Reliability Engineering and System Safety 167 (2017) 177–183

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Application of systems theoretic process analysis to a lane keeping assist system

Haneet Singh Mahajan*, Thomas Bradley, Sudeep Pasricha

College of Engineering, Colorado State University, Fort Collins, CO, United States

ARTICLE INFO

Keywords:
Hazard analysis
Systems theory
STPA
STAMP
Lane keeping assist
ADAS

ABSTRACT

The implementation of autonomous vehicles involves an increase in the number and depth of system interactions in comparison to user-driven cars. There is a corresponding need to address the system safety implications of autonomy. Traditional hazard analysis techniques are not designed to identify hazardous states caused by system interactions. An emerging technique based on systems theory, Systems Theoretic Process Analysis (STPA), allows for inclusion of system-level causal factors by focusing on component interactions. This study researches the application of STPA to a lane keeping assist system, resulting in identification of design constraints and requirements needed to engineer a safer system.

1. Introduction

Technologies associated with connected and autonomous vehicles are nearing mass-market introduction to the mainstream automotive industry. These vehicles incorporate technologies and systems to identify navigation routes, and detect and avoid obstacles with the help of sensors, radars and advanced controls.

Prototype and limited-production connected and autonomous vehicles (CAVs) [1,6,13] are under continued development, and it is universally understood that this development must address safety concerns through a process of system safety and safety-driven design. The concept of system safety is a part of risk management processes based on identification and analysis of hazards using a systems-based approach [22]. Simple automotive systems might consist of processes, sensors, and actuators, but CAV systems will include more complexity in automation (software), models of context, and human-machine interface [20]. With an increase in the complexity of automotive systems, the hazards associated with them have become more complex, thereby demanding changes in the hazard analysis methods that are used to derive safety-driven design requirements.

NPR 8715.3C defines hazard as “a state or a set of conditions, internal or external to a system, that has the potential to cause harm [7].” For this study, we consider that ‘Hazard’ and ‘Failure’ are different terms based on the scope of the analysis. Hazards are associated with safety analyses, whereas failure is studied under reliability engineering [19]. A comprehensive hazard analysis should include hazards that may arise from component failure and unsafe component interactions. Traditional approaches to safety analyses are based on reliability analysis techniques, such as fault tree analysis and event tree analysis [11]. Additionally, ISO 26262, a functional safety standard for the development of electrical/electronic (E/E) systems in road vehicles, defines functional safety as “absence of unreasonable risk due to hazards caused by malfunctioning behavior of electrical/electronic systems [14].” This might hold true for conventional automotive systems, but is not true for more complex systems, including those that involve autonomy. In complex systems, accidents may arise from unintended or unsafe interactions between components that have not failed. The ISO 26262 definition of safety focuses on malfunctioning behavior of E/E systems, but does not consider hazards that can be caused by unsafe interactions even when the E/E systems are not malfunctioning.

Hazard analysis techniques based on systems theory can consider that accidents arise from interactions among components and that there is more than a single causal variable or factor leading to a hazard. An emerging hazard analysis technique, Systems Theoretic Process Analysis (STPA) based on the STAMP (Systems Theoretic Accident Model and Processes) causality model allows for inclusion of new causal factors, such as software flaws, complex human decision-making errors and component interactions that are not identified by traditional hazard analysis techniques [20]. STPA focuses on identifying Unsafe Control Actions (UCAs), and developing design constraints and requirements based on the UCAs. The focus in terms of safety is shifted from *preventing failures* to *enforcing safety constraints*. The STAMP model allows inclusion of interactions and dependencies, leading to a better qualitative model of system behavior [15]. CAVs are software-intensive systems with com-

* Corresponding author.
E-mail addresses: haneetsingh3@gmail.com (H.S. Mahajan), thomas.bradley@colostate.edu (T. Bradley), sudeep@colostate.edu (S. Pasricha).

http://dx.doi.org/10.1016/j.ress.2017.05.037
Received 13 March 2017; Received in revised form 10 May 2017; Accepted 24 May 2017
Available online 25 May 2017

plicated system interactions, which make them particularly amenable to STAMP and other hazard analysis techniques based on systems theory.

An exemplar subsystem for CAVs and intelligent transportation systems is the Lane Keeping Assist (LKA) system. The lane on which the vehicle is travelling is detected and controlled by the LKA system using various input means, such as cameras and GPS position sensors [16,17,21]. LKA systems are a crucial feature of Advanced Driver Assistance Systems (ADAS) that warn the driver in a hazardous situation and/or assist in avoiding an accident. ADAS can minimize accidents by avoiding hazards, and can assist the driver in maintaining safe operating conditions. Functionally, an LKA system, as a component of ADAS, detects lane departure and is responsible for calculating the torque and steering angle needed to steer the car back into its lane [24].

With this understanding of the importance of LKA and other emerging automotive systems in improving the safety of CAVs, this project evaluates STPA using an LKA system case study. This study seeks to use STPA to analyze a proposed control structure of such a system, so as to identify UCAs and derive safety-driven design constraints and requirements.

2. Review of hazard analysis techniques

There are three major hazard analysis techniques suggested by ISO 26262: Fault Tree Analysis (FTA), Failure Modes and Effects Analysis (FMEA), and Hazard and Operability (HAZOP) Analysis [14]. This section reviews these traditional techniques to discuss their utility for hazard analysis of complex systems.

2.1. Fault tree analysis (FTA)

FTA is a deductive (top-down) analysis, that is, the causes for an event are identified and resolved after the event is defined. The goal of this analysis is to determine root causes and calculate the probability of occurrence of hazards or accidents. FTA can be applied to complex dynamic systems by modeling the combinations of fault events that lead to a hazard/accident, called a fault tree [11].

The analysis begins with a top-level event (hazard, accident, or any undesired activity), which flows down into various stages (intermediate events) and ends with the causal factors involved in the occurrence of the event. The FTA involves the following steps:

1) Define the undesired event (top event)
2) Define the scope and boundary of the system
3) Construct fault tree
4) Perform qualitative analysis (single point failures, common cause failures, cut sets)
5) Perform quantitative analysis (probability of top event and importance of basic events)
6) Make decisions and recommendations based on analysis [18]

The entities in the fault tree are connected using logic gates to define the relationship between events and states of the system. Once the fault tree is complete, the probability of occurrence of top events is calculated. Fault trees can alternatively be converted to reliability block diagrams for the same purpose [25,26].

2.2. Failure modes and effects analysis (FMEA)

FMEA is an inductive (bottom-up) reliability engineering analysis technique used to identify potential failure modes and their consequences. Identifying the failure modes is a brainstorming exercise and usually involves people with experience with the system. Once the failure modes are identified, the system design can include preventive and/or mitigative measures. The failure modes are prioritized to be able to classify them based on their effect on the system. The classification system is based on three factors:

1) Severity: The impact of the failure
2) Occurrence: The probability of the failure occurring
3) Detection: The probability of the failure being detected before its impact is realized [26]

All the factors can be on a scale from either 1–5 or 1–10 scale, based on the precision required for the analysis. These three factors are multiplied to form a Risk Priority Number (RPN) metric. This metric reflects the priority of the failure mode. Once the RPNs are calculated for all the identified failure modes, they are prioritized and high-risk failure modes are eliminated or mitigated. The failure modes are resolved either by design improvement or monitoring. There is no definitive RPN threshold over which action must be taken, but as a rule of thumb, teams may focus on the top 20% of the highest RPNs [3].

2.3. Hazard and operability analysis

HAZOP analysis is based on a slightly different accident model than FTA and FMEA, as it uses guidewords which measure the deviation from system parameters [19]. Using guidewords such as *none, more than, less than*, the deviations can be identified and traced to possible causes. HAZOP allows measurement of deviation from expected behavior, especially for software, as observed in [25].

ISO 26262 – Part 3 suggests using HAZOP along with FMEA to identify hazards [14]. HAZOP can use engineering artifacts such as process flow diagrams and piping and instrumentation diagrams to break the system into nodes. Each node is then analyzed to identify deviations from expected behavior, determine causes and their effects. This leads to a cause-consequence analysis that can be used to develop safeguards to detect, prevent, control or mitigate the effects of the hazard [9].

2.4. Limitations of these hazard analysis techniques

Although all of the methods described have utility in analysis and design for reliability, they have limited utility in design for safety in complex systems. Application of traditional techniques such as HAZOP and FTA on software-intensive or complex systems has proven to be tedious, time-consuming and error-prone [23].

FMEA and FTA are based on a chain-of-failure event model, where each hazard or accident is considered to be caused by component failures. Due to this nature of traditional hazard analysis methods, the prevention measures usually involve increasing the reliability of the components or introducing homogeneous redundancy. In many complex systems, unsafe controls can result without component failure, and reliability may not equate with safety. It is important to understand the difference between reliability and safety to realize the need for a new hazard analysis approach. A system can be reliable but unsafe. For example, the LKA system might force the car to stay in lane when the driver is actually trying to avoid an accident by switching lanes. The software in the LKA is reliable, as it detects lane change and steers the car back into its lane, but it is an unsafe action as the driver is aware of a situation, which can cause an accident, that the software is not. On the other hand, a system can be safe, but unreliable. Extending the previous example, the driver is being unreliable when the lane change is performed without the use of a *turn-indicator*, but the action is performed to avoid an accident, and is thus considered a safe action. STPA allows the segregation of safety and reliability by shifting the focus from reliability theory to systems theory and focusing on component interaction accidents, in addition to component failures.

Another weakness of FTA and FMEA is that they are developed through the elicitation of failures or faults, which can lead to discrepancies among analyses. For example, FTA and FMEA seek to quantify the causal factors of failure, causing the analysts to exclude factors that cannot be quantified, either due to their stochastic nature or due to a lack of data. In contrast, STPA focuses on identifying lack of control instead of deviations and allows for inclusion of new causal factors that include software flaws, human decision-making errors, and organizational factors [20]. In general, it is important to include human and organizational factors to allow identifications of hazards resulting from human error, leading to a model that allows human error analysis and prediction [15]. STPA is more inclusive and systematic as it uses engineering artifacts to identify scenarios that might lead to an accident.

3. Systems theoretic process analysis

The goal of STPA is similar to that of any other hazard analysis technique, that is, to identify and analyze hazards and losses to allow their mitigation and/or prevention. Under STPA, a hazard is defined as "a system state or set of conditions that together with a worst-case set of environmental conditions, will lead to an accident (loss) [19]." STPA is a top-down analysis with demonstrated success in developing high-level safety requirements and constraints at the initial stages of system development that are refined as the design evolves [2,5,12,27].

STPA focuses on three basic concepts – safety constraints, hierarchical control structure, and process models. The analysis identifies the safety constraints that were violated for a hazard to occur and investigates the inadequacy of the controls designed to enforce the safety constraints. These safety constraints allow behavioral control of components and enforce safety. Hierarchy in a control structure is crucial to understand the different levels of complexity and to enforce safety constraints on a component level, thus allowing them to be enforced collectively on a system level as well. The control structure is useful in understanding the complexity of component interactions as all components are modeled to be dynamic. This allows safety to be treated as a control problem and not a reliability problem. The process model embodies everything required to control the process effectively. It could be as simple as a few variables and as complex as a model containing many state variables, functional modes and transitions [19].

The STPA technique can be applied during any stage of the system life cycle. When this technique is used for safety-guided design, it allows for safety constraints and requirements to be refined and traced to individual subsystems and components. STPA works on a functional control diagram and uses guidewords based on lack of control to assist in the analysis. The design of the subsystems and the functional control block diagram lead to definitions of a hazard, safety constraints and requirements that can be traced back to the system components as the analysis proceeds. This analysis does not yield a probability related to a hazard as attempting to quantify different aspects of a hazard might lead to omission of stochastic or infrequent causal factors.

The STPA process presented in this paper has four main sequential steps:

1) Define high-level hazards, requirements and constraints based on system-level functionality.
2) Identify hazardous states for different control functions based on the control block diagram, caused by potential lack of control, which can be due to:
   a) A control action being provided but not required;
   b) A control action being required but not provided;
   c) A control action being provided too early or too late (wrong timing);
   d) A control action being provided for the incorrect duration of time;
3) Determine how each hazardous state could occur, that is, determining causal factors by considering the components involved and their interactions in the control loop.
4) Develop additional constraints and requirements based on the identified hazardous states.

The presented process can then be repeated through the various phases of system development, leading to a refined set of constraints and requirements for each subsystem and the interactions between them, which result in safety-driven design processes.

4. Application of STPA to an LKA system

4.1. Accidents and losses

The first step in any safety analysis is to define the accidents and losses related to the system. An accident is "an undesired or unplanned event that results in a loss, including loss of human life or human injury, property damage, environmental pollution, mission loss, etc. [19]"

4.2. High-level hazards, constraints and requirements

The next step is to define what a hazard is for an LKA system and specify some high-level constraints and requirements. We define a set of high-level hazards for an LKA system as:

1) Absence of warning when vehicle moves out of lane, resulting in a collision.
2) No corrective action provided by the system when the car moves out of lane, leading to a collision.
3) Corrective action being provided when it isn't required, resulting in a collision.
4) Corrective action (torque to the steering) being provided in the wrong direction, causing a collision.

These system-level hazards are identified based on an initial functional understanding of an LKA system. Based on these definitions, certain high-level requirements can be defined:

1) The LKA system shall warn the driver when the vehicle is switching lanes without using a *turn-indicator*
2) The LKA system shall provide corrective action if the driver doesn't respond to the warning signs and the vehicle continues to move out of a lane

At this stage of the STPA, these requirements assist in developing a general understanding of the system. But due to the structure of requirements, all the necessary information or ideas may not be clear to the design teams. Therefore, there is a need to define certain design constraints.

1) The LKA system must not allow the vehicle to switch lanes without the correct *turn-indicator* being actuated
2) The LKA system must not perform corrective action if the correct *turn-indicator* is actuated (if the direction of deviation is in the same direction as the *turn-indicator*)
3) The LKA system must verify that corrective action has been performed either from its inputs or feedback from the electrical steering system

4.3. Control structure

After having defined the initial hazards, constraints and requirements, the next step is to develop a control structure. Fig. 1 shows the proposed control structure for an LKA system using a camera as an input for surroundings. The control structure is generalized and does not represent any specific LKA system. However, it does include necessary information to understand the system-level interactions.

Now, STPA is performed on the various control functions present in the control structure. Each control function is analyzed to identify hazardous states (or UCAs) as mentioned in the *Identify Hazardous States* step of the analysis. Each unique UCA and requirement is labeled (H# for hazards, R# for requirements) for tracing different requirements to different hazards, and to simplify the analysis in cases of repetition of hazardous states. Some of the requirements that we derive will verify the need for some of the functions present in the control structure, whereas others lead to inclusion of new functions or subsystems. System-level constraints and requirements allow the design teams to understand the interactions between components better and that is the purpose of the

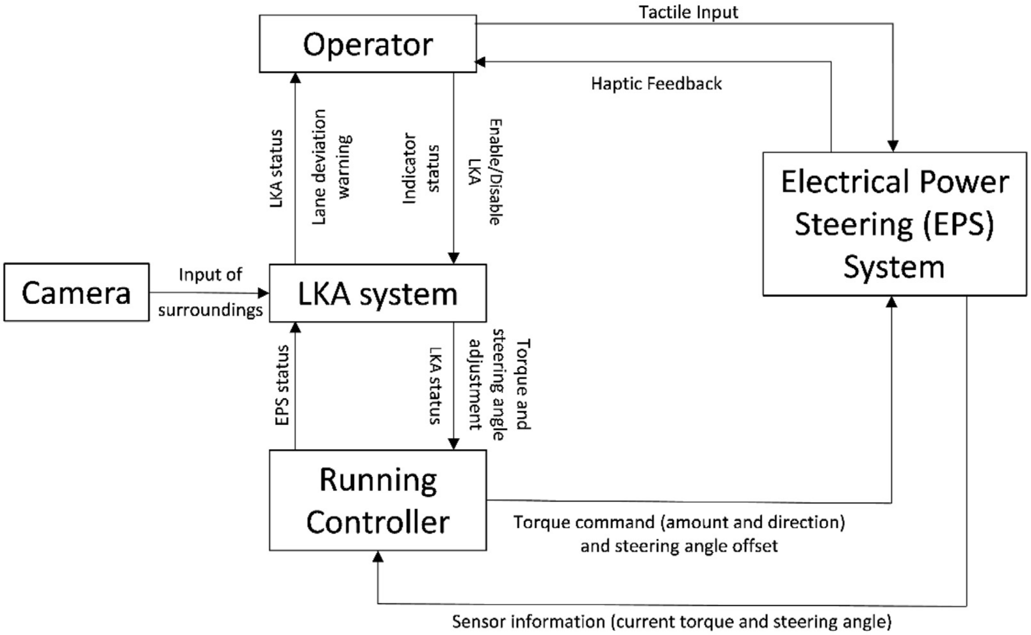
Operator
Tactile Input
Haptic Feedback
LKA status
Lane deviation warning
Indicator status
Enable/Disable LKA
Electrical Power Steering (EPS) System
Camera
Input of surroundings
LKA system
EPS status
LKA status
Torque and steering angle adjustment
Running Controller
Torque command (amount and direction) and steering angle offset
Sensor information (current torque and steering angle)

Fig. 1. Initial control structure showing high-level system interactions.

Table 1
STPA for torque and steering adjustments function.

Control function	**Unsafe control actions (UCAs)**			
	Required but not provided	*Provided but not required*	*Provided but wrong timing*	*Provided but incorrect duration*
Torque and steering angle adjustment (from LKA to running controller)	H1: Torque request isn't transferred, while vehicle continues to drive out of lane	H2: Unexpected torque to the steering	H3: Controller sends torque request at the wrong time	H4: Controller continues to send torque request
	Causal factor(s)			
	1. Incorrect input from camera to LKA.	1. LKA is enabled when it shouldn't be	1. Incorrect input from camera	1. Incorrect input from camera
	2. Misinterpreted lane markings by LKA (system thinks vehicle is in lane)	2. Incorrect camera input	2. Incorrect processing of deviation by LKA	2. LKA frozen
	3. Incorrect *turn-indicator* status transmitted to LKA	3. EPS status not communicated to LKA	3. *Turn-indicator* malfunction	3. EPS status not communicated to LKA
	4. LKA is disabled		4. EPS status communication is delayed	
	Constraint(s)			
	Camera check; accurate detection and processing of lane markings	Camera check; continuous communication of EPS status to LKA; LKA refresh rate	LKA processing time; incorrect refresh rate; camera cycle rate	LKA processing time; camera cycle rate; continuous EPS status communication
	Requirement(s)			
		R1: The running controller shall send the current EPS status signal to the LKA once the torque command has been implemented R2: The running controller shall update the LKA system if there is a mismatch between the sensor information from EPS and the EPS status stored in LKA	R3: The LKA system shall continuously monitor and verify the camera input with the current EPS status	R4: The running controller shall refresh the LKA system if the LKA status is frozen

analysis presented in this paper. The results of the analysis can be used by teams planning to design an LKA system and integrate it within a vehicle, provided a system-level analysis is performed with all the necessary system interactions. The constraints and requirements can assist the teams in gaining a system-level view before manufacturing and integrating the system for testing purposes.

4.4. STPA

Tables 1–4 show the results of STPA for the control functions described in Fig. 2. All the control functions are analyzed, but the ones that do not lead to system-level requirements or constraints are not presented in this paper. Certain constraints presented in the tables are presented as

Table 2
STPA for lane deviation warning to operator.

Control function	**Unsafe control actions (UCAs)**			
	Required but not provided	*Provided but not required*	*Provided but wrong timing*	*Provided but incorrect duration*
Lane deviation warning to operator	H5: Operator does not provide corrective action	H6: Wrong warning misdirecting driver, possibly leading to incorrect torque request to running controller	H6: Wrong warning misdirecting driver, possibly leading to incorrect torque request to running controller	H6: Wrong warning misdirecting driver, possibly leading to incorrect torque request to running controller
	Causal factor(s)			
	1. Incorrect input from camera 2. LKA is disabled when the operator thinks it is enabled 3. Incorrect *turn-indicator* status	1. Incorrect input from camera 2. LKA is enabled when it shouldn't be 3. Incorrect *turn-indicator* status	1. Incorrect input from camera 2. LKA is enabled when it shouldn't be 3. Incorrect *turn-indicator* status	1. Incorrect input from camera 2. LKA is enabled when it shouldn't be 3. Incorrect *turn-indicator* status
	Constraint(s)			
	Initial camera check; camera self-health check; *turn-indicator* check	Initial camera check; camera self-health check; *turn-indicator* check	Initial camera check; camera self-health check; *turn-indicator* check	Initial camera check; camera self-health check; *turn-indicator* check
	Requirement(s)			
	R5: The running controller shall confirm the LKA is functional with the operator when the system is enabled	R6: The LKA shall verify driver responsiveness before providing warnings and/or corrective action	R6: The LKA shall verify driver responsiveness before providing warnings and/or corrective action	R6: The LKA shall verify driver responsiveness before providing warnings and/or corrective action

Table 3
STPA for sensor information being sent from EPS to running controller.

Control function	**Unsafe control actions (UCAs)**			
	Required but not provided	*Provided but not required*	*Provided but wrong timing*	*Provided but incorrect duration*
Sensor information to running controller	H7: Controller is unaware of any changes implemented by the EPS	N/A	H7: Controller is unaware of any changes implemented by the EPS	N/A
	Causal factor(s)			
	Sensor malfunction		Sensor malfunction	
	Constraint(s)			
	Sensor diagnostics		Sensor diagnostics	
	Requirement(s)			
	R7: The running controller shall transfer torque requests to EPS only if sensor information is received		R7: The running controller shall transfer torque requests to EPS only if sensor information is received	

Table 4
STPA for LKA status being sent to the operator.

Control function	**Unsafe control actions (UCAs)**			
	Required but not provided	*Provided but not required*	*Provided but wrong timing*	*Provided but incorrect duration*
LKA status to operator	H8: Operator is unsure if LKA is on or not	H9: LKA is on when not needed	H8: Operator is unsure if LKA is on or not	H8: Operator is unsure if LKA is on or not
	Causal factor(s)			
	1. Communication breakdown between LKA and operator 2. LKA malfunction	LKA malfunction	1. Communication breakdown between LKA and operator 2. LKA malfunction	1. Communication breakdown between LKA and operator 2. LKA malfunction
	Constraint(s)			
		Incorrect enable signal		
	Requirement(s)			
		R8: The running controller shall verify operator intention to enable LKA		

a guide to design teams, such as initial camera check, *turn-indicator* relay check, camera fidelity check, and sensor diagnostics. These constraints imply that the subsystems should be functional before the interactions between them are analyzed, and the design teams can develop further requirements and constraints to satisfy the system-level constraints.

These tables show the analysis of the control functions that led to system-level constraints and requirements, thus encouraging safety-driven design decisions.

One of the outcomes of the analysis is a set of requirements that corresponds to one or more hazards. These requirements are used to understand the safety implications of existing control functions and, if

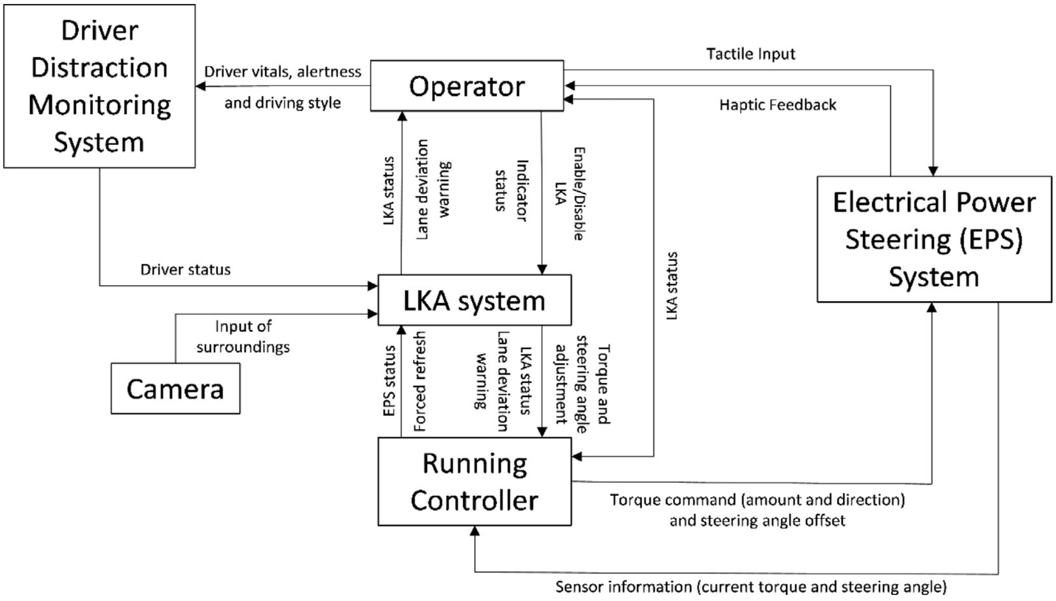
Driver Distraction Monitoring System
Driver vitals, alertness and driving style
Operator
Tactile Input
Haptic Feedback
Electrical Power Steering (EPS) System
LKA status
Lane deviation warning
Indicator status
Enable/Disable LKA
LKA status
Driver status
LKA system
Input of surroundings
Camera
EPS status
Forced refresh
Lane deviation warning
LKA status
Torque and steering angle adjustment
Running Controller
Torque command (amount and direction) and steering angle offset
Sensor information (current torque and steering angle)

Fig. 2. Updated control structure including changes derived from various requirements.

Table 5
List of hazards.

Hazard number	Hazard
H1	Torque request isn't transferred, while vehicle continues to drive out of lane
H2	Unexpected torque to the steering
H3	Controller sends torque request at the wrong time
H4	Controller continues to send torque request
H5	Operator does not provide corrective action
H6	Wrong warning misdirecting driver, possibly leading to incorrect torque request to running controller
H7	Controller is unaware of any changes implemented by the EPS
H8	Operator is unsure if LKA is on or not
H9	LKA is on when not needed

required, modify the control structure to a safer one. Fig. 2 shows the updated control structure based on some of the requirements developed during the analysis. The changes made to the control structure include three new control functions: forced refresh from running controller to the LKA system (driven by R4), lane deviation warning from the LKA system to the running controller (driven by R7 and R8), LKA status verification function between the running controller and the operator (driven by R5 and R8) and lastly, there is an additional subsystem – Driver Distraction Monitoring System (DDMS, driven by R6). All these changes are driven by the requirements mentioned in parenthesis and constrain the operation of the system to avoid hazards.

5. Results

This STPA was performed for an LKA system in the concept development stage, considering hazards and causal factors based on system interactions and not just failure events. The analysis is the first step in the development and integration of such a complex system. The identification of unsafe control actions and the causal analysis led to the development of system-level constraints and requirements, which ensure that the system satisfies the top-level safety goals. The requirements also allow the design teams to understand the intent of different subsystems and control functions. The process, when carried out from the beginning of the development of a system, encourages safety-driven design decisions. Tables 5 and 6 show the list of hazards and requirements achieved through this analysis. Fig. 3 shows how the requirements can be traced to various hazards.

Table 6
List of requirements.

Requirement number	Requirement
R1	The running controller shall send the current EPS status signal to the LKA once the torque command has been implemented
R2	The running controller shall update the LKA system if there is a mismatch between the sensor information from EPS and the EPS status stored in LKA
R3	The LKA system shall continuously monitor and verify the camera input with the current EPS status
R4	The running controller shall refresh the LKA system if the LKA status is frozen
R5	The running controller shall confirm the LKA is functional with the operator when the system is enabled
R6	The LKA shall verify driver responsiveness before providing warnings and/or corrective action
R7	The running controller shall transfer torque requests to EPS only if sensor information is received
R8	The running controller shall verify operator intention to enable LKA

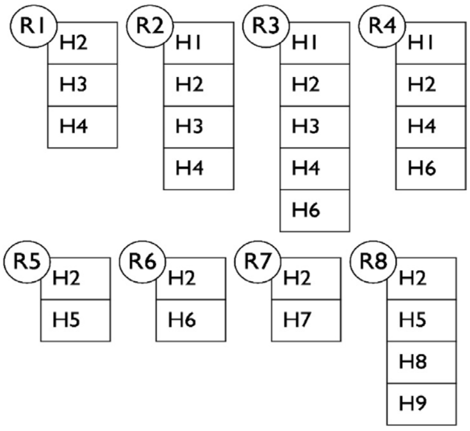

Fig. 3. Traceability between requirements and hazards.

6. Discussion

By using this LKA system as a case study for the application of STPA, we can identify some key outcomes and challenges in the application of STPA to this system.

6.1. Human-machine interface (HMI)

First, one of the key challenges in application of STPA is that it has no special considerations for deriving unsafe control actions that result from human error. Human behavior is considered analogous to software. In this study, the human operator is an integral component of the control loop who must perform functions associated with actuation, sensing, and feedback. Many of the unsafe control actions that are associated with the LKA system were found to be associated with control loops that included the human operator. It would be beneficial to incorporate a model of human decision-making in hazardous situations to allow systems to better react and adjust to the operator's behavior. To develop requirements associated with these operator functions, this study proposed that a DDMS is an integral component of a safety-driven LKA system design. Functionally, this DDMS is providing assurance that the driver is performing the intended actuation, sensing, and feedback control actions. In the literature on DDMS [4,8,17,24], the primary functions of a DDMS that are described include assessing fatigue and distraction. In the safety context, it is not sufficient to assess impairment, instead, the DDMS must be assessing the driver's function including aspects of situational awareness, decision making, and performance [10]. This is a more inclusive set of requirements than has been the domain of DDMS research and development to date, but this study points to its importance as an aspect of system safety in the LKA and other HMI-centric systems.

6.2. STPA in a requirements engineering process

The analysis performed in this paper has been used to derive safety-driven design requirements (R1-R8) at system-level. These requirements are inputs to the subsystem design processes for each of the components, functions, and interfaces described in the control structure proposed in Fig. 3. As the design progresses, and as changes are made to the system during the design and testing phases, they can be analyzed by updating the control structure and repeating the process demonstrated in this paper. These result demonstrate that STPA can be used to identify additional hazardous states and develop safety-critical constraints and requirements, even during the initial stages of development.

7. Future scope

A more detailed analysis can be performed by including other subsystems involved in autonomous vehicles and their interactions, for example, the interactions between an LKA system and an adaptive cruise control system. The results of the analysis would be more holistic and comprehensive, and would provide guidance for the design of safer autonomous vehicles.

The hazard analysis process can be continued throughout the development stage to improve the safety of the system. The design teams can use the hazard analysis and the control structure to develop individual subsystem requirements and constraints, and then update the control structure with more specific interactions and functions. This can improve the analysis and make it more specific during the design and testing stages.

Acknowledgments

The support of the US Department of Energy and various industrial sponsors through the EcoCAR 3 program is gratefully acknowledged.

References

[1] Ackerman E. Google's autonomous car takes to the streets 2010.
[2] Alemzadeh H, Chen D, Lewis A, Kalbarczyk Z, Raman J, Leveson N, et al. Systems-theoretic safety assessment of robotic telesurgical systems. In: International conference on computer safety, reliability, and security. Springer International Publishing; 2015, September. p. 213–27.
[3] Ben-Daya M. Failure mode and effect analysis. In: Handbook of maintenance management and engineering. London: Springer; 2009. p. 75–90.
[4] Bergasa LM, Nuevo J, Sotelo MA, Barea R, Lopez ME. Real-time system for monitoring driver vigilance. IEEE Trans Intell Transp Syst 2006;7(1):63–77.
[5] Bjerga T, Aven T, Zio E. Uncertainty treatment in risk analysis of complex systems: The cases of STAMP and FRAM. Reliability Engineering & System Safety 2016;156:203–9.
[6] Broggi A, Bertozzi M, Fascioli A, Conte G. The experience of the ARGO autonomous vehicle. Singapore: World Scientific Publishing; 1999.
[7] Dezfuli H, Benjamin A, Everett C, Smith C, Stamatelatos M, Youngblood R. NASA system safety handbook, vol. 1; 2011. System Safety Framework and Concepts for Implementation.
[8] Dong Y, Hu Z, Uchimura K, Murayama N. Driver inattention monitoring system for intelligent vehicles: a review. IEEE Trans Intell Transp Syst 2011;12(2):596–614.
[9] Dunjó J, Fthenakis V, Vílchez JA, Arnaldos J. Hazard and operability (HAZOP) analysis. A literature review. J Hazard Mater 2010;173(1):19–32.
[10] Endsley MR. Toward a theory of situation awareness in dynamic systems. Hum. Factors 1995;37(1):32–64.
[11] Ericson CA. Hazard analysis techniques for system safety. John Wiley & Sons; 2015.
[12] Fleming CH, Spencer M, Thomas J, Leveson N, Wilkinson C. Safety assurance in NextGen and complex transportation systems. Saf Sci 2013;55:173–87.
[13] Funkhouser K. Paving the road ahead: autonomous vehicles, products liability, and the need for a new approach. Utah L Rev 2013;437.
[14] International Organization for Standardization. ISO 26262: road vehicles – functional safety. Geneva: Author; 2011.
[15] Kariuki SG, Löwe K. Integrating human factors into process hazard analysis. Reliab Eng Syst Saf 2007;92(12):1764–73.
[16] Kim, HW. (2012). *U.S. Patent No. 8,095,266*. Washington, DC: U.S. Patent and Trademark Office.
[17] Kutila M, Jokela M, Markkula G, Rue MR. Driver distraction detection with a camera vision system. Image processing, 2007. ICIP 2007. IEEE international conference on, vol. 6. IEEE; 2007, September. VI-201.
[18] Lambert, H.E. (2004). *Use of fault tree analysis for automotive reliability and safety analysis* (No. 2004-01-1537). SAE Technical Paper.
[19] Leveson N. Engineering a safer world: systems thinking applied to safety. MIT Press; 2011.
[20] Leveson N. A systems approach to risk management through leading safety indicators. Reliab Eng Syst Saf 2015;136.
[21] Mineta, K., Unoura, K., & Ikeda, T. (2003). *Development of a lane mark recognition system for a lane keeping assist system* (No. 2003-01-0281). SAE Technical Paper.
[22] Moriarty B. System safety engineering and management. John Wiley & Sons; 1990.
[23] Papadopoulos Y, McDermid J, Mavrides A, Scheidler C, Maruhn M. Model-based semiautomatic safety analysis of programmable systems in automotive applications. In IEEE Int. Conf. Advanced Driver Assistence Systems (ADAS2001) 2001:53–7.
[24] Pohl J, Birk W, Westervall L. A driver-distraction-based lane-keeping assistance system. Proc Inst Mech Eng Part I 2007;221(4):541–52.
[25] Sinha P. Architectural design and reliability analysis of a fail-operational brake-by-wire system from ISO 26262 perspectives. Reliab Eng Syst Saf 2011;96(10):1349–59.
[26] Stamatis DH. Failure mode and effect analysis: FMEA from theory to execution. ASQ Quality Press; 2003.
[27] Stringfellow MV, Leveson NG, Owens BD. Safety-driven design for software-intensive aerospace and automotive systems. Proc IEEE 2010;98(4):515–25.