Accessing SUMo from MATLAB

Using traci4matlab

INITIALIZATION

> Traffic control interface(TraCI), helps us use SUMo through other softwares. For MATLAB it is available as traci4matlab.

> SUMo versions from 0.20.1 come with in-built traci4matlab. Latest version of SUMo is 0.32.0.

> First we add SUMO_HOME as an environment variable in our computer and traci4matlab in our MATLAB path.

> Then a java path text file has to be created in the preference directory of MATLAB to traci4matlab.jar.

> After this traci_test.m can be run to confirm installation.

STARTING SUMo from MATLAB

import traci.constants

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

Starting TraCI

```
% Initialize TraCI
traci.init();

traci.inductionloop.subscribe('0');
for i=1:length(steps)

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

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

Acessing SUMo variables

The SUMO objects are grouped in thirteen domains: gui, lane, poi, simulation, trafficlights, vehicletype, edge, inductionloop, junction, multientryexit, polygon, route, and vehicle.

The general structure to access or modify a SUMO object is:

traci.<domain>.<get/set_wrapper()>,

Where domain can take any of the domains listed previously and get/set_wrapper() are the functions to acces the values (get) or modify (set) the attributes of the object of interest.

>If the velocity's value of the vehicle with ID veh_1 in the current time step is required, the command current_speed_veh1 = traci.vehicle.getSpeed('veh_1') can be executed.

>Some other commands:

```
Command Window
>> help traci.lane
Contents of traci.lane:

getAllowed                      - Get the allowed vehicle
getCO2Emission                  - Returns the CO2 emissic
getCOEmission                   - Returns the CO emission
getContextSubscriptionResults   - Get the context subscri
getDisallowed                   - Get the disallowed vehi
getEdgeID                       -
getFuelConsumption              - Get the fuel consumptic
getHCEmission                   - Returns the HC emission
getIDList                       - Get the IDs of the lane
getLastStepHaltingNumber        - Get the number of halti
getLastStepLength               - Get the mean vehicle le
getLastStepMeanSpeed            - Get the average speed c
getLastStepOccupancy            - Get the percentage of c
getLastStepVehicleIDs           - Get the IDs of the vehi
getLastStepVehicleNumber        - Get the number vehicles
getLength                       -
getLinkNumber                   -
```

Getting Results

For example, suppose that it's desired to make a TraCI subscription to acces the values of the attributes LAST_STEP_VEHICLE_NUMBER and LAST_STEP_MEAN_SPEED of the *induction loop* with ID '0'. In this case, the command shown in figure 12 shall be used. Note that the import traci.constants command must be issued at the beginning of the *script*, as explained in the step 1.

```
traci.inductionloop.subscribe('0',{constants.LAST_STEP_VEHICLE_NUMBER,..
    constants.LAST_STEP_MEAN_SPEED});
```

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

Closing TraCI

traci.close()