 인버터(Invertor)로 구성된 모터 드라이버(Motor driver)가 사용된다.

EPS ECU의 감지기, 제어기, 그리고 작동기는 제어 명령 모터 토크를 제어되는 대상인 EPS에 제공하여 동력 조향 보조 및 자율주행 보조가 차량 수준에서 제공될 수 있도록 한다. 모터 토크가 제공되고 나면 각 감지기들은 각각 내부 온도, 조향 토크, 모터 전류, 그리고 모터 위치까지 4개의 피드백을 제어기로 전달하게 된다.

EPS ECU는 이중화 설계가 적용되어 마스터(Master)와 슬래이브(Slave)의 두 개 ECU가 전체 EPS를 구성하고 있다. 이를 위해 두 개의 ECU는 CAN(Controller Area Network)을 통해 서로의 상태를 확인하여 항시 최소 하나의 ECU는 가용할 수 있도록 할 뿐만 아니라, EPS의 현재 상태를 전달하거나 제어에 필요한 데이터를 전송 받을 수 있도록 SDV의 다른 전기/전자(Electrical/Electronic) 시스템과 통신을 수행한다.

4.2 EPS ECU의 위험한 사건과 안전 요구사항

EPS ECU의 위험원은, 동력 조향 보조와 자율주행 보조의 두 가지 차량 수준 기능에 SAE J2980[18]의 6가지 안내어를 조합한 후 인스펙션(Inspection) 및 워크스루(Walk-through)를 통해 실제로 EPS에서 발생한 경우를 식별하였다. 식별 결과, 동력 조향 보조로부터 5개의 위험원을, 자율주행 보조로부터 1개의 위험원을 식별하였다.

EPS ECU의 UCA는 작동기 역할을 수행하는 모터와 그 제어 명령인 모터 토크, CAN 통신을 제공하는 인터페이스와 그 제어 명령인 통신 송신을 대상으로 STPA를 위한 7가지 안내어[14,15]를 조합한 후, 위험원과 마찬가지로 인스펙션 및 워크스루를 수행하였다. 수행 결과, 모터 및 모터 토크로부터 6개의 UCA를, CAN 통신 인터페이스와 통신 송신으로부터 1개의 UCA를 식별하였다.

EPS ECU에서 발생할 수 있는 위험한 사건을 식별하기 위해 주행 조건, 장소 조건, 도로 조건, 그리고 환경 조건의 네 가지 운행 조건과 DDT(Dynamic Driving Task), OEDR(Object and Event Detection and Response), DDT

Table 5 Safety goals(safety requirements at draft) of EPS ECU at HL Mando

ID	Safety goals	ASIL
SG_1	Sudden loss of assistance shall be prevented (Include case of insufficient steering)	D
SG_2	Unintended steering shall be prevented (Include case of excessive steering and reverse steering)	D
SG_3	Unintended wheel motion shall be prevented (Include case of wheel oscillation)	D
SG_4	Lock steering shall be prevented	D
SG_5	Loss of ADAS function shall be prevented	D

Fall-back, ODD(Operational Design Domain)의 자율주행 수준에 따른 네 가지 요소를 조합하였다. 여기서 자율주행 수준을 위한 네 가지 요소는 SAE J3016[20]에서 정의하고 있는 내용을 기준으로 하였다. 그 결과, 동력 조향 보조와 관련된 UCA에서는 931건의 위험한 사건이 식별되었고, 자율주행 보조와 관련된 UCA에서는 36가지의 위험한 사건이 식별되었다. 그리고 전체 967건의 위험한 사건으로부터 Table 5와 같이 ASIL(Automotive Safety Integrity Level) D 수준의 안전 무결 요구사항을 가지는 총 5개의 초기 안전 기능 요구사항인 안전 목표(Safety Goal)가 식별되었다. 안전 목표는 EPS ECU가 분석을 통해 안전 기능을 구현하여 정말 위험원이 충분히 제거되었는지 확인해야 하는 5개의 개발 지표로 사용된다.

4.3 EPS ECU의 상세 제어 구조

EPS ECU에는 다양한 요구사항들이 기능으로써 구현된다. EPS를 구동하기 위한 기본 기능(Basic functions)과 부가 기능(Value added functions), 협조제어 기능(Integrated control functions), 감지기로부터 입력된 피드백을 제어하거나 작동기의 제어 명령을 수행하기 위한 외부 처리 기능 (External processing functions), 시스템이 가진 여러 요소에 대해 2차적인 연산을 수행하는 내부 처리 기능(Internal processing functions)이 EPS ECU의 기능으로써 구현된다. 여기서 외부 및 내부 처리 기능은 다른 전기/전자 시스템과의 통신, EPS를 구성하는 ECU 간의 통신, 차량의 고장에 대한 진단이나 소프트웨어 업데이트를 수행하는 진단 도구와의 통신에 대한 요구사항들을 포함한다.

EPS ECU의 안전 기능은 위험원 분석을 통해 식별된 ASIL D 수준의 안전 기능 요구사항을 시스템 수준으로 상세화 한 것이다. 특히 안전 기능은 일반적인 기능들에 대한 모니터링에 따라 시스템이 최종적으로 안전 상태로 갈 수 있도록 제어한다.

안전 기능은 크게 시스템을 구성하는 구조적 요소들에 대한 고장을 모니터링 하여 제어하는 안전 기능, 기능적 요소들에 대한 고장을 모니터링 하여 제어하는 안전 기능, 통신을 통해 송신 및 수신되는 데이터들에 대한 무결성을 모니터링 하여 제어하는 안전 기능으로 나뉘어

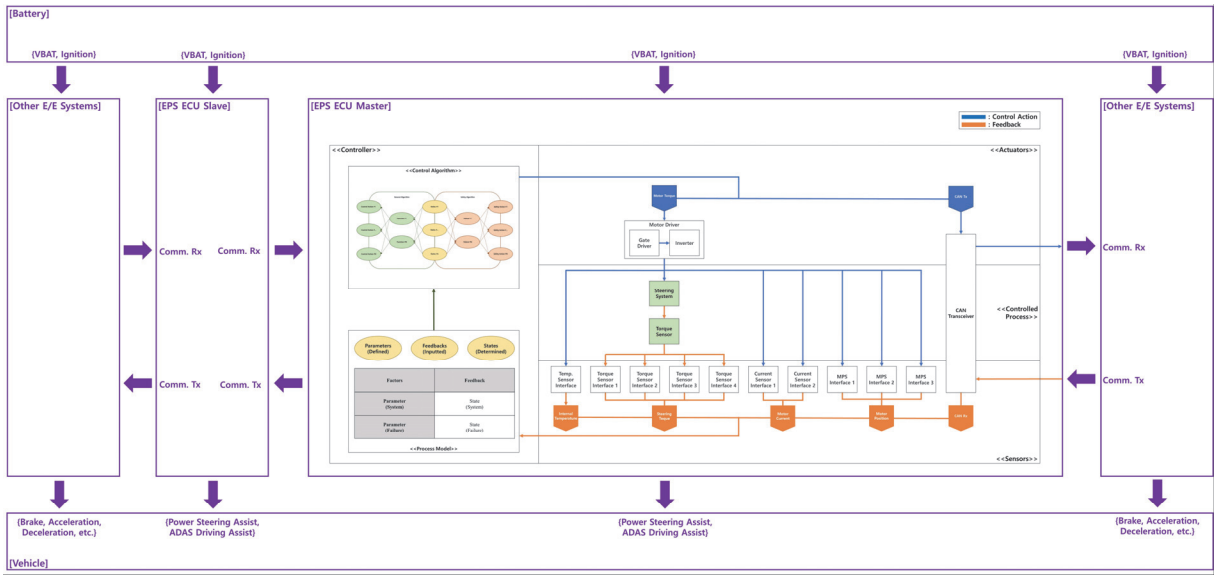

Fig. 7 Overall control structure of EPS ECU at HL Mando

진다. 여기서 안전 기능은 EPS ECU의 안전 상태를 유도해 나아가는 시스템 성능 저하 전략 및 고장 알람 전략의 두 가지 안전과 관련된 행동에 의해 제어된다.

Fig. 7은 예비 제어를 포함하여 EPS ECU의 전체적이 제어 구조를 표현한 것이고, Fig. 8에서 작성된 상태 다이어그램(State-chart diagram)은 EPS ECU의 시스템 성능 저하 전략 및 고장 알람 전략을 표현하는 예시의 하나로 시스템이 안전 기능을 통해 안전 상태로 천이 되어 가는 과정을 보여준다.

4.4 EPS ECU의 정량적 및 정성적 안전성 검증

EPS ECU의 안전성은 정량적 측면과 정성적 측면 두 가지 관점에서 검증하였다. ISO 26262[1]에서는 ASIL D를 가지는 시스템에 대해 고장 전파 관계 및 종속 고장 관계를 고려하여 HAM과 PMHF의 두 가지 정량적 지표가 달성되는지 평가하도록 요구하고 있다.

본 논문에서는 정량적 안전 분석을 수행하기 위한 다양한 신뢰성 기반 분석 방법론의 기법들 중에서도 안전 분석 도구 중의 하나인 medini analyze의 FTA 및 FMEDA를 통합한 방안[21]을 사용하였다. FTA를 통해 구성한 FT에 대해 Cut-set을 수행하여 시스템이 가지는 고장이 단일점 결함인지, 혹은 다중점 결함인지 판단하여 FMEDA를 수행하여 HAM을 평가하고, 이를 통해 식별 및 적용한 각 고장에 알맞은 안전 메커니즘(Safety mechanism)을 다시 FTA에 적용하여 수정된 FT로부터 PMHF를 평가하였다.

정성적 측면의 안전 분석에서는, 상세 제어 구조에서 시스템 성능 저하 전략 및 고장 알람 전략을 하나의 상태 다이어그램에 작성하였던 Fig. 8을 기준으로 하였다. 시스템이 가지는 안전 기능과 그에 따라 변경되는 상태들을 기준으로 3.4절에서 정성적 안전 분석을 위해 정의하였던 5가지 안전 제약사항을 적용하여, 고장이 발생하였을 때 실제 Fig. 8에 정의된 상태 천이 순서에 따라 안전 상태가 달성되었는지 확인하였다.

특히 EPS ECU를 대상으로 수행하는 시스템 및 소프트웨어 수준의 테스팅 결과와 교차 검증을 하여, 실제로

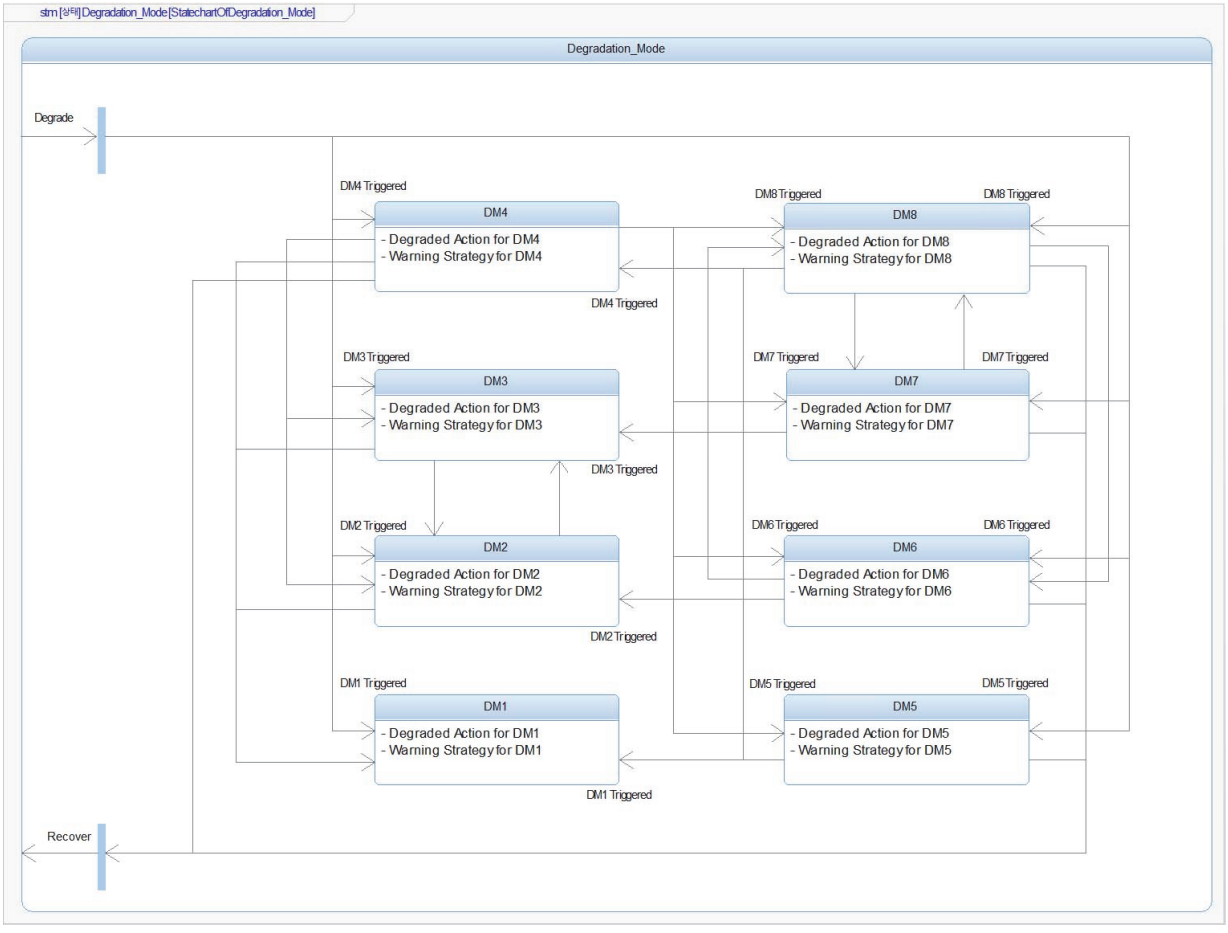

Fig. 8 Example about degradation concept of EPS ECU at HL Mando for safety behavior

도 관련된 사항이 시스템 V&V(Verification & Validation) 단계에서 검토되었음을 확인하였다. 이를 통해 최종적으로 고객의 요구사항에 부합하도록 안전성까지 검증된 소프트웨어 정의 조향 제어기의 개발을 완료하였으며, 또한 본 연구에서 제안하는 자동차 안전공학을 위한 MBSE 체계가 실제 SDV를 구성하는 제어기 시스템의 안전성을 실질적으로 보증함을 확인하였다.

5. 결 론

본 논문에서는 소프트웨어 정의 조향 제어기를 위한 자동차 안전공학 전략을 제안하였다. 특히 STPA를 시스템이 가지고 있는 제어 구조를 기능, 행동, 그리고 구조까지 세 가지의 관점에서 모델링하고, 각 관점에 적합한 분석 기법을 적용하였다. 이를 통해 설계와 분석, 그리고 검증까지 이어질 수 있는 SDV를 위한 MBSE 체계를 구성하고자 하였다.

본 연구를 통해 달성하고자 하는 목표는 세 가지이다. 첫째, 소프트웨어와 하드웨어의 수많은 상호작용으로 기능이 수행되는 SDV의 특성으로 인해 분석 과정에서 미처 파악하지 못하고 누락될 수 있는 사고의 위험원을 방지하고자 하였다. 둘째, 항공 분야의 AC 20-115D[22]가 DO-178C[23]를 준수해 항공 시스템의 안전성을 체계적으로 확보한 것과 같이, ISO 26262 및 ISO 21448을 준수하여 SDV의 안전성 뿐만 아니라 신뢰성까지 체계적으로 확보하기 위한 방안을 수립하고자 하였다. 셋째, HL만도에서 실제 개발 중인 EPS에 적용함으로써 SDV를 구성하는 소프트웨어 정의 조향 제어기에 대한 안전 활동 사례를 확보하고자 하였다.

향후 연구에서는 제안한 MBSE 체계에 따라 제어 구조를 설계하기 위해 파악했던 시스템의 행동을 대상으로 시제 논리(Temporal Logic)를 적용하여, Jeong 등[24]이 제안하였던 바와 같이 인적 오류(Human error)로 잘못된 설계된 모델에 의해 위험원 분석이 불완전하게 수행될 수 있는 경우를 회피하고, Kwon[25]과 Jeong 등[26]이 제안하였던 바와 같이 동적 모델로부터 안전 무결 요구사항을 정량적 정형 검증하여 보다 효율적인 안전 분석을 수행하고자 한다. 그리고 한 발 더 나아가 조수희 등[27], Chang 등[27]과 Kwon 등[29]이 제안하였던 다양한 연구에서 처럼 강화 학습 기술을 위험원 분석 또는 안전 분석에 적용하여 인간이 예상할 수 없는 부분까지 확장하여 안전성을 보증하기 위한 체계를 구성하거나, 최해서 등[30], 한승재 등[31]의 여러 연구와 같이 시뮬레이션 기술을 접목하여 공학적 체계를 넘어 실제 시스템 개발을 위한 V&V 영역까지 MBSE 체계를 확장해보고자 한다.

References

1) ISO, “ISO 26262:2018, Road Vehicles – Functional Safety,” International Organization for Standardization, 2018.
2) ISO, “ISO 21448:2022, Road Vehicles – Safety of the Intended Functionality,” International Organization for Standardization, 2022.
3) IEC, “IEC 61508:2010, Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems,” International Electrotechnical Commission, Geneva, 2010.
4) H. Pentti and H. Atte, Failure Mode and Effects Analysis of Software-Based Automation Systems, Radiation and Nuclear Safety Authority (STUK), Helsinki, 2002.
5) C. A. Ericson II, Hazard Analysis Techniques for System Safety, Wiley-Interscience, Hoboken, 2005.
6) P. Koopman, “A Case Study of Toyota Unintended Acceleration and Software Safety,” Proceedings of the High Integrity Software Conference, pp.1–55, 2015.
7) N. G. Levenson, Engineering a Safer World: Systems Thinking Applied to Safety, MIT Press, Cambridge, 2005.
8) R. S. Martínez, System Theoretic Process Analysis of Electric Power Steering for Automotive Applications, M. S. Thesis, Massachusetts Institute of Technology, Cambridge, 2015.
9) S. R. Do, Applying STPA to Establish a Safety Analysis System Based on ISO 26262, Ph.D. Dissertation, Sangmyung University, Seoul, 2016.
10) S. R. Do, “A Study on the Performing of Safety Analysis of ISO 26262 Development Phase Applying STPA Based on STAMP,” Transactions of KSAE, Vol.27, No.12, pp.965–975, 2019.
11) A. Abdulkhaleq, A System-Theoretic Safety Engineering Approach for Software-Intensive Systems, Ph.D. Dissertation, Stuttgart University, Stuttgart, 2017.
12) A. Abdulkhaleq, S. Wagner, D. Lammering, H. Boehmert and P. Blueher, “Using STPA in Compliance with ISO 26262 for Developing a Safe Architecture for Fully Automated Vehicles,” Lecture Notes in Informatics: Automotive Safety and Security 2017, pp.11–24, 2017.
13) D. H. Jeong, S. H. Park, G. H. Kwon and J. K. Kim, “System Theoretic Analysis and Reliability-Based Analysis for Preventing Accidents of Software-Intensive Systems,” Proceedings of the KIIT Conference, pp.154–159, 2021.

14) N. G. Levenson and J. P. Thomas, STPA Handbook, MIT Press, Cambridge, 2018.
15) SAE, “SAE J3187:2023, System Theoretic Process Analysis (STPA) Recommended Practices for Evaluations of Safety-Critical Systems in Any Industry,” Society of Automotive Engineers, Warrendale, 2023.
16) A. K. Goel, S. Rugaber and S. Vattam, Structure, Behavior, and Function of Complex Systems: The Structure, Behavior, and Function Modeling Language, Cambridge University Press, Cambridge, 2009.
17) VDA QMC, “Automotive SPICE Process Assessment/Reference Model 4.0,” Verband der Automobilindustrie, Berlin, 2023.
18) SAE, “SAE J2980:2017, Considerations for ISO 26262 ASIL Hazard Classification,” Society of Automotive Engineers, Warrendale, 2017.
19) IEC, “IEC 61882:2016, Hazard and Operability Studies (HAZOP Studies) – Application Guide,” International Electrotechnical Commission, Geneva, 2016.
20) SAE, “SAE J3016:2021, Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles,” Society of Automotive Engineers, Warrendale, 2021.
21) D. H. Jeong, Assuring E/E System Safety through Model-Based Analysis, https://www.autoelectronics.co.kr/article/articleView.asp?idx=4160, 2021.
22) FAA, “AC 20-115D: Airborne Software Assurance,” Federal Aviation Administration, Washington, 2017.
23) RTCA, “DO-178C: Software Considerations in Airborne Systems and Equipment Certification,” Radio Technical Commission for Aeronautics, Washington, 2011.
24) D. H. Jeong, J. W. Park, M. H. Kim and G. H. Kwon, “Synthesizing Control Algorithm in GR(1) Specification for System Theoretic Hazard Analysis,” Proceedings of the APIC-IST 2022, Seoul, pp.269–271, 2022.
25) G. H. Kwon, “Quantitative Verification of Safety Integrity Level in IEC 61508,” The Journal of KIIT, Vol.16, No.9, pp.43–50, 2018.
26) D. H. Jeong, R. G. Kwon and G. H. Kwon, “Probabilistic SIL Verification of a Synthesized Fault-Tolerant Model for Reliable Safety Assessment,” IEEE Access, Vol.13, pp.33148–33156, 2025.
27) S. H. Jo, R. G. Kwon and G. H. Kwon, “Safety Evaluation for Reinforcement Learning Model: Case Study of Autonomous Driving,” The Journal of KIIT, Vol.21, No.8, pp.165–174, 2023.
28) J. Y. Chang, R. G. Kwon and G. H. Kwon, “STPA-RL: Integrating Reinforcement Learning into STPA for Loss Scenario Exploration,” Applied Sciences, Vol.14, No.7, Paper No.2916, 2024.
29) R. G. Kwon, G. H. Kwon, S. H. Park, J. Y. Chang and S. H. Jo, “Applying Quantitative Model Checking to Analyze Safety in Reinforcement Learning,” IEEE Access, Vol.12, pp.18957–18971, 2024.
30) H. S. Choi, S. J. Han, J. W. Jeon, S. M. Ahn and J. W. Yoo, “Simulation-Based SOTIF Hazard Analysis and Risk Assessment Methodology for Autonomous Driving System,” Transactions of KSAE, Vol.32, No.4, pp.331–347, 2024.
31) S. J. Han, J. G. Byeon, S. M. Ahn and J. W. Yoo, “Derivation of STPA Safety Analysis and V&V Methodology for Ensuring SOTIF in Autonomous Driving System,” Transactions of KSAE, Vol.33, No.1, pp.61–80, 2025.
32) D. H. Jeong, “Model-Based Safety Engineering Strategy Based on System Theoretic for Software-Defined Vehicles,” KSAE Annual Conference Proceedings, pp.2163–2172, 2024.