MATLAB-SUMo INTERFACE WITH

TRACI4MATLAB

ABHINAV ANAND
Summer Project Intern
IIT-Kharagpur

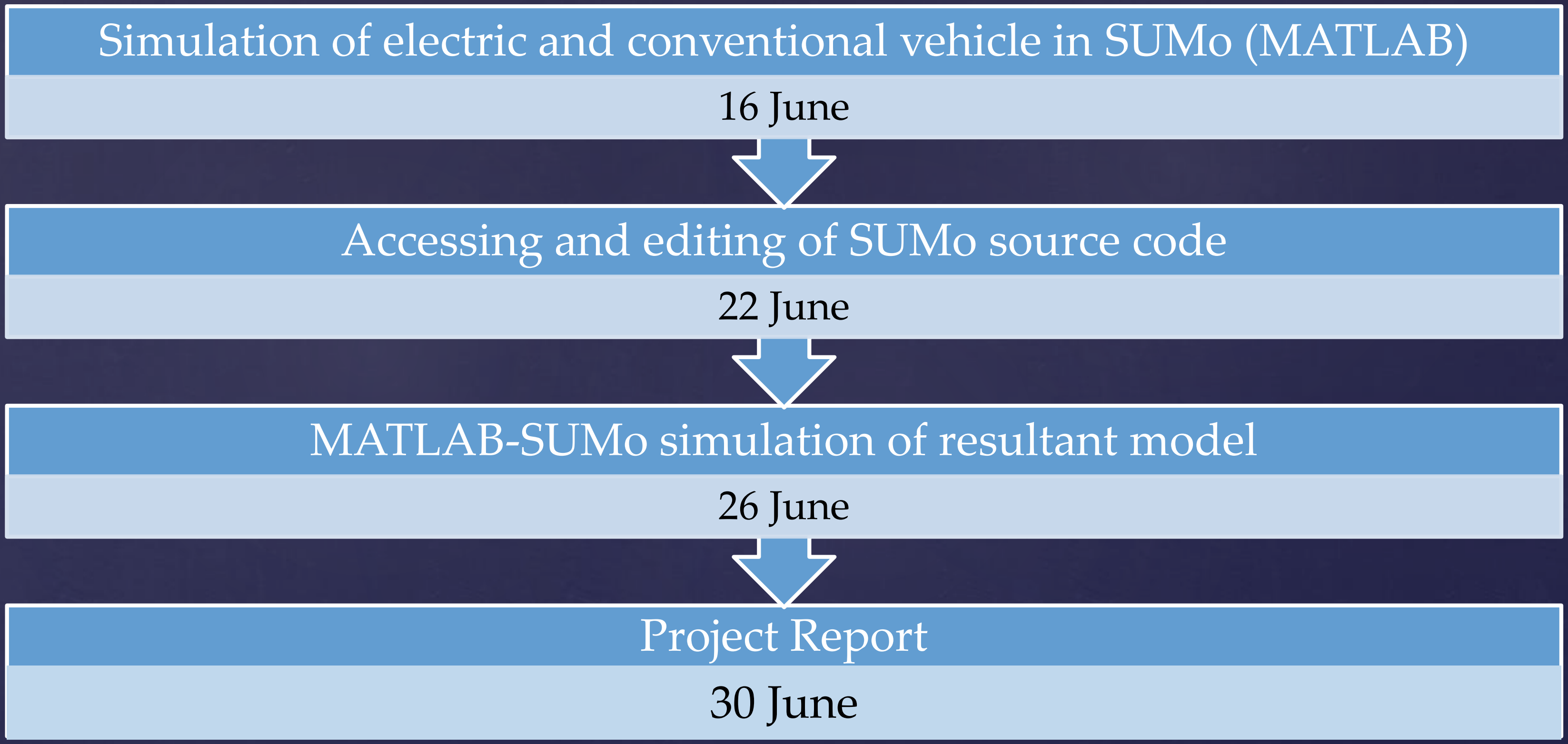
OBJECTIVES:

- Understand the working of Electric vehicle and conventional vehicle simulation in SUMo through MATLAB

- Access the vehicle models for energy/emission calculation and edit the source code to get required hybrid vehicle model

- Simulate the resultant model in SUMo through Matlab

TIMELINE



Simulation of electric and conventional vehicle in SUMo (MATLAB)

16 June

Accessing and editing of SUMo source code

22 June

MATLAB-SUMo simulation of resultant model

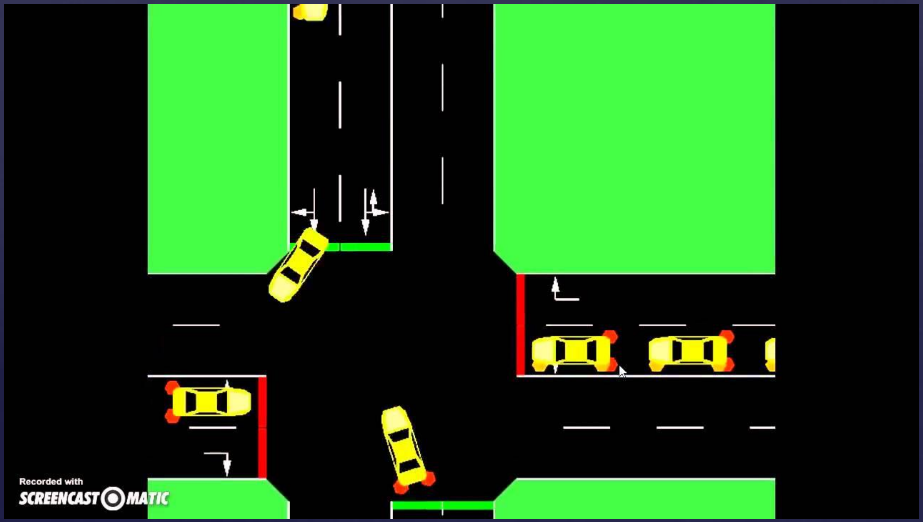
26 June

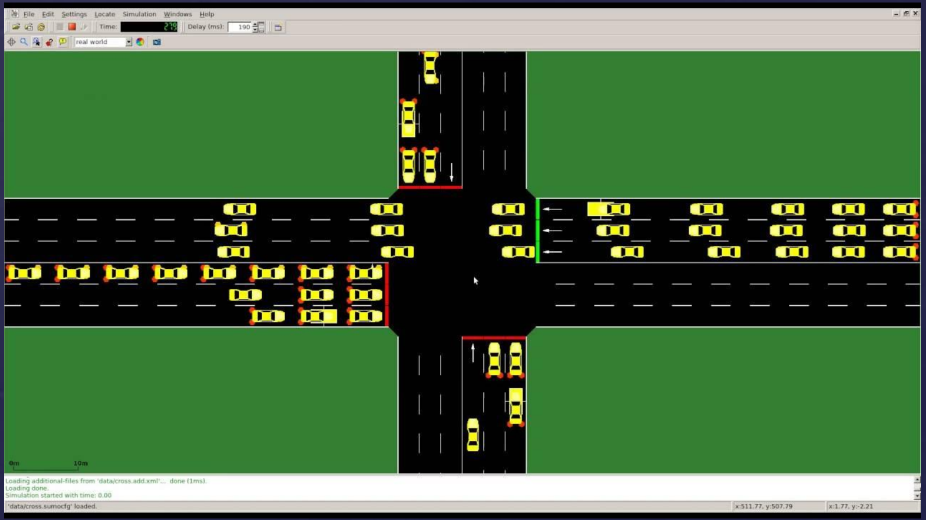
Project Report

30 June

About SUMo

- SUMo is a simulation software based on C++ that simulates traffic and traffic lights on a map where turns, paths, lanes, speed and vehicle density is defined.
- The simulation platform offers many features, like ***microscopic simulation*** (small area of map), ***online interaction***(access sumo using other software like TRACI) and the simulation of ***multimodal traffic*** (cars, trucks, buses, pedestrians etc.)



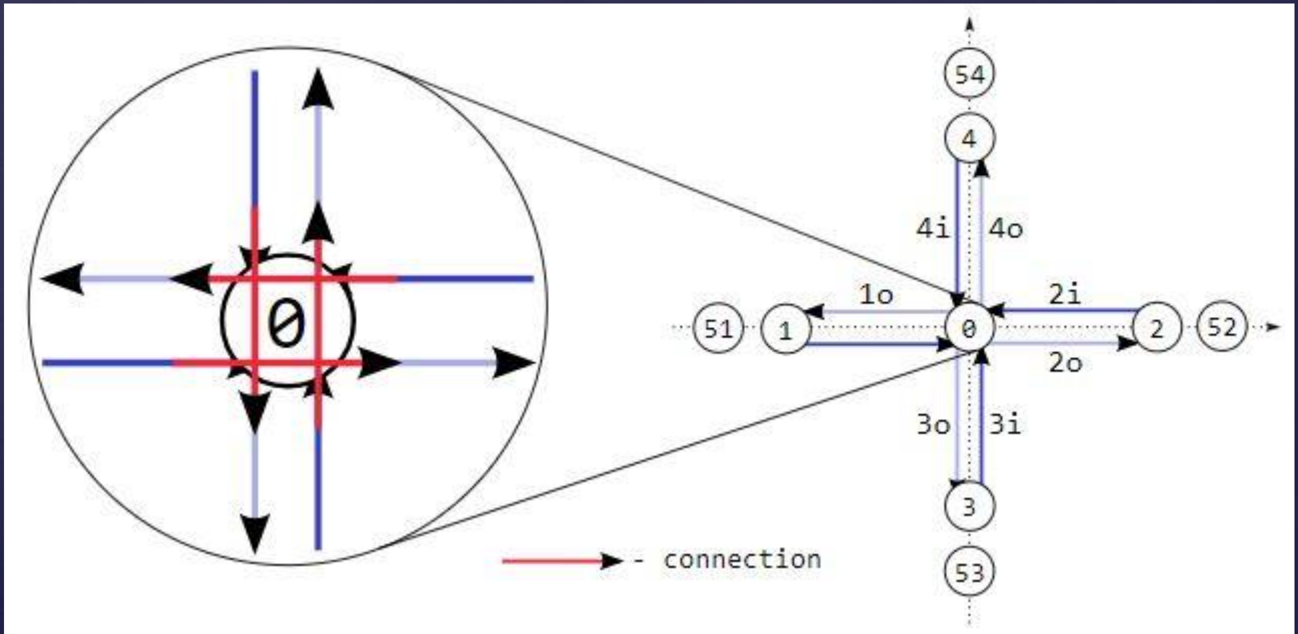


```
Loading additional-files from 'data/cross.add.xml'... done (1ms).
Loading done.
Simulation started with time: 0.00
'data/cross.sumocfg' loaded.
x:511.77, y:507.79
x:1.77, y:-2.21
```

Steps in TraCI modelling

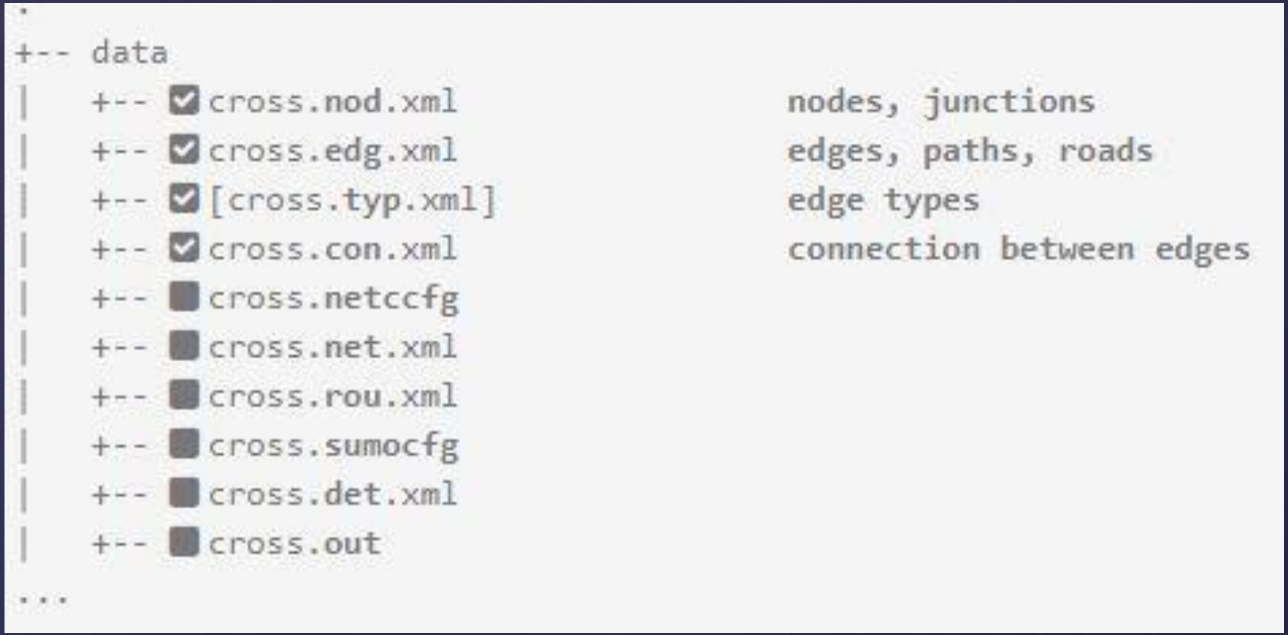
Building a road network

- Each SUMO model is built on a network of roads, junctions, traffic lights, and other infrastructure items such as induction loop detectors. Compared to the vehicles and their movements, these are the **static** elements of the model. For TraCI, we separate these definitions from the Python scripts in a data folder.

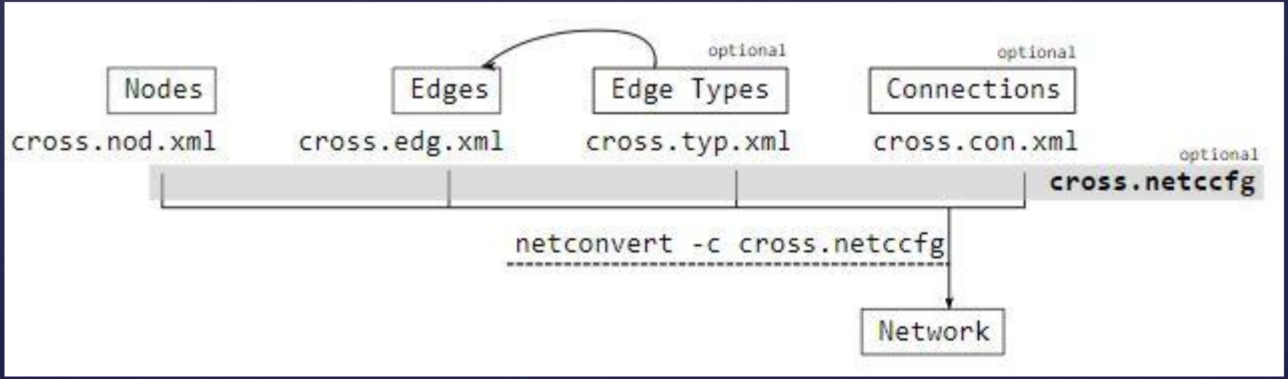


Generating the .net.xml file

We use the functions netconvert and *.netccfg to compile all the data files created.



```
.
+-- data
|   +-- ☑cross.nod.xml                         nodes, junctions
|   +-- ☑cross.edg.xml                         edges, paths, roads
|   +-- ☑[cross.typ.xml]                       edge types
|   +-- ☑cross.con.xml                         connection between edges
|   +-- ■cross.netccfg
|   +-- ■cross.net.xml
|   +-- ■cross.rou.xml
|   +-- ■cross.sumocfg
|   +-- ■cross.det.xml
|   +-- ■cross.out
...
```



Add Vehicles and Routes with *.rou.xml

To get traffic moving on our network, we first need to define vehicles and routes for vehicles to travel along.

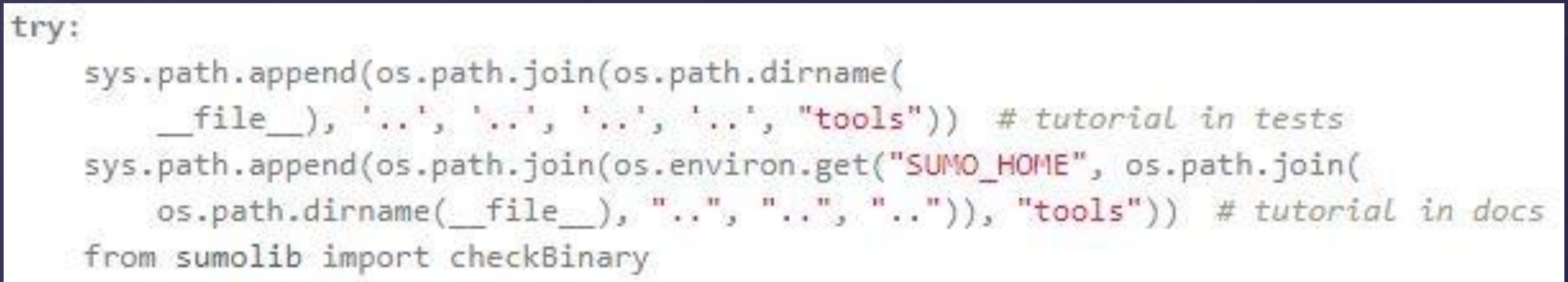
This is done in the cross.rou.xml file

Generate Traffic

Generate_routefile()

Control Traffic with Traffic Lights and Induction Loop

trafficlights.setPhase(string,int) ->

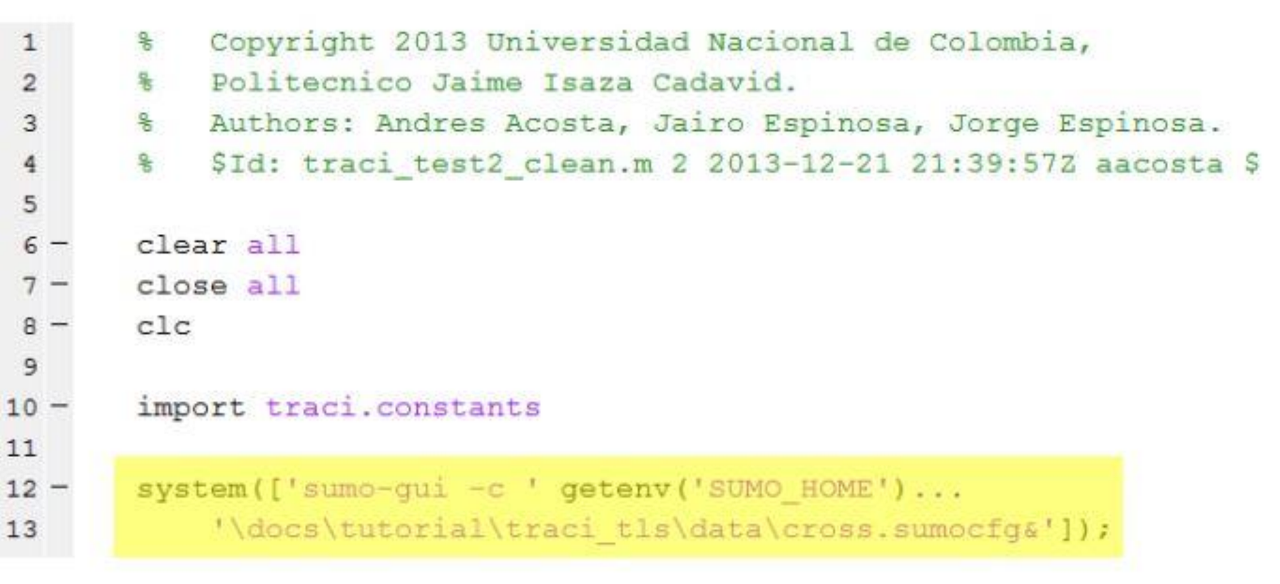


```
try:
    sys.path.append(os.path.join(os.path.dirname(
        __file__), '..', '..', '..', '..', "tools"))  # tutorial in tests
    sys.path.append(os.path.join(os.environ.get("SUMO_HOME", os.path.join(
        os.path.dirname(__file__), "..", "..", "..")), "tools"))  # tutorial in docs
    from sumolib import checkBinary
```

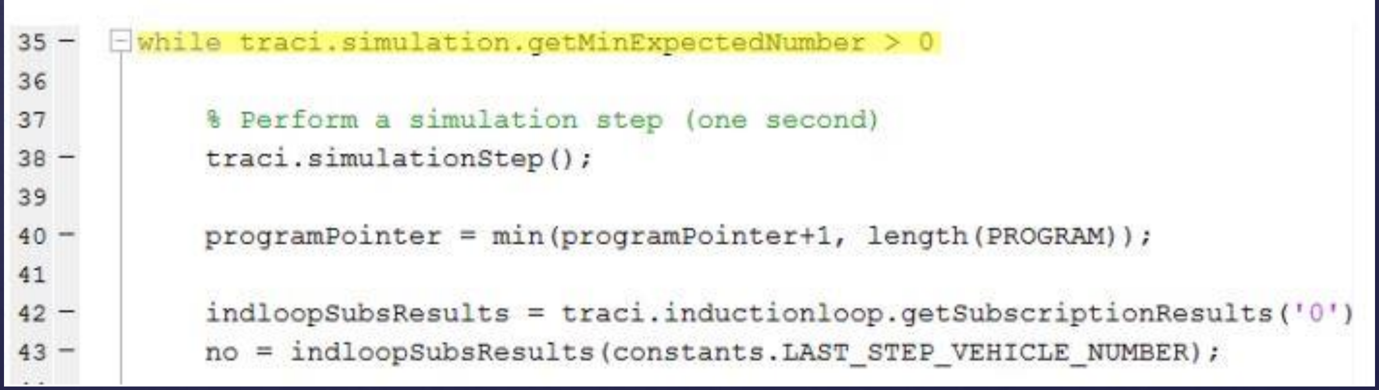
- The try block tries to locate and include the TraCI python module traci and sumolib from
  the $SUMO_HOME/tools directory.

- **Finally**, once import has been completed, the script triggers the sumo model and then execute
  the run() function to establish the connection with the following code section

MATLAB

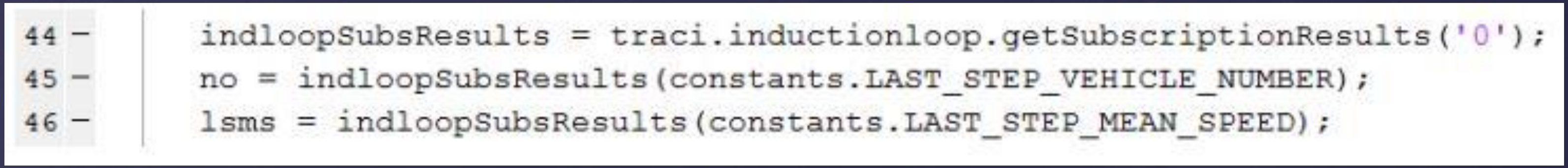


```matlab
%   Copyright 2013 Universidad Nacional de Colombia,
%   Politecnico Jaime Isaza Cadavid.
%   Authors: Andres Acosta, Jairo Espinosa, Jorge Espinosa.
%   $Id: traci_test2_clean.m 2 2013-12-21 21:39:57Z aacosta $

clear all
close all
clc

import traci.constants

system(['sumo-gui -c ' getenv('SUMO_HOME')...
    '\docs\tutorial\traci_tls\data\cross.sumocfg&']);
```



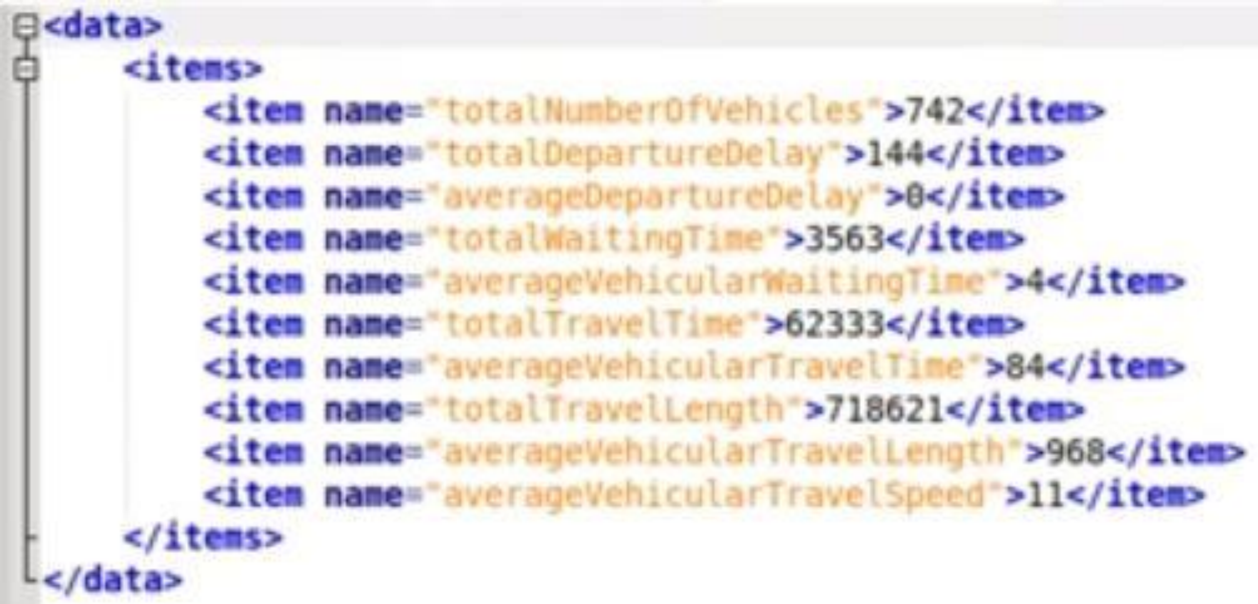
```matlab
while traci.simulation.getMinExpectedNumber > 0

    % Perform a simulation step (one second)
    traci.simulationStep();

    programPointer = min(programPointer+1, length(PROGRAM));

    indloopSubsResults = traci.inductionloop.getSubscriptionResults('0')
    no = indloopSubsResults(constants.LAST_STEP_VEHICLE_NUMBER);
```

Getting results from MATLAB



```
indloopSubsResults = traci.inductionloop.getSubscriptionResults('0');
no = indloopSubsResults(constants.LAST_STEP_VEHICLE_NUMBER);
lsms = indloopSubsResults(constants.LAST_STEP_MEAN_SPEED);
```

The results from TraCI simulation are in the form of TraCI constants which are the traffic parameters for a given map system



```
<data>
    <items>
        <item name="totalNumberOfVehicles">742</item>
        <item name="totalDepartureDelay">144</item>
        <item name="averageDepartureDelay">0</item>
        <item name="totalWaitingTime">3563</item>
        <item name="averageVehicularWaitingTime">4</item>
        <item name="totalTravelTime">62333</item>
        <item name="averageVehicularTravelTime">84</item>
        <item name="totalTravelLength">718621</item>
        <item name="averageVehicularTravelLength">968</item>
        <item name="averageVehicularTravelSpeed">11</item>
    </items>
</data>
```