CONSIDERING SAFETY AND SECURITY IN AV FUNCTIONS

by

Shefali Sharma

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2019

Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Autonomous vehicles (AVs) are coming to our streets. Due to the presence of highly complex software systems in AVs, a new hazard analysis technique is needed to meet stringent safety standards. Also, safety and security are inter-dependent and inter-related aspects of AV. They are focused on shielding the vehicles from deliberate attacks (security issue) as well as accidental failures (safety concern), that might lead to loss of lives and injuries to the occupants. So, the current research work has two key components: functional safety and cybersecurity of the autonomous systems.

For the safety analysis, we have applied System Theoretic Process Analysis (STPA), which is built on Systems Theoretic Accident Modeling and Processes (STAMP). STAMP is a powerful tool that can identify, define, analyze, and mitigate hazards from the earliest conceptual stage of development 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 thesis describes the STPA process used to generate system design requirements for an Autonomous Emergency Braking (AEB) system using a top-down analysis approach for the system safety. The research 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.

Controlling a physical process is associated with dependability requirements in a cyber-physical system (CPS). Cyberattacks can lead to the dependability requirements not being in the acceptable range. Thus, monitoring of the cyber-physical system becomes inevitable for the detection of the deviations in the system from normal operation. One of the main issues is understanding the rationale behind these variations in a reliable manner. Understanding the reason for the variation is crucial in the execution of accurate and time-based control resolution, for mitigating the cyberattacks as well as other reasons of reduced dependability. Currently, we are using evidential networks to solve the reliability issue. In the present work, we are presenting a cyber-physical system analysis where the evidential networks are used for the detection of attacks.

The results obtained from the STPA analysis, which provides the technical safety requirements, can be combined with the EN analysis, which can be used efficiently to detect the quality of the used sensor to justify whether the CPS is suitable for the safe and secure design.

Acknowledgements

I would like to thank my supervisor Prof. Sebastian Fischmeister for support and giving me an opportunity to work on such a unique system. Your unwavering vision and enthusiasm inspired me to pull through when I felt lost.

Dedication

To my family, whom I cannot express enough gratitude for supporting me, no matter what decision I made.

Contents

List of Figures **x**

List of Tables **xi**

1 Introduction **1**
1.1 Functional safety 1
1.2 Cybersecurity 2
1.3 Motivation 3
1.4 Contributions 4
1.5 Organization of the thesis 4

I **5**

2 Introduction to STAMP and STPA **6**
2.1 STAMP 6
2.2 STPA 7

3 FuSa of AEB **9**
3.1 Background: How the analysis started 10
3.2 The AEB subsystem 11

4 Methodology **13**
4.1 Scope 15
4.1.1 Assumptions 15
4.1.2 Accidents 16
4.1.3 **System level hazards** 18
4.1.4 High-level safety constraints 19
4.2 STPA Step 1 19

4.2.1 Safety control structure 19
4.2.2 Unsafe control actions 21
4.2.3 **Safety constraints** 22
4.3 STPA Step 2 22
4.3.1 **Causal factors and causal accident scenarios** . . . 22
4.3.2 Rationale table 25
4.3.3 Refined safety constraints and technical safety requirements 25

5 Results- Lessons learned 27
5.1 Lessons learned by applying STPA 27
5.2 Future scope 29

II 30

6 Importance of cybersecurity in cyber-physical systems 31
6.1 Evidence fusion for state inference 32
6.1.1 Modeling of relationship 32
6.2 Methodology for inference of current state 33
6.2.1 Dempster-Shafer theory 34
6.2.2 Belief and plausibility 34
6.2.3 Discernment frame 35
6.2.4 Mass function 36
6.2.5 Dempster's combination rule 37
6.3 Evidential Networks (EN) 38
6.3.1 Operations in the EN 39
6.3.2 Decision making in EN 40

7 Cybersecurity attack analysis using EN 41
7.1 High-level states 41
7.1.1 **Normal:** 41
7.1.2 **Error in controller command:** 42
7.1.3 **Controller malicious:** 42
7.1.4 **Manipulated communication:** 43
7.2 Threat scenarios 43
7.3 Description of the attack scenarios 44
7.4 Analysis using EN 48

7.5 Evaluation 52

8 Results 55

9 Integration of safety and security 57

Bibliography 59

Appendix 62

List of Figures

3.1 Threshold distances for the braking system 11

4.1 STPA methodology 14
4.2 Control loop structure 20
4.3 Control loop structure 23

7.1 Block Diagram of the treadmill system 42
7.2 Architectural diagram of the treadmill 50

List of Tables

4.1 Structural approach for causal factor identification 24

6.1 Linguistic scales for mapping design. [14] 33

7.1 Threat scenarios 45
7.2 Erroneous scenarios 46
7.3 TPR and FPR description [14] [26] 51
7.4 Evidential network tuple description 51
7.5 Variables of the node status 52
7.6 Relation implication rules [14] for mapping the sensors and the mass function Mass Function Relation implication rules . . 53
7.7 Different configurations for sensor reliability [14] (m3, m9, m14 represent the reliability of TM, HIDS, and NIDS respectively) 53
7.8 Table showing the decision probabilities 54

8.1 Table for TPR for the combination of HIDS and TM sensors . 55
8.2 Table for FPR for the combination of HIDS and TM sensors . 56

9.1 For GPS sensor 62
9.2 TPR and FPR for TM [26] 62

Chapter 1

Introduction

We are living in a world where many decisions are made for us by some form of software. At the same time, the criticality of said decisions is increasing. Autonomous driving is one example. A number of sensors are used in Autonomous vehicles for the perception of their surroundings, such as RADAR, Lidar, GPS, Camera,etc.. The presence of the Advanced control systems helps in getting the sensory information for navigating the appropriate path by avoiding the obstacles and follow relevant signage. Nowadays, security is also becoming relevant due to the increased interconnectivity because earlier physical access to the vehicle was needed to make a security breach. But now the vehicle can be remotely accessed and attacks can be launched from anywhere across the globe and simultaneously attack number of vehicles at the same time causing loss of human life. Thus, safety and security become crucial aspects as human life is involved in these cases.

1.1 Functional safety

The core of the overall safety of the system is functional safety, which relies on the automatic protection functioning accurately in response to the information obtained from the data or malfunction in an anticipated way (fail-safe). The automotive system shall be modeled to precisely control apparent hardware malfunctions, software failures, human errors, and operational/environmental stress.

For achieving automotive functional safety, every specified safety function shall be carried out properly, and the performance level required of specific safety function shall meet. A step by step manner described as below to accomplish the safety function compliance: [12]:

1. Identify the expected safety functions: This implies that the risks associated as well as safety functions must be identified.

2. Ensure that the design intent is fulfilled by the safety function, including under requirements of inaccurate operator data and failure styles.

3. Assess the risk-reduction demanded by the safety function: This shall comprise of automotive safety integrity level (ASIL) assessment.

4. Conduct functional safety inspections for monitoring and evaluating the process. This assessment shall indicate that suitable safety lifecycle management procedures were employed consistently and entirely in the appropriate lifecycle stages of the produce.

5. Verify that the system adheres to the designated ASIL by defining the possibility of hazardous failure, verifying minimum redundancy levels, as well as examining the precise abilities of the AV.

The safety of the system cannot be concluded without examining the environment and the other systems around with which the system communicates. Functional safety is substantially end-to-end in reach. In recent times, usually, the software intensively commands and controls the safety-critical functionalities. Thus, the functionality and accurate performance of the software are an indispensable part of the functional safety engineering effort, which at the system level guarantees a tolerable risk.

1.2 Cybersecurity

Cybersecurity refers to preserving programs, networks, and systems from digital attacks. The cyber-attacks are typically focused on getting access to the vital information and then modifying, or ruining that information, extorting money from users, or disrupting regular business means. Implementation of efficient cybersecurity measures is challenging these days as the number of devices are more than humans, and attackers are getting more ingenious [13].

Coordinated cyber attacks in cyber-physical systems (CPS) might cause cascading failures over large areas of the critical systems operations and are credible threats. Recently, privacy and security issues of CPSs are becoming critical and urgent. Due to the intimate interplay amongst cyber and physical spaces in CPS, the effect of cyber attacks is no longer confined to only the cyberspace but will be passed on to the physical systems as well. Due to the level of interoperability and scalability which CPSs maintain, attacks and other misfortunes on the CPS will most likely lead to increasing cascading failures and power outages. CPS networks and their devices have more complex assumptions and objectives on what needs to be protected in comparison to the conventional IT in the regular cyber domain.

1.3 Motivation

Nowadays, vehicles are getting more and more reliant on electronics, and manufacturers are increasingly turning to innovations in electronics and software to give them a competitive edge. A modernized luxury vehicle might possess up to 100 distinct embedded processors running over 100 million lines of code. With such a huge involvement of software, it is practically improbable to get it all right. It has been estimated that 60-70 percent of vehicle recalls involving software. [30]

The risk posed by the defects in the software, as well as attacks in automotive systems, is intense. This is due to the fact that the software is responsible for controlling the safety-critical aspects of the vehicle. Therefore, the development of safety-critical automotive software requires a precise approach and strict standards applicable worldwide. [30] The challenges presented above highlight an opportunity for a platform for addressing the concerns associated to the safety and security of the autonomous vehicles.

The proposed framework would ideally represent real-world scenarios, connecting the evidential network to practical applications. In this regard, autonomous driving seems to be a suitable choice; a wrong decision by the software would result in an accident that could claim people's lives. So, these concerns motivated for carrying out the safety and security analysis and approach to contribute to enhancing the efficiency of the current designs.

1.4 Contributions

In this thesis, we describe a functional safety approach to assess the suitability of the functionality of L4 AV. Work has been carried out to improve the ad-hoc approach to make the procedure more systematic. A cyber-physical system that can be used to help validate anomaly detection using Evidential network approach.

1.5 Organization of the thesis

The thesis is divided into two parts I and II: Part I discusses the work on the safety analysis of autonomous vehicle, whereas Part B discusses the security analysis of a cyber-physical system. Chapter 2 provides an insight into the STPA and STAMP methodology. Chapter 3 discusses the guiding requirements and description of the design of the system. Chapter 4 provides a brief design overview of the leading platform, then delve into details that impacted the system's adherence to the requirements. Chapter 5 discuss the results obtained from the analysis and lessons learned for future work. Then, comes the Part II, Chapter 6 discuss the details of Cyber-physical systems and some overview of the Dempster-Shafer theory and the Evidential networks. Chapter 7 discusses the architecture and the design of the system to be analyzed for the attacks and validates the system through a case study by using the EN to detect anomalies and evaluates the performance of the system. Chapter 8 discusses the results and future work that can improve the approach. Lastly, Chapter 9 integrates the safety and security aspects.

Part I

Chapter 2

Introduction to STAMP and STPA

Nowadays, ADAS technology is facilitating the AV functionality. The autonomous vehicle's complex system architecture and the usage of complex SoC along with the rapid rate of adoption, it is imperative for Tier-1 and semiconductor suppliers to be persistent in their collaborative endeavor to design for functional safety and the mitigation of cybersecurity threats affecting 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 share similar objectives as other techniques of hazard analysis. It is responsible for recognization of scenarios that could lead to hazardous situations and thus shall be removed or managed in the initial stage. STPA is modeled to address increasingly prevalent component interaction accidents, along with component failure accidents. The accidents can be an outcome of design imperfections or unsafe interactions amongst non-failing (operational) components [3]. Also, the causes identified by other techniques are subsets of the ones recognized using STPA [4].

2.1 STAMP

STAMP (System-Theoretic Accident Model and Processes) is built on systems theory. It is also known as an accident causality model, which displays the theoretical framework for STPA. It extends the conventional design of causality ahead of a series of directly associated failure events or component failures, in order to incorporate further complicated processes and unsafe interactions amidst system components. STAMP does not treat safety as a failure prevention problem but rather as a dynamic control problem. Some of the benefits of using STAMP are as follows [11]:

1. It operates top-down first than bottom-up and thus works on very complex systems.

2. It incorporates causal factors in accidents and different kinds of losses instead of handling them independently which includes software, humans, organizations, safety culture, etc..

3. It supports generating more robust tools, such as STPA, organizational risk analysis, accident analysis, identification and administration of leading signs of progressing risk, etc.

As STAMP employs to any emanating characteristic, STPA can be utilized for any feature of the system, and thus can be used for cybersecurity as well. STAMP shall not be confused as an analysis method; rather, as a model or set of presumptions regarding how accidents happen. It is a substitute to the chain-of-failure-events which underlies the traditional safety analysis approaches (such as fault tree analysis (FTA), failure modes and effects criticality analysis (FMECA), event tree analysis (ETA), and hazard and operability analysis (HAZOP)).

2.2 STPA

STPA (System-Theoretic Process Analysis) is a comparatively new hazard analysis method with basis on an extended model of accident causation. In addition to component failures, STPA believes that accidents can also be induced by hazardous interplays of system components, none of which may have failed. The benefits of STPA in comparison to the conventional hazard/risk analysis techniques are as follows [11]:

1. Examination of very complex systems for both intentional and unin-

tentional functionality is handled using STPA. The “Unknown unknowns” which were earlier discovered only in operations can be recognized early in the development method and can either be excluded or mitigated.

2. For the system engineering method and model-based system engineering, STPA process itself can be easily integrated with the current system design.

3. STPA can be commenced in early concept analysis and is not like the traditional hazard analysis methods to assist in recognizing safety requirements and the constraints for the system. These can then be used in the early design stages for the safety and security of the system. Thus, it helps in reducing the expensive rework associated when design imperfections are recognized late in advancement or while in operations.

4. Generally, in case of complex systems, the documentation of system functionality that is often absent but STPA provides it thoroughly.

5. STPA includes all likely causal factors in hazard analysis including the software and human drivers in the analysis, assuring that the hazard analysis comprises of all probable potential losses .

STPA has been compared with the traditional hazard analysis methods, such as FTA, FMECA, ETA, and HAZOP [11]. The comparison reveals that STPA was able to recognize all the causal scenarios encountered in case of an accident than the further traditional reviews. Also, STPA recognized numerous others, generally related to software and non-failure situations that the conventional methods were not able to recognize. Sometimes, there has been an accident where the analysts could not reason about; only STPA found the root cause for reasoning about the accident. Also, STPA turned out to be economical in considering the time and resources required than conventional methods.

Chapter 3

FuSa of AEB

The presented work 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 which will be updated in later steps of the analysis. The final control diagram 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 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 work 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 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 the safety, and security design changes part of a continuous improvement that must be at the heart of every safety culture.

3.1 Background: How the analysis started

The research 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 of the existing physical architecture and find possible security and safety issues. We needed to select vehicle functionalities that played an important role in vehicle and

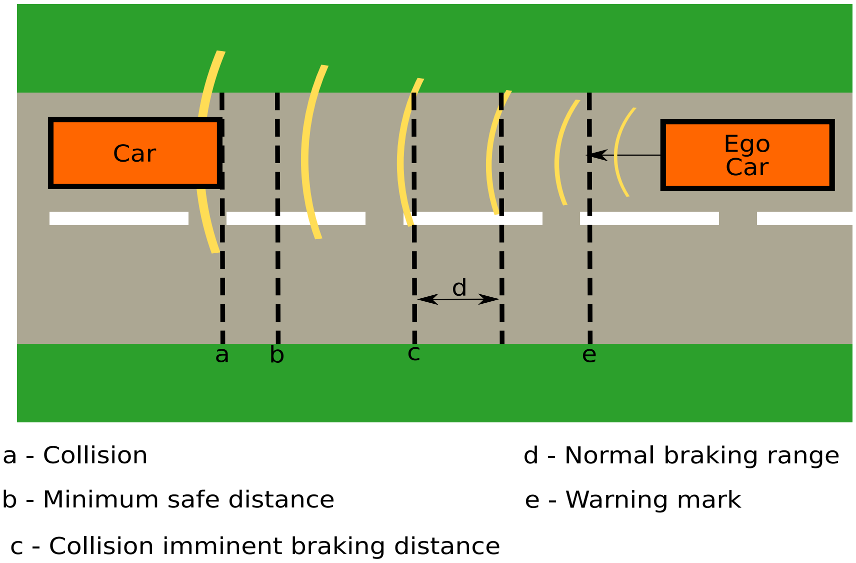

Figure 3.1: Threshold distances for the braking system

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.

3.2 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 standard 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

3.1, 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.

Chapter 4

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 4.1 presents the methodology we are using for the STPA analysis [4]:

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

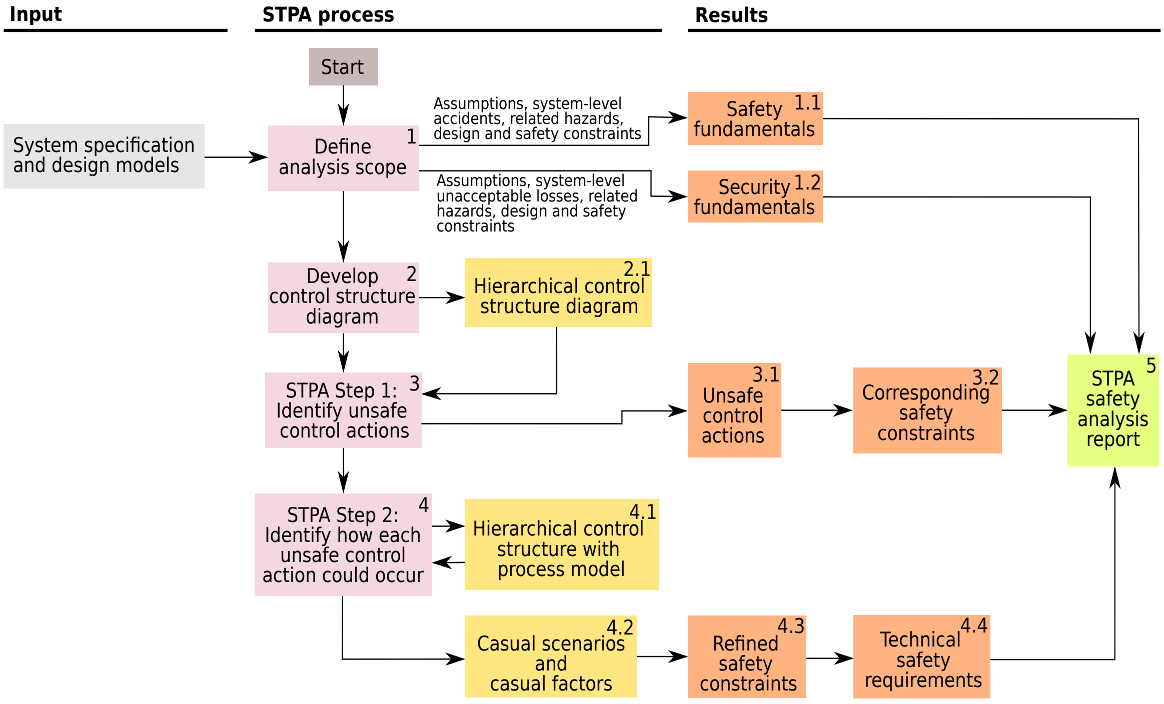
Input
STPA process
Results
Start
System specification and design models
Define analysis scope 1
Assumptions, system-level accidents, related hazards, design and safety constraints
Assumptions, system-level unacceptable losses, related hazards, design and safety constraints
Safety fundamentals 1.1
Security fundamentals 1.2
Develop control structure diagram 2
Hierarchical control structure diagram 2.1
STPA Step 1: Identify unsafe control actions 3
Unsafe control actions 3.1
Corresponding safety constraints 3.2
STPA safety analysis report 5
STPA Step 2: Identify how each unsafe control action could occur 4
Hierarchical control structure with process model 4.1
Casual scenarios and casual factors 4.2
Refined safety constraints 4.3
Technical safety requirements 4.4

Figure 4.1: STPA methodology

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 4.1. 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.

4.1 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.*"

4.1.1 Assumptions

After defining the scope, the next step is to define specific 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 2: AEB strives to minimize results attributed to multiple hazards
Assumption 3: AV tires are not damaged (e.g., tire bursting while driving)
Assumption 4: The brake system and sensors are properly cared for and maintained (e.g., clean camera lenses)
Assumption 5: Path prediction of surrounding mobile objects is available to the AEB system

Assumption 6: Brake force value is idempotent and does not keep BFC data history
Assumption 7: Passengers are wearing seat belts and are seated in a position supportive of braking maneuvers
Assumption 8: All components are working appropriately at the time of production
Assumption 9: No human interaction during L4 DDT

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, Assumption 1. The analysis considers that the tires are in perfect conditions at the time of analysis. Also, the brake system and sensors are in perfect condition as well. 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. The analysis does not consider the human intervention and there is no manufacturing defect being considered.

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.

4.1.2 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 [5]:

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 current approach is mainly the brainstorming process, and by systematically structuring the accident identification, we could consider scenarios which we might miss while brainstorming. So, we realized that for accident identification, we could start by defining subsets and then analyzing accidents as the members of the set. This structuring would give a more systematic style to the ad-hoc approach of analysis of accidents.

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

4.1.3 System level hazards

System-level hazards can be the ultimate reason for 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) provides 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).

4.1.4 High-level safety constraints

High-level safety constraints define the initial set of safety requirements for the system. These are considered as the constraints for the requirement definition.

4.2 STPA Step 1

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

4.2.1 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 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 the instruction is taking place without considering the complete internal functionality of the various components associated with it.

Controller: Here, in the 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 below:

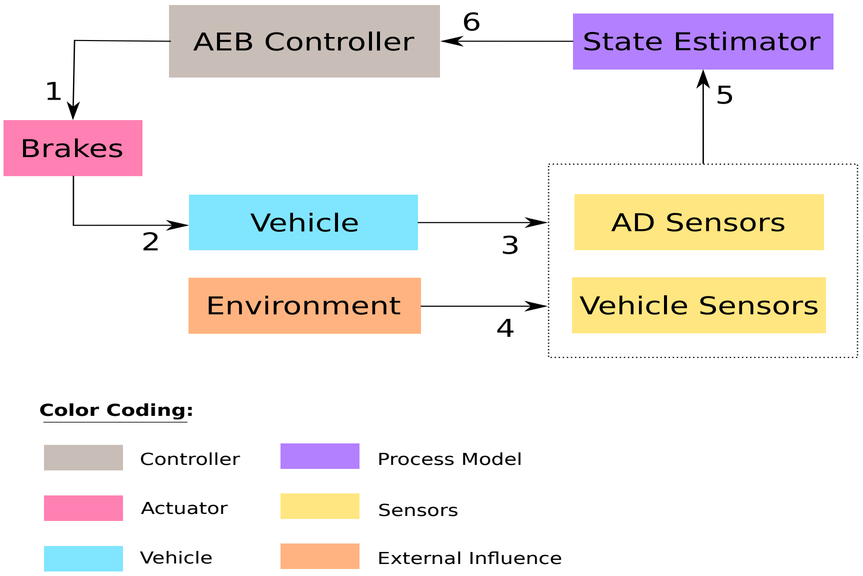

Figure 4.2: Control loop structure

After the identification of control actions, the next step is to recognize the likely causes of unsafe control.

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 amount of description to be present in the control loop structure. 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.

4.2.2 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 define the reasons of these unsafe control actions. The UCAs are defined using the control actions that can cause accidents. So, this analysis is considering two control actions for the analysis using 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 recognize the likely reasons for unsafe control.

The following are the reason, which could lead to the controller to lead to unsafe control [8]:

1. A control action is not given when needed for safety.

2. An unsafe control action is provided.

3. A probably safe control action is provided, but it is either provided too early or too late (at the incorrect time), and the order is incorrect.

4. A control action needed for safety is finished too early or applied for a prolonged time.

We considered these four categories as a basis for classification of the control table entries. The unsafe control actions considered in the analysis 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.

4.2.3 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.

4.3 STPA Step 2

This section identifies the reasons behind the unsafe control actions.

4.3.1 Causal factors and causal accident scenarios

After the recognizing the unsafe control actions, we followed STPA Step 2 (Figure 4.2) to identify the probable reasons of unsafe control actions, to understand the reason of their presence and how to prevent their occurrence [2].

However, accidents can still take place despite the absence of unsafe control actions. For example, accurate and reliable control actions are presented, but not accomplished by other components in the design. The identification of the causal factors can identify a breach of safety constraints notwithstanding safe control actions; this is important.

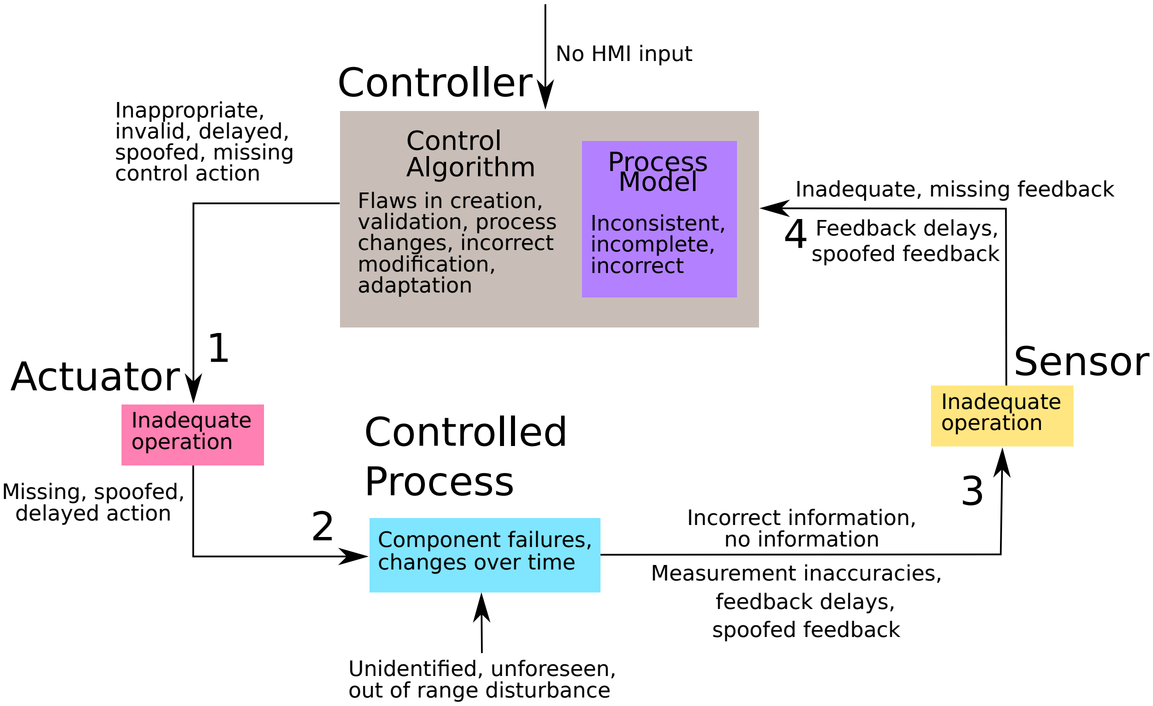
No HMI input
Controller
Inappropriate, invalid, delayed, spoofed, missing control action
Control Algorithm
Flaws in creation, validation, process changes, incorrect modification, adaptation
Process Model
Inconsistent, incomplete, incorrect
Inadequate, missing feedback
4
Feedback delays, spoofed feedback
1
Actuator
Inadequate operation
Sensor
Inadequate operation
Controlled Process
Missing, spoofed, delayed action
2
Component failures, changes over time
3
Incorrect information, no information
Measurement inaccuracies, feedback delays, spoofed feedback
Unidentified, unforeseen, out of range disturbance

Figure 4.3: Control loop structure

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.

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	

Table 4.1: Structural approach for causal factor identification

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 enough 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 4.1 provides a systematic and structured approach to analyzing 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.

For example one of the causal factor considered in the current analysis is:

CF1.4 [6] states that OPP computation fails due to AV speed sensor information is missing due to:

CF1.4a: due to spoofing
CF1.4b: due to component failure
CF1.4c: due to sensors electrical requirements not being met
CF1.4d: due to communication failure

Considering the actions and the reasons behind them as presented in the table, the causal factors can be identified systematically. The causal factor analysis is a feedback-based approach. The loop in the figure is iterated after one complete circle of considering the causal factors is applied. It is rechecked to see if there persist some scenarios that could lead the system to an unsafe state. This process is repeated until we get the technical safety requirements, which could lead the system to a safe state.

4.3.2 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.

4.3.3 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. One of the TSR considered in the analysis is [5]:

TSR1.1b: Sensor interface, as defined by the AEB controller architecture, shall be FOP (Fail-operational) and compliant to ASIL D

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.

Chapter 5

Results- Lessons learned

This chapter discusses the contribution of this work to making an ad-hoc STPA more systematic. 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.

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.

5.1 Lessons learned by applying STPA

During the analysis, we learned lessons, which will be useful in structuring future analysis systematically [6]. The lessons are summarized below:

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 priority is to define boundaries which are defined as the assumptions for the analysis.

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.

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.

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.

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.

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 between 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.

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 with the list of changes that the current architecture might incorporate in order to be safer and more secure.

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 considered while doing the analysis.

5.2 Future scope

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.

Part II

Chapter 6

Importance of cybersecurity in cyber-physical systems

For a safe and secure system design of real-time advanced engineering systems, high reliability is of paramount importance. Cyber-attacks affect the cyber-physical systems (CPS), as they are dependent on information and communication technology (ICT), and thus lead to improper operation of the system or device [14]. One of the critical threats in a system is the undetected and unauthorized sensor measurement manipulation. As the outcome of the system state is dependent on the state estimation so that a successful attack might cause incorrect or unintended decisions by the machine or the operator [15]. An attacker could introduce wicked control instructions that can cause unforeseen or improper behavior in the system, which is not suitable for the desired operation. For maintaining the control decision-making process, automatic measurement of data needs to be supported for providing state awareness. The various system states such as erroneous, malicious, and standard system states shall be considered of equal importance as each of these will affect the behavior of the complete system. The evidential networks are projected as a resolution for understanding the system states correctly. An evidential network (EN) [16] is "*a graph structure that encodes knowledge about variables in a system and the relationship between these variables.*" So, as per Dempster-Shafer Theory, the information obtained is encoded in belief structures [17]. Here in the current work, the proposed reasoning unit provides state awareness which answers about the causality of the system. With the collaboration of the information provided by different sensors, the system shall recognize the current state of the system that is caused by the underlying events.

For handling the uncertainty in the system, evidential networks are used in the present scenario as the current design of the treadmill system integrates various types of sensors. The sensors used in the current design provide varied data to operate with different level of dependability and reliability.

The evidential networks can be used for reasoning about the sensor evidence accurately from the cybersecurity domain. The problem of state inference is processed using the evidential networks in the current work. All the states: normal, malicious, and erroneous are considered to be of equal interest as all the varying states are accountable for estimating the performance of the entire system. With the provided a-priori information about the sensor's reliability, complete analysis regarding the proper combination of sensor evidence from various kinds of sensors can be done [14]. Thus, for placing a level of trust on the results obtained through the uncertainty, the evidential networks can be helpful.

6.1 Evidence fusion for state inference

There were some challenges which we came across while working on the CPS cybersecurity analysis. One of the obstacles is the correct understanding of sensory evidence in the explicit context of the system. Evidential network shows that various types of sensors require to be managed uniquely to assess the complete system state. Insufficient work has been done in analyzing the challenges of the cyber-physical systems while evidential networks and DS theory are well examined [14]. Also, the plan of relation implication rules shall be designed to lessen the error probabilities and complexity of the system design. The trustworthiness and performance based on a-priori understanding shall be handled with immense thought and in a distinctive manner depending upon the format in which the information is prepared.

6.1.1 Modeling of relationship

Relation implication rules are defined in paper [14] as the causal associations among the different sets of variables and are dependent on expert insight.

S.No.	8-Element Scale	5-Element Scale	4-Element Scale
1	Probable (99%)	Probable (99%)	Probable (99%)
2	Very Likely (85%)	Likely (74.5%)	
3	Likely (71%)		Likely (67%)
4	Possible (57%)	Possible (50%)	
5	Potentially (43%)		Possible (33%)
6	Feasible (29%)	Feasible (25.5%)	
7	Improbable (15%)		
8	Unlikely (1%)	Unlikely (1%)	Unlikely (1%)

Table 6.1: Linguistic scales for mapping design. [14]

This rule design is a two-step process: The first step involves identifying the relevant relationships, and the second step is to specify a level of belief into the represented causality. The first step can be analyzed with the safety analysis techniques such as STPA [4] and FMEA [19], which provide processes based on feedback. However, no well-established strategy is available as a second step. For the formalization of expert knowledge, usage of scale is proposed by Ou et al. [18] for supporting the human rational. However, no evidence exists to support the belief that the reduction of the probability scale to less number of discrete steps limits the performance of the evidential network. The table above presents three distinct scales that separate the mapping space into a distinct amount of even parts. For example, in the table, the “Improbable” is 15% in the 8-element scale, 33.3% in the 4-element scale, or 25.5% when mapped to the 5-element range. The end cases of the mapping (“Unlikely” and “Probable”) stays the same as Zomlot et al. [20] has already assessed their influence.

6.2 Methodology for inference of current state

The research work applies the theory of evidential networks for the identification of the causality between the system states and sensor alerts. Evidential networks are established on the Dempster-Shafer (DS) theory of evidence. The current section shall provide an insight to the Dempster-Shafer (DS) theory and Evidential networks for understanding the system state inference.

6.2.1 Dempster-Shafer theory

As discussed in [21], the Dempster–Shafer theory (DST) which is also known as the theory of belief functions or as evidence theory, gives a framework for reasoning with uncertainty, that has links to other contexts such as imprecise probability theories and possibility. The degrees of belief is based on the belief functions for one problem on the probabilities for a similar problem. The mathematical properties of probabilities might not be there in the degrees of belief themselves; the relation between two questions determine how much they differ.

The ideas behind Dempster–Shafer theory are linked depending upon whether the two concerns are independent or dependent. In case of related question, the degrees of belief are obtained from subjective probabilities, and in case of independent items of evidence, Dempster's rule [22] is used for combining such degrees of belief. The degree of trust in a proposition depends primarily upon the number of answers (to the related questions) containing the plan and the subjective probability of each solution.

6.2.2 Belief and plausibility

he framework of Shafer's work [17] allows for trust related to the hypotheses to be expressed as intervals, which is restricted by two states, belief (or support) and plausibility:

belief $\leq$ plausibility

As a primary step, the subjective probabilities (masses) are allocated to all subsets of the frame. This is usually the case where only a restricted number of sets will have non-zero mass (focal elements). The total of the masses of whole subsets constitutes the belief in a hypothesis. The degree of belief forms the lower bound on the probability that undeviatingly promotes either the presented hypothesis or a more precise one. The strength of evidence is generally marked by the Belief (usually denoted Bel) that measures in support of a proposition p. It varies from 0 (symbolizing no evidence) to 1 (expressing certainty). Plausibility can be obtained by having a total of the masses of all sets, which has a non-empty intersection with the hypothesis. Also, plausibility is one minus the total of the masses of all sets having an empty intersection with the hypothesis. Belief is an upper bound on the likelihood that the hypothesis could be accurate. Plausibility means that it “could possibly be the true state of the system” up to that value, as there is only enough evidence which contradicts the hypothesis. Plausibility (which is expressed as Pl) is determined as Pl(p) = 1 - Bel(p̃). It also varies from 0 to 1 and estimates the extent to which evidence in support of (p̃) gives place for belief in p.

6.2.3 Discernment frame

he D-S evidence theory, as described in [21] commences with establishing the frame of discernment (FD). The FD is defined as a finite nonempty exhaustive set of mutually exclusive possibilities, expressed as X here in the current explanation, which incorporates all the fundamental proposition of the problem:

Let us assume that X is the universe: the set describing all the potential states of a system under attention. The power set of X contains all the potential subsets, perceived as 2^X. 2n elements are present in the 2^X.
2^X has a power set which is the set of every subsets of X, including the void set Φ.
The calculations included in this section are from the D-S theory [21].

So, if:
X = {A,B}

Hence,

$2^X = \{\Phi, \{A\}, \{B\}, X\}$

The propositions regarding the actual state of the system can be represented by the elements of the power set, by including all and solely the states where the proposition is valid.

A belief mass is assigned to every element of the power set. The function $m : 2^X \longrightarrow [0, 1]$

is known as a basic belief assignment (BBA), in case it has the following two characteristic features.

First of all, the empty set has zero mass:

$m(\Phi) = 0$

Secondly, the total masses of the rest of the constituents of the power set is equal to 1:

$\sum_{(A \in 2^X)} m(A) = 1$

6.2.4 Mass function

The mass m(A) of A, is an assigned member of the power set. It displays the segment of all applicable and accessible evidence that substantiates the assertion that the real state relates to A but not to a distinct subset of A. The content of m(A) does not makes additional claims regarding any of the subgroups. It is concerned only to the set A, having their masses respectively.

The calculations included in this section are from the D-S theory [21].

The two bounds of a probability period i.e., the upper and the lower can be established using the mass assignments. This interval is limited by two non-additive continuous measures known as belief and plausibility and comprises the precise probability of a set of interest:

$bel(A) \leq P(A) \leq pl(A)$

The belief bel(A) for a set A is described as the aggregate of all the masses of subsets of the set of interest:

$bel(A) = \sum_{(B|B \subseteq A)} m(B)$

The plausibility pl(A) is described as the aggregate of all the portions of the sets B which intersect the set of interest A:

$Pl(A) = \sum_{(B|B \cap A \neq \phi)} m(B)$

Both the measures are linked to each other as discussed below:

$Pl(A) = 1 - bel(\bar{A})$

Conversely, given the belief measure bel(B) for a finite A, for all subsets B of A, the masses m(A) can be calculated using the inverse function presented below:
$m(A) = \sum_{(B|B \subseteq A)} (-1)^{|A-B|} bel(B)$

where $\|A - B\|$ is the difference of the cardinalities of the sets.

As is evident from the equations presented above, in case of a finite set X, for the deduction of one out of three i.e., mass, plausibility, or belief, we only need one value. Also, in case of an infinite X, the mass function cannot be well established, whereas the plausibility and belief functions can be well described.

6.2.5 Dempster's combination rule

In some specific situations, similar to our case, there arises a problem of linking two independent sets of probability mass assignments. Dempster's rule of combination works as a suitable fusion operator where we have diverse sources, and the sources represent their beliefs on the frame in expressions of belief constraints. This rule neglects all the contrary (non-shared) belief and acquires a commonly shared belief among multiple sources by a normalization factor. Cumulative fusion pays great attention to the fact that no probability mass is ignored and all probability masses from the varied sources are presented in the derived belief.

The m1 and m2 (set of masses) are used for calculating the combination (called the joint mass) as presented below:
The calculations included in this section are from the D-S theory [21].

$m_{1,2}(\phi) = 0$

$m_{1,2}(A) = (m_1 \bigoplus m_2)(A) = \left(\frac{1}{1-k}\right) \sum_{(B\cap C=A\neq\phi)} m_1(B)m_2(C)$

where
K describes the conflict amongst the two mass sets.

$K = \sum_{(B\cap C\neq\phi)} m_1(B)m_2(C)$

6.3 Evidential Networks (EN)

DS theory is used for defining an evidential network (EN), which is essentially a structure for information description and reasoning. EN is used for representing a real-world dilemma in an interlinked system of variables. EN is used in reliability engineering for dealing with the aleatory, epistemic uncertainties. The conditional dependencies amongst the variables are represented in a description range which integrates uncertainty as to belief masses [23].

An evidential network is represented as follows:

$EN = \{V, Xv, Mv, \bigoplus, \downarrow\}$
described as follows:
- $V = \{x1, ..., xn\}$ is the set of variables in the system design;
- $\bigoplus$ refers to the combination operator;
- $Mv = U\{M_D : D \subseteq V\}$ is containing the set of all mass functions in the system design;
- $\downarrow$ defines the marginalization operator.
- $Xv = \{Xv : x \in V\}$ is containing the set of frames of the variables;

$\bigoplus$ M represents the combination of all mass functions in the system model and is referred to as a joint mass function of an evidential network. It is used for inferring the information regarding higher-level system states by utilizing the information related to the variable relationships and the low-level evidence.

6.3.1 Operations in the EN

The joint mass function is computed using two evidential operations: marginalization and vacuous extension. Let us consider that there are two domains D and D', $D' \subseteq D$. X_D and XD' denote the frame of discernment for D and D' respectively [14].

Vacuous extension: It is the mass function described on the domain D', m'_D, to domain D is expressed as:

$$m_{D'}^{\uparrow D}(A) = \begin{cases} \mathrm{m}'_D(B) & \text{if } A = B * X_{D/D'} \\ 0 & \text{Otherwise} \end{cases}$$

where D \ D' represents the complement of D' in D.

Marginalization as defined on D is a projection of a mass function ,m_D, into the domain D':

$m_{D'}^{\downarrow D}(B) = \sum_{(A \subseteq B * X_{D/D'})} m_D(A)$

IF-THEN rule is usually practiced by domain specialists for providing their subjective judgments to describe the causality amongst variables, such as "if G then H." The degrees of confidence estimating uncertainty shall be appended to information rules in the case where uncertain knowledge is involved.For example "if G then H" with a specific degree of trust $p \in [\alpha, \beta], 0 \leq \alpha \leq \beta \leq 1$. A relationship is estimated to hold a minimum and maximum degree represented as α and β. A relation implication rule represents a relation between consequence and conditions. The framework of the DS theory can be used for conceptually express the relation implication rule along with uncertainty measures. Consider that DA and DB are two disjoint domains connected with frames X_{DA} and X_{DB} respectively, and A $\subseteq X_{DA}, B \subseteq X_{DB}$. A relation implication rule would be as follows:

$A \subseteq X_{DA} => B \subseteq\ X_{DB}$ with $p \in [\alpha, \beta], 0 \leq \alpha \leq \beta \leq 1$

The ballooning extension mechanism and the principle of minimum commitment can be used for representing the mass function over the product

space [25] for the above rule can be expressed as $X_{DC} = X_{DB} * X_{DA}$ on domain $D_C = D_B \cup D_A$:
The calculations included in this section are from the D-S theory [14].

$$\mathrm{m}_{DC}(C) = \begin{cases} \alpha & \text{if } C = (B * A) \cup (X_{DB} * A^C) \\ 1{-}\beta & \text{if } \quad C = (B^C * A) \cup (X_{DB} * A^C) \\ \ \beta - \alpha & \text{if } \quad C = X_{DB} * X_{DA} \end{cases}$$

where A^C depicts A's complement in X_{DA}, B^C shows B's complement in X_{DB}.

6.3.2 Decision making in EN

The belief functions cannot be used directly for making decisions because belief is induced by uncertainty, i.e., classified dispositions that manage our response [24]. Interpreted within a standardized approach, this normally commences the generation of a model to quantify beliefs that are associated directly to "rational" agent behavior within decision contexts. Hence, at the pignistic level, beliefs are quantified by probability functions. But only when a choice is actually involved, then the probability functions are employed to quantify our belief.So, pignistic probability distribution is the counterpart of the mass functions as per the classical probability theory shall be used instead. For example, assume that m_D is a mass function described on a subset of variables D beside corresponding frame X_D. The pignistic probability which is basically the pignistic transformation of m_D is established as follows [24]:

$$BetP(\theta) = \sum_{(\theta \in A \subseteq X_D)} m_D(A)/|A|$$

where $|A|$ shows the absolute number of components in A. BetP is a measure for quantifying the belief of human in terms of classical Bayesian probabilities, is the DS complement of the subjective probability.

Chapter 7

Cybersecurity attack analysis using EN

The current work is an assessment of the current state of the system based on the sensor evidence using evidential network modeling. The treadmill simulates as an infinite long highway where the vehicle moves. The system has a central workstation that controls the system. A camera is also present to track the location coordinates of the car. The interactions amongst the complete system is depicted in Figure 7.1.

7.1 High-level states

For the stability of the treadmill system, it is critical to recognize and define the current state of the system accurately as the decisions will be crucial from safety and security point of view. We identified the following system states:

7.1.1 Normal:

Complete system is working as it is expected to work under the given conditions.

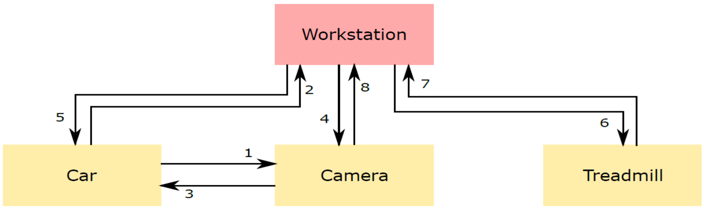

Figure 7.1: Block Diagram of the treadmill system

7.1.2 Error in controller command:

The system has error and is not working appropriately:

- The commands issued by the controller are erroneous (due to human error or arbitrary fault)
- Workstation issues erroneous commands
- Camera issues erroneous commands
- Car issues erroneous commands
- Treadmill issues erroneous commands

7.1.3 Controller malicious:

The system has malicious activities and is not working appropriately:

- The controller is compromised (malicious)
- Workstation controller is compromised

- Camera controller is compromised
- Car controller is compromised
- Treadmill controller is compromised

7.1.4 Manipulated communication:

The control commands are as expected but are manipulated in the communication network:

- Topics from Car to Camera are manipulated in the communication network
- Topics from Car to Workstation are manipulated in the communication network
- Topics from Camera to Car are manipulated in the communication network
- Topics from Camera to Workstation are manipulated in the communication network
- Topics from Workstation to Car are manipulated in the communication network
- Topics from Workstation to Treadmill are manipulated in the communication network
- Topics from Treadmill to Workstation are manipulated in the communication network
- Topics from Workstation to Camera are manipulated in the communication network

7.2 Threat scenarios

In this section, we are considering the threat scenarios that might lead to an attack to the system under considertaion. The malicious code might hijack the existing communication link in the system.

The hijacker can identify a vulnerability in the controller and thus gain full control over the treadmill controller.

1. Camera Controller
2. Workstation controller
3. Car controller
4. Treadmill controller

The system is considering the following detection mechanism present in the system for detecting the threats:

1. TM (Telemetry monitor): It monitors the position of the vehicle, the speed as well as the steering angle.
2. HIDS (Host Intrusion Detection System): It monitors the memory usage, checksum as well as CPU usage per second.
3. NIDS (Network Intrusion Detection System): It monitors the network for malicious activity

The current analysis is considering the CAPEC attack list. We simulated the attacks on the physical system, and the results obtained are compared with the evidential network analysis results.

7.3 Description of the attack scenarios

The various attacks simulated in the research are taken from the CAPEC list of attacks [29] and the definitions are verbatim in this section as per the standard CAPEC attack list.

1. Content spoofing (148):"*An adversary modifies content to make it contain something other than what the original content producer intended while keeping the apparent source of the content unchanged.*"

   Scenario: GPS attack, the location coordinates are changed. Now Car sends the location to the camera. The contents of the location topic

S.No.	Attack Types	Scenario	Attack on	Detection mechanism	Detection parameters
1.	Content Spoofing	GPS Spoofing	Path-3 (Car)	TM, NIDS	Check position value, throttle value, spikes in the dataset
2.	MITM	Delay (Camera_delay)	Path-8 (Car)	NIDS	Ideally there should be missing data
3.	Command Injection	Command_injection (Workstation issued command is tampered)	Path-5 (Car)	NIDS, TM	Check position value, throttle value
4.	Contaminate Resource	Use z parameter to send battery info, extra data in current topic	Car	HIDS, TM	Check position value
5.	Software Integrity Attack	Binary change (copy of node, make it logy)	Treadmill	HIDS	Using checksum
6.	Flooding	GPS jamming (gps_flooding)	Path-2	NIDS	Static checks
7.	Excessive Allocation	Memory allocated to attacker	Workstation	HIDS	Memory usage
8.	Traffic Injection	Encoder_fault (generated enough traffic to manipulate the information)	Treadmill	NIDS	Detector for checking the bounce
9.	Obstruction	Using dead spot on car	Camera	TM	Check position value
10	Config. /Env. manipulation	PID value controller	Car	HIDS	Checksum on config files
11.	Malicious Logic Insertion	Copy of node, edit the command value of the car in the node file (an evil version of itself)	Car	HIDS	Using checksum, memory and CPU usage

Table 7.1: Threat scenarios

S.No.	**Attack Types**	**Scenario**	**Attack on**	**Detection mechanism**	**Detection parameters**
12.	Fault Injection	Remove the decoder wheel	Treadmill	Watchdog detector (WD)	using WD
13.	E-Stop	Stop button gets pushed (operator error)	Car	E-stop detector (ED)	Using stop

Table 7.2: Erroneous scenarios

are changed by the spoofing attack, thus giving incorrect information to the camera about the current car location.

2. Man-In-The-Middle (384): "*This attack can allow the attacker to gain unauthorized privileges within the application, or conduct attacks such as phishing, deceptive strategies to spread malware, or traditional web-application attacks.*"

   Scenario: The delay will be added in the car position data. So, the position will be reported incorrectly because the system will assume that it is the current location of the car whereas the data is x sec delayed and now the car is present at some another location.

3. Command injection (248): "*An adversary looking to execute a command of their choosing, injects new items into an existing command, thus modifying interpretation away from what was intended. Commands in this context are often standalone strings that are interpreted by a downstream component and cause-specific responses.*"

   Scenario: Workstation issues command to the car, but due to command injection threat it is modified, and now it becomes a different action to perform.

4. Contaminate resource (548): "*An adversary contaminates organizational information systems (including devices and networks) by causing them to handle information of a classification/sensitivity for which they have not been authorized.*"

Scenario: The z-parameter of the location does not have any information currently, but this attack will exploit the z-parameter to send the battery information of the car. Thus, sending extra data in the current topic.

5. Software Integrity attack (184): "*An attacker initiates a series of events designed to cause a user, program, server, or device to perform actions which undermine the integrity of software code, device data structures, or device firmware, achieving the modification of the target's integrity to achieve an insecure state.*"

   Scenario: Change in the binary. The attack shall create a copy of the node and will send information with delay thus making it logy.

6. Flooding (125): "*An adversary consumes the resources of a target by rapidly engaging in many interactions with the target. This type of attack generally exposes a weakness in rate limiting or flow. When successful, this attack prevents legitimate users from accessing the service and can cause the target to crash.*"

   Scenario: There will be a GPS jamming. Car flooded camera node by sending a massive number of location data entries (E.g., If earlier the car location was sent every 100 ms now it is every 1 ms).

7. Excessive allocation (130): "*An adversary causes the target to allocate excessive resources to servicing the attackers' request, thereby reducing the resources available for legitimate services and degrading or denying services.*"

   Scenario: A large chunk of workstation memory is consumed by the attacker node thus depriving the other nodes from using the workstation.

8. Traffic Injection (594): "*An adversary injects traffic into the target's network connection. The adversary is therefore able to degrade or disrupt the connection and potentially modify the content. This is not a flooding attack, as the adversary is not focusing on exhausting resources. Instead, the adversary is crafting a specific input to affect the system in a way.*"

   Scenario: The encoder is at fault as the attack generated enough traffic to manipulate the information in the encoder.

9. Obstruction (607): "*An attacker obstructs the interactions between system components. By interrupting or disabling these interactions, an adversary can often force the system into a degraded state or even to fail.*"

   Scenario: The attack can affect the interactions amongst camera and car by the manipulating the dead spots on the camera.

10. Configuration/Environment Manipulation (176): "*An attacker manipulates files or settings external to a target application which affect the behavior of that application.*"

    Scenario: Modifying the PID value controller.

11. Malicious Logic Insertion (441): "*An adversary installs or adds malicious logic (also known as malware) into a seemingly benign component of a fielded system. This logic is often hidden from the user of the system and works behind the scenes to achieve negative impacts.*"

    Scenario: The attack makes a copy of the node, edits the command value of the node. Malicious logic is inserted at the car node.

12. Fault Injection (624): "*The adversary uses disruptive signals or events (e.g. electromagnetic pulses, laser pulses, clock glitches, etc.) to cause faulty behavior in electronic devices.*"

    Scenario: Remove the decoder wheel or slow down the vehicle (considering it to be erroneous scenario).

13. E-stop: A stop button can be pushed, causing the system to stop. It could be an operator error.

7.4 Analysis using EN

The analysis considers two parameters to categorize the various attacks on the system. Theses are identified as below:

Control Status (CS): A vulnerability in the controller is exploited by the attack and thus the attacker gains full control over the treadmill controller:

- Content Spoofing (SP)

- Command Injection (CI)
- Contaminate Resource (CR)
- Traffic Injection (TI)
- Obstruction (OB)
- Config. /Env. Manipulation (CEM)
- Malicious Logic Insertion (MLI)
- Fault injection (FI)
- E-Stop (ES)

Manipulation (MP) The malicious code might hijack the existing communication link in the system:

- Man-In-The-Middle (MITM)
- Software Integrity Attack (SIA)
- Flooding (FL)
- Excessive Allocation (EA)

The analysis requires detection mechanism sensor's TPR (True Positive Rate) and FPR (False Positive Rate) which are listed in the table 7.3 for the current work:

Evidential network tuple is:

EN $= \{V, \theta_V, M_V, \bigoplus, \downarrow\}$

Core Scenarios:

Due to the combination of various scenarios, several scenarios can occur (See Appendix). Here, we are considering only one scenario per the four core

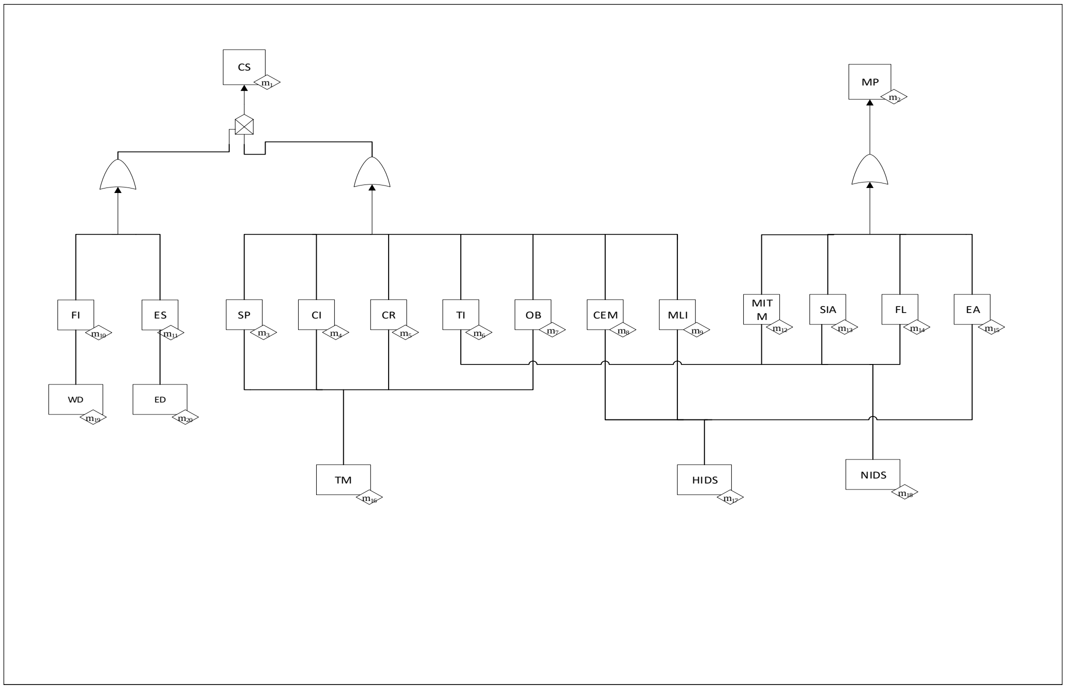

Figure 7.2: Architectural diagram of the treadmill

S.No.	**Sensor**	**TPR**	**FPR**
1	HIDS (Normal)	30%	5%
2	HIDS (Optimal)	50%	2%
3	HIDS (Low)	15%	10%
4	NIDS	70%	1%
5	TM (Normal)	22.68%	8.75%
6	TM (PFA)	93.33%	42.5%
7	TM (Mirror)	90.78%	40%

Table 7.3: TPR and FPR description [14] [26]

Scenarios	**Details**
VH	SP, CI, TI, OB, CEM, MLI, MITM, SIA, FL, EA, TM, HIDS, NIDS
VS	CS, MP
V	$V_H U V_S$
X_{VH}	$X_{SP}, X_{CI}, \ldots\ldots.$
X_{VS}	$X_{CS}, X_{MP}, \ldots\ldots.$
X_V	$X_{VH} U X_{VS}$
M_V	$m_1, m_2, \ldots\ldots. m_{18}$
Operations	$\bigoplus$, $\downarrow$, Evidential operations

Table 7.4: Evidential network tuple description

scenarios to continue the analysis.

1. Normal operation: Everything works as per the system design

2. Scenarios-1, The output shall be CS =0, MP = 0
Controller attack: The attack infects the controller directly, and now the erroneous commands are implemented by the controller itself. The attack is directly on the controller of the system.

3. Scenarios-7, The output shall be CS =1, MP = 0
Error in Controller : The controller acts in an erroneous manner and displaces the car position by 5 cm.

4. Scenarios-2, The output shall be CS =2, MP = 0
Manipulated operation: The attack infects the communication between the

S.No.	Variable name	Detail	Frame	Description
1	CS	Control Status	0,1,2	0- Normal, 1- Malicious, 2- Erroneous
2	MP	Manipulation	0,1	0- Normal, 1- Manipulate
3	TM	Sensor TM	0,1	0-Inactive, 1- Active
4	HIDS	Sensor HIDS	0,1	0-Inactive, 1- Active
5	NIDS	Sensor NIDS	0,1	0-Inactive, 1- Active
6	MLI	Malicious Logic Insertion	0,1	0-False, 1- True
7	FL	Flooding	0,1	0-False, 1- True
8	SP	Spoofing	0,1	0-False, 1- True

Table 7.5: Variables of the node status

car and workstation with malware but disguises the attack.

5. Scenarios-4, The output shall be CS =0, MP = 1
Malicious operation: The attack infects the communication between the car and workstation with an attack.

7.5 Evaluation

Different configurations of sensor reliability: We chose 3 different configurations for each sensor, so thus we obtain 27 different configurations for sensor reliability.

The mass functions are used to derive a conceptual decision based upon the sensor configurations used in the system. The masses of various sensor configurations are combined to derive the TPR and FPR probabilities for making a decision.

Mass function	**Relation Implication Rule**
m3, m16	Relation between SP and TM
SP =1, TM =1	With confidence between 0.46 and 1
SP =0, TM =0	With confidence between 0.08 and 1
m9, m17	Relation between MLI and HIDS
MLI =1, HIDS =1	With confidence between 0.72 and 1
MLI =0, HIDS =0	With confidence between 0.16 and 1
m14, m18	Relation between FL and NIDS
FL =1, NIDS =1	With confidence between 0.96 and 1
FL =0, NIDS =0	With confidence between 0.52 and 1

Table 7.6: Relation implication rules [14] for mapping the sensors and the mass function Mass Function Relation implication rules

Scenarios	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
ma3	0.46	0.46	0.46	0.37	0.37	0.37	0.388	0.388	0.388
mb3	0.08	0.08	0.08	0.788	0.788	0.788	0.73	0.73	0.73
ma9	0.72	0.2	0.92	0.72	0.2	0.92	0.72	0.2	0.92
mb9	0.16	0.02	0.32	0.16	0.02	0.32	0.16	0.02	0.32
mb14	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.96
mb14	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52

Table 7.7: Different configurations for sensor reliability [14] (m3, m9, m14 represent the reliability of TM, HIDS, and NIDS respectively)

S.No.	**Mass function HIDS**	**m ((0,0), (1,1), (0,1))**	**m ((0,0), (1,1), (1,0))**	**Mass function (1)**	**m ((0,0), (1,1), (0,1))**	**m ((0,0), (1,1), (1,0))**	**Mass function (1)**	**m ((0,0), (1,1), (0,1))**	**m ((0,0), (1,1), (1,0))**	**TPR**	**FPR**	**Decision True**	**Decision False**
1	0.72	0.0448	0.6048	0.46	0.0432	0.4232	0.96	0.0208	0.4608	0.877089	0.105074	Very likely and 1	Unlikely and 1
2	0.2	0.016	0.196	0.46	0.0432	0.4232	0.96	0.0208	0.4608	0.749948	0.078092	Likely and 1	Unlikely and 1
3	0.92	0.0256	0.6256	0.46	0.0432	0.4232	0.96	0.0208	0.4608	0.883558	0.087086	Very likely and 1	Unlikely and 1
4	0.72	0.0448	0.6048	0.37	0.49644	0.07844	0.96	0.0208	0.4608	0.803623	0.529004	Likely and 1	Potentially and 1
5	0.2	0.016	0.196	0.37	0.49644	0.07844	0.96	0.0208	0.4608	0.600488	0.514803	Possible and 1	Potentially and 1
6	0.92	0.0256	0.6256	0.37	0.49644	0.07844	0.96	0.0208	0.4608	0.813959	0.519537	Likely and 1	Potentially and 1
7	0.72	0.0448	0.6048	0.388	0.44676	0.10476	0.96	0.0208	0.4608	0.809232	0.482537	Likely and 1	Potentially and 1
8	0.2	0.016	0.196	0.388	0.44676	0.10476	0.96	0.0208	0.4608	0.611898	0.466935	Possible and 1	Potentially and 1
9	0.92	0.0256	0.6256	0.388	0.44676	0.10476	0.96	0.0208	0.4608	0.819272	0.472136	Likely and 1	Potentially and 1

Table 7.8: Table showing the decision probabilities

Chapter 8

Results

The analysis of the study shows the performance comparison that can be done based upon the sensor configuration to come up with the configuration with a trade-off between the system performance and resources required for it. Cost is an important factor while choosing the sensors for a system. If the performance between the two configurations is not significant, but there is a significant cost difference so we might choose the optimum configuration sensors. The process has been automated using java code to perform the simulations and can check a number of configurations before implementing the architecture. For example, the tables in this chapter shows the different configurations of HIDS and TM sensors and the outcome of the final TPR of the system.

Sensor type	**Bad HIDS**	**Medium HIDS**	**Good HIDS**
Bad TM	0.600	0.804	0.814
Medium TM	0.619	0.809	0.819
Good TM	0.749	0.877	0.883

Table 8.1: Table for TPR for the combination of HIDS and TM sensors

As can be clearly visualized, one poor accuracy sensor is not affecting the overall TPR of the complete system. Hence based upon other factors such as cost and resource requirement we can have a tradeoff and choose the medium performance sensor if the overall TPR required from the system is not stringent.

Similarly, for the FPR, the table below depicts the effect of different configurations of sensor on the overall FPR.

Sensor type	**Bad HIDS**	**Medium HIDS**	**Good HIDS**
Bad TM	0.515	0.529	0.519
Medium TM	0.467	0.482	0.472
Good TM	0.078	0.105	0.087

Table 8.2: Table for FPR for the combination of HIDS and TM sensors

As can be clearly visualized, one poor accuracy sensor does not have a significant impact on the overall FPR of the entire system. The present network architecture has been realized as a generic code for simulating the system design to assess the suitability as per the application requirement.

Hence, Evidential networks can be utilized to evaluate the sensor performance and the attack probabilities in a system configuration.

Future scope: This can be extended to real-time system for evaluating the attack probabilities based upon system performance.

Chapter 9

Integration of safety and security

The results obtained from STPA analysis, which provides the technical safety requirements can be combined with the EN analysis which can be used efficiently to detect the quality of the used sensor to justify whether the CPS is an ideal fit for the safe and secure design. The STPA gives the technical safety specifications which shall be satisfied for a safe and secure system. The EN provides the reasoning using the current work for determining the efficiency of the system to resist attack. So, for satisfying the security constraints on the design of AV, EN provides a framework for verifying the components used in the system design. Thus, the two works presented in the thesis as Part- A and Part-B converge together to build a reasoning-based relationship where EN supports the implementation of the STPA results. For example, one of the TSR is: TSR1.1c- Sensor interface, as defined by the AEB controller architecture, shall be secure following the recommended practices from SAE J3061. Now in order to satisfy this requirement EN analysis has been done on the sensors and based upon the results, the appropriate sensor configuration as per the requirement is chosen for the design.

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 .

For the security aspect, comparative analysis of the current approach with the standard Bayesian analysis could be next step. Also, the analysis can be potentially expanded to include the real-time systems analysis.

Bibliography

[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.

[11] Nancy Leveson, John P Thomas. STPA Handbook. MIT press,2018.

[12] Wikipidea https //en.wikipedia.org/wiki/Functional_safety

[13] https //www.cisco.com/c/en/us/products/security/what-is-cybersecurity.html

[14] Ivo Friedberg, Xin Hong, Kieran McLaughlin, Paul Smith, Paul Miller, Evidential Network Modeling for Cyber-Physical System State Inference, Security Analytics and Intelligence for Cyber Physical Systems - IEEE Access, June 2017

[15] A. Giani, E. Bitar, M. Garcia, M. McQueen, P. Khargonekar, and K. Poolla, "Smart Grid Data Integrity Attacks," Smart Grid, IEEE Transactions on, vol. 4, no. 3, pp. 1244–1253, sep 2013.

[16] P. P. Shenoy, "A valuation-based language for expert systems,"International Journal of Approximate Reasoning, vol. 3, pp.383–411, 1989.

[17] G. Shafer, A Mathematical Theory of Evidence. Princeton University Press, 1976.

[18] X. Ou, S. R. Rajagopalan, and S. Sakthivelmurugan, "An Empirical Approach to Modeling Uncertainty in Intrusion Analysis," in 2009 Annual Computer Security Applications Conference, 2009, pp. 494–503.

[19] D. H. Stamatis, Failure mode and effect analysis: FMEA from theory to execution. ASQ Quality Press, 2003.

[20] L. Zomlot, S. C. Sundaramurthy, K. Luo, X. Ou, and S. R. Rajagopalan, "Prioritizing intrusion analysis using Dempster-Shafer theory," in Proceedings of the 4th ACM workshop on Security and artificial intelligence - AISec '11. New York, New York, USA: ACM Press, Oct 2011, pp. 59–70.

[21] https //en.wikipedia.org/wiki/Dempster%E2%80%93Shafer_theory

[22] Dempster, Arthur P.; A generalization of Bayesian inference, Journal of the Royal Statistical Society, Series B, Vol. 30, pp. 205–247, 1968

[23] Jianping Yang, Hong-Zhong Huang, Yu Liu1 and Yan-Feng Li, Evidential Networks for Fault Tree Analysis with Imprecise Knowledge, Int. J. Turbo Jet-Engines, Vol. 29 (2012), pp. 111–122

[24] P. Smets, "Constructing the pignistic probability function in a context of uncertainty," in Procs. of UAI, 1990, pp. 29–40.

[25] R. Haenni and N. Lehmann, "Probabilistic argumentation systems: a new perspective on dempster-shafer theory," International Journal of Intelligent Systems, vol. 18, pp. 93–106,

[26] Stijn Van Winsen, THREAT MODELLING FOR FUTURE VEHICLES, Master of Science - Computer Science - Kerckhoffs Institute, Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, January 2017

[27] Mahmudur Rahman, Mozhgan Azimpourkivi, Umut Topkara, Bogdan Carbunar, Video Liveness for Citizen Journalism: Attacks and Defenses

[28] Reliability data for Safety Instrumented Systems, PDS handbook 2010 edition

[29] https://capec.mitre.org/

[30] https://www.testandverification.com/thought-leadership/iso26262-a-static-analysis-tools-perspective-19-october-2015/

Appendix

Te attack probabilities for GPS sensor is calculates using SAE J3061 as follows:

Sensor	S	O	P	F	C	Attack Potential (A)	Attack Probability (P)	Risk
GPS	1	3	3	0	1	7	5	R3

Table 9.1: For GPS sensor

Using the calculations from [14], the TPR and FPR for various attacks (without training) for the location detection:

Attack	**TPR**	**FPR**
PFA	93.33	42.5
Sandwich	22.68	8.75
Ipc mirror	90.78	40.0

Table 9.2: TPR and FPR for TM [26]

Alpha calculation for TM:

Sandwich attack:

PPV (TM=1) = 0.22/ (0.22+0.08) = 0.22/0.30 = 0.73
PPV (TM=0) = 0.92/ (0.92 + 0.78) = 0.92/1.70 =0.54

Calculating alpha:

α (TM =1) = PPV – (1-PPV) = 0.73-1 + 0.73 = 0.46
α (TM =0) = PPV – (1-PPV) = 0.54-1 + 0.54 = 0.08

PFA: 93.33 %, 42.5%
PPV (TM=1) = 0.93/ (0.93+0.42) = 0.93/1.35 = 0.688
PPV (TM=0) = 0.57/ (0.57 + 0.067) = 0.57/0.637 =0.894

Calculating alpha:
α (TM =1) = PPV – (1-PPV) = 0.688-1 + 0.688 = 0.37

α (TM =0) = PPV – (1-PPV) = 0.894-1 + 0.894 = 0.788

Mirror: 90.78%, 40%

PPV (TM=1) = 0.9078/ (0.9078+0.4) = 0.9078/1.3078 = 0.694
PPV (TM=0) = 0.60/ (0.922 + 0.60) = 0.6/0.6922 =0.866

Calculating alpha:

α (TM =1) = PPV – (1-PPV) = 0.694-1 + 0.694 = 0.388
α (TM =0) = PPV – (1-PPV) = 0.866-1 + 0.866 = 0.73

Alpha calculation for HIDS:

Realistic
PPV (HIDS=1) = 0.3/ (0.3+0.05) = 0.3/0.35 = 0.86
PPV (HIDS =0) = 0.95/(0.95 + 0.7) = 0.95/1.65 =0.58

Calculating alpha:

α (HIDS =1) = PPV – (1-PPV) = 0.86-1 + 0.86 = 0.72
α (HIDS =0) = PPV – (1-PPV) = 0.58-1 + 0.58 = 0.16

Low

PPV (HIDS=1) = 0.15/ (0.15+0.1) = 0.15/0.25 = 0.6
PPV (HIDS =0) = 0.90/ (0.90 + 0.85) = 0.90/1.75 =0.51

Calculating alpha:

α (HIDS =1) = PPV – (1-PPV) = 0.6-1 + 0.6 = 0.2
α (HIDS =0) = PPV – (1-PPV) = 0.51-1 + 0.51 = 0.02

High

PPV (HIDS=1) = 0.5/ (0.5+0.02) = 0.5/0.52 = 0.96
PPV (HIDS =0) = 0.98/ (0.98 + 0.5) = 0.98/1.48 =0.66

Calculating alpha:

α (HIDS =1) = PPV – (1-PPV) = 0.96-1 + 0.96 = 0.92
α (HIDS =0) = PPV – (1-PPV) = 0.66-1 + 0.66 = 0.32

Alpha calculation for NIDS:

PPV (NIDS =1) = 0.7/ (0.7+0.01) = 0.70/0.71 = 0.98
PPV (NIDS =0) = 0.99/ (0.99 + 0.3) = 0.99/1.29 =0.76

Calculating alpha:
α (NIDS =1) = PPV – (1-PPV) = 0.98-1 + 0.98 = 0.96
α (NIDS =0) = PPV – (1-PPV) = 0.76-1 + 0.76 = 0.52

Calculation of Domain knowledge (Dempster Shafer):
For HIDS: Realistic
ma9 ((1,1), (1,0), (0,0)) = 0.72
ma9 ((1,1), (1,0), (0,1), (0,0)) = 1- 0.72 = 0.28

mb9 ((0,0), (1,1), (0,1)) = 0.16
mb9 ((1,1), (1,0), (0,1), (0,0)) = 1- 0.16 = 0.84

Domain knowledge:

m9 ((0,0), (1,1)) = 0.1152
m9 ((0,0), (1,1), (0,1)) = 0.0448
m9 ((0,0), (1,1), (1,0)) = 0.6048
m9 ((0,0), (1,1), (1,0), (0,1)) = 0.2352

For HIDS: Low
ma9 ((1,1), (1,0), (0,0)) = 0.2
ma9 ((1,1), (1,0), (0,1), (0,0)) = 1- 0.2 = 0.8

mb9 ((0,0), (1,1), (0,1)) = 0.02
mb9 ((1,1), (1,0), (0,1), (0,0)) = 1- 0.02 = 0.98

Domain knowledge:

m9 ((0,0), (1,1)) = 0.004
m9 ((0,0), (1,1), (0,1)) = 0.016
m9 ((0,0), (1,1), (1,0)) = 0.196
m9 ((0,0), (1,1), (1,0), (0,1)) = 0.784

For HIDS: High
ma9 ((1,1), (1,0), (0,0)) = 0.92
ma9 ((1,1), (1,0), (0,1), (0,0)) = 1- 0.92 = 0.08

mb9 ((0,0), (1,1), (0,1)) = 0.32
mb9 ((1,1), (1,0), (0,1), (0,0)) = 1- 0.32 = 0.68

Domain knowledge:

m9 ((0,0), (1,1)) = 0.294
m9 ((0,0), (1,1), (0,1)) = 0.0256
m9 ((0,0), (1,1), (1,0)) = 0.6256
m9 ((0,0), (1,1), (1,0), (0,1)) = 0.054

For NIDS
ma14 ((1,1), (1,0), (0,0)) = 0.96
ma14 ((1,1), (1,0), (0,1), (0,0)) = 1- 0.96 = 0.04

mb14 ((0,0), (1,1), (0,1)) = 0.52
mb14 ((1,1), (1,0), (0,1), (0,0)) = 1- 0.52 = 0.48

Domain knowledge:

m14 ((0,0), (1,1)) = 0.4992
m14 ((0,0), (1,1), (0,1)) = 0.0208
m14 ((0,0), (1,1), (1,0)) = 0.4608
m14 ((0,0), (1,1), (1,0), (0,1)) = 0.0192

For TM: Sandwich

ma3 ((1,1), (1,0), (0,0)) = 0.92
ma3 ((1,1), (1,0), (0,1), (0,0)) = 1- 0.92 = 0.08

mb3 ((0,0), (1,1), (0,1)) = 0.32
mb3 ((1,1), (1,0), (0,1), (0,0)) = 1- 0.32 = 0.68

Domain knowledge:

m3 ((0,0), (1,1)) = 0.294
m3 ((0,0), (1,1), (0,1)) = 0.0256
m3 ((0,0), (1,1), (1,0)) = 0.6256
m3 ((0,0), (1,1), (1,0), (0,1)) = 0.054

For TM: PFA

ma3 ((1,1), (1,0), (0,0)) = 0.37
ma3 ((1,1), (1,0), (0,1), (0,0)) = 1- 0.37 = 0.63

mb3 ((0,0), (1,1), (0,1)) = 0.78
mb3 ((1,1), (1,0), (0,1), (0,0)) = 1- 0.78 = 0.22

Domain knowledge:

m3 ((0,0), (1,1)) = 0.2886
m3 ((0,0), (1,1), (0,1)) = 0.4914
m3 ((0,0), (1,1), (1,0)) = 0.0814
m3 ((0,0), (1,1), (1,0), (0,1)) = 0.1386

For TM: Mirror

ma3 ((1,1), (1,0), (0,0)) = 0.38
ma3 ((1,1), (1,0), (0,1), (0,0)) = 1- 0.38 = 0.62

mb3 ((0,0), (1,1), (0,1)) = 0.73
mb3 ((1,1), (1,0), (0,1), (0,0)) = 1- 0.73 = 0.27

Domain knowledge:

m3 ((0,0), (1,1)) = 0.2774
m3 ((0,0), (1,1), (0,1)) = 0.4526
m3 ((0,0), (1,1), (1,0)) = 0.1026
m3 ((0,0), (1,1), (1,0), (0,1)) = 0.1674

Failure rate for watchdog detector:
Number of lines of code * 25/1000
since it has 500 lines of code so the probability of error is 2.5% or 0.025

Failure rate for switch: [28]

$[\lambda, \lambda] = [0.034, 0.023]$

So [60, 40] = [0.566, 0.377, 0.057] (lower and upper probs)