Journal of KIIT. Vol. 16, No. 11, pp. 133-144, Nov. 30, 2018. pISSN 1598-8619, eISSN 2093-7571 133
http://dx.doi.org/10.14801/jkiit.2018.16.11.133

철도 시스템의 안전성 향상을 위한 하이브리드 위험원 분석

정대희*, 권기현**

Hybrid Hazard Analysis for Improving Safety of Railway System

Daehui Jeong*, Gihwon Kwon**

이 논문은 과학기술정보통신부 및 정보통신기술진흥센터의 대학 ICT연구센터육성지원사업의 연구결과로 수행되었음 (IITP-2015-0-00445)

요 약

철도 시스템의 안전성 표준인 IEC 62278은 위험원 분석을 통해서 철도 시스템이 가질 수 있는 위험원을 예방하거나 또는 제어하도록 요구한다. 만약 위험원 분석이 충분하지 않으면 사고가 발생할 가능성이 높기 때문에, 위험원 분석을 보다 철저히 수행할 필요가 있다. 본 논문에서는 기존의 신뢰성 기반 방법과 시스템 이론적 방법을 상호 결합한 하이브리드 위험원 분석을 제안한다. 제안하는 방법은 기존 위험원 방법을 상호 보완하는 것으로서, 시스템 구성 요소의 고장으로 인한 위험원과 구성 요소들 간의 상호작용으로 인해 발생되는 제어 위험원을 함께 분석한다. 열차간의 속도를 자동 제어하는 다중 적응형 순항 제어 장치의 안전 보호 시스템 개발에 적용한 결과, 기존 방법들 보다 안전 요구사항을 더 많이 추출하여, 위험원으로부터 시스템을 보호함을 확인하였다.

Abstract

IEC 62278, the Railway System Safety Standard, requires for hazard analysis to prevent or control the hazard that the railway system may have. If hazard analysis is not performed sufficiently, there is a high probability that accidents will occur. For this reason, hazard analysis methods are actively studied. In this paper, we propose the hybrid hazard analysis method to combine two representative hazard analysis methods: reliability-based and system-theoretic. As the proposed method is complementary to existing ones, it covers both the hazard caused by failure of components and the hazard occurred from the unintended control between components. It applies to the development of a safety protection mechanism for multiple cruise control system that automatically control the speed of trains to avoid the collision among trains. As a result, we drive more safety requirements than the existing analysis methods and it turns out that the safety requirements protect the trains with respect to the identified hazards.

Keywords
railway system, hazard analysis, reliability-based hazard analysis, system-theoretic hazard analysis

* 경기대학교 컴퓨터공학부 석사과정
- ORCID: https://orcid.org/0000-0001-6157-1294
** 경기대학교 컴퓨터공학부 교수(교신저자)
- ORCID: https://orcid.org/0000-0002-8221-4939

· Received: Sep. 07, 2018, Revised: Nov. 19, 2018, Accepted: Nov. 22, 2018
· Corresponding Author: Gihwon Kwon
Dept. of Computer Engineering, Kyonggi University, 154-42, Gwangyosan-ro, Suwon-si, Kyonggi-do, Korea.
Tel.: +82-31-249-9666, Email: khkwon@kgu.ac.kr

Ⅰ. 서 론

IEC 62278은 철도 분야의 시스템 안전성을 돕는 국제 표준으로서, 안전한 철도 임베디드 시스템의 개발을 위해서 개념단계에서 폐기단계까지 14단계에 걸쳐서 다양한 안전성 활동을 정의하고 있다[1]. 특히, 위험원 분석(Hazard Analysis)은 시스템의 위험원을 식별하고 평가하여, 안전 대책을 파악하는 안전성 활동이다. 만약 위험원 분석을 충분히 수행하지 않으면, 사고가 발생할 가능성이 높기 때문에, 위험원 분석은 중요한 안전성 활동이다.

사용되고 있는 위험원 분석 방법들은 신뢰성 기반(reliability-based)과 시스템 이론적(system-theoretic)으로 구분된다. 신뢰성 기반에는 ETA(Event Tree Analysis), FMEA(Failure Modes and Effects Analysis), FTA(Fault Tree Analysis), HAZOP(HAZard and OPerability study), PHA(Preliminary Hazard Analysis) 등이 있다[2]. 이들은, 시스템을 구성하는 각 요소의 고장(Failure)을 중심으로 위험원의 발생 원인과 위험원으로 인한 피해 영향간의 인과 관계를 분석하여, 위험원의 발생 회피 또는 피해 영향을 저감하는 안전 대책을 찾는다. 분석 초점이 구성 요소의 고장에 있기 때문에, 구성 요소들 간의 상호작용으로 인해 발생되는 제어 위험원은 다루기 어렵다. 한편, 시스템 이론적 방법에는 STPA(System Theoretic Process Analysis)가 있다[3]. 이것은 시스템 구성 요소간의 상호작용을 중심으로, 잘못된 제어(Control)로 인해서 발생할 수 있는 위험원을 분석한다. 분석의 초점이 구성 요소간의 상호작용이기 때문에, 구성 요소의 단일 고장으로 인한 위험원은 다루기 어렵다.

구성 요소의 고장으로 인한 위험원 및 상호작용으로 인한 제어 위험원도 다루고자 본 논문에서는 두 가지 방법을 결합한 하이브리드 위험원 분석을 다음과 같이 제안한다. 첫째, 시스템 명세로부터 시스템 기능을 파악한 후에, HAZOP을 이용하여 시스템이 가질 수 있는 고장 유형, 즉 위험원을 식별한다. 둘째, FTA를 수행하여 위험원의 발생 원인을 파악한다. 셋째, STPA를 수행하여 안전하지 않은 제어를 분석한다. 넷째, FTA와 ETA를 수행하여 위험원의 예방 대책과 저감 대책을 식별한다.

제안 방법의 유효성을 확인하기 위하여, 디오라마 모형 철도를 이용해 열차 세 대의 속도를 자동으로 제어하는 다중 적응형 순항 제어 장치를 위한 안전 보호 시스템을 개발하였다. 제안한 방법으로 시스템의 위험원을 분석하여 안전 요구사항을 추출하였으며, 이를 바탕으로 안전 보호 시스템을 구현하여 위험원으로부터 열차를 보호함을 확인하였다. 또한, 신뢰성 기반 위험원 분석과 시스템 이론적 위험원 분석 방법을 각각 개별적으로 적용한 것보다, 제안하는 하이브리드 위험원 분석이 더 많은 안전 요구 사항을 식별하여, 위험원 분석을 더 철저하게 수행하였다.

본 논문의 구성은 다음과 같다. 2장에서 신뢰성 기반 및 시스템 이론적 위험원 분석 방법을 각각 살펴본다. 3장에서는 제안하는 위험원 분석 방법의 개념 및 절차를 소개한다. 4장에서는 제안한 방법을 사례 시스템 개발에 적용한 결과를 기술하고, 5장에서는 결론 및 향후 연구를 소개한다.

Ⅱ. 배경 지식

2.1 신뢰성 기반의 위험원 분석

철도 분야 기능 안전 표준인 IEC 62278은 신뢰성 기반의 위험원 분석 방법을 사용할 것을 권고한다[1]. 이 방법은 시스템 구성 요소의 고장을 중심으로 위험원을 분석한다. 위험원 분석을 통해서 어떤 원인으로 인하여 고장이 발생하였고, 이로 인하여 시스템에 어떠한 피해를 미치는지 그 영향을 파악하여, 위험원을 회피하거나 또는 피해 영향을 저감하는 안전 대책을 파악한다.

2017년 제작된 “철도 분야 신뢰·안전 가이드”는 신뢰성 기반의 다양한 위험원 분석 방법을 소개하고 있어서, 프로젝트의 규모와 조직에 상황에 맞게 이들을 조합하여 위험원 분석을 수행할 수 있다[4]. 예를 들어, 그림 1 좌측처럼 네 단계로 위험원을 분석할 수 있다. 첫째, 고장 유형 및 기능 고장 식별에서는 시스템 명세로부터 시스템이 갖추어야 할 기능을 파악하고, 기능으로부터 발생될 수 있는 고장 유형과 기능 고장을 식별한다.

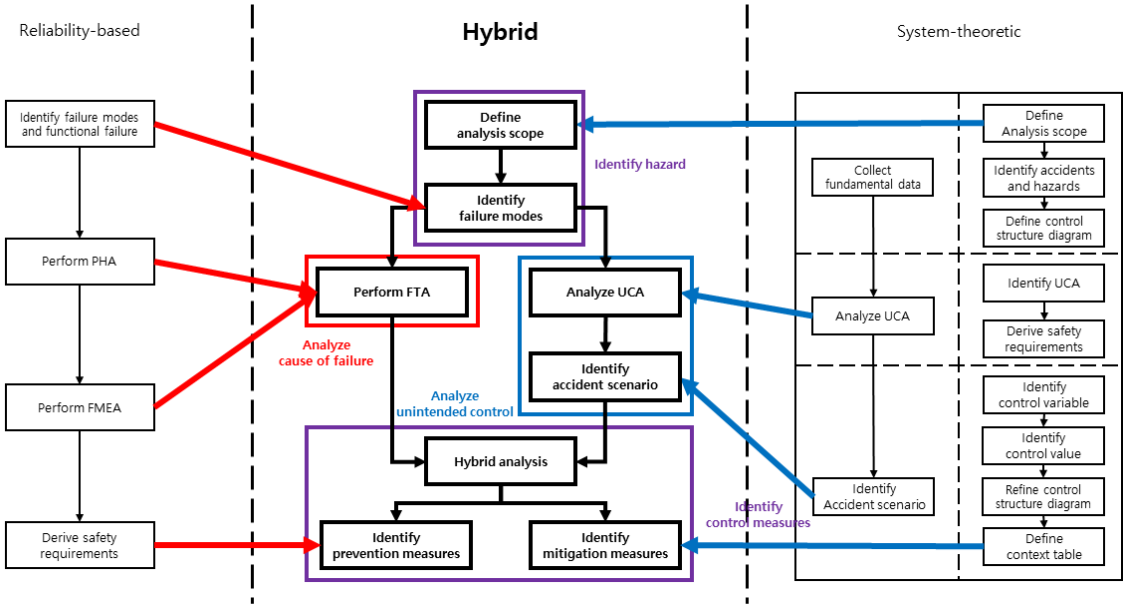

그림 1. 제안하는 위험원 분석 방법
Fig. 1. Proposed hazard analysis

둘째, PHA를 통해서 식별된 고장 유형 및 기능 고장을 발생시킨 원인과 발생되는 영향을 분석한다. 셋째, FMEA 단계에서는 PHA를 통해 분석된 원인 및 영향마다 어느 정도의 빈도로 발생되고 얼마만큼 심각한 피해를 입히는지를 파악한다. 넷째, 이러한 분석 결과로부터 안전 요구사항을 도출한다.

2.2 시스템 이론적 위험원 분석

STPA는 시스템 이론에 기반을 둔 대표적 위험원 분석 방법이며, 수행 절차는 그림 1 우측과 같이 기초 자료 수집, UCA(Unsafe Control Action) 분석, 사고 시나리오 식별의 순서로 진행된다[5]. 첫째, 기초 자료 수집에서는 시스템 수준의 설계 문서를 바탕으로 분석 범위를 정하고, 시스템에서 발생 가능한 사고와 위험원을 식별하고, 제어 구조 다이어그램을 작성한 후에 이를 바탕으로 제어 명령을 식별한다. 둘째, UCA 분석에서는 제어 명령 별로 네 개의 키워드를 적용하여 UCA를 식별하고 초기 수준의 안전 요구사항을 식별한다. 여기서 키워드는 'Wrongly provided', 'Not provided', 'Provided wrong timing or order', 'Stopped too soon or applied too long'의 네 가지이다. 셋째, 사고 시나리오 식별에서는 제어 변수와 제어 값을 포함하는 제어 모형(Process Model)을 식별하여 이를 포함한 제어 구조 다이어그램을 재작성하고, 상황표를 작성하여, 이를 바탕으로 초기 수준의 안전 요구사항을 재작성하거나 추가한다.

III. 하이브리드 위험원 분석

3.1 기존 분석의 문제점

철도 안전 시스템에서 전기전자제어(E/E/PE) 사용이 증가함에 따라서 시스템의 복잡도가 높아지고, 시스템 구성 요소 사이의 상호작용으로 인한 위험원이 늘고 있다. 이로 인해서, 위험원을 분석할 때에는 구성 요소의 고장으로 인한 위험원 및 구성 요소 사이의 상호작용으로 인한 위험원도 함께 고려해야 한다. 그러나 전 장에 살펴본 바와 같이, 기존 방법으로는 이들을 모두 분석하기 어렵다.

신뢰성 기반 분석은 구성 요소의 고장을 중심으로 위험원을 분석하기 때문에, 상호작용으로 인한 제어 위험원은 다루기 어렵다. 예를 들어 '열차 A'와 '열차 B'가 운행된다고 가정하자. 또한, 신뢰성

기반의 위험원 분석으로 “속도 제어 명령 전달에도 불구하고 두 열차 사이의 위치 정보 변경이 3번 이상 변화가 없을 경우 Fail-safe 한다”라는 안전 요구사항이 도출되었다고 가정하자. 만약 시스템 구성 요소의 고장이 발생하여 위치가 3회 이상 변화되지 않는다면, 요구사항에 따라서 ‘Fail-safe’를 발동해서 전체 시스템을 비상 정지한다. 그러나 두 열차 사이의 거리가 감소하였을 때, 앞 열차인 ‘열차 A’에게 ‘속도 감속’ 제어 명령을 전달할 경우, 두 열차 사이의 위치 정보는 계속 변화하는 상태이기 때문에 위의 안전 요구사항은 수행되지 않고, 마침내 ‘열차 A’와 ‘열차 B’의 추돌이 발생할 수 있다.

한편, 시스템 이론적 분석은 상호작용을 중심으로 위험원을 분석하는 만큼, 고장으로 인한 시스템 동작 과정상의 위험원을 다루는 데에는 한계가 있다. 예를 들어, 시스템 이론 기반 분석으로 도출된 안전 요구사항이 “열차 A와의 거리가 감소하면 열차 B는 감속해야만 한다”라고 가정하자. 이는 ‘열차 A’와 ‘열차 B’ 사이의 거리가 감소하였을 때, 후속 열차인 ‘열차 B’에게 ‘속도 증가’와 같은 의도하지 않은 제어 명령을 전달하는 경우를 방지한다. 하지만, 위치 감지기의 오동작이 발생하면, 실제로 두 열차간의 거리가 감소하였음에도 불구하고, 안전 요구사항이 수행되지 않아서, 마침내 ‘열차 A’와 ‘열차 B’의 추돌이 발생할 수 있다.

이처럼, 신뢰성 기반 분석과 시스템 이론적 분석은 고장으로 인한 위험원과 상호작용으로 인한 위험원 모두를 분석하기는 어렵다. 만약 상호보완적인 두 분석을 결합한다면 하나의 분석만으로는 발견할 수 없었던 위험원을 다른 분석으로 찾아낼 수 있을 것이다. 지금까지, 철도 시스템 위험원 분석에 신뢰성 기반 방법을 사용하거나 또는 시스템 이론적 방법을 사용한 연구들이 있었다[6][7]. 따라서 본 논문에서는 이 둘을 결합한 하이브리드 위험원 분석 방법을 연구한다.

3.2 위험원 분석 절차

하이브리드 위험원 분석은 그림 1 중앙과 같이, 네 개의 활동으로 구성된다. 첫째, 위험원 식별은 분석하고자 하는 시스템의 범위를 파악해 시스템 명세로부터 해당 범위내의 시스템 기능을 식별하고 제어 구조 다이어그램을 작성한다. 그리고 HAZOP을 이용해 기능으로부터 발생할 수 있는 고장 유형을 식별한다. 둘째, 고장 원인 분석은 FTA를 수행해 식별된 고장 유형이 어떠한 원인으로부터 발생되었는지를 분석한다. 셋째, 의도치 않은 제어 분석에서는 제어 구조 다이어그램으로부터 의도치 않은 제어를 분석하고 사고 시나리오를 식별해서, 고장 유형으로 인하여 어떠한 잘못된 제어 명령이 발생할 수 있는지를 분석한다. 마지막으로, 제어 대책 식별은 위험원으로부터 시스템을 보호하는 예방 대책과 저감 대책을 식별한다. 이들 활동들을 자세히 설명하면 다음과 같다.

3.2.1 위험원 식별

위험원 식별은 분석 범위 파악과 고장 유형 식별로 이루어진다. 분석 범위 파악에서는 분석하고자 하는 목표 시스템의 목적이 무엇인지를 파악하여 시스템 목적 달성을 위한 기능을 식별하고 이를 바탕으로 제어 구조 다이어그램을 작성한다. 여기서 제어 구조 다이어그램에 포함되는 요소들에는 제어기(Controller), 작동기(Actuator), 감지기(Sensor), 제어되는 프로세스(Controlled Process)가 있으며, 기능으로부터 시스템이 가지는 제어 구조를 파악하여 제어 명령을 추출한다. 고장 유형 식별에서는 HAZOP의 매개 변수와 안내어를 이용하여 기능으로부터 어떠한 고장 유형이 발생될 수 있는지를 파악한다[8]. 철도 분야를 다루기 때문에, 사용되는 매개 변수와 안내어는 HAZOP-KR(HAZOP for Korean Railway)을 따른다[9].

3.2.2 고장 원인 분석

고장 원인 분석에서는 FTA를 수행하여 위험원 식별에서 파악된 고장 유형을 발생시키는 원인을 분석한다. 고장 유형을 정상 사상(Top Event)으로 하여 결함 트리를 작성해, 고장 유형을 발생시키는 고장 원인을 파악한다. 단일 고장 원인 또는 복수 고장 원인으로 고장 유형이 발생될 수 있다.

3.2.3 의도치 않은 제어 분석

의도치 않은 제어는 STPA 활동에서 UCA 분석과 사고 시나리오 식별을 수행한다. 위험원 식별에서 작성하였던 제어 구조 다이어그램으로부터 시스템이 가질 수 있는 UCA를 식별하고 어떤 고장 유형으로 인해 UCA가 야기되는지를 파악한다. 이후에, 제어 변수와 제어 값으로 구성된 제어 모형을 포함하여 제어 구조 다이어그램을 재작성 한다. 그리고 상황표를 작성해서 사고 시나리오를 파악하고, 이로부터 UCA 분석 시에는 미처 파악하지 못하였던, 잘못된 제어 명령이 있는지를 확인한다.

3.2.4 제어 대책 식별

제어 대책 식별을 위해서는 하이브리드 분석이 수행된다. 고장 유형을 중심으로 고장 원인 분석과 의도치 않은 제어 분석에서 분석한 결과들을 정리하고 위험도를 추정해, 시스템의 위험원을 제어하기 위한 제어 대책을 식별한다. 여기서 위험도의 추정에는 발생 빈도와 심각도 두 가지 요소를 사용한다. 또한 위험도와 발생 빈도, 심각도의 추정 기준은 표준에 제시된 바를 따랐다[1].

제어 대책으로는 고장 유형의 발생 빈도를 줄이기 위한 예방 대책과 심각도를 줄이기 위한 저감 대책을 식별한다. 예방 대책은 고장 원인 분석에서 작성하였던 결함 트리를 이용한다. 결함 트리에서 파악한 고장 유형의 원인으로부터 발생 빈도를 추정하고 이를 낮추기 위한 대책을 찾아낸다. 여기서 찾아낸 방법은 이중화와 같이 고장 원인의 고장률을 낮춤으로써 전체 고장 유형의 고장률을 낮추고, 결과적으로 발생 빈도를 낮춘다. 저감 대책은 의도치 않은 제어 분석에서 식별된 사고 시나리오와 ETA를 이용한다. ETA를 수행해 사고 시나리오로 인해 발생되는 영향과 심각도를 분석하고, 이를 낮추기 위한 방법을 파악한다. 이 때 파악된 방법은 사고 시나리오가 ETA의 판단 기준에 부합하도록 만들어 결과적으로 심각도를 낮춘다.

Ⅳ. 적용 사례

4.1 목표 시스템

제안 방법의 유효성 확인을 위하여 사례 연구를 수행하였다. 사례 연구로는 디오라마 모형 철도를 이용하여 열차간의 속도를 제어하는 다중 적응형 순항 제어장치인 MACC(Multiple Adaptive Cruise Control)의 안전 기능을 개발하는 것이다. 그림 2와 같이 세 열차간의 거리를 감지하여 자동으로 안전거리를 유지한다. 만약 안전거리가 유지되지 않으면, 열차 추돌 같은 사고가 발생하기 때문에 철저한 위험원 분석이 요구된다. 이를 위하여 본 논문에서 제안하는 방법으로 위험원 분석을 수행하여 안전 요구사항을 찾아내고, 이를 바탕으로 시스템을 개발한다. 과연 찾아낸 요구사항이 정확한지 그리고 기존의 위험원 분석 방법에 비해서 얼마나 더 철저한지를 사례 연구를 통해서 비교 분석하고자 한다.

MACC시스템은 그림 2와 같이 ‘Position Sensors’, ‘Detectors’, ‘DCC’, ‘MACC Software’, ‘Decoders’들로 구성된다. ‘Position Sensors’는 열차가 감지되면 ‘Detectors’로 감지된 정보를 전달하고, ‘Detectors’는 ‘Position Sensors’가 열차를 감지하면 해당 지역 정보를 ‘DCC’로 전달한다. ‘DCC’는 ‘Detectors’에서 전달된 정보를 저장하거나 ‘MACC Software’에서 전달된 제어 명령에 따라서 ‘Decoders’로 열차 제어 정보를 전달한다. ‘MACC Software’는 ‘DCC’에서 전달되는 정보를 감시하여 ‘DCC’에 열차 제어명령을 전달하고, ‘Decoders’는 ‘DCC’에서 전달된 열차 제어 정보에 따라 열차를 제어한다. 표 1은 MACC 시스템의 기능을 나타낸다.

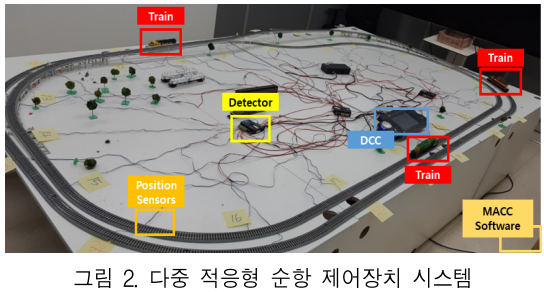

그림 2. 다중 적응형 순항 제어장치 시스템
Fig. 2. Multiple adaptive cruise control system

표 1. 시스템 기능
Table 1. System function

System Level	Components Level	Detailed Functions
Calculate distance	'Position Sensors' control	Detect train
	'Detectors' control	Detect train position
Maintain safety distance	'DCC' control	Save train position data
		Send train position data
	'MACC Software' control	Check safety distance
		Control train speed
Control speed	'Decoders' control	Perform control order

4.2 신뢰성 기반 위험원 분석 결과

신뢰성 기반 위험원 분석은 시스템 기능에서 고장 유형 및 기능 고장을 식별하는 것으로부터 시작한다. 그래서, MACC 시스템이 기능을 수행함에 있어서 고장을 유발시킬 수 있는 요소를 파악한다. 예로서 '열차속도 제어'의 기능 수행에는 'Decoders'가 사용된다. 속도 제어 명령을 수행해 열차의 속도를 변환하는 'Decoders'는 '제어 명령 수행 실패'의 기능 고장이 발생될 수 있고, 이는 '속도 제어 실패'라는 고장 유형을 야기한다.

그 후에 PHA와 FMEA를 수행하여 고장 유형 및 기능 고장을 일으키는 원인과 영향을 바탕으로, 구성 요소가 가지는 고장 유형으로 인해 발생되는 고장 영향을 분석하고 위험도를 추정한다. 또한, 이를 제어하기 위한 제어 조치를 표 2와 같이 수립한다.

4.3 시스템 이론적 위험원 분석 결과

STPA는 기초 자료를 수집하는 것으로부터 시작한다. 분석 대상인 MACC 시스템은 세 열차를 제어한다. 만약 MACC 기능이 정상적으로 작동하지 않으면 열차 간 추돌 사고가 발생한다. 이러한 사고를 일으킬 수 있는 위험원은 표 3과 같다.

기초 자료 수집의 마지막은 제어 구조 다이어그램을 작성하고 제어 명령을 식별하는 것이다. MACC 시스템의 구성 정보와 기능을 바탕으로 작성된 제어 구조 다이어그램은 그림 3과 같으며, 제어 명령은 '가속', '감속', '유지', '완전 정지'가 있다.

표 2. FMEA 결과
Table 2. Results of FMEA

Failure modes	Functional failure	Failure effects	Occurrence	Severity	Failure cause	Control measures
Calculate distance fail	'Position Sensors' control fail	Detect train fail	Probable	Critical	1. 'Position Sensors' failure	1. Component redundancy 2. Component fault detection design 3. Emergency stop via fail-safe
	'Detectors' control fail	Detect train position fail	Remote	Critical	1. 'Detectors' failure	
Maintain safety distance fail	'DCC' control fail	Save train position data fail	Improbable	Critical	1. 'DCC' failure 2. Communicate interface error	
		Send train position data fail	Improbable	Critical		
	'MACC Software' control fail	Check safety distance fail	Improbable	Critical	1. Wrong algorithm 2. CPU failure	
		Control train speed fail	Improbable	Critical		
Control speed fail	'Decoders' control fail	Perform control order fail	Remote	Critical	1. 'Decoders' failure 2. Communicate interface error	

표 3. 위험원 목록
Table 3. List of Hazards

Hazards
The control software does not receive the position data of 'Train A' or 'Train B', or 'Train C'.
The control software does not accelerate 'Train A' even though the distance between 'Train A' and 'Train B' has decreased.
The control software does not decelerate 'Train B' even though the distance between 'Train A' and 'Train B' has decreased.
The control software does not fully stop 'Train B' even though the distance between 'Train A' and 'Train B' is less than the safe distance.
The control software does not accelerate 'Train B' even though the distance between 'Train B' and 'Train C' has decreased.
The control software does not decelerate 'Train C' even though the distance between 'Train B' and 'Train C' has decreased.
The control software does not fully stop 'Train C' even though the distance between 'Train B' and 'Train C' is less than the safe distance.

제어 명령이 식별되면 UCA를 식별한다. 표 4는 전체 UCA 중 '열차 A'와 관련된 UCA를 식별한 것으로, 7개 UCA가 위험원과 관련 있다. 예를 들어 UCA 중 하나인 "열차 B와의 거리가 감소하였음에도 불구하고 가속하지 않는다"는 "제어 소프트웨어가 열차 A 또는 열차 B 또는 열차 C의 위치 정보를 전달받지 못한다" 혹은 "제어 소프트웨어가 열차 A와 열차 B 사이의 거리가 감소하였음에도 불구하고 열차 A를 가속하지 못한다"의 두 위험원으로 인해 발생될 수 있다.

STPA 마지막 활동은 사고 시나리오의 식별이다. 제어 모형을 포함해 제어 구조 다이어그램을 재작성 하고, 상황표를 작성해 안전 요구사항을 재작성 및 추가한다. MACC 시스템의 제어 모형은, 제어 명령에 영향을 미치는 '열차 A'와 '열차 B' 사이의 거리와 '열차 B'와 '열차 C' 사이의 거리가 제어 변수가 되고, 각각 '증가', '감소', '유지', '안전거리 미만'의 제어 값을 가진다. 이후 상황표는 제어 명령이 전달되지 못한 경우와 전달된 경우로 나누어, 제어 명령 별로 제어 변수와 제어 값을 조합해 작성한다. 표 5는 UCA와 상황표로부터 도출된 전체 안전 요구사항 중에서 '열차 A'와 관련된 것이다.

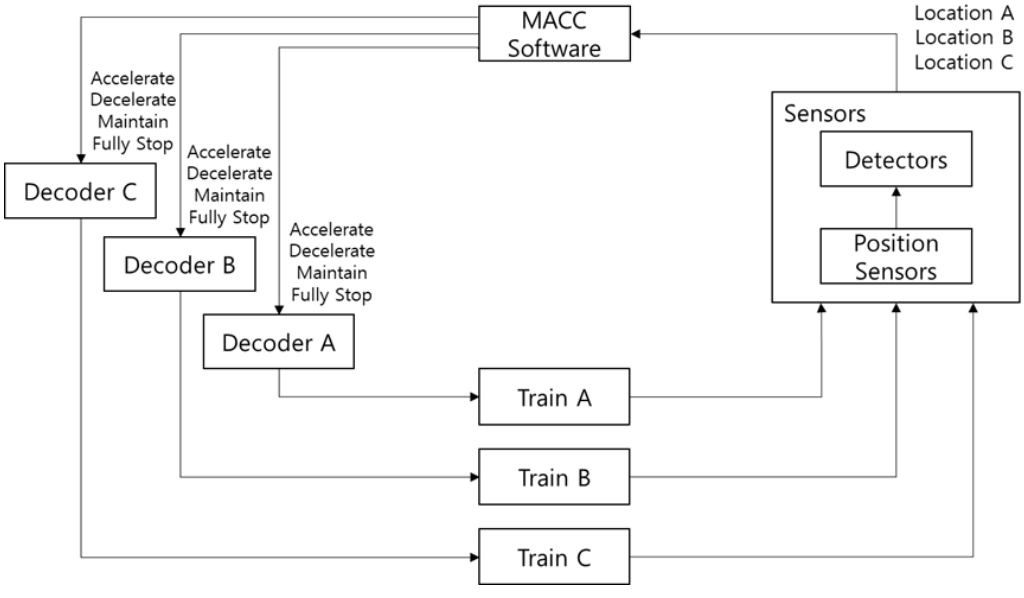

그림 3. MACC 제어 구조 다이어그램
Fig. 3. Control structure diagram for MACC

표 4. '열차 A'와 관련된 UCA
Table 4. UCA related to 'Train A'

UCA	Not provided	Wrongly provided	Provided wrong timing or order	Stopped too soon or applied too long
accelerate	Acceleration is not performed, even though the distance from 'Train B' has decreased.	Not hazardous	Acceleration is performed late, even though the distance from 'Train B' has decreased.	Acceleration is stopped, even though the distance from 'Train B' still have decreased.
decelerate	Not hazardous	Deceleration is performed, even though the distance from 'Train B' has decreased.	Not hazardous	Not hazardous
maintain	Maintainment is not performed, even though the distance from 'Train B' has keep.	Maintainment is performed, even though the distance from 'Train B' has not keep.	Not hazardous	Not hazardous
fully stop	Not hazardous	Fully stopping is performed, even though the distance from 'Train B' is safety distance.	Not hazardous	Not hazardous

표 5. '열차 A'와 관련된 안전 요구사항
Table 5. Safety requirements related to 'Train A'

Safety Requirements
If the distance from 'Train B' decreases, 'Train A' must accelerate.
If the distance from 'Train B' decreases, 'Train A' must not be accelerated late.
If the distance to 'Train B' is still decreasing, 'Train A' must not stop acceleration.
If the distance from 'Train B' decreases, 'Train A' must not decelerate.
If the distance to 'Train B' is still keeping, 'Train A' must be maintained.
If the distance to 'Train B' is not keeping, 'Train A' must not be maintained.
If the distance to 'Train B' is the safety distance, 'Train A' must not be fully stopped.
If the distance to 'Train B' is less than the safety distance, 'Train A' must accelerate.
If the distance from 'Train B' increases, 'Train A' may have to decelerate.*
If the distance from 'Train B' increases, 'Train A' may not have to accelerate.*
If the distance from 'Train B' increases, 'Train A' may not have to accelerate early.*
If the distance from 'Train B' increases, 'Train A' may not have to accelerate late.*

4.4 하이브리드 위험원 분석 결과

하이브리드 위험원 분석은 분석 범위를 파악하는 것으로부터 시작된다. 분석하고자 하는 MACC 시스템은 세 대의 열차가 운행될 때 각 열차 사이의 거리를 감지해 항상 안전거리를 유지하도록 자동으로 속도를 조절하는 시스템이다. 이를 유즈케이스 다이어그램으로 분석하면 사용자로는 'Trains'가 있고, 'To detect distance', 'To control train speed', 'To maintain safety distance'의 세 유즈케이스를 가진다. 또한 제어 구조 다이어그램은 그림 3과 같다.

분석 범위가 파악되면 고장 유형을 식별한다. 고장 유형은 파악된 유즈케이스를 기준으로 한다. 표 6은 식별된 고장 유형으로, 표와 같은 고장 유형이 존재할 때 시스템이 해당 유즈케이스를 요구할 경우, 시스템의 위험원이 됨을 의미한다.

식별된 고장 유형은 FTA를 수행하여 고장 원인을 분석하였다. 그림 4는 표 6의 고장 유형 중 "Distance is not detected"에 대한 FTA 수행 결과로, 'Position Sensors 고장'과 'Detectors 고장'을 고장 원인으로 가진다. 또한 이 고장 유형은 두 고장 원인 중 하나만 성립되어도 발생된다.

표 6. HAZOP-KR을 이용한 고장 유형
Table 6. Failure modes using HAZOP-KR

Use-case	Parameters	Guidewords	Failure modes
To detect distance	Interface	No	Distance detect interface is not worked
	Action	No	Distance is not detected
		Early	Distance is detected too early
		Late	Distance is detected too late
	Outside	Part of	Something is detected instead of trains
	Data	No	Distance data is not created or sent
		Early	Distance data is created or sent too early
		Late	Distance data is created or sent too late
To maintain safety distance	Action	No	Safety distance is not maintained
		Early	Safety distance is maintained too early
		Late	Safety distance is maintained too late
	Data	No	Maintenance order data is not created or sent
		Early	Maintenance order data is created or sent too early
		Late	Maintenance order data is created or sent too late
To control trains speed	Interface	No	Speed control interface is not worked
	Action	No	Speed is not controlled
		Early	Speed is controlled too early
		Late	Speed is controlled too late
	Data	No	Speed data is not received
		Early	Speed data is received too early
		Late	Speed data is received too late
-	Outside	Other than	Unexpected case is occurred

다음으로는 고장 유형으로 인한 의도치 않은 제어를 분석하기 위해 UCA를 식별 및 분석하고 사고 시나리오를 식별한다. UCA는 시스템 이론 기반 분석에서와 마찬가지로 표 4와 같다.

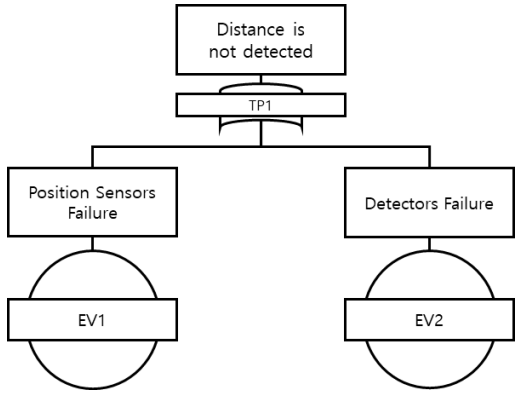

그림 4. "Distance is not detected"의 FTA 결과
Fig. 4. FTA result from "Distance is not detected"

예를 들어 "열차 A는 열차 B와의 거리가 감소하였음에도 불구하고 가속하지 않는다"는 "Distance detect interface is not worked"를 비롯한 15개의 고장 유형으로 인해 발생될 수 있다. 그 이후에 진행되는 사고 시나리오 식별은 시스템 이론적 방법과 같다.

마지막으로 하이브리드 분석을 통해 제어 대책을 식별한다. 하이브리드 분석을 위해서는 사고 시나리오 중 위험한 경우에 대하여 고장 유형과 고장 원인을 정리하는 작업이 필요하다. 표 7과 표 8은 표 4에서 식별한 위험한 사고 시나리오 중 하나인 "열차 A와 열차 B 사이의 거리가 감소하였음에도 제어 명령인 열차 A에 가속 명령이 전달되지 않는다"에 대한 하이브리드 분석표이다. 이 사고 시나리오는 15개의 고장 유형과 관련이 있고, 고장 유형은 각자의 고장 원인을 가진다. 예를 들어 '잘못된 Position Sensors 연결'로 인한 "Distance detect interface is not worked"가 발생하면 '열차 A'와 '열차 B' 사이의 거리가 감소하여도 해당 정보를 전달받지 못해 제어 명령 '열차 A 가속'이 전달되지 않게 된다.

하이브리드 분석표가 작성되면 예방 대책과 저감 대책을 식별한다. 예방 대책은 FTA, 즉 결함 트리를 이용한다. 예를 들어 표 7의 고장 유형 중의 하나인 "Distance is not detected"는 두 개의 원인인 'Position Sensors 고장'과 'Detectors 고장'을 가진다. 이 고장 유형은 두 고장 원인 중 하나만 성립되어도 발생하기 때문에 'Position Sensors'와 'Detectors'의 고장률을 동시에 감소시켰을 때 가장 발생 빈도를 낮출 수 있다. 그러기 위해 'Position Sensors'와 'Detectors'에 이

중화 혹은 결함 감지 설계가 요구된다.

저감 대책의 경우는 사고 시나리오와 ETA를 이용한다. 예를 들어 표 7의 사고 시나리오 “열차 A와 열차 B 사이의 거리가 감소하였음에도 제어 명령인 열차 A의 가속이 전달되지 않는다”를 방지하기 위하여 이를 부정적으로 표현한 “열차 A와 열차 B 사이의 거리가 감소하였을 때 제어 명령 열차 A 가속은 반드시 전달되어야 한다”라는 안전 요구사항을 도출하고, 이를 달성하기 위해 필요한 요소를 판단 기준으로 하여 그림 5와 같이 ETA를 수행한다. 분석 결과를 살펴보면 현재 시스템은 ‘Fail-safe’는 구현되어 있지만, 비상 상황을 대비한 모니터링이 이루어지지 않고 있기 때문에 심각도는 ‘Catastrophic’이다. 그렇기 때문에 모니터링 시스템을 구축해서 위험한 시나리오가 발생하는 경우를 감지하여 이를 제어한다면 심각도는 ‘Marginal’로 감소한다.

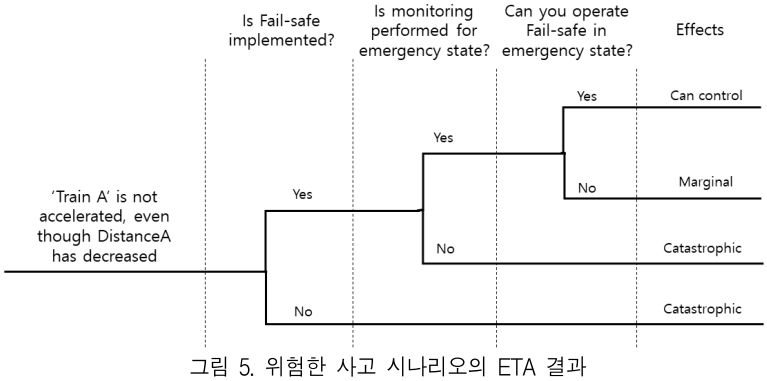

그림 5. 위험한 사고 시나리오의 ETA 결과

Fig. 5. ETA result from one dangerous accident scenario

표 7. 위험한 사고 시나리오의 하이브리드 분석표

Table 7. Hybrid analysis table from one dangerous accident scenario

Control actoin	Type	DistanceA	Failure modes	Cause of failure
‘Train A’ accelerate	control action is not delivered	Decrease	Distance detect interface is not worked	Incorrect Position Sensors connection
				Incorrect Detectors connection
			Distance is not detected	Positions Sensors failure
				Detectors failure
			Distance is detected too late	Positions Sensors failure
				Detectors failure
			Distance data is not created or sent	Incorrect Position Sensors connection
				Incorrect Detectors connection
				Incorrect DCC connection
				Positions Sensors failure
				Detectors failure
				DCC failure
			Distance data is created or sent too late	Incorrect Position Sensors connection
				Incorrect Detectors connection
				Incorrect DCC connection
				Positions Sensors failure
				Detectors failure
				DCC failure
			Safety distance is not maintained	CPU failure
				Incorrect MACC Software algorithm
				Incorrect safety distance definition
			Safety distance is maintained too late	Software internal computer failure
				Incorrect MACC Software algorithm
				Incorrect safety distance definition

표 8. 위험한 사고 시나리오의 하이브리드 분석표 (계속)
Table 8. Hybrid analysis table from one dangerous accident scenario (Continue)

Control actoin	Type	DistanceA	Failure modes	Cause of failure
'Train A' accelerate	control action is not delivered	Decrease	Maintenance order is not created or sent	Software internal computer failure
				Incorrect MACC Software algorithm
				Incorrect safety distance definition
			Maintenance order is created or sent too late	Software internal computer failure
				Incorrect MACC Software algorithm
				Incorrect safety distance definition
			Speed control interface is not worked	Incorrect DCC connection
				Incorrect Decoders connection
			Speed is not controlled	DCC failure
				Decoders failure
			Speed is controlled too late	DCC failure
				Decoders failure
			Speed data is not received	Incorrect DCC connection
				Incorrect Decoders connection
				DCC failure
				Decoders failure
			Speed data is received too late	Incorrect DCC connection
				Incorrect Decoders connection
				DCC failure
				Decoders failure
			Unexpected case is occurred	A natural disaster
				etc.

V. 결 론

본 논문에서는 하이브리드 위험원 분석을 제안하여, 복잡도가 높아진 철도 안전 시스템의 위험원을 분석할 때 구성 요소의 고장으로 인한 위험원뿐만 아니라 상호작용으로 인한 위험원도 고려하였다.

제안 방법의 유효성 확인을 위해서 사례 연구로 MACC 시스템을 개발하였다. 제안 방법을 이용하여 위험원 분석을 수행하였을 뿐만 아니라, 기존 방법들인 신뢰성 기반 방법과 시스템 이론적 방법으로도 위험원을 분석하였다. 분석 결과를 살펴보면 신뢰성 기반 방법은 시스템 구성 요소의 고장을 방지 또는 회피하기 위한 안전 요구사항이 도출되었고, 시스템 이론적 방법은 시스템의 상호작용으로 인한 위험원으로부터 잘못된 제어 명령을 제어하기 위한 안전 요구사항이 도출되었다. 반면에, 제안한 하이브리드 방법은 예방 대책과 저감 대책을 도출하여 고장 원인뿐만 아니라 의도치 않은 제어로부터도 시스템을 보호할 수 있는 안전 요구사항을 추출하여서, 기존 방법에 비해서 위험원을 더 철저히 분석할 수 있었다. 사례 연구로 구현한 MACC 시스템에서, '열차 A'와 관련해 식별된 안전 요구사항 수를 살펴보면, 신뢰성 기반 방법은 11개, 시스템 이론적 방법은 12개였지만, 하이브리드 기법은 23개로 기존 방법의 위험원을 모두 다루었다.

본 연구에서는 하이브리드 위험원 분석을 통해 시스템의 고장 위험원과 제어 위험원, 즉 두 가지 측면을 모두 고려함으로써 철도 임베디드 시스템의 안전성 향상에 기여하고자 하였다. 다만, 시스템의 규모와 목적에 비해 너무 과한 안전 요구사항이 도출될 수 있다는 우려가 있다. 향후 연구로는 제안한 방법을 더 많은 사례에 적용하는 것과, 도출된 안전 요구사항의 적절성 검증 및 안전 요구사항의 명세 방법을 연구하고자 한다.

References

[1] IIEC, "IEC 62278:2002, Railway Applications, Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS)", International Electrotechnical Commission, 2002.

[2] C. A. Ericson, "Hazard Analysis Techniques for System Safety", Wiley Publishing, 2005.

[3] N. Leveson, "Engineering a safer world: Systems Thinking Applied to Safety", MIT Press, 2005.

[4] NIPA, "SW Safety Guide at Railway Sector", National Information Promotion Agency, 2017.

[5] A. Adulkhaleq, "A System-Theoretic Safety Engineering Approach for Software-Intensive Systems", Stuttguart University, Ph.D. Dissertation, 2017.

[6] European Railway Agency, "Functional Safety Analysis of ETCS DMI", Safety Report, 2009.

[7] M. Ouyang, et. al,, "STAMP-based analysis on the railway accident and accident spreading: Taking the China-Jiaoji railway accident for example", Safety Science, Vol. 48, No. 5, pp. 544-555, Jun. 2010.

[8] S. R. Trammell and B. J. Davis, "Using a Modified Hazop/FMEA Methodology for Assessing System Risk", Proc. of Engineering Management for Applied Technology, IEEE Xplore, pp. 47-53, Aug. 2001.

[9] J. Hwang, et. al,, "A Study on the HAZOP-KR for Hazard Analysis of Train Control Systems", Journal of the Korean Society for Railway, Vol. 13, No. 4, pp. 396-403, Jan. 2010.

저자소개

정 대 희 (Daehui Jeong)

2017년 2월 : 경기대학교 컴퓨터과학과(공학사)
2017년 2월 ~ 현재 : 경기대학교 컴퓨터공학부 석사과정
관심분야 : 소프트웨어 공학, 정형 검증, 소프트웨어 안전성, 시스템 안전성 분석

권 기 현 (Gihwon Kwon)

1985년 2월 : 경기대학교 전자계산학과(이학사)
1987년 8월 : 중앙대학교 전자계산학과(이학석사)
1991년 2월 : 중앙대학교 전자계산학과(공학박사)
1991년 2월 ~ 현재 : 경기대학교 컴퓨터공학부 교수
1999년 ~ 2000년 : 미국 카네기멜론대학 전산학과 연구교수
2006년 ~ 2007년 : 미국 카네기멜론대학 전산학과 연구교수
2014년 ~ 2016년 : 한국정보과학회 소프트웨어공학 소사이어티 회장
관심분야 : 소프트웨어 공학, 정형 검증, 소프트웨어 안전성, 시스템 안전성 분석