Derivation and Validation of System Requirements for Automated Vehicles

A thesis submitted by

Emmett R. Lepp

in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

Tufts University

May 2025

Adviser: Jason Rife

Abstract

Modern day vehicles come equipped with numerous Advanced Driver-Assistance Systems (ADAS) including blind spot detection, adaptive cruise control, and collision detection. These improvements, while impactful, are insufficient in eliminating the number of vehicular accidents and fatalities experienced in the United States. In 2023, over 40,000 vehicular fatalities occurred in the United States, representing a slight decrease since the previous year [1]. Automated Vehicles (AVs) have the potential to further reduce the number of fatalities sustained everyday by motorists.

This dissertation analyzes system-level requirements for AVs using the System-Theoretic Process Analysis (STPA), with the goal of contributing to making AVs as safe as possible. Specifically, STPA is used to derive recommendations for design changes that make vehicles more resilient and that reduce the risk of critical failures. A case study, focusing on three specific functionalities of automated vehicles, validated the efficacy of the process developed in this work. Together, the SPTA results, and complementary simulation studies, serve as a successful proof of concept for a design process that could be applied at a more detailed level to specify and validate AV system requirements.

Acknowledgments

This work would not have been possible without the support of numerous people. First, I would like to thank the United States Army and the Charles Stark Draper Laboratory, for giving me the opportunity to broaden my experiences and expertise over the last two years.

I would like to thank all the members of my thesis committee for working with me on this endeavor. To Jason, thank you for constantly pushing me with hard questions, forcing me to think deeply and see the broad impact my research had the potential to deliver. I appreciate all your feedback, our weekly meetings, and all your support. To Tiffany, thank you for exposing me to the world of STPA and challenging my engineering worldview. My experience at Draper has been eye-opening and has allowed me to experience the possibility of a future career. To Pratap, thank you for serving as an amazing engineering mentor and constantly driving me to defend my judgment and become a better engineer and researcher. To my ASAR group mates thank you for constantly listening to my research ideas, pushing back on my nonsensical presentations, and challenging me to be a better researcher.

Lastly, I would like to thank my amazing partner and family for all their support over the last two years. Sammy, I cannot begin to thank you for all you do for me and for constantly pushing me to keep working towards finishing this work. These last two years have been an amazing experience and there is no one else I would rather have gone on this journey with. To my family, thank you for your unwavering support and your willingness to listen to and help reform my ideas.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Context

Vehicular driving carries inherent risk. While a driver has the ability to control their own vehicle, the environment in which they operate their vehicles is an external factor outside their control. Assuming the driver operates perfectly (which is often not the case), accidents can still occur due to environmental factors, negligent drivers, vehicular failures, and design errors. Creating a system that heightens situational awareness and responsiveness for improved operator safety, regardless of external contributors, allows for an overall safer operating environment.

1.2 Motivation

Throughout modern history a multitude of improvements have been made to improve vehicle safety: antilock braking systems, crumple zones, and seatbelts to increase the survivability for vehicle occupants. While mitigating unfavorable outcomes in the event of a crash is important, improving vehicle performance to stop crashes from occurring in the first place would drastically reduce the number of vehicle fatalities that occur each year.

In 2023, drivers across the United States were involved in a multitude of vehicle crashes resulting in 40,990 fatalities [1]. This number is staggeringly high and represents the dangerous conditions that drivers are faced with every day. With the emergence of devices such as smartphones and in-vehicle displays, drivers are plagued with a multitude of distractions which decrease their alertness and increase their likelihood of being involved in a crash. Creating a vehicle to assist the human driver serves as a potential solution to prevent these fatalities from occurring.

Significant engineering effort has focused on creating vehicles that can assist drivers, rather than aiming to replace the driver altogether. These Systems are collectively called Advanced Driver-Assistance Systems (ADAS). As it stands currently, advancements in this ADAS have had significant impact on safety; however, ADAS are still not perfect, with automation assisted vehicles sometimes still unable to prevent vehicular crashes.

Highly automated vehicles have the potential for further improving vehicle safety, but increased automation is not necessarily the only solution. A 2015 study by the University of Michigan Transportation Research Institute found that automated vehicles (Avs) were more likely to be involved in car crashes than their conventional counterparts [2].

More recently, data from the California Department of Motor Vehicles indicates that AVs were more likely than human-operated vehicles to be involved in rear-end collisions and less likely to be involved in side swipe or head on collisions [3]. While anecdotal, evidence from individual AV crashes, such as a fatal 2016 Tesla crash, indicate safety gaps exist [4]. Although increased integration and improvements of Automated Vehicles (Avs) have been accomplished since these reports were published, it is nonetheless clear that AV technology is in its early stages. More work is needed to create design processes to promote a safer driving platform for all users.

1.3 Background

1.3.1 Automated Vehicles

Modern production vehicles are equipped with a variety of technologies, capable of assisting drivers in making more informed decisions and enhancing system safety. The promise of these emerging technologies has led to a push for autonomous driving to reduce the role of humans in driving and the associated risk of making potentially incorrect choices. The Society of

Automotive Engineers (SAE) has created classifications regarding the spectrum of control between full human driving and fully automated driving. The SAE defines six levels in all (from zero to five). Levels zero to two describe vehicles in which human drivers are responsible for executing necessary control actions, whereas levels three to five describe vehicles where automated vehicle features take control of vehicle function [5].

Currently within the United States, SAE Level 2 (SL2) vehicles are the most readily available form of automated driving [6]. These vehicles push ADAS capabilities to the point where the vehicle acceleration, breaking, and steering can be fully automated under certain conditions; however, the driver must be fully alert at all times to detect anomalous operation and to take control if the ADAS functions fail. As of this writing, companies have produced vehicles with higher levels of automation, as used in Waymo's driverless taxi service; however, SAE level three (SL3) and higher vehicles are not yet available for purchase by consumers.

1.3.2 System Design

Growing complexity and reliance on software and complex critical technologies have given rise to new risks within SL2 architectures. Hence, the importance has increased for incorporating Human System Integration (HSI), System Security Engineering (SSE), System Safety Engineering (Safety), Reliability, Availability, & Maintainability (RAM), and Survivability principles during a system's concept development. Embedding these disciplines within the systems engineering lifecycle can promote flexibility, adaptability, and resilience. These areas are achieved by mitigation of undesired system behaviors which collectively control risk of unacceptable loss and unrecoverable downtime. The following factors must be Pareto optimized to promote the best chance of achieving the AV's mission objectives: situational awareness, decision making, and responsiveness/latency.

In this dissertation, the key innovation is to apply System Theoretic Process Analysis (STPA) as the mechanism to enhance the overall SL2 system design. STPA provides a foundation for problem framing, for system architecture requirements derivation, and for verification and validation. STPA additionally informs and guides each of the systems engineering processes to ensure individual components (sometimes called specialty- engineering deliverables) are adequately integrated and embedded within core systems engineering deliverables. This cohesion is valuable for compliance, traceability, and validation needs.

This document describes the application of STPA's four-step process to identify losses, hazards, and a control structure for use in the engineering design process. As an important complement to the standard STPA process, this dissertation introduces a complementary simulation tool, with the goal of testing the efficacy of risk mitigation strategies that emerge from applying STPA to the design of vehicles with SL2 capabilities.

1.4 Contribution

This thesis provides two main contributions:

(i) Application of System Theoretic Process Analysis (STPA) to the design of SL2 vehicles

(ii) Development of a simulation strategy to test risk-mitigation strategies that emerge from STPA.

This work also represents a first step toward strengthening STPA to enhance its capabilities to support rapid and iterative redesign for increasingly complex automation.

The first contribution involves applying STPA to a new type of system. STPA has not previously been applied to SL2 vehicles. It is important to note, however that STPA has been applied to several other automation topics related to roadway vehicles [7-9] and other types of automated vehicle [10, 11]. STPA is a relatively new approach to system safety, so opportunities

exist to uncover latent failure modes in SL2 driving, and possibly, to promote safety in continuing iterations of SL2 system design.

The second contribution focuses on extending STPA by introducing a simulation process that allows testing of design concepts generated by STPA with the simulation operating at a level of resolution consistent with the needs of early-stage design. This work shows that using a well-scoped simulation in conjunction with the STPA process can identify use cases where a design improvement can provide benefit. I call my simulation method Local Investigation of Mitigation Efficacy (LIME).

1.6 Thesis Structure

This thesis is structured to support the two contributions listed above, with a chapter dedicated to each.

Chapter 2 discusses the STPA methodology, providing the background necessary for an application to SL2 vehicles. After describing the method and its application, the chapter identifies three key scenarios spanning a range of human/ automation interactions. In each of these scenarios, STPA allows the identification of risk mitigation strategy. Although the risk-mitigation strategies identified here are not necessarily new to the larger field of SL2 design, these applications show the potential for a future, more thorough exploration to identify as yet unrecognized failure modes.

Chapter 3 takes the mitigations generated through the STPA process in Chapter 2 and introduces a focused simulation process that rapidly rests the efficacy of these design changes on in a focused region of the larger design space. By construction, this analysis is not meant to be exhaustive; the idea is that the early-stage design process is inherently iterative, so a limited

validation (such as that introduced in Chapter 3) is appropriate at this stage, with a more thorough validation left to later stages in the design process.

Chapter 2: Systems Theoretic Process Analysis for Development of System Requirements

2.1 Introduction

In recent years, automated vehicles have been touted as the future of automotive and autonomous engineering. However, even innovative technology does not solve all the problems a driver may face during a routine driving event. Examples can be seen involving these automated vehicles preforming incorrectly and resulting in both monetary and physical costs for the motorist [12, 13].

System Theoretic Process Analysis (STPA) is a hazard analysis technique which can be used early in the design process to derive necessary system requirements [14]. STPA provides a different approach compared to other forms of fault analysis such as fault trees or failure modes and effects analysis (FMEA) by analyzing complex systems and the interactions between their components, early enough in the design process. Additionally, STPA accounts for hardware and software, design errors, failures, threats, and human interactions between systems when compiling the lists of possible hazards a systems may encounter [14]. Other research has been done to compare the results of FMEA and STPA analyses, indicating large overlaps between the two methods in finding similar faults modes [7, 15].

Automated vehicles have a wide range of functionalities and practical applications in protecting drivers from potential harm. This analysis focuses on Society of Automotive Engineers (SAE) Level 2 (SL2) vehicles which provide driver support features to assist in the safe maneuvering of the vehicle but forces the driver to stay involved in the operation of the vehicle. Actions such as lane centering, adaptive cruise control, and automated breaking are utilized by SL2 vehicles to accomplish these goals [5]. As more work is done to advance the

development of these systems, utilizing an alternative approach such as STPA can provide insight into how SL2 vehicles can evolve to produce a safer system.

This contribution of this chapter is the application of the STPA model to SL2 automated vehicle to derive system losses, hazards, and control action hazardous contexts that result in loss scenarios, if not mitigated against. Examining a select few of these scenarios, mitigations are suggested to reduce the overall risk for SL2 vehicle operation. The application of this form of hazard analysis of driver assisted automated vehicles has not been investigated thoroughly and has the potential to help design vehicles to perform safely and protect the vehicle's driver. This process may expose areas of the design, which may previously be overlooked, directing suggested improvements and allowing novel design changes to remedy flaws in the current layout.

2.2 Background

STPA is an approach for hazard and accident analysis that identifies undesired system behaviors that emerge from interactions between system components. STPA is built upon System Theory concepts, and is a process to protect against emergent interactions, both inadvertent and intentional [14, 16]. Focusing on these interactions is where STPA promotes the discovery of interconnections that may have otherwise been excluded by other forms of analysis, which generally occur later in the lifecycle. STPA provides the most advantage when utilized early in the design process to create and test safety requirements [14, 17]. However, there are some drawbacks to the STPA process including a high upfront cost of learning how to apply the methodology as well as a lack of traditionally assigned risk probabilities seen in other forms of hazard analysis [18]. While it is still an emerging method of hazard analysis, STPA's marked advantages help distinguish it from other forms of fault detections due to its iterative nature and

ability to provide traceability of design recommendations to direct requirements put forward by the customer.

The subject of automated vehicles has been studied using the STPA process in other examples found in the literature [7, 19-22]. These analyses focus their attention on subsystems of automated vehicles: software faults, electronic control systems, and lane keeping systems or fully autonomous driving systems [9, 19, 23, 24]. These analyses, through their application of the STPA methodology, provide a detailed breakdown of where improvements to these subsystems may be made. This paper attempts to build on previous work applying STPA to automated vehicles, but with a particular focus on the analysis of SL2 vehicle systems with driver involvement. Prior literature contains no examples of applications of STPA to SL2 vehicle systems.

Other methods of hazard analysis, such as Failure Mode and Effects Analysis (FMEA) or Fault Tree Analysis (FTA), are also used to determine system safety and reliability. FMEA is a process for detecting faults in systems to prevent future accidents from occurring [25]. This form of analysis involves evaluating the risk of failure that may occur in a product and suggesting potential changes due to the severity, occurrence, and ability of the failure to be detected. Fault Tree Analysis is a process that involves breaking down a physical system into a series of logical gates, which helps guide the process to discover potential modes of failure [26]. FTA takes a more quantitative approach to evaluate the fault tree through the use of Monte Carlo simulations or deterministic methods, ultimately arriving at the end goal of hazard analysis. Compared to these two methods, STPA can be utilized earlier in the design process and does not require the application of quantitative measures of risk tolerances to derive system design. These advantages

allow for more iteration and innovation throughout the design process in order to deliver on a customer's objectives and goals, which cannot be achieved without System Safety.

STPA has been applied to a variety of systems ranging from aircraft to underwater vehicles. These other studies highlight the efficacy of STPA to detect faults and to promote fault mitigation through redesign [10, 11, 27]. Additionally, this process can also be applied to non-physical systems to detect and prevent potential cyber security and software shortfalls [28-30]. The range of application areas to which SPTA can be applied suggests that the tool is flexible and scalable for a broad use of applications.

2.3 Methods

The application of STPA requires that the user have knowledge about how the system of interest behaves, as well as how components interact with one another. Studying component interactions, rather than the probability of component failure, provides insight into failure modes that other forms of hazard analysis may overlook. The process of conducting analysis using STPA follows a four-step procedure, outlined in Figure 1 [14].

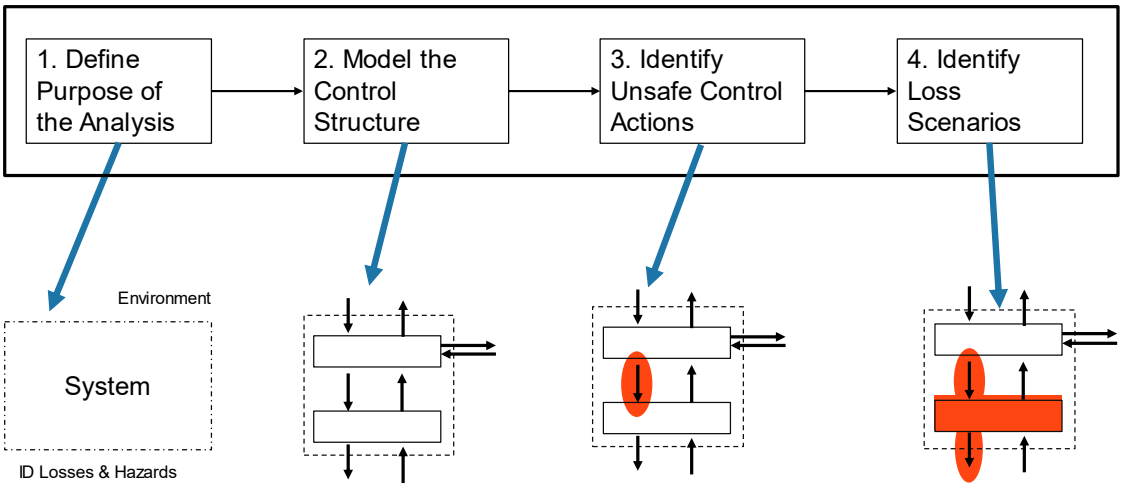

Figure 1 : STPA Flowchart.

The first step involves framing the scope of analysis to include what systems, subsystems, and external factors will be considered in the analysis. Within this step, the unacceptable losses, or undesirable situations to the stakeholder, are generated, focusing the analysis on a specific subset of the functionality of the system [14]. These losses can be defined as loss of life, loss of mission capability, loss of competitive advantage, reputation, and customer satisfaction, or even loss of critical information. Additionally, this step involves constructing conditions that, in the extreme case, could lead to the identified losses. These unsafe conditions, known as hazards, represent the worst-case scenario, and describe states that need to be prevented on a system-level as compared to an individual component basis. Hazards take the general form as seen below in Equation 1 where each value, represented by being surrounded by brackets, is part of a set [14]. This equation provides the basic structure for hazard formation, linking the hazard to the system of interest and the undesirable loss scenarios generated in the first step of the methodology.

$$< Hazard > = < System > \ AND < Unsafe\ Condition > AND < Link\ to\ Losses > \quad (1)$$

These hazards are then used to develop system constraints, or system conditions that must be satisfied to prevent the hazards. Constraints take the form seen below in Equation 2, where system constraints are made from the system of interest and linked to the unsafe situation which leads to the hazardous situation, resulting in a need to enforce conditions to prevent the hazard from emerging [14].

$$< System\ Constaint > = < System > AND < Condition\ to\ Enforce > AND \ < Link\ to\ Hazards > \quad (2)$$

Iterating this step can provide refinement to hazards and system constraints by breaking down and prescribing more fine-tuned constraints for use in the design process.

The second step in this process involves creating a control structure, or a hierarchical representation of the functionality and responsibilities of the system of interest [14]. This system is made up of a controller, which makes the decisions; control actions, which reflect commands made to enact goals; a control process, or thing being acted upon; and a control algorithm and process model, which depict the internal decision-making process and beliefs of the controller. Figure 2 depicts an example of a simple control structure, illustrating the relationship between the control and controlled process linked, by control actions and feedback.

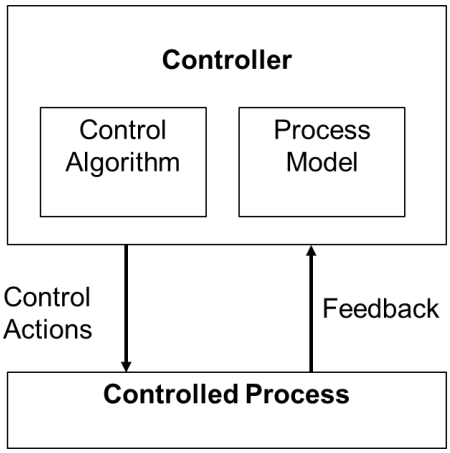

Figure 2: Basic Controller Diagram.

While Figure 2 can represent a simple, low-level model of controller and process interaction; the figure is general enough that it can also represent the basic principle of hierarchical control. Controllers placed higher along the vertical axis are control processes designated with higher levels of authority. Higher-level controllers provide commands in the form of control actions and receive feedback used to inform decision making. It is of note that this hierarchical model is not intended to be a physical or executable model but rather should illustrate internal and external feedback loops. These information flows have been widely recognized as critical to safety analyses [31].

Accurately identifying the proper control structure involves a process of abstracting the system of interest at a high level and treating subsystems as a black box early in the design process. This level of abstraction focuses the analysis on key commands and feedback systems that comprise the information flows of the system. As design iteration occurs, this black box model can be expanded to represent subsystems. This process embeds the system constraints generated in step two of the STPA methodology to create controllers capable of enforcing the constraints and preventing hazards from emerging. Establishing a detailed and representative control structure sets up the third and fourth steps of the STPA method which probes for possible weakness within the current design.

Step three of STPA analyzes each control action to determine how these actions could lead to negative scenarios when in a certain context. Unsafe control actions (UCA) fit into one of four types: not providing control action causes a hazard; providing a control action causes a hazard; control action provided too early, too late, or out of order; and control action stopped too soon, or applied too long [14]. Step three introduces worst-case scenarios to identify and classify how each controller in a system may lead to an undesirable situation. By iterating on this process, analysts can generate new hazards as necessary to expose hazards that may emerge as a result of improper control actions.

The last step of this process is identify loss scenarios, which are causal factors that can lead to UCAs and subsequent hazards [14]. This portion of the analysis aims to answer the question of why UCAs occur. Some potential scenarios that may cause improper control actions include physical controller failures, missing control inputs, or acting on incorrect information. This stage requires some abstraction to explain why UCAs may occur, allowing for traceability

that relates hazards to losses. As such, loss scenarios provide key insight into areas of shortcomings within a current design, and they enable potential improvement.

2.4 Application to Automated Vehicles

This section provides an overview of applying the STPA framework to automated vehicles. This analysis focuses on accomplishing the desired mission statement as defined below and within Figure 3:

The SL2 Automated Vehicle is a system to demonstrate lane changing, velocity control, and emergency mode switching by means of sensor fusion and adaptive cruise control; in order to safely accomplish the desired routing plans.

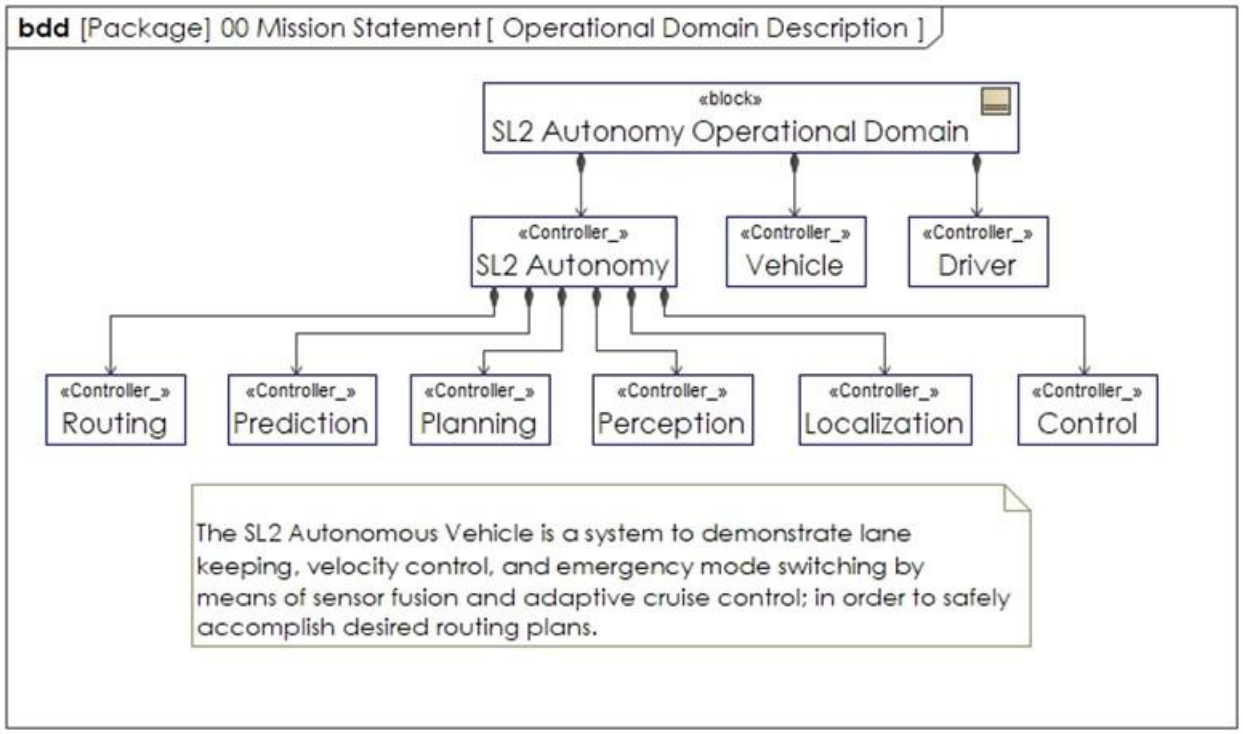

Figure 3: System of Interest- Operational Domain Description (ODD) and Hierarchical Decomposition.

2.4.1 Four Steps of STPA

For this analysis, the system of interest is an SL2 automated vehicle, driving along a multi-lane road, with other vehicles in the environment. The SL2 vehicle system consists of

sensors, actuators, software linking the two, and the human driver. In particular, this work focuses on three distinct mission modes: velocity control, elective lane changing, and mode switching. Any of these mission modes could result in unacceptable outcome of the system, where those outcomes are listed in Table 1. The losses were determined iteratively, inspired by other similar applications of STPA to automated platforms [8, 14].

Table 1: STPA Losses.

L-1: Loss or Damage to Vehicle
L-2: Loss or Damage to Environment, Infrastructure, and Landmarks
L-3: Loss of Life or Injury to People
L-4: Loss of Competitive Advantage, Reputation, or Customer Satisfaction
L-5: Loss of Mission
L-5.1: Loss of automated process
L-5.2: Loss of driver's ability to control vehicle

The above losses were used to identify *hazards*, or conditions that may lead to a loss, as illustrated in Table 2. The table documents the system hazards, connecting them to associated losses. Similar to the losses, determining these hazards was done iteratively to ensure that the scope of the analysis fully encapsulated operations in the three mission modes. With the losses and hazards defined, the system of interest could then be used to analyze the vehicle's design to develop potential mitigations and controls to remedy unacceptable hazards.

Table 2: STPA Hazards and Associated Losses.

Hazards	**Associated Losses**
H-1: Vehicle does not maintain minimum separation to other vehicles	L-1, L-2, L-3, L-5
H-2: Vehicle does not maintain lane integrity	L-1, L-2, L-3, L-5
H-3: Vehicle leaves designated navigation path	L-1, L-2, L-3, L-4, L-5
H-4: Vehicle integrity is lost	L-1, L-3, L-4
H-5: Vehicle does not maintain safe distance from nearby objects	L-1, L-2, L-5,

H-6: Vehicle enters a dangerous/unknown area	L-3, L-5
H-7: Vehicle violates safe operating speed	L-1, L-2, L-4, L-5
H-8: Vehicle exceeds safe operating envelope for lateral/ longitudinal control forces	L-1, L-2, L-4, L-5
H-9: Vehicle occupants exposed to harmful effects/ health hazards	L-3, L-4

Step two of the STPA model is to decompose the system into a control system which establishes control actions and feedback between controllers and control processes. Figure 4, depicts the high-level control diagram establishing the hierarchical responsibilities and responses between the driver, the automated process, and the vehicle itself. At this level, the automated process is treated as a black box, with its inner workings abstracted to highlight the control action semantic context between the three main systems within the operational domain.

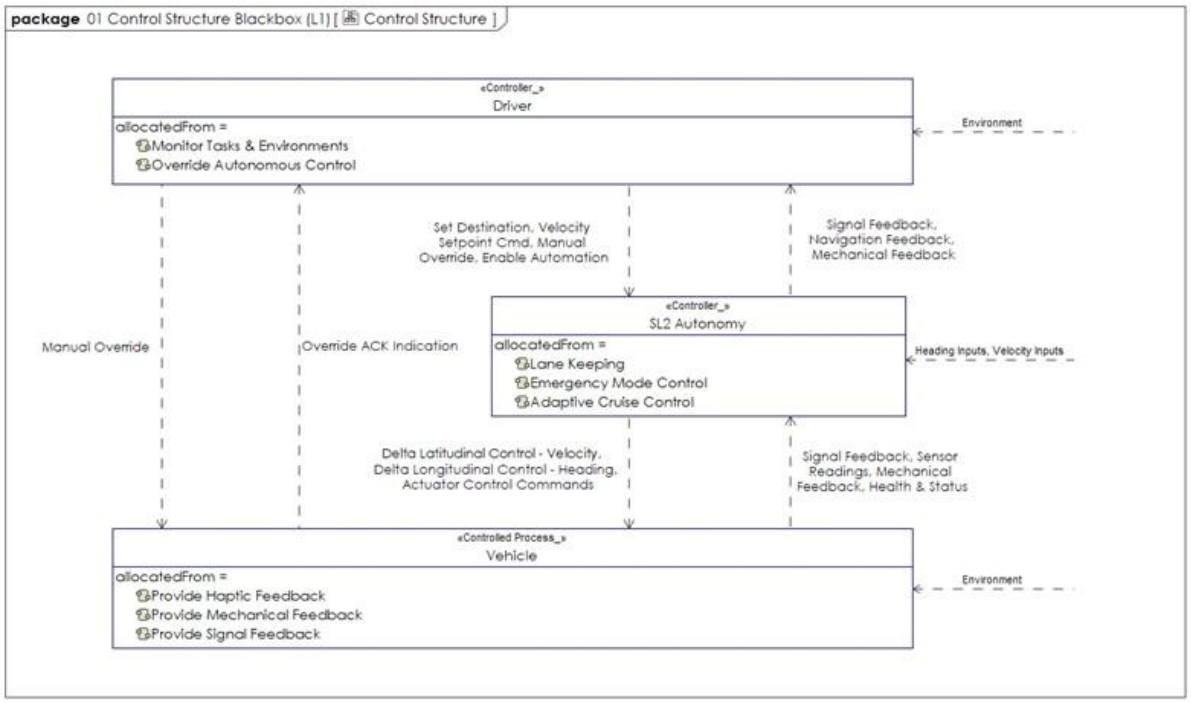

Figure 4: High Level Control Structure for Automated Vehicles.

Expanding the automated process to expose the subsystems, allows for a more detailed understanding of the system's internal hierarchy. To this end, Figure 5 takes Figure 4 and adds

the next level of detail. As shown in Figure 5, the SL2 automated process is made up of a variety of subsystems including routing, planning, perception, control, and localization. Figure 5 illustrates this hierarchical control diagram with the expanded subsystems and associated control actions and information feedback. The bottom level of this diagram represents the vehicle's physical hardware, with which the automated process communicates.

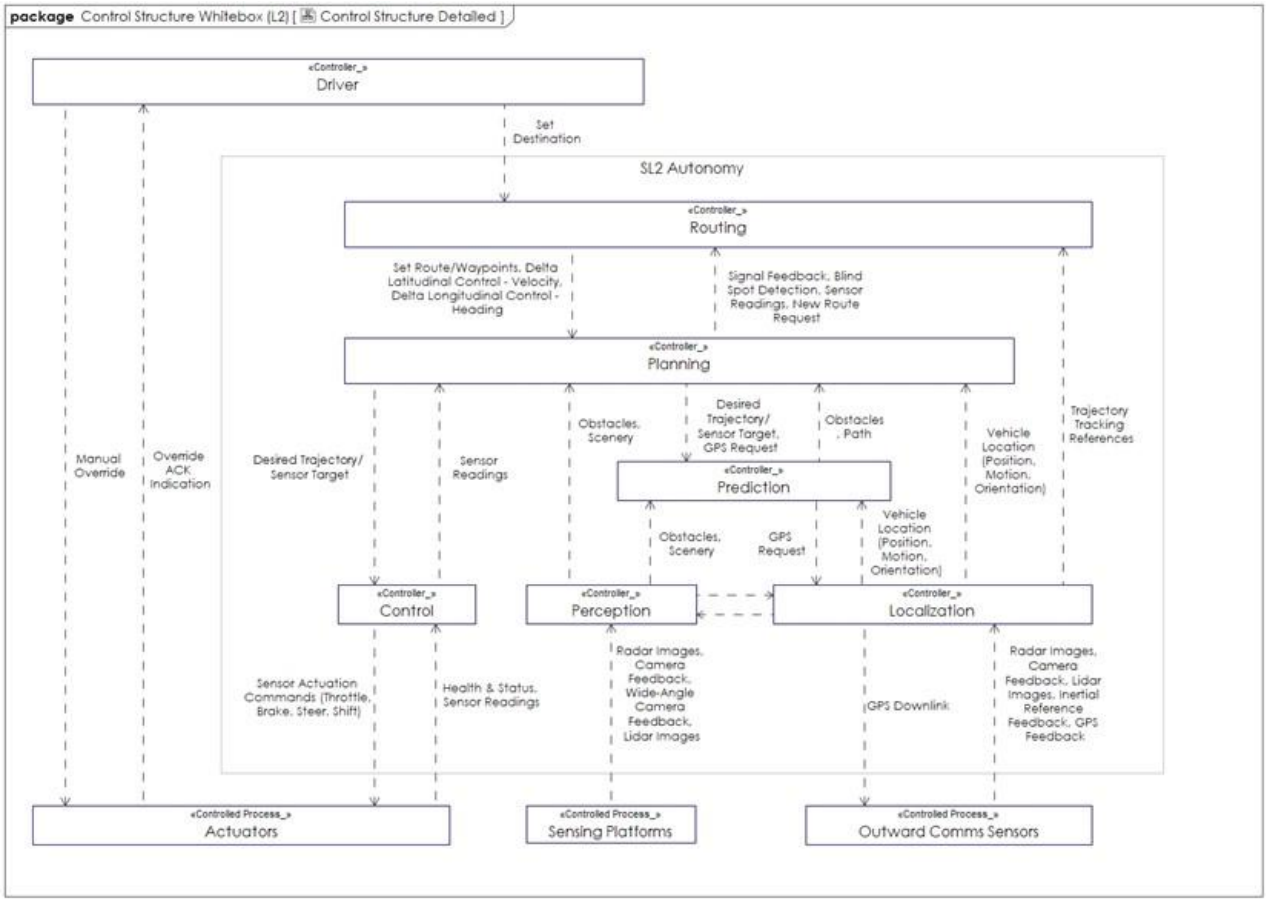

Figure 5: Hierarchical Control Structure for Automated Process with Subsystems.

Another way to model the internal function of an SL2 automation system is to visualize information flows that occur during operation. Figure 6 depicts this information flow between controllers. This figure provides insight into the following step of the analysis by highlighting information linkages between components that may be faulty due to delayed response, improper information, or even inaction which can cause the vehicle to malfunction. Additionally, this

diagram is a useful visualization for approaches to mitigating hazards, for example, by adding new blocks to the diagram. This figure also introduces external networks, Vehicle to Infrastructure (V2I) and Vehicle to Vehicle (V2V), which are essential to understanding the routing of information. These external blocks provide the vehicle with the ability to communicate with other vehicles as well as pieces of technology capable of sharing information about road conditions, traffic speed, or other data not derivable onboard the vehicle.

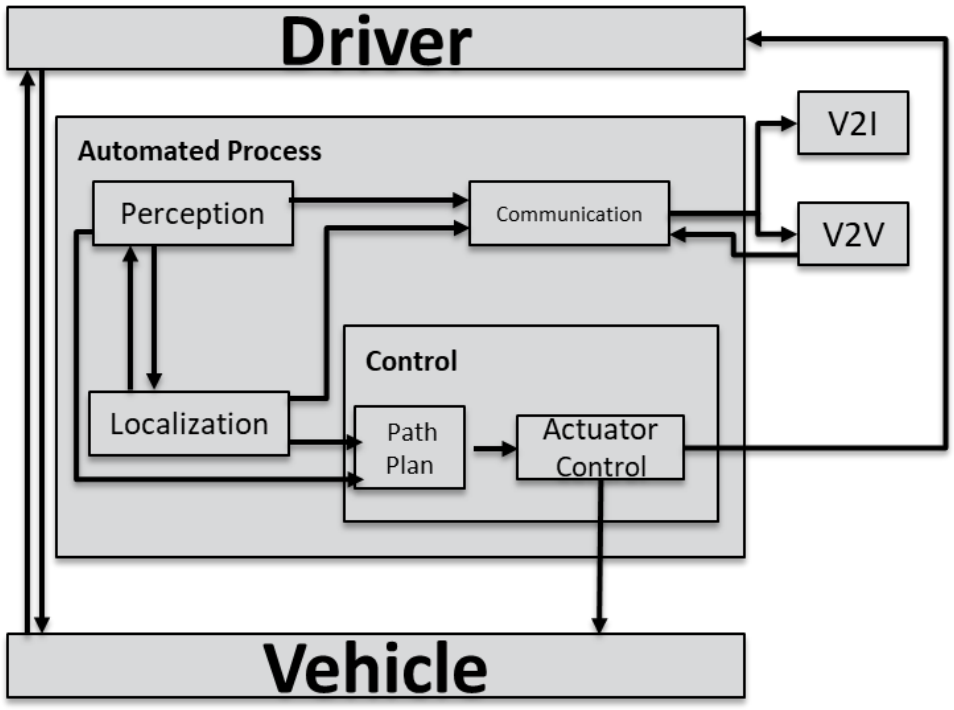

Figure 6: Action Diagram of Automated Vehicle.

Continuing to step three of the STPA process, each unsafe control action can be examined to determine how the system can malfunction. Rather than exhaustively work through the full table of UCAs, I opt in this thesis to focus only on three specific examples. Each UCA addressed is related to one of the selected mission modes of (i) velocity control, (ii) elective lane changing, and (iii) emergency mode switching. These representative examples serve to illustrate the

efficacy of STPA identifying component interaction misbehaviors that inform engineering controls being incorporated as remedies during the design process.

STPA step four involves the identification of loss scenarios to contextualize UCAs and potential mitigations. In the next subsection, this work illustrates traceability from loss scenarios to UCAs for each of the three mission modes enumerated above.

2.4.2 Three Mission Modes

Mission mode one focuses on a vehicle's velocity control, and its ability to enforce SC1-1: "vehicle must maintain TBD [m] minimum separation distance to other objects." This separation distance is illustrated in Figure 6. The separation constraint forces the vehicle to modulate its velocity to maintain the prescribed separation distance. In the scenario when the vehicle receives faulty ranging data from a perception sensor, due to sensor error or malfunction, the vehicle may operate on faulty information causing a violation of the constraint SC1-1.

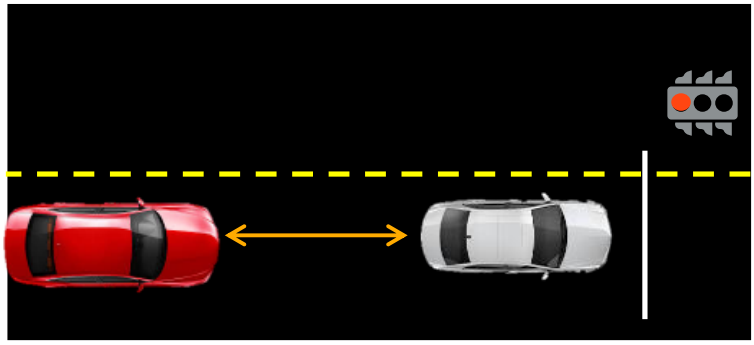

Figure 7: Illustration of Loss Scenario 1.

The traceability of these combining factors can be seen below in Table 4, which defines the hazardous context of the Braking command and resulting scenarios that can lead to violation of SC1-1, that leads to a loss of safety margin and possibly a collision.

Table 3: Braking Hazardous Contexts (UCAs).

Control Action	(Type I) Not providing causes hazard	(Type II) Providing causes hazard	(Type III) Too early, too late, out of order	(Type IV) Stopped too soon, applied too long
Brake	UCA1-1: Control does not provide the brake command when minimum separation distance with object has been violated, entering collision zone. UCA1-2: Control does not provide the brake command when making a turn.	UCA2-1: Control provides the brake command too aggressively when the brakes have degraded past 50% life. UCA2-2: Control provides the brake command too aggressively when roadways are slippery from weather conditions. UCA2-3: Control provides the brake command when the velocity setpoint has not been achieved.	UCA3-1: Control provides brake command too early when rear vehicle does not have sufficient response time to decelerate to avoid collision. UCA3-2: Control provides brake command too late when collision is imminent.	UCA4-1: Control releases brake command too early before vehicle decelerates to avoid collision.

Table 4: Loss Scenario 1: Following Failure.

UCA 1-1:	UCA 1-2:	UCA 3-2:	UCA 4-1:
Control does not provide brake command when minimum separation distance with object has been violated entering the collision zone	Control does not provide brake command when making a turn.	Control provides brake command too late when collision is imminent.	Control releases brake command too early before vehicle decelerates to avoid collision
Loss Scenario (Inadequate Control Algorithm | Incorrect Feedback)			
SCN-1: Because the camera & radar object detectors (vehicle, pedestrian) cannot characterize and track the targets of interest due to poor weather conditions, which cause noisy sensor readings of the forward-facing objects.			

Mission mode two involves the vehicle's inability to maintain a lateral separation when changing lanes. This unsafe situation is a violation of constraint SC5-1: "Vehicle must maintain

[TBD] safe lateral distance from nearby objects, vehicle, and pedestrians." In line with mission mode 2, the following example serves as a potential loss scenario which may occur in an off-nominal setting.

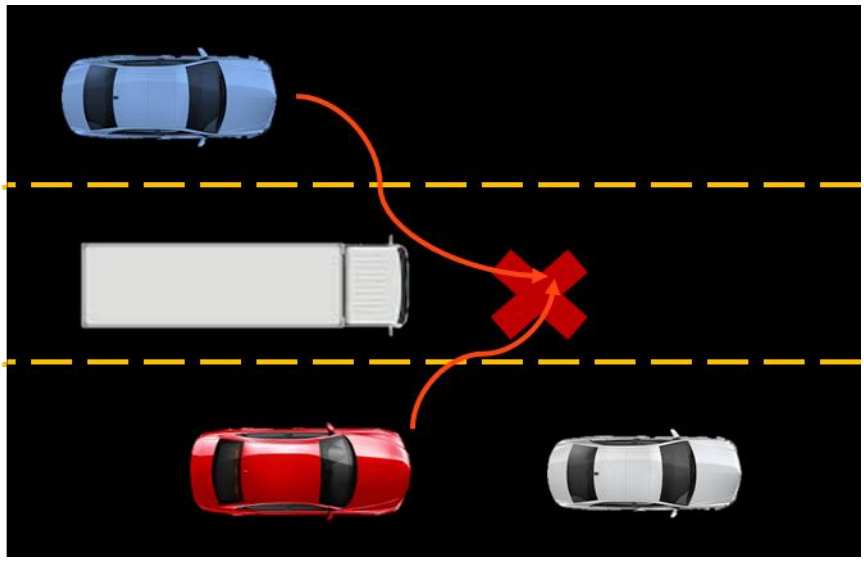

Figure 8: Illustration of Loss Scenario 2.

As a mechanism that might cause a loss of lateral separation, we consider an elective lane change scenario. As described above, SL2 automation only allows a vehicle to maintain speed and separation within its own lane; however, a driver might prefer to pass a slower vehicle immediately ahead. In this case, the driver might elect to disable SL2 driving in order to initiate a lane change. If a second vehicle were to initiate a lane change at the same time, a loss of lateral and longitudinal separation (and possibly a collision) might occur. This scenario is illustrated in Figure 8, where the SL2 vehicle (red) approaches a slower car prompting the driver to override SL2 automation in order to switch lanes. A third vehicle (blue) decides to switch lanes into the same location, which threatens the mission.

The UCAs for this situation are summarized in Table 5Table 5. Inaccurate localization or sensor information is provided to the planning and prediction controllers due to latency issues

associated with recognizing and tracking the third vehicle's intent, when the driver disengages from the SL2 automated driving to make a lane change.

Table 5: Hazardous Contexts for Planning Controllers

Control Action	**(Type I) Not providing causes hazard**	**(Type II) Providing causes hazard**	**(Type III) Too early, too late, out of order**	**(Type IV) Stopped too soon, applied too long**
Desired Trajectory	UCA 5-1: Planning does not provide the desired trajectory command when human initiates lane change.	UCA6-1: Planning provides the desired trajectory command when routing to dangerous location/lane.	UCA 7-1: Planning provides desired trajectory command too late after manual override was engaged.	UCA 8-1: Planning stopped the desired trajectory command too soon before latitudinal and longitudinal control achieved desired waypoint.
GPS Request	UCA 9-1: Prediction does not provide the GPS Request command when the vehicle location is unknown. UCA 9-2: Localization does not provide the GPS downlink command when the vehicle location is unknown.	UCA 10-1: Planning provides the GPS Request command when the satellite uplink/downlink channels are inactive, and collision is imminent.	UCA 11-1: Localization provides the GPS Request command too late when collision is imminent and SL2 automated vehicles have no line of sight with horizon.	UCA 12-1: Localization stopped GPS Request command prior to data exfil completion.
New Route Request	UCA 13-1: Planning does not provide the New Route Request command when the predicated obstacle path intersects with the vehicle planned trajectory.	N/A	N/A	N/A

Table 6: Loss Scenario 2 Vehicle Merging Failure.

UCA 5-1:	**UCA 13-1:**	**UCA 8-1:**	**UCA 11-1:**
Planning does not provide the desired trajectory command when humans initiate lane change.	Planning does not provide the New Route Request command when the predicated obstacle path intersects with the vehicle planned trajectory.	Planning stopped the desired trajectory command too soon before latitudinal and longitudinal control achieved desired waypoint.	Localization provides the GPS Request command too late when collision is imminent and SL2 automated vehicle has no line of sight with horizon.
Loss Scenario (Inadequate Control Algorithm | Incorrect Feedback)			
SCN 2: Because the merging vehicle was not detected due to latent perception feedback associated with characterizing and tracking the other vehicle's intent. Without proper detection the vehicles will likely merge into one another causing a collision. **SCN 3: Because the driver initiates a lane change, leaving the SL2 automation, steering the vehicle into a potentially dangerous area.**			

Mission mode three explores the idea of an emergency mode switch triggered by the automation, which occurs if the automation decides its function is compromised. The sudden transition of low-level control responsibility from the SL2 automation to the human pilot is recognized as a major limitation of the SL2 architecture. In this loss scenario, we consider the particular case when an emergency situation cannot be resolved by the automation, such as in the case when a vehicle ahead of the SL2 vehicle brakes suddenly, in such a way that a collision would be imminent if the SL2 vehicle remains in the same lane. The SL2 vehicle might transition control abruptly to the human in this case, asking the human driver to initiate an emergency lane change.

In this example the vehicle attempts to maintain the in-lane separation constraint (SC-1) but consequently transitions control to the driver in a way that fails to adhere to constraint SC2-1: "Vehicle must maintain lane integrity to TBD certainty" and constraint SC 6-1: "Vehicle must remain on known/designated drivable areas." It is up to the driver in this simulation to manage the risks associated with violations of SC2-1 and SC6-1. If the driver cannot maintain these constraints, a potential collision might occur. A visualization of this loss scenario is illustrated

below in Figure 9. This figure represents the difference in output between the maximum possible response and the automated vehicle's planned route. In this scenario, the vehicle enters into an unknown decision and reverts control back to the driver, who is unable to act fast enough to avoid collision with an object in the roadway.

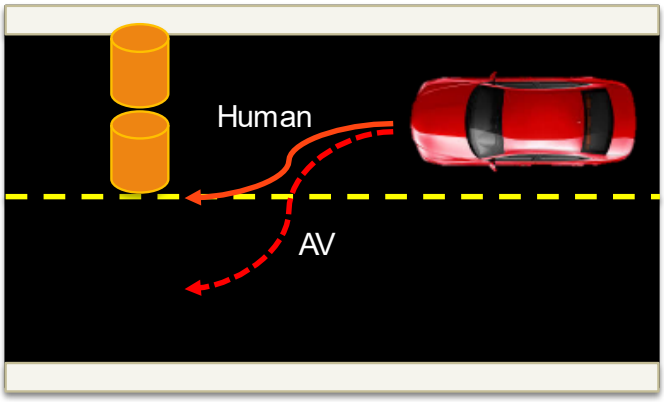

Figure 9: Illustration of Loss Scenario 3.

In an example of how the driver might act, we consider a scenario in which the driver provides insufficiently aggressive steering, perhaps out of a panicked uncertainty about whether the adjacent lane is clear. If the driver applies insufficiently aggressive steering, the driver may fail to clear the rapidly breaking vehicle ahead, causing a loss of in-track separation. These UCAs and the resulting loss scenarios are summarized in Table 8.

Table 7: Hazardous Context for Planning Controller in Emergency Mode Switch.

Control Action	**(Type I) Not providing causes hazard**	**(Type II) Providing causes hazard**	**(Type III) Too early, too late, out of order**	**(Type IV) Stopped too soon, applied too long**
Manual Override	UCA 14-1 Driver does not provide the manual override command during inclement weather when the environment is noisy.	UCA 15-1: Driver provides the manual override command when vehicle is on route to desired trajectory.	UCA 16-1: Driver provides the manual override command too late when there is no visual line of sight of existing obstacles.	UCA 17-1: Driver provides the manual override command too long when lateral separation is reached.
Desired Trajectory	N/A	N/A	UCA 18-1: Planning provides the desired trajectory command too late after, after manual override was engaged.	N/A

Table 8: Loss Scenario 3 Emergency Mode Switch.

UCA 14-1:	**UCA 15-1:**	**UCA 16-1:**	**UCA 18-1:**
Driver does not provide the manual override command during inclement weather when the environment is noisy.	Driver provides the manual override command when vehicle is on route to desired trajectory.	. Driver provides the manual override command too late when there is no visual line of sight of existing obstacles.	Planning provides the desired trajectory command too late after, after manual override was engaged.
Loss Scenario (Inadequate Control Algorithm | Incorrect Feedback)			
SCN 4: The driver does not have the clearance to maneuver without causing a collision to neighboring cars. **SCN 5: Because the planning controller has control authority to correct driver input when the unnecessary to maintain control authority. This generally occurs at a response rate beyond the driver's ability. Automation overcorrecting an already abrupt maneuver can lead to displacement causing a crash/fatality.**			

` To remedy these three loss scenarios, new system requirements are needed. Those requirements might be satisfied by adding new risk mitigations to the system. Table 9 introduces recommendations for each of the three loss scenarios described above.

Table 9: Summarized Loss Scenarios and Mitigations.

SCN ID	Summarized Flaw	Recommended Requirements	Recommendation Type
SCN-1:	Sensors provided different measurements for hazard in front of vehicle.	Propose voting algorithm to fuse the sensor measurements and take the most common or most trusted measure.	Signal fusion to ensure data sent to planning controller is accurate.
SCN-3:	Improper routing information sent to the planning controller.	Routing information must be checked against external vehicle communication to ensure vehicle is not moving to an occupied space.	Check planning information versus V2V and V2I information.
SCN-4:	Driver provides an inadequate amount of heading change in an emergency mode switching scenario.	Propose weighting the automated process input into the system so that the human driver can safely navigate around potential objects.	Control structure changes to prevent scenario.

These recommendations, while promising, need to be evaluated to ensure compliance with the requirements they are attempting to enforce. Simulation-based evaluations will be explored in Chapter 3: Simulation to Verify Results of STPA Model.

2.5 Conclusion

While much improvement has been made in the realm of automated vehicles in past years to increase their safety, more work is needed to ensure future development of even safer systems. Using the STPA method, SAE level 2 vehicles were analyzed to determine points of interest where vehicle behaviors are hazardous, potentially leading to crashes and fatalities. When STPA is utilized early in the design process, the tool can be useful to identify ways to improve the system.

This research found three key areas of improvement within automated vehicles, one for each of the three mission modes: velocity control, elective lane changing, and emergency mode

switching. By instituting these proposed mitigations, vehicles can reduce the frequency they enter dangerous situations.

Selecting three mission modes allowed for a case study to demonstrate the application of STPA towards SL2 vehicle automation for the first time.

Chapter 3: Simulation to Verify Results of STPA Model

3.1 Introduction

Vehicle manufacturers sometimes propose design changes theoretically capable of assisting drivers and augmenting safety, but which in application serve the opposite effect. For example, Volkswagen introduced a touch-sensitive steering wheel control panel which managed speed and other functions in certain vehicle models [33]. To increase the likelihood that proposed design changes are beneficial, early-stage simulation is useful. Unfortunately, many existing simulation methods are too complex for use in the early stages of design, due to a requirement for intensive computation or for inclusion of as-yet-unspecified design details. This chapter introduces an analysis approach with complexity tuned specifically to the level of modeling used in STPA; the goal is that the analysis should provide meaningful assessment of the design recommendations produced by STPA, while maintaining a high-level of abstraction, similar to that of STPA.

The primary contribution in this chapter is the derivation of the Local Investigation of Mitigation Efficacy (LIME) tool that can be used as a complement to the STPA process. To test our proposed process, it is applied to the design recommendations that were found using the STPA methodology in the previous chapter. Recalling that these analyses did not identify entirely new design advice; rather, they re-identified design recommendations previously proposed in the open literature. As such, we expected our analysis tool to demonstrate the correct result, namely that these mitigations are beneficial. The analysis confirmed this hypothesis, suggesting its utility in a further refined STPA analysis of SL2 automation.

3.2 Background

When utilized early in the design phase, STPA has the potential to highlight areas of improvement in a system's structure. These recommendations, while they may seem promising, need detailed testing to determine their efficacy in addressing identified problem areas. Ideally, an analysis would simulate the problem identified by the STPA process. The simulation should include a high-level abstraction of the system automation, an abstraction similar in complexity to the information-flow modeling used to define the STPA process.

3.2.1 Model Based System Engineering (MBSE)

In the systems-engineering community, simulation-based approaches are called Model Based System Engineering (MBSE). MBSE tools are useful for design, calibration, and validation of generated system requirements [34]. MBSE tools can also serve to reduce waste by directing engineering efforts to prevent future integration issues. When executed in parallel to the STPA process, the MBSE validation has the potential to inform design iterations by checking the effectiveness of assigned constraints, mitigations, and system outcomes [35]. For purely software systems, MBSE has been incorporated into STPA for embedded model checking [36].

The challenge for combining MBSE with STPA for automated vehicle systems is to define the correct level of resolution for the simulation. While other literature has analyzed components of automated vehicles to derive system constraints or simulated vehicle performance in different environments, this work focuses on combining the derived system requirements with simulated vehicle performance to evaluate design recommendations made during the STPA process [31, 34]. An appropriate MBSE methodology should be no more complicated than required to show that a design alternative can mitigate a potential loss. It is not necessary to fully characterize the severity or risk before and after mitigation; rather, it is simply necessary to show

that there is likely to be some reduction in risk or some reduction in severity, as highlighted in Figure 10.

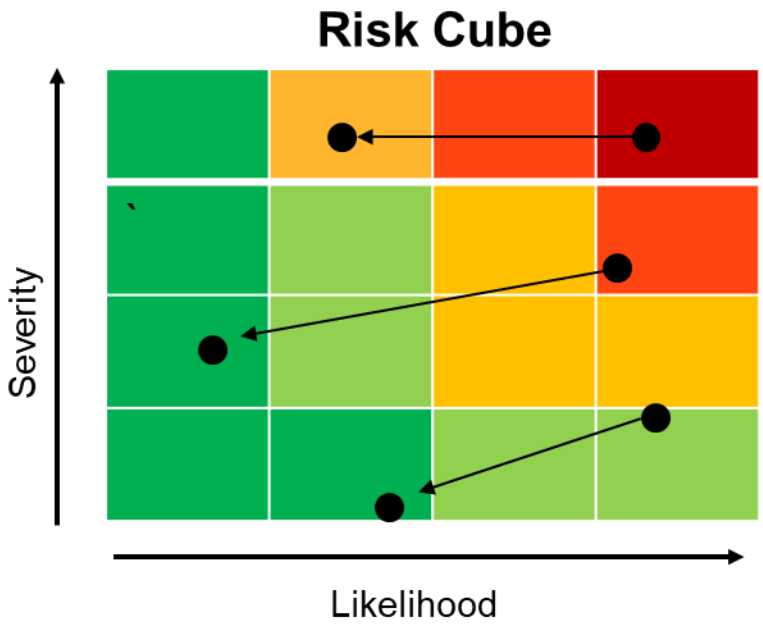

Figure 10: Risk Cube depicting proposed mitigation approaches.

3.2.2 Modeling SL2 Driving

Ideally, it would be possible to apply an MBSE tool to simulate a range of automated hardware systems; as a starting point, this chapter only considers the specific case of modeling SL2 driving. Existing models of fully autonomous systems have been used for verification; however, these models are insufficient for simulating SL2 driving as they fail to model human involvement within the system and fail to encapsulate early stage design recommendations [37, 38]. In the next section, we introduce a particular simulation-based evaluation methodology where the simulation is customized to SL2 driving.

To show that this verification process is functional, we will apply it to the three mission modes identified in the prior chapter. These mission modes have known solutions that have been identified previously in the literature; as such, they are good candidates to confirm that our

proposed SL2 driving tool and associated evaluation methodology is appropriate to analyze the benefit of these known solutions. The remainder of this section provides additional background for each mission mode and related solutions.

Mission mode 1 involves a sensor fault. Based on STPA analysis, a reasonable mitigation is a multi-sensor voting algorithm. In the research literature, voting algorithms have been utilized as a method of sensor fusion to deliver an end result without relying too heavily on information from one specific sensor [39]. In practice, taking these multiple sensor reading and using them to acquiesce to an agreed measurement, provides a distinct advantage in reducing the amount of information required to come to a solution, while potentially eliminating an erroneous data source [40]. The drawback of voting algorithms is cost, in the sense that redundant sensors must be deployed, even though their data is not needed except in rare corner fault cases.

Mission mode 2 involves a merging fault when the driver de-activates automation. Based on STPA analysis, a reasonable mitigation is to exploit Vehicle-to-Vehicle (V2V) communication. Vehicle communication networks provide a means for vehicles and their drivers to send and receive information about their operating environment, possibly reducing the number of vehicle accidents [41]. Challenges remain associated with communication latency, certification, and spectrum therefore more research is needed to achieve safety benefits; however, the ability to prevent collisions between merging vehicles is one potential benefit of V2V communication [42].

Mission mode 3 involves emergency mode switching triggered by a hazard in the roadway. Based on STPA analysis, a reasonable mitigation is the introduction of a driver-assist capability to help drivers change lanes safely. Emergency mode switching is believed to be the biggest safety risk facing SL2 automated vehicles [43]. As such, safe emergency mode switching

controllers are needed to support transitions to human control. Fortunately, it is believed that mode-switching controllers can be made safe for a range of applications including shared autonomy and vehicle platooning [44, 45].

3.3 Methodology

By testing STPA generated solutions to known problems we can test the value of those concepts and compare them to existing solutions found in the literature. To accomplish the goal of verification of design parameter compliance and efficacy, a model was defined to accomplish three goals of velocity control, lane changing, and accomplishing navigational touring information- which align to the three mission modes as outlined in Chapter 2. These are a limited set of possible mission modes for SL2 driving; however, these three particular modes offer a span of space of shared autonomy by including faults that occur during automatic control (Mission Mode 1), during human-initiated control transition (Mission Mode 2), and during an emergency-response control transition (Mission Mode 3).

A common simulation-based testing procedure was applied to analyze all three mission modes. This section details that methodology, including the approach used to define the evaluation space and the models employed in the simulation.

3.3.1 Evaluation Procedure

The novelty of my approach is in defining an evaluation procedure with a level of complexity well-matched to STPA. STPA is a rigorous approach to mapping out information flow structures through a complex system, one that can be used to identify problems with those control actions and information flows that could result in an undesirable outcome (also known as a *loss*). One of the most important aspects of STPA is abstraction. The system should be

abstracted as much as possible, with the abstraction level just sufficient to reveal key information flows across the system but not lose them. This abstraction makes STPA a flexible design tool, one that can be deployed early and iteratively in the design process.

To evaluate design changes proposed during iterative design with STPA, an evaluation tool is needed with a similar level of abstraction. It does not make sense, for instance, to define an evaluation tool that tests a design change for all possible configurations or that computes a probabilistic assessment (for example, as evidence to support a node in a fault tree), bur they require too much specific design documentation and too much computation to implement during flexible, early-phase design.

My key insight is to introduce an evaluation procedure that tests only proof of concept. In other words, instead of searching the entire operational space to evaluate the design change, I propose to select one example that embodies a loss identified in the STPA procedure. If a simulation can be used to show that the proposed mitigation partially or fully resolves that loss, then it is generally worth refining the proposed design change and considering how to incorporate the change into the next phase of design. In that sense, the evaluation tool needs only to explore a compact parameter space in the immediate vicinity of a particular loss example, in order to assess how well the design changes partially or fully resolves the issue in that localized space.

It is useful to visualize the local parameter space as a multi-dimensional hypercube, As illustrated in Figure 11, the hypercube takes the form of a square where there are only two main parameters needed to define the operational space in the immediate vicinity of a loss. By discretizing the parameter space (i.e. the square shown in Figure 11) and running a simulation for each discretized combination of parameter values, the simulation can be used to asses the degree

to which the mitigation expands the envelope of safe operation in the immediate vicinity of a loss event.

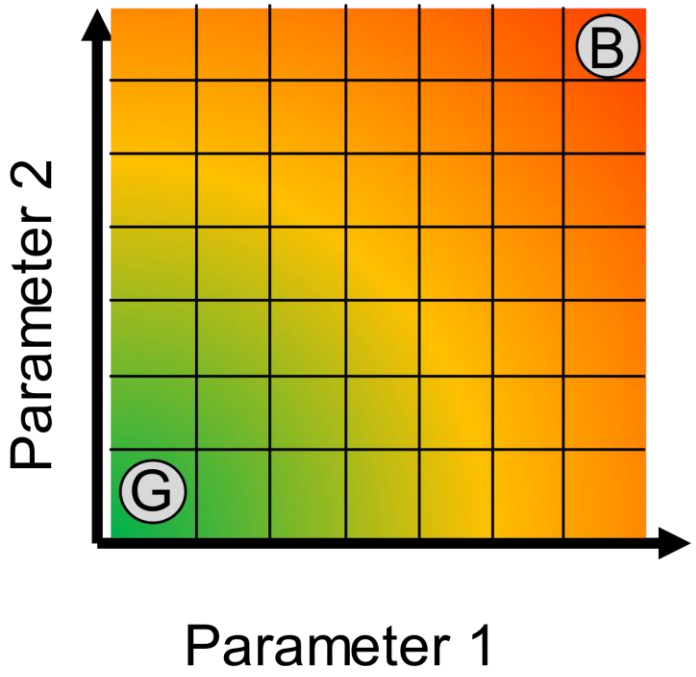

Figure 11: Example Figure of Proposed LIME Tool for Evaluation Model Extension Efficacy.

To define the local search space, an engineer must take two steps. First, the engineer must identify the key parameters the influence the outcome of the simulation. Second, the engineer must define the range of those parameters. The first step requires critical thinking about the appropriate level of abstraction for the simulation and about the associated parameter space. Identifying important parameters may require a sensitivity analysis to further direct the analysis. The second step requires identifying the parameters associated with a loss and then expanding the parameter space slightly. Again, this process may require some iteration on the part of the engineer utilizing the tool. To simplify this process, we recommend the engineer focus on identifying only two points in the operational parameter space, one case in which the simulation

results in a loss (the bad case in Figure 11 labeled *B)* and one case in which the parameters are adjusted enough to prevent the loss (the good case labeled *G)*.

Because this proposed process of defining a local parameter exploration is an art, rather than a science, it is important to provide a clear example. The result of the methodology section outlines a simulation of SL2 driving, created at a level of abstraction similar to that of STPA from Chapter 2. Once that simulation methodology is defined, we apply it to the three mission modes, in each case identifying a local parameter space that illustrates the transition between a loss case and a safe case.

3.3.2 Mathematical Model for Simulation

Since our methodology only needs to evaluate a proof-of-concept example for a given design modification, it is acceptable to simulate a single operational case. As such, we define the simulation for an Operational Domain Description (ODD) associated with a common type of operation: driving on multiple parallel, adjacent lanes with traffic moving in the same direction. A reasonable level of abstraction for this case is to consider the interaction of only two vehicles, where both vehicles have all the capabilities of typical SL2 system, with the driver supervising control of vehicle function with assistance in the form of automated steering and velocity control [6].

The simulation evaluates a vehicle's performance for a specific set of parameters describing the ODD. By varying those parameters, it is possible to sweep a local region of the parameter space in the vicinity of a loss, as described in Figure 11. The number of occurrences of system safety constraint violations that occur across the local space can be tallied, indicative of a vehicle's performance under a given set of conditions.

To simulate the two SL2 vehicles in adjacent lanes, as depicted in Figure 12, a dynamic simulation is needed. The simulation should capture critical aspects of the system identified by the STPA process including sensors and vehicle communication networks. The detailed mechanical design of the vehicles is not important; as such we model each vehicle as a point mass moving along a 2D plane (see right hand side of Figure 12). A loss of separation event occurs if the two point-masses draw too close. A loss of separation event, representing a collision or near collision, is considered an unfavorable outcome (a *B* on Figure 11). If a loss-of-separation does not occur, the outcome of the simulation is considered to be favorable (a *G* on Figure 11).

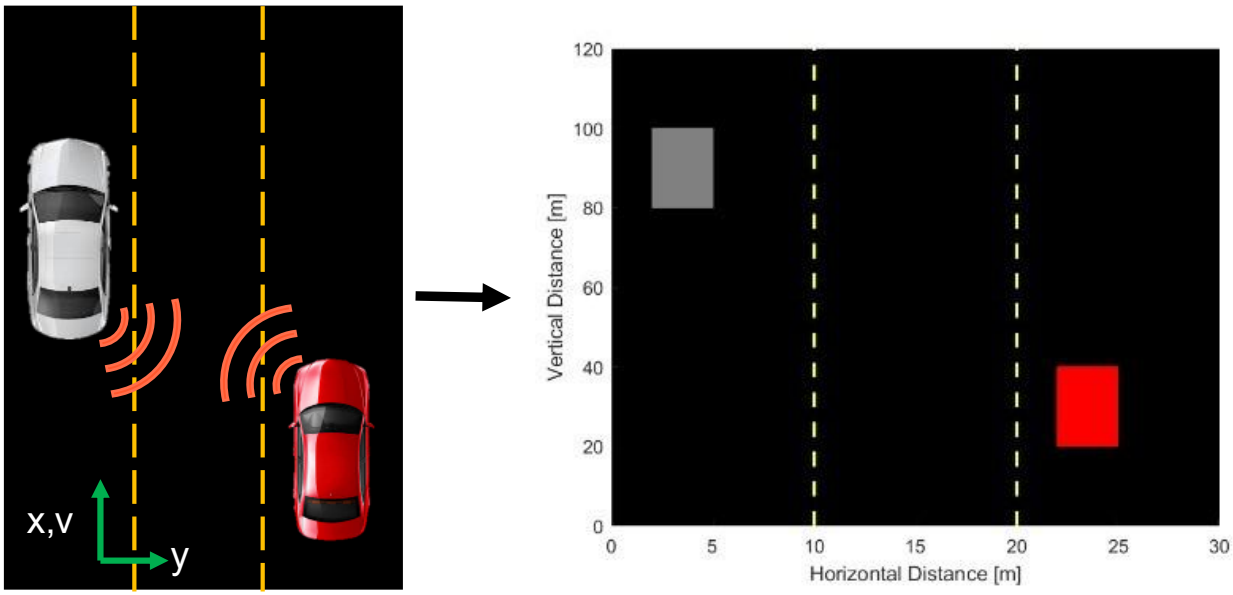

Figure 12: Depiction of Operation Domain Description (ODD).

The dynamic model involves one discrete state and three continuous states for each vehicle. The discreate state is a driving mode from the list {*drive, brake/stop, follow, change lanes, exit*}. The three continuous states are along-track position x, along-track velocity v, and lateral position y. At each time step these states are updated according to rules described below

The discrete state is updated using a finite state machine (FSM), An FSM can model vehicle performance based on a given set of vehicle parameters and simulated localization, perception, and control information [46]. This chapter uses a specific FSM with the structure shown below in Figure 13, the discrete state for each vehicle begins in the "drive" mode, in which a vehicle maintains its desired speed. From this state a vehicle may enter either "brake" if the distance between the current and lead vehicle is less than the threshold value; or choose to change lanes if the driver desires.

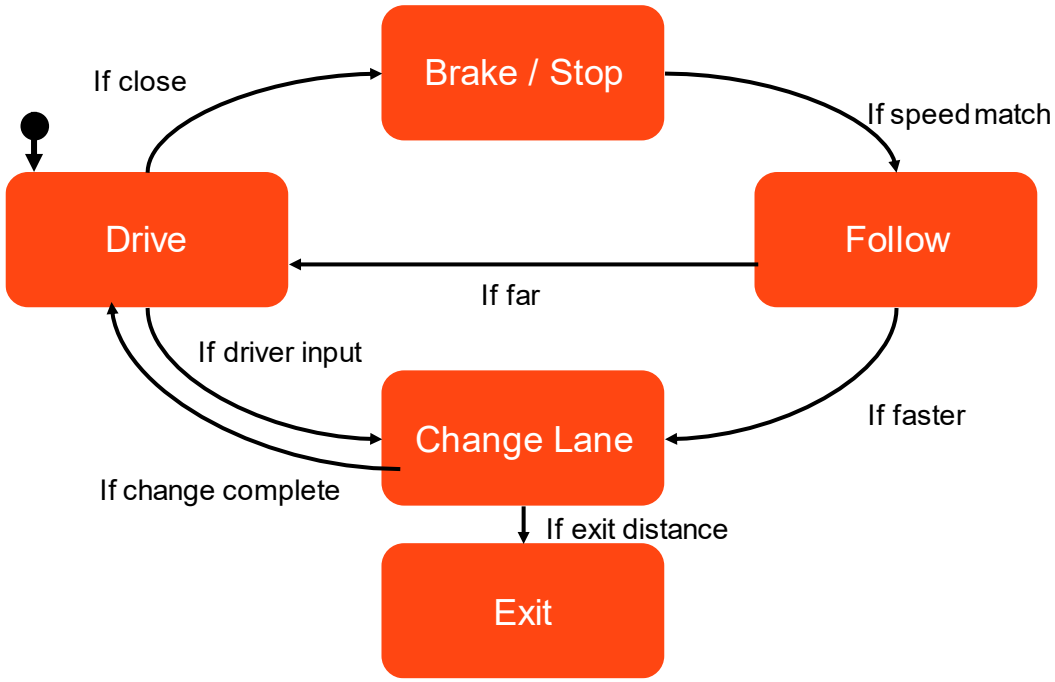

Figure 13: Finite State Machine Depiction of the System.

The continuous states for each vehicle are updated at each time step by applying dynamic equations appropriate to the discrete mode of operation. In all discrete states, the along-track position is advanced with a forward Euler approximation that depends on the vehicle's previous position (x_{k-1}) and velocity (v_{k-1}) as well as the time interval (Δt).

$$x_k = x_{k-1} + v_k \Delta t \tag{3}$$

The cross-track or lateral position of the vehicle usually remains constant, except during a lane change. While a vehicle is executing a lane change, the lateral position is specified geometrically by a sigmoid function $g(k: k^*, \delta, \alpha)$.

$$g(k: k^*, \delta, \alpha) = \frac{\delta}{1 + e^{-\alpha(k-k^*)}} \tag{4}$$

In this function, the time step k is compared to k^*, the time at which the lane-change maneuver began. The speed of the lane change is defined by the parameter α. The direction of the lane change is defined by the parameter δ, where $\delta = -1$ for moving to a left lane and $\delta = 1$ for movement to a right lane.

The lateral position during the lane change in the "change lane" case is referenced to the initial value y (the y position when the lane change is initiated); the shift is computed by multiplying the sigmoid function by an estimated lane width ϕw where w represents the true lane width and ϕ is a scale factor representing estimation error of vehicle heading change. The ϕ term allows for modeling of emergency driver response, modulating the amount of heading change the vehicle makes in scenarios where the driver must initiate a lane change without having full situation awareness. Combining the constant $y_k$modes and "change lane" case gives the below expression for cross-track position.

$$y_k = \begin{cases} y_{k-1} & \{Drive, Brake, Follow, Exit\} \\ \phi w * g(k: k^*, \delta, \alpha) + y & \{Change\ Lane\} \\ \phi w + y & Transition\ from\ Change\ Lane \end{cases} \tag{5}$$

The velocity used in determining the along-track vehicle position is dependent on the discrete state of the FSM. Typically, the vehicle travels at the self-selected speed, $v_{prefered}$. If the vehicle is following a leader, then the vehicle matches the lead vehicle's speed v_{lead}. In the discrete state of "brake/stop" the vehicle decelerates at a constant rate, until the separation

distance has been restored. The deceleration rate is modeled as having the same value for all time and all vehicles. The velocity update can be summarized as follows:

$$v_k = \begin{cases} v_{prefered} & \{Drive, Change\ Lane, Exit\} \\ v_{lead} & \{Follow\} \\ \max\left(v_{k-1} - \alpha\Delta t, 0\right) & \{Brake\} \end{cases} \quad (6)$$

Note that a special case has been added for the case when the discrete state of the FSM leaves the transition "change lane" mode. Because the sigmoid function exponentially converges toward its final value (without ever fully converging), the lateral state is set to its final limit value of $\phi w + y$, upon transition out of the "change lane" mode.

Next, with the updates of the three continuous states defined, it is important to define the gates governing the transitions of the FSM's discrete state. In all there are seven possible state transitions, as indicated by the seven arrows in Figure 13. The conditions that trigger the transition in each case are summarized in Table 10.

Table 10: Gating Conditions for FSM.

From	To	Gating Condition
Drive	Brake/Stop	$(x_1 - x_2) \leq x_{threshold}$
Brake/Stop	Follow	$(v_1 - v_2) < v_{threhold}$
Follow	Drive	$(x_1 - x_2) > x_{threhold}$
Follow	Change Lane	$v_{preffered} > v_{lead}$ AND $(y_1 - y_2) > y_{threshold}$
Drive	Change Lane	$DriverInput = True$
Change Lane	Drive	$y_k = \phi w + y$
Change Lane	Exit	$x_k = x_{max}$

The calculation of the separation distance can be seen below in Equation 7 where the absolute difference of a vehicle's position is used to simulate data readings that would be available to perception systems such as lidar. When the distance between two vehicles violates the minimum required threshold, the vehicle must slow to match the speed of the leading vehicle

and reestablish the threshold following distance. This violation of the following threshold triggers the "if close" gate, forcing the trailing vehicle to brake and slow its speed before entering into another state.

$$(x_1 - x_2) \leq x_{threshold} \rightarrow brake \tag{7}$$

Equation 8 governs the control action administered within the brake block, dictating that when the velocities between the two vehicles have been matched the "if speed match" gate triggered and the vehicle enters the following mode, holding a constant velocity equivalent to the lead vehicle.

$$(v_1 - v_2) < v_{threshold} \rightarrow follow \tag{8}$$

In the case where the lead vehicle speeds up or leaves the road the gating condition of "if far" is met. This condition is summarized in Equation 9 in which the distance between the two vehicles exceeds the following threshold distance, therefore the vehicle accelerates to its preferred cruising velocity and drives until it encounters another vehicle or object in the road.

$$(x_1 - x_2) > threshold \rightarrow drive \tag{9}$$

The last transition gate is to exit the simulation once the vehicle has reached its required exit distance, as prescribed in the vehicle's given parameters. This ends the simulation for the vehicle of interest but allows other vehicles to continue operating on the simulated road as the finite state machine is utilized independently by all vehicles at every timestep.

3.3.3 General Simulation Set Up

The simulation is a deterministic model of driving performance, representing an open loop control model for each automated vehicle. The simulated environment consists of three straight, concurrent lanes of fixed length with constant width and length. Some parameters were

kept constant for simplicity in order to model the desired mission modes. The parameters for this simulation can be seen in Table 11, which represent the variables used in the basic simulation.

Table 11: General Parameter Values.

Parameter	**Value**
Acceleration,a	2 m/s^2
Vehicle 1 Starting Speed	40 m/s
Vehicle 1 Starting Position (along-track)	40 m
Turn Steepness, α	2
Number of Vehicles	2
Road Length	1000 m
Lane Width, w	10 m
Along-track position threshold, $x_{threshold}$	20 m
Along-track velocity threshold $v_{threhold}$	0.01 m/s
Cross-track threshold, $y_{threshold}$	2 m

In the basic simulation, each vehicle enters the road, traveling at its desired speed in its starting lane. Unless otherwise specified, the vehicle starts in the far-right lane (denoted as lane position 1). When this vehicle perceives another vehicle or object in its simulated ranging sensor data, it then acts to either slow its speed to match the vehicle moving upstream of it, or it initiates a lane change. This decision is indicative of the choice a driver must make when driving down the road and sometimes provides a difficult choice for the driver. The simulated vehicle continues driving down the road, resuming at its optimal speed if it is able, until it reaches its terminal point along the length of the road. At this point, the vehicle exits the roadway and the simulation is complete. This process is applied independently to each vehicle and updated at each timestep, with the vehicle that is further downstream of the origin being the primary agent to update its position and velocity.

3.3.4 Specific Simulation Set Up

In order to alter the general SL2 vehicle simulation to demonstrate the effectiveness of the proposed LIME tool, the basic simulation was customized to represent three specific mission modes: velocity control, elective lane change, and emergency mode switching. Using these three examples, the LIME tool explores a local region of the parameter space for each mission mode to analyze the effect that STPA proposed mitigations have on system safety. While limited in its scope, these three case studies allow for the comparison of vehicle performance with and without the suggested mitigations to assess their effectiveness to prevent system constraint violations.

The LIME tool involves creating a grid of the parameter space by identifying two key parameters for each mission mode and discretizing between the good and bad cases identified in each of these scenarios. In order to best utilize this methodology, the span of the entire plausible parameter space should be examined, in order to map system performance across the design space. With the entire design space mapped, the LIME methodology focuses on an area of operation within the design space that most embodies realistic operating conditions. Selecting this local region for analysis of mitigation performance allows the designer to know whether the proposed mitigation provides some benefit for its designed purpose.

In the SL2 vehicle analysis a space of twenty-five total tests, representing a combination of the parameter range used in each mission mode, was chosen to illustrate the level of resolution to illustrate the early-stage design advantage this tool provides. The below tables Table 12, Table 13, and Table 14 encompass the parameter values used in each mission mode, as well as the increment denoting the grid resolution. It should be noted that the speed parameter is repeated in each mission mode to elicit vehicle reactions across the range of possible operating conditions for SL2 vehicles.

The values in Table 12, demonstrate the values used to describe initial conditions in mission mode 1. These parameters describe the initial along-track position of the following vehicle as well as its desire speed, as referenced to the lead vehicle. This range of vehicle parameters provides a reasonable span of potential vehicle characteristics for operation within the simulated environment. With these inputs, the difference in along-track position serves as the output variable of interest, where a violation of the along-track position threshold serves as the output for this mission mode.

To modify the general simulated for velocity control the two base vehicles were placed in the same lane, with the upstream vehicle being assigned a slower speed in order to allow for testing of the vehicle's voting algorithm. In this specific scenario the downstream vehicle's relative position and velocity were modified, reflective of the values of "delta speed" and "delta position" in Table 12.

Table 12: Mission Mode 1 Specific Parameter Values.

Parameter	**Starting Value**	**Ending Value**	**Increment**
Delta Speed	3 m/s	11 m/s	2 m/s
Delta Position	-28 m	3 m	-16 m

The values in Table 13, demonstrate the values used to describe the initial conditions in mission mode 1. These parameters describe the speed at which the SL2 vehicle operates and the turn delay in making the desired lane change (k^* in the sigmoid function shown in Equation 4). These values applied to the SL2 vehicle serve as the input to the second mission mode, where violations of the cross-track position threshold are the output variable of interest.

In order to create a simulated model of the loss scenario of the elective lane changes, the two vehicles were placed in outside lanes with a human desired lane change initiated at a fixed time. In this case, the vehicle of interest had a varied turn delay, as shown in Equation 10, impacted when this vehicle initiated a turn. Additionally, in this scenario the speed was also varied to determine if velocity had an impact on the performance of the vehicles.

Table 13: Mission Mode 2 Specific Parameter Values.

Parameter	**Starting Value**	**Ending Value**	**Increment**
Delta Speed	3 m/s	11 m/s	2 m/s
Turn Delay	1.3 s	0.1 s	0.3 s

The values in Table 14, demonstrate the values used to describe initial conditions in mission mode 3. These parameters describe the speed at which the vehicle approaches the road obstacle as well as the level of influence the automated vehicle has on the change in heading input the driver makes in an emergency situation. Given the nature of this mission mode, the output can be derived from the violation of along-track and cross-track thresholds, resulting in a vehicular collision.

Table 14: Mission Mode 3 Specific Parameter Values.

Parameter	**Starting Value**	**Ending Value**	**Increment**
Delta Speed	3 m/s	11 m/s	2 m/s
AV Weight	10%	50%	10%

Lastly, modifying the general simulation for the emergency mode switch example involved introducing a stationary point mass, indicative of an object or a vehicle, which has come to a stop in the lane. In this case the vehicle's velocity was modified, which provided varied time in making and executing the decision for the human driver to avoid hitting the object in the roadway.

3.3.5 Model Extensions

Mission mode 1, vehicle following and velocity control, occurs when the vehicle decides to maintain its desired speed when it has already violated the following threshold as established in SC1-1: "vehicle maintains a following distance of 20 meters." In the simulation this flawed information is generated due to sensor errors of the perception systems, which are operating at the bounds of 20% error, representing an overestimation of the distance between the two vehicles. In the simulation the perception system is made up of lidar, radar, and depth camera capabilities which provide simulated data for the distance between the vehicle of interest and other vehicles on the road. A visualization of this error and corresponding system constraint violation can be seen below in Figure 14. Frame 1 illustrates the multiple erroneous sensor readings, causing the vehicle to enter within the prescribed following threshold. In frame 2, The voting algorithm allows for a consensus to be reached, causing the vehicle of interest to reduce speed to maintain a safe following distance.

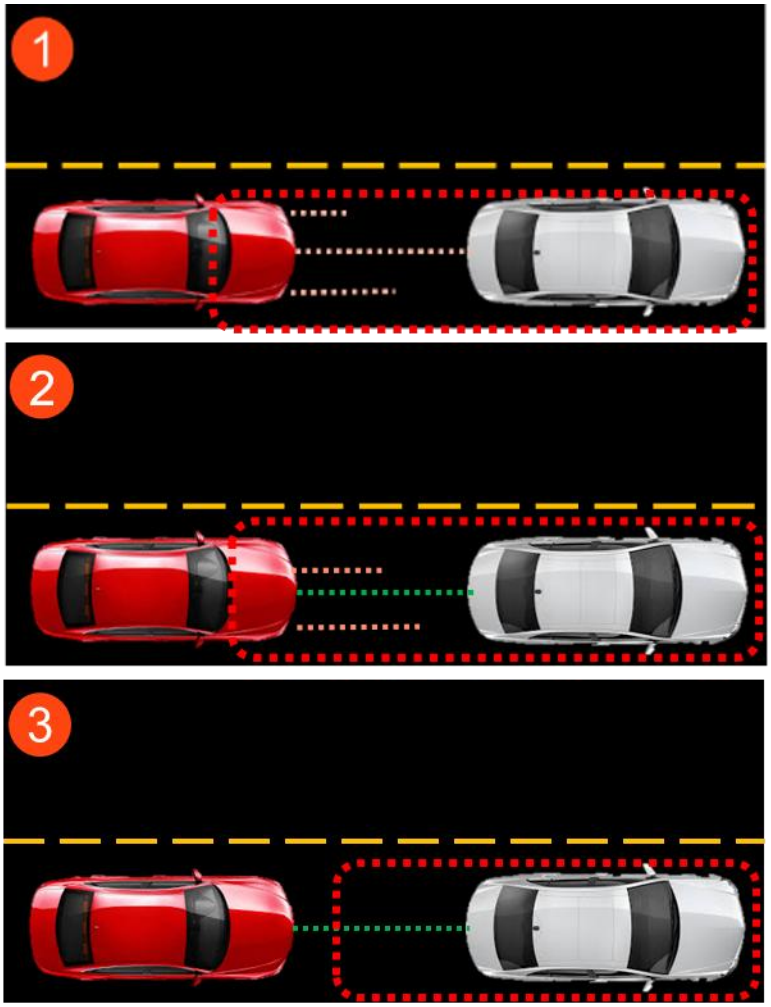

Figure 14: Visualization of Voting Mitigation.

To remedy this issue a voting algorithm was introduced to the simulation which takes the faulty sensor readings from and decides what measurement to use based on what value has consensus. In practice, when a sensor reading such as the lidar sensor provides an overestimation of the distance between the two vehicles, but the radar and stereo camera are within a similar range, the perception controller combines the two agreeing measurements and passes that information forward. In the case of complete disagreement between all sensor platforms, the vehicle reverts to its most trusted sensor platform, which in this simulation is radar. This algorithm allows impact from everyday sensor errors, due to sensor malfunction or weather

conditions to be decreased allowing the vehicle to operate within the bounds of its system safety constraints.

Mission mode 2, lane keeping and lateral control, focuses on a vehicle's ability to maintain its lane position while enforcing the SC5-1: "vehicle must maintain safe lateral distance of 20 meters from other vehicles, pedestrians, and objects." This mode of the simulation involves the execution of a s-curve lane change based off of the driver's input the system but fails when faulty localization information allows the vehicle to violate the system safety constraint. The trajectory of the s-curve lane change is given below in Equation 4, which denotes the sigmoid function in which the vehicle uses to change lanes [47, 48]. In this simulation the steepness of the curve was kept constant, but the value of the time delay was varied to represent latency and reaction difference of the drivers.

The below figure, Figure 15, depicts a violation of this system safety constraint while conducting a driver started lane change. This violation can be mitigated through the use of a model extension of an integrated vehicle to vehicle (V2V) network, which allows two vehicles to deconflict their intended completion of desired routing plans. Frame 2 illustrates the communication network in action, in which the vehicle of interest can interpret where the silver vehicle is intending to go and call off its planned lane change. As a result, the red vehicle waits until the lane is clear in order to complete the desired routing information, ensuring compliance with the system constraints generated earlier in the design process.

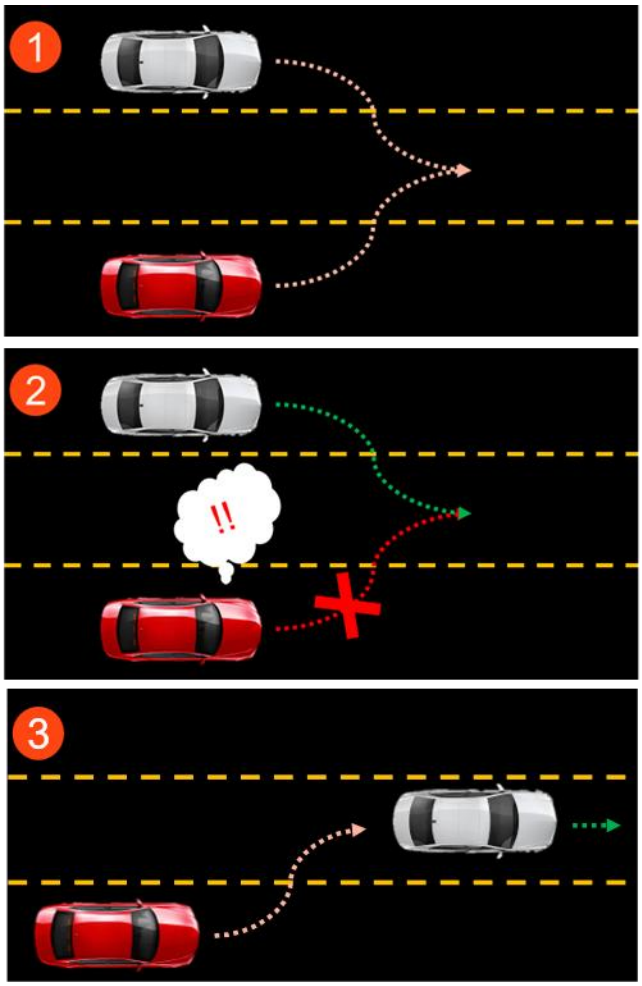

Figure 15: Visualization of V2V Mitigation.

This mitigation would allow vehicles to communicate not only where they are, but also where they intend to go. Within the simulation, a vehicle violates the system safety constraint when a vehicle enters within the same lane as another vehicle as illustrated in the above figure. Implementing the proposed V2V system allows for the deconfliction of path planning information and the abandonment of the lane change by the vehicle of interest. The decision to abandon a lane change is made by the vehicle who has completed less of the lane change and as a result has more ability to act. In practice this V2V network could act like a blind spot warning light among current vehicles, indicating drivers that another vehicle is intending to travel to a waypoint along their path to abandon their current trajectory.

In the last mission mode, focusing on emergency mode switching back to the driver given an unknown scenario, tests the vehicle's ability to maintain system safety constraints related to the traveling to unknown areas while implementing control measures to avoid a collision. Figure 16 illustrates the execution of an emergency mode switching controller to assist the driver in executing the necessary evasive maneuver. In frame 1, the vehicle notices an object in the roadway, while the driver is not paying full attention to the road. The vehicle reverts control to the driver, who due to their inattentiveness, does not have the time to properly execute a maneuver around the obstacle. This mitigation would provide haptic feedback in order to assist the driver into making the correct movement to swerve around the obstacle. As an amplification of driver response, this controller would allow the vehicle to avoid collision with the obstacle while ensuring that the driver is the executor of the control action.

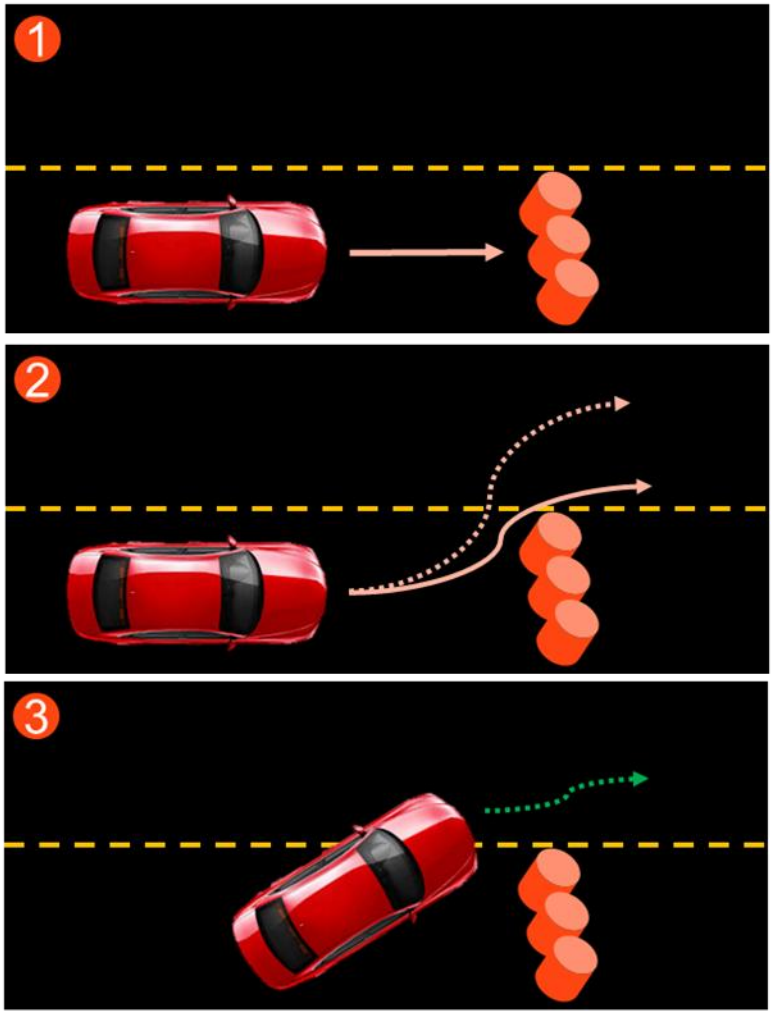

Figure 16: Visualization of Emergency Mode Switch Mitigation.

Integrating these mitigations allows for comparison of vehicle architecture states to determine if the imposed system-level changes assist the vehicle in carrying out the constraints and provide for a safer system.

3.4 Results and Discussion

This research aims to show the MBSE is a useful approach that can be used to meaningfully test the found mitigations through the STPA method to evaluate their effectiveness. Applying these mitigations and simulation framework over the given parameter space allows for

the performance of these mitigations to be tested to determine if they succeed at their mission of reducing the risk these vehicles operate with. Mitigations that succeed in this goal will reduce the number of system constraint violations, allowing the vehicle to perform its desired mission within set safety bounds. These simulations are essential to determine if the applied system constraints serve as realistic limitations for vehicle performance as well as ensuring the proposed mitigations reduce the number of occurrences of these violations without introducing any new faults. Simulations were conducted to test the three mission modes and corresponding mitigations, as outlined in section 3.3.3 Mode-Specific Parameters, and highlighted in the following subsections.

3.4.1 Mission Mode 1

The results of mission mode 1, illustrated in Figure 17, show that instituting the voting algorithm improves the vehicle's performance for a range of starting conditions. This mitigation provides the most improvement at lower values of $\delta\ velocity$ and larger values of $\delta\ distance$. This case represents the top left corner of the graph, in which the vehicle of interest starts 28 meters from the lead vehicle as compared to the bottom right in which the vehicle starts closer and moves much quicker than the vehicle to its front.

Each square on the plot represents a simulation outcome, in which if the system constraint is ever violated, the scenario is considered a loss. Given the initial conditions of the vehicle of interest, when a scenario violates the system constraint and enters within the threshold following distance, the outcome is considered a loss. When the voting mitigation is added and no system constraint is violated, the outcome reverts to a positive one: highlighting a positive improvement of vehicle function. This simulation also considered cases where adding a mitigation would provide a disadvantage by introducing hazardous scenarios, represented by a

black outcome, as well as scenarios where no system constraint is violated, represented by a gray outcome.

Examining the results in Figure 17, illustrates that the voting algorithm provides the most advantage in environments when the vehicle is given time to act and utilize perception information to execute safer driving practices. It should be noted that instances where the vehicle enters the threshold but quickly decides to slow at a speed are still counted as a hazard, even if no more potential threat is present. Refining the timestep used in this simulation and expanding the range of values used in the simulation would provide more fidelity in the response and potentially illustrate the robust nature of the voting algorithm to provide a distinct advantage in continuous time.

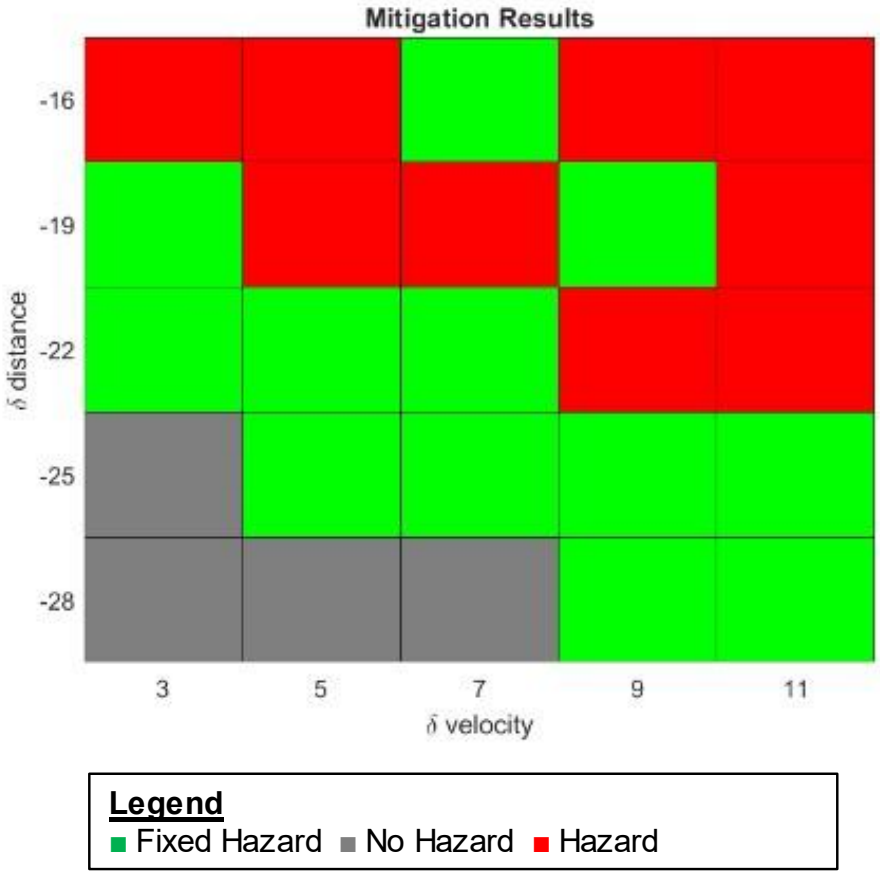

Figure 17: Results Mission Mode 1.

3.4.2 Mission Mode 2

Examining the results of mission mode 2 illustrates a broad effectiveness of improved vehicle communication networks regardless of initial conditions. A hazard is encountered when the lateral distance between vehicles violated the system constraint that two vehicles must not be in the same lane at the same time. These results, shown in Figure 18, demonstrate that when the lane change is instituted for low values of t_0 the vehicle performance improves, which results in a safer ride. These results are indicative of a timely lane change instituted by the driver allowing for deconfliction to occur. Improving this V2V communication network can give vehicles with more information (i.e., vehicles who may be able to see more of the roadway) the ability to abort a lane change to avoid violation of system constraints and prevent collision. This scenario, which

focuses on operator-introduced lane changes, allows for deconfliction of lane changing maneuvers but could also be used to provide more efficient routing information to deliver vehicles through traffic in a quicker manner through interconnected information flows.

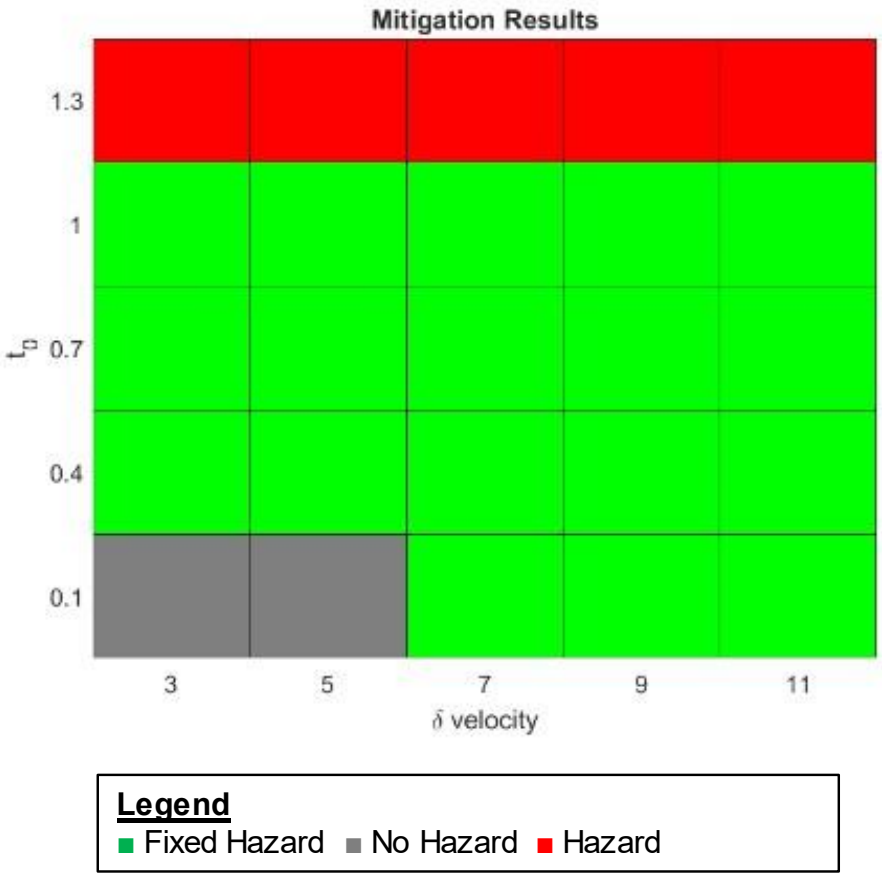

Figure 18: Results Mission Mode 2.

3.4.3 Mission Mode 3

The last mission mode, emergency mode switch controller, provides an assist in amplifying correct human action to prevent hazardous conditions. Increasing the weight of the automated vehicle's input provides a bias towards the correct action allowing the human driver to correctly steer the vehicle away from the potential collision. For this mission mode, a hazard is encountered when the heading change provided by the human driver is insufficient to clear the hazard in the roadway. This difference of heading input between the driver and the vehicle's

suggested pathing indicates that the human's input is insufficient to clear the obstacle, and a collision should be expected. Adding the mitigation of the emergency mode switch controller provides amplification of human performance and corrects heading inputs, while still allowing the human to maintain control. This controller provides input, but not overall control, to guide the human driver to safely avoid a collision with an object in the road.

The results in Figure 19 indicate that higher levels of automated vehicle weight in the decision-making process allow the driver to avoid hitting the obstacle, regardless of the driver's speed. While the driver still needs to initiate the transition to avoid the obstacle, having more input from the automated process allows for a sufficient heading change to avoid collision. This mission mode provides a basic example of the difficulties with transitioning control back to the driver in SL2 vehicles in emergency situations, and the corresponding mitigation serves as an introductory example for how this problem may be combatted in future automated vehicles.

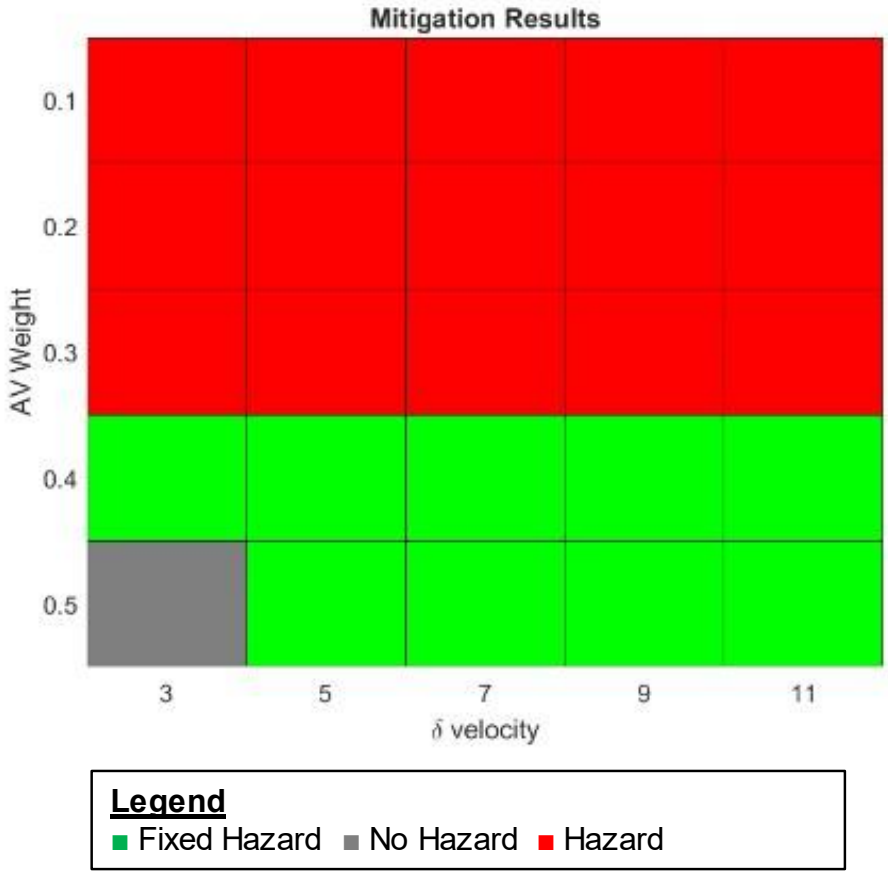

Figure 19: Results Mission Mode 3.

The mitigation strategies examined in this chapter present one potential solution to solve vehicle capabilities which are of interest in this analysis. While these simple simulations provide insight into basic vehicle performance in line with the system constraints, iterating on this analysis would provide more insight into other vehicle capabilities as well as performance.

This work represents a small subset of the potential mitigations and area of operations that vehicles face but provides a starting point for future analysis to continue to identify and analyze other areas of design flaws which can be improved to create a safer automated vehicle. Additionally, this analysis serves as a tool that can be used to identify and mitigate a wide range of undiscovered failure modes. Now that the STPA process is defined in Chapter 2, there are

more potential fault modes to explore and mitigate with the use of this tool to assess the value of future proposed mitigations.

3.5 Conclusion

Currently, STPA lacks a clear methodology to test design changes as they emerge early in the design process to determine if proposed mitigation strategies are worth continuing exploration and investment. While the field of MBSE allows for validation of system requirements which were generated during the STPA process in Chapter 2, the proposed LIME tool serves to augment the current STPA methodology by providing a form of rapid simulation-based testing of the effectiveness of design changes.

Utilizing a simple simulation of SL2 automated vehicles, the generated system requirements and corresponding mitigation approaches were tested on multi-lane roads with traffic traveling in the same direction. These simulations tested an automated vehicle's ability to execute the desired mission modes of velocity control, lane change, and emergency mode switching. The results indicate that the mitigation approaches derived to prevent the vehicle from entering hazardous conditions are capable of reducing the risk these vehicles are subject to within the local environment examined in this analysis.

Utilizing known edge cases of SL2 vehicle performance and mitigation strategies found in literature, this work serves as a proof of concept, highlighting that the identified solutions to these problems work as expected. The results of these mitigation strategies: voting algorithms, vehicle communication networks, and emergency mode switching controllers demonstrate the usefulness of this tool in illustrating the effectiveness of continuing to develop these mitigation strategies given their effect on creating a safer vehicle system.

Chapter 4: Conclusion

4.1 Summary

The modern day driving environment is inherently dangerous, with drivers subject to 1.26 fatalities per 100 million miles traveled by all vehicles [1]. Automated vehicles, specifically SL2 automated vehicles, aim to enhance the driver's functionality while keeping them in control, and serve as a potential tool that may be used to mediate the risk drivers face. Identifying and developing changes in automated vehicle design is essential to increasing the safety of these vehicles. While current automated vehicle design has improved some driver assistance features, more work is needed to ensure these capabilities match up with the ever-changing nature of the vehicle operating environment. Through rigorous design and testing, the future development of SL2 vehicles has the potential to save lives and reduce the reliance on human drivers acting perfectly.

4.2 Contribution

This thesis provided two main points of contribution:

(i) Application of System Theoretic Process Analysis (STPA) to the design of SL2 vehicles

(ii) Development of a simulation strategy to test risk mitigation strategies that emerge from STPA.

First, this work applied the hazard analysis technique to System Theoretic Process Analysis (STPA) to SL2 vehicles systems, an application which has not been previously executed. This application serves as a novel use of this process both in scope and outcome-deriving a series of three expected loss scenarios encapsulating three mission modes of velocity control, lane changing, and emergency mode switching. The three expected loss scenarios were

representative of known corner cases where SL2 vehicles fail to execute safe operating controls. To remedy these three scenarios, the STPA process was utilized to generate three mitigation approaches, representative of common design improvements seen in literature.

Second, these mitigation strategies were tested through simulation of vehicle performance, utilizing the developed Local Investigation of Mitigation Efficacy (LIME) tool to determine their effectiveness in reducing vehicle risk. This tool augments the STPA process as well as early design work by allowing engineers to evaluate suggested design change performance early in the design process to determine if further development is warranted. Developing a deterministic model of driving performance representative of the typical operating environments of these vehicles, allowed for each mitigation strategy to be tested in its intended use case. In each described mission mode: velocity control, elective lane change, and emergency mode switching; the proposed mitigation reduced the number of occurrences of system constraint violations, emphasizing the mitigations' ability to deliver a safer driving environment for the vehicle. The use of this simulation for each mission mode highlighted that these known loss scenarios and corresponding mitigation strategies, serve as a proof of concept for the usefulness of the LIME tool for use in assessing early design change effectiveness before implementation.

4.3 Future Work

Future work in this area of research should involve the application of this framework to other emerging areas of design defect, the expansion of the simulation aspect to encompass more realistic scenarios, as well as a metric to evaluate increased performance of the mitigations.

4.3.1 Future Design Improvements

With this thesis as a guideline, examining other emerging loss scenarios emerging from the SL2 vehicle, other than those expected from known vehicle outcomes, would provide key insight into currently unknown areas of design improvement. Iterating between simulation and the STPA methodology may help direct auto manufacturers find formerly unknown areas of potential faults through highlighting areas of previously unknown interaction between components and their control actions.

Expanding on the STPA analysis to turn some components treated as black boxes during this iteration would increase the applicability of the generated results. To accomplish the desired goal of analysis and simulation of the SL2 vehicle, certain systems such as localization systems or the path planning controller were abstracted for simplicity. While a limitation of this current research, returning the analysis to expand on these controllers and develop a control process model which reflects the true performance of these systems would embody a more representative approach to the vehicle system of interest and provide more accurate representations of hazardous outcomes emerging from unsafe interactions between these component control actions.

Additionally, future work should include consultation from subject matter experts. Similar to a FMEA, incorporating input from individuals with the most expertise on automated vehicles would provide key results more accurately, as they have the knowledge that comes with the experience of working with the system of interest for a long time. In a similar fashion, this analysis should be reproduced by other STPA professionals to ensure the accuracy of the results due to the novel nature of the scope of the analysis as the whole SL2 automated vehicle.

4.3.2 Advanced Simulation Outcomes

For the generated mitigation strategies to be effectively implemented within an automated vehicle, rigorous simulation and testing is essential before its incorporation. Advancing the scope of the simulation tools used to more accurately model driver and vehicle performance would accomplish this goal through providing a more realistic model of real-world driving scenarios. This research was limited in its consideration of constrained vehicle dynamics restricted to constant acceleration along concurrent straight lane roads. Adding vehicle features such as variable acceleration, curved roadways, and driving features such as stop lights or rotaries would provide increased complexity which would increase the known performance of the designed system mitigations. Vehicles operate in a multitude of operating conditions: urban environments, highspeed interstates, and unpaved rural roads; adapting the simulation to evaluate vehicle performance would drastically improve the ability to declare the efficacy of the suggested improvements.

Additionally, the simulation should be expanded to model other forms of vehicle faults, which were not included in this analysis as they did not fit within the selected mission modes of velocity control, lane changing, and emergency mode switching. Improving these features would allow for a more broadly applicable simulation capable of representing the expected areas of performance. This work notably focused its analysis on areas of known improvement (i.e., the mitigation approach is known to provide benefit to the system), further analysis should focus on cases where mitigation strategies may cause problems with the execution of the system task. Highlighting these “harmful” cases would verify that the LIME tool suggested in this methodology would confirm that the suggested changes do not provide advantage to the vehicle

system. Testing this negative case is still needed to validate that this approach to supplement the STPA process provides usefulness to engineers and vehicle designers.

4.3.3 Improved Performance Metrics

Adding an improved ability to evaluate the effectiveness of proposed mitigations would provide a key advantage to determine whether these design improvements should be implemented. Incorporating a cost vs effectiveness feature to this analysis would provide feedback to the design process, allowing designers to select combinations of improvements that would create the safest system with the lowest cost. This process would involve integrating the improved simulation capabilities and accounting for potential costs of incorporating these systems into future vehicles.

4.4 Impacts

In conclusion, this work has generated a framework for future analysis, while validating the approach to reduce the risk these vehicles are subject to. This process has the potential to impact the future development of both SL2 automated vehicles as well as a variety of other technologically dependent systems.

4.4.1 Focused Impacts

The novel application of this form of hazard analysis towards SL2 vehicles provides a baseline for the losses, hazards, control framework, and unsafe control actions found during the STPA process. This has an expressed impact on the future of automated vehicle design by illustrating the process in which future vehicle design changes can be implemented and tested to improve their respective systems. This approach coupled with the proposed improvements would provide fidelity in the results derived from their potential impact on the vehicle's performance.

Providing advanced insight into the potential performance advantages early in the design process would allow auto manufacturers to test design changes before implementation and the completion of the design phase. This change would allow for safety features that directly impact a driver's safety to directly target the areas of emerging failure that may not be commonly identified.

4.4.2 Broader Impacts

While outside the scope of this research, this process may also be applied to other systems to identify design flaws, determine potential remedies, and test those mitigations to design safer or more efficient systems. Extending the application of this framework would provide a tool that designers can use to iterate on the control structures that govern these systems to eliminate unsafe interactions between components. When utilized early in the design process, this tool has the potential to reduce costs by directing engineers to focus their attention on the most cost-effective strategies that can deliver results and reduce system downtime.

The potential applications are not limited to systems with incorporated levels of automation (although this process dissertation provides insights into how to think about these systems) but rather should be focused on systems with varying levels of complexity. The benefit of using this process provides a way to abstract complex systems operating in equally complex environments and identify areas of improvement, regardless of the system being analyzed. Some potential applications include other high complexity systems such as unmanned ariel vehicles (UAVs), aircraft, underwater vehicles, as well as cyber security systems.

Bibliography

1. Administration, N.H.S.T., *Early Estimates of Motor Vehicle Traffic Fatalties and Fatality Rate by Sub-Categories in 2023*, D.o. Transportation, Editor. 2024.
2. Schoettle, B. and M. Sivak, *A preliminary analysis of real-world crashes involving self-driving vehicles.* University of Michigan Transportation Research Institute, 2015.
3. Novat, N., et al., *A comparative study of collision types between automated and conventional vehicles using Bayesian probabilistic inferences.* Journal of Safety Research, 2023. **84**: p. 251-260.
4. Banks, V.A., K.L. Plant, and N.A. Stanton, *Driver error or designer error: Using the Perceptual Cycle Model to explore the circumstances surrounding the fatal Tesla crash on 7th May 2016.* Safety Science, 2018. **108**: p. 278-285.
5. International, S., *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, in *J3016_202104*. 2014.
6. Teoh, E.R., *What's in a name? Drivers' perceptions of the use of five SAE Level 2 driving automation systems.* Journal of safety research, 2020. **72**: p. 145-151.
7. Shahzad, A., *Comparison of STPA with FMEA for analyzing safety of autonomous driving system*. 2023.
8. Sabaliauskaite, G., L.S. Liew, and J. Cui, *Integrating autonomous vehicle safety and security analysis using STPA method and the six-step model.* International Journal on Advances in Security, 2018. **11**(1&2): p. 160-169.
9. Jianbo, H., Z. Lei, and X. Shukui, *Safety analysis of wheel brake system based on STAMP/STPA and Monte Carlo simulation.* Journal of Systems Engineering and Electronics, 2018. **29**(6): p. 1327-1339.
10. Scarinci, A., et al., *Requirement generation for highly integrated aircraft systems through STPA: an application.* Journal of Aerospace Information Systems, 2019. **16**(1): p. 9-21.
11. Dghaym, D., et al., *An STPA-based formal composition framework for trustworthy autonomous maritime systems.* Safety science, 2021. **136**: p. 105139.
12. Press, A., *11 more people killed in crashes involving automated-tech vehicles*, in *MoneyWatch*. 2022.
13. Reuters, *US investigates 2.4m Tesla self-driving vehicles after reported collisions*, in *Reuters*. 2024.
14. Leveson, N.G. and J.P. Thomas, *STPA handbook.* Cambridge, MA, USA, 2018.
15. Sulaman, S.M., et al., *Comparison of the FMEA and STPA safety analysis methods–a case study.* Software quality journal, 2019. **27**: p. 349-387.
16. Whitchurch, G.G. and L.L. Constantine, *Systems theory*, in *Sourcebook of family theories and methods: A contextual approach*. 1993, Springer. p. 325-355.
17. Leveson, N.G., *Engineering a safer world: Systems thinking applied to safety*. 2016: The MIT Press.
18. Gunaratnam, K., et al., *Hazard analysis techniques, methods and approaches: a review.* International Journal of Advanced Research in Engineering Innovation, 2022. **4**(1): p. 23-34.
19. Abdulkhaleq, A., et al., *A systematic approach based on STPA for developing a dependable architecture for fully automated driving vehicles.* Procedia Engineering, 2017. **179**: p. 41-51.

20. Leveson, N.G., *New Safety Technologies for the Automotive Industry*. 2006, SAE Technical Paper.
21. Chen, L., J. Jiao, and T. Zhao, *A novel hazard analysis and risk assessment approach for road vehicle functional safety through integrating STPA with FMEA.* Applied Sciences, 2020. **10**(21): p. 7400.
22. Sulaman, S.M., et al. *Hazard analysis of collision avoidance system using STPA*. in *International Conference on Information Systems for Crisis Response and Management (ISCRAM 2014)*. 2014. Pennsylvania State University.
23. Mahajan, H.S., T. Bradley, and S. Pasricha, *Application of systems theoretic process analysis to a lane keeping assist system.* Reliability Engineering & System Safety, 2017. **167**: p. 177-183.
24. Hommes, Q.V.E. *Safety analysis approaches for automotive electronic control systems*. in *Society of Automotive Engineers' Meeting*. 2015.
25. McDermott, R.E., R.J. Mikulak, and M.R. Beauregard, *FMEA.* New York: Taylor & Francis Group, 2009.
26. Lee, W.-S., et al., *Fault tree analysis, methods, and applications ✐ a review.* IEEE transactions on reliability, 1985. **34**(3): p. 194-203.
27. Chaal, M., et al., *A framework to model the STPA hierarchical control structure of an autonomous ship.* Safety Science, 2020. **132**: p. 104939.
28. De Souza, N.P., et al., *Extending STPA with STRIDE to identify cybersecurity loss scenarios.* Journal of Information Security and Applications, 2020. **55**: p. 102620.
29. Friedberg, I., et al., *STPA-SafeSec: Safety and security analysis for cyber-physical systems.* Journal of information security and applications, 2017. **34**: p. 183-196.
30. Gkoktsis, G. and L. Peters. *The Cyber Safe Position: An STPA for Safety, Security, and Resilience Co-Engineering Approach*. in *Proceedings of the 19th International Conference on Availability, Reliability and Security*. 2024.
31. Vaddi, S.S., et al. *Air-Ground Integrated Concept for Surface Conflict Detection and Resolution*. 2012.
32. Kramin, V., *Red car on white background*.
33. Fitzgerald, J. *Volkswagen's Killing Its Touch-Sensitive Steering-Wheel Controls. Goodbye and Good Riddance*. 2022.
34. Madni, A.M. and M. Sievers, *Model-based systems engineering: Motivation, current status, and research opportunities.* Systems Engineering, 2018. **21**(3): p. 172-190.
35. Dakwat, A.L. and E. Villani, *System safety assessment based on STPA and model checking.* Safety science, 2018. **109**: p. 130-143.
36. Abdulkhaleq, A., S. Wagner, and N. Leveson, *A Comprehensive Safety Engineering Approach for Software-Intensive Systems Based on STPA.* Procedia Engineering, 2015. **128**: p. 2-11.
37. Nejati, S., et al. *Evaluating model testing and model checking for finding requirements violations in Simulink models*. in *Proceedings of the 2019 27th acm joint meeting on european software engineering conference and symposium on the foundations of software engineering*. 2019.
38. Arcile, J., R. Devillers, and H. Klaudel, *VerifCar: a framework for modeling and model checking communicating autonomous vehicles.* Autonomous Agents and Multi-Agent Systems, 2019. **33**(3): p. 353-381.

39. Rokach, L., O. Maimon, and R. Arbel, *Selective voting—getting more for less in sensor fusion.* International Journal of Pattern Recognition and Artificial Intelligence, 2006. **20**(03): p. 329-350.
40. Parker, J.R. *Multiple sensors, voting methods, and target value analysis*. in *Signal Processing, Sensor Fusion, and Target Recognition VIII*. 1999. SPIE.
41. Harding, J., et al., *Vehicle-to-vehicle communications: readiness of V2V technology for application*. 2014, United States. National Highway Traffic Safety Administration.
42. Dey, K.C., et al., *Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication in a heterogeneous wireless network–Performance evaluation.* Transportation Research Part C: Emerging Technologies, 2016. **68**: p. 168-184.
43. Hu, L., et al., *The Challenges of Driving Mode Switching in Automated Vehicles: A Review.* IEEE Transactions on Vehicular Technology, 2024. **73**(2): p. 1777-1791.
44. Yan, Y., et al., *Discrete Multi-Objective Switching Topology Sliding Mode Control of Connected Autonomous Vehicles With Packet Loss.* IEEE Transactions on Intelligent Vehicles, 2023. **8**(4): p. 2926-2938.
45. Qu, T., et al., *Multi-mode switching-based model predictive control approach for longitudinal autonomous driving with acceleration estimation.* IET Intelligent Transport Systems, 2020. **14**(14): p. 2102-2112.
46. Martello, T., J.H. Rife, and H. Wassaf. *Proximity-Event Quantification for Navigating Automated Vehicles in Concurrent Traffic*. in *Proceedings of the 2024 International Technical Meeting of The Institute of Navigation*. 2024.
47. Huang, X., W. Zhang, and P. Li, *A Path Planning Method for Vehicle Overtaking Maneuver Using Sigmoid Functions.* IFAC-PapersOnLine, 2019. **52**(8): p. 422-427.
48. Chen, X., et al., *A Sigmoid-Based Car-Following Model to Improve Acceleration Stability in Traffic Oscillation and Following Failure in Free Flow.* IEEE Transactions on Intelligent Transportation Systems, 2024. **25**(8): p. 9039-9057.