Software Requirements Specification (SRS)

Active Parking Assist 1

Authors: Kevin Dittman, Matt Schleusener, Michael Williams, Daniel Thompson, Yi Shi

Customer: Eileen Davidson

Instructor: Betty Cheng

1 Introduction
Below is the Software Requirement Specification for the Active Park Assist (APA) system in Ford vehicles. The system requirements, functionality, and desired behavior will be described. The APA system can be activated by the driver through the Human Machine Interface in the vehicle when the vehicle’s speed is under 5 mph. The system will then scan its surroundings using close range radar sensors and display a series of parking spots for the driver to choose. Once one is selected, the system will take control of the vehicle and steer itself into the spot and shift into park. The system will also monitor for obstacles while undertaking this maneuver and brake if one is detected. The motivation for this system is to prevent injury to both passengers and outside pedestrians, as well as provide ease of use for the driver.

1.1 Purpose
This SRS document is intended to provide a detailed and accurate representation of an Active Parking Assist system. It will give descriptions of the system and its components, along with their interactions and reactions based on various external and internal conditions. This document is intended for the use of any interested stakeholder or developer, and will be used to assist in the creation of such a system.

1.2 Scope
This document describes the Active Park Assist system to be created. This system will make parking easier for drivers and reduce accidents.

The system will allow a vehicle to automatically park itself in either a parallel or perpendicular parking space. The system will identify parking spots, allow the user to accept or reject each spot as it is presented, and once a spot is selected, it will control the

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 chengb.cse.msu.edu)
gear, speed, and steering of the vehicle in order to maneuver it into the parking spot. It will utilize its sensors to ensure that the vehicle does not bump into any obstacles, including other parked vehicles. Parking will only initiate if the path is clear, but if an obstacle is detected in the way during the maneuver, the system will stop the vehicle and cancel the maneuver, returning control to the driver. It will not be able to determine if it is currently safe to begin parking, but instead will rely on the driver’s judgement.

1.3 Definitions, acronyms, and abbreviations
In the table below, various acronyms and definitions that are used throughout the document.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Parking Assist</td>
<td>The name of the system being designed, meant to park a car automatically within certain constraints.</td>
</tr>
<tr>
<td>(APA)</td>
<td></td>
</tr>
<tr>
<td>Controller Area Network</td>
<td>Serves as the main method of communication via electronic signals between the various systems and subsystems of a vehicle.</td>
</tr>
<tr>
<td>bus (CAN bus)</td>
<td></td>
</tr>
<tr>
<td>Driver</td>
<td>The operator of the vehicle, able to interact with the controls of the vehicle and the HMI.</td>
</tr>
<tr>
<td>Human Machine Interface</td>
<td>The touchscreen display set in the middle dash of the car which serves as the main point of interaction between the driver and the system for initiation.</td>
</tr>
<tr>
<td>(HMI)</td>
<td></td>
</tr>
<tr>
<td>Shift-by-wire transmission</td>
<td>A specific type of transmission where the gear is electronically controlled, either manually through an interface by the driver or automatically by another system.</td>
</tr>
<tr>
<td>Customer</td>
<td>Ford Motor Company.</td>
</tr>
<tr>
<td>MPH</td>
<td>Miles per hour.</td>
</tr>
<tr>
<td>Parallel Parking</td>
<td>Parking in a spot parallel to a line from the front to the rear of the vehicle.</td>
</tr>
<tr>
<td>Perpendicular Parking</td>
<td>Parking in a spot perpendicular to a line running from the front to the rear of the vehicle.</td>
</tr>
</tbody>
</table>
### Sensors
Ultrasonic sensors on all sides of the vehicle, and visual sensors (cameras) on the front and rear of the vehicle.

<table>
<thead>
<tr>
<th>Sensors</th>
<th>Ultrasonic sensors on all sides of the vehicle, and visual sensors (cameras) on the front and rear of the vehicle.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steering Control Subsystem</td>
<td>Subsystem of the vehicle which controls the steering of the front wheels.</td>
</tr>
<tr>
<td>User</td>
<td>See driver</td>
</tr>
<tr>
<td>Vehicle Position Subsystem</td>
<td>Subsystem of the vehicle which utilizes sensors to recognize objects surrounding the vehicle, available space between said objects, and threat of collision with them.</td>
</tr>
<tr>
<td>Brake Control Subsystem</td>
<td>Vehicle subsystem which controls the deceleration of the vehicle via the vehicle’s brakes.</td>
</tr>
</tbody>
</table>

### 1.4 Organization
The remainder of this document contains the following sections: section two is the overall description, which outlines the key functionality and characteristics of the system. Section three is the specific requirements as specified by Ford Motor Company. Section four is the modeling requirements, containing models to represent objects, behavior and states of the system. Section five is the prototype displaying the user interface. Section six is the references used in the creation of this document. Section 7 is the contact information for Dr. Cheng.

### 2 Overall Description
This section provides an overview of the APA system and clarifies how the system fits into any larger systems, namely the vehicle itself. It also provides a summary of the functions our system is expected to perform, and the required and expected characteristics of any user. Following that, any assumptions or constraints made about the environment, system, or user are detailed. Finally, any capabilities beyond the scope of the system will be listed.

### 2.1 Product Perspective
The system will exist in and function as a part of a vehicle. This vehicle will use “by-wire” technology for all control systems, allowing this system to directly issue commands to the control systems. These control systems are assumed to exist and be fully functional, specifically systems to control acceleration, braking, steering, and what gear the car is in. Additionally, input from the sensors will be available to the system.

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 chengb.cse.msu.edu)
Figure 2.1 is a system diagram that represents all the subsystems the APA must communicate with. A rectangle represents a subsystem while an arrow represents the direction of communication.

![System Diagram](image)

Figure 2.1: System Diagram

The driver will always initiate the system through the HMI. The system will ensure that it cannot be initiated when all of the conditions for starting are not met, namely the speed limit of 5 mph, and that the system has no serious or recurring failures in either the software or hardware. The vehicle and all of its subsystems - including the APA - communicate on a CAN bus, which uses electronic signals to send messages between such components.

### 2.2 Product Functions

The software of the APA will be able to identify parking spots and maneuver the vehicle into these parking spots as the user specifies. The software will be able to identify obstacles and abort the parking procedure if necessary. The software will be able to identify security threats and faults in the system or dependent systems and will deactivate the APA feature if necessary.

Figure 2.2 is a high-level goal diagram of the APA system. The top rectangle represents the overarching goal while the lower rectangles represent dependencies. The primary goal for the APA system is to safely park the vehicle. This depends upon the system being able to detect obstacles, detect faults in the system, and detect security threats. If the system operates as designed and is able to perform these three functionalities, the vehicle will park itself in a safe manner in the given time limit of 120 seconds.

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 chengb.cse.msu.edu)
2.3 User Characteristics
A user of the APA system is assumed to have the following characteristics: the user is to be physically and mentally capable of safely operating a vehicle, and is familiar with the laws pertaining to such operation. The user is also to have read the user’s manual and knows how to correctly operate the system. Finally, the user is to have a valid driver’s license.

2.4 Constraints
The APA system has constraints that can limit its functionality. For proper operation, the HMI software must able to communicate with the embedded systems of the subsystem involved. If the system is active for over 120 seconds, the system will brake the vehicle and abort, and the user will have to reinitiate the sequence. To help prevent malicious attacks or false information being provided to any subsystem, the system will use a key system in which one module will provide a specific identifier that another module must accept to communicate safely.

The system will be constrained to vehicles with shift-by-wire transmission, as it is a requirement of the system to shift gears for operation. As a safety constraint, if any hardware component of any of the subsystem fails, the user will not be able to activate the APA system. Another safety constraint is the system must avoid collision with obstacles that are detected through sensors during a parking maneuver.

2.5 Assumptions and Dependencies
The following assumptions were made during the design of the system: the driver is aware and capable of making intelligent and timely decisions, the system is capable of accurately and quickly detecting errors within the software and hardware, and the vehicle

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 chengb.cse.msu.edu)
subsystems used by the APA system are functional, and will correctly report their functionality or lack thereof, should that be necessary. As the system is comprised of a number of subsystems, it will be assumed that all of the subsystems performed their functionality as expected. Although this system will detect failure and subsequently prevent itself from operating, it is expected that hardware components of the subsystem have their own failsafe procedures in place as well.

2.6 Appportioning of Requirements

The following functionality is beyond the scope of this project, and therefore not implemented: the system autonomously exiting a parking spot and the handling of obstacles through means other than braking. For example, the system will not attempt to swerve to miss an obstacle.

3 Specific Requirements

Below are the functional, safety, and security requirements as specified by Ford Motor Company.

Functional Requirements:
1. Once a parking event is initiated, it shall complete within 120 seconds.
2. The system will ensure that any parking spots presented to the user are at least 1.2 times the car’s length for parallel and at least standard parking space width for perpendicular.

Safety Requirements:
3. System shall prevent the vehicle from hitting obstacles that move into its path during the parking maneuver.
4. A single point failure of any sensor input shall be detectable.
5. The vehicle must be going under 5 mph in order to activate the system.
6. If the customer brakes during the maneuver, the system will abort.
7. Any collision will cause the system to brake the vehicle to a stop and abort the system.
8. If any part of a control subsystem fails, the customer will not be able to activate the system.
9. The system must detect obstacles in the path of the vehicle.
10. If there is an object in its path, then the system will apply the brakes until the driver either aborts or choose to continue.

Security Requirements:

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 chengb.cse.msu.edu)
11. 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.
12. The system must be able to detect security faults in itself, including but not limited to miscommunications and repeated transmissions of messages.

4 Modeling Requirements

Included in this section are the following models and their use: a Use Case Diagram to represent observable behavior of the system, a Domain Model to depict all objects in the system, several State Diagrams to show the states of the system and how it transitions among these states, and a series of Sequence Diagrams to represent sample scenarios.

Figure 4.1 is a domain model for the APA system, with a data dictionary following. This provides an overview of the system structure. A box represents an object in the system. An arrow represents an association between two objects. The verbs within each box represent a function name.

Figure 4.1: Domain Model

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 chengb.cse.msu.edu)
The driver can interact with the HMI, which then interfaces with the controller to process the entered commands. The controller serves as an interface between the sensors, the control subsystems, and the HMI, processing the information from the sensors, issuing the necessary commands to the subsystems, and relaying the camera feed to the HMI.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera</td>
<td>Class to describe the class of Camera</td>
</tr>
</tbody>
</table>

**Attributes**

**Operations**

- GiveData (): void
  
  Sends data to the controller

**Relationships**

- Gives camera data to the controller. Inherits from the Sensors class

**UML Extensions**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APA Controller</td>
<td>Class to describe the controller of the APA system</td>
</tr>
</tbody>
</table>

**Attributes**

**Operations**

- DetectObstacle (): bool
  
  Detects if an obstacle is in the path of the parking

- IdentifySpot (): bool
  
  Tries to identify a parking spot

- GetCameraFeed (): void
  
  Gets the camera feed

- ParallelPark(): void
  
  Starts a parallel parking maneuver

- PerpendicularPark(): void
  
  Starts a perpendicular parking maneuver

- DisableSystem(): void
  
  Disables the system

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 chengb.cse.msu.edu)
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DetectFailure(): bool</td>
<td>Checks for any security or system failures</td>
</tr>
<tr>
<td>VerifyInput():void</td>
<td>Ensures the input from the HMI is valid</td>
</tr>
</tbody>
</table>

**Relationships**

Receives commands and sends data to the HMI. Also receives data from the sensors

**UML Extensions**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver</td>
<td>Class for the driver of the vehicle</td>
</tr>
</tbody>
</table>

**Attributes**

**Operations**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use(): void</td>
<td>Function for the driver to interact with the HMI</td>
</tr>
</tbody>
</table>

**Relationships**

Interacts with the HMI Class

**UML Extensions**

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMI</td>
<td>Class to describe the HMI within the vehicle</td>
</tr>
</tbody>
</table>

**Attributes**

**Operations**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VerifyInput (): bool</td>
<td>Verifies the input of the driver</td>
</tr>
<tr>
<td>DisplayCameraFeed (): void</td>
<td>Toggles the camera feed</td>
</tr>
<tr>
<td>StartParking(): void</td>
<td>Starts the parking process</td>
</tr>
<tr>
<td>CancelParking(): void</td>
<td>Cancels the parking process</td>
</tr>
<tr>
<td>DisplaySuccess():void</td>
<td>Notifies user of successful parking</td>
</tr>
</tbody>
</table>

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 chengb.cse.msu.edu)
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DisplayError():void</td>
<td>Notifies user of failure in system</td>
</tr>
<tr>
<td>SelectType(string type):void</td>
<td>Starts parking for the selected type (parallel vs. perpendicular)</td>
</tr>
</tbody>
</table>

Relationships: Gives commands and receives data from the controller

### UML Extensions

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obstacle</td>
<td>Class to describe the class of Obstacle</td>
</tr>
<tr>
<td>Attributes</td>
<td></td>
</tr>
<tr>
<td>Operations</td>
<td></td>
</tr>
<tr>
<td>Relationships</td>
<td>Ultrasonic uses this class to send data to the controller</td>
</tr>
<tr>
<td>UML Extensions</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parking Spot</td>
<td>Class to describe the class of Parking Spot</td>
</tr>
<tr>
<td>Attributes</td>
<td></td>
</tr>
<tr>
<td>Operations</td>
<td></td>
</tr>
<tr>
<td>Relationships</td>
<td>Ultrasonic uses this class to send data to the controller</td>
</tr>
<tr>
<td>UML Extensions</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle Position Subsystems</td>
<td>Class to describe the abstract class of sensor</td>
</tr>
</tbody>
</table>

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 chengb.cse.msu.edu)
<table>
<thead>
<tr>
<th>Attributes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Operations</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GiveData (): void</th>
<th>Sends data to the controller</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Relationships</th>
<th></th>
</tr>
</thead>
</table>

| Gives data to the controller. The ultrasonic and camera class inherit this class |  |

<table>
<thead>
<tr>
<th>UML Extensions</th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
</table>

| Ultrasonic | Class to describe the class of Ultrasonic |

<table>
<thead>
<tr>
<th>Attributes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Operations</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GiveData (): void</th>
<th>Sends data to the controller</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Relationships</th>
<th></th>
</tr>
</thead>
</table>

| Gives data to the controller. Inherits from the Sensors class. Uses the Obstacle and Parking Spot class to describe detection |  |

<table>
<thead>
<tr>
<th>UML Extensions</th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
</table>

| Brake Control Subsystems | Controls braking based on input from controller |

<table>
<thead>
<tr>
<th>Attributes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Operations</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Brake (): void</th>
<th>Brakes the car</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>StopBraking(): void</th>
<th>Stops braking the car</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Relationships</th>
<th></th>
</tr>
</thead>
</table>

| Receives input from APA Controller |  |

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 chengb.cse.msu.edu)
### Powertrain Management Subsystems

**Description**: Controls gear based on input from controller

<table>
<thead>
<tr>
<th>Attributes</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChangeGear(string gear): void</td>
</tr>
</tbody>
</table>

**Relationships**: Receives input from APA Controller

### Steering Control Subsystems

**Description**: Controls steering based on input from controller

<table>
<thead>
<tr>
<th>Attributes</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdjustSteering(): void</td>
</tr>
</tbody>
</table>

**Relationships**: Receives input from APA Controller

---

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 chengb.cse.msu.edu)
Figure 4.2 is a use case diagram with descriptions, showing the possible interactions for the system. The large rectangle represents the system boundary. The ovals within the system boundary represent use cases which are specific actions of the system. A stick figure represents an actor that interacts with the system. A line from an actor to a use case represents the actor is responsible for or can be responsible for that specific use case.
The driver can initiate parking through the HMI, picking either perpendicular or parallel. The HMI will verify that the input was valid, and begin the process of identifying spots. When a spot is found, the car will park. At any point, the driver or system may cancel the maneuver. The driver can either interact with the car through the brakes or through the HMI. The system will monitor surroundings with sensors, detecting any obstacles or collisions and brake or cancel accordingly. The system will also monitor itself for any points of failure, and if one is found deactivate itself.

**Figure 4.2: Use Case Model**

**Use Case: Start Parking Perpendicular**
*Actors:* Driver  
*Description:* Driver selects perpendicular parking, and the system begins to search for spots if the vehicle is not going above 5 miles per hour.  
*Type:* Primary  
*Includes:* Identify Spot  
*Extends:* N/A  
*Cross-refs:* 1, 5  
*Use cases:* Verify Input, Park

**Use Case: Start Parking Parallel**
*Actors:* Driver  
*Description:* Driver selects parallel parking, and the system begins to search for spots if the vehicle is not going above 5 miles per hour.  
*Type:* Primary  
*Includes:* Identify Spot  
*Extends:* N/A  
*Cross-refs:* 1, 5  
*Use cases:* Verify Input, Park

**Use Case: Identify Spot**
*Actors:* Sensors  
*Description:* Sensors scan the area around the car and relay potential parking spots to the user through the HMI.  
*Type:* Primary  
*Includes:* Verify Input  
*Extends:* N/A

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 chengb.cse.msu.edu)
Cross-refs: 2
Use cases: Verify Input, Park

Use Case: Verify Input
Actors: HMI
Description: The HMI verifies the user input to ensure no false starts.
Type: Primary
Includes: Parking
Extends: N/A
Cross-refs: 11
Use cases: Park

Use Case: Park
Actors: Steering and Acceleration
Description: The system controls the vehicle to park in the selected spot.
Type: Primary
Includes: N/A
Extends: N/A
Cross-refs: 1, 2
Use cases: Park

Use Case: Detect Failure
Actors: HMI
Description: The HMI will scan for failures in the sensors and send a failure signal to the system if one is detected. If a sensor failure is detected before the system is initiated, the active parking assist will not be able to be activated.
Type: Primary
Includes: Disable System
Extends: N/A
Cross-refs: 4, 8
Use cases: Disable System

Use Case: Detect Security Fault
Actors: HMI
Description: The HMI will scan for security faults and send a failure signal to the system if one is detected. If there is a single security threat, handle it and ignore. If they are repeated, abort and disable the system, notifying the user for service.
Type: Primary

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 chengb.cse.msu.edu)
Use Case: Detect Driver Interaction  
Actors: Braking System  
Description: If the driver interacts with the brakes, the system will receive a signal to abort. If the driver interacts with the accelerator, the vehicle will accelerate up to five miles per hour. 
Type: Primary  
Includes: Disable System  
Extends: N/A  
Cross-refs: 6  
Use cases: Disable System

Use Case: Detect Collision  
Actors: Driver  
Description: The sensors will continually scan to detect a collision with the vehicle. If one occurs the sensors will send an abort signal to the system. 
Type: Primary  
Includes: Disable System  
Extends: N/A  
Cross-refs: 7  
Use cases: Disable System

Use Case: Disable System  
Actors: HMI, Braking System, Sensors  
Description: If an abort signal is received from the HMI, Braking System, or Sensors the parking process will be aborted and the driver will be notified via a message on the HMI. 
Type: Primary  
Includes: N/A  
Extends: N/A  
Cross-refs: 4, 6, 7, 12  
Use cases: N/A

Use Case: Display Camera Image  
Actors: HMI  
Description: Receives environment data from cameras  
Type: Primary

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 chengb.cse.msu.edu)
Use Case: Cancel Parking
Actors: Driver, Obstacle, Sensors
Description: Cancel the parking process
Type: Primary
Includes: N/A
Extends: N/A
Cross-refs: 3, 6, 7
Use cases: Cancel Parking

Use Case: Detect Obstacle While Parking
Actors: Sensors
Description: Stops the vehicle when an obstacle has been detected while parking
Type: Primary
Includes: Brake
Extends: N/A
Cross-refs: 3, 9, 10
Use cases: Brake

Use Case: Brake
Actors: Sensors
Description: Brake the system
Type: Secondary
Includes: N/A
Extends: N/A
Cross-refs: 3, 10
Use cases: Brake
Figure 4.3 is a state diagram that shows how the system transitions from an idle state to its various other states while finding a parking spot and during the action of parking. The main aspect of the state diagram is that at any point if the action is cancelled, regardless of the reason, the vehicle will stop and return to idle. The black dot in the diagram represents the start of any driver interaction with the APA. The roundtangles represent states the system can be in, while the directional arrows show which states the current state can progress to. The text found in or above the arrows describes how the system transfers from one state to another.
Figure 4.3: Overall System State Diagram

Figure 4.4 is a state diagram for the HMI. At any time, if an error is detected the HMI will display an error message. The driver can also cancel parking at any state to return to an idle state. If no error is detected and the parking maneuver is successfully completed, a success message will be displayed. In the end, the HMI will always finish in an idle state.
Figure 4.5 is a state diagram for the APA Controller. The controller constantly detects for a failure and displays an according error if one is found. If no errors are found, the controller begins parallel or perpendicular parking. While parking, if an obstacle is detected, the controller reverts to an idle state.

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 chengb.cse.msu.edu)
Figure 4.5: APA Controller State Diagram

Figure 4.6 is a state diagram for the Brakes. The brakes start in an idle stage while the vehicle begins parking. When the vehicle needs to brake, the brakes are activated.
Figures 4.7-4.9 are three sequence diagrams that express the progression of certain scenarios the APA system may encounter. In the sequence diagrams, the boxes on the top represent objects of the system. The thin, vertical rectangles represent the time an object

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 chengb.cse.msu.edu)
Scenario One (Figure 4.7) demonstrates how the system will execute a simple parallel-parking maneuver. The driver initiates the APA and specifies parallel parking. The HMI verifies that the driver did in fact initiate the request, and displays the camera feed. Next the controller will identify a parking spot and begin the parking process. This starts with checking for obstacles then entering reverse, adjusting the steering accordingly, brake accordingly, and entering park once the vehicle is in the spot. Upon completion, a success message is displayed on the HMI.

![Diagram of a Successful Parallel Park](image)

**Figure 4.7: Sequence Diagram of a Successful Parallel Park**

Scenario Two (Figure 4.8) is how the system deals with a failure. The driver initiates the APA and specifies parallel parking. The HMI verifies that the driver did in fact initiate the request, and displays the camera feed. Next the controller will identify a parking spot and begin the parking process. This starts with checking for obstacles then entering reverse, adjusting the steering accordingly, brake accordingly, and entering park once the vehicle is in the spot. Upon completion, a success message is displayed on the HMI.

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 chengb.cse.msu.edu)
and begin the parking process. When the system detects a failure during the maneuver. An error message displays on the HMI. Then the system brakes the vehicle, adjusts the steering accordingly and disables the system.

![Sequence Diagram of a Parking Maneuver Failure Due to an Obstacle](image)

**Figure 4.8: Sequence Diagram of a Parking Maneuver Failure Due to an Obstacle**

Scenario Three (Figure 4.9) represents how the system handles malicious or erroneous attacks. The system verifies that the driver did in fact request a parking maneuver. If this

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 chengb.cse.msu.edu)
is true, parking continues as in scenario one. If this is false, the parking process is canceled.

A fault or a malicious attack tells the HMI to start parking. The HMI attempts to verify the input, and ignores it if it is invalid or passes it on if it is not. When the controller receives input, it also attempts to verify it, checking that it is from the HMI and not a malicious attack, and tells the HMI to cancel parking if it is not valid. The controller will be able to decode the signature of the key that is sent along with the message. In a similar vein, to protect against repeated retransmissions of the same message, there are time stamps that will allow the controller to distinguish between repeated and valid messages.

5 Prototype

The prototype will show a screen mockup of the vehicle HMI, the current state of the system, and provide possible interactions with the system, whether that is user interactions or outside events that affect the state of the system.

5.1 How to Run Prototype

The prototype is usable within a browser window. Using the AngularJS framework, it will give an accurate model of what the system will look like and how the system reacts to interactions. The prototype only requires you to use a browser with Javascript enabled.

http://www.cse.msu.edu/~cse435/Projects/F2016/Groups/APA1/web/prototypes.html
5.2 Sample Scenarios

Scenario One: Normal Use

Figure 5.1: Prototype Screenshot of Default State

The system starts in the default state. From here, the user selects “Start Parking”.

Figure 5.2: Prototype Screenshot of Parking Spot Search

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 chengb.cse.msu.edu)
In the search state, the system looks for a parking spot.

The system finds a parking spot and asks the user to select the spot, continue searching, or cancel the search.

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 chengb.cse.msu.edu)
The system then performs the parking maneuver. The user can cancel the maneuver by hitting the brakes or pushing the “Cancel Parking” button.

After parking successfully, the system enters park and displays a success message.

Scenario Two: Obstacle Detection
If the system detects an obstacle, the parking maneuver will be cancelled and the vehicle will be placed in park. A message describing what happened will appear on the HMI.

Scenario Three: Failure Detection

If a hardware or security failure happens, the system will cancel parking and put the vehicle in park. It will display a message and alert the driver to service the vehicle based on the severity of the failure.

6 References


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 cse.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.

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 chengb.cse.msu.edu)