Design Document

ARC - Autonomous RC

Tao Chen, Cierra Shawe, Daniel Stoyer

June 5, 2017

Contents

1 INTRODUCTION

Research into consumer/hobbyist, high performance RC vehicles was requested by Oregon State University via Mr. Kevin McGrath. This project was requested to determine if it is possible to apply high-speed performance during autonomous navigation and obstacle avoidance to a modified RC car at a cost less than four thousand dollars (USD). Autonomous RC (ARC) sought to push the boundaries of what is possible for autonomous RC vehicles. Our research shows that components are decreasing in cost and increasing in performance. The cost-barrier to autonomous research is decreasing dramatically. Our documentation and parts list provides would-be researchers a launching point to continue the work we started in ARC. Our client was the same person who requested the project, Mr. Kevin McGrath. Mr. McGrath is an instructor at Oregon State University. [Who are the members of our team?]The ARC team members are Tao Chen, Cierra Shawe, and Daniel Stoyer. [What were their roles?]Tao was our software and robotics expert, he worked extensively with our software package and got our car working in simulation, he was responsible for the areas of Motion Model, Path Planning, and Autonomous Algorithms (e.g. obstacle avoidance, parallel parking, etc.). Cierra was our electronics and hardware expert, she designed all the mounting hardware used to anchor the sensors to the RC car and did all the wiring/soldering, she was responsible for the ares of Vision Systems, Sensors, and Hardware. Dan was team leader, responsible for making sure the team was on track to hit milestones and Capstone deadlines on time. He was also responsible for overseeing the areas of Image Analysis, User Interfaces, and Radio Communication. [What was the role of the client? (i.e. supervision only, participate in development, etc.)]

2 ORIGINAL REQUIREMENTS DOCUMENT

Software Requirements Specification

ARC - Autonomous RC

Senior Capstone Project

Oregon State University

Fall 2016

Tao Chen, Cierra Shawe, Daniel Stoyer

Version 1.0

December 1, 2016

D. Kevin McGrath | Date

Tao Chen | Date

Cierra Shawe | Date

Daniel Stoyer | Date

Contents

1 INTRODUCTION

This document aims to provide an overview and list of requirements for group 44, Autonomous RC (ARC). This section provides a purpose and scope description, list of abbreviations and acronyms, and an overview of everything included in this SRS document. As this is a research project, some of the sections within the document may not apply to the project.

1.1 Purpose

A detailed description of the requirements for the ”Autonomous RC System” or ARCS will be provided within this document. System constraints, interface decisions, and interactions with other external applications and hardware will also be explained. This document is primarily intended to be a customer proposal for approval and a development team reference for the first version of the system.

1.2 Scope

ARCS is a software-hardware interface designed to retrofit RC cars for autonomous operation, using commodity hardware. The software and hardware specifications should be available free to download and modify at the users will.

Users should be able to purchase and install the specified hardware from major online retailes such as Amazon.com or hobby sites such as Sparkfun.com. Once the vehicle is assembled, it should be able to autonomously navigate to a given destination using GPS, or a within pre-defined boundaries, such as a room. Control software will need to be loaded onto the main processing unit and any other control hardware needed. Other software required will be a control panel to monitor the vehicle and determine it’s behavior, such as navigating to a point, or on a track.

Hardware that we expect to need:

- Base station that includes a transceiver
- An RC car with servo steering and DC motors
- Main processing unit (a computer)
- Transceiver to send and receive information on the car
- Controller to send signals to sensors and actuation devices (motors, servos, etc...)
- Vision system to aide in obstacle avoidance
- GPS unit to aide in navigation

ARCS will be expected to be able to receive input from a user base station, and react within the environment based on a destination or path that the system receives. It should also be able to navigate to the destination without user intervention, as fast as possible.

1.3 Definitions, Acronyms, and Abbreviations

1) **IMUs:** Inertial measurement unit. Used to measure acceleration, angular acceleration, and orientation of the vehicle.

2) **Operator/User:** The person who is giving commands such as destination to the system.
3) **Protocol:** Defines the data format to be transferred.
4) **Telemetry Data:** Data that contains the status information of the vehicle, such as speed, temperature, location, battery, etc.
5) **Emergency:** Emergencies include, but are not limited to:
   - The vehicle drifts off course significantly.
   - Vehicle flips upside down.
   - On-board components fall off.
6) **Visual Unit:** Visual components that provide image streams to the main processing unit. The primary computer will extract information from the images, such as road condition, obstacles, and depth.
7) **Actuators:** Motors and servos.
8) **Initialization:** From the system perspective, the initialization process will include self-checking and sensor calibration. From the user's perspective, it contains making sure all the hardware is attached, battery is at least 80% full, and giving the system a destination.
9) **Success:** The vehicle successfully navigates itself to the destination.
10) **ARC:** Autonomous Remote Controlled is our team name
11) **ARCS:** ARC System, which refers to both hardware and software components built for the project.
12) **RC:** Remote Controlled
13) **AV:** Autonomous Vehicle
14) **Main Processing Unit:** The unit which handles all high level decisions and input processing.
15) **Stakeholder:** Any user that will operate ARCS.
16) **Companion Computer:** The computer on board the RC vehicle that performs heavy computation, such as image analysis.
17) **Ground Station:** The remote computer that sends user commands to the vehicle and receives telemetry from the vehicle.

1.4 Overview

The remainder of this document includes three sections and the appendixes.

Section two provides an overview of system functions and intersystem interaction. System constraints and assumptions are also addressed.

Section three provides the requirement specification in detailed terms and system interface descriptions. Different specification techniques are used in order to specify requirements for different audiences.

Section four priorities requirements and includes motivation for the chose prioritization and discusses why other methods were not chosen.

Appendixes at the end of the document include results of the requirement prioritization and a release plan based on the requirements. [**?**]

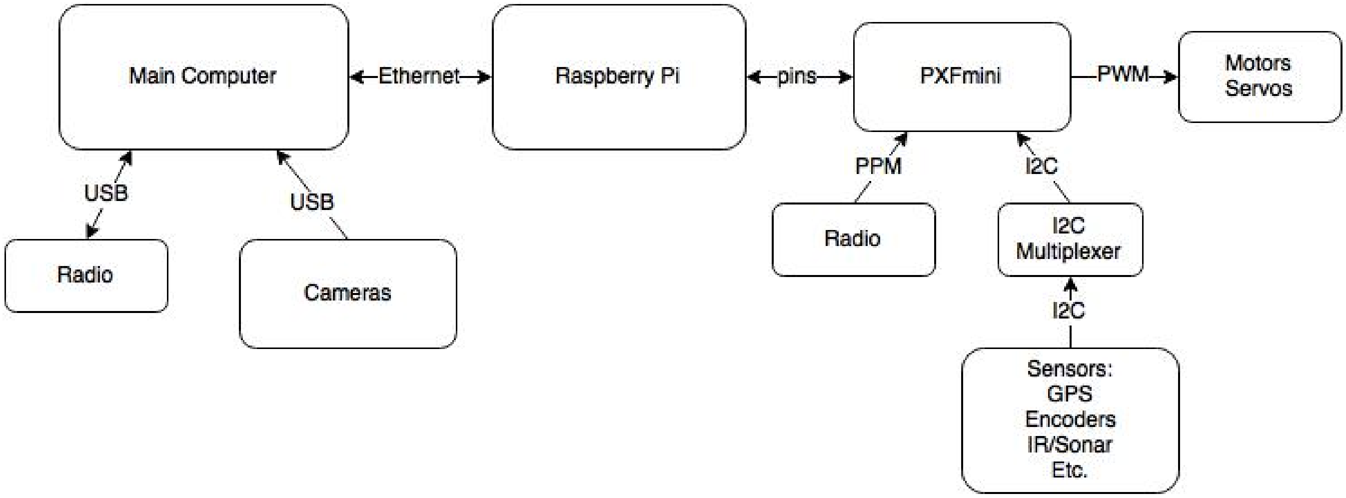


Fig. 1. Block diagram of hardware flow

2 Overall Description

This section will explain the system in its context to create a better understanding of how the system interacts with other systems. In addition, basic functionality will be introduced and addressed. Stakeholders will be defined and functionality for each type of stakeholder will be adressed. Lastly, constraints and assumptions for the system will be presented.

2.1 Product Perspective

ARCS will be designed to integrate into an RC car using commodity hardware, and open to anyone who is interested in using it. This makes ARCS a component of a larger system, namely the RC car.

ARCS will consist of three parts:

1) Base-station used for user interaction
2) Hardware attached to RC car to be able to connect to base-station and operate the vehicle
3) Software implementation for control and communication between hardware and software

In order for the user to interact with the vehicle, commands will be sent via some form of receiver to the car. This will need to be done via a base-station that has software able of providing a way to communicate with the receiver, which then transmits data to the receiver on the car, which is then handled by the on-board computer. The control flow is described in figure 1.

2.1.1 System interfaces

There are a total of 5 system interfaces where the system can communicate with the outside world.

1) Sensors: Sensors will have a two-way communication with a secondary computer unit, where filtering and smoothing will happen before reliable data will be passed to the main processing unit. The programs that reside in the secondary unit will utilize various methods to generate reliable results based on the raw data. At start-up, there will be a script executed by the system to correctly configure and calibrate each sensor. Sensors may include: battery monitor, temperature sensor, GPS, encoders, IMUs, and any other sensors that are needed for autonomous operation.
2) Radio: This is the portal of the system where operator/user can monitor the status of the vehicle. Different protocols will be implemented for telemetry data, which will be displayed to the operator/user. This portal also allows operator/user to take control over the computer in case of emergencies.
3) Visual unit: This is the interface where the visual unit can pass streams of images to the system.
4) Actuator: The system issues commands to the motors and servos via this interface.
5) User interface: This interface is a different interface than the radio interface even though they both allow humans to interact with the system. The user interface will be disable after the vehicle starts maneuvering. This interface allows user to input operation modes and desired destinations into the system.

2.1.2 User interfaces

The user interface will be a graphical user interface (GUI) that can be interacted with by a user. Anyone who knows how to operate a mouse or touch screen will be al. A map makes it easier for users to pin point destinations and view the current location of the vehicle. An error messages will be generated the vehicle and the station exceeds the maximum radio range, or the vehicle is low on battery.

With proper training (in less than 30 minutes), one can understand all the indicators of the system to know the status of the vehicle.

The GUI is a single window/page arrangement. A large portion of the window is dedicated to the map. A section of the GUI will be used to display telemetry data. Error messages will cover at least 10% of the screen in order to alert the user of an issue.

2.1.3 Hardware Interfaces

Two main hardware interfaces currently exist within ARCS.

1) Vision processing
2) Telemetry collection and actuator control

Vision processing, from either a stereoscopic or depth camera, will be interfaced through an i5 or better processor from an on board computer using a control board, which will be used as an interface between a computer that is doing high level computations, the servos, motors, and other telemetry devices.

Telemetry collection and actuator control will be handled by a PXFmini and interfaced through a board with a 40pin connector, such as the Raspberry Pi. This abstracts a lot of control onto the PXFmini.

2.1.4 Software Interfaces

- We require three operating systems. One for a remote PC for user input, one for the primary computer for on-car data analysis, and one for the secondary computer for on-car control.
- We require a user interface to be able to give the car destination commands.
- We require software that takes input from hardware, such as video cameras, and analyze the data
- Sensor analysis software needs to be able to read data from sensors, such IMUs, and generate usable information to pass on to
- Path finding software needs to be able to integrate and analyze data from mapping and localization software products and determine a path to follow in real time.
- The primary and secondary computers will need to talk with one another. The primary computer will need to receive data from the secondary computer, analyze the data and send corresponding commands back to the secondary computer. The secondary computer will need to receive commands from the primary computer, and send data to the primary computer.
- A software interface will required on the secondary computer to convert commands received from the primary computer to instructions usable by pre-existing RC car on-board controller.
- We will require separate interfaces between the secondary computer and the visual/spacial sensors, GPS, speed sensor, and the IMU.

2.1.5 Communications Interfaces

- Radio communication will required to be able to send commands to the car and to be able to receive feedback from the car. The radio frequency will be in 27 MHz or 49 MHz range. We need software that allows us to send and receive radio signals to and from the car.
- LAN communication will be used to transmit data between the on-car computers. Existing network protocols will be utilized to perform the transmissions.
- A software interface will be required to send the commands from the secondary computer to the pre-existing RC car on-board controller.

2.1.6 Memory constraints

Since we are unsure of what hardware will be used, we are not able to set any constraints on the memory requirements that the system has to meet.

2.1.7 Operations

In most cases, operations of the system will be isolated from the user/operator, excluding emergencies and initialization.

1) At initailization, user/operator has to thoroughly check the body of the vehicle, making usre that all components are firmly attached as well as the bettary level is at least 80%;
2) Destination will be given by the user at the station during initialization;
3) User is allowed to take over control from the system in case of emergencies via the remote control;

4) User/operator is resposible for monitoring the status of the system, such as battery level, speed, location, distance, radio signal strength, pre-planned path, etc.

2.1.8 Site Adaptation Requirements

For research and prototype purposes, no site adaptation requirements are relevant.

2.2 Product Functions

This system utilizes resources from more than one computer and controls multiple actuators. The system is designed to minimize user interference during operations, however, monitoring is required. Maintainability of the system is not considered to be user-oriented, which means a user/operator should not be worried about maintaining the system. The system will be made available publicly (open source), however, only users with extensive robotics experience should attempt at modifying the system outside of the given parameters.

2.3 User Characteristics

Our system should be able to be built and operated by a user with at least two years of university engineering coursework and one two years, a user with over five years of technical experience through work in either an electrical or software industry, and a RC hobbyist with over five years of experience.

2.4 Constraints

As this is a research project, constraints may rapidly change or new constraints may be retroactively added to better address problems found or the needs of the system.

Current constraints for the project include:

1) Real time image processing will limit the vehicle's ability to maneuver through obstacles.
2) Telemetry data displayed to the user will be slightly delayed due to long distance. No solutions to this issue are currently available.
3) The vehicle's natural structure will inevitably cause more uncertainty on predictions, which requires the use of more complicated algorithms versus what is required for ideal conditions.
4) Hardware may fall off during operation which will force the system to halt. All components must be securely fastened and adequately protected in the case of an emergency.
5) Telemetry data transfer are limited to radio during outdoor operations.
6) The success rate are set to 80% and above.
7) For research purpose only, the vehicle will only be required to navigate around static objects in an environment. This is to reduce the complexity of the initial system.

2.5 Assumptions and Dependencies

As a research project, the main assumption is currently this is possible.

2.6 Apportionment of Requirements

See next page.

Fig. 2. ARC Fall Schedule Gantt Chart

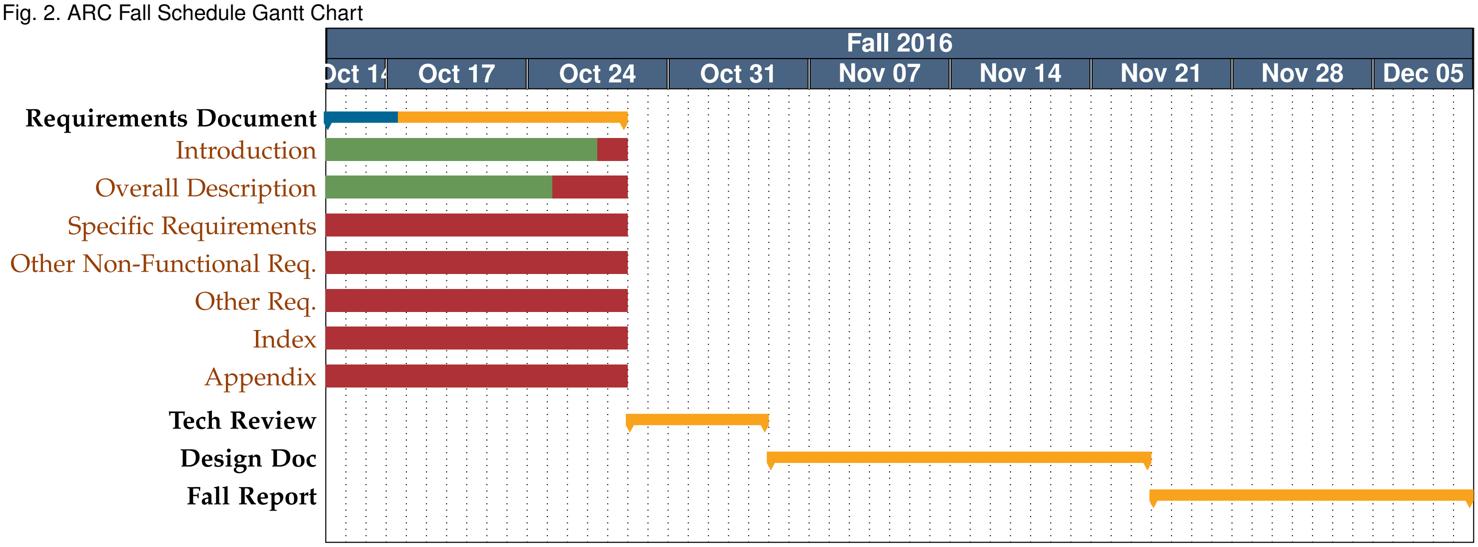


Fig. 3. ARC Winter Schedule Gantt Chart

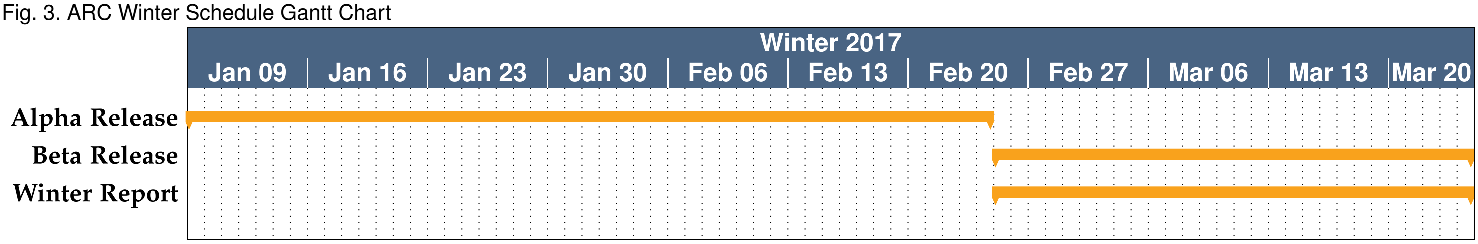
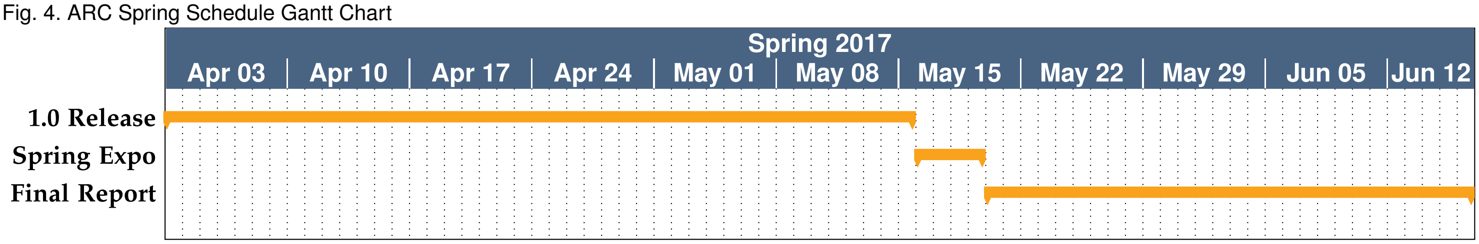


Fig. 4. ARC Spring Schedule Gantt Chart



3 Specific Requirements

This section describes the hardware, software, and performance requirements necessary for the ARC system to function. It lays out requirements for input and output of the system through interfaces. It also covers the major systems features and functional requirements for those features.

3.1 External interface requirements

This section describes all of the interfaces required for input to and output from the ARC system.

3.1.1 User interfaces

There should be a remote station (commonly referred to as a ground station) where all information about the vehicle can be seen. The ground station should allow user control of the vehicle including setting way-points, adjusting parameters for vehicle behavior, and direct control over vehicle movement.

3.1.2 Hardware interfaces

There should be hardware interfaces for manual control of the vehicle. This interface should be able to override software control so that users can safely commandeer control of the vehicle from the autopilot. Other hardware interfaces should be between a network interface between the autopilot and the companion computer on the vehicle. There should also be a hardware interface between the autopilot and the vehicle control system. We will use existing protocols for communication between the autopilot and the companion computer and the vehicle.

3.1.3 Software interfaces

There should be software interfaces between the user interfaces and the hardware interfaces. These interfaces will be handled by the operating system on both the ground station and the companion computer.

3.1.4 Communications interfaces

3.2 System features

Describes what the ARC system can be expected to do in terms of its major features.

3.2.1 System Feature: Image Analysis

3.2.1.1 Functional requirement image analysis 1:

1) ID: FR-IA1
2) Title: Fast Image Processing.
3) Description: Image analysis needs to be at a rate of around 15 or more frames per second.
4) Rationale: In order for the vehicle to move fast, images will need to be acquired and processed very fast.
5) Dependencies: N/A

3.2.1.2 Functional requirement image analysis 2:

1) ID: FR-IA2
2) Title: Depth finding
3) Description: Image analysis needs to determine how far away objects are.
4) Rationale: The proximity of objects will determine timing for speed and turning.
5) Dependencies: FR-IA1

3.2.1.3 Functional requirement image analysis 3:

1) ID: FR-IA3
2) Title: Object height
3) Description: Objects that are required to be detected, must be at least 12” tall.
4) Rationale: The car should be able to avoid obstacles that are taller than the car itself.
5) Dependencies: FR-IA2

3.2.2 System Feature: Sensors

3.2.2.1 Functional requirement sensor 1:

1) ID: FR-SN1
2) Title: GPS Data Collection
3) Description: When outside, GPS coordinates should be able to be obtained from a GPS unit and be accurate up to 10 meters, in an open space with no trees within 50 meters.
4) Rationale: Within an open field, this accuracy should be able to be obtained in order to provide reasonable locational accuracy to be used in navigation.
5) Dependencies: FR-HW1, FR-HW2

3.2.2.2 Functional requirement sensor 2:

1) ID: FR-SN2
2) Title: GPS Data Processing
3) Description: Data collected from the GPS unit will need to be relayed to the user through the main computational effort.
4) Rationale: A user needs to be able to view the location of the car at any time.
5) Dependencies: FR-HW1, FR-HW2, FR-SN1

3.2.3 System Feature: Navigation

The ARC vehicle needs to navigate a course autonomously. The following functional requirements describe what is required to achieve this.

3.2.3.1 Functional requirement navigation 1:

1) ID: FR-NAV1

2) Title: Motor Control
3) Description: To limit the uncertainty caused by the motor under 25% when going forward and backward.
4) Rationale: Maximize the accuracy of motor control to make prediction easier.
5) Dependencies: Probabilistic Analysis for motion

3.2.3.2 Functional requirement navigation 2:

1) ID: FR-NAV2
2) Title: Servo Control
3) Description: To limit the uncertainty caused by the servo under 25% when turning.
4) Rationale: Maximize the accuracy of servo control to make prediction easier.
5) Dependencies:

3.2.3.3 Functional requirement navigation 3:

1) ID: FR-NAV3
2) Title: Probabilistic Analysis for motion
3) Description: Use probability to estimate the current location of the vehicle with less than 25% error.
4) Rationale: Tires may slip and wobble, which will cause uncertainty on the location of the vehicle.
5) Dependencies: Motion Model

3.2.3.4 Functional requirement navigation 4:

1) ID: FR-NAV4
2) Title: Motion Model
3) Description: Motion model is significant as it decides what commands should be given by the computer under various circumstances.
4) Rationale: This is like learning how to drive a car. Knowing the configuration of the car is critical for the computer to know how to control it. Different combinations of speed and turning angle will result in different paths. Under different surface and weight distributions, the vehicle will also act differently.
5) Dependencies: Motor Control & Servo Control

3.2.3.5 Functional requirement navigation 5:

1) ID: FR-NAV5
2) Title: Global Path Planning
3) Description: When operating outdoor, the vehicle needs to go from one location to another following a legal path that is determined by algorithms using sensor data.
4) Rationale: A legal path is a path that goes on concrete surfaces and does not run into any objects.
5) Dependencies: Obstacle Avoidance, Sensors, Probabilistic Analysis for Motion

3.2.3.6 Functional requirement navigation 6:

1) ID: FR-NAV6
2) Title: Local Path Planning

3) Description: When operating indoor and given the map of the indoor area, the vehicle needs to go from one location to another within the area following a legal path that is determined by algorithms using sensor data.
4) Rationale: A legal path is a path that does not run into any objects.
5) Dependencies: Obstacle Avoidance, Sensors, Probabilistic Analysis for Motion

3.2.3.7 Functional requirement navigation 7:

1) ID: FR-NAV7
2) Title: Obstacle Avoidance
3) Description: When approaching an object, the vehicle needs to decide whether to turn left or right and by how much. The algorithm can overwrite the path planned by the path planning algorithms.
4) Rationale: When operating both indoor and outdoor, crashing into objects needs to be avoided.
5) Dependencies: Sensors, Control System, Probabilistic Analysis for Motion

3.2.3.8 Functional requirement navigation 8:

1) ID: FR-NAV8
2) Title: Parallel Parking
3) Description: Given sensor data and the estimate of the current location, the algorithm outputs the next optimal action (turning the front wheel a certain angle and go forward/backward at a certain speed).
4) Rationale: A dedicated algorithm for parallel parking is necessary because path planning algorithms do not promise the orientation of the vehicle. The parallel parking algorithm will make sure the vehicle is align with the curb/wall when finished.
5) Dependencies: Sensors, Control System, Probabilistic Analysis for Motion

3.2.4 System Feature: Hardware mounting

3.2.4.1 Functional requirement 1:

1) ID: FR-HW1
2) Title: Minimize Port and Hardware Exposure
3) Description: Seal any unused ports, encase electronic components to minimize environmental exposure.
4) Rationale: Minimizing dust and other elements to electronic components to reduce wear and tear.
5) Dependencies: N/A

3.2.4.2 Functional requirement 2:

1) ID: FR-HW2
2) Title: Secure Hardware Mounting
3) Description: Hardware should be mounted in such a way, that in the event that the vehicle rolls over or high-speed impact, hardware remains in place and moves no more than 1 cm from its original location.
4) Rationale: In the event of an impact or rollover, the hardware should remain in place, and not detach from the vehicle.
5) Dependencies: FR-HW1

3.2.5 System Feature: Communications

3.2.5.1 Functional Requirement Comms 1:

1) ID: FR-CM1
2) Title: Telemetry
3) Description: Telemetry needs to be transmitted from the vehicle to a ground station.
4) Rationale: Users need to see the current state of the vehicle at all times.
5) Dependencies: N/A

3.2.5.2 Functional Requirement Comms 2:

1) ID: FR-CM2
2) Title: Vehicle Control
3) Description: Control signals, such as ”start”, ”stop”, ”go to way-point”, etc. needs to be sent to the vehicle. Signals should be received and processed in under 2 seconds.
4) Rationale: Users need to have a way to command the vehicle remotely.
5) Dependencies: N/A

3.2.5.3 Functional Requirement Comms 3:

1) ID: FR-CM3
2) Title: Emergency Stop
3) Description: When sent a signal to stop, the car needs to stop within 2 seconds from when the command is sent.
4) Rationale: A failsafe must be in place if the car is ”going rogue”.
5) Dependencies: FR-CM2

3.3 Performance Requirements

This section specifies numerical requirements placed on ARC system:

- The system only supports one user at a time.
- The system only provides one interface for user interaction at any single moment.
- The system supports no more than 5 waypoints as inputs.
- The system is capable of driving the vehicle at maximum 10 miles per hour.
- The system is capable of navigating in an indoor environment of area no larger than 500 square feet.
- The system is capable of navigating outdoor without space constraints.

3.4 Design constraints

The ARC RC vehicle will be guaranteed to communication with the ground station within 2km due to limitations of the telemetry radio.

3.5 Software System Attributes

As this is a research project, requirements found along the way will be appended onto this section.

3.6 Other Requirements

As this is a research project, requirements found along the way will be appended onto this section.