

Software Requirements Specification (SRS)

Active Park Assist

Team: APA1

Authors: Madison Bowden, Kunyu Chen, Dylan Iseler, Ben Read, Daniel Webb

Customer: Eileen Davidson, Ford Motor Company

Instructor: Dr. Betty Cheng

1. Introduction

This document outlines the software requirements for the Active Park Assist project and is organized as follows: Section 1 provides a brief introduction to the project and the Software Requirements Specifications, Section 2 outlines the product, Section 3 explicitly states the project requirements, Section 4 provides models of the requirements, Section 5 includes a rapid prototype of the product, followed by references in Section 6 and the point of contact in Section 7.

1.1 Purpose

The purpose of the Software Requirements Specification is to define assumptions and requirements for the product. The goal is to provide a high-level understanding of the project and clarity for all stakeholders involved. This should converge understanding among the customers, management, and development teams and be the starting point for development.

1.2 Scope

The software product to be produced is an Active Park Assist feature, which is part of an embedded automotive system. This APA feature is intended to accurately and safely park a vehicle without driver intervention. It is also motivated by reducing the stress of parking and prevent collisions while perpendicular or parallel parking.

When activated by a user, APA will select a perpendicular or parallel parking space (based on the user's selection for parking type), and autonomously shift gears, accelerate, brake, and steer the vehicle into the verified spot without colliding with parked cars or other obstacles that may cross the vehicle trajectory and obstruct the vehicle's motion. Additionally, the software will also ensure that the parking request was initiated by the driver, not an external attack.

1.3 Definitions, acronyms, and abbreviations

The following definitions from Active Park Assist Project Description [4]

- APA – Active Park Assist:** The product to be developed for safely and accurately parking the vehicle outlined in these specifications
- ASIL – Automotive Safety Integrity Level:** An automotive standard for which the APA software must conform to the highest level
- BCS – Brake Control Subsystem:** A subsystem of APA that accepts inputs from the Park Control Subsystem to slow the vehicle in order to meet the required trajectory by applying the brakes
- HMI – Human Machine Interface:** A subsystem of APA that accepts driver inputs, displays camera information, and handles telltales/warnings.
- PCS – Park Control Subsystem:** A subsystem of APA that manages the Active Park Assist feature. It accepts the driver input from the HMI subsystem, calculates the vehicle trajectory based on information from the Vehicle Position Subsystem, and issues commands to the other subsystems.
- PMS – Powertrain Management Subsystem:** A subsystem of APA that accepts inputs from the Park Control Subsystem to accelerate the vehicle and select the gear lever position in order to meet the required trajectory.
- SCS – Steering Control Subsystem:** A subsystem of APA that accepts inputs from the Park Control Subsystem to steer the vehicle in order to meet the required trajectory.
- SRS – Software Requirements Specification:** This document outlining the APA product software requirements
- VPS – Vehicle Position Subsystem:** A subsystem of APA that processes data from the vehicle's cameras/radar in order to identify parking spots and verify vehicle position throughout the duration of a parking event.

1.4 Organization

Following this Introduction in Section 1, the rest of the SRS is outlined as follows:

Section 2 – Provides a more detailed overall description for the product, its functions, constraints, assumptions/dependencies, and requirements beyond the scope of this product release.

Section 3 – Enumerates specific requirements.

Section 4 – Models the requirements graphically with UML notation, specifically defining the use cases, domain, scenarios with sequence diagrams, and state diagrams consistent with the previous models.

Section 5 – Describes a rapid prototype for the product, how to run it, and demonstrates sample scenarios.

Section 6 – Outlines references used in this work.

Section 7 – Provides a point of contact for any questions or concerns regarding this document or any of the artifacts generated herein.

2. Overall Description

This section introduces the APA feature, namely the context, primary functions, constraints, expectations about the user, assumptions and dependencies, and

Template based on IEEE Std 830-1998 for SRS. Modifications (content and ordering of information) have been made by Betty H.C. Cheng, Michigan State University (chengb at msu.edu)

requirements determined to be out of scope for the current version of APA that are developed based on customer negotiations.

2.1. Product Perspective

APA is one of many features of an autonomous driving system. As such, it is an embedded automotive system that coordinates with other systems within the vehicle. Being an embedded system constrains the possible hardware and software that can be used. Additionally, APA itself coordinates the work of autonomously parking by utilizing several existing subsystems that are defined in SRS 1.3. These existing software interfaces that must be used are the BCS, SCS, VPS, PMS, and the HMI, which must work with the PCS that is being developed to manage the APA feature holistically. The HMI is the user interface that allows the driver to control APA setup and view the surrounds via external cameras and obstacle warnings. The system is also constrained by the vehicle's hardware, including sharing its battery, WiFi, Bluetooth and CAN capabilities with the vehicle's other systems. Additionally, it faces the constraints of the sensors that are mounted around the vehicle (which are used by VPS). The quality of the cameras, ultrasonic, and radar systems all impact the quality of the APA system, and any fault in that hardware must be considered for safety and quality assurance.

2.2. Product Functions

APA must first be turned on by the driver through the HMI interface. The system will then prompt the user to select the type of parking desired (parallel or perpendicular) through the HMI. After the driver has selected the type, the cameras in the VPS will begin scanning the area to identify possible parking spaces. When a space has been identified, the HMI will prompt the user to validate the selected option and trajectory. After the user has validated the selection, APA will take over steering (with SCS), braking (BCS), acceleration/gear shifting (PMS) and use VPS to maneuver the vehicle from its current location to the target space, coordinated by PCS. While on the path to the target space, APA must be monitoring for obstacles with the ultrasonic sensors, cameras, and radar. If the system detects an obstacle, the system will come to a stop, notify the driver of the obstacle, and deactivate autonomous parking. The driver then receives control of the vehicle. During the maneuver, the HMI will display the surroundings/camera views and any warnings for the driver. When the vehicle reaches the target, PMS will shift into park to end the maneuver. Once the vehicle is successfully parked, the HMI will notify the driver of the end of the maneuver and that APA is deactivated, returning controls to the driver. If at any time during the maneuver the driver steps on the brake pedal or moves the steering wheel, APA override occurs, and the driver regains control of the vehicle. The system must also be robust against malicious cyber-attacks and verify that the controls come from the driver/APA, not an external party. In the event of a cyberattack, APA will be disabled. In the event of a subsystem/sensor failure, the vehicle will come to a stop, notify the driver of the malfunction and advise the driver to take the vehicle to a dealership. APA will be deactivated until the error is resolved.

2.3. User Characteristics

The user is the person driving the vehicle. It is expected that he or she should be a legally authorized driver and able to follow the HMI instructions to activate and set up APA. The system should be made in a way that any driver can easily use the system without intervention, as few assumptions about the user are made. The driver must also

be able to drive the vehicle once APA has been deactivated. It is likely that most users of the system will be new drivers or elderly drivers who may have more trouble parking.

2.4. Constraints

Security

The system must have a means of verifying that the driver has initiated the request, and the request was not a result of a fault in the HMI system, or a malicious 3rd party attack. The system shall also maintain a firewall to block unauthorized commands that come from outside the vehicle, either over Bluetooth or WIFI.

Safety

The system shall stop the vehicle, return control to the driver, and deactivate itself when obstacles move into the vehicle path during the parking maneuver. Once a parking event is initiated, it shall be complete in no more than 20 seconds. A single point failure of any sensor input shall be detectable. Sensor failure will result in APA being deactivated and the driver being notified to go to the dealership for repair.

2.5. Assumptions and Dependencies

1. The user interacts with the HMI through a touch-screen interface.
2. The software receives user interaction messages (either from the touch screen, steering wheel, or brakes) correctly, quickly, and in order.
3. The Parking Control Subsystem uses CAN to interface with the necessary subsystems, and data packets are delivered correctly, on time, and in order. The protocol recovers from errors in data packet transmission.
4. The hardware in the vehicle has adequate primary and secondary memory to perform all necessary computations.
5. The Parking Control Subsystem always correctly identifies error states in external sensors.
6. The Vehicle Positioning Subsystem always identifies parallel and perpendicular parking spots correctly.
7. The vehicle is in working mechanical condition (transmission, brakes, steering, accelerator, engine, sensors).
8. The sensors and accompanying software always identify obstacles accurately.
9. The driver must be going < 7 mph in order for the system to accurately identify a spot.

2.6. Apportioning of Requirements

1. The algorithm for calculating the vehicle trajectory.
2. The computer vision/sensing software for identifying parking spaces.
3. Specific technology for detecting external intrusion.
4. HMI should connect to the users phone via Bluetooth and also to the vehicle WIFI.

3. Specific Requirements

1. System allows vehicle to automatically park in parallel or perpendicular space
2. System interacts with user through HMI interface.
3. System will accelerate/brake, choose forward/reverse as needed, and steer along the calculated trajectory into the selected space.
4. System must be activated through HMI interface.
5. System allows driver to select type of parking space (parallel/perpendicular).
6. System uses 1 camera mounted in the front of the vehicle and 1 camera mounted in the rear of the vehicle to identify parking spaces.
7. System uses 4 ultrasonic sensors on each side of the vehicle to measure the space between vehicles for parallel parking and 2 ultrasonic sensors on the front and back.
8. System identifies spots that are large enough for the vehicle (1.2 times the vehicle length for parallel parking or 9 feet for perpendicular) and plans the trajectory to the spot.
9. Driver validates the spot selected, and the vehicle's path for the spot through HMI.
10. The vehicle must have automatic transmission.
11. Only after selection is validated, APA will shift the transmission into reverse/drive as needed, accelerate, brake, and steer vehicle into the spot.
12. Radar/camera system will continuously monitor vehicle position while maneuvering to ensure no collisions with other parked vehicles/obstacles (solid, immovable objects, people or animals).
13. Radar sensors adjacent to cameras to detect objects within 10 feet of the vehicle on its trajectory, specifically while reversing.
14. High quality cameras must have a view of 170 degrees and 50 feet behind the vehicle.
15. Cameras must display the video scene to the dashboard via Bluetooth or CAN with the vehicle, the vehicle's trajectory and surroundings and an aerial view.
16. After the maneuver, APA shifts into Park to finish the parking process.
17. After the maneuver, APA is inactive, the HMI indicates the end of the parking process, and the driver resumes control of the vehicle.
18. The system should work without driver intervention.
19. The system should deactivate whenever the driver steps on the brake or moves the steering wheel. The driver should be notified that APA has aborted through HMI.
20. The system will identify obstacles that have moved into the path after the maneuver started and stop the vehicle, shut down APA, and return control to the driver.
21. The system will verify that the driver started the request to park rather than a fault in HMI system or malicious attack
22. The maneuver shall be completed in no more than 20 seconds.
23. A single failure of any sensor input shall be detectable, and the driver should be notified of the malfunction and directed to go to the dealership. The system will also shut down when the malfunction is encountered.
24. System should be able to distinguish between obstacles (solid, immovable objects, people, or animals), snow, curbs, debris on the ground, and vehicles.
25. The system must be ASIL d certified.

4. Modeling Requirements

The models below are intended to visually show the logical paths for the APA system as interpreted by the development team based on the abovementioned requirements. The section follows with four types of models: use case diagrams to show high-level functionality, domain model to show key entities and their relationships, sequence diagrams to show changes over time in the system for the scenarios posed by the customer, and a state diagram to show how the entities in the system can interact with each other over time.

4.1. Use Case Diagrams

Figure 1 below depicts use cases for this system. The external actor Driver is shown on the left, outside of the system boundary (blue box). The use cases (defined below in Table 1) are circled inside of the system boundary and include: Activate APA, Select Parallel or Perpendicular Parking, Validate Parking Spot & Trajectory Choice, Deactivate APA, Identify Parking Spot & Path, Park Vehicle, Detect & Avoid Obstacles, Steer Control, Acceleration & Gear Shifting Control, Brake Control.

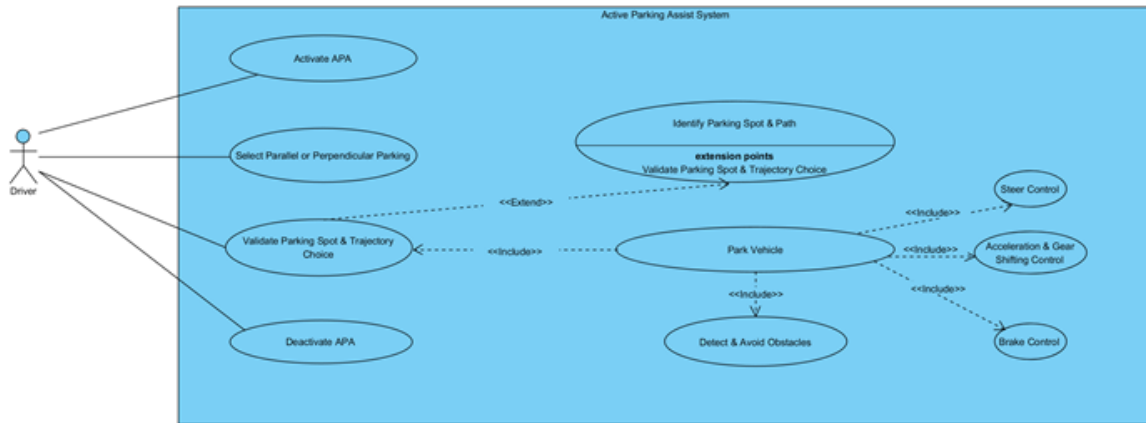


Figure 1: APA Use Case Diagram

System Boundary: Border of blue box

System includes: Park Control Subsystem, Powertrain Management Subsystem, HMI Subsystem, Brake Control Subsystem, Steering Control Subsystem, Vehicle Position Subsystem

Table 1: Use Case Descriptions

Use Case:	Activate APA
Actors:	Driver, System
Description:	The Driver must first use the Human Machine Interface (HMI) to activate Active Park Assist (APA) before any of the APA features are available. Once activated, the Driver must then Select Parallel or Perpendicular Parking to use other APA features. While the APA is active, the HMI must indicate this is so (APA button will be lit). The APA system must be ASIL d certified and properly working for APA to function. If any malfunctions are detected while being activated, or at any other time while parking, the system will notify the Driver through the

	HMI and indicate that the vehicle should be taken to a dealership for repair.
Type:	Primary & Essential
Includes:	N/A
Extends:	N/A
Cross-refs:	2, 4, 25
Use Cases:	Activate APA must be done before any other features of APA are active to the Driver, beginning with Select Parallel or Perpendicular Parking

Use Case:	Deactivate APA
Actors:	Driver, System
Description:	<p>APA is deactivated in the following scenarios:</p> <ol style="list-style-type: none"> 1. Driver depresses the brakes 2. Driver moves the steering wheel 3. APA has completed the parking maneuver <p>To deactivate APA, all APA subsystems and functions are turned off (VPS, HMI, PCS, BCS, SCS, PMS). Once APA has been deactivated and the vehicle is in park, Driver is notified of APA deactivation through HMI notification. The Driver then resumes control of the vehicle.</p>
Type:	Primary & Essential
Includes:	N/A
Extends:	N/A
Cross-refs:	17, 19
Use Cases:	Park Vehicle will Deactivate APA upon successful maneuver

Use Case:	Select Parallel or Perpendicular Parking
Actors:	Driver
Description:	After Driver has activated APA through Activate APA, the Driver selects desired parking type as Parallel or Perpendicular through the HMI.
Type:	Primary & Essential
Includes:	N/A
Extends:	N/A
Cross-refs:	2, 5
Use Cases:	Required step in fully activating the system after Activate APA Select Parallel or Perpendicular Parking used in Identify Parking Spot & Trajectory and therefore must be completed beforehand

Use Case:	Identify Parking Spot & Trajectory
Actors:	System
Description:	After the system has been activated and the Driver selected the parking type, the system begins scanning for available parking spaces of the appropriate type and size for the vehicle (1.2 times the length of the vehicle for parallel or 9 feet for perpendicular) using the mounted cameras, ultrasonic sensors, and radar and plans out the Trajectory of the vehicle.
Type:	Primary & Essential
Includes:	N/A
Extends:	N/A
Cross-refs:	6, 7, 8, 14

Use Cases:	Select Parallel or Perpendicular Parking before APA identifies parking spot/trajectory Spot must be identified before Validate Parking Spot & Trajectory Choice and Park Vehicle commence
-------------------	--

Use Case:	Validate Parking Spot & Trajectory Choice
Actors:	Driver
Description:	After the system has been activated, the Driver has selected the parking type, and the system has found a parking space/trajectory, the Driver then must accept the identified parking spot and vehicle trajectory through the HMI. This is also a necessary step in ensuring APA security, as APA must only be activated by the Driver, not an external party.
Type:	Primary & Essential
Includes:	N/A
Extends:	Identify Parking Spot & Trajectory
Cross-refs:	2, 9, 15, 21
Use Cases:	Identify Parking Spot & Trajectory must be completed before validation Validation begins Park Vehicle

Use Case:	Park Vehicle
Actors:	System
Description:	After the parking spot and trajectory have been verified by the Driver in Validate Parking Spot & Trajectory Choice, APA uses vehicle subsystems to park the vehicle in the chosen spot by autonomously steering (SCS), accelerating (PMS), braking (BCS), and shifting gears (PMS). While parking, the vehicle must continuously avoid obstacles/collisions (through Detect & Avoid Obstacles) and allow for Driver override (with Deactivate APA). The process of completing the maneuver must be timely. Once the maneuver is completed, APA will shift into Park and Deactivate APA, thus alerting the Driver via HMI of parking completion. Any conflict in sensor data (coming from radar/camera sensors) will flash a notification via HMI indicating which sensor(s) may have failed and to take the vehicle to a dealership to get fixed. If a sensor has failed, APA will stop the vehicle and be then be deactivated, and the Driver resumes control.
Type:	Primary & Essential
Includes:	Validate Parking Spot & Trajectory Choice, Detect & Avoid Obstacles, Steer Control, Brake Control, Acceleration & Gear Shifting Control
Extends:	N/A
Cross-refs:	1, 3, 7, 10, 11, 15, 16, 17, 18, 19, 21, 22, 23
Use Cases:	Validate Parking Spot & Trajectory Choice must be completed first Continuously Detect & Avoid Obstacles while parking Uses Steer Control, Brake Control, Acceleration & Gear Shifting Control to park vehicle in selected space Deactivate APA to be done upon completion of maneuver

Use Case:	Detect & Avoid Obstacles
Actors:	System
Description:	While the system is active and in control of driving to Park Vehicle, APA uses cameras and sensors (VPS) to detect obstacles in the vehicle's Trajectory (within 10 feet of the vehicle) and uses PMS, SCS, and BCS to modify the vehicle's motion to avoid collisions with detected obstacles, coordinated by PCS and displays camera output and obstacle warnings to Driver via HMI. If an obstacle is detected, the vehicle must stop, notify the driver, and return control to the driver to complete the maneuver. The system must be able to differentiate between obstacles (including solid, immovable objects, humans, and animals), vehicles, snow/debris on the ground, and curbs.
Type:	Primary & Essential
Includes:	N/A
Extends:	N/A
Cross-refs:	7, 12, 13, 14, 20, 24
Use Cases:	Must continuously Detect & Avoid Obstacles to safely Park Vehicle

Use Case:	Steer Control
Actors:	System
Description:	APA will autonomously control the power steering capabilities of the vehicle to follow the chosen trajectory into the selected parking spot and preventing collisions with obstacles.
Type:	Primary & Essential
Includes:	N/A
Extends:	N/A
Cross-refs:	1, 3, 11
Use Cases:	Park Vehicle will use Steer Control to move the vehicle along the trajectory

Use Case:	Brake Control
Actors:	System
Description:	APA will autonomously control the braking system in order to safely complete the parking maneuver.
Type:	Primary & Essential
Includes:	N/A
Extends:	N/A
Cross-refs:	1, 3, 11
Use Cases:	Park Vehicle will use Brake Control to complete the maneuver and prevent collisions if obstacles are detected on the trajectory

Use Case:	Acceleration & Gear Shifting Control
Actors:	System
Description:	APA will modify the vehicle's speed and direction (forward vs. reverse) by accelerating or shifting gears as needed to travel along the trajectory to the selected parking space. The vehicle must have an automatic transmission for this to work.

Type:	Primary & Essential
Includes:	N/A
Extends:	N/A
Cross-refs:	1, 3, 10, 11, 16
Use Cases:	Park Vehicle uses Acceleration & Gear Shifting Control to control vehicle movement to complete the parking maneuver

4.2. Domain Model

The domain model in Figure 2 below depicts a high-level view of the key entities and their relationships to one another. Specific definitions for the key entities are defined in the data dictionary (Table 2).

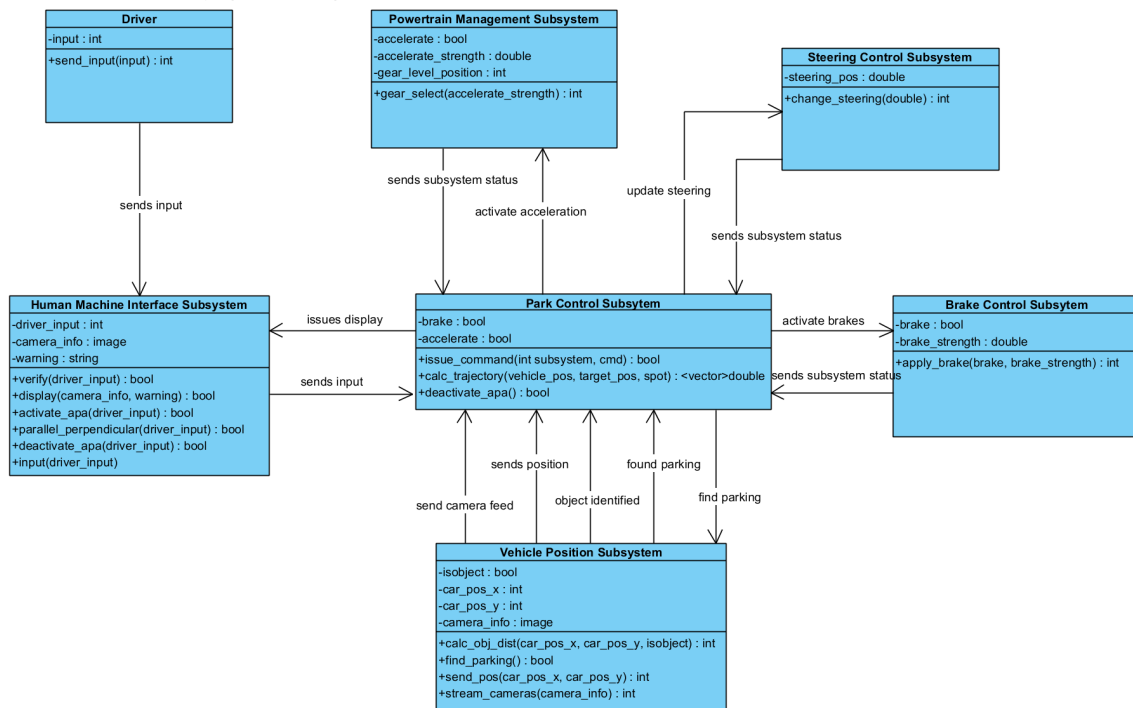


Figure 2: Domain Model

Table 2: Data Dictionary

Element Name	Description
Brake Control Subsystem	Brake Control Subsystem (BCS) accepts inputs from the Park Control Subsystem to slow the vehicle in order to meet the required trajectory.
Attributes	
	brake:bool
	Boolean for whether the system should engage the brakes. True to brake the car.

	brake_strength:double	A double corresponding to the strength of braking power to engage.
Operations		
	apply_brake(brake, brake_strength):int	If brake flag set, engage the brakes to amount brake strength. Returns system status.
Relationships	The BCS has an association with the Park Control System (PCS) since the BCS receives signals from PCS to engage/disengage brakes to control vehicle movement appropriately.	
UML Extensions	N/A	

Element Name	Description	
Driver	The Driver or user who interacts with APA via HMI.	
Attributes		
	input:int	The Driver selects a button which corresponds to an integer input.
Operations		
	send_input(input):int	The Driver's choice input is sent to the HMI to give instructions to APA.
Relationships	There's an association between Driver and HMI with the Driver pushing buttons to send signals to APA via HMI.	
UML Extensions	N/A	

Element Name	Description	
Human Machine Interface Subsystem	The Human Machine Interface Subsystem (HMI) accepts Driver inputs and displays the camera feed and warnings.	
Attributes		
	driver_input:int	An input signal from the Driver for their button selection.
	camera_info:image	The stream of images HMI receives from external cameras and displays for the Driver.
	warning:string	A signal or written warning to the Driver.
Operations		
	verify(driver_input):bool	Verify the Driver's selection and ensure the selection came from the Driver, not a third-party.
	display(camera_info, warning):string	HMI takes the video stream from cameras and displays that and

		any warnings to the audiovisual display.
	activate_apa(driver_input) : bool	Driver's selection to activate the APA system.
	parallel_perpendicular(driver_input) : bool	Driver chooses what type of parking space to look for: parallel or perpendicular.
	deactivate_apa(driver_input):bool	If the Driver decides to deactivate APA before the maneuver completes, HMI will send the signal to PCS.
	input(driver_input):void	HMI waits for the Driver to input something, so that HMI can then send the Driver's input to PCS with one of the other operations.
Relationships	There are associations between the Driver, which provides inputs to HMI, and Park Central Subsystem which receives signals from HMI/Driver to control behavior of APA.	
UML Extensions	N/A	

Element Name	Description	
Park Control Subsystem	The Park Control Subsystem (PCS) manages the Active Park Assist feature. It accepts the Driver input from the HMI subsystem, calculates the vehicle trajectory based on information from the Vehicle Position Subsystem, and issues commands to the other subsystems.	
Attributes		
	brake:bool	A Boolean flag indicating whether to activate the Brake Control Subsystem based on current position, speed, and trajectory, as well as the presence of obstacles.
	accelerate:bool	A Boolean flag indicating whether to activate the Powertrain Management Subsystem based on current position, speed, and trajectory, as well as the presence of obstacles to move the vehicle appropriately.
Operations		

	calc_trajectory(vehicle_pos, target_pos, spot): <vector>double	Takes inputs of position and target position, and the parking space to calculate the vehicle's needed trajectory to complete the parking maneuver.
	issue_command():bool	Park Control Subsystem uses the calculated trajectory to determine if another subsystem needs to do work and how much should be done for the vehicle to meet the required trajectory. Invokes operations in other subsystems and returns true on successful invocation.
	deactivate_apas():bool	Boolean indicating if PCS should end the parking session, stop managing the other parking subsystems, and return driver control.
Relationships	Park Control Subsystem has associations with all of the other subsystems as this is the manager of the APA feature and coordinates signals from all of the subsystems to complete the parking maneuver.	
UML Extensions	N/A	

Element Name	Description
Powertrain Management Subsystem	The Powertrain Management Subsystem (PMS) accepts inputs from the Park Control Subsystem to accelerate the vehicle and select the gear lever position in order to meet the required trajectory.
Attributes	
	accelerate:bool Boolean from Park Control Subsystem for activating acceleration.
	accelerate_strength:double A Double of acceleration (m/s^2) strength/amount needed to meet the trajectory coming from PCS.
	gear_lever_pos:int Position of the gear lever (forward, reverse, park).
Operations	
	gear_select(accelerate_strength):int Takes accelerate_strength to calculate which gear level position to be in. Returns system status.
Relationships	Associations with Park Control Subsystem which sends Powertrain Management signals for accelerating and the strength to accelerate.
UML Extensions	N/A

Element Name		Description
Steering Control Subsystem		The Steering Control Subsystem (SCS) accepts inputs from the Park Control Subsystem to steer the vehicle in order to meet the required trajectory.
Attributes		
	steering_pos:double	The angle of the steering wheel to control the vehicle's direction.
Operations		
	change_steering(int): int	Input integer from Park Control Subsystem that changes steering_pos. Returns system status.
Relationships	Association with Park Control Subsystem which updates the steering_pos to steer the vehicle along the correct path.	
UML Extensions	N/A	

Element Name		Description
Vehicle Position Subsystem		The Vehicle Position Subsystem (VPS) processes data from the vehicle's cameras/radar in order to identify parking spots and verify vehicle position throughout the duration of a parking event.
Attributes		
	isobject:bool	True if detected object on trajectory gets close.
	car_pos_x:int	Car's latitude
	car_pos_y:int	Car's longitude
	camera_info:image	Stream of images from the external cameras of the surroundings.
Operations		
	calc_obj_dist(car_pos_x, car_pos_y):int	Calculates distances between objects and the vehicle in order to prevent collisions.
	find_parking():bool	Uses the cameras to calculate distances between objects to find available parking space and returns true if at least one is found.
	send_pos(car_pos_x, car_pos_y):int	Sends car position to PCS. Returns 0 on success, otherwise failed.

	stream_cameras(camera_info):int	Sends camera feed to PCS to be displayed along with any warnings. Returns 0 on success, otherwise fail.
Relationships	Associations with Park Control Subsystem to send the vehicle's position and to send notification of an available parking spot or obstacles near the vehicle, and from PCS to VPS to start looking for a parking space and sending information to PCS.	
UML Extensions	N/A	

4.3. Sequence Diagrams

Sequence diagrams show object lifelines indicated by the dotted lines descending from the boxed object in order to depict how these objects interact over a given scenario. The sequence diagram descriptions are in the note boxes below the figures.

4.3.1. Scenario 1: Standard Parallel Parking

The driver activates the HMI interface. Then, the HMI interface asks what type of parking the driver wanted (parallel). With the driver's input, HMI activates the Parking Control Subsystem. The Parking Control Subsystem activates the Vehicle Position Subsystem to find a parking spot and check the distance between the vehicle and other objects. After the parking spot is found, the HMI will send a message to get the driver's permission to park. If validated by the driver, the Parking Control Subsystem will calculate a parking trajectory and manage the Powertrain Management Subsystem to control the speed of vehicle and the driving direction and manage the Break Control Subsystem to stop the car. After the vehicle finishes parking at the right spot, the Parking Control Subsystem will send a success message to the HMI, and the HMI will tell the driver the parking has finished. During the automatic parking, the driver can stop it whenever he or she wants. If the parking fails or cannot find parking spot, HMI will receive a message. The parking spot found is $>1.2x$ the length of the vehicle, and the vehicle speed is limited to < 5 mph while parking.

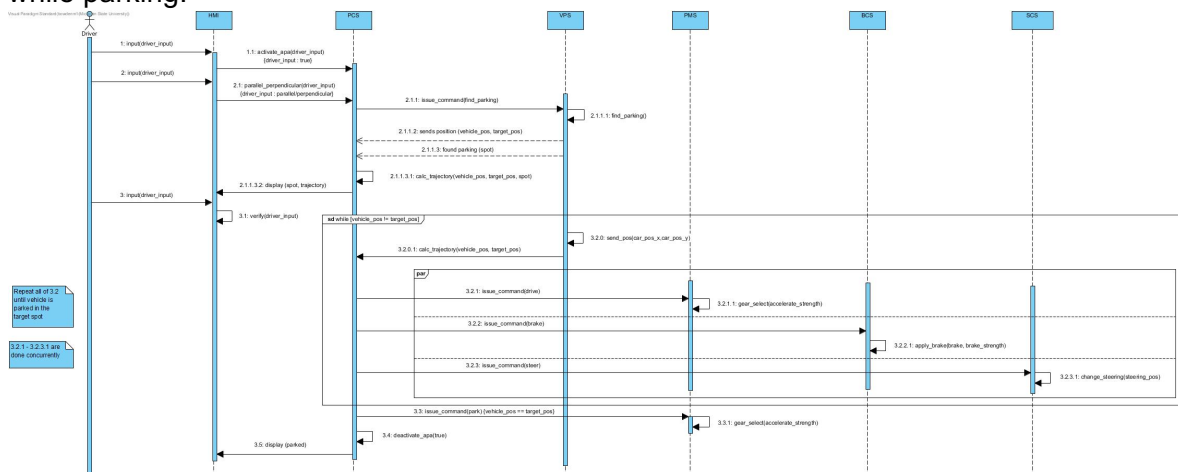


Figure 3: Sequence diagram for standard parallel parking

4.3.2. Scenario 2: Subsystem Failure

The Driver activates APA. Then, the HMI notifies PCS to be activated. The Driver then selects a parking type, and HMI sends the selection to PCS. PCS asks VPS to find a spot and to send the locations of the spot and vehicle. PCS uses this information to calculate a parking trajectory. PCS issues the results to HMI to be displayed. The Driver accepts the selection, and HMI verifies that the Driver made selection for security purposes. PCS then takes control of PMS and begins the parking maneuver. PCS applies the brakes using BCS, but BCS responds with an indication that it has encountered a fault. PCS commands PMS to shift the vehicle into park. PMS notifies PCS when this is complete. PCS deactivates APA and issues the fault code to HMI for display. The Driver sees the fault and must acknowledge it through HMI.

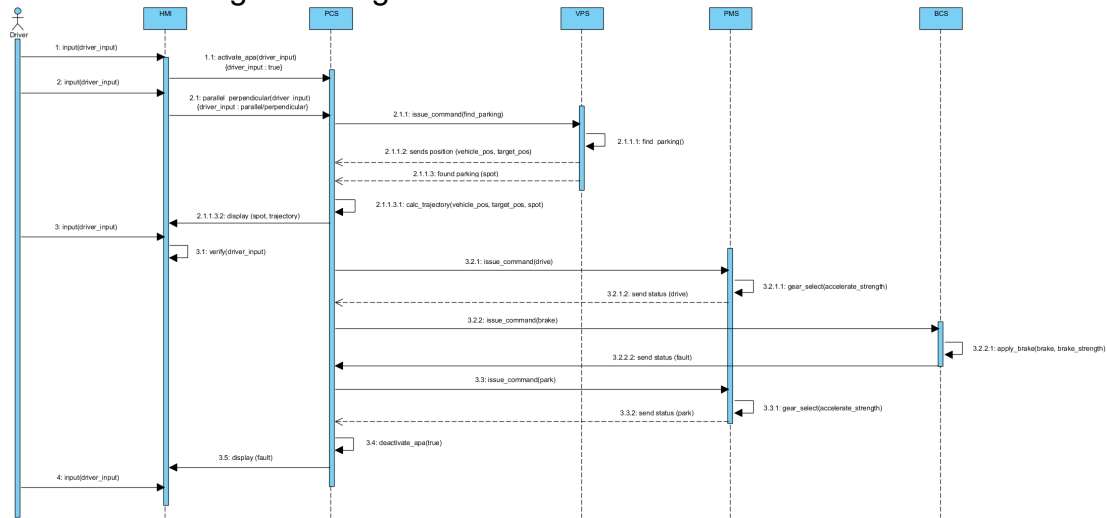


Figure 4: Sequence diagram for a subsystem malfunction

4.3.3. Scenario 3: Security Event

The Driver sends input to HMI. HMI verifies that it has come from the Driver. If true, then HMI activates APA, and APA begins issuing commands. If false, HMI ensures APA is deactivated since the Driver did not initiate the sequence.

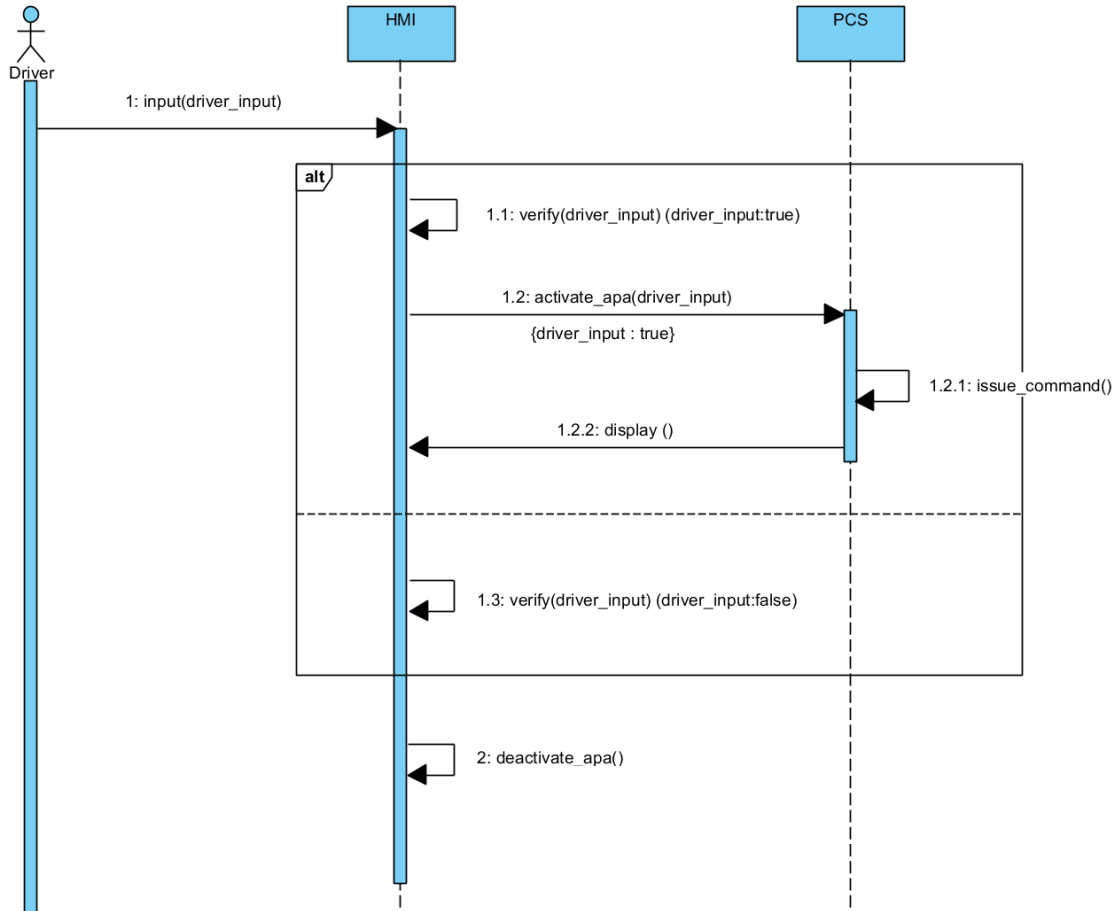


Figure 5: Sequence diagram for a security event

4.4. State Diagram

The state diagram below shows the states of the overall APA system and includes the possible states of the subsystems that make up the APA.

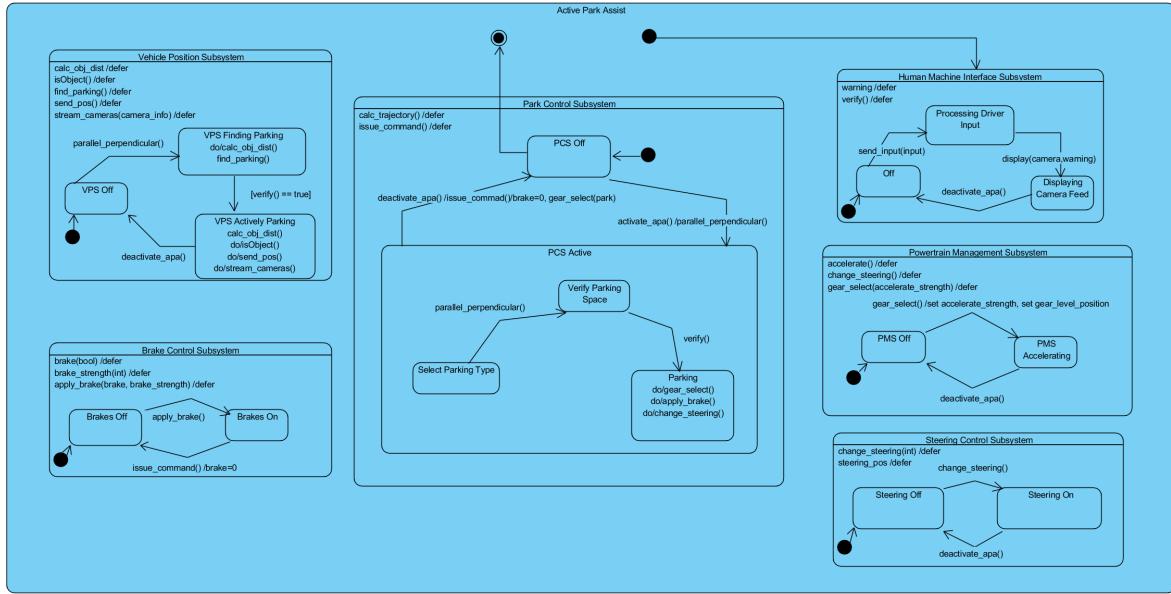


Figure 6: State diagram for APA system

5. Prototype

The purpose of the first prototype is to show the development team's interpretation of the customer's requirements and visually demonstrate the system's logic surrounding each of the customer-provided scenarios. The figures below provide examples of the user-experience with APA with the user-interface as the system progresses through various parking situations. After the user has selected various inputs to determine the simulation, a video will play showing the changes in speed with the speedometer (controlled by PMS and BCS), steering with the steering wheel position (SCS), gear-lever (PMS), and camera feed (composited aerial view from all the cameras/sensors)/warnings on the HMI (receiving input from the VPS) which are managed by the PCS. The details of the system are abstracted away in this prototype, only showing high level functionality. The screenshots below provide an abridged view of this experience.

5.1. How to Run Prototype

Prototype 1 is available online at <http://cse.msu.edu/~chenkuny/project/prototype1.html> as of November 18, 2019. Provided the user has an up-to-date internet browser and bandwidth, the prototype will run as-is, without any special plug-ins or downloads required in the browser of their choosing (tested on Google Chrome, Microsoft Edge, Safari, and Firefox on Windows 10, MacOS, and Debian Linux distributions).

The prototype page gives a description of the APA system and graphical instructions at the top of the page. The bottom of the page has the interactive HMI module, instructions for operating the HMI, and buttons for the user to select which scenario to view, given their inputs with the HMI.

The user can run the prototype by clicking on the diagram of the HMI to set up a session. As noted in the instructions for interacting with the HMI online, for the screens in which there is no user input option (ex. When not selecting parking type or verifying

the selected space/trajectory), the user *must* still click anywhere on the HMI screen to proceed to the following screen. This is a feature of the web interface, *not* the vehicle interface (those transitions occur based on time, not user inputs). If the buttons for selecting a scenario to be played are not clickable, the user has not finished interacting with the HMI module yet.

After that is completed, the user can select one of the options on the lower right to play the full scenario as a driver would experience it. These options specify the “environment” or scenario (for both parallel and perpendicular parking) include standard parking, obstacle detection and avoidance, a mechanical fault in one of the subsystems, and a security event.

5.2. Sample Scenarios

The following figures are pulled from the prototype to demonstrate the system’s functionality in the provided scenarios.

5.2.1. Scenario 1: Standard Parallel Parking Simulation

This first set of images demonstrate the driver-view while using APA in a standard parallel parking situation. From the customer specifications, in this case, the speed does not exceed five miles per hour, and the parking space must be at least 1.2 times the vehicle length or 9 feet.

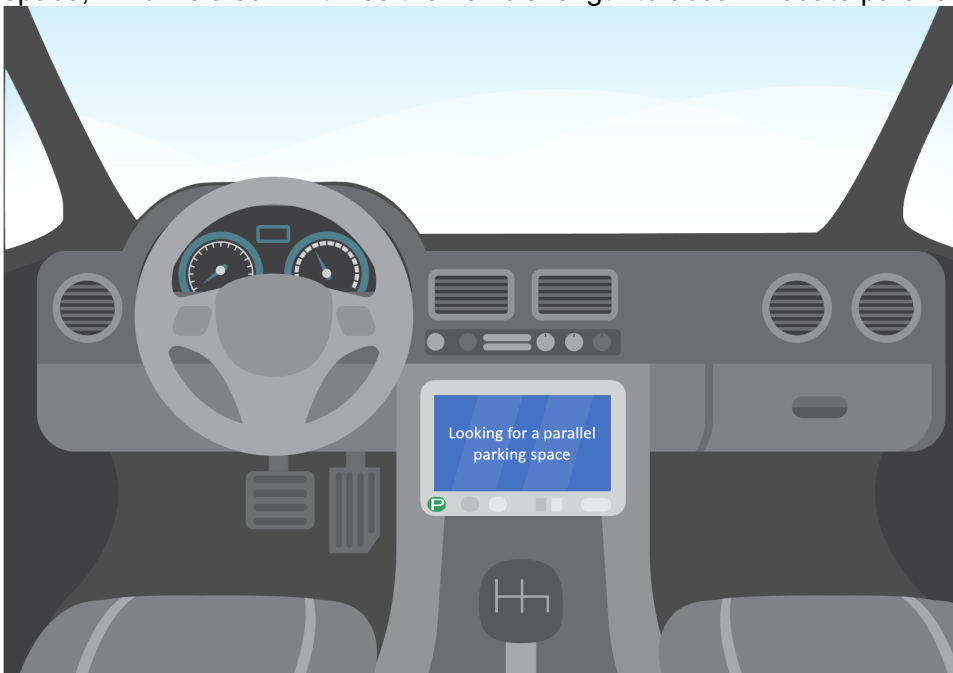
The figure below shows the intro screen to the APA system.



This figure is the screen for the user to select the type of parking desired.



The figure below indicates to the driver that APA is looking for the desired parking space, which is also 1.2 times the vehicle length to accommodate parallel parking.



This figure is the screen which prompts the driver for validation of the parking space and trajectory that were selected by the system as a security measure.



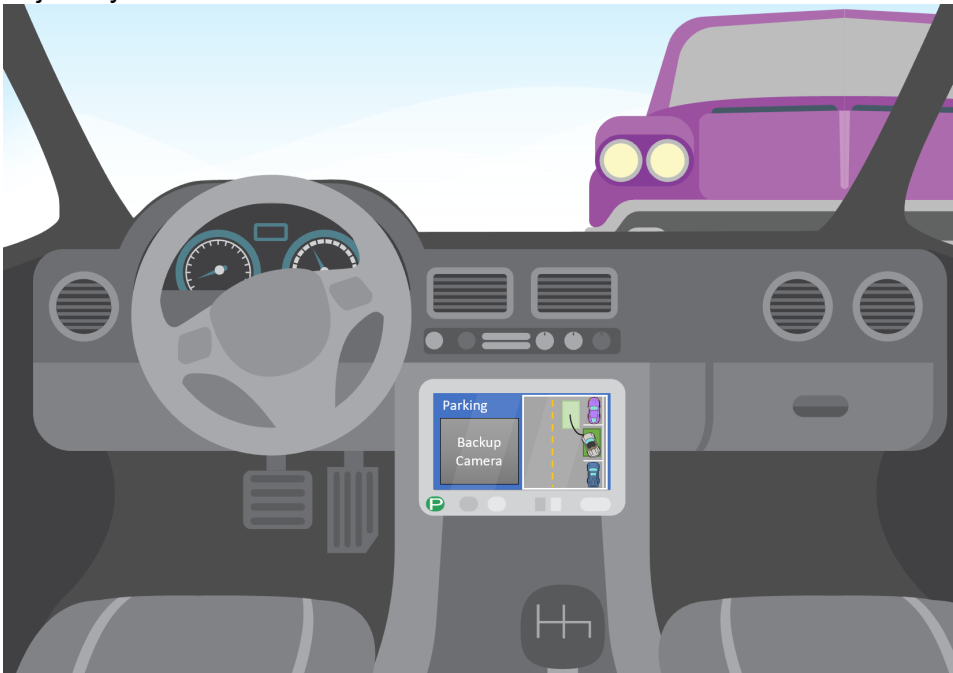
The figure below is one image of the parking simulation beginning, where the gear is shifted into drive, and the car accelerates, but does not exceed 5mph (as shown on the speedometer).



During the parallel parking maneuver, the backup camera will also be displayed, as well as the composite camera views creating the aerial picture on the right side of the HMI.



Another view of the parking maneuver in progress, this time with objects also in the surroundings. These are not seen as obstacles in this case, however, since the system knows these cars frame the desired parking space and do not cross the vehicle's trajectory.



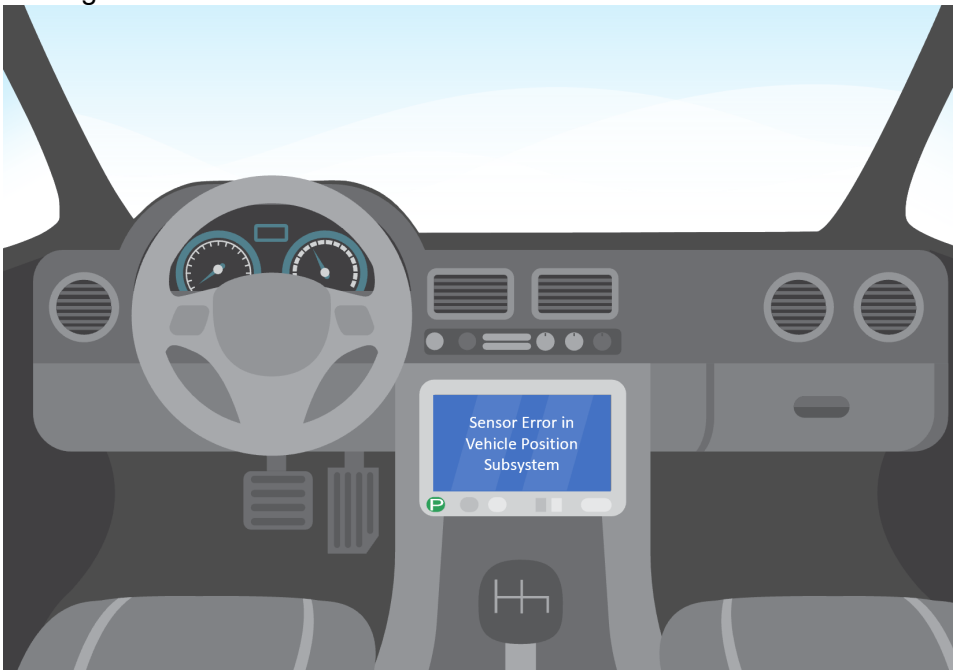
The figure below is the terminal screen, indicating the maneuver is over, the system is deactivated (as indicated by the APA button being in the off state), and that the driver has regained control of the vehicle.



5.2.2. Scenario 2: Subsystem Failure Simulation

These images demonstrate the behavior of the system in Scenario 2 proposed in the customer specifications, wherein there is a mechanical failure of one of the subsystems, specifically the sensors and cameras in VPS fail. In this case, the system is deactivated, and control is returned to the driver. The failure could occur at any time; however, the figures below show the case in which it occurs with the VPS while initially searching for a parking space.

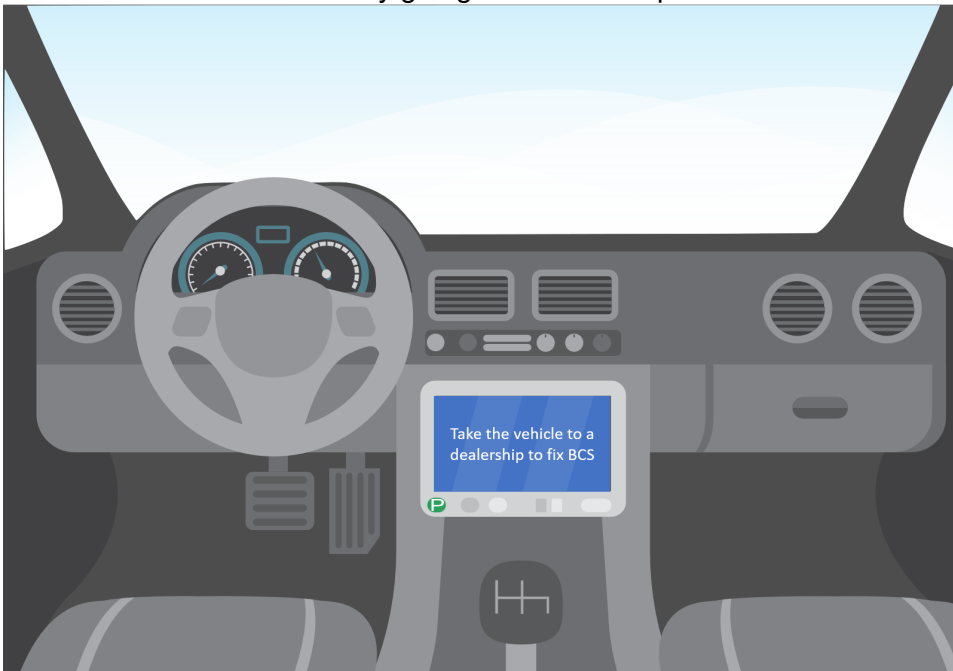
The system goes through the initial stages (figures 1-3 above) before encountering the VPS failure. The following images show the divergence in behavior from that point. This figure indicates to the driver the error has been encountered.



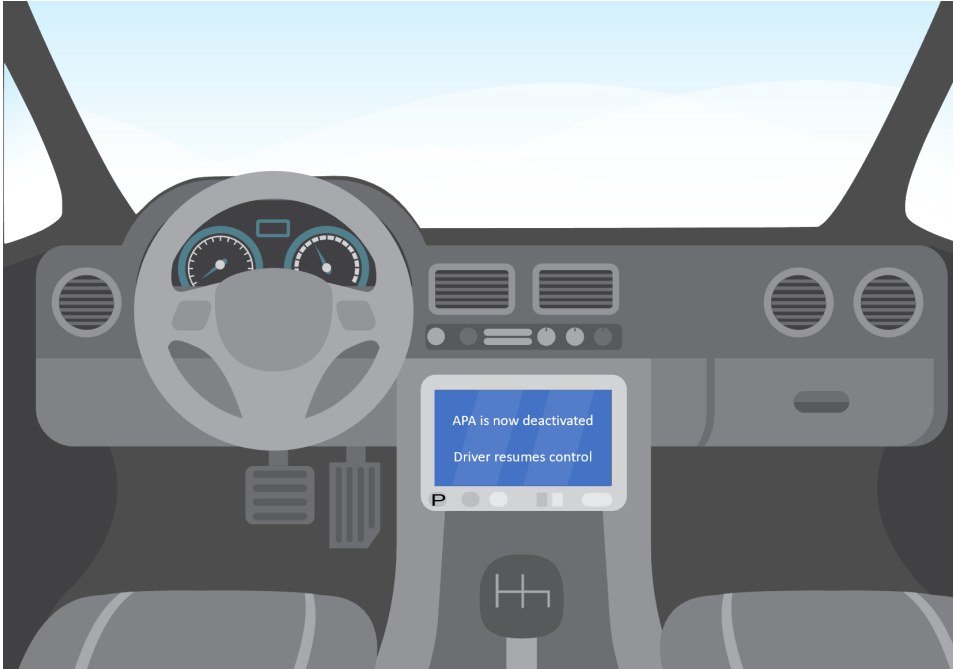
This figure explains in further detail for the driver what the failure caused behaviorally.



The figure below indicates that in the case of a malfunction, the HMI should tell the driver to fix the malfunction by going to a dealership.



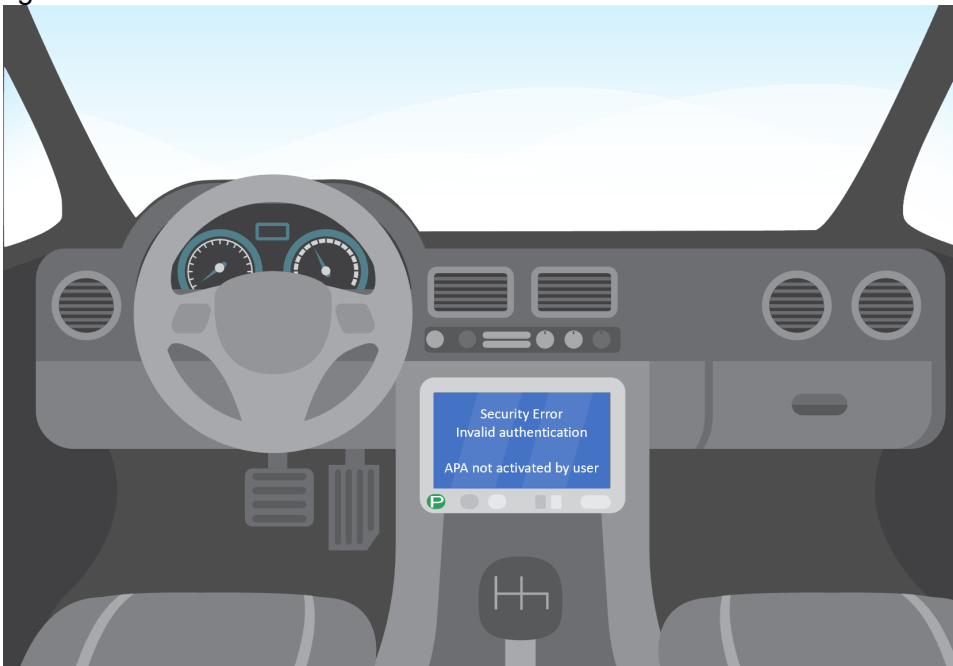
This figure indicates to the driver that the system could not recover, and rescinded control to the driver, who acknowledged this by pressing the APA button on the HMI, which is now off (no longer lighted green).



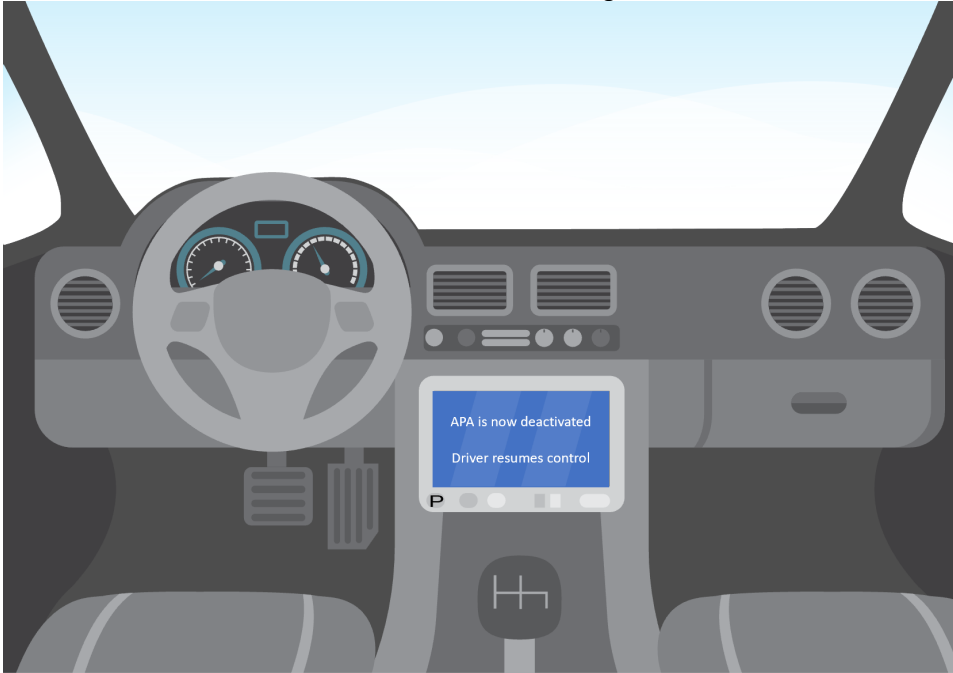
5.2.3. Scenario 3: Security Event Simulation

These images demonstrate the behavior of the system in Scenario 3 proposed in the customer specifications, wherein there is an attempted security breach by a malicious third-party. In this case, the third-party attempts to activate and execute an autonomous parking maneuver that was not initiated by the driver. This security event could occur at any time; however, this case shows the instance when an external party attempts to activate and control parking/driving.

The system detects that APA was not activated by the driver, but rather attempted to be activated by another external source, and the activation is therefore invalidated in the figure below.



Since the activation attempt failed, HMI informs the driver that the system is in fact off, and that the driver controls the vehicle in the figure below.



6. References

6.1. Background Information

- [1] Anon, (2019). [online] Available at: <https://www.mbusa.com/mercedes/technology/videos/detail/title-claclass/videoIdd165b63245537410VgnVCM100000ccec1e35RCRD> [Accessed 10 Nov. 2019].
- [2] CarMax, "Features You Auto Know: Automated Parking," *CarMax*, 27-Jul-2018. [Online]. Available: <https://www.carmax.com/articles/features-automated-parking>. [Accessed: 08-Nov-2019].
- [3] Cartelligent.com. (2019). *Should your new car have active park assist? | Cartelligent*. [online] Available at: <https://www.cartelligent.com/blog/should-your-new-car-have-active-park-assist> [Accessed 10 Nov. 2019].
- [4] E. Davidson. Active Park Assist Customer Specifications. (2019). Ford. Available at: <http://www.cse.msu.edu/~cse435/Projects/F2019/ProjectDescriptions/2019-APA-Ford-Davidson.pdf>
- [5] "Fords New Active Park Assist Aims To Curb Stress | Car ..." [Online]. Available: <https://www.auto123.com/en/news/fords-new-active-park-assist-aims-to-curb-stress/21325/>. [Accessed: 8-Nov-2019].
- [6] Owner.ford.com. (2019). [online] Available at: <https://owner.ford.com/support/how-tos/vehicle-features/convenience-and-comfort/active-park-assist.html> [Accessed 10 Nov. 2019].
- [7] S. Corby, "self-parking technologies in cars", WhichCar, 2015. [Online]. Available: <https://www.whichcar.com.au/car-advice/self-parking-technologies-in-cars>. [Accessed: 08- Nov- 2019].
- [8] S. Ma, H. Jiang, M. Han, J. Xie and C. Li, "Research on Automatic Parking Systems Based on Parking Scene Recognition," in *IEEE Access*, vol. 5, pp. 21901-21917, 2017.
doi: 10.1109/ACCESS.2017.2760201
- [9] S. Sedighi, D. Nguyen and K. Kuhnert, "A New Method of Clothoid-Based Path Planning Algorithm for Narrow Perpendicular Parking Spaces", Proceedings of the 5th International Conference on Mechatronics and Robotics Engineering, pp. 50-55, 2019. Available: <https://dl-acm-org.proxy2.cl.msu.edu/citation.cfm?id=3314512>. [Accessed 8 November 2019].

6.2. Project Site

- [10] Webpage for APA1 (2019), "Active Park Assist", Available online: <http://cse.msu.edu/~chenkuny/project/>, Michigan State University.

6.3. Prototype References

- [11] Dashboard Vectors. Vecteezy (2019). Available online at: vecteezy.com
- [12] Dynamic images with Image Map. Xul (2019). Available online at: <https://www.xul.fr/javascript/dynamic-images.php>

7. Point of Contact

For further information regarding this document and project, please contact **Prof. Betty H.C. Cheng** at Michigan State University (chengb at msu.edu). All materials in this document have been sanitized for proprietary data. The students and the instructor gratefully acknowledge the participation of our industrial collaborators.