Safety and Security Analysis of AEB for L4 Autonomous Vehicle Using STPA

Shefali Sharma
Electrical and Computer Eng., University of Waterloo, Waterloo, Canada
s335shar@uwaterloo.ca

Adan Flores
Electrical and Computer Eng., University of Waterloo,Waterloo, Canada
aaflores@uwaterloo.ca

Chris Hobbs
QNX Software Systems Limited, Kanata, Canada
chobbs@qnx.com

Jeff Stafford
Renesas Electronics America Inc., Farmington Hills, USA
jeff.stafford@renesas.com

Sebastian Fischmeister
Electrical and Computer Eng., University of Waterloo, Waterloo, Canada
sfischme@uwaterloo.ca

Abstract

Autonomous vehicles (AVs) are coming to our streets. Due to the presence of highly complex software systems in AVs, there is a need for a new hazard analysis technique to meet stringent safety standards. System Theoretic Process Analysis (STPA), based on Systems Theoretic Accident Modeling and Processes (STAMP), is a powerful tool that can identify, define, analyze and mitigate hazards from the earliest conceptual stage deployment to the operation of a system. Applying STPA to autonomous vehicles demonstrates STPA's applicability to preliminary hazard analysis, alternative available, developmental tests, organizational design, and functional design of each unique safety operation.

This paper describes the STPA process used to generate system design requirements for an Autonomous Emergency Braking (AEB) system using a top-down analysis approach to system safety. The paper makes the following contributions to practicing STPA for safety and security:

1. It describes the incorporation of safety and security analysis in one process and discusses the benefits of this;
2. It provides an improved, structural approach for scenario analysis, concentrating on safety and security;
3. It demonstrates the utility of STPA for gap analysis of existing designs in the automotive domain;
4. It provides lessons learned throughout the process of applying STPA and STPA-Sec [1].

2012 ACM Subject Classification Hardware → Safety critical systems; Networks → Cyber-physical networks

Keywords and phrases Functional Safety, Security, STAMP, STPA, STPA-Sec, ISO 26262, AEB, Advanced Driver Assistance Systems (ADAS), Automated Vehicles, SoC (System-On-Chip)

Digital Object Identifier 10.4230/OASIcs.ASD.2019.5

[1] STPA-Sec identifies and frames the security problems [10]

1 Introduction

AV functionality is rapidly adopted through ADAS technology today. Combining this rate of adoption with the complexity of the autonomous vehicle's system architecture and its use of complex SoC, it is essential for Tier-1[2] and semiconductor suppliers to be diligent in their collaborative effort to design for functional safety and for the mitigation of cybersecurity threats impacting functional safety.

STPA is a new hazard analysis technique and a new model of accident causation, based on systems theory rather than reliability theory [4]. STPA has the same goals as any other hazard analysis technique, that is, to recognize scenarios leading to identified hazards so that they can be eliminated or controlled. STPA, however, has an innovative theoretical basis or accident causality model. STPA is designed to address increasingly common component interaction accidents, along with component failure accidents, which can result from design flaws or unsafe interactions among non-failing (operational) components [3]. In fact, the causes identified using STPA are a superset of those identified by other techniques [4].

This paper provides an example of applying STPA to an AEB system primarily designed for functional safety as well as to mitigate risks associated with cybersecurity vulnerabilities. In this, we have combined functional safety analysis with safety-relevant security analysis.

A methodology is defined to analyze functional safety and cybersecurity, first for the AEB system, and then for the interactions, searching specifically for security vulnerabilities that might contribute to safety hazards.

The next step in the analysis is the identification of accidents and unacceptable losses along with accident hazards and unacceptable loss hazards. We define accident hazards and unacceptable loss hazards, keeping in mind that the implementation of the AEB system is on an L4 AV. Because of the level of autonomy of the vehicle, it is safe to assume there is no driver interaction for the control of the vehicle or the AEB system. In this analysis, the system hazards lead to high-level system constraints and further refinement in STPA Steps 1 and 2.

As we move forward in the analysis, while applying the STPA process, additional dependencies are going to be identified. Knowing this, we can define a basic initial high-level control structure [3] which will be updated in later steps of the analysis. The final control diagram [4] captures the dependencies from both a safety and cybersecurity perspective.

From the high-level control diagram, the next step is to identify CAs (Control Actions). Evaluation of potential hazardous sources is shown in the refined control diagram, considering all of the diagram's inputs and outputs. We also considered component failure, but the analysis is not limited to this. Instead, it presents all aspects of the system's performance, including cybersecurity features negatively impacting functional safety. From this analysis, we are defining a set of causal factors and causal accident scenarios.

The novelty of this paper lies in the addition of a more systematic approach to the conventional STPA approach. Identifying the scenarios by analyzing the components associated with the control flow, and the causal factors corresponding to each scenario, constitutes the next step. From the causal factors, we are refining the safety constraints so that they can produce technical safety requirements (TSRs). Comparison of the TSRs against an existing

[2] Companies which supply components directly to the original equipment manufacturer (OEM), that set up the chain.
[3] See Figure 3.
[4] See Figure 4.

autonomous vehicle design (the autonomy vehicles designed as ASIL-D L4 fail-operational systems) is carried out to identify design gaps for future improvement. This gap analysis on an existing system demonstrates how to make safety and security design changes part of a continuous improvement that must be at the heart of every safety culture.

2 How the analysis started

We started by reviewing an existing autonomous vehicle in need of formal safety analysis. The initial plan was to use a conventional Hazards Analysis and Risk Assessment (HARA) analysis because the group already had experience using this method. But then we learned about STPA and decided to assess its suitability for a system of this scale. We had read reports of its application to much larger systems [3] and wanted to determine whether it would scale to a single, embedded system. Using this approach, we can generate high-level safety constraints in the early stages of development. These constraints can then be tailored to generate detailed safety requirements on individual components of the analyzed system[8].

To avoid biasing our results, we established that the safety analysis should be as general as possible without being directly involved with the current implementation . Thus the result of this analysis was a list of technical safety requirements which we could use to perform an analysis on the current physical architecture and find possible security and safety issues.

We needed to select vehicle functionalities that played an important role in vehicle and occupant safety. The vehicle component also had to be a part of a well-contained function to complete the analysis in the time span available. For these reasons, we selected the L4 AEB function for our analysis.

2.1 The AEB subsystem

An AEB system of L4 AV aid in avoiding accidents by identifying potential collisions with the help of a perception system (LIDAR, RADAR, stereo vision, etc.), computing localization, path planning and determining object trajectory. If a collision is unavoidable, these systems prepare the vehicle to minimize the impact by lowering its speed. It is important to note that the AEB itself is independent of the normal braking system of the vehicle. Once the AEB has identified a potential threat, it takes control of the braking system to mitigate the threat. This functionality has a significant effect on the safety of the vehicle and its occupants, making it an excellent vehicle subsystem for our analysis.

When looking at the distances between the vehicles as shown in Figure 1, we can establish safety thresholds. The first threshold is the warning distance that notifies the AV when the proximity between ego vehicle and the vehicle in front is becoming dangerous; it is recommended for the ego vehicle to start slowing down and increasing the distance between the vehicles. At this distance, the probability of a collision is low. The next threshold is the normal braking limit. At this distance, the normal braking system of the vehicle starts slowing down the vehicle. If the braking system is unable to slow down the vehicle and increase distance, the vehicle will reach the Collision Imminent Braking distance (CIBd) and will activate the AEB system. At this point the collision probability is high, and the AEB needs to take immediate action. The AEB's objective is to stop or slow down the vehicle before it reaches the Minimum Safe Distance (MSD). The MSD threshold is the only fixed value amongst all the thresholds. The rest of the values are dependent on the road conditions (weather and road surface) and the speed of the vehicle.

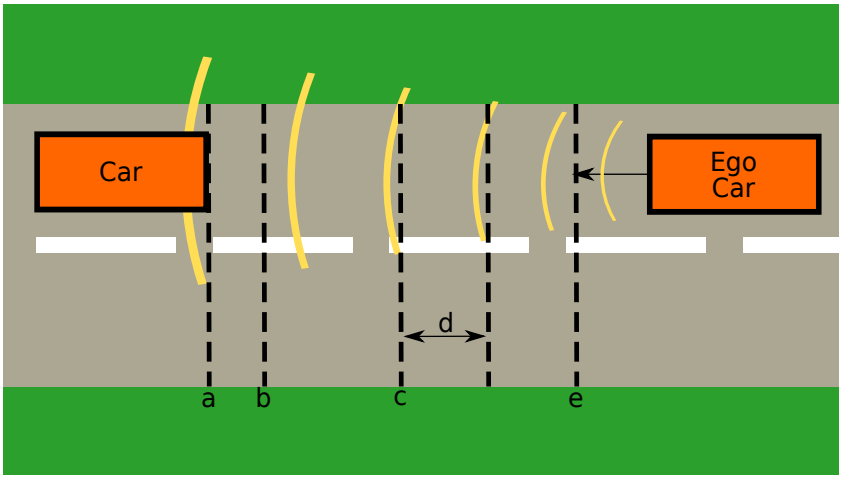

Figure 1 Threshold distances for the braking system.

3 Methodology

The methodology used in the current approach combines safety and security analysis. This approach considers the functional safety and the security-affecting safety. Figure 2 presents the methodology we are using for the STPA analysis[2]:

1. Define analysis scope
   a. Accidents
   b. Hazards
   c. High-level constraints
2. Develop control structure diagram
3. Identify unsafe control actions
   a. Unsafe control actions
   b. Corresponding safety constraints
4. Identify the occurrence of unsafe control actions
   a. Hierarchical control structure with the process model
   b. Causal factors, scenarios, and refined safety constraints
   c. Technical safety constraints

The elements 1(c), 3(b), and 4(c) constitute the STPA analysis report which defines the safety constraints for a safer and more secure system.

The analysis considers a detailed analysis of various blocks of Figure 2. The constituents of the multiple blocks are referred with an identifier as the various parts of each block, to serve as a starting ground for the next block.

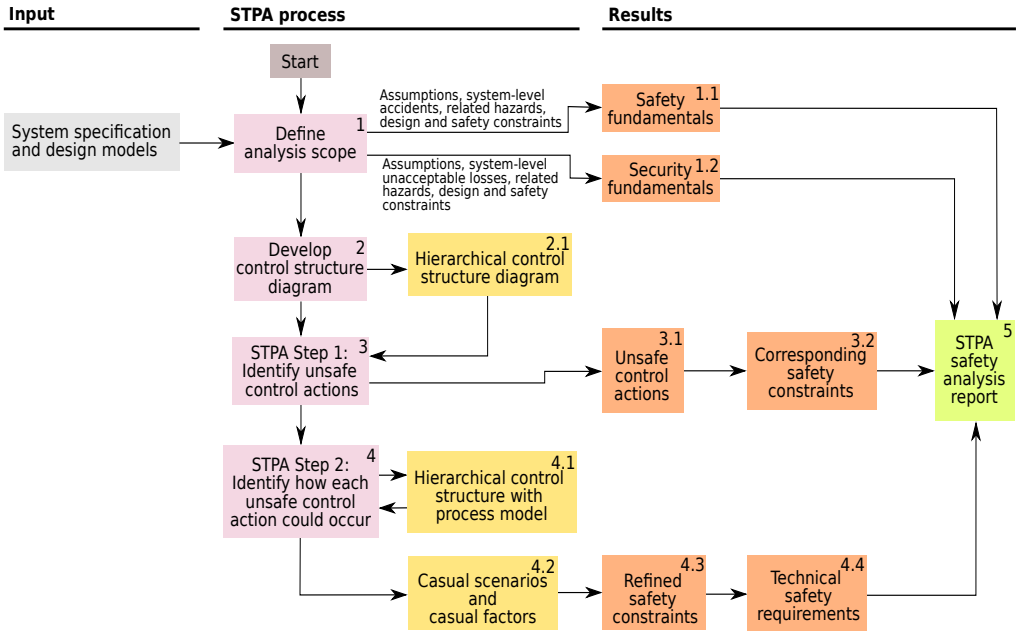

Figure 2 STPA methodology.

3.1 Scope

The methodology begins by defining the scope of the analysis. For the system under consideration, the scope is as follows: "*The analysis presents functional safety analysis for AEB for an AV using vehicle state and environmental data analysis to contribute to the safety of the passengers and environment.*"

3.1.1 Assumptions

After defining the scope, the next step is to define certain conditions that serve as the basis for analysis development. Thus, the analysis considers certain assumptions related to the working conditions. These conditions are also helpful in setting the limits to the analysis. Although, the authors recognize that it would be beneficial to further analyze the assumptions from the perspective of an expanded scope. Here are a couple of examples:[5]:

Assumption 1: AEB functions for collisions from all angles, not just traditional forward-collisions (no lateral maneuvering or acceleration commanded, considering only the brake actuation).

Assumption 5: Path prediction of surrounding mobile objects is available to the AEB system.

There are certain logical conditions behind including these assumptions in the analysis. General cases assume collision primarily from the front. This analysis, however, also examines projected paths of side objects relative to the AV projected path. Hence, the Assumption 1.

[5] These are some of the assumptions we are referring here from the analysis. To have consistency with the report [6], we are using the same identifiers.

The analysis considers an assumption about the availability of data from the surroundings, such as for calculating the collision imminent braking distance and path prediction from the surrounding mobile objects. Hence, Assumption 5.

Some of the assumptions also consider certain conditions outside the scope of the analysis. For example:

- The variation in braking performance based on the mechanical condition of AV tires,
- The sensor performance can be negatively impacted by maintenance or improper care,
- No manufacturing defects and
- All the components are correctly working as they are quality checked and properly maintained.

3.1.2 Accidents

An accident is an undesired or unplanned event that results in the loss of a human life, human injury, property damage, etc. The accidents considered in the analysis are:

A1: The AV collides with a mobile object.
A2: The AV collides with an immobile object.
A3: The AV passengers injured without collision.

In defining the accidents, we first discussed various scenarios that the AV can encounter on the road. Next, we grouped the elements of the scenarios into different categories: vehicles, pedestrians, cyclists, stationary objects, etc. As the analysis was evolving, these subsets posed certain problems; for example, a dustbin could start off as a stationary object, but due to the wind, could start rolling on the road and become non-stationary. We decided that instead of defining it by its current state of activity, we can describe it with its innate ability. So after refinement we devised two subsets: mobile and immobile. For example: if a mailbox were on an HD map, it would be an immobile object. If that same mailbox were blown from its bolts by high wind and became non-stationary, it would be a mobile object requiring identification of the AD sensor system because it is no longer in its original position as shown in the HD map. Here, "mobile" is anything that can move, irrespective of the external influence. Thus A1 and A2 are considered as two potential accidents for the analysis. Also, as in the definition of accident, anything that causes harm to human occupants needs to be considered and is stated as A3. While sitting inside the AV, under certain circumstances such as sudden braking (braking deceleration exceeds the safety physics to passengers) can harm the occupants even when there is no collision.

The next step in the analysis was to define system-level hazards. These are the system states or set of conditions, which together with a particular set of worst-case environmental conditions, would probably lead to an accident.

3.1.3 System level hazards

System level hazards can lead to accidents considered in the analysis. Some of the hazards are listed below:

AH1: AV does not maintain Minimum Safe Distance (MSD) from a Forward Mobile Object (FMO).
AH2: AV does not maintain MSD from Prohibited Area (PA).
AH3: AV occupants exposed to unhealthy g-forces in vehicle exceeding the safety threshold of AV.

Maintaining a safe distance from a vehicle in front is a necessary condition for AEB. If the vehicle is unable to keep MSD from a forward mobile object, then this could be the potential cause of an accident and thus become a hazard that could lead to an accident. The condition for the MSD from an FMO is a prerequisite for the safety of the AV. There are certain areas which have restricted access to traffic. The AVs should ensure that they do not enter such areas and this has been considered – in the analysis as AH2. PA can mean any area – military field, recent accident site, landslide site, etc., – AV's design is not suitable for L4 functionality in a PA. The thresholds predefined in the system related to BFC (Braking Force Command) shall always be complied with because they have the potential to harm the occupants if they exceed a certain threshold level and thus constitute a hazard for the analysis (AH3).

After the identification of hazards, the next step was to describe high-level constraints. These prevent the accident from occurring. Thus, HLCs (High-Level Constraints) provide the set of requirements with which the system shall comply to be functionally safe. These are defined consistently to have traceability to the corresponding hazards. Using a consistent structure can be helpful for the automation of the process. Although this analysis doesn't automate the process, consistency in the structure helped in having a symmetric structure.

During this analysis, we were struggling with the question of whether we should generate two different reports relating to safety and security or whether they should be merged into one. We realized that safety and security are closely interlinked and therefore merged them into one single analysis.

For example: If the AV speed sensor information is spoofed (security threat) then it can lead to a hazardous scenario ultimately leading to an accident (safety threat).

If, due to delayed EPS sensor information (safety threat), BFC fails to set the braking force = 0% even after the removal of earlier hazard, this situation could lead to an unnecessary halt, and thus personal identifiable information of occupants could be inferred (security threat).

3.1.4 High-level safety constraints

High-level safety constraints define the initial set of safety requirements for the system.

3.2 STPA Step 1

The identification of unsafe control actions and the corresponding safety constraints are discussed in this section.

3.2.1 Safety control structure

The control structure is a preliminary process model for the system. It is a functional decomposition of the system. While working on the control structure, we faced certain challenges such as level of detail to be considered. For the sake of a systematic and structured approach, a control structure is the most crucial thing for the safety analysis . We should only consider the blocks responsible for significant functionality such as controller, actuator, process, and feedback. The structure is only a generic one and does not consider the level of granularity. It gives us an overview of how the execution of instruction is taking place without considering the complete internal functionality of the various components involved. Here follows the description of various blocks within the analysis:

Controller: In the system under analysis, the AEB controller is responsible for generating and controlling the BFC.
Actuator: In this system, brakes are the actuator responsible for implementing the BFCs.
Controlled process: The AEB controls the braking of the vehicle.
Feedback: The feedback from the vehicle state and the surrounding environment through the sensors is collected in the state estimator, and thus constitutes the feedback network.

The control structure for the system under consideration is as shown in the Figure 3.

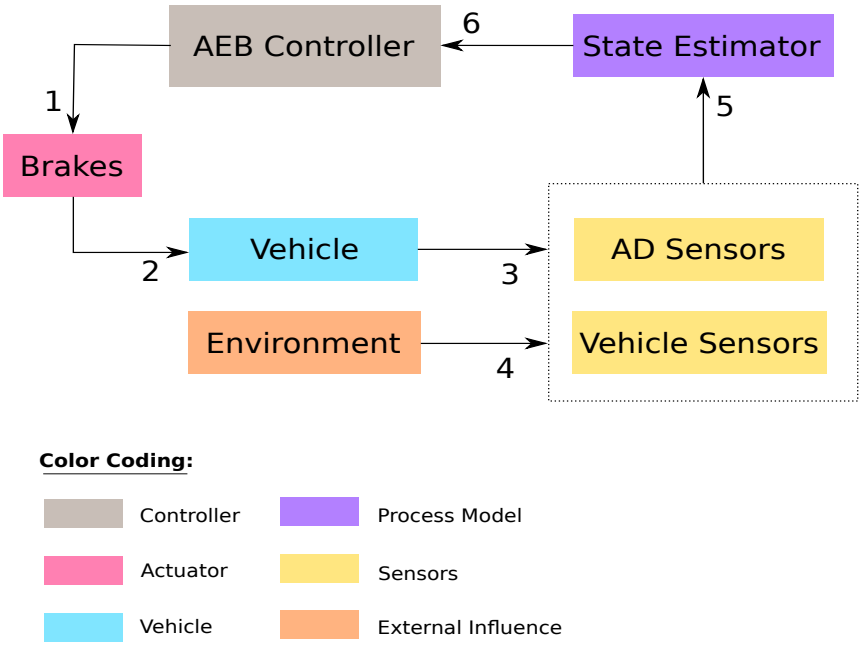

Figure 3 Control loop structure.

By following the STPA process diligently, through detailed use of refined control diagrams, we have a reference to verify that the hazards identification is adequate, and through continued refinement, a benchmark for the design to support continuous improvement over the life of the item. During the analysis, we struggled with the level of detail to be present in the control loop diagram. After creating several revisions of the control loop, we concluded that it should be generic in form and that a further level of detail would not add value to the analysis. For the Control Loop, it shall be in basic generic form and the later stages shall consider the details.

3.2.2 Unsafe control actions

This step performs the identification of the unsafe control actions each component can create which helps in refining the safety requirements and constraints of the system. It will determine the causes of these unsafe control actions. The UCAs are defined using the control actions that can lead to accidents. So, this analysis is considering two control actions based on the control diagram. Here we have taken the BFC (Braking Force Command) coming from the controller; it is only the command and not the force. Two states considered in the analysis are: BFC disengaged (0%), and BFC engaged (modulated engagement ranging from 0% – 100%). After the identification of control actions, the next step is to identify the potential causes of unsafe control.

The four ways a controller can provide unsafe control are the following [3]:

1. A control action required for safety is not provided.
2. An unsafe control action is provided.
3. A potentially safe control action is provided too late or too early (at the wrong time) or in the wrong sequence.
4. A control action required for safety is stopped too soon or applied for too long.

We considered these four categories as a basis for identification of the control table entries. Some of the unsafe control actions considered are listed here:

UCA 1: AEB does not provide BFC when AV is at a closer distance than the CIBd.
UCA 3: AEB does not provide required braking force value when AV is at a closer distance than the CIBd.

If BFC is not applied even when the AV is within the CIBd from an object, then this can be a potential unsafe control action, which could lead to an accident. Hence, UCA 1 belongs to the category of "control action required, but not provided." Another UCA is when the BFC is applied, but the braking force $<$ RDR (Required Deceleration Rate) can also lead to an accident, and is therefore an unsafe control action. Similarly, other UCAs are considered, based upon the time of application of BFC and the total time span of BFC application. Thus, the UCA table is formed.

3.2.3 Safety constraints

The UCAs help to find reasons behind unsafe actions and guide design engineers to eliminate or control them. We referred to table 1 for UCAs, and SCs sets the requirements for the systems. The refined safety constraints are defined in a consistent language as follows:

SC 1: AEB shall provide BFC when AV is at a closer distance than the CIBd.
SC 3: AEB shall provide required braking force value when AV is at a closer distance than the CIBd.

3.3 STPA Step 2

This section identifies the reasons behind the unsafe control actions.

3.3.1 Causal factors and causal accident scenarios

After the identification of the unsafe control actions, we followed STPA Step 2 (Figure 2) to identify the potential causes of unsafe control actions, to understand their presence and how to prevent their occurrence [9]. However, accidents can still occur even without unsafe control actions if, for example, correct and safe control actions are provided, but not executed by other components in the system. The identification of the causal factors can identify a violation of safety constraints despite safe control actions; this is important.

To study the scenarios and causal factors corresponding to each UCA, we made a structured approach:

1. Identify scenarios using UCAs.
2. Identify causal factors corresponding to each scenario by analysing the components associated with the control flow diagram.

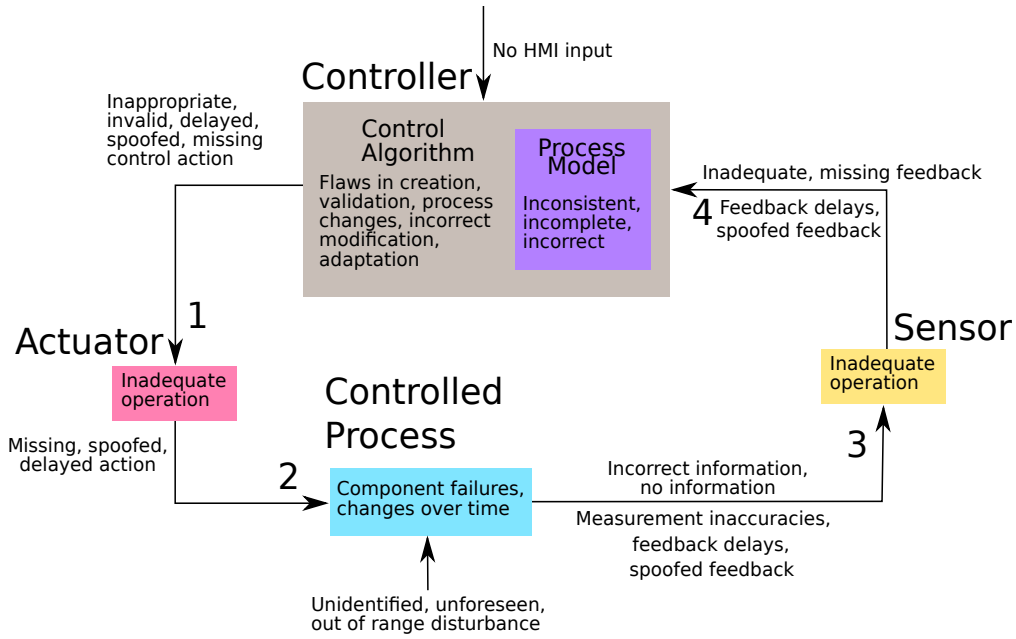

Figure 4 Control loop structure.

Table 1 Table for identification of causal factors.

Blocks	Actions	Reasons
Sensors, Controller, Actuator, Controlled process	Missing information	Due to spoofing, component failure, electrical requirements not met, communication failure
	Inadequate information	
	Incorrect information	
	Delayed information	

We studied the STPA problem as a whole and took parts from the methods available from various researchers in the field [4], [9]. Then we created a hybrid to perform the required analysis.

For example: For the scenario table we tried to use the conventional STPA approach, but found that for our analysis the basic scenarios are sufficient and other detailed scenarios (scenarios arising from feedback issues, etc.) merely lead to redundant scenarios. We created twelve scenarios, but when we started defining the causal factors after three scenarios, they began to repeat and became redundant for our purposes. So we removed the detailed scenarios and analyzed only basic, generic scenarios.

Table 1 provides a systematic and structured approach to analyzing the causal factors. For each of the four blocks in the control structure, we considered four actions: the information is missing, inadequate, incorrect and delayed. The reasons behind these unsafe actions could be spoofing, component failure, electrical requirements not met or communication failure. Considering these actions and the reasons behind them, the causal factors can be identified.

3.3.2 Rationale table

The analysis uses a supporting table for the causal factor entries. It verifies the table entries and explains the thought process behind the causal factors. It can serve as a reference table for refined safety constraints and technical safety requirement tables.

Rationale for CF1.1a (Causal Factors). If the OPP (Object Predicted Path) is calculated incorrectly, there is the potential for the actual object path to be closer to the AV path than calculated. In this case, the controller will not send the BFC command, even though the autonomous vehicle's predicted path has reached the minimum safe distance from the object's predicted path. An image processing performance fault could prevent the correct calculation required for the identification of an object that is within the MSD of AV.

It was recognized and accepted that some rationale repeated itself. When this occurred, we reviewed the causal factor table for correctness and appropriateness, and if it still provided a distinctly different CF (Causal Factor), then the repeated rationale conditions were accepted. The repeated nature is suitable for automation and desirable, as long as it is applied to each unique and new CF.

3.3.3 Refined safety constraints and Technical safety requirements

After the identification of reasons behind the UCAs, the constraints on the system were redefined to eliminate or avoid the causes behind the UCAs. These new safety constraints created from the causal factors contained the rationale tables.

Technical safety requirements: This step is responsible for the implementation of refined safety constraints on the system. These represent the technical requirements for a safe system. We used these TSRs to make the gap analysis for the already existing architecture and modified the design of the system.

4 Results: Lessons learned

This section discusses the contribution of this work to making an adhoc STPA more systematic. During the analysis we learned lessons, which will be useful in structuring future analysis systematically. The lessons are summarised below:

▶ **Lesson Learned 1.** We realized that certain factors could act as a basis for the analysis development which could have an impact on the definition of the safety fundamentals. The first priority is to define boundaries which are defined as the assumptions for the analysis.

▶ **Lesson Learned 2.** We realized that the identification of accidents and hazards lacks a systematic approach. SOTIF (Safety Of The Intended Function) details from current PAS (Public Available Specification) can be useful for better structuring. We tried to make the identification of accidents and hazards systematic by considering the various scenarios in a symmetric way. The purpose of such a systematic approach is to get rid of the current brainstorming process and in its place, to establish a concrete, automatic method of scenario identification for the analysis.

▶ **Lesson Learned 3.** From our analysis we realized that the control diagram must represent the basic blocks with generic functionalities and terms. The control diagram is essential and must represent a complete overview of the function under consideration. During the analysis, the control loop serves a reference block and the representation of the control structure keeps the analysis streamlined.

▶ **Lesson Learned 4.** We have created one single report considering safety and security hazards that threaten safety. Because the safety and security issues are often interlinked, one such report, addressing both problems, is an efficient way to analyze them.

▶ **Lesson Learned 5.** The novelty of our current work is the systematic analysis of causal factors. The approach presented in Table 1 avoids unnecessary mental exercise. Here we predefined certain actions and the possible reasons for those actions. By correlating actions and reasons, using permutation and combination, the causal factors are devised. Since one of our motives is to automate this process using this constructive approach, we can automate the causal factor generation as well.

▶ **Lesson Learned 6.** Making a rationale table for each causal factor table is undoubtedly useful as it lists the logic behind the causal factors and serves as a reference for further steps. The cause-effect relationship of the unsafe actions is exploited in the rationale table. The use of rationale tables helps to identify flaws in the original causal factors and thus works as a checkpoint for those factors.

▶ **Lesson Learned 7.** While using this analysis for finding the gaps in the existing architecture we realized that any architecture could make use of it. We performed the analysis independently of the current design and later compared the technical safety requirements with the existing design. By using a generic rather than the specific approach we found that more extensive applications are possible. The analysis can be used for evaluating any existing AEB system. The gaps provided us the with the list of changes that the current architecture might incorporate in order to be safer and more secure.

▶ **Lesson Learned 8.** Another important lesson learned is about the residual risk inherent in any system. Residual risk refers to some risks which are present but acceptable, in our system. The assumptions made in the analysis are part of the residual risk. The integration of the outcome of this analysis with ISO standard is also an area where we should consider the presence of residual risk which is an integral part of the safety analysis and should be taken into account while doing the analysis.

5 Related work

STPA proved to be a more powerful and useful technique for evaluating safety-critical systems in the automotive domain by identifying the potential accident scenarios that include the entire accident process, including design errors, software flaws, component interaction accidents and human decision-making errors contributing to accidents [1].

Both ISO 26262 and STPA are based on a systems engineering framework in which a system is considered to be more than merely the sum of its parts [5]. The development and top-down analysis are common to both. While ISO 26262 [7] emphasizes the importance of considering the context of a system in achieving safety (including the role of safety management and safety culture), there seems to be no consensus on whether ISO 26262 considers the context to be a part of the hazard analysis of an item. On the other hand, STPA includes all relevant aspects of the system's environment, including the driver.

6 Conclusions

In the safety analysis, the STPA process has been used to generate system design requirements for an Automatic Emergency Brake (AEB) using a top-down analysis approach to system safety. The STPA analysis provides an improved structured approach for scenario analysis,

concentrating on safety and security. We learned lessons while applying the STPA process. The STPA has benefits, but needs to be integrated with the ISO to produce more efficient results. Doing Functional safety analysis and cyber security analysis in parallel is efficient and effective, but tool support is required. STPA is a structured and systematic approach that reduces mental exercise.

7 Future work

The next step can be a comparative study, comparing the analysis with standard ISO. Further, the analysis can potentially be expanded beyond the AEB module to cover the complete functionality of AVs.

References

1 Asim Abdulkhaleq and Stefan Wagner. Experiences with applying STPA to software-intensive systems in the automotive domain. *Stuttgart*, 2013.

2 Asim Abdulkhaleq, Stefan Wagner, Daniel Lammering, Hagen Boehmert, and Pierre Blueher. Using STPA in Compliance with ISO 26262 for Developing a Safe Architecture for Fully Automated Vehicles. *arXiv preprint*, 2017. `arXiv:1703.03657`.

3 N Leveson. An STPA Primer, Version 1. *Massachusetts Institute of Technology*, pages 22–65, 2013.

4 Nancy Leveson. *Engineering a safer world: Systems thinking applied to safety*. MIT press, 2011.

5 Archana Mallya, Vera Pantelic, Morayo Adedjouma, Mark Lawford, and Alan Wassyng. Using STPA in an ISO 26262 Compliant Process. In *International Conference on Computer Safety, Reliability, and Security*, pages 117–129. Springer, 2016.

6 Shefali Sharma Adan Flores Chris Hobbs Jeff Stafford and Sebastian Fischmeister. Functional Safety and Cybersecurity Assessment of L4 Autonomous Emergency Braking System. *University of Waterloo*, 2018.

7 Standard. ISO 26262 Road vehicles–Functional Safety. *ISO*, 2011.

8 John Thomas. Systems Theoretic Process Analysis (STPA) Tutorial, 2013.

9 John P Thomas IV. *Extending and automating a systems-theoretic hazard analysis for requirements generation and analysis*. PhD thesis, Massachusetts Institute of Technology, 2013.

10 W Young. STPA-SEC for cyber security mission assurance. *Eng Syst. Div. Syst. Eng. Res. Lab*, 2014.