Software Requirements Specification (SRS)

Active Park Assist 3

Authors: Peyton Ritchie, Igli Ndj, Cobian Gleason, Lazaro Cruz, Don Madhawa Jayaweera

Customer: Ms. Eileen Davidson (Ford Motor Company)

Instructor: Prof. Betty H.C. Cheng

1 Introduction

This Software Requirements Specification (SRS) document provides an outline for the Active Park Assist (APA) system requirements. The sections that are provided in this document include an overall project description, specific requirements of the system, models to explain the project functionality and flow, and a prototype with an explanation. Section 1 will describe the purpose and scope of the project, as well as the organization of the document. Section 2 will explain the product perspective and functions, as well as characteristics, constraints, assumptions, and dependencies. Section 3 will list the specific requirements of the project. Section 4 will contain and explain all of the modeling requirements for the system, including a use case diagram, class diagram, sequence diagrams, and a state diagram. Section 5 will explain the prototype and discuss some sample scenarios of the system.

1.1 Purpose

The purpose of the SRS document is to outline the requirements and functionality for the development of the APA system by Ford Motor Company. This document will further define the different aspects related to the system for the purpose of verifying them with the customer. The intended audience of the document are the current and future engineers and the sponsors involved in the building of the APA system.

1.2 Scope

The APA system is an embedded system that is used when the driver wants to park the vehicle. The system takes control of the vehicle and maneuvers the vehicle into the parking spot. The system will use the existing hardware
infrastructure on the vehicle to automate the parking process with minimal driver interaction. The application domain of this project is in the automotive domain and interacts with existing subsystems in the vehicle to execute the parking process.

Once the system is activated, the digital interface will prompt the driver to choose either parallel or perpendicular for the type of parking they would like. The system will proceed to find available parking spots of the chosen type and ask the user to select a spot. Then the system will take control and park the vehicle in the chosen spot according to a calculated trajectory. Prior to the confirmation, the parking spot is verified by the system and will detect any obstacles which will then trigger the system to issue warnings to the driver. If there is no verified parking spot, the system will send an alert to the driver.

The detection will range up to 6 feet and the system will not be able to park in diagonal parking spots. Warnings will be given to the driver throughout the parking process to alert them of any obstacles or system faults. The system will also be operable in various types of weather conditions such as rain, snow, fog, etc.

1.3 Definitions, acronyms, and abbreviations

Terms and Abbreviations

1. **Active Park Assist (APA)**: the system outlined in this document that is responsible for parking the vehicle
2. **Brake Control Subsystem (BCS)**: accepts inputs from the Park Control Subsystem to brake the vehicle in order to meet the required trajectory
3. **Human Machine Interface (HMI)**: accepts inputs from the driver, displays the camera information, and issues any warnings to the driver
4. **Park Control Subsystem (PCS)**: issues commands to the necessary subsystems based on the inputs received from the HMI subsystem and the trajectory calculated by the Vehicle Positioning System
5. **Powertrain Management Subsystem (PMS)**: accepts inputs from the Park Control Subsystem to accelerate the vehicle and select the gear position to meet the required trajectory
6. **Steering Control Subsystem (SCS)**: accepts inputs from the Park Control Subsystem to position the vehicle based on steering to meet the required trajectory
7. **Software Requirements Specification (SRS)**: this document that outlines the APA system requirements
8. **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 process.
1.4 Organization

The rest of this document will include an overall description of the system, which will explain the product perspective, product functions, user characteristics, system constraints, assumptions, and requirements. It will also include a list of specific system requirements, modeling requirements, a prototype description and sample scenarios.

The content of this SRS document is organized as follows:

1. Introduction
   1.1. Purpose
   1.2. Scope
   1.3. Definitions, Acronyms and Abbreviations
   1.4. Organization
2. Overall Description
   2.1. Product Perspective
   2.2. Product Functions
   2.3. User Characteristics
   2.4. Constraints
   2.5. Assumptions and Dependencies
   2.6. Apportioning of Requirements
3. Specific Requirements
4. Modeling Requirements
5. Prototype
   5.1. How to Run Prototype
   5.2. Sample Scenarios
6. References
7. Point of Contact

2 Overall Description

This section will detail the project development and the intended functionality of the system. The product perspective section will describe the context of the product and the constraints relating to the interfaces. Further examination of functionality, user characteristics and constraints of the system are present under this section. The section will end with an explanation of assumptions, dependencies and apportioning of the 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 msu.edu)
2.1 Product Perspective

Parking can be stressful for many drivers, especially parallel parking. There may be a risk of collision during parking the vehicle because the driver was not aware of all the obstacles surrounding the vehicle. To avoid such accidents and driver discomfort, the APA system will be built to automate the parking process.

The system will be built as an add-on to the current working system. The parking system will utilize three of the vehicle subsystems. The APA system will be incorporated into the HMI subsystem. During the parking process, the system will need to interact with the powertrain management, brake control, and steering control subsystems in order to properly position the vehicle.

For hardware constraints, radar sensors and cameras will be present to detect surrounding obstacles and position of the vehicle. These will interact with the VPS by feeding information back to it.

Throughout the parking maneuver, the camera feed should be displayed on the HMI as well. The camera feed will allow the user to be aware of the surroundings which will decrease the chance of a collision. If an error occurs, an appropriate warning will be displayed to the user through the HMI.

![Data Flow Diagram](image)

Figure 1: Data Flow Diagram

2.2 Product Functions

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)
This section describes the functionality of the APA system under the assumption of normal operation and a successful parking process.

The driver will interact with the HMI to enable the APA system and choose the type of parking operation. The system will send the verified parking spots for the driver to choose from. Once parking spot is selected the APA system will commence the parking procedure. The HMI then takes the user's input and sends it to the PCS. The PCS receives the vehicle position from the VPS and calculates a trajectory to the parking spot and sends inputs based on the calculated trajectory to the BCS, SCS, and PMS. Once the vehicle has been successfully parked, the HMI will notify the user and the system will disable itself.

This section’s description includes a high-level goal diagram depicting the goals of the system, shown below in Figure 2.

![Figure 2: High-Level Goal Diagram](image)

### 2.3 User Characteristics

A user of the APA system is expected to have a valid driver's license, which ensures that they are familiar with automotive vehicles and the parking process in general. The user is also expected to have some amount of technological experience to navigate through the HMI system via reading. To ensure that a user with minimal technical expertise will be able to use the APA system, the HMI will display simple instructions and actions for the user. The most important attribute of a user is their attentiveness. Since this will vary from user to user, the
system will provide camera feeds and notifications through the HMI to help users pay attention while the parking process is being executed.

### 2.4 Constraints

The APA system has a set range of detection it must operate in to prevent collision with obstacles. The system will disable if any object enters within 6 inches of the vehicle’s detection range. This is necessary in order to secure the safety-critical collision avoidance of the system.

If any of the radars or cameras become obstructed while a parking process is being executed, the system will disengage, relinquish control to the driver and alert the driver that the system is no longer parking the vehicle.

### 2.5 Assumptions and Dependencies

It will be assumed that the hardware will be researched and chosen by another team. For this iteration, we are not responsible for any of the requirements or specifications of the hardware, such as cameras and radar or how they are integrated into the vehicle.

The APA system relies on other systems within the vehicle in order to operate in any weather conditions. Systems such as traction control, AWD, lane keeping, ABS, etc, will provide a sufficient and consistent level of vehicle operation in any road conditions. Environment conditions are not taken into account for this reason.

It is assumed that the user is able to react appropriately to any warnings. In doing so, this will minimize the risk of collision and protect the driver and others as much as possible.

### 2.6 Apportioning of Requirements

Requirements for the system’s hardware will be set aside for future versions. Hardware utilities will have a constraint on their size. They must be small enough to cleanly incorporate into the existing vehicle frame without extending out. Any exposed hardware would allow for it to be easily damaged which would compromise the integrity of the parking system. Cameras should be equipped on the vehicle so that there is viewing angle on both sides and the back, as these are areas of the car that the driver cannot directly see.
3 Specific Requirements

1. Functions
   1.1. System shall identify available parking spots that correspond to the type of parking chosen through the HMI.
   1.2. Driver should be able to choose a spot from available spots shown through the HMI.
   1.3. Verify selected spot with the driver through HMI subsystem.
   1.4. The system will Accelerate/brake accordingly to maintain target trajectory into selected spot.
   1.5. The system will Steer accordingly into spot based on calculated trajectory.
   1.6. When the driver brakes, the system should suspend and relinquish control to the user.
   1.7. The system changes the vehicle state into park once the parking process has been completed.
   1.8. The system shall prevent the vehicle from hitting obstacles that are in its path during the parking maneuver by using the cameras and radars.
   1.9. The system will identify whether there are irregular curbs on the side and the system must respond to those accordingly.
   1.10. Vehicle should be able to steer, brake, and check position multiple times in order to get the vehicle in the parking spot.

2. Interface
   2.1. If the driver needs to initiate the automated parking procedure the driver needs to enable the system through the HMI subsystem. Otherwise the system is in an idle state.
   2.2. Allow driver to choose between parallel and perpendicular spot through the HMI subsystem.
   2.3. Each time the car performs a parking maneuver the HMI should display the process through the camera.
   2.4. When system has finished the parking maneuver, it will notify the driver through the HMI then relinquish control back to the driver by putting vehicle in park mode and disable the system.

3. Safety
   3.1. The vehicle will not move faster than 7 MPH while parking in progress for the safety of the driver and others.
   3.2. Once a parking event is initiated, it shall be completed in no more than 20 seconds.

4. Reliability
   4.1. System warns the driver if the vehicle is going to collide with an object or pedestrian through the HMI along with a warning sound.

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)
4.2. A single point failure of any sensor input shall be detectable and give a warning to the driver through the HMI, relinquish control and disable the system.

5. **Availability**
   5.1. The system must function in any weather conditions, where the camera/radar is not compromised.
   5.2. The system should function on any road conditions.

4 **Modeling Requirements**

**Use Case Diagram and Descriptions**

Figure 3 represents a use case diagram that displays the interactions of the driver with the APA system and parking spots. The driver can either enable or disable the system. If enabled the driver can choose a parking spot in the display provided by the HMI and begin the parking process. The driver still has the ability to disable the system whenever the driver wants. The system is then responsible for detecting obstacles and available parking spots surrounding the vehicle.

The surrounding data which is collected by the camera and the radar is sent to the VPS subsystem to identify valid parking spots. The information about the spots will be sent to the HMI subsystem for the driver to choose. The trajectory is then calculated for the chosen parking spot and the PCS subsystem will issue commands to other subsystems to steer the vehicle to the parking spot. The system has the ability to steer, brake and accelerate the vehicle when needed.
Use case: Enable System

Actors: Driver

Description: The driver can enable the Active Parking System to park the vehicle and turns on the rest of the Active Park Assist system.

Type: Primary and essential

Includes: Park Vehicle

Extends: None

Cross-refs: 1.1, 1.2, 1.6, 2.2

Use cases: Park Vehicle

Use case: Disable System

Actors: Driver

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)
### Description:
The driver can disable the Active Parking System through the HMI or by applying the brakes. When the system is disabled, the rest of the system turns off as well.

<table>
<thead>
<tr>
<th>Type:</th>
<th>Primary and essential</th>
</tr>
</thead>
<tbody>
<tr>
<td>Includes:</td>
<td>None</td>
</tr>
<tr>
<td>Extends:</td>
<td>None</td>
</tr>
<tr>
<td>Cross-refs:</td>
<td>1.6, 5.1, 5.2</td>
</tr>
<tr>
<td>Use cases:</td>
<td>None</td>
</tr>
</tbody>
</table>

### Use case: Choose Parking Spot

<table>
<thead>
<tr>
<th>Actors:</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>The VPS will verify parking spots and vehicle position. Then the HMI will display parking options to the driver. The driver will choose a parallel or perpendicular parking spot and then the system will park the car into the parking spot.</td>
</tr>
<tr>
<td>Type:</td>
<td>Primary and essential</td>
</tr>
<tr>
<td>Includes:</td>
<td>Park Vehicle</td>
</tr>
<tr>
<td>Extends:</td>
<td>None</td>
</tr>
<tr>
<td>Cross-refs:</td>
<td>1.2, 1.3</td>
</tr>
<tr>
<td>Use cases:</td>
<td>Park Vehicle</td>
</tr>
</tbody>
</table>

### Use case: Detect Surrounding

<table>
<thead>
<tr>
<th>Actors:</th>
<th>Pedestrian, Obstacles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>The camera and the radar will collect data of obstacles surrounding the vehicle and sends the camera visual data to the display of the HMI subsystem for the driver to select a valid parking spot, and send data collected from the camera and radar to the VPS to calculate the vehicle</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 msu.edu)
<table>
<thead>
<tr>
<th>Use case</th>
<th>Park Vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>System</td>
</tr>
<tr>
<td>Description</td>
<td>Once the driver selects the parking spot and the type of parking (parallel / perpendicular) via the HMI, the information will be sent to the PCS. PCS issue commands to the other subsystems (BCS, PMS, SCS) based on the trajectory calculated by VPS to park the vehicle while avoiding any potential risks involving collisions.</td>
</tr>
<tr>
<td>Type</td>
<td>Primary and essential</td>
</tr>
<tr>
<td>Includes</td>
<td>Steer Vehicle, Break Vehicle, Accelerate Vehicle</td>
</tr>
<tr>
<td>Extends</td>
<td>None</td>
</tr>
<tr>
<td>Cross-refs</td>
<td>1.1, 1.2, 1.3, 1.4, 1.5, 1.8, 1.9, 2.1, 2.2, 2.3, 2.4, 3.1, 3.2</td>
</tr>
<tr>
<td>Use cases</td>
<td>Choose Parking Spot, Enabled</td>
</tr>
</tbody>
</table>

<p>| Use case         | Steer Vehicle                                                                |
| Actors           | Driver                                                                       |
| Description      | The SCS accepts input from the PCS to steer the vehicle in order to maintain the required trajectory. |
| Type             | Primary and essential                                                         |</p>
<table>
<thead>
<tr>
<th>Use case:</th>
<th>Brake Vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors:</td>
<td>System</td>
</tr>
<tr>
<td>Description:</td>
<td>The BCS accepts input from the PCS to brake the vehicle in order to maintain the required trajectory.</td>
</tr>
<tr>
<td>Type:</td>
<td>Primary and essential</td>
</tr>
<tr>
<td>Includes:</td>
<td>None</td>
</tr>
<tr>
<td>Extends:</td>
<td>None</td>
</tr>
<tr>
<td>Cross-refs:</td>
<td>1.4, 1.6, 1.8, 1.9, 1.10, 5.1, 5.2</td>
</tr>
<tr>
<td>Use cases:</td>
<td>Park Vehicle</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use case:</th>
<th>Accelerate Vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors:</td>
<td>System</td>
</tr>
<tr>
<td>Description:</td>
<td>The PMS accepts input from the PCS to accelerate the vehicle in order to maintain the required trajectory.</td>
</tr>
<tr>
<td>Type:</td>
<td>Primary and essential</td>
</tr>
<tr>
<td>Includes:</td>
<td>None</td>
</tr>
<tr>
<td>Extends:</td>
<td>None</td>
</tr>
<tr>
<td>Cross-refs:</td>
<td>1.4, 1.8, 1.9, 5.1, 5.2</td>
</tr>
<tr>
<td>Use cases:</td>
<td>Park Vehicle</td>
</tr>
</tbody>
</table>
Class Diagram and Data Dictionary

Figure 4 below is a high level Class Diagram of the APA system. Below the diagram is a data dictionary which goes into further detail of each class in the diagram and how they interact with each other.

### Element Name

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

**APASystem**: Main system for Active Park Assist

**Attributes**:

- enabled: boolean

**Operations**:

None

---

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)
### BrakeControlSubsystem

**Element Name:** BrakeControlSubsystem  
**Description:** Accepts input from the ParkControlSubsystem to apply the brakes in order to meet the required trajectory

**Attributes:**
- None

**Operations:**
- **Brake(strength : float) : void**  
  Applies brakes according to Park Control Subsystem operation call

**Relationships**
- One to one relationship with ParkControlSubsystem

**UML extensions**
- None

---

### Camera

**Element Name:** Camera  
**Description:** Sends information of the area around the vehicle to the VehiclePositionSubsystem and HMI

**Attributes:**
- None

**Operations:**
- **sendData() : void**  
  Send the collected data to the Vehicle Positioning Subsystem
- **turnOn(): void**  
  Turn camera on
- **turnOff(): void**  
  Turn camera off

**Relationships**
- The VehiclePositionSubsystem depends on it
- Interacts with HMI

**UML extensions**
- None

---

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)
### Driver

Person driving the vehicle and controlling the system

#### Attributes:
None

#### Operations:
- **pressBrake()**: void
  If the system is enabled, this operation will disable the system
- **pressDisableButton()**: void
  If the system is enabled, this operation will disable through the system (from the HMI)
- **turnWheel()**: void
  If the system is enabled and a parking operation is in progress, this will disable the system
- **pressGas()**: void
  If the system is enabled and a parking operation is in progress, this will disable the system

#### Relationships
One to one relationship with HMI

#### UML extensions
None

### Element Name

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMI</td>
<td>Accepts driver input, displays camera information and handles warnings</td>
</tr>
</tbody>
</table>

#### Attributes:

- **spotChosen**: The physical spot chosen by the system

#### Operations:
- **enableSystem()**: void
  Activates the system
- **disableSystem()**: void
  Deactivates the system (by either pressing the brake, taking over steering, or though the HMI)
- **chooseParkType()**: void
  Driver chooses between parallel or perpendicular parking
- **chooseSpot()**: void
  Driver chooses a spot from the available spots identified by the system of the type they chose
- **displayMessage()**: void
  Displays warning or current action sent from the Park Control Subsystem

---

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)
<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obstacle</td>
<td>Any obstacle the vehicle detects during the parking process</td>
</tr>
<tr>
<td>Attributes:</td>
<td></td>
</tr>
<tr>
<td>- position: Position</td>
<td>The position of the obstacle</td>
</tr>
<tr>
<td>Operations:</td>
<td>None</td>
</tr>
<tr>
<td>Relationships</td>
<td>Many to one relationship with VehiclePositionSubsystem</td>
</tr>
<tr>
<td>UML extensions</td>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParkControlSubsystem</td>
<td>Accepts the driver input from the HMI subsystem, calculates the vehicle trajectory based on information from the Vehicle Position Subsystem and calls operations of the other subsystems</td>
</tr>
<tr>
<td>Attributes:</td>
<td></td>
</tr>
<tr>
<td>- systemActive : boolean</td>
<td>Checks whether the Park Control Subsystem is in an active state or not</td>
</tr>
<tr>
<td>Operations:</td>
<td></td>
</tr>
<tr>
<td>- calculateTrajectory(): void</td>
<td>Based on the information received from the other subsystems, this operation will calculate the path the</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 msu.edu)
<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParkingSpot</td>
<td>A parking spot that is available for the system to park the vehicle in</td>
</tr>
</tbody>
</table>

**Attributes:**

- width: float
- length: float
- depth: float

**Operations:**

- Park(): void

**Relationships**

One to one relationship with HMI, One to one relationship with ParkControlSubsystem

**UML extensions**

None

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParallelSpot</td>
<td>A ParkingSpot that is a parallel spot</td>
</tr>
</tbody>
</table>

**Attributes:**
<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PerpendicularSpot</td>
<td>A ParkingSpot that is a perpendicular spot</td>
</tr>
<tr>
<td>Attributes:</td>
<td></td>
</tr>
<tr>
<td>- width: float</td>
<td>Width of the parking spot</td>
</tr>
<tr>
<td>- length: float</td>
<td>Length of the parking spot</td>
</tr>
<tr>
<td>- depth: float</td>
<td>Depth of the parking spot</td>
</tr>
<tr>
<td>Operations:</td>
<td></td>
</tr>
<tr>
<td>- Park() : void</td>
<td>Parking operation for a perpendicular spot</td>
</tr>
<tr>
<td>Relationships</td>
<td>Inheritance from parking spot.</td>
</tr>
<tr>
<td>UML extensions</td>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PowertrainManagementSubSystem</td>
<td>Accepts input from the Park Control Subsystem to accelerate the vehicle and select the gear lever position in order to meet the required trajectory.</td>
</tr>
<tr>
<td>Attributes:</td>
<td>None</td>
</tr>
<tr>
<td>Operations:</td>
<td></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 msu.edu)
<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radar</td>
<td>Gathers information from the vehicle’s surroundings and sends it to the VehiclePositionSubsystem</td>
</tr>
<tr>
<td>Attributes:</td>
<td>None</td>
</tr>
<tr>
<td>Operations:</td>
<td></td>
</tr>
<tr>
<td>- sendData() : data Data</td>
<td>Send the data collected to the Vehicle Positioning Subsystem</td>
</tr>
<tr>
<td>- turnOn(): void</td>
<td>Turn radar on</td>
</tr>
<tr>
<td>- turnOff(): void</td>
<td>Turn radar off</td>
</tr>
<tr>
<td>Relationships</td>
<td>VehiclePositionSubsystem depends on it</td>
</tr>
<tr>
<td>UML extensions</td>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SteeringControlSubsystem</td>
<td>Accept input from the Park Control Subsystem to steer the vehicle according to the calculated trajectory</td>
</tr>
<tr>
<td>Attributes:</td>
<td>None</td>
</tr>
<tr>
<td>Operations:</td>
<td></td>
</tr>
<tr>
<td>- Steer(angle : float) : void</td>
<td>Steer the vehicle according to Park Control Subsystem information</td>
</tr>
</tbody>
</table>
Element Name | Description
---|---
VehiclePositionSubsystem | 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:**
None

**Operations:**
- IdentifyAvailableSpots() : spots : ParkingSpot[ ] Identifies available parking spots for the driver to choose from
- IdentifyObstacles() : void Identifies obstacles surrounding the vehicle.
- VerifyPosition() : void Gets called continuously as the vehicle is in the parking process to ensure the position is correct

**Relationships**
Depends on Camera and Radar, one to one relationship with ParkControlSystem and Obstacle

**UML extensions**
None

**Representative Scenarios of System**

Figure 5 is a sequence diagram that displays a process of the APA system if the parking maneuver was successful. On the other hand, Figure 6 is another sequence diagram that displays the APA system process in a situation where the vehicle detects an obstacle during parking.

---

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)
Figure 5: Sequence Diagram of successful park

When the driver turns the system on, the HIL will prompt the driver to select a parking type (parallel or perpendicular).

HIL enables the control system, which collects data and identifies valid spots.

Valid spots are displayed for the user to select. Parking will initiate once the user selects one.

Parking control system will park the vehicle, and the HIL will notify the user when the process has completed.

Figure 6: Sequence diagram of parking with an obstacle interruption

When the driver turns the system on, the HIL will prompt the driver to select a parking type (parallel or perpendicular).

HIL enables the control system, which collects data and identifies valid spots.

Valid spots are displayed for the user to select. Parking will initiate once the user selects one.

Parking process will begin. The parking control subsystem will pull the positioning subsystem for updates on vehicle position and surrounding obstacles.

If an obstacle is detected to be 6 inches or closer, a message is sent to disable the system. The user is notified and the parking maneuver will end without being completed.

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)
5 Prototype

The prototype will show the functionality of the HMI system and how the user interacts with the HMI. Specifically, it will show how the user navigates the HMI to have the vehicle park into the chosen parking spot of a chosen. It will also show how the APA system will handle different scenarios during the parking process.

5.1 How to Run Prototype

The prototype will be provided via the web that is accessible on any device with no networking constraints. The prototype V1 will provide a number of interfaces with descriptive captions/annotations showing what the user will experience while operating the system.

Prototype V2 will be more interactive than V1. It will allow the user to select settings for the scenario, such as the type of parking spot and if the parking maneuver executes as normal or not. They can select issues to occur during the
process such as an obstacle, hardware fault, or security breach of the system. They can then click “Run Simulation” and the prototype will display a series of images of what the HMI will display for the scenario. This is what the driver will see when the APA system is running and it will show the types of options they have while interacting with the HMI.

Prototype: http://cse.msu.edu/~gleaso95/prototype.html

5.2 Sample Scenarios

The scenario below is of a user interacting with the APA system. The user will be able to turn on the system through the HMI and select a parking spot type, either parallel or perpendicular (Figure 8). In figure 9, the user has selected to park the vehicle in a perpendicular spot and the user will be shown the valid parking spots they are able to select.

Once the parking spot has been chosen, the APA system will proceed to park the vehicle into the selected parking space. The HMI will provide the user a visual display of the process (Figure 10).

If the system detects an unexpected obstacle, the system will display a warning to the user and disable the system (Figure 11).

![Figure 8: Driver turns on the system and selects a parking spot type](image-url)
Figure 9: HMI displays valid spots for the driver to choose from

Figure 10: HMI shows surroundings as parking process occurs
Figure 11: If vehicle detects an obstacle, it will stop and system will be disabled

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 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.