RIO min\_dist\_to\_stop\_bar Analysis Code

This document outlines code modifications to the “Real-time Intersection Optimizer” (RIO) python project, as part of the Autonomous Vehicles at Intelligent intersections and Advanced Networks (AVIAN) project at the University of Florida Transportation Institute (UFTI), that support the investigation of optimal minimum distance to stop bar values (min\_dist\_to\_stop\_bar). In this context, the optimal min\_dist\_to\_stop\_bar value is defined as the value that creates the minimum total travel time delay and outputs executed trajectories that are implementable. These code modifications allow you to run the code a chosen number of simulations at different min\_dist\_to\_stop\_bar values. In addition to outputting the original csv and plot files found in the original RIO for each value, the code also outputs a csv file that includes individual vehicle and total travel time delay and plots of the executed trajectories for each vehicle in their respective lanes. Finally, and most importantly, the code outputs a final csv file that records each total travel time delay for their respective min\_dist\_to\_stop\_bar values. The outputs are shown as follows:

For each simulation:

veh\_delay\_(min\_dist\_to\_stop\_bar value).csv

executed\_trajectories\_(min\_dist\_to\_stop\_bar value).png

For entire code:

veh\_delay\_total.csv

The reference file for line numbers and specific changes are from “GaleStad” branch in github on June 28th, 2019, found here: <https://github.com/Pourmehrab/RIO/tree/GaleStad>.

**main.py**

Import csv in line 5. Inserte the main function for the analysis modifications, delay\_analysis(min\_dist), on line 172 of the main.py file. Remove the original run\_rio(args) command. The new delay\_analysis(min\_dist) function, along with the new run\_rio(args) function, is referenced inside a “for” loop in what is now line 371. This “for” loop is where you can set a range for your desired min\_dist values.. The “for” loop introduces the variable min\_dist, therefore this variable must be referenced in the run\_rio and delay\_analysis functions. Add the variable to the run\_rio function in line 21. The variable should also be added to different calls of the Intersection function in line 55, SimTraffic function in line 59, and the MCF\_SPaT function in line 71.

**config.py**

Add the variable min\_dist to the function load\_inter\_parms in line 117. Next, change the min\_dist\_to\_stop\_bar value to the variable min\_dist for the intersection you want to analyze.

**intersection.py**

In the Intersection class, add the min\_dist variable to the \_init\_ function in line 20, and to the call of the load\_inter\_params function in line 24.

**traffic.py**

These analysis modifications only relate to the SimTraffic class. RealTimeTraffic can be ignored. In line 269, add the min\_dist variable to the \_init\_ function. More script is needed in lines 316-324 for the creation of the necessary csv files. Also, you will need to change the names of the trj\_point\_level and trj\_veh\_level csv files in line 309 and 311.

**signal.py**

In line 37, insert the min\_dist variable into the \_init\_function. Edit the script in line 58 in order to create new file name. Add the min\_dist variable to the \_init\_ function in the MCF\_SPaT class in line 258, and to the function call in line 275.