Software Design Document

for

Baja SAE Drivetrain Optimization

Version 1.0 approved

Prepared by Krystal Bernal, Edit Vosganians, Roi Chico, Bowen Waugh, and Huang Xu

Cal State LA Baja SAE

11/28/2019
# Table of Contents

Table of Contents 2
Revision History 4

1. Introduction 5
   1.1 Purpose 5
   1.2 Document Conventions 5
   1.3 Intended Audience and Reading Suggestions 5
   1.4 System Overview 5

2. Design Considerations 7
   2.1 Assumptions and Dependencies 7
   2.2 General Constraints 7
   2.3 Goals and Guidelines 8
   2.4 Development Methods 8

3. Architectural Strategies 10

4. System Architecture 12
   4.1 Level 0 DFD 12
   4.2 Level 1 DFD 12

5. Policies and Tactics 14

6. Detailed System Design 15
   6.1 - Fast Data Collection Module 15
   6.2 - Slow Data Collection Module 15
   6.3 - Graphical User Interface Module(GUI) 16
   6.4 - Data Analysis Module 16

7. Detailed Lower Level Component Design 18

8. User Interface 20
   8.1 Overview of the User Interface 20

9. Database Design 21

10. Requirements Validation and Verification 22

Glossary 23

References 24
# Revision History

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
<th>Reason for Changes</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Submission</td>
<td>11/28/19</td>
<td>First Draft</td>
<td>v 1.0</td>
</tr>
</tbody>
</table>
1. Introduction

This SRS document is intended for the Baja SAE Drivetrain Optimization project.

1.1 Purpose

The purpose of this software is to assist the Baja SAE student-run team in optimizing the drivetrain of the competition vehicle. The software functions on two sets of RPI's, versions 3 and 4, running on Python 3.7. The goal is to be able to determine the ideal configuration of bolts and springs of the CVT’s based on sensor readings and analysis on the gathered data.

1.2 Document Conventions

Bold - Section or Table Headings

*Italicics with Underline* - Used for terms defined in the Glossary

*Italicics* - Figure name

1.3 Intended Audience and Reading Suggestions

This document is intended for use by the Cal State LA Baja SAE team members and faculty advisors as they use the Drivetrain Optimization data acquisition system. More specifically, the Electrical and Drivetrain subsections of the team may use this document to understand and further develop analysis methodology for optimizing the vehicle and presenting results to competition judges.

1.4 System Overview

The main function of the Drivetrain Optimization system is to provide feedback about the Cal State LA Baja SAE vehicle drivetrain configuration (Figure 1). The drivetrain for the Baja vehicle includes the engine, Continuously Variable Transmission (CVT), and the Gearbox. The part of the drivetrain our project focuses on is the CVT. The CVT consists of a Primary and Secondary, both of which are rotating mechanical parts connected by a belt. Inside of the Primary are weights and springs that can be changed to be heavier or stiffer, respectively.
Inside of the Secondary is a return spring that can also be substituted for one with a different stiffness.

Figure 1: Components of Cal State LA Baja SAE drivetrain

The configuration of the weights and springs inside the Primary and Secondary affects the ratio of their rotational speeds throughout a course. Exploring how these ratios change from one configuration to another allows the mechanical engineers to determine whether the power output by the engine is efficiently reaching the tires after passing through the rest of the system. The configuration of the Secondary return spring affects how quickly the CVT engages, affecting vehicle performance as well. There are five potentially unique tunes we need to find that correspond to the best performance for each dynamic competition event: Hill Climb, Tractor Pull, Acceleration, Maneuverability, and Endurance.

The Drivetrain Optimization system will record the rotational speed of the CVT Primary and Secondary, the change in position over time of the Secondary, the course environment, and the vehicle position and speed during test time. At analysis time, we will calculate the CVT ratio and compare it to the course, vehicle position, and vehicle speed to determine which tune performed the best and predict which type of tunes are the best for the defined competition events.
2. Design Considerations

2.1 Assumptions and Dependencies

The software is fully dependent on Python version 3.7; it will not be fully functional on any previous version of the language. The assumption is that the software will be utilized on the two dedicated RPI’s and no other device.

2.2 General Constraints

<table>
<thead>
<tr>
<th>Hardware Limitation</th>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component</td>
<td></td>
</tr>
<tr>
<td>VL6180X Distance Sensor Range</td>
<td>5mm - 100mm</td>
</tr>
<tr>
<td>55140 Series Hall Effect Sensor Range</td>
<td>10mm - 20 mm</td>
</tr>
<tr>
<td>VK162 GPS Module</td>
<td>No walls, clear sky or deadzone.</td>
</tr>
<tr>
<td>Pi Camera V1</td>
<td>1024 x 768 image resolution for object recognition to work.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Software Constraints</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor Name</td>
<td></td>
</tr>
<tr>
<td>55140 Series Hall Effect Sensor</td>
<td>The system has no constraints to our knowledge.</td>
</tr>
<tr>
<td>VL6180X Distance Sensor Range</td>
<td>The system shall gather data in units of inches.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RPi3 Constraints</th>
<th>Requirement Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The system is limited to the maximum memory size of SD memory card of 64 GB.

### Graphical User Interface Constraints

<table>
<thead>
<tr>
<th>Requirement Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The system shall receive data from RPi4 and RPi3.</td>
</tr>
<tr>
<td>The system shall use a screen of 5” 800 x 480 to display information.</td>
</tr>
</tbody>
</table>

#### 2.3 Goals and Guidelines

The goal of this software is to assist the Baja SAE student-run organization in optimizing the drivetrain of their competition vehicle. Other software goals include the following:

1. The system software shall be fast and reliable
   - a. The system shall be easy to understand
   - b. The system shall be easy to read
   - c. The system shall be easy to use
2. The GUI shall be user-friendly and informative
   - a. The design shall be easy to view on the monitor
   - b. The design shall display the appropriate information
3. The system software shall provide data analysis with correct filters
   - a. The system shall filter through outliers in the data
   - b. The system shall have troubleshooting procedures in place
   - c. The system shall calculate results using reliable data

#### 2.4 Development Methods

Our project development follows the Waterfall project management methodology. Requirements were set by the Cal State LA Baja SAE team at the beginning of the project, and development of modules follows a sequential order. This type of project development ensures that the software agrees with the Baja team’s build and testing timeline. Revisions to the software and its architecture would impede its availability for use during testing of the vehicle, so any changes may only be requested after preliminary collection and analysis of data is provided. Such changes will be left to a future software team and/or the Baja members responsible for programming and data acquisition.
3. Architectural Strategies

Python IDLE

Used as primary IDE for software and source code construction.

Gmap Library API

The Gmap Library API was utilized to integrate GPS data with the Google Maps web mapping service. Using this, several coordinates were plotted as to show the route taken by a vehicle with the mounted GPS.

ImageAI Library

The ImageAI Library is the key library that our systems uses for object detection. It provides an easy to understand and customizable interface to integrate image recognition into our system.

ADC Library Adafruit_MCP3008 Adafruit_GPIO.SPI

This library was essential in converting the analog data that we recorded from the sensors into digital data. It is employed by our project to handle the general inputs and outputs needed to be processed.

ADC Library V16180x

In order for the distance sensor to record data properly, this library was needed. It contains the necessary dependencies in order for our system to use the distance sensor.

Data Acquisition Hardware

The data acquisition hardware that is included in our system is comprised of Hall Effect sensors, a Distance Ranging sensor, Camera, GPS module, and an MCP3008 Analog to Digital converter.

System Hardware

The primary hardware of our system is comprised of a Raspberry Pi 3 and a Raspberry Pi 4. The Raspberry Pi 3 handles the collection of data which does not require high hardware performance. The Raspberry Pi 4 handles the collection of data which is heavily reliant on the performance of its operating system.

External Hardware
The hardware needed to interface with our system are basic computer peripherals, including a bluetooth keyboard, a mouse with a USB cord, and a screen with an HDMI cord.

Data Interpretation

Data collected by the sensors will be interpreted by the Fast Data Collection Module and the Slow Data Collection Module.
4. System Architecture

4.1 Level 0 DFD

At the top-most organizational level, our software accepts data from the sensors into two Raspberry Pi’s that are responsible for filtering, saving, and processing the inputs. In addition, the RPi3 is responsible for the GUI module of the system which will display the appropriate information requested by the Baja SAE team.

4.2 Level 1 DFD

4.2.1 Microprocessor 1 - RPi 4 Data Collection

The first RPi is used for the “fast data collection” system to gather the data generated by the RPM sensors and the distance sensor. The RPi will collect the data and run it through pre-set filters in order to sift through the noise for each sensor. The information sent to the second RPi
through the TCP/IP connection for processing. This is done to ensure that the speed of the data collection is not compromised.

### 4.2.2 Microprocessor 2 - RPi 3 Data Collection

The second RPi is used for the “slow data collection” system to gather the data generated by the GPS module and PiCamera, process and calculate all the data gathered, and generate the GUI display consisting of the information required by the Baja SAE team. This is where all the information is going to be stored including all the .csv and .jpg files.

### 4.2.3 GUI

The GUI is responsible for being the only interactive portion with the user. The GUI will accept the time, date, and CVT configuration from the user and use the inputs in its display. In addition, it will display the CVT Secondary return rate, CVT Primary and Secondary ratio RPM and some live sensor data being collected as a feed.
5. Policies and Tactics

Policies and tactics will be provided in a separate document as a user manual.
6. Detailed System Design

6.1 - Fast Data Collection Module

6.1.1 Responsibilities
The Fast Data Collection Module interprets the data collected from the Hall Effect Sensors and the Distance sensor, and sends them to the Slow Data Collection Module.

6.1.2 Constraints
See section 2.2.

6.1.3 Composition
The Fast Data Collection Module interacts with two Hall Effect Sensor, one Distance Sensor and one Raspberry Pi 4. A TCP/IP Connection shall be used for communication between the Raspberry Pi 3 and Raspberry Pi 4.

6.1.4 Uses/Interactions
The Fast Data Collection Module will interact with the Slow Data Collection Module. It will communicate with the Slow Data Collection Module by sending the data it collects.

6.1.5 Resources
The Fast Data Collection Module will utilize sensors to gather information.

6.2 - Slow Data Collection Module

6.2.1 Responsibilities
The Slow Data Collection Module will store GLL data from the GPS module, as well as record images every three seconds of the course environment. After sending appropriate processed data to the GUI, the Slow Data Collection Module will save all data into a central location.

6.2.2 Constraints
See section 2.2

6.2.3 Composition
The Slow Data Collection Module interacts with one GPS module, one Camera and one Raspberry Pi 3. A TCP/IP Connection shall be used for communication between the Raspberry Pi 3 and Raspberry Pi 4.

6.2.4 Uses/Interactions

The Slow Data Collection Module will interact with the Fast Data Collection Module and GUI. The Slow Data Collection Module will retrieve data from the Fast Data Collection Module and send appropriate processed data to the GUI.

6.2.5 Resources

The Slow Data Collection Module will utilize the data from the Fast Data Collection Module and hardware sensors to gather information.

6.3 - Graphical User Interface Module (GUI)

6.3.1 Responsibilities

The GUI will display data including the CVT Secondary Return data, the CVT Primary and Secondary RPM ratio and live sensor data that has been collected.

6.3.2 Constraints

See section 2.2.

6.3.3 Composition

The GUI will include the software to display data and output it to a screen.

6.3.4 Uses/Interactions

The GUI interacts with the Slow Data Collection Module. It receives data from the Slow Data Collection Module and display them on the screen.

6.3.5 Resources

The GUI will utilize the data obtained from the Slow Data Collection Module to gather information.

6.4 - Data Analysis Module

6.4.1 Responsibilities

The Data Analysis Module will determine the best weights and springs to use and the best clocking and stiffness to use in the CVT Secondary for each respective
course type. It will also track the path taken during the course runs and classify each recorded image, such as an uphill slope, turns and rocks.

6.4.2 Constraints

See section 2.2.

6.4.3 Composition

The Data Analysis Module includes the software to determine the best configuration for the vehicle.

6.4.4 Uses/Interactions

The Data Analysis Module does not interact with other modules.

6.4.5 Resources

The Data Analysis Module will utilize the Data form the Fast Data Collection Module and the Slow Data Collection Module.
7. Detailed Lower Level Component Design

New files will be added in future revisions of the document as the project progresses.

7.1 HallEffect.py

7.1.1 Classification
This file will implement the methods that will allow the two Hall Effect sensors to work.

7.1.2 Processing Narrative (PSPEC)
It is a python file that will collect data from the Primary and Secondary CVT and outputting an RPM.

7.1.3 Interface Description
This file does not need an interface, it only collects data.

7.1.4 Processing Detail
The data collected will be processed within the microprocessor.

7.1.4.1 Restrictions/Limitations
The system has no constraints to our knowledge.

7.1.4.2 Performance Issues
The microprocessor needs to have a fast CPU for it to collect accurate data.

7.1.4.3 Design Constraints
Hall Effect sensor needs to be between 10mm - 20mm from the magnet for it to be able to collect accurate data.

7.2 PiCam.py

7.2.1 Classification
This file will implement the methods that will allow the Pi Camera to work.

7.1.2 Processing Narrative (PSPEC)
It is a python file that will take pictures every 3 seconds.

7.1.3 Interface Description
This file does not need an interface, it only take pictures.

7.1.4 Processing Detail
The images collected will be processed from an on site computer.
7.1.4.1 Restrictions/Limitations
The system has no constraints to our knowledge.

7.1.4.2 Performance Issues
Pi camera needs a 1024 x 768 image resolution for object recognition to work.

7.1.4.3 Design Constraints
Pi camera needs to be placed in an angle where it can see the whole view of the driver for it to capture the best image.

7.3 GPS.py

7.3.1 Classification
This file will implement the methods that will allow the GPS module to work.

7.1.2 Processing Narrative (PSPEC)
It is a python file that will collect data from the GPS module and when finished will be able to plot it into a map.

7.1.3 Interface Description
This file does not need an interface, it only collects data.

7.1.4 Processing Detail
The data collected will be processed within the microprocessor.

7.1.4.1 Restrictions/Limitations
The system has no constraints to our knowledge.

7.1.4.2 Performance Issues
GPS module will have trouble if speed limit is going over 40 mph.

7.1.4.3 Design Constraints
GPS module needs to be placed in the highest elevation of the car for it to get the best accurate coordinates.
8. User Interface

8.1 Overview of the User Interface

A more detailed design of the GUI will be determined in a future revision.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The GUI shall provide a way to input the test date, time, CVT Primary</td>
<td>and Secondary weight and spring configurations, and course type.</td>
</tr>
<tr>
<td>The GUI shall display the CVT Secondary return rate.</td>
<td></td>
</tr>
<tr>
<td>The GUI shall display the ratio of RPMs of the CVT Primary and Secondary</td>
<td></td>
</tr>
<tr>
<td>The GUI may be required to show more live sensor data as required by</td>
<td>the Cal State LA Baja SAE team.</td>
</tr>
<tr>
<td>The system shall receive data from RPi4 and RPi3.</td>
<td></td>
</tr>
<tr>
<td>The system shall use a screen of 5” 800 x 480 to display information.</td>
<td></td>
</tr>
</tbody>
</table>
9. Database Design

The system does not have a database.
10. Requirements Validation and Verification

This section will be updated in a future revision.
Glossary

Abbreviations and Acronyms

- GPIO - General Purpose Input and Output
- RPi 3 - Raspberry Pi 3
- RPi 4 - Raspberry Pi 4
- OS - Operating System
- GUI - Graphical User Interface
- ADC - Analog to Digital Converter
- DFD - Data Flow Diagram
- NMEA - National Marine Electronics Association
- GLL - Geographic Latitude and Longitude

Definitions

- **CVT** - Continuously Variable Transmission. A type of clutch that has a Primary and Secondary component, which are connected by a belt.

- **Primary** - A rotating body connected to the engine by the engine output shaft. It drives the Secondary member by transferring power through a belt. The primary may be tuned by changing the internal weights and springs.

- **Secondary** - A rotating body connected to the Gearbox by the Gearbox input shaft. It is driven by the Primary through a belt. The Secondary may be tuned by clocking and changing the stiffness of its return spring.

- **Gearbox** - A mechanical device whose internals consists of gears that create a speed reduction from the input rotational speed to the output rotational speed. The input rotational speed comes from the Secondary and the output rotational speed goes to the vehicle wheel.
References

Documents

1.5.1 Analog Inputs for Raspberry Pi Using the MCP3008


1.5.5 NMEA Reference Manual


1.5.6 Collegiate Design Series Baja SAE® Rules 2020

Baja SAE, September 8, 2019. Revision A.

Setup Guides

- Distance Sensor Setup Guide: 

- Analogue Sensors On The Raspberry Pi Using An MCP3008: 
  https://www.raspberrypi-spy.co.uk/2013/10/analogue-sensors-on-the-raspberry-pi-using-an-mcp3008/