APPLICATION OF STPA TO THE INTEGRATION OF MULTIPLE CONTROL SYSTEMS: A CASE STUDY AND NEW APPROACH

by

Matthew Seth Placke

B.S. Mechanical Engineering
North Carolina State University, 2012

SUBMITTED TO THE ENGINEERING SYSTEMS DIVISION IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN ENGINEERING SYSTEMS
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
JUNE 2014

Signature of the Author: ____
Engineering Systems Division
May 9, 2014

Certified by: ____
Nancy Leveson
Professor of Aeronautics and Astronautics and Engineering Systems
Thesis Co-Supervisor

Certified by: ____
John Thomas
Research Engineer, Department of Aeronautics and Astronautics
Thesis Co-Supervisor

Accepted by: ____
Richard Larson
Professor of Engineering Systems and Civil and Environmental Engineering
Chair, Engineering Systems Division Academic Committee

Page intentionally left blank.

APPLICATION OF STPA TO THE INTEGRATION OF MULTIPLE CONTROL SYSTEMS: A CASE STUDY AND NEW APPROACH

by
Matthew Seth Placke

B.S. Mechanical Engineering
North Carolina State University, 2012

Submitted to the Engineering Systems Division on May 9, 2014
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Engineering Systems

ABSTRACT

A new approach for analyzing multiple control systems within the STPA framework has been developed and demonstrated. The new approach meets the growing need of system engineers to analyze integrated control systems, that may or may not have been developed in a coordinated manner, and assess them for safety and performance. This need comes from the increasing proliferation of embedded control systems across domains including defense, energy, healthcare, automotive, aerospace, and consumer products. When multiple embedded control systems are integrated together, they have the potential to operate in uncoordinated and conflicting ways which might hinder their performance and lead to unsafe behavior.

This new approach provides a means for engineers to analyze the integration of control systems, beginning during concept development and continuing through the design process. The approach leverages the results of STPA Step 1 and guides the analyst in identifying instances of potential conflict between controllers. The method is demonstrated through a case study from the automotive domain, the integration of three driver assistance systems. The first application of the new approach identified instances of conflict amongst the three systems that would prohibit their successful operation in the field. Following the presentation of the case study, suggestions for future work and use in practice are provided.

Thesis Co-Supervisor: Nancy Leveson
Title: Professor of Aeronautics and Astronautics and Engineering Systems

Thesis Co-Supervisor: John Thomas
Title: Research Engineer, Department of Aeronautics and Astronautics

Page intentionally left blank.

ACKNOWLEDGEMENTS

I first thank my Lord and Savior Jesus Christ for His grace and provision throughout my life—He has blessed me beyond what I deserve and led me step-by-step in all seasons. I will continue to believe and rest in this promise from scripture:

> Proverbs 3:5-6 – "Trust in the LORD with all your heart, and do not lean on your own understanding. In all your ways acknowledge him, and he will make straight your paths."

I am deeply grateful to my parents, Eric and Dawn Placke, for how faithfully and enthusiastically they have supported me these last 24 years. I would not have been prepared for my time at NC State and MIT if not for their constant encouragement, sacrifice, and love. To my brother Sam, you have been a consistent source of encouragement, wisdom, and friendship—thank you.

To my wife Michelle, I love you very much! I never could have imagined that my time at MIT would include meeting and marrying you—what a blessing. Thank you for supporting me so graciously this past year; I look forward to spending the rest of my life with you!

I thank my advisor, Professor Nancy Leveson, for giving me the opportunity to come to MIT and explore new ways of thinking about engineering. Thank you for pushing me to grow and teaching me to challenge assumptions. I also thank Dr. John Thomas for his guidance throughout my thesis work and job search. To the rest of the group: Cody, John, Cameron, Dajiang, Adam, Soshi, Dan, Kip, and Aubrey – thanks for your encouragement, insightful feedback, and for making 33-407 a pleasant place to wrestle through MIT.

Page intentionally left blank.

Table of Contents

List of Figures

List of Tables

Page intentionally left blank.

1. Introduction

1.1. Motivation

Integrated software and hardware, linked through embedded controllers, are increasingly used across industries to both replace legacy physical systems and to introduce new functionality. Aircraft avionics, medical devices, automotive, industrial control and real time financial systems are just a few examples of industries that have seen a rapid increase in the presence of these systems. When multiple controllers are integrated into one system, they have the potential to interact in both the cyber and physical domains. Cyber interactions often arise when controllers use data from the same sensors or pass information on a common communication network. Physical interactions occur when controllers have authority over the same actuators or ability to influence the coupled dynamics of the system. The interaction of subsystems (embedded controllers are just one example) within a larger integrated system yields emergent properties that may be positive, such as enhanced functionality, or negative, such as degradations in safety. The ability to anticipate and manage controller interaction at an architectural level during concept development will enable system engineers to take advantage of positive couplings and design-out or mitigate negative ones.

Systems-Theoretic Process Analysis (STPA), a hazard analysis methodology developed by Nancy Leveson at MIT, is a technique that promises to meet this need. Since its inception, STPA has proven successful as a hazard analysis technique in domains ranging from nuclear power operation to space vehicle design [1]. This thesis applies STPA to a system with multiple embedded controllers to analyze their integration at an architectural level. The focus of the analysis will be functional safety and enabling safety driven design. Because of its rapid growth in the use of embedded systems, the automotive industry is presented as the application example.

1.2. Background

The automobiles we drive today are not just cosmetically different than those from the beginning of the twentieth century, and even those from the 1970's and 1980's; they have technologically evolved from mechanical and hydraulic machines to an elaborate electro-mechanical system with integrated control software. Software was initially used to replace mechanical functions in legacy systems such as Cruise Control and Anti-lock Brakes, bringing more flexible and precise operation. As their usefulness was validated and capabilities expanded, embedded controllers brought new functionality to the vehicle such as traction control and expanded entertainment systems. It is estimated that most consumer vehicles now have 60-80 microprocessors on-board that manage functions ranging from stability control to navigation guidance [2]. The complexity of the modern automobile arises from the tight coupling between components that is difficult to understand and manage, requiring expertise in domains raging from material science to human-computer interaction.

Because of this evolution, OEM system engineers must now manage requirements and development efforts across a broad range of technology suppliers. The nature of a complex system is that its apparent behavior emerges from the combined behavior of its components, which may themselves be systems. This holds true for automobiles as embedded systems become coupled when they are included together on the vehicle platform. OEM system engineers must produce subsystem requirements that will ensure not only stand-alone functionality, but continued functionality in the presence of other subsystems. Software-driven vehicle complexity has changed the nature of the design process and opened the design space such that upwards of 80% of car innovations are driven by computing systems [2]. Manufacturers are now often relying on suppliers to do specialized technology development while focusing their efforts on system integration [3].

This thesis analyzes the integration of cyber-physical systems, those that network complex embedded devices to control physical hardware components [4]. A common example of a cyber-physical system is Adaptive Cruise Control (ACC) that takes readings of physical data from sensors, processes them using software in an embedded controller, and then controls physical systems including propulsion and braking. Another cyber-physical subsystem that may control the brakes is the Anti-lock Braking System (ABS). Designed to prevent skidding by rapidly cycling brake pressure, the ABS system also takes data from sensors around the vehicle to be processed in its control software before actuating the brakes. These two systems, ACC and ABS, have the potential to interact in both the cyber (electronic) and physical domains.

In the cyber domain, ACC and ABS draw information from some of the same sensors and send electronic commands over the same network. In the physical domain, both controllers can actuate the brake system, which in turn directly affects the dynamics of the vehicle. These interactions may be positive, negative or inconsequential. A positive, complementary interaction occurs when ACC is slowing the vehicle down but is unable to do so quickly enough without skidding, so ABS takes over and pulses the brakes. A negative, conflicting interaction occurs when ACC is attempting to accelerate the vehicle but the ABS begins to pulse the brakes. This negative interaction will make the vehicle difficult to control and might lead to an accident. Ideally, opportunities for all types of interactions (positive, neutral, and negative) will be considered during system design. The presence of complementary interactions should be assured as their associated emergent behavior is often driving the value of the integrated system. Conflicting interactions, such as the scenario described above, should be designed out or mitigated to ensure functionality and safety of the final system.

This is not a trivial task for the System Engineers managing the development process and producing subsystem requirements. It is the requirements that guide the development of subsystems, so they must be both correct and complete to produce a fully functional, integrated system. Conflicts that are not accounted for during requirements development may be caught during system testing and validation. However, exhaustive testing is difficult for large electro-mechanical systems and practically impossible for software systems of any appreciable complexity [5]. Similarly, anticipated synergies (positive, complementary interactions) may not be realized and the associated emergent functionality not present. As integration and testing occur shortly before production, there is significant time pressure to mitigate issues and ensure functionality and safety. This ‘fire-fighting’ period is costly in terms of required labor and its impact on the project timeline. Adding to the time pressure, at this stage in the development process the potential solution space is often limited as various features of the design are frozen and engineers may be limited to adding-on fixes rather than changing inherent design flaws [6].

Even with considerable effort, some potential conflicts are not triggered in system testing and do not occur until operation in the field. Design flaws resulting in conflicts between embedded controllers is considered a possible cause of the rash of unintended acceleration events that were widely publicized in 2009 and 2010 [7]. In response to the accidents, many large recalls were issued and investigations undertaken by the National Highway Traffic Safety Administration (NHTSA) [5]. No specific electronic features were implicated in the accidents, but evidence of software errors leading to controller conflict would be difficult to find after-the-fact and thus the possibility has not been ruled out. In response to these events, NHTSA commissioned a report from the National Research Council titled *The Safety Promise and Challenge of Automotive Electronics*. The report notes that “electronics systems have become critical to the functioning of the modern automobile" and that the "challenge of the electronics intensive vehicle stems from the highly interactive nature of the electronic control systems." These observations led to the conclusion that "proliferating and increasingly interconnected electronics systems are creating opportunities to improve vehicle safety and reliability as well as demands for addressing new system safety and cybersecurity risks" [5].

These new risks are the motivation for this thesis, which will introduce an analysis method that enables engineers to identify potential conflicts between embedded controllers and consider them proactively rather than reactively. Doing so early in the development process at the architectural level, during concept development, will streamline the design process and improve the quality of fielded systems.

1.3. Objectives and Approach

The thesis is organized around four primary objectives as follows:

- **Assess the ability of current analysis methods to guide the integration of embedded controllers.**

  In the next section, Chapter 2, the author surveys common hazard analysis techniques used in the automotive industry and assesses their ability to analyze the integration of controllers during concept development. A comparison is made between the techniques and it is concluded that an analysis gap exists that STPA may be able to fill.

- **Apply STPA to an example set of embedded controllers and assess its efficacy.**

  Chapter 3 demonstrates how the traditional application of STPA may be used to meet this system engineering need by applying it to a set of three embedded control systems. Chapter 4 demonstrates how STPA Step 1 can be iterated with increased formality to generate executable requirements. The analysis focuses on the high level logic requirements and functional allocation of the independent systems. After the analysis is presented, the ability of the method to identify conflicting interactions is assessed and it is concluded that additional guidance would be beneficial. The details of the example systems have been fictionalized for this research and do not reflect the production intent of any particular manufacturer. The focus of this work is method development and not design or safety evaluation.

- **Apply a new approach to analyze the integration of controllers and identify conflicts.**

In response to the results of Chapters 3 and 4, Chapter 5 proposes a new approach that leverages the results of STPA Step 1 to analyze the integration of existing control systems. The new approach is motivated and then introduced in a general form. Application to the automotive case study is presented and initial results are provided, indicating the approach is a promising solution.

- **Assess progress and make recommendations for further development.**

Chapter 6 concludes the thesis by reflecting on the prior sections and suggesting a path for the future. The author assesses how well STPA and the new enhancement meet the analysis gap and what work is left to be done by listing the contributions of the thesis.

Page intentionally left blank.

2. Analysis Techniques

2.1. Introduction to Analysis Techniques

Chapter 1 described a problem currently being experienced in many industries, that the increasing scope and complexity of embedded control systems is making the design process very difficult. Specifically, as increasing numbers of control systems are integrated, it is very difficult to assess them holistically and make judgments about their performance with regards to functional safety. As the case study in this thesis comes from the automotive industry, this chapter will review several safety analysis methods commonly used by automotive engineers. Other methods may be used in practice; however, those reviewed here are employed most frequently and are often the foundation of other techniques used across many industries. Four analysis techniques are discussed briefly by introducing their basic concepts. Following their introduction, the techniques are compared and one is selected for further exploration and development.

2.2. Failure Modes and Effect Analysis

Failure Modes and Effect Analysis (FMEA) is an analysis technique that evaluates a system to identify possible failures of components (or subsystems) and asses the effects of those failures [8]. The FMEA technique was originally developed by reliability engineers working on military systems in the 1950's and has since been adopted and adapted by a variety of industries including defense, automotive, energy, and others [9].

There are many potential types of FMEAs, though they can generally be grouped into two categories which are distinguished by the type of reasoning used during the analysis [8]. The most common type of FMEA is referred to as hardware FMEA and employs inductive reasoning by starting from known failure modes and analyzing their effects. The second type of FMEA, a functional FMEA, relies on deductive reasoning to deduce which modes may lead to a particular failure. More variants may be defined by considering when the FMEA takes place and by differentiating whether the system is physically realized or is an engineered process. The FMEA is commonly extended to a Failure Modes, Effects, and Criticality Analysis (FMECA) by adding information about the probability and criticality of failure modes.

The analysis for both FMEA/FMECA variants follows similar steps, beginning by identifying and listing the parts of the system (components or subsystems) and their failure modes. Then, for each failure mode, the effects on other components and the effects on the overall system are recorded. The results of a FMEA/FMECA are generally recorded in a table such as that shown in Table 1 which has been adapted from the example in [8].

Table 1: Sample FMEA Worksheet

Component (or subsystem)	Failure Modes	Cause(s)	Effect(s)	Criticality
Hoist Motors	Inoperative	loss of power; defective circuitry; defective bearings	load cannot be raised or lowered	3
	Fails to Engage	broken springs; worn linings	load holding torque of motor brake will be lost; redundant motor brake with electric load brake and motor control will hold load	3

Once the FMEA/FMECA has been completed, the assembled information may be used to make decisions about the system. Failures that have a high criticality level and potential to significantly impact the system should be studied further and possibly eliminated from the system. Often the form is appended with additional columns that record the next steps as recommended by the analysis team. The FMEA/FMECA may be iterated as the design process progresses, different variants being used as appropriate.

The FMEA/FMECA technique can effectively identify single point failures in the system and characterize their consequences. The simplicity of the method is positive in that it can be applied to a wide variety of systems along most phases of the design process. However, the method is not well suited to assess multiple failure scenarios that may occur in complex, integrated systems [10]. Furthermore, the FMEA/FMECA technique's focus on failures prevents it from identifying scenarios that arise from the integration of systems in which no failure has occurred. As discussed in Chapter 1, conflicts may arise when correctly functioning systems interact if their individual requirements do not consider combined operation or are otherwise flawed. FMEA/FMECA's are only capable of identifying a subset of the conflict scenarios that may occur when integrating embedded control systems.

2.3. Fault Tree Analysis

In contrast to FMEAs, which are largely inductive, Fault Tree Analysis is a deductive method that identifies causes of undesired events. FTA was originally developed during the early 1960's at Bell Laboratories to evaluate the *Minuteman* missile launch system and then further developed by the Boeing Company to be both a quantitative and qualitative technique, but has since been adopted by many other industries as a part of their system safety process [9] [10]. Fundamentally, a FTA uses Boolean Logic to describe the combinations of causal events that lead to some particular undesired event, often referred to as a hazard. If qualitative, the output of the analysis is a listing of combinations of things (events, environmental factors, failures, errors, etc.) that will lead to a hazard. If quantitative, the FTA will produce a probability of the hazard's occurrence during a given time interval [9].

Every FTA begins by identifying the top event which may be broad, such as vehicle crash, or specific, such as tire blow-out. Next, causal events that may lead to the top event are identified and placed as branches underneath the top event. A standardized set of symbols is used to build the tree of faults and failures that are combined using AND gates and OR gates. An example fault tree is given in Figure 1.

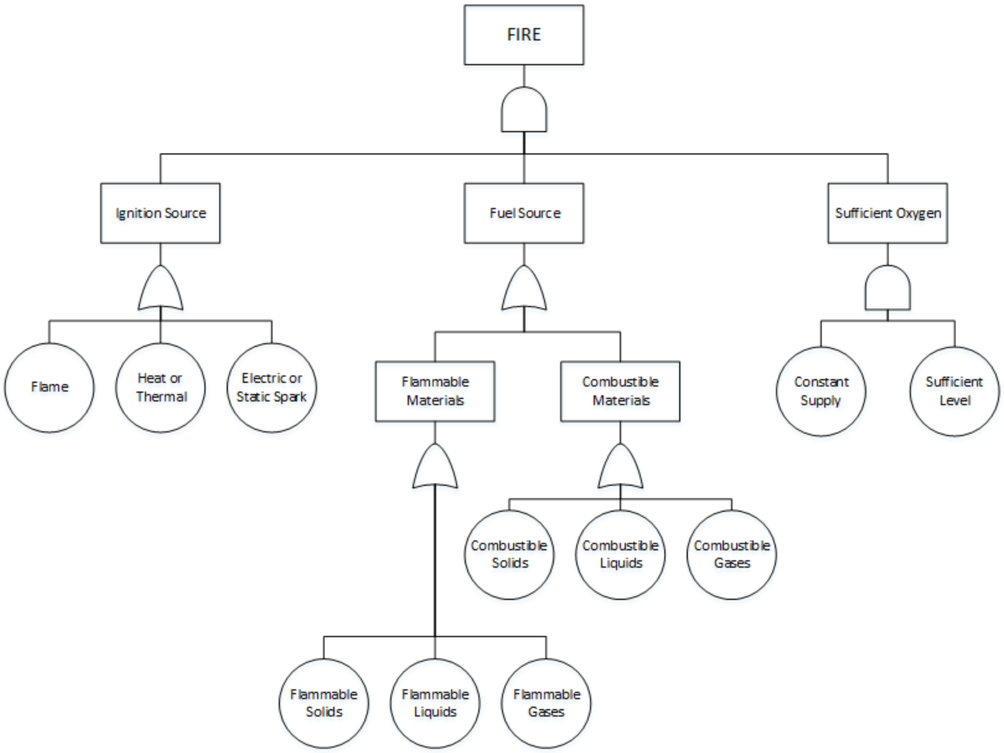


Figure 1: Example Fault Tree [7]

After construction of the fault tree, a logically equivalent form relating the basic events to the top event is developed, referred to as a cut set. If probabilities are assigned to the basic events, the likelihood of a cut set occurring may be calculated using probability theory. Cut sets are used to further focus the analysis and help analysts prioritize high level causes of a top event for further study and potential mitigation.

FTA is a top down technique, in that it starts by identifying a single undesired event and then works to levels of greater detail by identifying causal events. This is a powerful attribute that prevents portions of the system from being completely unexplored. However, FTA provides little guidance as to the analyst throughout the analysis. Although it is a powerful technique for analyzing the electromechanical systems prevalent during its development, FTA is not particularly capable of analyzing systems with significant software such as embedded control systems. Software does not fail like physical components and FTA does not provide a means to analyze flawed requirements that may lead to undesired software behavior [6]. Furthermore, the construction of a Fault Tree requires some appreciable level of design detail and thus FTA during concept development is unlikely to produce meaningful results [10]. These limitations prevent FTA from effectively addressing the problems introduced in Chapter 1.

2.4. System Theoretic Process Analysis

Systems Theoretic Process Analysis (STPA) is a hazard analysis technique introduced by Leveson [6]. STPA differs fundamentally from FMEA and FTA in that is rooted in Systems Theory rather than Reliability Theory, specifically the Systems Theoretic Accident Model and Processes (STAMP) [11]. STAMP focuses on the emergent properties of engineered systems rather than the reliability of individual components and thus is an effective means for holistically analyzing complex systems.

STPA begins by defining the system of interest and analyzing accidents that the system should not experience. Accidents are considered to be unacceptable losses. Hazards are system states that may lead to an accident. The system is modeled in terms of a functional control structure comprised of hierarchical control loops which may have both social and technical components and be represented at varying levels of abstraction. Figure 2 is an example of a simple, generic system control structure in which some process is controlled by automated and human controllers who act on a process through actuators and receive feedback information through sensors.

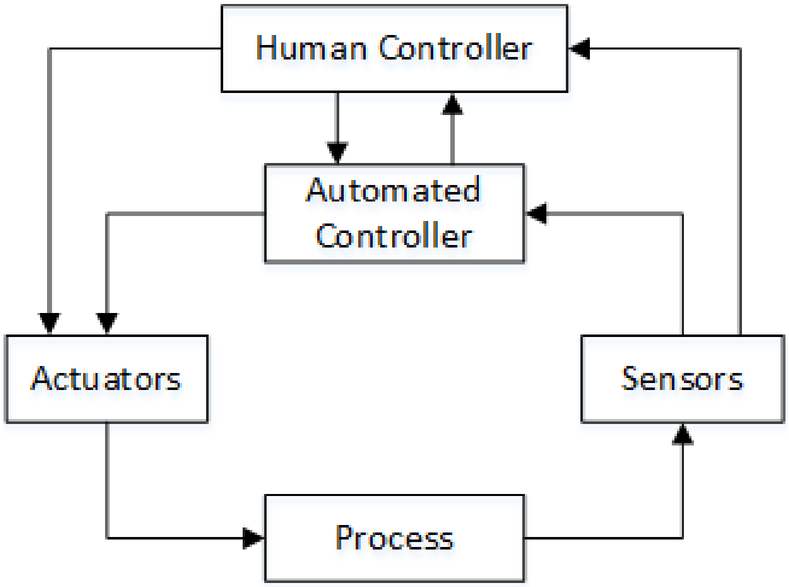


Figure 2: Generic Control Loop

After this initial preparation, the first step of STPA is to identify when the controllers may issue control actions, realized by actuators, in a manner that is unsafe or otherwise inappropriate. There are four ways in which a control action may be hazardous:

- A control action required for safety is not issued.
- A control action that is issued when it is unsafe to do so.
- A control action is issued too soon or too late.
- A continuous control action is applied too long or stopped to soon.

Instances when these four types of Unsafe Control Action (UCA) may occur are often recorded in a table for easy reference. Once they have been identified, UCA may be translated into safety constraints – high level constraints on system behavior that must be enforced to ensure an accident does not occur.

STPA then moves to a second step which identifies reasons that a UCA may be issued and ways in which a correct control action may be issued but not executed properly. These two types of causal factors are both important and require some diligence to identify. A control loop with guidewords to prompt the analysts, such as that shown in Figure 3, is used to ensure a thorough analysis.

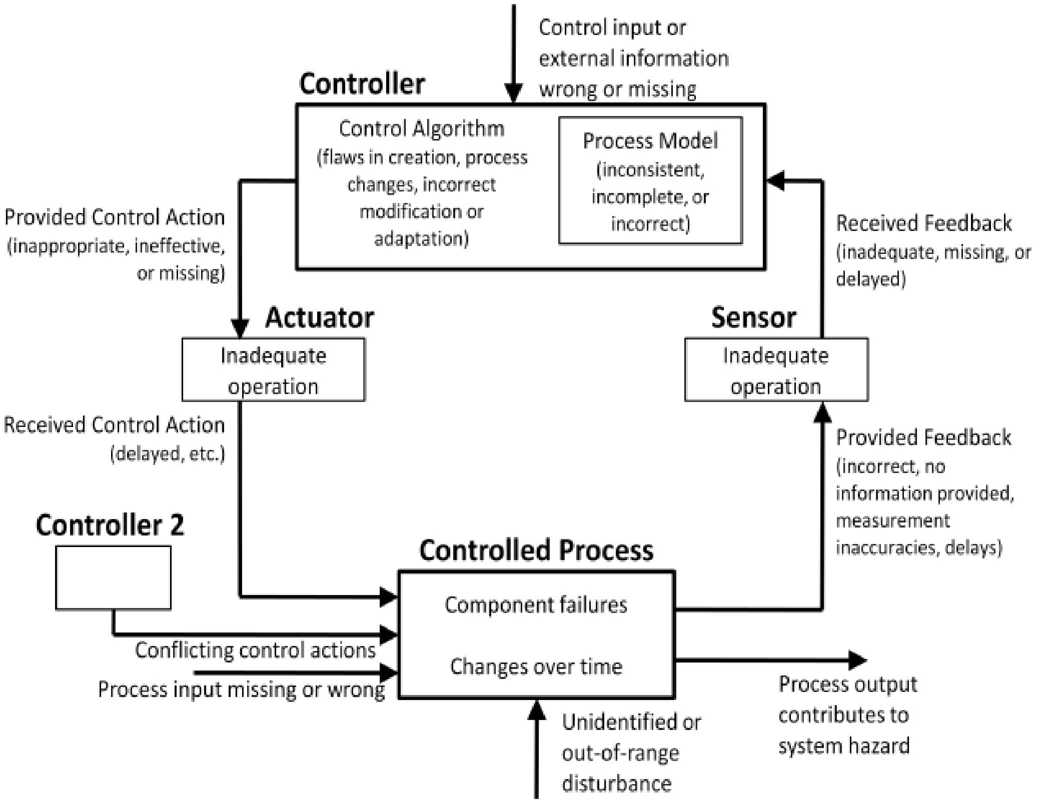

Control input or external information wrong or missing
Controller
Control Algorithm (flaws in creation, process changes, incorrect modification or adaptation)
Process Model (inconsistent, incomplete, or incorrect)
Provided Control Action (inappropriate, ineffective, or missing)
Received Feedback (inadequate, missing, or delayed)
Actuator
Inadequate operation
Sensor
Inadequate operation
Received Control Action (delayed, etc.)
Provided Feedback (incorrect, no information provided, measurement inaccuracies, delays)
Controller 2
Controlled Process
Component failures
Changes over time
Conflicting control actions
Process input missing or wrong
Unidentified or out-of-range disturbance
Process output contributes to system hazard

Figure 3: STPA Step 2 Guide Loop [6]

Once identified, the causal factors leading to accidents may be used to write requirements for the components of the system. Alternatively, if the causal factor concerns the behavior of a subsystem, steps 1 and 2 may be iterated to analyze the subsystem in greater depth. The results of the causal factor analysis may be used to eliminate hazardous scenarios or, if not elimination is not possible, reduce their occurrence.

STPA is a top-down methodology capable of analyzing complex systems and capturing interactive effects between subsystems. Significant guidance is provided throughout the process to ensure the system in question is analyzed in a methodical manner. Because the method can be used at many levels of abstraction, STPA is useful beginning at the concept development stage and through the entire design process.

2.5. Choosing an Analysis Technique

It is necessary to choose an analysis technique that is both methodical and flexible. The challenges outlined in Chapter 1 are not confined to one domain, so an analysis technique must not be constrained by disciplinary or application boundaries. Additionally, the analysis technique must be applicable during concept development to prevent wasted design effort and promote top-down, safety-driven design. The potential analysis techniques are compared in Table 2 using a set of criteria that measure their potential to analyze a set of complex embedded control systems.

Table 2: Analysis Technique Selection Criteria

Criteria	FMEA/FMECA	FTA	STPA
Top-down, deductive reasoning	no	yes	yes
Systematic and guided analysis process	no	no	yes
Meaningful results during concept development	yes	no	yes
Identifies non-failure related events	no	no	yes
Intended to analyze interaction of multiple systems	no	no	yes
Can handle software requirements effectively	no	no	yes

STPA is selected for further exploration because it has greater potential, in comparison to the other three techniques, to meet the needs of system engineers seeking to integrate complex control systems. Chapters 3 and 4 describe the application of STPA to an automotive case study.

Page intentionally left blank.

3. Initial STPA Case Study

3.1. Overview

This chapter presents the traditional application of STPA to three automotive subsystems. The subsystems will be referred to as features, a term often used in the industry. A feature is a subsystem, or collection of subsystems, that together provide a unified functionality from the driver's perspective. Anti-Lock Brakes is an example of a feature that incorporates multiple components (sensors, valves, controller, etc.) to provide a function—non-skid braking. The term feature will be used because the classification as either systems or subsystems is context dependent. When talking about an individual feature, it is appropriate to refer to it as a system that has subsystems. When considering the integration of these features into the larger automobile system they are appropriately subsystems (which still contain subsystems). One can see that a perspective-neutral term will be useful to avoid confusion.

All three features in this case study rely on embedded controllers to control components of the vehicle's brake and propulsion systems. The features described are representative of those currently being developed and offered by automotive manufacturers; however, the analysis presented here is not meant to represent or critique a specific manufacturer's system. The details of the example features are fictional, created for the purposes of this research work. The goal of this work is to show how STPA can be used to guide the design of a system beginning during concept development. To accomplish this, the analysis is begun at a high level of abstraction when specific implementation details are not known and may be influenced by the outcomes of the first analysis iteration.

While the analysis is presented straight through, some iteration did occur during its development. When this is the case, comments will be provided about the analysis' development and how it could be iterated further to greater levels of detail. The features are briefly described before the background common to all three is presented, including a template for their control structures, accidents and hazards. After the common foundation is laid, STPA Step 1 is presented for each feature, followed by a selected portion of STPA Step 2.

It is recognized that analyzing these features individually, at least initially, is not ideal and does not align with the top-down methodology put forth by System Engineering. Ideally, one would start with the highest level of system goals that, through decomposition, would eventually branch into features to be developed by (potentially) separate engineering teams. In this case, the flow down of functionality and requirements would prevent many of the calamities that results from integrating systems designed in an uncoordinated manner. Analyzing features individually does not allow the analyst to make conclusions regarding the behavior of the integrated system.

However, it is often the case in industry that an engineer (or engineering team) is responsible for integrating legacy systems that were designed separately. It is often not practical to start with a blank sheet of paper and develop an integrated system from scratch. The industry may be moving so fast that a company would lose its competitive advantage or the cost of doing so may be significantly greater than the available budget. In addition, systems that were developed using System Engineering methods eventually need to be modified, updated and/or upgraded. These situations require an analysis method that can analyze the system as is and guide the analyst to make conclusions regarding the whole system.

This chapter presents the analysis of the three features and then comments on what is required to responsibly analyze them as an integrated system, re-aligning us with the top-down methodology of System Engineering. Developing an approach to analyze these existing systems as an integrated one is the purpose and subject of Chapter 5.

3.2. Features

The three features are briefly introduced here; more detailed descriptions accompany the presentation of each feature’s control structure. Each of the features is implemented as a software upgrade that implements existing vehicle hardware. These fictional descriptions, and additional details that follow in later sections, have been created by a team of researchers at MIT for research purposes and do not reflect the production intent of any manufacturers.

3.2.1. Auto-Hold

Auto-Hold (AH) is an automatic braking feature that holds the vehicle and prevents rollback, allowing the driver to remove his foot from the brake pedal when the vehicle is at a standstill. When the vehicle is brought to rest using the brakes, the AH feature will maintain the necessary brake pressure to keep the vehicle from moving by capturing the pressure already in the system. Once sufficient wheel torque (other than idle torque) is supplied to move the vehicle forward, the feature will disengage and release the braking system back to its normal state. The AH feature uses existing hardware components of the braking system including the Anti-Lock Braking System (ABS) and Electronic Parking Brake (EPB).

3.2.2. Engine Stop-Start

Engine Stop-Start (SS) is a feature designed to reduce fuel consumption and emissions by turning off an engine that would otherwise be idling while the vehicle is stopped. When the vehicle comes to a complete stop, the engine is automatically turned off and then restarted before motion resumes.

3.2.3. Adaptive Cruise Control with Stop & Go

Adaptive Cruise Control w/Stop-Go (Stop-Go or SG) is an enhanced version of the legacy cruise control feature. With traditional cruise control, a driver may pre-set a speed for the vehicle to maintain, allowing him to remove his foot from the accelerator pedal. Traditional cruise-control systems allow the driver to set the speed, increment the set speed up or down, temporarily increase speed using the accelerator pedal (such as when passing), and disengage the feature using the driver controls or the brake pedal.

Adaptive Cruise Control (ACC) builds upon this traditional architecture in that it intelligently considers the distance to vehicles and other objects ahead of the primary vehicle in the same lane. Radar is mounted in the front of the vehicle that reports the range and range-rate to a controller that may utilize the braking and propulsion systems to maintain a safe trailing distance[1]. As with speed, the driver can change the desired trailing gap through the feature controls.

The Stop-Go capability further builds upon this architecture, enabling the ACC system to bring the vehicle to a full stop and resume motion when following a target vehicle. This is intended to be used in stop-and-go traffic when the vehicle may momentarily come to a stop before moving forward in concert with the surrounding vehicles.

[1] Trailing gap may be implemented as a time-gap to the leading vehicle so that it can vary dynamically with speed.

3.3. System Engineering Foundation

STPA should be integrated into a project's system engineering efforts, beginning with the identification of accidents and hazards. Following this and unique to STPA, a functional control structure of the system is developed on which to base the analysis.

3.3.1. Scope and Assumptions

The subject of this analysis is the three embedded electronic features described above: Auto-Hold, Engine Stop-Start, and Adaptive Cruise Control with Stop-Go. The goal of this thesis is to analyze these three features while developing a generalized method for controller integration. Not all aspects of the vehicle are analyzed, and as such, the following assumptions and bounds are used to focus the analysis:

- **Longitudinal Motion** - These three feature systems are used to control the vehicle's longitudinal motion and are not able to directly influence the steering system[2]. This analysis will not encompass the vehicle's steering system except to note when the feature may hinder its operation.
- **Automatic Transmission** – It is assumed that these features are being implemented on a vehicle with an automatic transmission.
- **Electronic Parking Brake** – It is assumed that the features are being implemented on a vehicle that has an Electronic Parking Brake (EPB). Many automotive manufacturers are offering EPB and it is trending to become an industry standard. Including an EPB means introducing additional control logic into the vehicle that can directly affect its longitudinal motion.
- **Hydraulic Service Brake System** – It is assumed that the features are being implemented on a vehicle that uses a hydraulic service brake system.

It should also be noted that this analysis is conducted at a high-level of abstraction, early in the design process of these features. The analysis is not meant to represent a specific manufacturer or supplier's implementation of the feature concepts.

[2] Engine Stop-Start may indirectly affect the steering system by cutting power to the hydraulic pump that provides pressure for power-steering. If this occurs, the driver will still be able to move the wheels but it will require greater effort.

3.3.2. System-level Accidents

An **accident** is any undesired or unplanned event that results in a loss [6]. What is considered a loss is subjective and up to the system stakeholders to identify. Common losses to be avoided include human life, human injury, property damage, and pollution. This idea may be extended to include loss of functionality such that mission loss or system malfunction may also be considered accidents. Four accidents relevant to automobiles are listed below with examples and brief descriptions. This set of accidents encompasses loss of life, injury and property damage. This set of four accidents captures the losses of concern in most automotive contexts; however, a few known exceptions and omissions are discussed after the presentation of accidents.

A-1: Two or more vehicles collide

Example: A trailing vehicle rear-ends a leading vehicle in city traffic.

Description: This accident captures any collision between two or more vehicles, which may occur with or without the other accidents.

A-2: Vehicle Collides with non-fixed (mobile) obstacle[3]

Example: A vehicle collides with a pedestrian or biker.

Description: Non-fixed obstacle is meant to capture entities other than vehicles that are mobile and capable of controlling movement. This includes, but is not limited to: pedestrians, bikers, animals, etc. This is distinct from terrain, because the capability to move presents a different challenge and requires fundamentally different strategies such as anticipating future movement, tracking current position, and negotiating right of way.

A-3: Vehicle crashes into terrain[4] (fixed obstacle)

Example: A vehicle traveling through an underpass collides with the structural supports of a bridge.

Description: Terrain is meant to capture all fixed obstacles that a vehicle may interact with. This includes, but is not limited to: guard rails, vegetation, bridges, signage, etc.

[3] Other obstacle' includes pedestrians, bikers, animals, etc.

[4] 'Terrain' includes fixed, permanent objects such as guard rails, trees, bridges, signage, pavement, etc.

Terrain captures a class of obstacles that are not capable of interacting with the vehicle and thus it is the sole responsibility of the vehicle to avoid a collision.

A-4: Vehicle occupants injured without vehicle collision

Examples: Excessive deceleration causes whiplash. Subsystem misuse or failure results in a vehicle fire without a collision.

Description: This is meant to capture scenarios in which the health or well-being of the vehicle's occupants is compromised without a collision.

Table 3 summarizes the accidents discussed above. These accidents provided the basis for the hazard identification that is presented next section

Table 3: System Level Accidents

Accident	Description
A-1	Two or more vehicles collide
A-2	Vehicle collides with non-fixed obstacle
A-3	Vehicle crashes into terrain
A-4	Vehicle occupants injured without vehicle collision

Not captured by this list of accidents are scenarios that include: vehicle maintenance, performance driving, and off-road driving. For example, a mechanic reaching under the vehicle to change the oil but being injured by moving components is not an accident captured by the list in Table 3. Such scenarios require that the individual have additional training or experience that would not be expected of the average driver interfacing with the features to be analyzed. Additionally, each of these three omitted scenarios is far from the intended usage of the average automobile for transporting people and cargo. It is expected that the individual involved will consciously weigh his experience and expertise against the potential risk of the activity. It is possible to conduct a hazard analysis to encompass these situations by changing the assumptions and defining a new set of accidents.

3.3.3. System-level Hazards

The next step after identifying the unacceptable losses and associated accidents is to identify the system level hazards. A **hazard** is a system state, or set of conditions, that may lead to an accident (loss) [6]. An important distinction between accidents and hazards is that accidents may involve factors outside the system boundary (i.e. outside of the designer's control) while hazards should focus on what is within the system boundary and in the available design space. For example, icy road conditions during winter driving cannot be controlled by the engineers designing tire treads; they may lead to A-1, 2, or 3 but are outside the available design space. The engineering effort must focus on preventing hazards that are within the available design space. Continuing the example, while the road conditions cannot be controlled, the performance of the tires' materials in cold weather and on low friction surfaces are within the design space and should be considered diligently to prevent poor performance which may lead to an accident. Provided below are examples and brief descriptions of the hazardous states that may lead the accidents identified previously:

H-1: Vehicle does not maintain safe distance[5] from nearby vehicles

Examples: Vehicle stops suddenly with another vehicle behind, accelerates into vehicle in front, vehicle rolls backwards towards another vehicle, vehicle changes lanes when adjacent space is not available.

Description: This hazard describes vehicles that become too close to each other, and it may be broken down into sub-hazards based upon direction. The three logical subdivisions are longitude, latitude and vertical. The decision to divide the hazard or leave it as a comprehensive set is up to the analyst and can be revised as needed to aid the analysis process.

H-2: Vehicle does not maintain safe distance from terrain and other obstacles

Examples: Vehicle experiences a near miss with a pedestrian, animal, biker, bridge, etc…

Description: This hazard is similar to H-1 but refers to vehicles that are too close to non-vehicular objects.

[5] Safe distance is potentially subjective and a function of many things including: vehicle speed, obstacle speed, vehicle braking and acceleration capacity, road conditions, etc. Thus it is dependent on the driver, environment, traffic system, and design of the vehicle.

H-3: Vehicle enters uncontrollable/unrecoverable state

Examples: Driving too fast, turning too much for a given speed, going too fast over a speed bumps, too fast for weather conditions, unable to stop, unable to go, driver's commands not heeded, loss of vehicle system such as braking or power steering, etc

Description: This hazard captures situations in which the driver may be unable to use the vehicle's systems to gain control. There are many potential ways this hazard could potentially arise, including through actions of the driver, vehicle automation behavior, design flaws, flawed requirements, outside influences, etc.

H-4: Vehicle occupants exposed to harmful effects and/or health hazards

Examples: Vehicle or engine fire, excessive temperature (occupant heat exhaustion, burns from hot surfaces), smoke, excessive deceleration (seatbelt rash) or acceleration, loud noise, sharp edges, too fast over rough terrain, etc.

Description: This hazard captures additional problems that may occur during nominal dynamic-operation of the vehicle and its subsystems. It is possible for the examples listed above to occur while the vehicle is operating appropriately in the larger traffic system; however the occupants of the vehicle may still be at risk for injury.

Table 4 summarizes the hazards that are described above. These hazards are used to guide the rest of the analysis and provide a foundation for traceability in the scenarios that will result.

Table 4: System Level Hazards

Hazard	Description	Accident
H-1	Vehicle does not maintain safe distance from nearby vehicles	A-1
H-2	Vehicle does not maintain safe distance from terrain and other obstacles	A-2, A-3
H-3	Vehicle enters uncontrollable/unrecoverable state	A-1, A-2, A-3, A-4
H-4	Vehicle occupants exposed to harmful effects and/or health hazards	A-4

3.3.4. Control Structure Template

Identifying accidents and hazards is standard practice in system-safety analyses; the additional foundational piece required to perform STPA is a system control structure. The system control structure is a functional representation of the system as hierarchical control loops. Depicting the system with a functional control structure provides a means for representing many pieces of information in a simple, graphical manner that can be understood by analyst from different disciplines. The same information regarding system design and operation may be embedded in engineering documents such as wiring diagrams, pseudocode, or requirements documentation; however, the basic functional information may be hidden behind the design details of the relevant discipline. The control structure consists of functional blocks connected by arrows that represent control actions and feedback. When making the control structure, the analyst is assigning responsibilities to each functional block and connecting them in a hierarchy. Doing so in a neutral (discipline non-specific) manner ensures that all system stakeholders have a common understanding of the system's design and operation.

The control structures used in this analysis have a common format that stems from similar architectures of the three features. As has been previously mentioned, each feature is a cyber-physical system in which multiple hardware components are managed by an embedded controller. From a functional standpoint, the common architecture has three basic hierarchical levels: the human driver, the software module, and the physical vehicle with its subsystems. The same driver-software-vehicle system can be represented in much greater detail with many more levels of hierarchy, though for a high level analysis this level of abstraction is appropriate. In a safety guided design effort, each of the functional blocks and their links may be expanded to greater detail as design spaces are narrowed and implementation decisions made. Figure 4 below gives a template for the control structures that are a part of this analysis.

The blocks in Figure 4 represent functional entities that may or may not correspond to physical entities. The convention used here is that blocks higher on the structure have authority over blocks lower on the structure and that control actions flow down while feedback flows up and to the side. For visual simplicity, only one arrow in a given direction is shown between two blocks—so an arrow represents a path rather than an individual command or piece of feedback. Multiple terms labeling an arrow represents the set of control actions or feedback variables (depending on direction) that may exist on the path indicated by the arrow. In this template, the braking and propulsion systems are considered subsystems of the physical vehicle and thus are represented a solid blocks with a larger, dotted box. This construct, solid boxes grouped in a dotted box, is used to indicate that the relationships of these subsystems to the vehicle and its dynamics are different than the purely logical relationship between the software module and the physical subsystems.

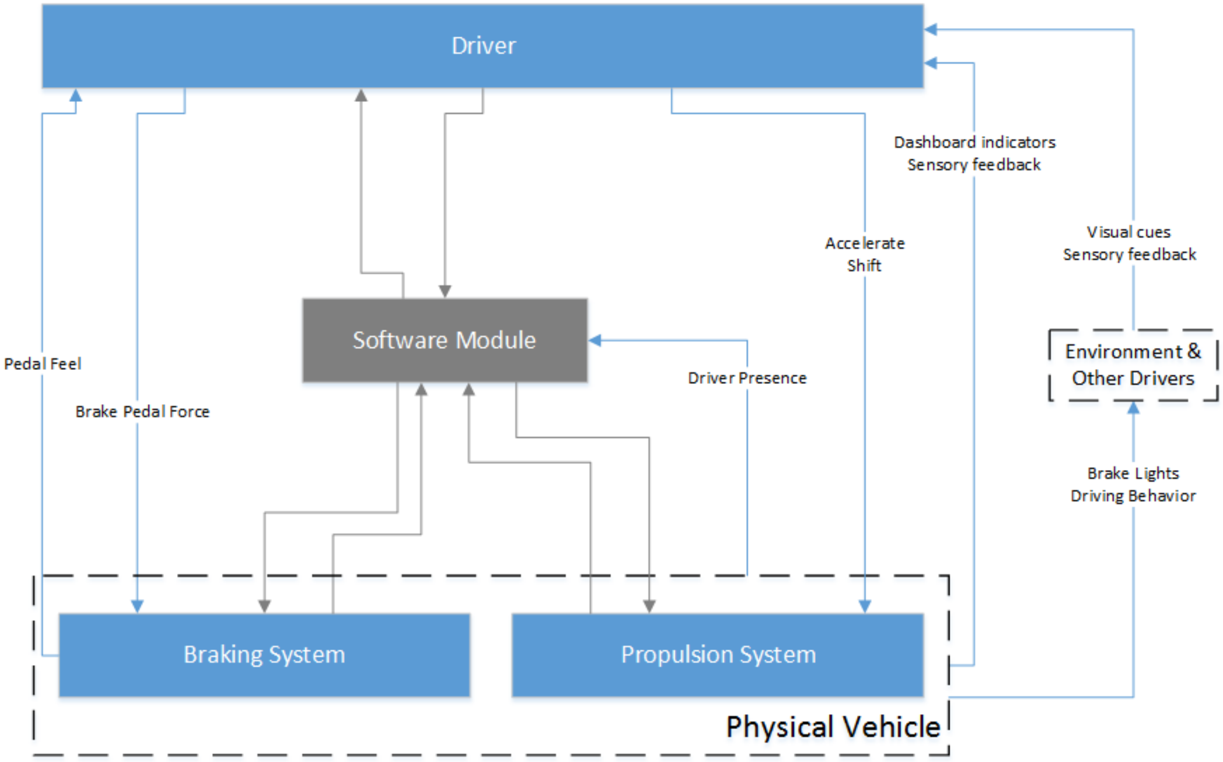


Figure 4: Control Structure Template

The blue blocks and paths in Figure 4 are common to all three features in the case study. The grey portion is the software module and its unique control and feedback paths. Each of these features is a software addition to the vehicle that relies mostly on existing hardware. None of the features alter the other functional blocks at this level of analysis and thus can share this common template. The black dashed-rectangle labeled "Environment and Other Drivers" is included in recognition that the driver-vehicle system does not exist in isolation. Furthermore, the purpose of many vehicle features is to aid the driver in interacting with the surrounding environment, e.g. ACC w/SG. This functional block is meant to represent the surroundings of the driver including, but not limited to: weather, road conditions, traffic conditions, the actions of other drivers, pedestrians, bikers, animals, etc. The vehicle contributes to this environment (perceived by other drivers) in two primary ways: the vehicle brake lights (activated by the brake pedal switch) and observable driving behavior.

The feedback and control actions common to all features are given below in Table 5 and Table 6. This control structure template may easily be extended and generalized for any automotive application.

Table 5: Common Driver Control Actions (Figure 4)

Control Action	Description
Brake Pedal Force	Represents the driver's ability to apply friction to the wheels by increasing the pressure in the hydraulic brake lines by stepping on the brake pedal. The travel of the brake pedal is traditionally monitored by an electronic switch.
Accelerate	The driver uses the accelerator pedal to open the throttle valve and allow more air into the engine. The Engine Control Unit (ECU) monitors this process and facilitates the injection of the appropriate amount of fuel. From the driver's perspective, pressing the accelerator pedal will increase the engine torque and move the vehicle when in gear.
Shift	The driver may select the range of the vehicle's transmission (automatic). The choices considered here are Park (P), Reverse (R), Neutral (N), Drive (D), and Low (L). This analysis does not distinguish between a traditional mechanical mechanism and an electronically controlled or "shift-by-wire" system. It is assumed that for either implementation, the transmission state is monitored electronically.

Table 6: Common Feedback Signals (Figure 4)

Feedback Signal	Description
Visual Cues	What the driver sees out the windshield and windows including (but not limited to): other vehicles; lane markings; road/weather conditions; terrain; pedestrians; animals; etc…
Sensory Feedback	What the driver feels that informs driving patterns and decisions such as: stiffness of steering column; acceleration/deceleration during braking, accelerating, and turning; road conditions; etc…
Dashboard Indicators	Encompasses general vehicular information presented through the instrument cluster including: speedometer, tachometer, odometer, fuel gage, gearshift indicator and warning indicators for: seat belts, parking-brake, check-engine, low fuel, low oil pressure, low tire pressure, engine-temperature, and airbag faults.
Brake Pedal Feel	Tactile feedback that the driver feels directly from the brake pedal when it is stepped on. This physical feedback includes the backpressure felt when pressing on the pedal and the amount of vibration. Abnormal backpressure can indicate a major problem such as a loss of power brakes or hydraulic system failure while abnormal vibration indicates other physical problems.
Driver Presence	This is an indicator as to whether or not the driver is residing their seat. This may be determined a number of ways, a specific implementation is not prescribed.

3.4. Auto-Hold

As described in Section 3.2.1, Auto-Hold (AH) is a feature that can take control of the braking system to hold the vehicle. A more detailed description of the feature is given along with its control structure and operating modes in Section 3.4.1. Following this introduction, Section 3.4.2 identifies the ways that Auto-Hold can issue an unsafe command leading to a hazard. Section 3.4.3 then identifies reasons (causal factors) that Auto-Hold may issue an unsafe command and ways that a potentially safe command may become unsafe.

3.4.1. AH – Control Structure and System Operation

The AH feature is implemented through an embedded controller that takes data from sensors around the vehicle and in turn can control mechanical components of the braking system. The details given here are specific to the fictional case study developed for this research; other implementation strategies may be used in practice. To hold the vehicle at a standstill, AH engages a valve that captures and maintains the pressure in the brake lines. If this pressure becomes insufficient to hold the vehicle at a standstill, AH may increase the pressure in the brake lines using the ABS pump. To disengage, AH releases the valve which allows the brake pressure to dissipate and gives control of the brakes back to the driver's service brake pedal. The feature's operating modes and control actions are given before the control structure is shown.

Modes:

HOLD-MODE: While in HOLD-MODE, AH is responsible for keeping the vehicle at a standstill. It may do so by increasing brake pressure or using the parking brake in some circumstances. The system exits HOLD-MODE by issuing the RELEASE command, it may do so when another system takes responsibility for holding the vehicle or propulsion torque is sufficient to move the vehicle.

ENABLED/DISABLED: The AH feature is either ENABLED or DISABLED at all times. Upon vehicle startup, the AH feature sets to DISABLED. The feature may then be ENABLED, and subsequently DISABLED, by the driver using a pushbutton. When ENABLED, the AH computer module monitors brake pressure and wheel speed so that when the brakes are used to bring the car to a stop, the system enters HOLD-MODE. The computer module may not issue commands when it is not ENABLED. After exiting HOLD-MODE, the AH system returns to DISABLED and must be ENABLED by the driver before engaging again. AH relays to the driver its state, ENABLED or DISABLED, through visual feedback, the form of which is not specified at this level of analysis.

Control Actions:

HOLD: When AH is ENABLED and the vehicle is brought to rest using the brake pedal, HOLD is issued to capture the existing brake pressure and place the feature in HOLD-MODE. AH identifies this situation by monitoring brake-pressure and wheels speed that are provided by the braking system.

ADDITIONAL PRESSURE: When the system is in HOLD-MODE and the wheels begin to rotate, the ADDITIONAL-PRESSURE command is issued that increases brake pressure (using the ABS pump) until the vehicle comes to rest.

RELEASE: When the system is in HOLD-MODE and one of two conditions is met, RELEASE is issued to release the valve and return the brake system to normal operation. 1) The propulsion torque is sufficient to move the vehicle. 2) Another system takes responsibility for holding the vehicle.

APPLY EPB: When the AH system is in HOLD-MODE, it may engage the EPB. This provision is for off-nominal cases which will be identified through the analysis.

The functional control structure for AH is given in Figure 5 following the template provided in Figure 4.

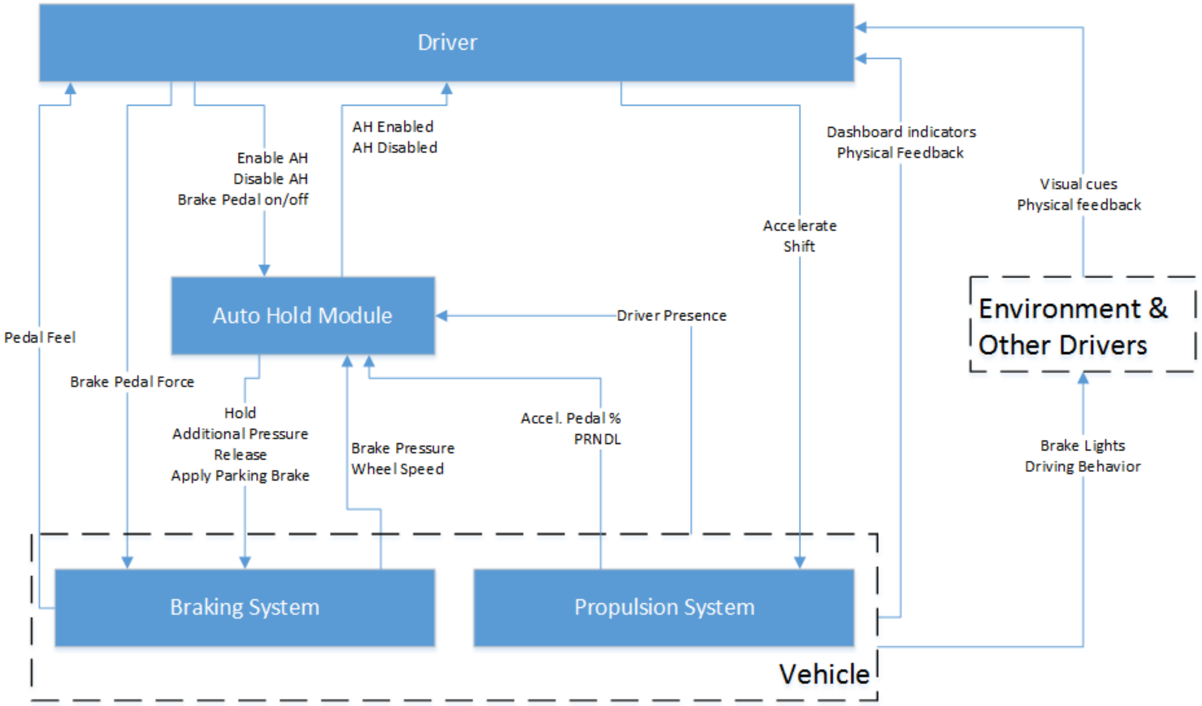


Figure 5: Auto-Hold Control Structure

3.4.2. AH – STPA Step 1 – Identifying Unsafe Control Actions

After completing the System Engineering Foundation, we begin STPA Step 1 that identifies how commands issued by the Auto-Hold Module may be unsafe. As is described in Section 2.4, there are four ways that a control action may be unsafe which can be organized in a table to guide brainstorming and aid recording. Table 7 presents ways that commands (control actions) issued by the Auto-Hold Module may be hazardous.

Table 7: Unsafe Control Actions for Auto-Hold

Controller: Auto-Hold Module				
Control Action	**Not Provided**	**Provided**	**Too Soon/Late**	**Too Long/Short**
HOLD	UCA-AH-1: Not providing HOLD is hazardous if AH is active and the vehicle comes to rest with the brake pedal on [H-1,2,3]	UCA-AH-2: Providing HOLD is hazardous if driver is applying the accelerator [H-1,3]	UCA-AH-3: Providing HOLD is hazardous if the driver has inactivated AH [H-1,2,3]	
		UCA-AH-4: Providing HOLD is hazardous if AH is DISABLED [H-1,3]	UCA-AH-5: Providing HOLD is hazardous if there is sufficient wheel torque [H-1,2,3]	
		UCA-AH-6: Providing HOLD is hazardous if AH is ENABLED and vehicle is not at rest [H-1,3]	UCA-AH-7: Providing HOLD is hazardous if the required time at rest has not been met [H-1,3]	
ADDITIONAL-PRESSURE	UCA-AH-8: Not providing ADDITIONAL-PRESSURE is hazardous if AH is in HOLD-MODE and vehicle is slipping [H-1,2,3]	UCA-AH-9: Providing ADDITIONAL-PRESSURE is hazardous if AH is not in HOLD-MODE [H-1,3]		
		UCA-AH-10: Providing ADDITIONAL-PRESSURE is hazardous if it exceeds brake system specs [H-3]		

Controller: Auto-Hold Module				
Control Action	**Not Provided**	**Provided**	**Too Soon/Late**	**Too Long/Short**
RELEASE	UCA-AH-11: Not providing RELEASE is hazardous if driver has commanded sufficient wheel torque via the accelerator pedal [H-1,2,3]	UCA-AH-12: Providing RELEASE is hazardous if AH is in HOLD-MODE and driver has not commanded sufficient wheel torque [H-1,3]	UCA-AH-13: Providing RELEASE before the there is sufficient wheel torque is hazardous [H-1,3]	
	UCA-AH-14: Not providing RELEASE is hazardous if the driver DISABLES AH [H-1,3]			
APPLY EPB	UCA-AH-15: It is hazardous not to provide APPLY EPB if the driver has released AH w/o sufficient wheel torque or brake pedal pressure [H-1,2,3]	UCA-AH-16: It is hazardous for AH to provide APPLY EPB if AH is not in HOLD-MODE [H-1,3]		

Note that Table 7 does not have any entries in the Too Long/Short column. The control actions that AH may issue are considered discrete commands rather than continuous. This means that ADDITIONAL-PRESSURE is meant to step up the brake pressure in pre-defined increments rather than continuously increase the pressure. Multiple ADDITIONAL-PRESSURE commands may be issued in sequence if necessary.

After identifying instances in which the control actions may lead to a hazard, steps should be taken to ensure that they are not issued in such situations. Safety constraints may be written, such that if followed the system will not enter a hazardous state as the result of issuing one of these control actions. The safety constraints on Auto-Hold's behavior are given in Table 8.

Table 8: Auto-Hold Safety Constraints

Safety Constraint	Related UCAs	Rationale
SC-AH1: AH shall not provide any commands if the feature is not ENABLED.	UCA-AH-3,4	AH may interfere with the vehicle dynamics and prevent the driver from fully utilizing the service brake.
SC-AH-2: AH shall not provide commands, other than HOLD, if it is not in HOLD-MODE.	UCA-AH-4,9,16	AH may interfere with the vehicle dynamics and prevent the driver from fully utilizing the service brake.
SC-AH-3: AH shall not interfere with the driver's ability to accelerate.	UCA-AH-2,5,11	AH should not compromise the driver's ability to perform basic vehicle functions.
SC-AH-4: AH shall not enter HOLD-MODE while the vehicle is in motion.	UCA-AH-5,6	AH may interfere with the vehicle dynamics when it is not expected by the driver.
SC-AH-5: AH must keep the vehicle at a standstill when in HOLD-MODE until the RELEASE criteria are satisfied.	UCA-AH-8,12,13,15	Premature release may leave the vehicle's velocity uncontrolled if the driver is not prepared.
SC-AH-6: AH should issue RELEASE when the criteria are met.	UCA-AH-11	AH should not compromise the driver's ability to perform basic vehicle functions and exercise high-level control over its features.
SC-AH-7: AH should not compromise the mechanical integrity of the vehicle.	UCA-AH-10	AH is an add-on feature, it should not compromise the basic vehicle functions.
SC-AH-8: AH must engage and disengage when the driver normally expects.	UCA-AH-1,14	

Close inspection of Table 7 and Table 8 will reveal some conflicts between constraints on the design of Auto-Hold. For instance, SC-AH-5 states that: "AH must keep the vehicle at a stand-still when in HOLD-MODE until the RELEASE criteria are met." The criteria referenced indicate that to issue RELEASE, the service brake must be applied, propulsion torque sufficient to move the vehicle present, or the EPB issued. UCA-AH-14 states that "Not providing RELEASE is hazardous if the driver DISABLES AH [H-1,3]" because it may prevent the driver from exercising full control over the service brake. This violates SC-AH-5 as the driver may press the AH button at any time in an attempt to disable the feature. This conflict presents the designers of the system with a design decision, an opportunity to prioritize goals and hazards in specifying the implementation of AH.

As was mentioned previously, one goal of this thesis is to provide some insight as to how STPA can help to refine a high-level concept. As the analysis moves forward, more design decisions will be flagged, although the majority of discussion concerning them will be postponed until the analysis is iterated and formalized. In fact, being able to rigorously identify and precisely define conflicts that lead to design decisions is partially the motivation behind the next two chapters, 4 and 5.

3.4.3. AH – STPA Step 2 – Identifying Causal Factors

STPA Step 1 identifies when it is unsafe and/or dysfunctional for a controller to issue or not issue commands. STPA Step 2 builds upon this and guides the analyst in identifying reasons that the controller may issue a command when inappropriate, not issue a command when necessary, and why a correctly issued command may be executed incorrectly. This guidance comes from Figure 3 which has a basic control loop depicting the fulfillment (or not) of a single control action: A controller issues commands to an actuator that then acts on some controlled process. Aspects of this controlled process are measured by sensors that provide feedback to the controller. The controller takes this feedback and builds a process model, i.e., its understanding of the state of the controlled process, which it then uses in conjunction with the control algorithm to make decisions about issuing a given command.

The Step 2 control loop associated with the ADDITIONAL-PRESSURE command is shown in Figure 6 which labels various parts of the loop for reference. This loop provides greater detail regarding the link between the Auto-Hold Control Module and Physical Vehicle in Figure 5.

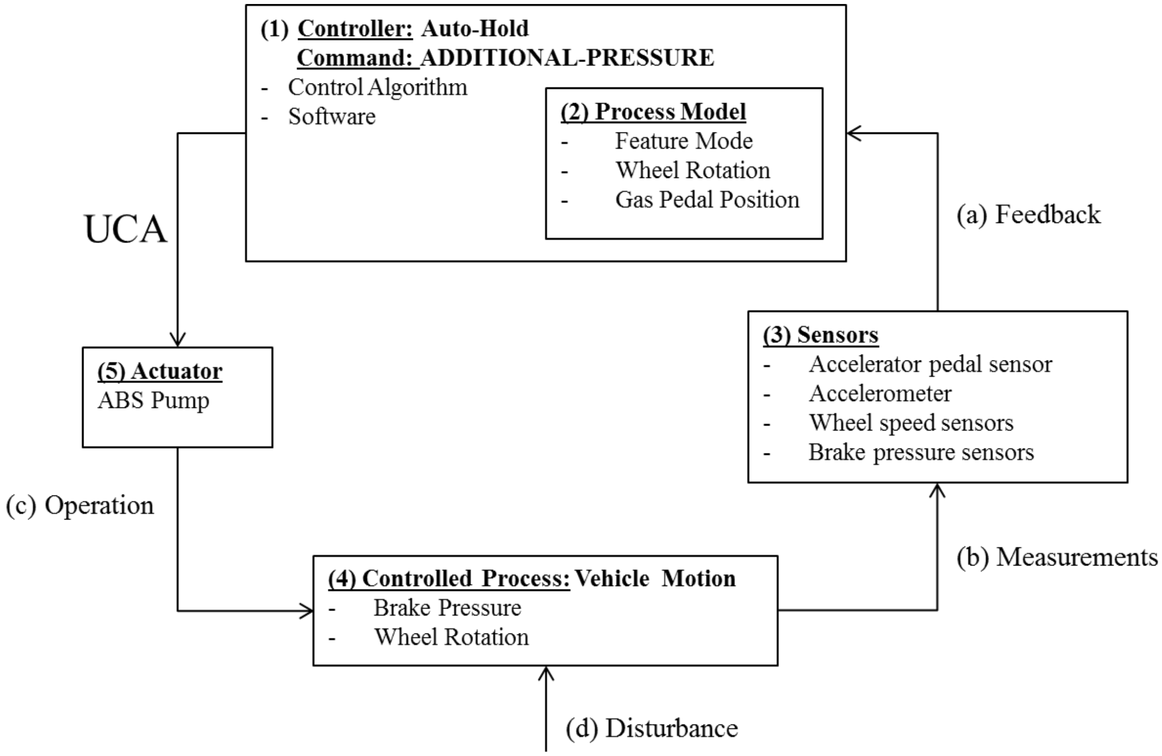


Figure 6: STPA Step 2 Guide for UCA-AH-8

3.4.3.1. Identifying Causes of an Unsafe ADDITIONAL-PRESSURE Command

Table 9 suggests causal factors that may lead to the unsafe issuing of ADDITIONAL-PRESSURE by inspecting the control loop shown in Figure 6. The specific UCA considered, from Table 7, is:

> **UCA-AH-8: Not providing ADDITIONAL-PRESSURE is hazardous if AH is in HOLD-MODE and vehicle is slipping [H-1,2,3]**

Table 9: UCA-AH-8 Causal Factors

Component/Link	Associated Causal Factors Leading to UCA-AH-8
(1) AH ADDITIONAL PRESSURE Control Algorithm	• Algorithm does not respond quickly when the Process Model changes in response to the wheels rotating.
(2) Process Model[6] • Wheels Rotating: [Yes, No] • HOLD-MODE: [Yes, No]	• AH does not correctly combine data from the different wheel speed sensors and does not detect the vehicle slipping. • While in HOLD-MODE, the AH computer restarts and initializes to HOLD-MODE: No and thus does not take responsibility for holding the vehicle.
(3) Sensors • Wheel Speed Sensors	• Wheels speed sensors degrade over time from poor connections and/or corrosion and do not detect the vehicle slipping. • Wheel speed sensors return data at an insufficient rate. • Wheel speed sensors do not detect sufficiently small/slow changes in value. E.g. the wheel speed sensors do not detect very slow rolling.

These causal factors may be interpreted as contributing reasons that Auto-Hold will issue a command when inappropriate. This process of identifying causal factors may be completed for every UCA and the combined set can then be used to write requirements for the components of the system that will help enforce the safety constraints on the behavior of the feature given in Table 8.

[6] These Process Model Variables come from the analysis presented in Chapter 4.

3.4.3.2. Identifying Causes of a Violated AH Safety Constraint

The prior section identified reasons that a specific UCA may be issued, which is one way that unsafe control can occur. Another cause of unsafe control is that an appropriate control action is issued but not executed properly. Table 10 list reasons that the ADDITIONAL-PRESSURE command may not be executed properly even if issued correctly, which would then violate the safety constraint:

SC-AH-5: AH must keep the vehicle at a standstill when in HOLD-MODE until the RELEASE criteria are satisfied.

Table 10: AH ADDITIONAL PRESSURE Executed Incorrectly – Causal Factors

Component/Link	Associated Causal Factors
(4) Controlled Process • Brake Pressure • Wheel Rotation	• Seals in the brake system degrade over time such that braking pressure cannot be sustained to hold the vehicle. • Brake lines degrade due to weathering, salt, wear & tear, etc. and are not able to withstand the maximum pressure.
(5) Actuator • ABS Pump	• The pump physically degrades over time and is not able to reach its original max pressure. • The pump' seals degrade and begin to leak fluid when increasing pressure. • The pump does not have an adequate power supply to reach the required pressure. • The pump does not stop increasing brake pressure and compromises the integrity of the hydraulic system.

3.5. Engine Stop-Start

As described in Section 3.2.2, Engine Stop-Start (ESS) is a feature that can control the engine, turning it on and off to reduce engine idle time. A more detailed description of the feature is given along with its control structure and operating modes in Section 3.5.1. Following this, Section 0 identifies the ways that Stop-Start can issue an unsafe command leading to a hazard. Section 0 then identifies reasons (causal factors) that Engine Stop-Start may issue an unsafe command and ways that a potentially safe command may become unsafe.

3.5.1. ESS – Control Structure and System Operation

The ESS feature is implemented through an embedded controller that takes data from sensors around the vehicle and can turn the engine on and off using the electric starter. Engine Stop-Start is designed to turn the vehicle's engine off when the vehicle is at rest and to turn it on again before motion resumes. Engine Stop-Start is intended to be used at short traffic stops and not at the beginning and end of travel. Engine Stop-Start is also restricted in that it must not turn the engine off if doing so will compromise other vehicle systems through a loss of power. The feature's operating modes and control actions are given after the control structure is shown. The details given here are specific to the fictional case study developed for this research; other implementation strategies may be used in practice.

The functional control structure for ESS is given in Figure 7 following the template provided in Figure 4.

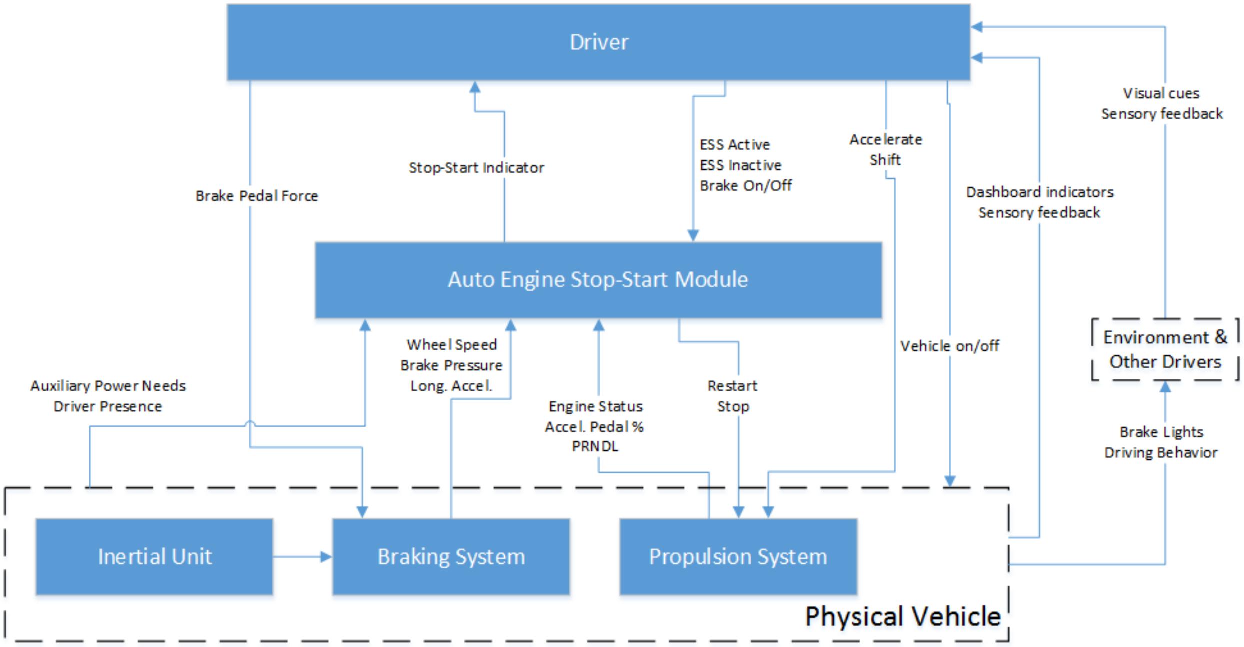


Figure 7: Auto Stop-Start Control Structure

The ESS feature employs an ESS computer module that is capable of issuing two actionable commands: STOP and RESTART. These commands and the system modes are described below.

Commands:

STOP: Once the vehicle is brought to rest using the brake pedal, *Auto-Stop* is issued which shuts down the engine.

RESTART: When the system is in AUTO-STOPPED and either the driver triggers (currently by releasing the brake pedal) motion or power needs arise, RESTART is issued which will restart the engine.

System Modes:

AUTO-STOPPED Mode: Once STOP has been issued; the system enters AUTO-STOPPED mode. In this mode the engine is off and the system is waiting for the driver to trigger a restart (by releasing the brake) or for power needs to exceed that supplied by the battery.

ENABLED/DISABLED Modes: The ESS feature can be enabled or disabled offline in the vehicle settings. When enabled, the ESS computer module waits to be triggered when the vehicle is being held at rest while in Drive.

3.5.2. ESS – STPA Step 1 – Identifying Unsafe Control Actions

Table 11 presents ways that commands issued by the Engine Stop-Start Module may be hazardous.

Table 11: Unsafe Control Actions for Engine Stop-Start

Controller: Auto Stop-Start Module				
Control Action	**Not Provided**	**Provided**	**Too Soon/Late**	**Too Long/Short**
RESTART	UCA-ESS-1: Not providing RESTART is hazardous if the driver releases the brake while system is in AUTO-STOPPED [H-3]	UCA-ESS-2: Providing RESTART is hazardous if the car is not AUTO-STOPPED [H-1,2,3]	UCA-ESS-3: It is hazardous to not provide RESTART within TBD time after driver releases brake while in 'standstill'	
	UCA-ESS-4: Not Providing RESTART is hazardous if a critical non-propulsion power need arises [H-3]	UCA-ESS-5: Providing RESTART is hazardous if the engine is running - damage to the starter mechanism [H-3]		
	UCA-ESS-6: Not Providing RESTART is hazardous is conditions for a favorable restart change [H-3]			
STOP		UCA-ESS-7: Providing STOP is hazardous if driver has not brought vehicle to stop w/brake [H-1,3]	UCA-ESS-8: Providing STOP is hazardous if it is done before vehicle is at rest [H-1,3]	
		UCA-ESS-9: Providing STOP is hazardous if there will not be sufficient power for RESTART [H-3]		

As was done with Auto-Hold, safety constraints for the Engine Stop-Start system are written based upon the Unsafe Control Actions identified above and presented in Table 12.

Table 12: Engine Stop-Start Safety Constraints

Safety Constraint	Related UCAs	Rationale
SC-ESS-1: ESS should not STOP the engine when the vehicle is in motion.	UCA-ESS-7,8	The engine is required for providing power to power-assisted systems such as steering and braking. Issuing STOP prematurely may surprise the driver and make it difficult to control the vehicle.
SC-ESS-2: ESS should RESTART the engine before motion resumes once AUTO-STOPPED.	UCA-ESS-1,3	The engine is required for providing power to power-assisted systems such as steering and braking. Not proving power may allow the vehicle to travel without control.
SC-ESS-3: ESS should not start the engine unless is currently AUTO-STOPPED.	UCA-ESS-2	ESS should not start the engine when it does not have authority to do so as it may lead to rollaway. Starting the engine when it is already on will unnecessarily wear on the start mechanism.
SC-ESS-4: ESS should not prevent the operation of other vehicle subsystems.	UCA-ESS-4,5	ESS should not prevent electrical systems from drawing the current they require.
SC-ESS-5: ESS should not operate when it is not able to complete a full operational cycle.	UCA-ESS-6,9	Shutting off the engine when a restart is not possible may strand the vehicle in a precarious location.

Enforcing the five safety constraints given in Table 12 will ensure that ESS does not lead to accident by putting the system into one of the four hazardous states that were identified in Section 3.3.3. These general constraints can be formalized which in turn will refine the resulting requirements regarding ESS behavior. This is done in the next chapter as the evolution of STPA is demonstrated.

3.5.3. ESS – STPA Step 2 – Identifying Causal Factors

Further study of the ESS RESTART command is used to demonstrate how the analysis may be continued to the identification of causal factors using STPA Step 2. A control loop corresponding to the ESS RESTART command is given in Figure 8. This loop shows that the ESS controller sends commands to the Engine Starter which in turn may start up the engine. The ESS controller receives feedback from several vehicle systems including the engine, transmission, power systems, and driver.

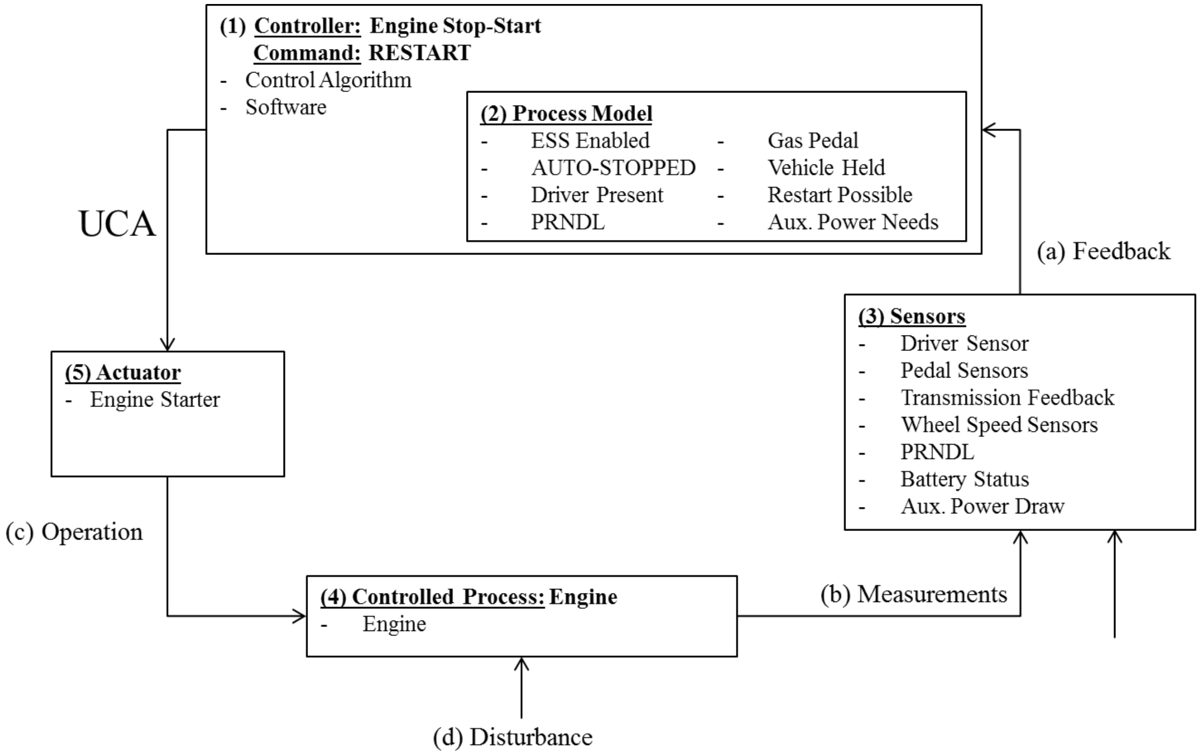


Figure 8: STPA Step 2 Guide for ESS

3.5.3.1. Identifying Causes of an Unsafe RESTART Command

Table 13 identifies causal factors that may lead to the unsafe issuing of the ESS RESTART command by inspecting the control loop shown in Figure 8. The specific UCA considered, from Table 11, is:

> **UCA-ESS-1: Not providing RESTART is hazardous if the driver releases the brake while system is in AUTO-STOPPED [H-3]**

Table 13: UCA-ESS-1 Causal Factors

Hazardous Scenario	Associated Causal Factors Leading to UCA-ESS-1
(1) ESS RESTART Control Algorithm	• Algorithm does not correctly predict battery voltage drain and is unable to engage the starter.
(2) Process Model[7] • ESS ENABLED: [Yes, No] • AUTO-STOPPED: [Yes, No] • Driver Present: [Yes, No] • PRNDL: [P,R,N,D,L] • Gas Pedal: [Pressed, Not Pressed] • Vehicle Held: [Yes, No] • Restart Possible: [Yes, No] • Aux. Power Needs: [High, Low]	• ESS does not abstract Vehicle Held correctly when combining inputs from various braking systems and believes the vehicle is held when it is not. • Auxiliary power needs are not monitored, or are not sufficiently anticipated, and the controller reports Low when the value should be High leading to an inability to Restart.
(3) Sensors • Accelerator Pedal Sensor • Driver Sensor • Wheel Speed Sensors • Transmission Feedback • Battery Charge Sensor • Power Bus Sensor	• Brake sensor fails or reports a false value such that the brake release is not capture and ESS does not anticipate motion. • Noise is not adequately filtered and the brake release is not identified. • Sensors do not detect sufficiently small/slow changes in value. E.g. the brake sensor does not detect a small shift in pedal position that may allow the vehicle to begin slipping.

[7] These Process Model Variables come from the analysis presented in Chapter 4.

3.5.3.2. Identifying Causes of a Violated ESS Safety Constraint

ESS may also put the vehicle in a hazardous state if it issues a control action correctly, but the action is not executed properly. Table 14 proposes reasons that the RESTART command may not be executed properly even if issued correctly, which would then violate the safety constraint:

SC-ESS-2: ESS should RESTART the engine before motion resumes once AUTO-STOPPED.

Table 14: ESS RESTART Executed Incorrectly – Causal Factors

Hazardous Scenario	Associated Causal Factors Leading to UCA-ESS-1
(4) Controlled Process • Engine	• Engine performance changes over time such that the actual torque does not match that assumed by the ESS algorithm. • Engine misfires during startup and allows the vehicle to roll. • Engine does not receive the proper air/fuel mixture and cannot start.
(5) Actuator • Engine Starter	• Starter degrades over time due to weathering and exposure and does not execute the RESTART command • Starter degrades due to excessive cycles • Starter does not operate within an acceptable timeframe, some TBD seconds. • The driver attempts to shut-off the vehicle as ESS commands a RESTART.

This investigation of causal factors should be completed for each of the UCA's and safety constraints associated with ESS. The aggregated list of scenarios and causal factors can then be used change the design of the system to eliminate, or reduce, the occurrence of unsafe control.

3.6. Adaptive Cruise Control with Stop & Go

As described in Section 3.2.3, Adaptive Cruise Control with Stop & Go (Stop-Go or SG) is a feature that can use the propulsion and braking systems to control the vehicle's velocity. A more detailed description of the feature is given along with its control structure and operating modes in Section3.6.1 before Section 3.6.2 identifies the ways that Stop-Go can issue an unsafe command leading to a hazard. Section 3.6.3 then identifies reasons (causal factors) that Stop-Go may issue an unsafe command and ways that a potentially safe command may become unsafe.

3.6.1. ACC w/SG – Control Structure and System Operation

The ACC w/SG feature is implemented through an embedded controller that takes data from sensors around the vehicle and in turn can control the braking and propulsion systems. The feature must be enabled and engaged by the driver before influencing the vehicle's dynamics. The details given here are specific to the fictional case study developed for this research; other implementation strategies may be used in practice. Figure 9 shows a high level vehicle control structure with a Stop-Go system following the template provided in Figure 4.

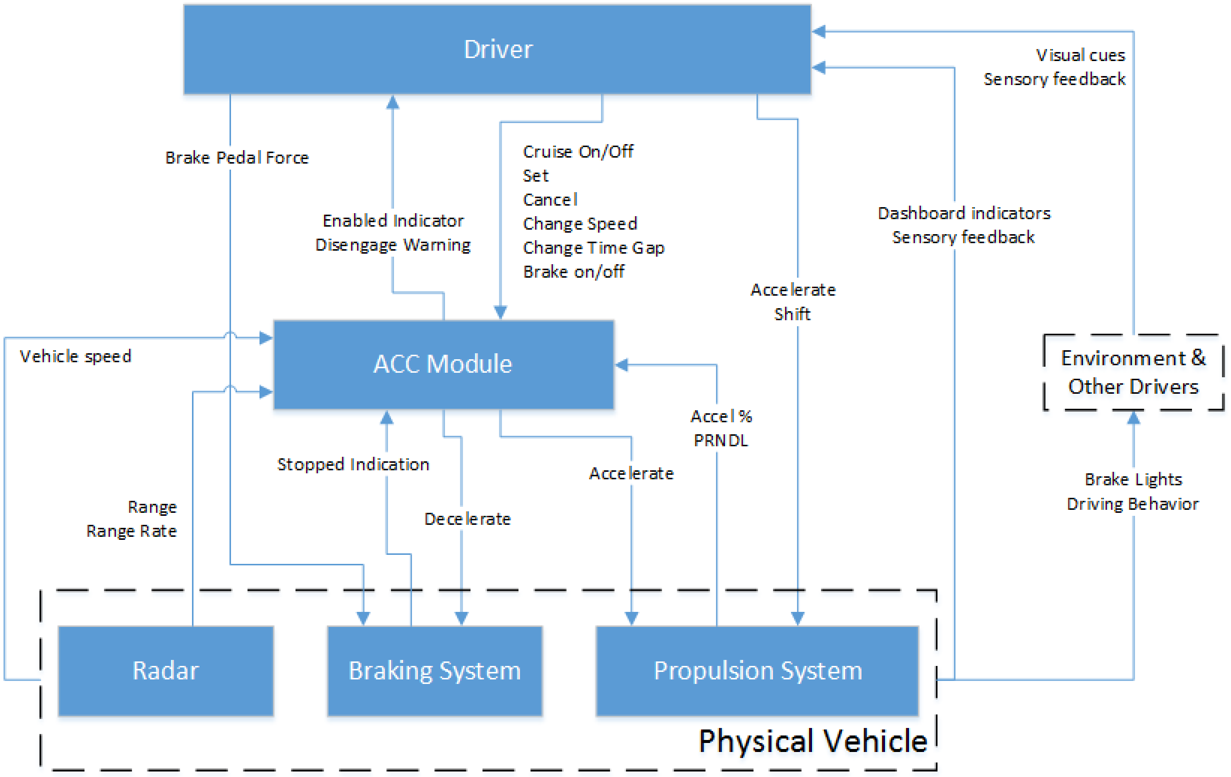


Figure 9: ACC w/Stop-Go Control Structure

The ACC w/SG feature employs a computer module that is capable of issuing four actionable commands: *Engage, Accelerate, Brake,* and *Cancel*. These commands and the system modes are described below.

Commands:

ENGAGE: If the feature is active (enabled) then it may be engaged by the driver using a button on or near the steering column. When the feature is engaged, it takes the current speed as the goal speed and begins to control the velocity of the vehicle to maintain that set speed while meeting the specified spacing requirements.

ACCELERATE: When the system is enabled, it may accelerate the vehicle using the propulsion system. At this level, the means of acceleration (constant vs. incremental vs. other) is not specified.

DECELERATE: When the system is enabled, it may decelerate the vehicle using the braking system. At this level, the means of deceleration (constant vs. incremental vs. other) is not specified.

CANCEL: This command disengages the feature and relinquishes control and influence over the vehicle's velocity.

System Modes:

CRUISE Mode: Once ENGAGE has been issued; the system enters CRUISE MODE. In this mode the ACC w/SG Module may control the velocity of the vehicle until one of the cancel criteria is met or the vehicle is turned off.

ENABLED/DISALED Modes: The ACC w/SG feature can be enabled or disabled via a Driver control on or near the steering column. When enabled, the ACC w/SG computer module waits to be triggered by the ENABLE command.

3.6.2. ACC w/SG – STPA Step 1 – Identifying Unsafe Control Actions

Table 15 presents ways that commands issued by the Stop-Start Module may be hazardous.

Table 15: Unsafe Control Actions for ACC w/Stop-Go

Controller: Auto-Hold Module				
Control Action	**Not Provided**	**Provided**	**Too Soon/Late**	**Too Long/Short**
ACCELERATE	UCA-SG-2: It is hazardous to not provide ACCELERATE when a vehicle or other mobile object is approaching at TBD rate/distance [H-1]	UCA-SG-3: It is hazardous to provide ACCELERATE if the vehicle is already at the preset speed [H-3]	UCA-SG-4: It is hazardous to provide ACCELERATE before ACC is engaged [H-1,2,3]	UCA-SG-5: It is hazardous to provide ACCELERATE too long such that the vehicle exceeds the preset speed [H-1,3]
		UCA-SG-6: It is hazardous to provide ACCELERATE if the vehicle is closer than the minimum safe distance from a leading vehicle [H-1]	UCA-SG-7: It is hazardous to issue ACCELERATE too late after the speed is set and ACC engaged [H-1,3]	UCA-SG-8: It is hazardous to provide ACCELERATE too long such that the vehicle violates a minimum safe trailing distance from a leading vehicle in the lane [H-1]
		UCA-SG-9: It is hazardous to provide ACCELEREATE if the vehicle is moving to collide with an object within TBD rate/distance in its trajectory [H-1,2]		
		UCA-SG-10: It is hazardous to provide ACCELERATE if the vehicle is at rest and the driver has not indicated it is safe to resume motion [H-3]		
		UCA-SG-11: It is hazardous to provide ACCELERATE if ACC is not enabled/engaged [H-3]		
		UCA-SG-12: It is hazardous to provide ACCELERATE if it closes a gap at a high range rate [H-1,2,3]		

Controller: Auto-Hold Module				
Control Action	**Not Provided**	**Provided**	**Too Soon/Late**	**Too Long/Short**
DECELERATE	UCA-SG-13: It is hazardous to not provide DECELERATE if there is an obstacle ahead in the lane and the range rate is negative [H-1,2]	UCA-SG-14: It is hazardous to provide DECELERATE too abruptly [H-1,3,4] - design conflict with box to the left	UCA-SG-15: It is hazardous if DELECERATE is issued too late after an obstacle (fixed or slowing vehicle) has been detected [H-1,2]	UCA-SG-16: It is hazardous if DECELERATE is not provided long enough (too short) to sufficiently decelerate the vehicle [H-1,2,3]
	UCA-SG-17: It is hazardous to not provide DECELERATE if the vehicle is closer than the minimum safe distance from a leading vehicle and range rate is not increasing [H-1]			
	UCA-SG-18: It is hazardous to not provide DECELERATE if the vehicle is closing faster than some TBD rate/distance on an obstacle ahead [H-1,3]	UCA-SG-19: It is hazardous to provide DECELERATE if it slows the vehicle's speed too much for the given roadway traffic conditions [H-1,3]		
		UCA-SG-20: It is hazardous to provide DECELERATE if ACC is not enabled/engaged [H-3]		
		UCA-SG-21: It is hazardous to provide DECELERATE when not commanded by driver and there is no obstacle ahead [H-1,3]		

As was done with previously, safety constraints for the ACC w/SG system may be written based upon the Unsafe Control Actions identified in Table 15. The high level safety constraints on Engine Stop-Start's behavior are given in Table 16.

Table 16: ACC w/SG Safety Constraints

Safety Constraint	Related UCAs	Rationale
SC-SG-1: SG may not exceed driver set limits on speed and following distance.	UCA-SG-2,3,5,6,8,9,12,13,17,18	ACC w/SG is intended to aid the driver in maintaining a safe speed and following distance. It does not have the feedback or intelligence to override driver input.
SC-SG-2: SG should issue commands that provide smooth vehicle acceleration (positive and negative).	UCA-SG-14,19	ACC w/SG should not decrease the ride quality during normal operation.
SC-SG-3: SG must issue commands to avoid a collision.	UCA-SG-15,16	Regularly contributing to collisions will greatly decrease the efficacy and value of the feature.
SC-SG-4: SG must not issue commands when it has not been enabled by the driver.	UCA-SG-4,8,11,20	ACC w/SG is meant to aid the driver at his discretion – not take over or correct driver actions.

As has been seen before with the other features, conflict exists between the safety constraints that were produced for SG. For example, SC-SG-2 requires that accelerate and decelerate commands provide smooth vehicle motion so as not to injure the passengers and startle other drivers. However, SC-SG-3 requires that the SG system must always avoid a collision, which may at times require sudden changes in acceleration – violating SC-SG-2. At this level of abstraction, the conflict is fairly broad and no tangible guidance is available for the design team that must mitigate the competing constraints. Further refining the Unsafe Control Actions will better define the instances when the constraints may conflict and ensure that the design team considers all relevant system states, this is demonstrated in Chapter 4.

3.6.3. ACC w/SG – STPA Step 2 – Identifying Causal Factors

As was done for AH and ESS, a guide for completing Step 2 with regards to the ACC w/SG ACCELERATE command is presented in Figure 10. The control loop depicts how the ACCELERATE command is realized by the Propulsion System to increase wheel speed, which is in turn monitored by wheel speed sensors. Other sensors provide information about the presence of the driver in the seat and the range to vehicles ahead in traffic. This loop provides greater detail regarding the link between the Auto-Hold Control Module and Physical Vehicle in Figure 9.

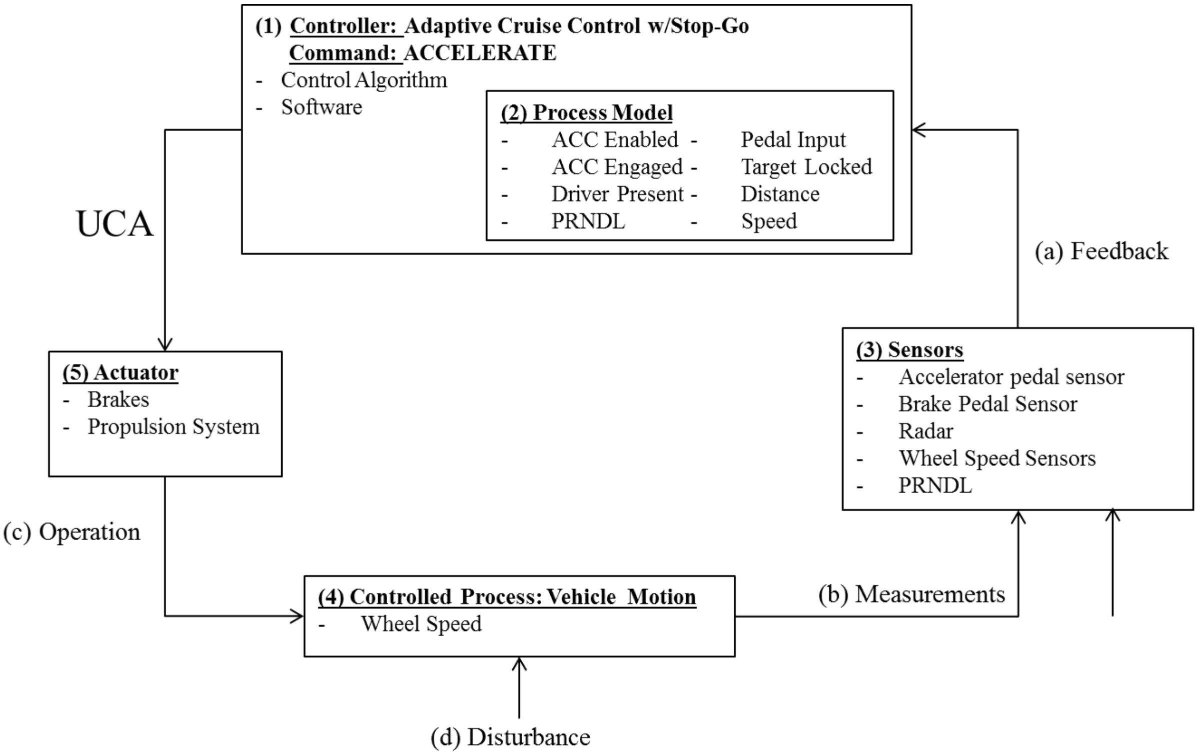


Figure 10: STPA Step 2 Guide for UCA-SG-9

3.6.3.1. Identifying Causes of an Unsafe RESTART Command

Table 17 presents the causal factors associated with UCA-SG-9, an inappropriate issuing of the ACCELERATE command:

UCA-SG-9: It is hazardous to provide ACCELEREATE if the vehicle is moving to collide with an object within TBD rate/distance in its trajectory [H-1, 2]

Table 17: UCA-SG-9 Causal Factors

Component/Link	Associated Causal Factors Leading to UCA-SG-9
(1) SG ACCELERATE Control Algorithm	• Algorithm does not correctly calculate the closing distance and rate to an object ahead. • At the time ACC w/SG is engaged, the target is too close to identify, calculate, and mitigate.
(2) Process Model[8] • ACC Enabled: [Yes, No] • ACC Engaged: [Yes, No] • Driver Present: [Yes, No] • PRNDL: [P,R,N,D,L] • Drive Pedal Input: [Yes, No] • Target Locked: [Yes, No] • Distance ≥ Threshold: [Yes, No] • Speed ≥ Threshold: [Yes, No]	• The controller does not lock onto a target and thus does not anticipate a collision. • Distance threshold is incorrect and allows the vehicle to get too close to the target.
(3) Sensors • Accelerator Pedal Sensor • Brake Pedal Sensor • Wheel Speed Sensors • Radar • Driver Presence Sensor(s)	• Radar is unable to detect targets due to road and weather conditions. • Radar data is not returned at a sufficient rate to avoid a collision. • Noise is not adequately filtered and a small target is not identified.

3.6.3.2. Identifying Causes of a Violated ACC w/SG Safety Constraint

As discussed previously, it is necessary to also consider when a control action that is correctly issued may lead to a hazard because it is not properly executed. An example for ACC w/SG is presented here in Table 18 at identifies way that a correctly issued DECELERATE command may not be executed and thus violate the safety constraint:

SC-SG-3: SG must issue commands to avoid a collision.

[8] These Process Model Variables come from the analysis presented in Chapter 4.

Table 18: ACC w/SG DECELERATE Executed Incorrectly – Causal Factors

Component/Link	Associated Causal Factors
(4) Controlled Process - Wheel Speed	- Tire treads are too low to provide adequate friction for decelerating. - Change in tire size unaccompanied by recalibration offsets the wheel speed readings which compromises the closing distance calculations.
(5) Actuator - Braking System	- Brakes degrade over time (alignment, pads, seals, hydraulic lines) and cannot match original performance. - Brake force is not adequate to meet required deceleration.

3.7. Summary and Next Steps

This chapter has presented the basic application of STPA to each of the three fictionalized feature systems: Auto-Hold, Engine Stop-Start, and Adaptive Cruise Control w/SG. After establishing a common set of accidents and hazards as well as a control structure template, the concept for each feature was described and analyzed. Instances when commands issued by controllers may be unsafe were identified and the process for finding causal factors leading to unsafe control was demonstrated. Even at this high level of abstraction, the results may be used to further develop the design of each feature. Not captured in this thesis is the discussion stimulated among MIT researchers and collaborators throughout the analysis process. This discussion serves to develop a common understanding of the feature systems among those involved in the design process which enhances cohesion amongst members who will later be responsible for implementing more detailed designs.

If one revisits the UCA tables (Table 7, Table 11, and Table 15), it can be seen that some entries (and the resulting safety constraints) concern both the system level hazards and feature functionality. Chapter 4 will demonstrate how iterating STPA Step 1 with a formal method can help distinguish between constraints required for safety and those required for system function. The requirements generated from this iteration will be the foundation for the new approach to analyzing the integrated system that is presented in Chapter 5.

Page intentionally left blank.

4. Generating Executable Requirements

4.1. Overview and Motivation

Chapter 3 presented the basic application of STPA to the three automated control systems: Auto-Hold, Engine Stop-Start, and Adaptive Cruise Control with Stop & Go. As was noted in the section summary, the results of the analysis are useful but not yet actionable at a design level. There are several ways to take the analysis to a greater level of detail, one being to reiterate using the method that has been already demonstrated. However, as the level of refinement increases, the analysis scope grows quickly and may be difficult for an individual analyst to manage. Another strategy is to iterate using methods that require increased formality, but produce more comprehensive and precise results. This strategy will be demonstrated here to iterate STPA Step 1, identifying unsafe control actions, for the three automated features. The additional detail that is presented in this chapter has been fictionalized for the purpose of this research, it is not meant to represent the specific implementation of a particular manufacturer.

4.2. Refining UCAs and Generating Requirements

A new method for identifying and specifying unsafe control actions was proposed by Thomas in [12], and thus will be referred to here as the Thomas Method. Thomas observed that the Unsafe Control Actions identified in STPA Step 1 (e.g. Table 7, Table 11, and Table 15) often have a common structure that may be formalized. This is demonstrated in Figure 11 using a rephrasing of UCA-H-2.

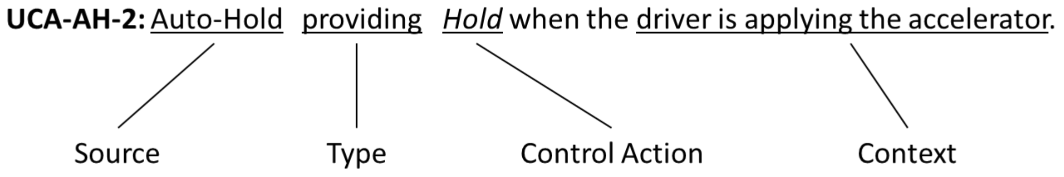

UCA-AH-2: Auto-Hold providing *Hold* when the driver is applying the accelerator.
Source
Type
Control Action
Context

Figure 11: Structure of a Hazardous Control Action

This four part structure is common to all the Unsafe Control Actions identified in Section 3. In general, the *source* and *control action* fields are found in the relevant System Control Structure developed as part of the System Engineering Foundation. The *type* may be provided or not-provided, in accordance with the four ways a control action may be unsafe[9] as described in Section 2.4. The *context* is then what

[9] Applied too late/soon or long/short are special cases of provided and not-provided.

distinguishes the issuing of a control action as safe or unsafe, dysfunctional or functional. If the contexts can be formalized, an enhanced set of UCAs can be generated by evaluating whether the providing or not-providing of a control action is appropriate in each potential context.

An individual context can be described by a set of *process model variables* – variables that describe a precise aspect of the system state. Different combinations of process model variables can be assembled to precisely define a context. Two examples Auto-Hold Process Model Variables (PMVs) and values that they may take are given in Figure 12.

Context Variables	**Context Values**
AH Enabled	Yes No
Brake Pedal	Pressed Not Pressed

Figure 12: Example Context Variables/Values

To fully evaluate a control action, each potential context in which it may be issued must be covered by the analysis. The relevant contexts are organized in a *context table* in which each row represents a context, or system state defined using the PMVs, for which the providing or not-providing of a control action is evaluated against hazards and functional goals. It is possible that a set of combinations (set of system states) may share a trait that leads to a common conclusion. It is also possible that a set of PMVs may be abstracted into a type of global variable that reduces the number of states an analyst must consider.

Once formalized and fully evaluated the context tables may be used to generate executable requirements. The requirements specify when a particular action must be issued to satisfy safety constraints and/or functional goals. In this case study, these requirements are presented using SpecTRM-RL tables [13] which may be used as black-box models in a simulation environment before detailed design is complete. More information on the Thomas Method, automated review, and automatically generating requirements may be found in [12].

The following sections present the application of this method to each of the three features: Auto-Hold, Engine Stop-Start, and Adaptive Cruise Control with Stop & Go. For each feature, the relevant PMVs are defined and a *context table* is presented for each control action. Following the context table, executable requirements are presented using SpecTRM-RL tables. General comments regarding the results of the tables are given in this section, a more detailed record of the analysis, including traceability to functional goals and system hazards, may be found in Appendix A.

4.3. Common Process Model Variables

As with components of the control structures, the three features share some common Process Model Variables. These common PMVs primarily concern vehicle level feedback (such as transmission range) or feedback concerning the driver (whether or not he is seated). These common PMVs are defined below in Table 19. Process Model Variables that are unique to each feature will be given in the appropriate section, prior to the presentation of the relevant context table.

Table 19: Common Context Variables

Driver Present ***Yes:*** *The driver is in the seat* ***No:*** *The driver is not in the seat or position is transitory* ***Implementation:*** *F(door sensor, seat belt, …)*	**[Yes, No]**
PRNDL ***P:*** *the transmission is in Park* ***R:*** *the transmission is in Reverse* ***N:*** *the transmission is in Neutral* ***D:*** *the transmission is in Drive* ***L:*** *the transmission is in Low* ***Implementation:*** *Propulsion system feedback*	**[P, R, N, D, L]**
Gas Pedal ***Pressed:*** *Accelerator pedal position is at least Y mm* ***Not Pressed:*** *Accelerator pedal position is less than detectable Y mm* ***Implementation:*** *Accelerator pedal potentiometer*	**[Pressed, Not Pressed]**
Brake Pedal ***Pressed:*** *Brake pedal travel is at least X mm* ***Not Pressed:*** *Brake pedal travel is less than detectable X mm* ***Implementation:*** *Brake pedal switch*	**[Pressed, Not Pressed]**
Wheels Rotating ***Yes:*** *Wheels are moving at least deg/min* ***No:*** *Wheels are stopped or less than detectable deg/min* ***Implementation:*** *Wheel speed sensors*	**[Yes, No]**

4.4. Auto-Hold

The initial application of STPA to the Auto-Hold system yielded eight high-level constraints on the feature's behavior (Table 8). This section demonstrates how the search for inadequate (unsafe or dysfunctional) control actions may be formalized to generate more specific constraints. Also, the generation of executable requirements is demonstrated for each of the four commands Auto-Hold may issue: HOLD, ADDITIONAL-PRESSURE, RELEASE, APPLY EPB.

The PMVs unique to Auto-Hold are presented before the context table for each command is shown. Some discussion of the context tables is provided, specifically regarding conflicts that were resolved during their development. Line-by-line reasoning and comments for each of the context tables may be found in Appendix A. Following this discussion, the requirements that govern Auto-Hold's behavior are provided.

The PMVs unique to Auto-Hold are defined in Table 20. The combination of these and the common PMVs (Table 19) are used to identify the contexts in which Auto-Hold will issue commands [HOLD, ADDITIONAL PRESSURE, RELEASE, APPLY EPB]. Each context will be evaluated to assess whether issuing a given command is appropriate or not. The output of this process is the set of reduced context tables shown in the following section.

Table 20: Auto-Hold Module Process Model Variables

AH Enabled ***Yes:*** *AH is allowed to enter hold-mode* ***No:*** *AH is not allowed to enter hold-mode – initial state* ***Implementation:*** *Driver push-button*	**[Yes, No]**
Hold-Mode ***Yes:*** *AH has control of brakes and is allowed to issue HOLD and ADDITIONAL PRESSURE* ***No:*** *AH does not have control of brakes and is not allowed to issue HOLD or ADDITIONAL PRESSURE- initial state* ***Implementation:*** *Brake valve state*	**[Yes, No]**
Torque Sufficient *Is engine torque applied to wheels by the propulsion system >= the torque required to hold still or move in direction of propulsion?*	**[Yes, No]**

The PMVs defined in Table 19 and Table 20 can be combined to describe a set of systems states in which Auto-Hold may issue a command. The full set of system states must be evaluated for a thorough analysis —meaning every combination of the PMVs and their values. This is quite a long list; fortunately some of the states, or contexts, have something in common that leads to a common conclusion. For example, this abstraction strategy may be used when considering the HOLD command. For all the contexts in which the Gas Pedal is pressed, it is hazardous for Auto-Hold to issue the HOLD command. Thus, all the contexts in which the Gas Pedal is pressed can be condensed into one row in the HOLD Context Table. Similar combinations and abstractions are also used in the tables associated with ADDITIONAL PRESSURE, RELEASE, and APPLY EPB.

The context tables for each command issued by Auto-Hold are provided as a set (as opposed to separately, with interspersed discussion) because they together describe the behavior of the Auto-Hold system. When completing the context tables, and thus generating the SpecTRM-RL requirements, it is necessary to consider the effect of decisions and changes on the entire feature system.

Table 21: AH – HOLD Context Table

Context ID	AH Enabled	Hold-Mode	Driver Present	PRNDL	Gas Pedal	Brake Pedal	Wheels Rotating	Providing Causes Hazard	Not Providing Causes Hazard	Providing Required for Function	Not Providing Require for Function
AH-H-1	Yes	Yes	Yes	P	Not Pressed	*	No	x			
AH-H-2	Yes	Yes	Yes	!=P	Not Pressed	*	No				
AH-H-3	Yes	No	Yes	P	Not Pressed	*	No	x			
AH-H-4	Yes	No	Yes	N	Not Pressed	*	No	x			x
AH-H-5	Yes	No	Yes	R	Not Pressed	*	No	x			x
AH-H-6	Yes	No	Yes	D,L	Not Pressed	Not Pressed	No	x			
AH-H-7	Yes	No	Yes	D,L	Not Pressed	Pressed	No		x	x	
AH-H-8	*	*	*	*	*	*	Yes	x			
AH-H-9	*	*	*	*	Pressed	*	*	x			
AH-H-10	*	*	No	*	*	*	*	x			
AH-H-11	No	*	*	*	*	*	*	x			

Table 22: AH – AP Context Table

Context ID	Hold-Mode	Gas Pedal	Wheels Rotating	Providing Causes Hazard	Not Providing Causes Hazard	Providing Required for Function	Not Providing Required for Function
AH-AP-1	Yes	Not Pressed	Yes		x	x	
AH-AP-2	Yes	Not Pressed	No	x			
AH-AP-3	Yes	Pressed	Yes		x		
AH-AP-4	Yes	Pressed	No	x			
AH-AP-5	No	*	*	x			

Table 23: AH – RELEASE Context Table

Context ID	AH Enabled	Hold-Mode	Driver Present	PRNDL	Gas Pedal	Propulsion Torque Sufficient	Brake Pedal	EPB On	Providing Causes Hazard	Not Providing Causes Hazard	Providing Required for Function	Not Providing Required for Function
AH-R-1	Yes	Yes	Yes	P	*	*	*	No		x		
AH-R-2	Yes	Yes	Yes	N	*	*	Not Pressed	No	x			
AH-R-3	Yes	Yes	Yes	N	*	*	Pressed	No	x			
AH-R-4	Yes	Yes	Yes	R,D,L	Pressed	Yes	*	No		x		x
AH-R-5	Yes	Yes	Yes	R,D,L	Pressed	No	*	No	x			
AH-R-6	Yes	Yes	Yes	R,D,L	Not Pressed	*	*	No	x			
AH-R-7	Yes	Yes	No	*	*	*	*	No	x			
AH-R-8	No	Yes	*	*	*	*	*	No	x			
AH-R-9	*	No	*	*	*	*	*	No				
AH-R-10	*	*	*	*	*	*	*	Yes				

Table 24: AH – APPLY EPB Context Table

Context ID	AH Enabled	Hold-Mode	Driver Present	PRNDL	Vehicle On	Providing Causes Hazard	Not Providing Causes Hazard	Providing Required for Function	Not Providing Require for Function
AH-EPB-1	Yes	Yes	Yes	!=N	Yes				x
AH-EPB-2	Yes	Yes	Yes	N	Yes			x	
AH-EPB-3	Yes	Yes	No	*	Yes		x		
AH-EPB-4	No	Yes	*	*	Yes		x		
AH-EPB-5	*	Yes	*	*	No		x		
AH-EPB-6	*	No	*	*	*	x			

Following the initial completion of STPA Step 1 for Auto-Hold, a conflict was identified surrounding the RELEASE Command. Specifically, that issuing RELEASE upon driver de-activation of AH may violate AH's responsibility to hold the vehicle still if no other mechanism (driver's foot, park pawl, or EPB) was prepared to do so. Resolving this conflict, and others like it, is necessary to design an AH feature that is both safe and functional. The process of populating, evaluating, and reducing the context tables forces the resolution of many of these conflicts. In this particular example the scenario where the Driver disables AH while it is engaged is captured by AH-R-8[10]. It is noted that providing RELEASE in this instance may be hazardous as explained above. To resolve this conflict, AH-EPB specifies that the EPB should be applied in this instance which then moves the RELEASE logic to AH-R-10. This set of specifications provides a means for AH to disengage when the driver requests, by applying the EPB to hold the vehicle at rest.

Another interesting result from the Auto-Hold context tables stems from AH-AP-3. This row in the context table states that when the system is in HOLD-MODE, the gas pedal is pressed, and the wheels begin to rotate, the ADDITIONAL-PRESSURE command should be issued. The result is that the AH feature will be attempting to hold the vehicle when the Driver may be attempting to accelerate. It is not the opinion of the author that automation should fight the human operator, which is aligned with the safety constraint *SC-AH-3: AH shall not interfere with the driver's ability to accelerate.* However, the logic in Table 22 concerns issuing the ADDITIONAL-PRESSURE command. The entering or exiting of HOLD-MODE is considered in Table 21. We must consider all four context tables together to make conclusions about the feature behavior in this scenario. In this scenario, the driver attempting to accelerate the vehicle from a held state will put the RELEASE logic in AH-R-4, which states that the system must issue RELEASE. At that point, the feature will no longer be engaged and ADDITIONAL-PRESSURE will not be issued to comply with AH-AP-5. If for some reason, RELEASE is not issued, AH will remain in HOLD-MODE and therefore must continue to keep the vehicle at a standstill–not doing so will violate the AH safety constraint *SC-AH-5: AH must keep the vehicle at a standstill when in HOLD-MODE*.

It is worthy to note the increased precision that this formal method provides over the safety constraints of the basic method presented in Section 3.4.2. More important than the increase in precision, the results of the formal method can be used to validate the results of the first iteration and identify UCAs that may have been missed using the informal method. Both the formal and informal methods may

[10] AH-R-8 is a Context ID from Table 23. A similar format will be used to refer to specific rows of the context tables for the rest of the thesis: CONTROLLER-COMMAND-ROW where the command may be abbreviated.

guide the analyst to useful results and either may be more appropriate at a given point in the design process. It is the opinion of the author and other team members that completing the basic method at the outset is a valuable exercise as it builds a common understanding of the system that guides the formal analysis.

From the context tables, requirements may be automatically generated to enforce behavior that is required for both safety and functionality. The SpecTRM-RL tables below specify when Auto-Hold must issue a particular command to satisfy both safety constraints and functional goals. The tables are read as AND-OR tables, such that the operator AND is applied between the rows in a given column, and the OR operator is applied between each column. In other words, each column represents a unique requirement. Each column is marked at the top with either a red ‘S’ or green ‘F’ to indicate the requirements traces to a safety constraint or functional goal, or both.

Triggering Conditions for AH HOLD Command:

		S-F
AH Enabled =	Yes	T
	No	
Hold-Mode =	Yes	
	No	T
Driver Present =	Yes	T
	No	
PRNDL =	P	
	R	
	N	
	D | L	T
Gas Pedal =	Pressed	
	Not Pressed	T
Brake Pedal =	Pressed	T
	Not Pressed	
Wheels Rotating =	Yes	
	No	T

Figure 13: AH HOLD - SpecTRM-RL Table

Triggering Conditions for AH ADDITIONAL-PRESSURE Command:

		S-F	S
Hold-Mode =	Yes	T	T
	No		
Gas Pedal =	Pressed		T
	Not Pressed	T	
Wheels Rotating =	Yes	T	T
	No		

Figure 14: AH ADDITIONAL-PRESSURE - SpecTRM-RL Table

Triggering Conditions for AH RELEASE Command:

		S	S-F	S-F
AH Enabled =	Yes	T	T	T
	No			
Hold-Mode =	Yes	T	T	T
	No			
Driver Present =	Yes	T	T	T
	No			
PRNDL =	P	T		
	R		T	
	N			
	D | L			T
Gas Pedal =	Pressed		T	T
	Not Pressed			
Brake Pedal =	Pressed			
	Not Pressed			
Propulsion Torque Sufficient =	Yes		T	T
	No			
EPB On =	Yes			
	No	T	T	T

Figure 15: AH RELEASE - SpecTRM-RL Table

Triggering Conditions for AH APPLY EPB Command:

		F	S	S	S
AH Enabled =	Yes	T	T		
	No			T	
Hold-Mode =	Yes	T	T	T	T
	No				
Driver Present =	Yes	T			
	No		T		
PRNDL =	P				
	R				
	N	T			
	D				
	L				
Vehicle On =	Yes	T	T	T	
	No				T

Figure 16: AH APPLY EPB - SpecTRM-RL Table

It is interesting to note the number of PMVs included in the requirements model for each control action. For instance, the requirements model for the ADDITITIONAL-PRESSURE command only includes 3 PMVs while the requirements models for HOLD and RELEASE include 7 and 8, respectively. The reason for this is that issuing the HOLD or RELEASE command involves shifting control of, and responsibility for, the brakes from one controller to another. The ADDITIONAL-PRESSURE command is issued when AH already has control of the brakes and is responsible for keeping the vehicle at rest, no mode change occurs. Requirements regarding commands that involve changes in the control mode of the larger vehicle system will likely be more complex than those that do not.

The SpecTRM-RL tables used here are black-box models of requirements in a state-machine-based language [13]. The requirements are functional in nature and do not contain design and implementation details. One goal of this thesis is to demonstrate the application of STPA during concept development and its potential to assist in safety guided design. These requirements may be refined as the design process continues. One means of refinement is to breakout PMVs into more detail. E.g. “Driver present” may be defined as a function of inputs including the driver’s seat belt, the door sensor, and a weight sensor in the seat. The requirements may also be refined with respect to timing (race conditions) to ensure that safe behavior is also acceptable from the user’s perspective. For instance, the ‘wheels not rotating’ condition required to issue HOLD may be further specified with a time requirement, greater than 2 seconds, to ensure that momentary stops do not engage the feature and bring the vehicle to rest when it may be inappropriate.

Refinement and abstraction are useful tools throughout the design process; however, it is important to not lose information necessary for traceability. For example, while the high-level logic governing ADDITIONAL-PRESSURE (Figure 14) is already less complex than other commands, it could be reduced further without effect on system behavior. However, doing so would eliminate the distinction between cases that deal only with safety vs. those that concern both safety and functionality. The logic should be kept in its current form so that future revisions can trace the specifications to safety constraints and functional goals that may change over time.

4.5. Engine Stop-Start

As was done for Auto-Hold, the unsafe control actions previously identified for ESS may be refined and formalized. The results of this process are shown here, beginning with the definition of PMVs unique to the ESS system in Table 25. The combination of these with the common context variables (Table 19) are used to identify the contexts in which ESS will issue commands [STOP, START, APPLY EPB]. These contexts are then evaluated to assess their impact on system safety and functionality.

Table 25: ESS Module (specific) Process Model Variables

ESS Enabled ***Yes:*** *ESS is allowed to auto-stopped mode* ***No:*** *ESS is not allowed to enter auto-stopped mode – initial state* ***Implementation:*** *Menu Option*	**[Yes, No]**
Auto-Stopped ***Yes:*** *ESS has shut off the engine* ***No:*** *ESS has not shut off the engine* ***Implementation:*** *Engine Control Module*	**[Yes, No]**
Vehicle Held ***Yes:*** *The vehicle is secured from longitudinal motion by one or more of the following: Electronic Park Brake, Park Gear, Service Brake* ***No:*** *The vehicle is free to move longitudinally* ***Implementation:*** *Electronic Park Brake, Park Gear, Service Brake*	**[Yes, No]**
Restart Possible ***Yes:*** *The battery has enough charge to restart the engine with some TBD margin* ***No:*** *The battery does not have enough charge to restart the engine and some TBD margin* ***Implementation:*** *Feedback from Propulsion System*	**[Yes, No]**
Auxiliary Power Needs ***Above:*** *The battery cannot satisfy the current demand for electrical power from vehicle subsystems* ***Below:*** *The battery can satisfy the current demand for electrical power from all vehicle subsystems* ***Implementation:*** *Feedback from the vehicle*	**[Above, Below]**

Continuing the process established with Auto-Hold, the context tables are presented as a set with discussion regarding system behavior following.

Table 26: ESS – STOP Context Table

Context ID	ESS Enabled	Auto-Stopped	Driver Present	PRNDL	Gas Pedal	Vehicle Held	Wheels Rotating	Restart Possible	Auxiliary Power Needs	Providing Causes Hazard	Not Providing Causes Hazard	Providing Required for Function	Not Providing Required for Function	FMVSS 102
ESS-STOP-1	No	*	*	*	*	*	*	*	*	x				
ESS-STOP-2	Yes	Yes	*	*	*	*	*	*	*					
ESS-STOP-3	Yes	No	No	*	*	*	*	*	*	-	-	-	-	-
ESS-STOP-4	Yes	No	Yes	P, R, N	*	*	*	*	*	x				prohibited
ESS-STOP-5	Yes	No	Yes	D, L	Yes	*	*	*	*	x			x	
ESS-STOP-6	Yes	No	Yes	D, L	No	No	*	*	*	x				
ESS-STOP-7	Yes	No	Yes	D, L	No	Yes	Yes	*	*	x				
ESS-STOP-8	Yes	No	Yes	D, L	No	Yes	No	No	*	x			x	
ESS-STOP-9	Yes	No	Yes	D, L	No	Yes	No	Yes	Above				x	
ESS-STOP-10	Yes	No	Yes	D, L	No	Yes	No	Yes	Below			x		

Table 27: ESS – APPLY EPB Context Table

Context ID	Auto-Stopped	Driver Present	Restart Possible	Providing Causes Hazard	Not Providing Causes Hazard	Providing Required for Function	Not Providing Required for Function
ESS-EPB-1	No	*	*	x			x
ESS-EPB-2	Yes	No	*		x		
ESS-EPB-3	Yes	Yes	No		x		
ESS-EPB-4	Yes	Yes	Yes				

Table 28: ESS – RESTART Context Table

Context ID	ESS Enabled	Auto-Stopped	Driver Present	PRNDL	Gas Pedal	Vehicle Held	Wheels Rotating	Restart Possible	Auxiliary Power Needs	Providing Causes Hazard	Not Providing Causes Hazard	Providing Required for Function	Not Providing Required for Function	FMVSS 102
ESS-RESTART-1	No	*	*	*	*	*	*	*	*	x				
ESS-RESTART-2	Yes	No	*	*	*	*	*	*	*	x				
ESS-RESTART-3	Yes	Yes	No	*	*	*	*	*	*	x			x	
ESS-RESTART-4	Yes	Yes	Yes	*	*	*	*	No	*	-	-	-	-	-
ESS-RESTART-5	Yes	Yes	Yes	P, N	*	*	*	Yes	Above			x		
ESS-RESTART-6	Yes	Yes	Yes	P, N	*	*	*	Yes	Below				x	
ESS-RESTART-7	Yes	Yes	Yes	R	*	Yes	*	Yes	*					required
ESS-RESTART-8	Yes	Yes	Yes	R	*	No	*	Yes	*		x			prohibited
ESS-RESTART-9	Yes	Yes	Yes	D, L	Yes	Yes	Yes	Yes	*		x			
ESS-RESTART-10	Yes	Yes	Yes	D, L	Yes	Yes	No	Yes	*	x		x		
ESS-RESTART-11	Yes	Yes	Yes	D, L	Yes	No	Yes	Yes	*		x	x		
ESS-RESTART-12	Yes	Yes	Yes	D, L	Yes	No	No	Yes	*	x		x		
ESS-RESTART-13	Yes	Yes	Yes	D, L	No	Yes	Yes	Yes	*		x			
ESS-RESTART-14	Yes	Yes	Yes	D, L	No	Yes	No	Yes	Below				x	
ESS-RESTART-15	Yes	Yes	Yes	D, L	No	Yes	No	Yes	Above			x		
ESS-RESTART-16	Yes	Yes	Yes	D, L	No	No	Yes	Yes	*		x			
ESS-RESTART-17	Yes	Yes	Yes	D, L	No	No	No	Yes	Below	x				
ESS-RESTART-18	Yes	Yes	Yes	D, L	No	No	No	Yes	Above	x		x		

The ESS RESTART command restarts the engine, after being stopped by ESS, to provide power to the vehicle systems before the driver resumes driving. Unlike ESS STOP, there are many contexts in which issuing ESS RESTART is required to meet both safety constraints and functional goals. Interestingly, there are several instances when these constraints and goals conflict – both with each other and with current FMVSS regulation [14]. Inspection of the ESS RESTART context table yields conflicts in the following rows: 8, 10, 12, and 18. These conflicts are described briefly here and in greater depth in Appendix A.

The first conflict, ESS-RESTART-8, occurs when issuing RESART is required to provide electrical power to the assisted steering and braking systems, but doing so violates the FMVSS 102 requirement S.3.1.3.1 c2 – "The engine may automatically restart in reverse gear only if the vehicle satisfies (1) and (2)" where (2) is, "When the engine is automatically stopped in a forward drive shift position and the driver selects Reverse, the engine does not start automatically if the service brake is not applied." This conflict is explored further in the next chapter when considering the interactions between controllers.

The second conflict, ESS-RESTART-10, may occur when a driver attempts to take off while AUTO-STOPPED on a hill. The driver will apply the gas pedal and once the engine is started there will be some delay before the propulsion torque is greater than both the force of gravity down the incline and any braking force still applied. During this delay, the propulsion torque will fight the braking force holding the vehicle. From a safety perspective, this should be avoided so that there is no unnecessary lurching when the brakes release and to minimize unnecessary wear and tear on the braking system. Functionally, the engine should be started (and producing torque) in response to the driver's request via the gas pedal. This conflict presents the designer of the system with a challenge - design the system such that it minimizes lurching and facilitates a smooth takeoff. This will require a rapid startup time for the engine and some intelligence in the electronic throttle. The next conflict, SS-RESTART-12, is similar, the only difference being that the vehicle is not actively held.

The final conflict associated with the RESTART command, ESS-RESTART-18, comes from considering ESS's impact on auxiliary vehicle systems (accessories). Functionally, ESS should not inhibit the operation of auxiliary systems and thus must restart when their power needs rise above what the battery can provide. If this occurs when the vehicle is not actively held, starting the engine will cause the vehicle to move when the driver may not be fully engaged and prepared to control the vehicle's motion. Design engineers must consider this and implement logic into the controls that anticipates drain of the battery and responds accordingly.

As was done with Auto-Hold, executable requirements are generated from the above context tables. In addition to safety and functionality, regulatory requirements are generated and identified with an orange R. Requirements that present a conflict between goals are flagged with an asterisk (*) to indicate they must be revisited before the design is developed further.

The requirements below indicate that while the STOP command should only be issued in one context, necessitated by a functional goal, RESTART must be issued in many contexts associated with functionality, safety, and regulatory compliance. Intuitively this makes sense as ESS puts the vehicle in a commonly accepted 'safe-state' – at rest with no power generation. While entering a 'safe-state' may be relatively straightforward, determining when to leave a known safe-state may be highly complicated. This is reflected in the relative number of requirements that are given for STOP and RESTART.

It is also interesting to note that ESS only issues the EPB when required for safety. The addition of the APPLY EPB command for safety and the complexity of the RESTART requirements indicate that the apparent simplicity of ESS, suggested by the single requirement for issuing STOP, may be misleading.

Triggering Conditions for ESS STOP Command:

		F
ESS Enabled =	Yes	T
	No	
Auto-Stopped =	Yes	
	No	T
Driver Present =	Yes	T
	No	
PRNDL =	P	
	R	
	N	
	D | L	T
Gas Pedal =	Pressed	
	Not Pressed	T
Vehicle Held =	Yes	T
	No	
Wheels Rotating =	Yes	
	No	T
Restart Possible =	Yes	T
	No	
Auxiliary Power Needs =	Above	
	Below	T

Figure 17: ESS STOP - SpecTRM-RL Table

Triggering Conditions for ESS RESTART Command:

		F	F	R*	S-R*	S	F*	S-F	F*	S	F	S	F*
ESS Enabled =	Yes	T	T	T	T	T	T	T	T	T	T	T	T
	No												
Auto-Stopped =	Yes	T	T	T	T	T	T	T	T	T	T	T	T
	No												
Driver Present =	Yes	T	T	T	T	T	T	T	T	T	T	T	T
	No												
PRNDL =	P	T											
	R			T	T								
	N		T										
	D | L					T	T	T	T	T	T	T	T
Gas Pedal =	Pressed					T	T	T	T				
	Not Pressed									T	T	T	T
Vehicle Held =	Yes			T		T	T			T	T		
	No				T			T	T			T	T
Wheels Rotating =	Yes					T		T		T		T	
	No						T		T		T		T
Restart Possible =	Yes	T	T	T	T	T		T	T	T	T	T	T
	No						T						
Auxiliary Power Needs =	Above	T	T										T
	Below												

Figure 18: ESS RESTART - SpecTRM-RL Table

Triggering Conditions for ESS APPLY EPB Command:

		S	S
Auto-Stopped =	Yes	T	T
	No		
Driver Present =	Yes		T
	No	T	
Restart Possible =	Yes		
	No		T

Figure 19: ESS APPLY EPB - SpecTRM-RL Table

4.6. Adaptive Cruise Control with Stop & Go

Following the same process as AH and ESS, STPA Step 1 is iterated with greater formality for ACC w/SG. An additional set of PMVs is defined in Table 29 that completes the process model of the ACC w/SG control module. The relevant context tables follow with discussion and presentation of executable requirements for the operation of ACC w/SG.

Four ACC w/SG control actions are considered here: ENGAGE, ACCELERATE, BRAKE, and DISENGAGE. The ENGAGE and DISENGAGE commands are not physically realized control actions; however, they cause the ACC w/SG controller to enter and exit operation and thus take and release control of the propulsion and braking systems. These mode changes are included in this analysis so that the requirements for each controller are complete, in that they fully define the functionality of each feature.

Table 29: ACC w/SG (unique) Process Model Variables

ACC Active ***Yes:*** *SS is allowed to engage* ***No:*** *SS is not allowed to engage* ***Implementation:*** *Driver Button/Switch*	**[Yes, No]**
ACC Engaged ***Yes:*** *ACC has engaged and is allowed to influence braking and propulsion* ***No:*** *ACC has not engaged* ***Implementation:*** *ACC Control Module Internal State*	**[Yes, No]**
Driver Pedal Input ***Yes:*** *Either the Brake pedal travel is at least X mm or the Gas pedal travel is at least Y mm* ***No:*** *Brake pedal travel and Gas pedal travel are both less than X mm and Y mm respectively* ***Implementation:*** *Brake Pedal Switch and Accelerator Pedal Potentiometer*	**[Yes, No]**
Target Locked ***Yes:*** *A same-lane target is identified and being actively tracked* ***No:*** *A same-lane target is not both identified and actively tracked* ***Implementation:*** *Forward Looking Vision System*	**[Yes, No]**
Distance >= Threshold ***Yes:*** *The distance to the target is greater than or equal to the set distance*[11] ***No:*** *The distance to the target is less than the set distance* ***Implementation:*** *Forward Looking Vision System*	**[Yes, No]**

[11] It is possible that the set distance may have an acceptable +/- margin and thus interpreted as a range

Speed >= Threshold ***Yes:*** *The vehicle speed is greater than or equal to the set speed*[12] ***No:*** *The vehicle speed is less than the set speed* ***Implementation:*** *Wheels Speed Sensors & Set Value*	**[Yes, No]**

The context tables and requirements for ACC w/SG are presented on the following pages. At this level of abstraction, there are no explicit conflicts between safety constraints and functional goals for SG; however, there is tight coupling between safety and functionality in the DECELERATE command logic (Table 32). There are three instances in which issuing the DECELERATE command is required for both safety and functionality: SG-DECELERATE-6; SG-DECELERATE-7; SG-DECELERATE-9. In the first of these three contexts, SG-DECELERATE-6, the command DECELERATE is required to decelerate the vehicle before it rear-ends the vehicle it is following. The other two contexts, SG-DECELERATE-7/9, are similar in that failing to issue the DECELERATE command will result in over-speed which violates the Driver-defined speed and may increase the vehicle speed such that it is difficult to control. This coupling is also seen in the requirements generated for the DECELERATE command (Figure 22), all three requirements are required for both safety and functionality. In contrast, the requirements for the other three command issued by ACC w/SG (ENGAGE, ACCELERATE, DISENGAGE) have distinct requirements for functionality and safety.

[12] It is possible that the set speed may have an acceptable +/- margin and thus interpreted as a range

Table 30: ACC w/SG ENGAGE Context Table

Context ID	ACC Enabled	ACC Engaged	Driver Present	PRNDL	Driver Brake Pedal Input	Wheels Rotating	Providing Causes Hazard	Not Providing Causes Hazard	Providing Functional	Not Providing Functional
SG-E-1	No	*	*	*	*	*	x			
SG-E-2	Yes	Yes	*	*	*	*				x
SG-E-3	Yes	No	No	*	*	*	x			
SG-E-4	Yes	No	Yes	!=D	*	*	x			x
SG-E-5	Yes	No	Yes	D	Pressed	*	x			
SG-E-6	Yes	No	Yes	D	Not Pressed	Yes			x	
SG-E-7	Yes	No	Yes	D	Not Pressed	No	x			

Table 31: ACC w/SG ACCELERATE Context Table

Context ID	ACC Enabled	ACC Engaged	Driver Present	PRNDL	Driver Pedal Input	Target Locked	Distance >= Threshold	Speed >= Threshold	Providing Causes Hazard	Not Providing Causes Hazard	Providing Functional	Not Providing Functional
SG-ACCEL-1	No	*	*	*	*	*	*	*	x			
SG-ACCEL-2	Yes	No	*	*	*	*	*	*	x			
SG-ACCEL-3	Yes	Yes	No	*	*	*	*	*	x			
SG-ACCEL-4	Yes	Yes	Yes	!=D	*	*	*	*	x			x
SG-ACCEL-5	Yes	Yes	Yes	D	Yes	*	*	*	x			
SG-ACCEL-6	Yes	Yes	Yes	D	No	Yes	No	*	x			x
SG-ACCEL-7	Yes	Yes	Yes	D	No	Yes	Yes	Yes	x			x
SG-ACCEL-8	Yes	Yes	Yes	D	No	Yes	Yes	No			x	
SG-ACCEL-9	Yes	Yes	Yes	D	No	No	*	Yes	x			x
SG-ACCEL-10	Yes	Yes	Yes	D	No	No	*	No			x	

Table 32: ACC w/SG BRAKE Context Table

Context ID	ACC Enabled	ACC Engaged	Driver Present	PRNDL	Driver Pedal Input	Target Locked	Distance >= Threshold	Speed >= Threshold	Providing Causes Hazard	Not Providing Causes Hazard	Providing Functional	Not Providing Functional
SG-DECEL-1	No	*	*	*	*	*	*	*	x			
SG-DECEL-2	Yes	No	*	*	*	*	*	*	x			
SG-DECEL-3	Yes	Yes	No	*	*	*	*	*				
SG-DECEL-4	Yes	Yes	Yes	!=D	*	*	*	*	x			
SG-DECEL-5	Yes	Yes	Yes	D	Yes	*	*	*	x			
SG-DECEL-6	Yes	Yes	Yes	D	No	Yes	No	*		x	x	
SG-DECEL-7	Yes	Yes	Yes	D	No	Yes	Yes	Yes		x	x	
SG-DECEL-8	Yes	Yes	Yes	D	No	Yes	Yes	No				x
SG-DECEL-9	Yes	Yes	Yes	D	No	No	*	Yes		x	x	
SG-DECEL-10	Yes	Yes	Yes	D	No	No	*	No				x

Table 33: ACC w/SG DISENGAGE Context Table

Context ID	ACC Enabled	ACC Engaged	Driver Present	PRNDL	Driver Brake Pedal Input	Target Locked	Wheels Rotating	Providing Causes Hazard	Not Providing Causes Hazard	Providing Functional	Not Providing Functional
SG-DE-1	No	Yes	*	*	*	*	*			x	
SG-DE-2	Yes	No	*	*	*	*	*			x	
SG-DE-3	Yes	Yes	No	*	*	*	*				
SG-DE-4	Yes	Yes	Yes	!=D	*	*	*			x	
SG-DE-5	Yes	Yes	Yes	D	Pressed	*	*		x		
SG-DE-6	Yes	Yes	Yes	D	Not Pressed	*	Yes				x
SG-DE-7	Yes	Yes	Yes	D	Not Pressed	No	No		x		
SG-DE-8	Yes	Yes	Yes	D	Not Pressed	Yes	No				x

Triggering Conditions for ACC w/SG ENGAGE:

		F
ACC Enabled =	No	
	Yes	T
ACC Engaged =	No	T
	Yes	
Driver Present =	No	
	Yes	T
PRNDL =	!=D	
	D	T
Driver Brake Pedal Input =	Yes	
	No	T
Target Locked =	No	
	Yes	
Speed Above Threshold =	Yes	
	No	T

Figure 20: ACC w/SG - SpecTRM-RL Table

Triggering Conditions for ACC w/SG ACCELERATE Command:

		F	F
ACC Enabled =	No		
	Yes	T	T
ACC Engaged =	No		
	Yes	T	T
Driver Present =	No		
	Yes	T	T
PRNDL =	!=D		
	D	T	T
Driver (Either) Pedal Input =	Yes		
	No	T	T
Target Locked =	No		
	Yes		T
Distance Above Threshold =	No	T	T
	Yes		
Speed Above Threshold =	Yes		
	No	T	

Figure 21: ACC w/SG ACCELERATE - SpecTRM-RL Table

Triggering Conditions for ACC w/SG DECELERATE Command:

		S-F	S-F	S-F
ACC Enabled =	No			
	Yes	T	T	T
ACC Engaged =	No			
	Yes	T	T	T
Driver Present =	No			
	Yes	T	T	T
PRNDL =	!=D			
	D	T	T	T
Driver (Either) Pedal Input =	Yes			
	No	T	T	T
Target Locked =	No			T
	Yes		T	
Distance Above Threshold =	No	T	T	
	Yes			
Speed Above Threshold =	Yes			T
	No	T		

Figure 22: ACC w/SG DECELERATE - SpecTRM-RL Table

Triggering Conditions for ACC w/SG DISENGAGE Command:

		F	F	F	S	S
ACC Enabled =	No	T				
	Yes		T	T	T	T
ACC Engaged =	No		T			
	Yes	T		T	T	T
Driver Present =	No					
	Yes			T	T	T
PRNDL =	!=D			T		
	D				T	T
Driver Brake Pedal Input =	Yes				T	
	No					T
Target Locked =	No					T
	Yes					
Speed Above Threshold =	Yes					
	No					T

Figure 23: ACC w/SG DISENGAGE - SpecTRM-RL Table

4.7. Summary and Next Steps

It has been demonstrated that iterating STPA Step 1 with a formal method can increase the specificity of the constraints and generate executable requirements while ensuring that all possible scenarios are considered. This case study is being completed at a level of abstraction appropriate during concept development, additional detail may be added as iterations continue through the design process. If this process is repeated at each level of design development, the flow down of safety hazards and functional goals will ensure that the system meets its high level requirements upon completion.

Review of the requirements presented for the three features reveals an interesting trend, i.e., that the control actions to engage a feature are required to be issued in fewer instances than the control action to disengage the feature. This contrast can be seen quite clearly by inspecting the requirements for ESS STOP and RESTART, Figure 17 and Figure 18, respectively. There is only one requirement for ESS to issue STOP, which engages the feature, but there are twelve requirements for ESS to issue RESTART, which exits the feature from AUTO-STOPPED mode. Additionally, the single requirement to issue STOP is only required for functionality, while the requirements to issue RESTART stem from functional, safety, and regulatory constraints. This pattern is repeated for ACC w/SG, where the 'positive' commands, ENGAGE and ACCELERATE, have few requirements that stem solely from functional goals. In contrast, the 'negative' commands, DECELERATE and DISENGAGE, each have a greater number of requirements that stem from safety constraints in addition to functional goals.

The AH feature is interesting in that its control actions for normal operation (HOLD, ADDITIONAL-PRESSURE, and RELEASE) do not have any purely functional requirements, they are all coupled with safety or exist solely to satisfy safety constraints. This means that AH may not be treated simply as an additional convenience feature added onto the existing vehicle platform, it must be regarded as a safety critical feature to be integrated into the vehicle.

These observed trends suggest that while a feature may be simple to engage, once active, the feature may have safety implications that outreach or constrain its functional purpose. ESS is intended to save fuel when the vehicle is already stopped; however, taking away propulsion power has the potential to render the vehicle stuck in place or uncontrollable in the event of rollaway. It is beyond the scope of this thesis to comment on the value evaluation of autonomous features; however, analysis such as what is presented here may be used by those who are responsible for such decisions.

In the context tables for each of the three feature systems, it was noted that there may be both conflicts and tight coupling between the safety constraints and functional goals. Coupling exists when a requirements for safety overlap with requirements for function. This idea of coupling is particularly prevalent for the DECELERATE command associated with ACC w/SG, where all three requirements generated (Figure 22) are necessary for both safety and functionality. The requirements for the ESS RESTART command (Figure 18) provide an example of conflicts between hazards, functional goals, and regulatory requirements. It is necessary to study both the couplings and conflicts between safety constraints and functional goals to understand the behavior of each feature and its impact on the larger vehicle system.

The conflicts in the requirements generally occur in what may be considered "off-nominal" situations, those other than the intended sequence of stimuli and commands for which the feature was developed. For example, the ESS feature should nominally RESTART the engine as the driver is preparing to take off and before the wheels begin to rotate. In this nominal scenario, it is assumed that the vehicle remains in the original transmission range, the auxiliary power needs stay sufficiently low and the driver remains in the vehicle simply waiting for the opportunity to move the vehicle forward. Off-nominal stimuli change the value of the PMVs in the RESTART control logic and place it into a context (state) that leads to a conflict between system level goals.

The analysis thus far has provided a significant amount of insight into the behavior of each system and the results thus far—safety constraints, example causal factors, executable requirements—are useful inputs to the design process. However, the analysis contains the assumption that each feature is deployed independently of the others, i.e. the vehicle only has one of the three features enabled at a time. The goal of this project is to guide the integration of these features rather than design them in isolation. To that end, the next chapter describes a new approach to analyze these features as an integrated system by leveraging the results produced thus far.

Page intentionally left blank.

5. Identifying Conflicts to Prevent Hazardous Interactions

5.1. Overview and Background

This chapter explores a new approach to analyze the integration of multiple control systems within the STPA framework. This case study is the first application of the new approach that is based on a proposal by Thomas [15]. In this chapter, the new approach is applied to a large multiple-controller system, and evaluated for feasibility and effectiveness in these systems.

The goal of the new approach is to leverage the results of STPA Step 1 to identify two types of scenarios: 1) when one controller may issue a command that leads to another controller issuing an unsafe command 2) when one controller performing as designed may prevent another controller from realizing its functional goals. Identifying these scenarios, conflicts between the control systems, presents system designers with information they need to design and implement improved systems.

As mentioned in the summary of Chapter 4, the analysis thus far has focused on the features individually. It has been shown that STPA can analyze each feature and the results indicate that, independently, each feature will accomplish its level purpose. However, there have been some hints that an integrated system may not work seamlessly. For instance, many of the conflicts in the ESS RESTART requirements (Figure 18) occur when the system must respond to a Driver action such as shifting or leaving the vehicle. Similarly, the AH module must issue the EPB when the vehicle is turned off while in HOLD-MODE. If a vehicle has both AH and ESS and the engine is automatically shutoff while in HOLD-MODE, AH will apply the EPB and render the vehicle stuck until the driver intervenes to disengage—taking away the intuitive behavior intended for both features. These are examples of the type of conflicts that need to be identified during concept development and then designed out of the system before implementation.

These conflicts arise for several reasons. Primarily, and perhaps most simply, features that are designed independently do not take each other's behavior into account—and cannot do so once installed on the same vehicle. For example, the AH feature (as presented in previous chapters) considers the Driver and Anti-Lock Brakes (ABS) as the only other controllers of the vehicle's brakes. AH then knows that when the vehicle is stopped (and ABS is irrelevant) it can assume complete control of the vehicle's brakes by closing a valve that captures the brake pressure, in turn holding the vehicle still while also locking out the driver’s ability to release (or increase) the captured pressure. Integrating ACC w/SG into a vehicle with AH introduces another potential controller of the brake system. If AH does not know when ACC w/SG will actuate or take control of the brakes, it cannot make appropriate decisions about those actions itself.

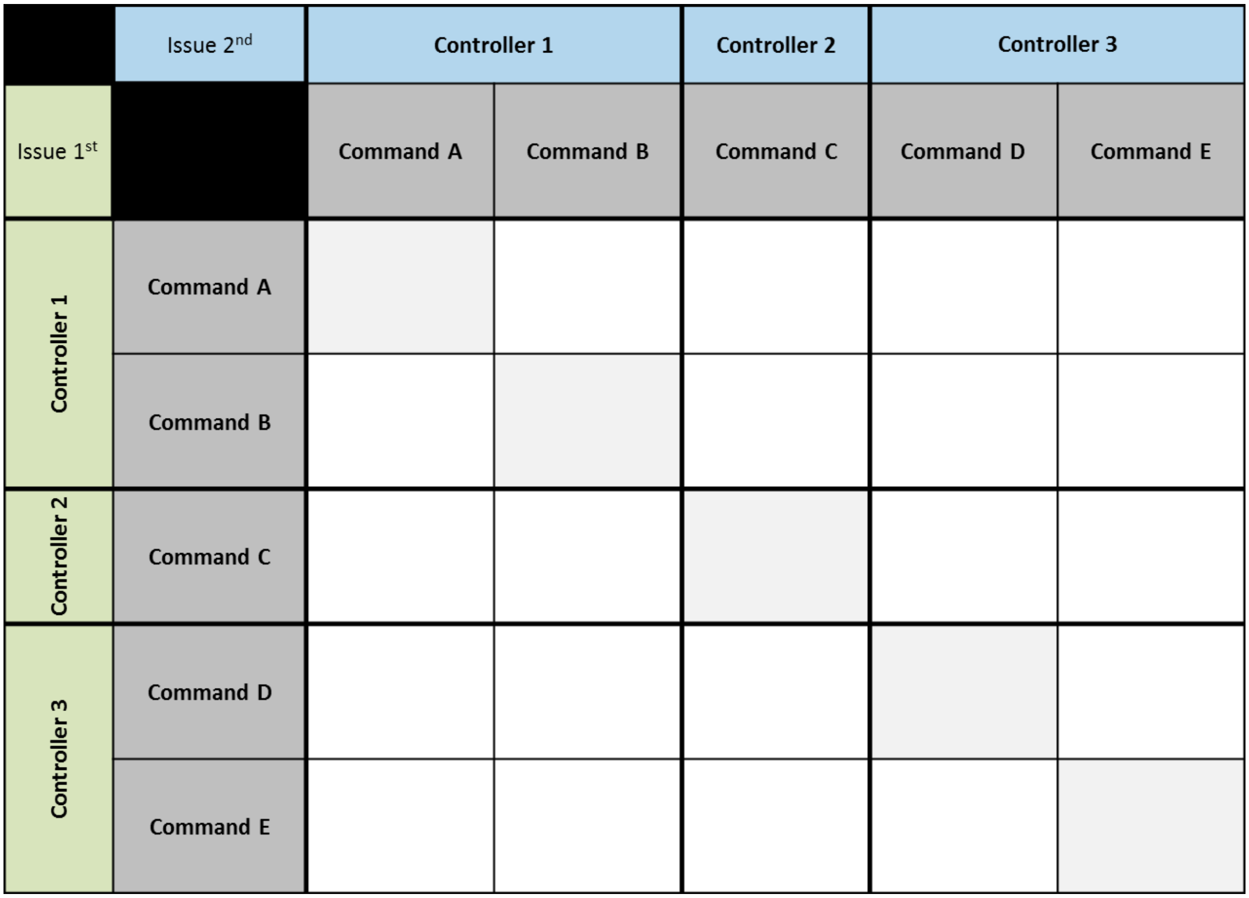
Another source of conflict is violated assumptions. Each feature is designed under a set of assumptions regarding both the larger vehicle system and the operating environment. These assumptions may no longer be valid once multiple features are integrated onto the same vehicle. A simple example of violated assumptions occurs when integrating ESS and ACC w/SG. When implemented in isolation, ACC w/SG assumes that engine power is always available for propulsion, power-assisted brakes, and steering. If ESS shuts off the engine, this assumption is violated and ACC w/SG may not operate correctly due to reduced amount of power available, now limited to that contained in the battery.

The next section proposes a method for leveraging the executable requirements from Chapter 4 to identify when controllers may conflict. After the method is introduced in generic terms, it is applied to the automotive case study and initial results are presented. The summary of this chapter comments on the initial results and how the method may be used in practice.

A method for analyzing multiple controllers with STPA was proposed and demonstrated in [16]. This method builds upon and extends that approach by leveraging the more formal results of the Thomas Method. Some comparison of the new approach to the old will be provided following the results of the case study.

5.2. Method for Analyzing Integrated Systems

To analyze the integration of multiple control systems it is necessary to understand how the issuing of one control action may affect the issuing of another. One way to consider all pairs of control actions is to make a table such as that shown in Figure 24.



	Issue 2nd	Controller 1		Controller 2	Controller 3	
Issue 1st		Command A	Command B	Command C	Command D	Command E
Controller 1	Command A					
	Command B					
Controller 2	Command C					
Controller 3	Command D					
	Command E					

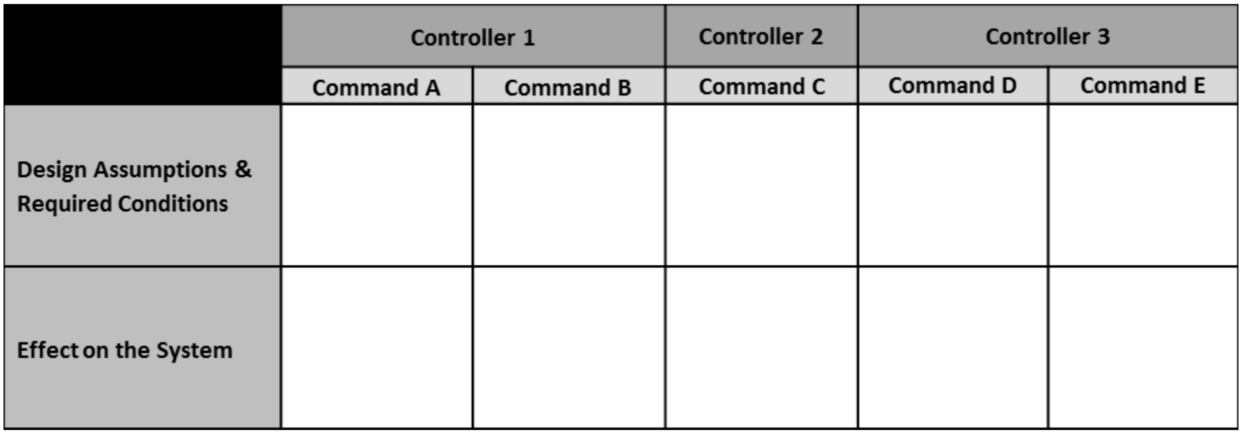
Figure 24: Brute Force Approach for 2 Control Actions

In this table, three controllers (1, 2 and 3) may issue commands (A, B, C, D, and E) within the same system. Each cell in the table represents the scenario in which one command is issued before another. The command corresponding to the cell's row is issued first; the command corresponding to the column is issued second. For a given cell, the issuing of one command before another may be hazardous, dysfunctional, necessary, or trivial. Filling in all the cells in the table will analyze every situation in which one of the five commands is issued before one of the other four. Scenarios that are hazardous of dysfunctional may be mitigated and/or designed out of the system; ones that are necessary may be ensured in the system design.

This brute-force approach will provide the information required; however, it is not scalable and therefore cannot be applied to any system of appreciable complexity. Also, it is very difficult to visualize the intersection space when considering combinations of control actions greater than two. Considering three commands at a time will necessitate a three dimensional table which may be visualized, but higher orders cannot be represented in this format. If *n* is the number of commands and *x* is the number considered together, the table scales according to $O(n^x)$.

While the brute force approach is not practical, a method for generating the same analysis and considering the same command interactions is necessary to ensure a complete set of conflict-free requirements. It was observed earlier that conflicts occur when either design assumptions are violated or one controller's behavior interferes with another. In essence, one controller can have an effect on the system that prevents another controller from operating in a manner that is both safe and functional.

This information can be captured in a *conditions table*. The conditions table is a *2n* table that keeps a record of the design assumptions, required conditions, and effect on the system associated with each command. A generic conditions table is presented in Figure 25.



	Controller 1		Controller 2	Controller 3	
	Command A	Command B	Command C	Command D	Command E
Design Assumptions & Required Conditions					
Effect on the System					

Figure 25: Conditions Table

One way that conflicts occur is when the effects of one command violate the assumptions and conditions of another. When this violation occurs, one or more controllers may be prevented from realizing their intended functions as the requirements for safely and effectively issuing a command are no longer met. Depending on the particular commands involved, this may have both performance and safety implications. An example from the automotive case study, ESS issuing the STOP command violates one of the conditions required for ACC w/SG's ACCELERATE command to be realized, i.e., that propulsion power is available which implies the engine is running.

Another way that conflicts occur is when the effects of one command satisfy the assumptions and conditions of another, triggering the second command to be issued at an unanticipated time or in an uncoordinated manner. This type of conflict is more subtle than the first that arises from commands effectively prohibiting each other. In this case, the requirements for an individual command to be safe and functional are still met; however, from a system perspective, the control amongst controllers is uncoordinated such that they may be competing against each other.

The conditions table can be searched for instances when these potential conflicts occur to provide the same information as the brute-force method described above, but in a format that grows on the order $O(2n)$ and thus is scalable for large systems. For small systems, the cells may be populated with natural language text, as was done for the UCA tables in Chapter 3, and then visually inspected for conflicts. A more powerful approach is to formalize the information in the table and allow a computer to search for, and flag, potential conflicts.

This approach may leverage the results of STPA Step 1 as the required conditions are equivalent to the executable requirements generated in Chapter 4. The “Effects on the System” and “Design Assumptions” are to be supplied by the analyst who is, or is closely working with, an engineer intimately familiar with the particular controller and system. In the case study, it was observed that identifying the “Effect(s) on the System” first prompted the analyst to think of “Design Assumptions” that may otherwise be left out.

5.3. Application to Case Study

5.3.1. Setup

This approach was applied to the three automotive features and the driver as an integrated system. The required conditions for the three features are equivalent to the executable requirements presented in Chapter 4; those for the Driver were specified during the development of the conditions table. Following the required conditions, the "Effect" of each command was specified in terms of the global set of Process Model Variables put forth in Chapter 4. Finally, the design assumptions for each command were specified, often prompted by inspecting the conditions for and effects of other controllers and commands. Iterative completion of the conditions table helps ensure that all known constraints (required conditions), assumptions, and effects are included. Some abstraction has been used when combining the requirements and assumptions to make the tables readable. For instance, Vehicle Held is an abstract variable that may be satisfied by either the Park gear, EPB or service brake.

To ensure readability and meet formatting constraints, the conditions table is presented in parts by controller. These four tables may be combined into one master conditions table as would be done by a computer for automated review. These tables are presented in pseudo-natural language for readability; formal notation, such as the SpecTRM-RL tables, is also possible. Prior to each table, a brief explanation of rationale is given, including how the results of previous chapters are leveraged.

The conditions, assumptions, and effects associated with Auto-Hold's commands are given in Table 34. The conditions required correspond to the constraints from Table 8 and triggering conditions in Figure 13–Figure 16. Combined, the constraints for the four commands (the first column) ensure that Auto-Hold will prevent the vehicle from rolling once it comes to a stop by capturing the brake pressure and not releasing until either the driver has taken control of the vehicle's motion or the vehicle is secured by another system (e.g. the EPB). To accomplish these goals, AH has the authority to take control of the braking systems, draw power from the battery, and apply the EPB. It does not have the authority to release the EPB or control other vehicle systems.

Table 34: AH Conditions Table

		Design Assumption & Conditions Required	**Effect on the System**
Auto-Hold	**HOLD (HOLD-MODE)**	**AH Enabled:** Yes **Wheels Rotating:** No **Brakes:** On **Driver Present:** Yes **Gas Pedal:** No **Range:** D **|** L	**Brakes:** Applied by AH
	RELEASE	**Brakes:** Applied by AH **Propulsion Torque:** Yes **AND Driver Present:** Yes **OR Vehicle Held:** Yes (i.e. **Range:** Park **|** **EPB:** Yes) **Range:** D **|** P	**Brakes:** Not Applied by AH
	ADDITIONAL-PRESSURE	**Brakes:** Applied by AH **Wheels Rotating:** Yes **Electrical Power:** Available (i.e. **Engine:** On **|** **Battery:** High) **Brake Pressure:** <Max	**Battery:** Reduce Charge **Brake Force:** Increased
	APPLY EPB	**AH Enabled:** Yes **Brakes:** Applied by AH **Driver Present**: Yes **AND Range:** Neutral **OR Driver Present:** No **Vehicle On:** Yes **– OR –** **AH Enabled:** No **Brakes:** Applied by AH **Vehicle On:** Yes **– OR –** **Brakes:** Applied by AH **Vehicle On**: No	**EPB:** Applied

ESS is similar to AH in that it can only control a limited number of vehicle systems including the engine and the EPB (apply only). ESS controls these systems to reduce idle torque by shutting off the engine when it is at a stop and restarting when either requested by the driver or engine power is required for vehicle control. The conditions and assumptions come from the constraints in Table 12 and triggering conditions of Figure 17–Figure 19. Table 35 presents the conditions table for the ESS controller.

Table 35: ESS Conditions Table

		Design Assumption & Conditions Required	**Effect on the System**
Engine Stop-Start	**STOP (AUTO-STOPPED)**	**SS Enabled:** Yes **AUTO-STOPPED:** Yes **Vehicle Held:** Yes (i.e. **Brake:** On **|** **EPB:** Yes) **Restart Possible:** Yes (i.e. **Battery Charge:** High) **Driver Present:** Yes **Gas Pedal:** No **Auxiliary Power Needs:** Low **Range:** !=P,R,N	**Engine:** Off **Idle Torque:** No **Electrical Power**: Off **AUTO-STOPPED:** Yes
	RESTART	**Vehicle Held:** Yes (i.e. **Brakes:** On **|** **Range:** Park **|** **EPB:** Yes) **Wheels Rotating:** No **Restart Possible:** Yes (i.e. **Battery Charge:** High) **Driver Present:** Yes **Range**:!=P,R,N	**Engine:** On **Idle Torque:** Yes **Electrical Power:** On - power reduced ~2s **AUTO-STOPPED:** No
	APPLY EPB	**Auto-Stopped:** Yes **Driver Present:** Yes **AND Restart Possible:** Yes **OR Restart Possible:** No	**EPB:** Applied

ACC w/SG has greater authority over the vehicle dynamics than either AH or ESS as it can actively control both the vehicle’s propulsion and braking systems to maintain a set speed and following distance. All three features will likely have different control modes, though AH and ESS should be transparent to the driver. ACC w/SG is different in that it will operate in one of two high-level modes that will be apparent to the driver, speed mode or distance mode, depending on the traffic conditions. While the feature must never exceed the speed set by the driver, when following a vehicle in traffic the logic will be slightly different than when traversing open road. This two-mode distinction is the reason for the *if-then* logic that is shown in Table 36, which describes the assumptions, conditions, and effects associated with ACC w/SG.

Table 36: ACC w/SG Conditions Table

		Design Assumption & Conditions Required	**Effect on the System**
Adaptive Cruise Control with Stop-Go	**Accelerate**	**ACC Enabled & Engaged:** Yes **Vehicle Held:** No (i.e. **Brakes:** Not Applied **& EPB:** No) **Driver Present:** Yes **Driver Gas Pedal:** No **Range:** D *If* **Target Locked:** Yes *Then* **Distance >= Threshold:** Yes **Speed <= Threshold:** Yes **Propulsion Power:** Available (i.e. **Engine:** On)	**Gas:** Applied by ACC **Speed:** Increased **Wheels Rotating:** Yes
	Brake	**ACC Enabled & Engaged:** Yes **Driver Pedal Input:** No **Gas:** Off **Driver Present:** Yes **Range:** D **Hydraulic Power:** Available (i.e. **Engine:** On | **Battery:** High)	**Brakes:** Applied by ACC **Speed:** Decreased

The Driver is the most complex controller in the vehicle system. His or her ability to incorporate many sources of feedback and intuition into decisions is unique and produces control strategies that are difficult to describe formally. Table 37 presents the Driver Conditions Table with a focus on when it is appropriate to issue complete four basic actions: leaving, shifting, accelerating, and braking.

Table 37: Driver Conditions Table

		Design Assumption & Conditions Required	**Effect on the System**
Driver	**Leave**	**Vehicle Held:** Yes (i.e. **Range:** Park **|** **EPB:** Yes) **Wheels Rotating:** No **Engine:** Off	**Driver Present:** No
	Shift	**Speed:** <TBD **New Range Available:** Yes	Transmission Range Change
	Gas	**Engine:** On **Range:** !=P **&** !=N **Vehicle Held:** No (i.e. **Brake:** Not Applied **& EPB:** Off)	**Wheels Rotating:** Yes
	Brake	**Hydraulic Power:** Available (i.e. **Engine:** On) **Brakes:** Not Applied	**Brakes:** Applied by Driver

5.3.2. Results

The conditions table above was searched for instances when the effects of each controller violate the required conditions and design assumptions of the others; the full list of potential conflicts is given in Appendix B. Each conflict is of the form:

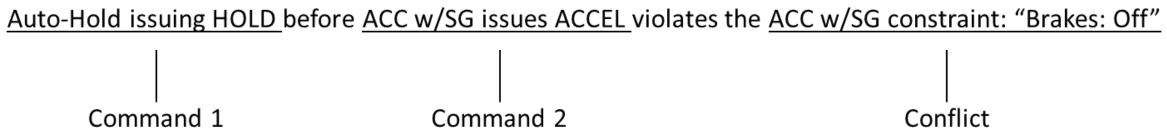

Auto-Hold issuing HOLD before ACC w/SG issues ACCEL violates the ACC w/SG constraint: "Brakes: Off"
Command 1
Command 2
Conflict

Figure 26: Example Conflict

The potential conflicts may be recorded in this form and reviewed by an engineer, or engineering team, which then determines an appropriate mitigation strategy. Three conflict scenarios are presented here to provide examples of results and to demonstrate how they may be used to inform the design process.

Some comments are made regarding the second cause of conflict, when one command triggers another in an uncoordinated manner; however, the focus of these examples is the violation of required conditions and design assumptions. The case study is being conducted at a high level of abstraction, such as the level of detail that may be available during concept development. Fully exploring the second type of conflict requires more information about the operation of the system than is available at the concept level, including implementation details such as timing requirements. Future work will further explore the other cause of potential conflicts, when the effect of one command triggers another such that they are issued in an uncoordinated fashion.

5.3.2.1. Conflict Scenario 1

The first example result is a conflict between Auto-Hold and Adaptive Cruise Control w/ Stop-Go. When both features are installed on a vehicle, they can conflict concerning operation of the EPB. The conflict was identified as:

> **Conflict 16:** AH APPLY EPB prior to ACC w/SG ACCELERATE violates the constraint *Vehicle Held: No*

When both AH and SG are installed and enabled in a vehicle, the following scenario is possible that would lead to a dysfunctional and potentially hazardous situation:

Scenario:

- Driver engages ACC w/SG to maintain a selected speed and safe following distance.
- Traffic slows to a stop; SG slows the vehicle and holds it at rest.
- Once at a stop, AH engages and captures the existing pressure in the brake lines.
- Some stimulus (see below) triggers AH to engage the EPB.

Outcome: ACC w/SG cannot ACCELEATE because the vehicle is held by the EPB…

There are several reasons that the AH feature may engage the EPB[13], one being the driver disabling the AH feature. The driver may do this if he thinks the AH system will interfere with SG, additional reasons that the driver would disable AH may be considered by completing STPA Step 2, specifically the process model of the Driver. Once the EPB is applied, the SG system is prevented from resuming vehicle motion when traffic allows because it does not have the authority to disengage the EPB.

At first glance, this conflict appears to lie outside the safety domain; however, it may have safety implications when the driver needs to quickly move the vehicle (i.e. it is stopped in an intersection) but is unable to do so because he or she must first recognize the EPB is applied, and then disengage and take control from SG.

This conflict may be resolved; three potential strategies are described below:

1. Require that ACC w/SG monitor the state of the EPB and allow it to disengage when appropriate. Allowing ACC w/SG to disengage the EPB resolves the conflict, but adds complexity

[13] There are other reasons that AH may issue APPLY EPB, including those in Section 0.

in that it must now be decided when it is appropriate for an automated system (ACC w/SG) to disengage the EPB.

2. Require that an issuing of the EPB turns other features ‘off’ and requires Driver intervention to disengage the EPB. This change is simpler than the first potential resolution and does not require giving automation the authority to disengage the EPB. However, not giving automated systems the ability to disengage the EPB means that each time it is issued, the Driver will be required to intervene. If the driver is required to intervene often, this strategy may reduce the transparency with which some features operate and thus reduce their value.
3. Require that AH does not engage when ACC w/SG is engaged as AH becomes redundant when ACC w/SG is holding the vehicle still. Note, this change does not prevent AH and ACC w/SG from using the same strategy and physical hardware to control the vehicle’s brakes. A design solution is that ACC w/SG effectively ‘implements’ AH when it brings the vehicle to a stop; however, this should be done within the ACC w/SG logic and the standalone AH system should remain disengaged.

As this conflict exists at a fairly high-level of abstraction and concerns the overlap of authority between Auto-Hold and Adaptive Cruise Control w/Stop-Go, the third potential resolution is recommended.

5.3.2.2. Conflict Scenario 2

A set of related conflicts exists between Auto-Hold and Engine Stop-Start, identified as:

Conflict 7: AH AP prior to ESS RESTART may violate the constraint *Battery Charge: High*

Conflict 19: ESS STOP prior to AH AP may violate *Hydraulic Power: Available*

Conflict 20: AH AP while ESS AUTO-STOPPED may violate *Auxiliary Power Needs: Low*

When both AH and SG are installed and enabled in a vehicle, the following scenario is possible that would lead to a dysfunctional and potentially hazardous situation:

Scenario:

- The vehicle battery has a low charge that is sufficient to restart the engine but not higher[14].
- The vehicle comes to a stop on an incline.
- AH engages first by capturing the current brake pressure to hold the vehicle.
- ESS then turns off the engine and consequently, idle torque disappears.
- Vehicle begins to roll because the captured brake pressure is no longer sufficient to hold the vehicle on the incline.

Outcome: AH will attempt to increase the braking pressure using the ADDITIONAL-PRESSURE command (per requirements shown in Figure 14) and ESS will attempt to RESTART the engine (per requirements shown in Figure 18); however, the low battery charge limits simultaneous power draw. As the battery voltage is already low, the drain from the RESTART effort may prevent the ABS pump from fully achieving the ADDITIONAL-PRESSURE command. If ADDITIONAL-PRESSURE is not successful, the vehicle will continue rolling until the engine starts and fully restores electrical power.

This raises both functionality and safety concerns. Functionally, the current logic for AH and ESS prevents them from both being used when the vehicle comes to rest on a positive grade. From a safety perspective, it is hazardous for the vehicle to be rolling when the driver does not have control of the brakes and the automated controller does not have sufficient power to control them either.

Requirements on both controller logic and feature components may be used to resolve this conflict:

[14] This may be the result of use-life, ambient temperature, or both.

1. The pump associated with Auto-Hold's ADDITIONAL-PRESSURE command should be able to operate at low voltage levels.
2. ESS should warn AH when it is about to RESTART (and drain the battery) so that AH may pre-increase the pressure accordingly.
3. The ESS logic associated with the battery voltage threshold should be high enough to guarantee simultaneous engine-restart and ABS pump operation.

Unlike the first example conflict scenario, this conflict between AH and ESS does not require that either be fundamentally changed or disabled. The conflict may be resolved by imposing additional requirements that were generated by considering the effect of each controller's actions on the other.

5.3.2.3. Conflict Scenario 3

The first two conflicts described were between pairs of controllers. This third conflict scenario is an example of conflict between three controllers: Auto-Hold, Engine Stop-Start, and the Driver. The following conflicts were detected using the new approach:

Conflict 21: AH RELEASE while AUTO-STOPPED violates the constraint *Engine: On*

Conflict 51: Driver SHIFT prior to ESS RESTART may violate *Range: !=P,R,N*

Again consider a vehicle that has both AH and ESS enabled:

Scenario:

- Vehicle comes to a stop, both the AH and ESS features engage successfully.
- Driver attempts to move the vehicle backward:
  - Driver shifts to Reverse
  - Driver applies the Accelerator Pedal

Outcome: The vehicle is effectively stuck, because:

- ESS is prevented from restarting the engine by FMVSS 102 [17].
- AH cannot RELEASE because there is insufficient wheel torque (Figure 15).

The stopping and starting of a vehicle engine is partially regulated by Federal Motor Vehicle Safety Standard (FMVSS) 102 [16]. The older version of this standard prohibited the engine starter from operating while the transmission shift lever is in either the forward or reverse drive position. This proves to be a barrier for new technologies, including hybrid-electric vehicles and idle-stop systems. In response to developments in those technologies, an updated version of FMVSS 102 prohibits the engine from automatically stopping in reverse gear but allows it to restart if automatically stopped while in a forward gear (i.e. followed the driver shifting to reverse) [17]. The engine may not automatically restart in reverse when the service brake is not applied and must restart automatically when it is applied. This means that in the scenario described above, the engine may not automatically restart until the driver applies the service brake: AH maintaining brake pressure is not sufficient.

With the engine off, the driver pressing the gas pedal does not produce any propulsion torque and AH may not issue the RELEASE command (see Figure 15). Thus, the vehicle is effectively stuck until the driver moves his foot to the service brake and the engine restarts. Once this happens, restored engine power will produce engine torque and if the driver presses the gas pedal, AH will issue RELEASE (see Figure 15). However, this sequence may take several seconds during which the vehicle is not fully controllable and the driver is required to assess the situation and determine the correct mitigation.

Resolving this conflict is not as straightforward as the first two examples as it involves several layers of control logic and regulation. Design engineers must prioritize the hazards associated with various solutions and choose one that is acceptable to all stakeholders.

5.4. Summary & Conclusions

This chapter has presented a new approach to analyzing multiple control systems that leverages the results of STPA Step 1 using the Thomas Method. The method takes the requirements generated by STPA Step 1 and considers them as a subset of the conditions required to safely and effectively issue a particular command. This subset is augmented by recording the assumptions embedded in the design and specification of each controller and its commands. Next, the effects of each command on the vehicle system are recorded in terms of PMV states. Once this information is recorded, the resulting conditions table is searched for conflicts between the effects of one command against the required conditions and assumptions of another. The conflicts identified are instances when one feature may inhibit the behavior of another feature, or prevent it from functioning completely. The conflicts should be reviewed by an engineer familiar with the system and used to further refine the designs of the features.

The new approach was executed over the course of several weeks, though it was being developed and refined during the application to the case study. This time may be reduced greatly in future applications. The time-intensive portions of the method are the development of the conditions table and the review of the output. The identification of conflicts may occur almost instantaneously if the conditions table is formalized and software executes the search. The time required to develop the conditions table largely depends on the complexity of the system to be analyzed in terms of scale and level of abstraction. The case study describes the features at a very high level of abstraction with less than 12 PMVs are associated with each control action. Also, time constraints for the control actions were not considered in the case study as they are considered implementation details to be determined further along in the design process when the dynamics of the systems are considered in greater depth. Review of the output, a list of conflicts that are potentially dysfunctional and/or hazardous, is the other time intensive portion of the method. The time required to review the output may be reduced by dividing it among members of an engineering team. Assuming that STPA Step 1 has been previously completed using the Thomas Method, it is estimated that the new approach may be completed by a small team of engineers over several days or a few weeks on a part time basis.

The output of this method provides engineers with information that can be used to identify scenarios in which the system design is flawed or inadequate as demonstrated by the three example scenarios. These three scenarios were perhaps not considered when the features were designed individually, though they may be encountered during normal operation of the vehicle. The conflicts that have been identified may be used to write requirements that will help ensure the feature systems work together in concert in all circumstances. Once engineers know which control action combinations lead to conflicts, they can evaluate the conflicts and decide which to mitigate and how.

By flagging the control action combinations that conflict, the new approach reduces the task of the engineers considerably from explicitly analyzing all combinations of control actions as would be required by the brute force method. Also, by identifying conflicts during the early stages of design, engineers may reduce the number of problems that arise during integration and testing of system prototypes. In fact, if the analysis is carried through STPA Step 2, the resulting scenarios may be used in system test to help define the test envelope and help ensure that the system is tested thoroughly when exhaustive testing is not possible.

As mentioned in the introduction of this chapter, an approach for analyzing multiple controllers was proposed and demonstrated by analyzing a complex spacecraft maneuver [16]. This new approach improves upon that proposed by Ishimatsu et. al. in several ways:

- **Greater guidance for evaluating the interaction of controllers and control actions.**
  The new approach provides additional guidance by using the PMVs as a basis for comparison of effects and conditions across controllers and control actions. These PMVs are defined when completing STPA Step 1 for each controller and its control actions and may continue to be used when analyzing the integrated system.
- **Considers individual control actions rather than the general output of controllers.**
  The older approach considers controllers as the base unit for comparison and classifies interactions based on their safety with respect to each issuing controller. The new method considers individual control actions as the base unit for analysis and is capable of identifying conflicts that arise within a single controller as well as those that arise between multiple controllers.

- **Improves scalability by reducing the growth rate of the analysis from exponential to linear.** The new method is a scalable means to analyze large numbers of controllers and control actions. The tables required by the old method cannot be visualized well when considering more than two controllers as each additional controller requires an additional dimension. The new approach reduces the growth rate so that the required input from the analyst grows linearly with the number of control actions to be considered.
- **Additional control actions may be appended, the analysis does not need to begin from scratch.** Another important distinction between the methods regards the evolution of the analysis as the system evolves. The new method simply requires that additional control actions are characterized in terms of the PMVs and appended to the conditions table rather than completing a new set of intersection tables.

The new approach demonstrated in this thesis extends STPA's ability to analyze the integration of multiple control systems for safety and functionality. Although the case study was successful, there may be opportunity for further development of the new approach. This is partially the subject of the next and final section, Chapter 6.

Page intentionally left blank.

6. Conclusion

6.1. Summary of Work

This thesis has demonstrated how STPA may be used to analyze multiple control systems that are to be integrated together. STPA was selected from amongst several common analysis techniques because of its foundation in systems theory and ability to analyze the system as a whole rather than simply in segments. An automotive example was selected because it is a high-profile industry that is experiencing a rapid growth in introduction of feature systems. A case study including three automotive features systems was created and used as an example throughout the thesis.

Chapter 3 presented the foundation for the STPA analysis as well as the initial results of STPA Step 1. The foundation for the analysis included defining the three feature systems using functional control structures as well as the accidents to be avoided in their design and operation. It was noted that a common control structure exists for the three features and a template control structure was presented that may be used in future analyses. STPA Step 1 was conducted using traditional methods to identify ways that the embedded controllers in each feature system may issue commands that are unsafe and/or dysfunctional. A small portion of STPA Step 2 was presented for each feature to demonstrate how causal factors contributing to inappropriate control may be identified.

Chapter 4 took the results from Chapter 3 and demonstrated how STPA may be iterated with increased formalism using the Thomas Method. The feature systems of the case study were further developed and described in greater detail, specifically by defining Process Model Variables that specify the state-space for each controller. Executable requirements were generated that govern the behavior of each feature and bound it to what is both safe and functional. The results of the analysis presented in Chapters 3 and 4 is recorded in greater detail in Appendix A using Intent Specifications. At the end of Chapter 4, it was concluded that additional guidance is needed to analyze the controllers in an integrated way and identify conflicts and other dysfunctional interactions that may occur.

Chapter 5 presents a new approach for analyzing the integration of control systems within the STPA framework. The new approach provides additional guidance and improves scalability compared to previously proposed methods for analyzing multiple controllers within STPA. The initial results of the new approach are promising as it successfully leveraged the results of STPA Step 1 (from Chapters 3 and 4) to identify several dozen ways in which the three feature systems may interfere with each other at a high level. It is concluded that this method should be further refined and adopted when appropriate during the execution of STPA.

6.2. Contributions

The work presented in this thesis has made several contributions to both academic researchers and industry collaborators. The following are the primary contributions resulting from this work:

- **Successful implementation and refinement of a new extension to STPA.**
  This new approach may be used in the future when STPA is being applied to systems with multiple controllers. The approach may be used as early as concept development and iterated in greater details as the design process moves forward to detailed design work. As it leverages the results of STPA executed with Thomas Method, the new approach does not require significantly more analysis time or effort.
- **Case study results that may be used by engineers in academia and industry.**
  The case study in this thesis is fictional, but realistic enough that the results may be used by those in industry designing similar systems. The end of Chapter 4 includes some comments on what the results of this work reveal about advanced features in automobiles. The results of Chapter 5 (further listed in Appendix B) are a demonstration of how, even at the concept level, automotive subsystems must be designed with the larger vehicle system in mind to avoid conflicts upon integration. As the conversation concerning advanced features in automobiles continues, it is the author's hope that decision makers will root their decisions in systems thinking analysis techniques such as STPA.
- **Addition to the example base for new users of STPA.**
  It is the hope of the author that this thesis provides a good example of STPA and the new method for future users including students and professionals. The case study is not so technically detailed that it is unapproachable by those with no automotive background and yet the results are meaningful and informative. For users in the automotive industry, a high level set of accidents and hazards has been presented that may be applied to any automotive system, potentially to be refined into sub-accidents and sub-hazards. Additionally, the control structure template provides a starting point for future analyses of embedded control systems.

6.3. Future Work

The new approach demonstrated in this thesis may be improved further; the author suggests the following avenues for future work:

- **Application to additional systems and in different domains.**
  Additional case studies, both in the automotive industry and elsewhere, will help further develop and refine the method. Ideally, future case studies will be conducted on real systems and the results validated accordingly.
- **Further explore conflicts that occur when one command inadvertently triggers another.**
  The new approach proposed in Chapter 5 can be used to identify two types of potential conflicts: 1) when the issuing of one command prohibits the issuing of a second command 2) when the issuing of one command inadvertently prompts the issuing of a second. The examples presented in Chapter 5 focus on the first type of potential conflict, as it is easier to identify at the concept level when implementation details are yet to come. Future work should further explore the second type of conflict, particularly the application of the new approach during later stages of the design process when more detail regarding operation, and thus lower level requirements, are available.
- **Explore connections and extensions to STPA Step 2 causal factor analysis.**
  Ishimatsu et. al. proposed in [16] an expanded guide for identifying causal factors in a multiple controller system. This vein of work should be continued in the spirit of the approach presented here, seeking to standardize the analysis across controllers such that results from one may inform the analysis of another.
- **Formalize the method and contribute to STPA tool development.**
  Several researchers, at MIT and elsewhere, are currently developing tools to automate parts of STPA. Upon further verification and refinement, the new approach should be integrated into these software packages to aid analysts in executing STPA.

Page intentionally left blank.

Bibliography

[1] MIT PSAS, "An STPA Primer," 2013.

[2] R. Charette, "This Car Runs on Code," IEEE Spectrum, 1 February 2009. [Online]. Available: http://spectrum.ieee.org/transportation/systems/this-car-runs-on-code. [Accessed 29 April 2014].

[3] Science Friday, *Interview with Dr. Ingolf Krueger,* NPR, 2010.

[4] Q. D. V. E. Hommes, "Applying System Theoretical Hazard Analysis Metod to Complex Automotive Cyber Physical System," in *ASME IDETC/CIE 2012*, Chicago, IL, 2012.

[5] Transportation Research Board, "The Safety Promise and Challenge of Automotive Electronics," National Research Council, 2012.

[6] N. Leveson, Engineering a Safer World, Cambridge: MIT Press, 2012.

[7] J. Yoshida, "Toyota Case: Inside Camry's Electronic Control Module," EE Times, 30 October 2013. [Online]. Available: http://www.eetimes.com/document.asp?doc_id=1319952. [Accessed 5 May 2014].

[8] J. Vincoli, Basic Guide to System Safety, Second Edition, Hoboken, New Jersey: John Wiley & Sons, Inc., 2006.

[9] M. Rausand and A. Hoyland, System Reliability Theory Models, Statistical Methods, and Applications Second Edition, Hoboken, New Jersey: John Wiley & Sons, Inc., 2004.

[10] N. Leveson, Safeware System Safety and Computers, Addison-Wesley, 1995.

[11] N. Leveson, "A New Accident Model for Engineering Safer Systems," *Safety Science,* vol. 42, no. 4, pp. 237-270, 2004.

[12] J. Thomas, *Extending and Automating a Systems-Theoretic Hazard Analysis for Requirements Generation and Analysis,* Cambridge, MA, 2013.

[13] N. Leveson, "Intent Specifications: An Approach to Building Human-Centered Specifications," *IEEE Transactions on Software Engineering,* vol. 26, pp. 15-35, 2000.

[14] National Highway Traffic Safety Administration (NHTSA), FMVSS 102.

[15] J. Thomas and M. S. Placke, "Analyzing Feature Interactions in Automobiles," in *STAMP Workshop 2014*, Cambride, MA, 2014.

[16] T. Ishimatsu, N. Leveson, J. Thomas, C. Fleming, M. Katahira, Y. Miyamoto, R. Ujiie, H. Nakao and N. Hoshino, "Hazard Analysis of Complex Spacecraft using Systems-Theoretic Process Analysis," *Journal of Spacecraft and Rockets,* vol. 51, no. 2, pp. 509-522, 2014.

[17] "Federal Motor Vehicle Safety Standards; Transmission Shift Position Sequence, Starter Interlock, and Transmission Braking Effect," *49 CFR Part 571,* 2005.

Appendix A: Intent Spec Results

This appendix records the results of the analysis conducted in Chapters 3 and 4 using Intent Specifications as developed by Leveson [13]. The results are recorded through the completion of STPA Step 1 using the Thomas Method for the three automotive features: Auto-Hold, Engine Stop-Start, and Adaptive Cruise Control w/Stop-Go. The Intent Specifications begin with high level statements about the systems' purpose and goals which are then decomposed to the level of black-box models in the form of SpecTRM-RL executable requirements. Where possible, hyperlinks are included to reference between specification levels and provide a means of tracing the reasoning of the analysis from one level to those above and below.

The details in this analysis are fictional but considered to be realistic. The results are not mean to be used for design or safety evaluation but to demonstrate the application of STPA and the new analysis approach to the integration of embedded control systems.

Level 1: System Purpose

System Goals

G1: The vehicle controls should prevent rollback when vehicle is stopped. [→FE1; ↓AH-G1]

G2: The vehicle controls should reduce idle time when the vehicle is stopped. [→FE2; ↓SS-G1]

G3: The vehicle controls should provide functionality to automatically maintain set speed and/or distance from the vehicle in front. [→FE3; ↓SG-G1, SG-G2]

Accident Definition

A1: Two or more vehicles collide [→H1; →H3]

A2: Vehicles collide with non-fixed obstacle[15] [→H2; →H3]

A3: Vehicle crashes into terrain[16] [→H2; →H3]

A4: Vehicle occupants injured without vehicle collision [→H3; →H4]

Preliminary Hazard Analysis

H1: Vehicle does not maintain safe distance from nearby vehicles [←A1; ↓AH-C1, AH-C2, SG-C1]

H2: Vehicle does not maintain safe distance from terrain & obstacles [←A2, A3; ↓AH-C1, AH-C2, SG-C1]

H3: Vehicle enters uncontrollable/unrecoverable state [←A1, A2, A3, A4; →DC3, DC4 ↓AH-C2, SS-C1, SS-C2, SG-C1, SG-C2, SG-C3]

H4: Vehicle occupants exposed to harmful effects and/or health hazards [←A4; ↓AH-C1]

System Design Constraints

DC1: The new features described in G1-G3 must be implemented with minimal new hardware into the vehicle architecture [↓AH-C3]

DC2: The new features must utilize existing vehicle infrastructure to the extent possible, such as existing power and communication networks in the vehicle architecture []

DC3: The driver must be able to enable or disable the new features to directly control the vehicle [←H3; ↓AH-C2.1, SS-C3, SG-C4]

DC6: The new features should respond consistently to driver input. [←H3]

[15] 'Other obstacle' includes pedestrians, bikers, animals, etc.
[16] 'Terrain' includes fixed, permanent objects such as guard rails, trees, bridges, signage, pavement, etc.

Assumptions

AS1: The vehicle has an automatic transmission.

AS2: The vehicle has an Electronic Park Brake (EBP).

AS3: The vehicle has a standard hydraulic brake system.

AS4: The vehicle has a standard, internal combustion propulsion system.

Level 2: System Design

Auto-Hold Goals:

AH-G1: Auto-Hold should hold the vehicle when at a standstill. [↑FE1, G1]

AH-G1.1: Auto-Hold is meant to be used while the vehicle is in a forward transmission range. []

AH-G1.2: Auto-Hold is intended for holding the vehicle for short durations during routine traffic stops. [↑DC5]

Auto-Hold Constraints:

AH-C1: Auto-Hold must not take control of the brakes when they are not already applied by another controller [↑H1, H2, H4]

Intent: *Doing so may reduce the ability of the driver to control the vehicle using either the brakes (unintended additive effect) or the propulsion system (unintended conflict).*

AH-C2: Once engaged, AH is responsible for maintaining the vehicle at a standstill until it is properly disengaged. [↑H1, H2, H3]

AH-C2.1: AH may only disengage when the vehicle is being held by another system or propulsion torque is sufficient to propel the vehicle. [↑DC3]

AH-C2.2: AH may not allow the wheels to roll while it is engaged. []

Intent: *If AH releases without disengaging then the vehicle may roll and hit an obstacle or another vehicle, but retain control of the brakes not allowing other systems to use them. The consequence of AH releasing without handing off holding responsibilities or sufficient propulsion torque is the same, it may impact another object.*

AH-C3: Auto-Hold should use the brake-valve hardware associated with the ABS system and should not introduce new hardware. [↑DC1]

AH-C4: Auto-Hold cannot influence the brakes when it is not in HOLD-MODE. [↑DC3]

Auto-Hold Modes:

AH-M1: Auto-Hold shall initialize to OFF-MODE (NOT-ENABLED) upon vehicle start. []

AH-M2: Auto-Hold will enter STANDBY-MODE (ENABLED) upon driver input. []

AH-M3: Auto-Hold will enter HOLD-MODE (ENGAGED) when automatically engaged. []

Engine Stop-Start Goals:

ESS-G1: Engine Stop-Start should shutoff then engine when the vehicle is at a standstill and restart before motion resumes to reduce fuel consumption. [↑FE2, G2]

ESS-G1.1: Engine Stop-Start is intended to shutoff and restart the engine for short durations at routine traffic stops. [↑DC5]

Engine Stop-Start Constraints:

ESS-C1: ESS cannot shutoff the engine when the vehicle is in motion [↑H3]

ESS-C1.1: ESS can only shutoff the engine when another system (may be the driver) is actively holding the vehicle.

ESS-C1.2: ESS should not shutoff when power is needed for steering and braking. [↑DC4]

Intent: *Shutting off the engine while the vehicle is in motion takes away power steering and brakes that are necessary to control the vehicle's motion. Also, shutting off the engine while the vehicle is in motion does not meet the functional goal.*

ESS-C2: ESS may only keep the engine off when the vehicle power needs are sufficiently low.

ESS-C2.1: ESS may not turn off the engine when a restart is not possible and should restart the engine before the battery is depleted such that this is the case. [↑H3]

ESS-C2.2: ESS should keep the engine on if non-propulsion/braking power needs are above that which the battery can provide and should restart if their power draw rises above the battery's capabilities.

Intent: *Shutting off the engine when there is not enough battery power to restart it renders the functionality useless and may leave the vehicle stranded. Not returning engine power when a non-critical system needs power (e.g. air conditioning) will frustrate users.*

ESS-C3: ESS cannot issue commands when it is not enabled. [↑DC3]

ACC w/Stop-Go Goals:

SG-G1: Stop-Go should maintain a steady, forward vehicle motion when engaged and it is safe to do so.

SG-G1.1: Stop-Go should maintain a driver-set speed when engaged in a forward transmission range. [↑FE3, G3]

SG-G1.2: Stop-Go should maintain a driver-set time/distance gap when engaged in a forward transmission range. [↑FE3, G4]

SG-G1.3: Stop-Go should follow a target vehicle in stop-and-go traffic.

ACC w/Stop-Go Constraints:

SG-C1: Stop-Go may not initially ENGAGE from a standstill. [↑H1, H2, H3]

Intent: *Automatic taking off from a standstill is not in the common driver mental model and thus the system should make sure that the driver is attentive and ready to steer the vehicle or cancel the function. The system may not initially engage from a standstill because the set (max) speed has not been set – only the distance.*

SG-C2: Stop-Go must yield to driver override. [↑H3]

SG-C3: Stop-Go may not violate driver-defined parameters. [↑H3]

SG-C4: Stop-Go may not issue commands when it is not engaged [↑ DC3].

SG-C4.1: Stop-Go should relinquish control of propulsion and braking if disengaged by the driver.

ACC w/SG Modes:

SG-M1: Auto-Hold shall initialize to OFF-MODE (NOT-ENABLED) upon vehicle start. []

SG-M2: Auto-Hold will enter STANDBY-MODE (ENABLED) upon driver input. []

SG-M3: Auto-Hold will enter CRUISE-MODE (ENGAGED) when … []

Level 3: Context Tables / Rationale / Black-box Models

Auto-Hold – HOLD

Context ID	AH Enabled	Hold-Mode	Driver Present	PRNDL	Gas Pedal	Brake Pedal	Wheels Rotating	Providing Causes Hazard	Not Providing Causes Hazard	Providing Required for Function	Not Providing Require for Function
AH-H-1	Yes	Yes	Yes	P	Not Pressed	*	No	x			
AH-H-2	Yes	Yes	Yes	!=P	Not Pressed	*	No				
AH-H-3	Yes	No	Yes	P	Not Pressed	*	No	x			
AH-H-4	Yes	No	Yes	N	Not Pressed	*	No	x			x
AH-H-5	Yes	No	Yes	R	Not Pressed	*	No	x			x
AH-H-6	Yes	No	Yes	D,L	Not Pressed	Not Pressed	No	x			
AH-H-7	Yes	No	Yes	D,L	Not Pressed	Pressed	No		x	x	
AH-H-8	*	*	*	*	*	*	Yes	x			
AH-H-9	*	*	*	*	Pressed	*	*	x			
AH-H-10	*	*	No	*	*	*	*	x			
AH-H-11	No	*	*	*	*	*	*	x			

AH-H-1: This context should only be achieved if the driver shifts to Park while AH is in HOLD-MODE. RELEASE should be issued [→AH-R-1] to avoid confusion regarding what is holding the vehicle [↑DC3]. The concern is that just prior to shifting out of Park, there may be three things holding the vehicle: Park gear, Driver's foot on service brake, and AH. When the driver is shifting he should have a clear understanding of what he controls to avoid mode confusion and associated unsafe actions [↑H3].

AH-H-2: This is not hazardous because it assumed that issuing HOLD has not affect when already in HOLD-MODE.

AH-H-3: HOLD should not be issued while in Park [↑DC3, H3; ←AH-H-1].

AH-H-4: HOLD should not be issued while in Neutral because it is not part of the functional intent, and there is not a safe way to exit HOLD-MODE while in Neutral without invoking another vehicle function [↑DC4; →AH-R-2]. Also, issuing HOLD when the Driver is not applying the brake pedal violate the condition that AH must not be issued when the brakes are not already applied [↑AH-C1]. Additionally, the intent of AH is to hold the vehicle in Drive and Low [↑AH-G1.1], not in Neutral.

AH-H-5: HOLD should not be issued while in Reverse because it is not part of the functional intent [↑AH-G1.1]. Additionally, issuing HOLD in Reverse may increase the mental workload of the Driver during a time when he is already tasked with a new steering orientation and focuses on the environment all-around the vehicle rather than in front [↑DC6]

AH-H-6: HOLD should not be issued when the brakes are not already applied [↑AH-C1].

AH-H-7: HOLD should be issued here to fulfill its functional intent [↑AH-G1].

AH-H-8: HOLD should not be issued when the vehicle's wheels are rotating, doing so may lead to a rear-end collision or abrupt deceleration [↑H3].

AH-H-9: HOLD should not be issued when the accelerator is pressed as this will create conflict in the vehicle's dynamics and may make the vehicle difficult to control [↑H3]

AH-H-10: HOLD should not be issued when the driver is not present because the vehicle mode will change without the driver's knowledge. This may lead to mode confusion and unexpected vehicle behavior [↑H3, DC6]

AH-H-11: HOLD may not be issued if AH is not enabled to avoid taking away control of the brakes from the driver when it is unexpected (mode confusion) [↑H1, H3, AH-C4].

H: HOLD Command		S-F
AH Enabled =	Yes	T
	No	
Hold-Mode =	Yes	
	No	T
Driver Present =	Yes	T
	No	
PRNDL =	P	
	R	
	N	
	D | L	T
Gas Pedal =	Pressed	
	Not Pressed	T
Brake Pedal =	Pressed	T
	Not Pressed	
Wheels Rotating =	Yes	
	No	T

Auto-Hold – ADDITIONAL-PRESSURE

Context ID	Hold-Mode	Gas Pedal	Wheels Rotating	Providing Causes Hazard	Not Providing Causes Hazard	Providing Required for Function	Not Providing Required for Function
AH-AP-1	Yes	Not Pressed	Yes		x	x	
AH-AP-2	Yes	Not Pressed	No	x			
AH-AP-3	Yes	Pressed	Yes		x		
AH-AP-4	Yes	Pressed	No	x			
AH-AP-5	No	*	*	x			

AH-AP-1: If the system is in HOLD-MODE and the wheels begin to rotate, ADDITIONAL-PRESSURE should be issued to keep the vehicle at a standstill [↑AH-C2.2].

AH-AP-2: If the system is in HOLD-MODE but the wheels are not rotating, ADDITIONAL-PRESSUE is not necessary and issuing it may cause excesses wear and tear in the braking system from increased pressure [↑DC7].

AH-AP-3: Should issue ADDITIONAL PRESSURE using the reasoning from [←AH-AP-1]. It is counter-intuitive to fight the driver; however, when in HOLD-MODE the driver doesn't have control of the brakes and they may not know it. To allow them to continue accelerating the vehicle without control of the brakes is hazardous [↑H3, DC3].

AH-AP-4: The wheels are not rotating, so issuing ADDITIONAL-PRESSURE may accelerate wear and tear in the brake system [↑DC7].

AH-AP-5: AH may not influence the brake system when it is not in HOLD-MODE and does not have the responsibility to do so [↑AH-C4]. Issuing additional pressure may interfere with other braking systems (service brake, ABS).

AH-AP-6: ADDITIONAL-PRESSURE may not be issued if AH is not enabled to avoid influencing the brakes from the driver when it is unexpected (mode confusion) [↑H1, H3, AH-C4].

AH: AP Command			S-F	S
	Hold-Mode =	Yes	T	T
		No		
	Gas Pedal =	Pressed		T
		Not Pressed	T	
	Wheels Rotating =	Yes	T	T
		No		

Auto-Hold – RELEASE

Context ID	AH Enabled	Hold-Mode	Driver Present	PRNDL	Gas Pedal	Propulsion Torque Sufficient	Brake Pedal	EPB On	Providing Causes Hazard	Not Providing Causes Hazard	Providing Required for Function	Not Providing Required for Function
AH-R-1	Yes	Yes	Yes	P	*	*	*	No		x		
AH-R-2	Yes	Yes	Yes	N	*	*	Not Pressed	No	x			
AH-R-3	Yes	Yes	Yes	N	*	*	Pressed	No	x			
AH-R-4	Yes	Yes	Yes	R,D,L	Pressed	Yes	*	No		x		x
AH-R-5	Yes	Yes	Yes	R,D,L	Pressed	No	*	No	x			
AH-R-6	Yes	Yes	Yes	R,D,L	Not Pressed	*	*	No	x			
AH-R-7	Yes	Yes	No	*	*	*	*	No	x			
AH-R-8	No	Yes	*	*	*	*	*	No	x			
AH-R-9	*	No	*	*	*	*	*	No				
AH-R-10	*	*	*	*	*	*	*	Yes				

AH-R-1: Not providing RELEASE may induce an incomplete or incorrect mental model in the driver regarding control of the brakes [↑AH-C2.1].

Reasoning: *AH should RELEASE so that the driver is able to take control of the brakes by pressing the pedal. He must apply the pedal to get out of Park (interlock requirement) and should not have to question whether his foot on the brake is actually control the brake system. Additionally, if the Park Gear and AH are working together to hold the vehicle, moving from Park may increase the load on AH past what ADDITIONAL-PRESSURE can respond to (e.g. if significant mass is added to the vehicle such as a trailer).The worst-case consequence of providing is that the vehicle may roll some due to gear lash in the transmission – though this is a known and accepted risk (it is currently experienced by most vehicles).*

AH-R-2: Cannot provide RELEASE because the vehicle will be free to roll which violates the constraint that HOLD-MODE can only be excited when another system will actively control the vehicle's motion [↑AH-C2.1].

Reasoning: *Providing RELEASE in Neutral w/o the service brake engaged will leave the vehicle free to roll. Not providing RELEASE may confuse a driver who does not fully understand AH; however, the vehicle will be 'stuck' in a safe state (at rest) and this is deemed the less hazardous scenario. It is recognized that this reasoning combined with [→AH-R-3] does not provide a means for exiting AH in Neutral, which the driver may want to do to roll such as if towing. This is addressed by allowing RELEASE once the EPB is engaged [→AH-R-10].*

AH-R-3: AH should not provide RELEASE when the vehicle is in Neutral with the brakes on because is not aligned with the function of the brake pedal and may confuse the driver's mental model [↑DC6.1]

Reasoning: *Not providing RELEASE will at work confuse the driver will leaving the vehicle 'stuck' in a safe state (at rest). Providing RELEASE means that the driver will be pressing the brake pedal to disengage the AH system. This would be the only situation in which this is true and it is counter-intuitive to press the brake pedal to release AH's control of the brakes. Further, if the driver does not know that he has been given control of th brakes, then he may shift to another gear expecting AH to cancel the creep torque when he release his foot. It is recognized that this reasoning combined with [←AH-R-2] does not provide a means for exiting AH in Neutral, which the driver may want to do to roll such as if towing. This is addressed by allowing RELEASE once the EPB is engaged [→AH-R-10].*

AH-R-4: AH should provide RELEASE when the commanded acceleration provides sufficient propulsion torque [↑AH-C2.1].

AH-R-5: AH should not provide RELEASE until the commanded acceleration provides sufficient propulsion torque [↑AH-C2.1]. Providing RELEASE prior will allow the vehicle to roll.

AH-R-6: AH should not provide RELEASE if acceleration has not been commanded and there is not another system holding the vehicle [↑AH-C2.1].

AH-R-7: AH should not provide RELEASE when the driver exits the vehicle as this will allow the vehicle to roll with no control over velocity or steering [↑AH-C2.1]. Once the EPB is deployed per [→AH-EPB-3], the condition [→AH-R-10] will apply and RELEASE may be issued satisfying [↑DC4].

AH-R-8: Can only enter this state if the driver disables AH while in HOLD-MODE, to satisfy [↑AH-C2.1] AH must remain in HOLD MODE until a satisfactory RELEASE condition is satisfied.

Reasoning: *It is possible to expand this state and account for every possible combination of gear, brake pedal and forward propulsion; however, this would mean the functionality of the AH button would change depending on the state of the inputs, violating [↑DC6.1]. The EPB should be issued per [→AH-EPB-4] which will allow RELEASE to be issued per [→AH-R-10].*

AH-R-9: It is assumed that issuing RELEASE when not in HOLD-MODE will have no effect.

AH-R-10: When the vehicle is in HOLD-MODE and the EPB is engaged, it is safe (but not required) to issue RELEASE [↑AH-C2.1].

AH-R-11: When the vehicle is in HOLD-MODE and turned off, it is necessary to issue RELEASE so that AH is not indefinitely holding the vehicle and violate [↑AH-G1.2]. Doing so w/o the EPB engaged will allow the vehicle to roll and violate [↑AH-C2.1].

AH: RELEASE Command		S	S-F	S-F
AH Enabled =	Yes	T	T	T
	No			
Hold-Mode =	Yes	T	T	T
	No			
Driver Present =	Yes	T	T	T
	No			
PRNDL =	P	T		
	R		T	
	N			
	D | L			T
Gas Pedal =	Pressed		T	T
	Not Pressed			
Brake Pedal =	Pressed			
	Not Pressed			
Propulsion Torque Sufficient =	Yes		T	T
	No			
EPB On =	Yes			
	No	T	T	T

Auto-Hold – APPLY EPB

Context ID	AH Enabled	Hold-Mode	Driver Present	PRNDL	Vehicle On	Providing Causes Hazard	Not Providing Causes Hazard	Providing Required for Function	Not Providing Require for Function
AH-EPB-1	Yes	Yes	Yes	!=N	Yes				x
AH-EPB-2	Yes	Yes	Yes	N	Yes			x	
AH-EPB-3	Yes	Yes	No	*	Yes		x		
AH-EPB-4	No	Yes	*	*	Yes		x		
AH-EPB-5	*	Yes	*	*	No		x		
AH-EPB-6	*	No	*	*	*	x			

AH-EPB-1: It is safe but not necessary to issue the EPB when in HOLD-MODE but not in NEUTRAL. Actually, there may be instances when this annoys the driver as the EPB will hold the vehicle even when the driver attempts to accelerate requiring him to disengage the EPB before resuming motion.

AH-EPB-2: When the vehicle is shifted to Neutral while in HOLD-MODE, the EPB should be issued to allow a means for exiting the mode [↑DC4]. This is the resolution outcome of the reasoning associated with the requirements [←AH-R-2, AH-R-3]

AH-EPB-3: If the driver leaves the vehicle while in HOLD-MODE, the EPB should be issued (followed by RELEAESE) to secure the vehicle while meeting its functional goal that it's only meant for short durations [↑AH-C2, AH-G1.2]. Also, having the vehicle held by the EPB rather than AH upon the driver's return is closer to his mental model for a standard startup and reduces the thought necessary to take control of the vehicle [↑DC6].

AH-EPB-4: If disabled while in HOLD-MODE, AH should honor the request to disengage but should secure the vehicle first with the EPB [↑AH-C2.1].

AH-EPB-5: If the vehicle is turned off while in HOLD-MODE, AH should engage the EPB to secure the vehicle to satisfy [↑AH-C2.1]. Remaining in HOLD-MODE without power for issuing ADDITIONAL PRESSURE may lead to a roll away.

AH-EPB-6: AH may not issue EPB if it is not enabled to avoid severely impacting the dynamics of the vehicle [↑H1, H3, AH-C4].

AH: APPLY EPB Command

		F	S	S	S
AH Enabled =	Yes	T	T		
	No			T	
Hold-Mode =	Yes	T	T	T	T
	No				
Driver Present =	Yes	T			
	No		T		
PRNDL =	P				
	R				
	N	T			
	D				
	L				
Vehicle On =	Yes	T	T	T	
	No				T

ESS – STOP

Context ID	ESS Enabled	Auto-Stopped	Driver Present	PRNDL	Gas Pedal	Vehicle Held	Wheels Rotating	Restart Possible	Auxiliary Power Needs	Providing Causes Hazard	Not Providing Causes Hazard	Providing Required for Function	Not Providing Required for Function	FMVSS 102
ESS-STOP-1	No	*	*	*	*	*	*	*	*	x				
ESS-STOP-2	Yes	Yes	*	*	*	*	*	*	*					
ESS-STOP-3	Yes	No	No	*	*	*	*	*	*	-	-	-	-	-
ESS-STOP-4	Yes	No	Yes	P, R, N	*	*	*	*	*	x				prohibited
ESS-STOP-5	Yes	No	Yes	D, L	Yes	*	*	*	*	x			x	
ESS-STOP-6	Yes	No	Yes	D, L	No	No	*	*	*	x				
ESS-STOP-7	Yes	No	Yes	D, L	No	Yes	Yes	*	*	x				
ESS-STOP-8	Yes	No	Yes	D, L	No	Yes	No	No	*	x			x	
ESS-STOP-9	Yes	No	Yes	D, L	No	Yes	No	Yes	Above				x	
ESS-STOP-10	Yes	No	Yes	D, L	No	Yes	No	Yes	Below			x		

ESS-STOP-1: Stop-Start should not issue STOP when it is not enabled because it has not been given command authority over the engine. Shutting off the engine may confuse the driver and leave them without the power systems (brakes and steering) necessary to control the vehicle. [↑SS-C3]

ESS-STOP-2: It is assumed that issuing the STOP command when already AUTO-STOPPED will have no effect.

ESS-STOP-3: In this scenario the vehicle's engine is running and the driver leaves. Required action TBD.

Rationale: *The appropriate resolution is dependent on what other systems are present. If SS is the lone active system, then the driver chose to leave the vehicle with the engine running and not held (assuming in gear other than Park – which is fine). It is not possible to mitigate every driver action, some responsibility lies with the driver to operate the vehicle correctly. If another system such as AH is present, then it may be holding the vehicle and the Driver may deem it safe to leave temporarily. In this scenario, a coordinated response is necessary so that the vehicle is held secure. Distributed and/or competing reactions may push the potentially hazardous situation to an accident.*

ESS-STOP-4: SS should not turn off the engine in transmission ranges other than forward (Drive and Low). Doing so may lead to driver confusion and is prohibited by FMVSS 102 S3.1.3. [↑ DC6]

ESS-STOP-5: SS should not turn off the engine when the driver is propelling the vehicle in a forward transmission range, doing so may take away power braking and steering while the vehicle is still moving [↑ SS-C1].

ESS-STOP-6: SS should not turn off the engine when the vehicle is not held as reducing idle torque may allow the vehicle to roll (w/o power steering and brakes) leading to a rear-end collision [↑ SS-C1.1].

ESS-STOP-7: SS should not turn off the engine when the vehicle is in motion, doing so reduces controllability by taking away power steering and brakes [↑ SS-C1.1].

ESS-STOP-8: SS should not turn off the engine when a restart is not possible, doing so may leave the vehicle stranded. In most cases this will be a functional violation of [↑ SSG1] but at an intersection it is a safety hazard because it may prevent the driver from moving out of the way of oncoming traffic [↑ DC4].

ESS-STOP-9: If the Non-Propulsion/Braking power needs are too high for the battery, turning off the engine will violate functional goal [↑ SS-C2.2], but will not be a safety hazard.

ESS-STOP-10: This is the intended operation of SS that meets its functional goals [↑ SSG1] without violating any constraints.

ESS: STOP Command

		F
ESS Enabled =	Yes	T
	No	
Auto-Stopped =	Yes	
	No	T
Driver Present =	Yes	T
	No	
PRNDL =	P	
	R	
	N	
	D | L	T
Gas Pedal =	Pressed	
	Not Pressed	T
Vehicle Held =	Yes	T
	No	
Wheels Rotating =	Yes	
	No	T
Restart Possible =	Yes	T
	No	
Auxiliary Power Needs =	Above	
	Below	T

ESS – RESTART

Context ID	ESS Enabled	Auto-Stopped	Driver Present	PRNDL	Gas Pedal	Vehicle Held	Wheels Rotating	Restart Possible	Auxiliary Power Needs	Providing Causes Hazard	Not Providing Causes Hazard	Providing Required for Function	Not Providing Required for Function	FMVSS 102
ESS-RESTART-1	No	*	*	*	*	*	*	*	*	x				
ESS-RESTART-2	Yes	No	*	*	*	*	*	*	*	x				
ESS-RESTART-3	Yes	Yes	No	*	*	*	*	*	*	x			x	
ESS-RESTART-4	Yes	Yes	Yes	*	*	*	*	No	*	-	-	-	-	-
ESS-RESTART-5	Yes	Yes	Yes	P, N	*	*	*	Yes	Above			x		
ESS-RESTART-6	Yes	Yes	Yes	P, N	*	*	*	Yes	Below				x	
ESS-RESTART-7	Yes	Yes	Yes	R	*	Yes	*	Yes	*					required
ESS-RESTART-8	Yes	Yes	Yes	R	*	No	*	Yes	*		x			prohibited
ESS-RESTART-9	Yes	Yes	Yes	D, L	Yes	Yes	Yes	Yes	*		x			
ESS-RESTART-10	Yes	Yes	Yes	D, L	Yes	Yes	No	Yes	*	x		x		
ESS-RESTART-11	Yes	Yes	Yes	D, L	Yes	No	Yes	Yes	*		x	x		
ESS-RESTART-12	Yes	Yes	Yes	D, L	Yes	No	No	Yes	*	x		x		
ESS-RESTART-13	Yes	Yes	Yes	D, L	No	Yes	Yes	Yes	*		x			
ESS-RESTART-14	Yes	Yes	Yes	D, L	No	Yes	No	Yes	Below				x	
ESS-RESTART-15	Yes	Yes	Yes	D, L	No	Yes	No	Yes	Above			x		
ESS-RESTART-16	Yes	Yes	Yes	D, L	No	No	Yes	Yes	*		x			
ESS-RESTART-17	Yes	Yes	Yes	D, L	No	No	No	Yes	Below	x				
ESS-RESTART-18	Yes	Yes	Yes	D, L	No	No	No	Yes	Above	x		x		

ESS-RESTART-1: Stop-Start should not issue START when it is not enabled because it has not been given command authority over the engine. Starting the engine may startle the driver (if present) and begin vehicle motion when he is not ready to control braking and steering. [↑SS-C3]

ESS-RESTART-2: It is hazardous to issue START when the vehicle is not AUTO-STOPPED just as it is hazardous to do so when the feature is not enabled [↑SS-C3]. Issuing the command may interfere with the STOP command or other subsystems.

ESS-RESTART-3: STOP-START should not issue START when the driver is not present as no one is available to control the vehicle's velocity (if in a forward transmission range) and allowing the engine to run unnecessarily violates the functional purpose to reduce engine idle time [↑ G1].

ESS-RESTART-4: SS cannot issue START effectively when a restart is not possible, which will likely be due to lack of available power from the battery but may also arise from mechanical or electrical failure. In this case SS should issue the EPB [??] to secure the vehicle.

ESS-RESTART-5: It is safe to restart the engine because power is not translated to the wheels in either Park or Neutral. If the auxiliary power needs are sufficiently high, then the engine should be restarted to meet them.

ESS-RESTART-6: It is safe to restart the engine because power is not translated to the wheels in either Park or Neutral, but starting the engine without a need violates the goal of reducing engine idle time [↑ G1].

ESS-RESTART-7: SS should issue START if the driver shifts to Reverse while AUTO-STOPPED as it is required by FMVSS 102. Doing so is safe in this case because the vehicle is being held and motion will not resume until the system moves to another state.

ESS-RESTART-8: SS should issue START if the driver shifts to Reverse while AUTO-STOPPED as the driver may need to move back to avoid a hazard, such as if stuck partway out in an intersection [↑ DC4]. FMVSS 102 prohibits restarting the engine in Reverse unless the brake is held which may conflict with the need of the driver in the scenario described.

Note: *FMVSS 102 specifically references the service brake; it does not comment on the use of a Park Brake and does not differentiate between the driver applying the service brake and automation doing so (Auto-Hold).*

ESS-RESTART-9: SS should provide START because the wheels are rotating and not doing so will inhibit the driver's ability to control motion using power steering and brakes [↑ SS-C1.2].

ESS-RESTART-10: Some drivers will pass through this state when attempting to take off from a hill, holding the brake until they feel the engine torque take them forward. If START is provided, then something has to resolve the conflict between the vehicle being held and the gas pedal being applied as they will fight each other at the vehicle level and may cause lurching [↑ H1, H3]. Functionally, SS should provide START if the driver request engine torque using the gas pedal [↑ SS-G1].

ESS-RESTART-11: Providing power via START may induce a lurch when engine torque engages; however, providing START is necessary so that the driver has full control of the power steering and braking systems to control the vehicle's motion [↑ SS-C1.2].

ESS-RESTART-12: It is possible that the vehicle will begin to move when idle torque engages; however, it is necessary to provided START and engine power to fully satisfy the restart clause of [↑ SS-G1].

ESS-RESTART-13: Not providing START is hazardous because the driver will not have power steering and braking available to control the vehicle's motion [↑ SS-C1.2].

Note: *Perhaps this case can be prevented by using the EPB to hold the vehicle while it is AUTO-STOPPED.*

ESS-RESTART-14: It is safe to issue START because the vehicle is held; however, doing so will defeat the goal to reduce unnecessary engine idle time [↑ G1].

ESS-RESTART-15: It is safe to issue START because the vehicle is held; it is functionally required as the auxiliary power needs exceed the capacity of the battery [↑ SS-C2.2].

ESS-RESTART-16: If START is issued then it is possible that the vehicle will begin to move more rapidly once idle torque engages. If the driver's foot remains off of the gas pedal the idle torque should not greatly affect the trajectory of the vehicle. If START is not issued, the vehicle may continue to move but the driver will not be able to control its motion with power steering and brakes. START should be provided to satisfy [↑ SS-C1.2].

ESS-RESTART-17: Providing START is hazardous in this case because nothing is holding the vehicle and it will move when creep torque engages [↑ H1, H2, H3]. Also, none of the restart criteria have been met, so the system should remain in AUTO-STOPPED mode to satisfy its functional goals of reducing engine idle time [↑ G1].

ESS-RESTART-18: Providing START is hazardous in this case because nothing is holding the vehicle and it will move when creep torque engages [↑ H1, H2, H3]. Not providing START violates the functional requirement that the engine must be restarted to meet auxiliary power needs [↑ SS-C2.2].

ESS: RESTART Command

		F	F	R*	S-R*	S	F*	S-F	F*	S	F	S	F*
ESS Enabled =	Yes	T	T	T	T	T	T	T	T	T	T	T	T
	No												
Auto-Stopped =	Yes	T	T	T	T	T	T	T	T	T	T	T	T
	No												
Driver Present =	Yes	T	T	T	T	T	T	T	T	T	T	T	T
	No												
PRNDL =	P	T											
	R			T	T								
	N		T										
	D | L					T	T	T	T	T	T	T	T
Gas Pedal =	Pressed					T	T	T	T				
	Not Pressed									T	T	T	T
Vehicle Held =	Yes			T		T	T			T	T		
	No				T			T	T			T	T
Wheels Rotating =	Yes					T		T		T		T	
	No						T		T		T		T
Restart Possible =	Yes	T	T	T	T	T		T	T	T	T	T	T
	No						T						
Auxiliary Power Needs =	Above	T	T										T
	Below												

ESS – APPLY EPB

Context ID	Auto-Stopped	Driver Present	Restart Possible	Providing Causes Hazard	Not Providing Causes Hazard	Providing Required for Function	Not Providing Required for Function
ESS-EPB-1	No	*	*	x			x
ESS-EPB-2	Yes	No	*		x		
ESS-EPB-3	Yes	Yes	No		x		
ESS-EPB-4	Yes	Yes	Yes				

ESS-EPB-1: SS should not attempt to apply the EPB when it is not Auto-Stopped, doing so is outside the bounds of SS goals and authority [↑ SS-G1, SS-C3].

ESS-EPB-2: SS should issue the EPB if it is AUTO-STOPPED and the driver leaves the vehicle [↑ DC5]. This may occur if the driver believes the vehicle is 'off' – either they forget about SS or are unaware.

ESS-EPB-3: SS should issue the EPB if it is AUTO-STOPPED and a restart is not possible. Without a restart it is possible that additional hydraulic pressure will not be available in the brakes and a rollaway may occur. This would occur if the vehicle fails to START or the battery is depleted in some other way while AUTO-STOPPED. [↑ DC4.1]

ESS-EPB-4: SS may issue the EPB when AUTO-STOPPED, this is safe but not necessary.

ESS: APPLY EPB Command			S	S
	Auto-Stopped =	Yes	T	T
		No		
	Driver Present =	Yes		T
		No	T	
	Restart Possible =	Yes		
		No		T

ACC w/SG – ENGAGE

Context ID	ACC Enabled	ACC Engaged	Driver Present	PRNDL	Driver Brake Pedal Input	Wheels Rotating	Providing Causes Hazard	Not Providing Causes Hazard	Providing Functional	Not Providing Functional
SG-E-1	No	*	*	*	*	*	x			
SG-E-2	Yes	Yes	*	*	*	*				x
SG-E-3	Yes	No	No	*	*	*	x			
SG-E-4	Yes	No	Yes	!=D	*	*	x			x
SG-E-5	Yes	No	Yes	D	Pressed	*	x			
SG-E-6	Yes	No	Yes	D	Not Pressed	Yes			x	
SG-E-7	Yes	No	Yes	D	Not Pressed	No	x			

SG-E-1: SG should not ENGAGE if the feature is not enabled [↑ SG-C4].

SG-E-2: It is assumed that issuing ENGAGE will have no effect if the feature is already in CRUISE mode.

SG-E-3: SG should not ENGAGE when the driver is not present as this may lead to vehicle motion without supervisory control or a means to steer [↑ H3].

SG-E-4: SG may not ENGAGE when not in Drive. Doing so violates the intent of maintaining cruise in a forward transmission range [↑ SG-G1] and may surprise the driver in other contexts [↑ DC3, DC4]

SG-E-5: SG may not ENGAGE if the brake pedal is pressed. The brake pedal is a means for the driver to disengage CRUISE [→ SG-DE-5] and allowing the same action to be taken while ENGAGING may confuse the driver [↑ DC4]. Also, issuing ENGAGE while the brakes are actuating may lead to a conflict at the vehicle dynamics level as the brakes and propulsion system conflict.

SG-E-6: SG should issue ENGAGE when the previous criteria are satisfied and the wheels are rotating. This is the nominal function of the feature and satisfies [↑ SG-G1].

SG-E-7: SG should not issue ENGAGE when the wheels are not rotating as doing so violates [↑ SG-C1.1].

ACC: ENGAGE Command

		F
ACC Enabled =	No	
	Yes	T
ACC Engaged =	No	T
	Yes	
Driver Present =	No	
	Yes	T
PRNDL =	!=D	
	D	T
Driver Brake Pedal Input =	Yes	
	No	T
Target Locked =	No	
	Yes	
Speed Above Threshold =	Yes	
	No	T

ACC w/SG – ACCELERATE

Context ID	ACC Enabled	ACC Engaged	Driver Present	PRNDL	Driver Pedal Input	Target Locked	Distance >= Threshold	Speed >= Threshold	Providing Causes Hazard	Not Providing Causes Hazard	Providing Functional	Not Providing Functional
SG-ACCEL-1	No	*	*	*	*	*	*	*	x			
SG-ACCEL-2	Yes	No	*	*	*	*	*	*	x			
SG-ACCEL-3	Yes	Yes	No	*	*	*	*	*	x			
SG-ACCEL-4	Yes	Yes	Yes	!=D	*	*	*	*	x			x
SG-ACCEL-5	Yes	Yes	Yes	D	Yes	*	*	*	x			
SG-ACCEL-6	Yes	Yes	Yes	D	No	Yes	No	*	x			x
SG-ACCEL-7	Yes	Yes	Yes	D	No	Yes	Yes	Yes	x			x
SG-ACCEL-8	Yes	Yes	Yes	D	No	Yes	Yes	No			x	
SG-ACCEL-9	Yes	Yes	Yes	D	No	No	*	Yes	x			x
SG-ACCEL-10	Yes	Yes	Yes	D	No	No	*	No			x	

SG-ACCEL-1: SG should not ACCELERATE if the feature is not enabled [↑ SG-C4].

SG-ACCEL-2: SG may not ACCELERATE if the feature is not engaged as no acceptable thresholds (speed or distance) have been set by the driver, a special case of violating [↑ SG-C3], and doing so may surprise the driver violating [↑ DC3, DC4].

SG-ACCEL-3: SG may not ACCELERATE when the driver is not present as there is no supervisory control over SG or the steering of the vehicle [↑ H3].

SG-ACCEL-4: SG may not ACCELERATE when not in Drive as this violates the functional intent [↑ SG-G1] and may surprise the driver when their workload is already increase, such as in Reverse, violating [↑ DC4].

SG-ACCEL-5: SG may not ACCELERATE when the driver is pressing a pedal [↑ SG-C2]. If the driver is pressing the gas, the potential additional acceleration from SG may exceed what the driver intended or can control. If the driver is pressing the brake, SG should defer and disengage per [→ SG-DE-5].

SG-ACCEL-6: SG may not ACCELERATE when the distance is less than the threshold set by the driver [↑ SG-G1.2, SG-C3].

SG-ACCEL-7: SG may not ACCELERATE when the speed is greater than the threshold set by the driver [↑ SG-G1.1, SG-C3].

SG-ACCEL-8: SG should ACCELERATE if the speed is less than the set speed and the distance gap is greater than the threshold to meet its functional purpose [↑ SG-G1].

SG-ACCEL-9: This is a variant of [← SG-ACCEL-7] in which no target is locked, thus the only constraint is speed. SG may not ACCELERATE when the speed is greater than the threshold set by the driver [↑ SG-G1.1, SG-C3].

SG-ACCEL-10: This is a variant of [← SG-ACCEL-8] in which no target is locked, thus the only constraint is speed. SG should ACCELERATE if the speed is less than the threshold set by the driver [↑ SG-G1].

ACC: ACCELERATE Command

		F	F
ACC Enabled =	No		
	Yes	T	T
ACC Engaged =	No		
	Yes	T	T
Driver Present =	No		
	Yes	T	T
PRNDL =	!=D		
	D	T	T
Driver (Either) Pedal Input =	Yes		
	No	T	T
Target Locked =	No		
	Yes		T
Distance Above Threshold =	No	T	T
	Yes		
Speed Above Threshold =	Yes		
	No	T	

ACC w/SG – DECELERATE

Context ID	ACC Enabled	ACC Engaged	Driver Present	PRNDL	Driver Pedal Input	Target Locked	Distance >= Threshold	Speed >= Threshold	Providing Causes Hazard	Not Providing Causes Hazard	Providing Functional	Not Providing Functional
SG-DECELERATE-1	No	*	*	*	*	*	*	*	x			
SG-DECELERATE-2	Yes	No	*	*	*	*	*	*	x			
SG-DECELERATE-3	Yes	Yes	No	*	*	*	*	*				
SG-DECELERATE-4	Yes	Yes	Yes	!=D	*	*	*	*	x			
SG-DECELERATE-5	Yes	Yes	Yes	D	Yes	*	*	*	x			
SG-DECELERATE-6	Yes	Yes	Yes	D	No	Yes	No	*		x	x	
SG-DECELERATE-7	Yes	Yes	Yes	D	No	Yes	Yes	Yes		x	x	
SG-DECELERATE-8	Yes	Yes	Yes	D	No	Yes	Yes	No				x
SG-DECELERATE-9	Yes	Yes	Yes	D	No	No	*	Yes		x	x	
SG-DECELERATE-10	Yes	Yes	Yes	D	No	No	*	No				x

SG-DECEL-1: SG should not DECELERATE if the feature is not enabled [↑ SG-C4].

SG-DECEL-2: SG should not DECELERATE if the feature is not engaged, as this may surprise the driver and interfere with his use of the brakes or other controller's use of the brakes [↑ SG-C4].

SG-DECEL-3: SG should not control the vehicle without a driver in the seat.

SG-DECEL-4: SG should not DECELERATE if the vehicle is shifted from drive, as this violates the functional purpose of the feature – that it will control vehicle motion while in drive [↑ SG-G1].

SG-DECEL-5: SG may not DECELERATE when the driver is pressing a pedal [↑ SG-C2]. If the driver is pressing the gas, the deceleration from SG may create a conflict in the vehicle's dynamics and make it difficult to control. If the driver is pressing the brake, SG should defer and disengage per [→ SG-DE-5] – failing to do so may interfere with the driver's braking.

SG-DECEL-6: SG should issue DECELERATE if the distance to a target is less than the acceptable threshold. Not doing so will violate [↑ SG-C3].

SG-DECEL-7: SG should DECELERATE if the vehicle speed is greater than the driver set threshold [↑ SG-C3].

SG-DECEL-8: SG should not provide DECELERATE if the distance and speed are acceptable because it will fail to meet its functional goal of maintaining the set speed [↑ SG-G1].

SG-DECEL-9: SG should provide DECELERATE if the speed is over the driver set speed [↑ SG-C3].

SG-DECEL-10: If there is no target in front (open road), SG should not provide DECELERATE if the speed is below the set speed as it will then fail to meet its functional goal [↑ SG-G1].

ACC: DECELERATE Command

		S-F	S-F	S-F
ACC Enabled =	No			
	Yes	T	T	T
ACC Engaged =	No			
	Yes	T	T	T
Driver Present =	No			
	Yes	T	T	T
PRNDL =	!=D			
	D	T	T	T
Driver (Either) Pedal Input =	Yes			
	No	T	T	T
Target Locked =	No			T
	Yes		T	
Distance Above Threshold =	No	T	T	
	Yes			
Speed Above Threshold =	Yes			T
	No	T		

ACC w/SG – DISENGAGE

Context ID	ACC Enabled	ACC Engaged	Driver Present	PRNDL	Driver Brake Pedal Input	Target Locked	Wheels Rotating	Providing Causes Hazard	Not Providing Causes Hazard	Providing Functional	Not Providing Functional
SG-DE-1	No	Yes	*	*	*	*	*			x	
SG-DE-2	Yes	No	*	*	*	*	*			x	
SG-DE-3	Yes	Yes	No	*	*	*	*				
SG-DE-4	Yes	Yes	Yes	!=D	*	*	*			x	
SG-DE-5	Yes	Yes	Yes	D	Pressed	*	*		x		
SG-DE-6	Yes	Yes	Yes	D	Not Pressed	*	Yes				x
SG-DE-7	Yes	Yes	Yes	D	Not Pressed	No	No		x		
SG-DE-8	Yes	Yes	Yes	D	Not Pressed	Yes	No				x

SG-DE-1: The way to get to this state is the driver pressing a button to CANCEL the feature; in that case it should DIS-ENGAGE [↑ SG-C2, SG-C4].

SG-DE-2: The way to get to this state is the driver pressing a button to CANCEL the feature; in that case it should DIS-ENGAGE [↑ SG-C2, SG-C4].

SG-DE-3: This may occur if the driver leaves while the vehicle is at a standstill, in this case the feature should continue to hold the vehicle. Ideally it should apply the EPB, but this is not a capability in the system as it is currently designed.

SG-DE-4: If the driver shifts from drive, SG should cancel in accordance with its functional goal to be available in forward transmission ranges [↑ SG-G1].

SG-DE-5: If the driver presses the brake, SG should CANCEL so that the driver immediately gains control of the vehicle's velocity [↑ SG-C2]. This is also the accepted model for cancel a Cruise-Control feature across the industry [↑ DC4.1].

SG-DE-6: SG should not cancel, this describes normal CRUISE mode.

SG-DE-7: SG should cancel if a target is lost while the vehicle is stopped [← SG-RE-6].

SG-DE-8: SG should not cancel because doing so will prevent it from fulfilling [↑ SG-G1.3].

ACC: DIS-ENGAGE Command

		F	F	F	S	S
ACC Enabled =	No	T				
	Yes		T	T	T	T
ACC Engaged =	No		T			
	Yes	T		T	T	T
Driver Present =	No					
	Yes			T	T	T
PRNDL =	!=D			T		
	D				T	T
Driver Brake Pedal Input =	Yes				T	
	No					T
Target Locked =	No					T
	Yes					
Speed Above Threshold =	Yes					
	No					T

Appendix B: Listing of Conflicts

The following conflicts exist between the conditions/assumptions of one command and effects of another in the Conditions Tables (Table 34, Table 35, Table 36, and Table 37). The number does not reflect any relative significance or prioritization but exist purely for reference. The details in this analysis are fictional but considered to be realistic. The results are not mean to be used for design or safety evaluation but to demonstrate the application of STPA and the new analysis approach to the integration of embedded control systems.

Conflict 1: AH HOLD prior to ACC w/SG ACCELERATE violates *Brakes: Not Applied*

Conflict 2: AH HOLD prior to Driver GAS violates *Brakes: Not Applied*

Conflict 3: AH HOLD prior to Driver BRAKING violates *Brakes: Not Applied*

Conflict 4: AH ADDITIONAL-PRESSURE prior to AH ADDITIONAL-PRESSURE may violate *Battery Charge: High*

Conflict 5: AH ADDITIONAL-PRESSURE prior to AH ADDITIONAL-PRESSURE may violate *Brake Pressure: <Max*

Conflict 6: AH AP prior to ESS STOP may violate *Battery Charge: High*

Conflict 7: AH AP prior to ESS RESTART may violate the constraint *Battery Charge: High*

Conflict 8: AH AP prior to ACC w/SG issuing ACCELERATE violates the constraint *Brakes: Not Applied*

Conflict 9: AH AP prior to the Driver BRAKING violates the constraint *Battery: High*

Conflict 10: AH RELEASE prior to AP violates the constraint *Brakes: Applied by AH*

Conflict 11: AH RELEASE prior to AH RELEASE violates the constraint *Brakes: Applied by AH*

Conflict 12: AH RELEASE prior to AH APPLY EPB violates the constraint *Brakes: Applied by AH*

Conflict 13: AH RELEASE prior to ESS STOP violates the constraint *Vehicle Held: Yes*

Conflict 14: AH RELEASE prior to ESS RESTART violates the constraint *Vehicle Held: Yes*

Conflict 15: AH RELEASE prior to the Driver LEAVING violates the constraint *Vehicle Held: Yes*

Conflict 16: AH AH APPLY EPB prior to ACC w/SG ACCELERATE violates the constraint *Vehicle Held: No*

Conflict 17: AH AH APPLY EPB prior to the Driver GAS violates the constraint *Vehicle Held: No*

Conflict 18: ESS STOP prior to AH HOLD may violate *Hydraulic Power: Available*

Conflict 19: ESS STOP prior to AH AP may violate *Hydraulic Power: Available*

Conflict 20: AH AP while ESS AUTO-STOPPED may violate *Auxiliary Power Needs: Low*

Conflict 21: AH RELEASE while AUTO-STOPPED violates the constraint *Engine: On*

Conflict 22: ESS STOP prior to ACC w/SG ACCELERATE violates *Propulsion Power: Available*

Conflict 23: ESS STOP prior to ACC DECELERATE may violate *Hydraulic Power: Available*

Conflict 24: ESS STOP prior to Driver GAS violates *Propulsion Power: Available*

Conflict 25: ESS STOP prior to Driver BRAKE may violate *Hydraulic Power: Available*

Conflict 26: ESS RESTART prior to AH HOLD may violate *Wheels Rotating: No*

Conflict 27: ESS RESTART prior to ESS RESTART violates *AUTO-STOPPED: Yes*

Conflict 28: ESS RESTART prior to ESS APPLY EPB violates *AUTO-STOPPED: Yes*

Conflict 29: ESS RESTART prior to Driver LEAVE violates *Engine: Off*

Conflict 30: ESS APPLY EPB prior to ACC w/SG ACCELERATE violates *Vehicle Held: No*

Conflict 31: ESS APPLY EPB prior to Driver GAS violates *Vehicle Held: No*

Conflict 32: ACC w/SG ACCELERATE prior to AH HOLD violates *Wheels Rotating: No*

Conflict 33: ACC w/SG ACCELERATE prior to AH APPLY EPB violates…

Conflict 34: ACC w/SG ACCELERATE prior to ESS STOP violates *Wheels Rotating: No*

Conflict 35: ACC w/SG ACCELERATE prior to ESS APPLY EPB violates...

Conflict 36: ACC w/SG ACCELERATE prior to ACC w/SG ACCEELERATE may violate *Speed <= Threshold: Yes*

Conflict 37: ACC w/SG ACCELERATE prior to ACC w/SG DECELERATE violates *GAS: Off*

Conflict 38: ACC w/SG ACCELERATE prior to Driver LEAVE violates *Wheels Rotating: No*

Conflict 39: ACC w/SG ACCELERATE prior to Driver SHIFT may violate *Speed: <TBD*

Conflict 40: ACC w/SG DECELERATE prior to ACC w/SG ACCELERATE violates *Brakes: Not applied*

Conflict 41: ACC w/SG DECELERATE prior to Driver GAS may violate *Vehicle Held: No*

Conflict 42: Driver LEAVE prior to AH HOLD violates *Driver Present: Yes*

Conflict 43: Driver LEAVE prior to AH RELEASE violates *Driver Present: Yes*

Conflict 44: Driver LEAVE prior to ESS STOP violates *Driver Present: Yes*

Conflict 45: Driver LEAVE prior to ESS RESTART violates *Driver Present: Yes*

Conflict 46: Driver LEAVE prior to ACC w/SG ACCELERATE violates *Driver Present: Yes*

Conflict 47: Driver LEAVE prior to ACC w/SG DECELERATE violates *Driver Present: Yes*

Conflict 48: Driver SHIFT prior to AH HOLD may violate *Range: D*

Conflict 49: Driver SHIFT prior to AH RELEASE may violate *Range: D | P*

Conflict 50: Driver SHIFT prior to ESS STOP may violate *Range: !=P,R,N*

Conflict 51: Driver SHIFT prior to ESS RESTART may violate *Range: !=P,R,N*

Conflict 52: Driver SHIFT prior to ACC w/SG ACCELERATE may violate *Range: D*

Conflict 53: Driver SHIFT prior to ACC w/SG DECELERATE may violate *Range: D*

Conflict 54: Driver SHIFT prior to ACC w/SG LEAVE may violate *Vehicle Held: Yes*

Conflict 55: Driver GAS prior to AH HOLD violates *Wheels Rotating: No and Gas Pedal: No*

Conflict 56: Driver GAS prior to AH APPLY EPB violates…

Conflict 57: Driver GAS prior to ESS STOP violates *Gas Pedal: No*

Conflict 58: Driver GAS prior to ESS APPLY EPB violates…

Conflict 59: Driver GAS prior to ACC w/SG ACCELERATE violates *Driver Pedal Input: No*

Conflict 60: Driver GAS prior to ACC w/SG DECELERATE violates *Driver Pedal Input: No*

Conflict 61: Driver GAS prior to Driver LEAVE violates *Wheels Rotating: No*

Conflict 62: Driver GAS prior to Driver SHIFT may violate *Speed: <TBD*

Conflict 63: Driver BRAKE prior to AH AP may violate *Brake Pressure: <Max*

Conflict 64: Driver BRAKE prior to ACC w/SG ACCELERATE violates *Driver Pedal Input: No*

Conflict 65: Driver BRAKE prior to ACC w/SG DECELERATE violates *Driver Pedal Input: No*

Conflict 66: Driver BRAKE prior to GAS violates *Vehicle Held: No*