

SUMO ENHANCEMENT FOR VEHICULAR COMMUNICATION DEVELOPMENT

Kit Guan Lim, Chun Hoe Lee, Renee Ka Yin Chin, Kiam Beng Yeo and Kenneth Tze Kin Teo

Modelling, Simulation and Computing Laboratory, Faculty of Engineering, Universiti Malaysia Sabah, Malaysia

Abstract

It is normal that every family is having at least one vehicle at their home as vehicles have become a daily needs for all of us. However, this also leads to the increased of road accidents where major causes are related to human errors which can be prevented. To tackle with this problem, vehicular ad hoc network (VANET) is introduced with the aim to make vehicles intelligent. In order to study the algorithm in VANET, a mobility simulator is needed for simulation purpose. In this case, SUMO is proved to be a good simulation tool in generating VANET environment while MATLAB is good for algorithm development. Yet, to develop a good simulation platform, modification on SUMO files are necessary. This paper discusses on the procedures in creating a left-hand traffic (LHT) simulation file that is suitable to be used in Malaysia. LHT simulation is not easy to achieve as modification on the road connection and traffic light files are required. This paper also showed the results of the simulation after SUMO files modification. Apart from that, this paper also showed the simulation of VANET environment using SUMO and MATLAB through a third party interfacing named TraCI4Matlab, which allows communication between MATLAB and SUMO simulator.

Keywords:

Mobility Simulator, SUMO, MATLAB, SUMO Files Modification

1. INTRODUCTION

Transportation has become an important role in our lives. So it is common that every family is having at least one vehicle in their house. However, this situation of vehicle increment has also led to an increase of traffic congestion and accidents every year. As the number of vehicles on the road increased, it also means higher congestion and higher probability of accidents happened. Chairman of International Organisation for Road Accident Prevention, Bob Joop Goos stated that human error is the main factor causing most of the accidents [1]. At the same time, it cannot be denied that driver's condition while driving is causing traffic congestion too. For the past few decades, traffic congestion and safety have become a hot topic with different approaches being researched. Some researchers tried to monitor traffic condition using trajectory clustering method to find out the factors causing heavy traffic congestion [2-4], while some of them trying to optimise the traffic light signal for optimum time allocation to ensure smooth traffic flow [5-7]. Apart from that, there are also researches known as vehicular ad hoc network (VANET) being carried out in making vehicles more intelligent [8]. Once intelligence has been given to vehicles, vehicles are able to react automatically under different circumstances to prevent unwanted accidents.

VANET is a type of network that uses Dedicated Short Range Communications (DSRC) and IEEE 802.11p as the standards for PHYSICAL layer address and MAC layer address configuration. The standardization process for IEEE 802.11p comes from the allocation of DSRC spectrum at a frequency of 5.9GHz and

bandwidth of 75MHz, which allows communication range of up to 1000m [9], [10]. These standards allow high message delivery rate with low delay among vehicles, leading to a smooth communications among the vehicles. However, creating a real VANET environment is costly, time consuming and inefficient, hence most of the researches are only done through simulation.

In order to obtain good results in VANET simulation, large number of nodes needs to be distributed in a realistic testbed to form a large vehicular network while evaluation is to be made under a controlled environment [11]. If simulation is performed in an uncontrolled environment, the results might be affected by other disturbance such as noise, making troubleshooting task more difficult to be carried out. On the other hand, to be able to make comparison among the proposed model with other models, there is a need to have a standardized benchmarks and test scenarios [12]. As user-friendly network simulators are becoming well-known and easily available, modelling of network protocols can be a lot easier. VANETs simulation requires two important components, namely the networking component and mobility component [13]. Networking component acts as the bridge for vehicles to communicate with each other, while mobility component generate the trace for vehicles' movement, forming a virtual vehicular network.

2. MOBILITY SIMULATORS

Mobility simulators are mainly used to generate the movement of vehicles' pattern under a certain trace [8]. Some of the widely used simulators are Quadstone Paramics [14], Traffic Software Integrated System (TSIS-CORSIM) [15] and SUMO [16]. Each of the software has its advantages and disadvantages in VANET development, but the combination of Simulation of Urban Mobility (SUMO) and MATLAB is one of the most convenient approaches in this development [17].

2.1 QUADSTONE PARAMICS

Quadstone PARAMICS is a microscopic traffic and pedestrian simulation software used by planning professionals to design efficient, economical, driver and pedestrian friendly transportation infrastructure. This software allows operational assessment for current and future year traffic conditions [14]. The name PARAMICS is an acronym derived from PARAllel computer MICropic Siumulation. In PARAMICS, movements for different types of vehicles can be modelled, where vehicle type is defined by physical characteristics such as length, height, width, weight and maximum speed. Pedestrian interaction with road network can also be modelled through the provision of pedestrian phases at signal-controlled intersections. Vehicles' motion in PARAMICS is alike with other microscopic traffic models, composed of a series of discrete steps, which result in the perception of motion when strung together, displaying vehicles

moving throughout the network [18]. The Fig.1 shows an example interface of the software, where user can set the route for the vehicles' movement.

2.2 VISSIM

VisSim is developed by Altair Engineering. It is a visual language for modelling and simulating nonlinear dynamic systems. Martin and Peter have conducted research, presenting the possibilities of validating microscopic traffic flow simulation model on microscopic and macroscopic level. VisSim implements a psycho-physical car-following model and provides a very realistic driving behaviour. The complex model offers many model parameters that can be calibrated using measurement data from driving experiments. The model parameters can be adjusted to reflect different traffic situations [19]. There is another study performed by Lu *et al.* where simulation platform is built by integrating MATLAB into VisSim. They proved that integrating both of the software solve the difficulties in doing secondary development for VisSim, which is to compute algorithm to solve the complex problems in the model [20]. In Fig.2, it can be seen that VisSim offers a 3D view at a specific junction, which is good for simulation and visualisation purpose.

2.3 TSIS-CORSIM

Traffic Software Integrated System (TSIS) is an integrated development environment that enables users to conduct traffic operation analysis. It contains tools that allow user to define and manage traffic analysis projects, create inputs for traffic simulation, execute traffic simulation models and interpret the results of the models used. While Corridor Simulation (CORSIM) consists of integrated set of two microscopic simulation models that represent the entire traffic environment. Combining TSIS and CORSIM, TSIS-CORSIM becomes a microscopic traffic simulation software package for signal systems, freeway system, or a combination of both signal and freeway system. Several successful TSIS-CORSIM applications are used to demonstrate that CORSIM is a very effective tool to perform traffic operations analysis and to evaluate new Intelligent Transportation System (ITS) technologies [21]. The Fig.3 shows an example of the TSIS interface, displaying the nodes for vehicular communication, where each nodes representing a vehicle.

3. DEVELOPMENT OF MOBILITY MODEL

Normally, developing a realistic simulation is not an easy task as compared to deriving the mobility of moving nodes based on the trace files recorded in real-world. However, if the vehicles' traces are produced using traffic simulation tools, users are able to take full control of the traffic scenario. This also opens up the possibility to allow network simulations to be integrated into traffic simulations and control the simulations flow [22].

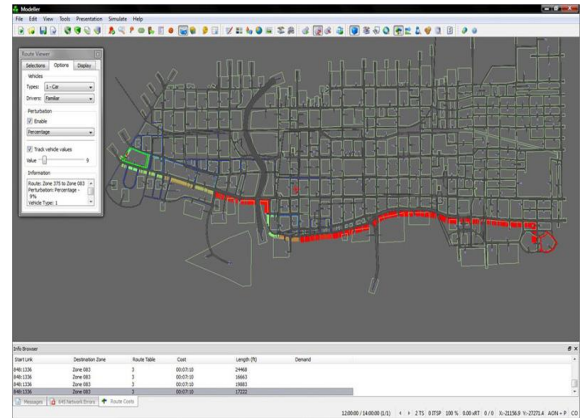


Fig.1. Example Workspace for Quadstone Paramics



Fig.2. Example Workspace for VisSim

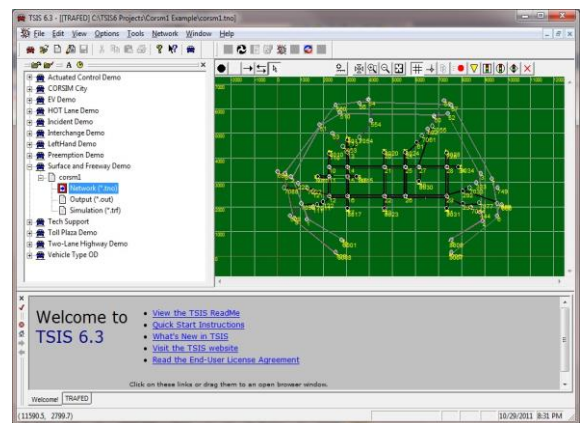


Fig.3. Example Workspace for TSIS-CORSIM

3.1 SUMO

Since 2001, SUMO has become an open source traffic simulation software that is widely used globally. It allows user to model traffic systems that include road vehicles and public transport, even pedestrians can also be modelled into the traffic systems. SUMO also included various supporting tools that can handle different task, such as route finding, importing network from open street map and etc. It can further be improved with self-defined models and provides various features to control the simulation remotely.

SUMO has been widely used to tackle with variety research project, for example, evaluating modern algorithm by increasing the performance of traffic lights. In 2005 when Pope's visit to City of Cologne and during Soccer World Cup 2006, SUMO is used to forecasts the traffic in the city. Apart from that, SUMO too is widely used by vehicular community as it can provide realistic vehicle traces in any map imported into it, while evaluation process can easily be taken placed with the help of a network simulator that can work with SUMO [16].

Over the past few decades, SUMO has evolved into a full featured suite of traffic modelling tool, which is able to read different source formats using its unique road network importer. The major contribution for such development is undertaken by Institute of Transportation Systems at German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt, DLR). The simulation in SUMO is time-discrete based with default step length of 1s, while minimum time step is 1ms. The simulation process for SUMO is capped at a maximum duration of 49 days [23].

SUMO also has a feature that allows users to import road network into it which simplified the process for developing mobility model. Through Java Open Street Map (JOSM) Editor, real world road network can be easily downloaded and edited, as shown in Fig.4. In the editor, modification such as inserting traffic lights, configuring road connections, defining number of lanes on a road, etc. can be made to suit the users' need.

The map is then further processed through SUMO's command *netconvert* to generate simulation network containing heuristically computed values. After that a *trip* file is generated to simulate vehicle flows on the network. The Fig.5 shows the road network simulated in SUMO with vehicles moving in random across the whole network.

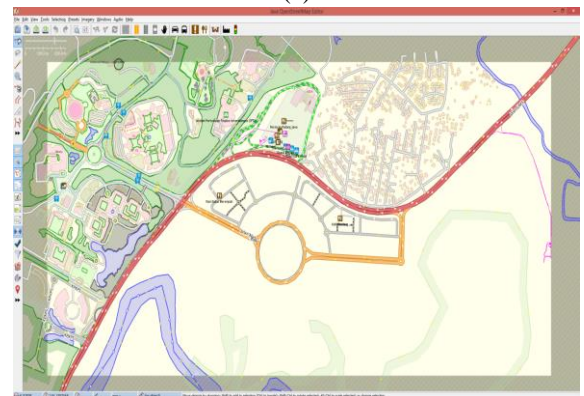
3.2 MATLAB

MATLAB on the other hand is a fourth-generation programming language with a multi-paradigm numerical computing environment. It allows user to plot different functions and data, performing matrix calculation, computing and implementing algorithms into it, or even creating a custom user interfaces and interface with programs written in other languages. In other word, MATLAB can also be a good alternative for developing mobility model for VANETs. There are some VANETs scenario simulations given by MathWorks that are open to public [24].

In Fig.6, an example of Urban City is simulated in MATLAB. This model has 9×9 intersections with 400 vehicles (black dot) moving on the road. To simulate the movement of vehicles, figure generated in MATLAB is updated every 0.1 seconds with new vehicle's position. Hence this simulation takes up a lot of processing power which may be too heavy to perform using older computers.



(a)

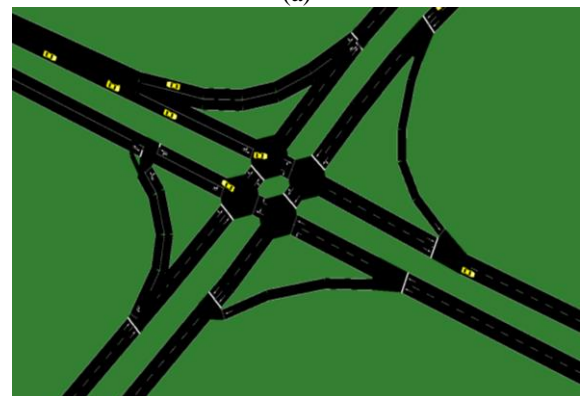


(b)

Fig.4. (a) Original OpenStreetMap Network (b) Network imported into JOSM Editor



(a)

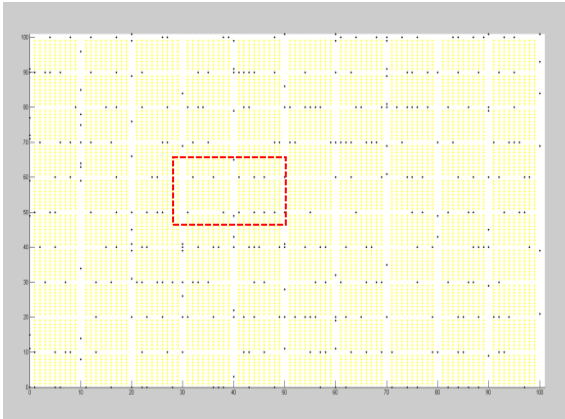


(b)

Fig.5. Network simulated in SUMO

3.3 TRACI FOR MATLAB

Although mobility model can be produced through MATLAB and SUMO, there is still some difficulties faced to perform study on VANETs. For instance, SUMO itself does not offer users to alter the vehicles flow with their developed algorithm. On the other hand, MATLAB allows users to compute algorithm within it, but the vehicle flow simulation will used up a lot of the processing power available in MATLAB, making it harder to perform further algorithm development. Hence, TraCI4Matlab is introduced to solve this situation.



(a)

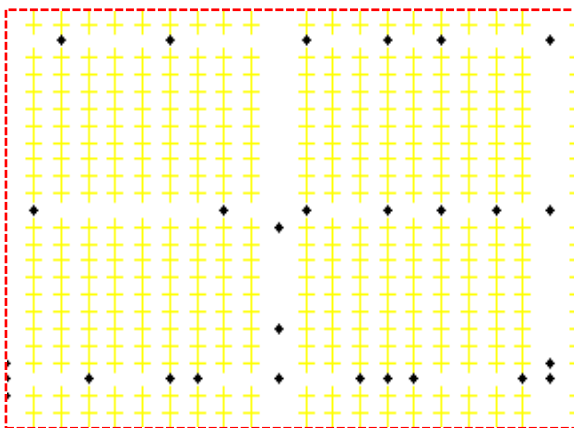


Fig.6. Urban VANET Simulation in MATLAB

TraCI4Matlab is an Application Programming Interface (API) developed in MATLAB that allows communication among applications written in MATLAB and SUMO simulator. TraCI4Matlab implement Traffic Control Interface (TraCI) application level protocol, built on top of the TCP/IP stack, so that the application developed in MATLAB, which is the client, can access and modify the simulation environment provided by the server, SUMO. TraCI4Matlab not only allows MATLAB to take control of SUMO objects such as vehicles, traffic lights, etc, but also allow MATLAB to enable traffic lights predictive control and dynamic route assignment among others [25].

```
<?xml version="1.0" encoding="UTF-8"?>

<configuration
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/sumo
  Configuration.xsd">

  <input>
    <net-file value="cross.net.xml"/>
    <route-files value="cross.rou.xml"/>
    <additional-files value="cross.det.xml"/>
  </input>

  <time>
    <begin value="0"/>
    <step-length value="1"/>
  </time>

  <report>
    <verbose value="true"/>
    <no-step-log value="true"/>
  </report>

  <traci_server>
    <remote-port value="8813"/>
  </traci_server>

</configuration>
```

Fig.7. SUMO configuration file

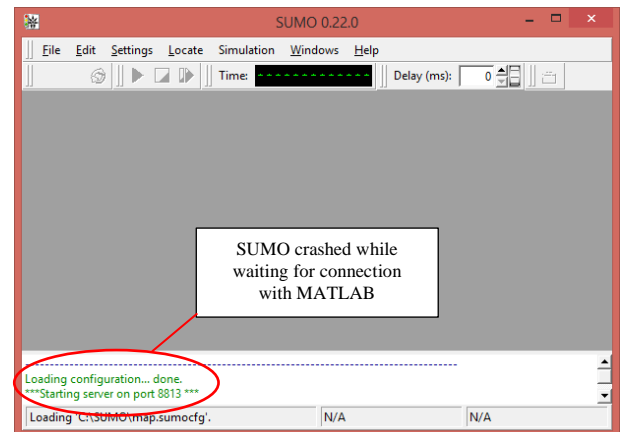


Fig.8. SUMO crashed while trying to communicate with MATLAB

In order to establish communication port for MATLAB and SUMO, SUMO configuration file need to be edited as shown in Fig.7. In the configuration file, input needs to be specified with the correct input files or else the simulation might not work properly. *step-length* value refers to the step time of the simulation, depending on the user, time-step can also be set to 0.1s for smoother simulation flow. *traci_server* declare a connection port for communication between MATLAB and SUMO. This part is important and must be included when running SUMO with MATLAB for TraCI to work, but must be excluded if simulation is initiated from SUMO. Otherwise SUMO will crash when trying to establish a connection to MATLAB, as shown in Fig.8. Normally the *traci_server* part is included when simulation is initiated from MATLAB, since MATLAB is the part to establish server port for SUMO to connect with it for further communication.

4. ENHANCEDMENT OF SUMO

Although SUMO and MATLAB can perform well in generating VANET scenario, but it is still insufficient to have a close to real world VANET environment. In SUMO, default simulation is set to have one vehicle being inserted into the simulation scenario every second. This means in order to have a large amount of vehicles moving on the road, simulation has to be run for a long time before further process being taken place. Even worse, if the map is large, it is very difficult to form a vehicular network on a certain area of the map. Furthermore, SUMO by default is designed to work in right-hand traffic (RHT) as in the European country. Here in Malaysia, left-hand traffic (LHT) is implemented, hence some modification needs to be done in SUMO so that simulation process is smooth and running in order.

The Fig.9 shows one of the unwanted situation occurred in SUMO when there is a junction in the simulation, which is in a mess when running the simulation as shown in Fig.10. This situation will not affect the simulation in SUMO, but it does not reflect the real world as such junction does not exist. Furthermore, the smoothness of the traffic flow is also affected by such junction where all the vehicles stopping in the middle of junction causing vehicles not able to move. Solving this problem might not be crucial but is kind of important for later stage of algorithm testing in VANET [26, 27], for example testing the efficiency of algorithm developed such as network coding that is used to improve nodes communication [28, 29].

Modifying the traffic junction is not very difficult but time consuming. First of all, the generated `<name>.net.xml` file from `netconvert` in SUMO needs to be separated into several parts for easier modification using the command `“..sumo-0.22.0\bin\netconvert” --sumo-net-file <name>.net.xml --plain-output-prefix <name2>”`. After executing such command, there will be files with the extension `<name2>.con.xml`, `<name2>.edg.xml`, `<name2>.nod.xml` and `<name2>.tll.xml`. In this paper, focus will be on `<name2>.con.xml` because this is the file with all the road connection information.

First step on modifying the connection file is to open the file `<name2>.con.xml` for editing. Notepad++ is recommended for easier viewing and editing. The next step is to look for the correct connection name for editing. For example, by referring to Fig.11, `130856267#1` is the junction need to be edited, the number behind the junction name after underscore is the lane number, modification according to Fig.12 will be able to change the way vehicle flows. After changing the road connection, traffic light that is affecting vehicle flow needs to be edited also. A sudden traffic light appeared in the middle of junction is unrealistic, but SUMO does not allow removal of the traffic lights, hence the only way in solving this trouble is to disable the traffic lights.

Traffic lights information is saved in the `<name2>.tll.xml` file.

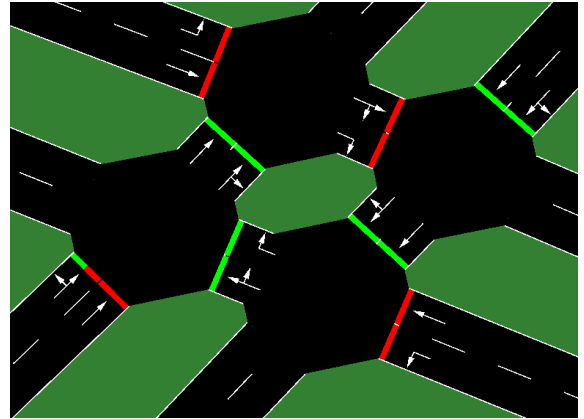


Fig.9. Cross-junction simulated in SUMO

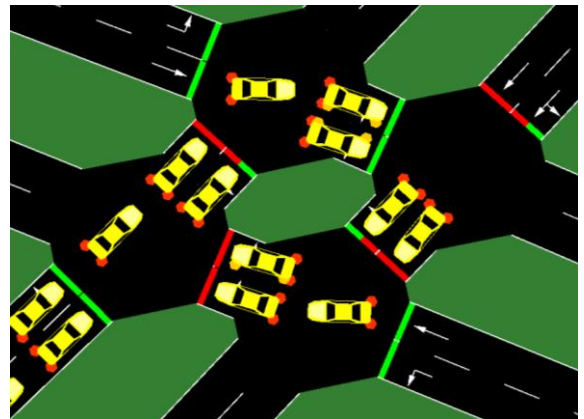


Fig.10. Vehicles simulation in SUMO

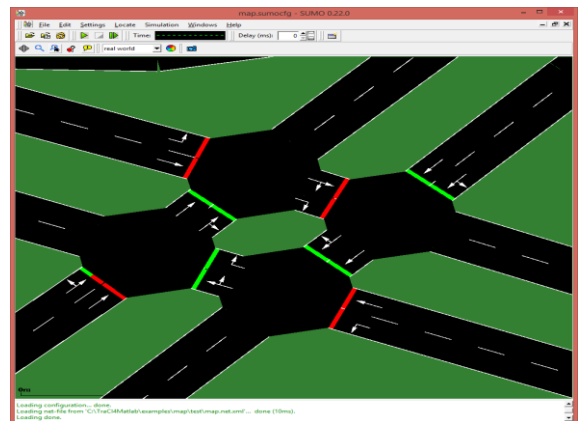


Fig.11. SUMO’s junction with name labelled

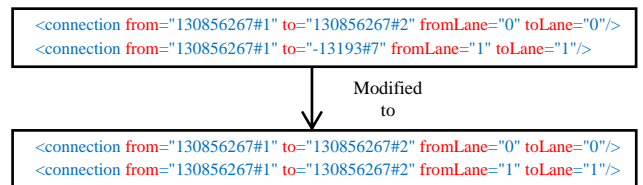


Fig.12. Road connection modification in `<name2>.con.xml`

```
<tlLogic id="1440171875" type="static" programID="0" offset="0">
  <phase duration="31" state="GGGr"/>
  <phase duration="9" state="yyrr"/>
  <phase duration="31" state="GrrGG"/>
  <phase duration="9" state="yrryy"/>
</tlLogic>
```

Modified to

```
<tlLogic id="1440171875" type="static" programID="0" offset="0">
  <phase duration="31" state="OOOr"/>
  <phase duration="9" state="OOOr"/>
  <phase duration="31" state="OOOG"/>
  <phase duration="9" state="OOOy"/>
  <phase duration="31" state="OOOr"/>
  <phase duration="9" state="OOOr"/>
  <phase duration="31" state="OOOr"/>
  <phase duration="9" state="OOOr"/>
</tlLogic>
```

Fig.13. Traffic light modification in `<name2>.tll.xml`

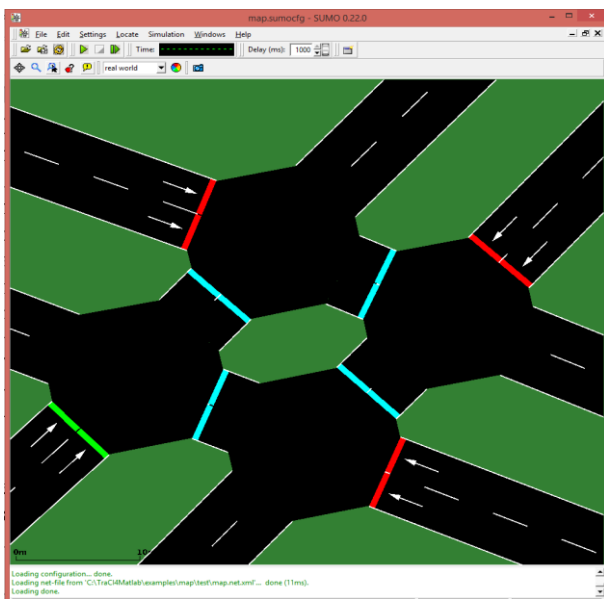


Fig.14. SUMO's junction after modification

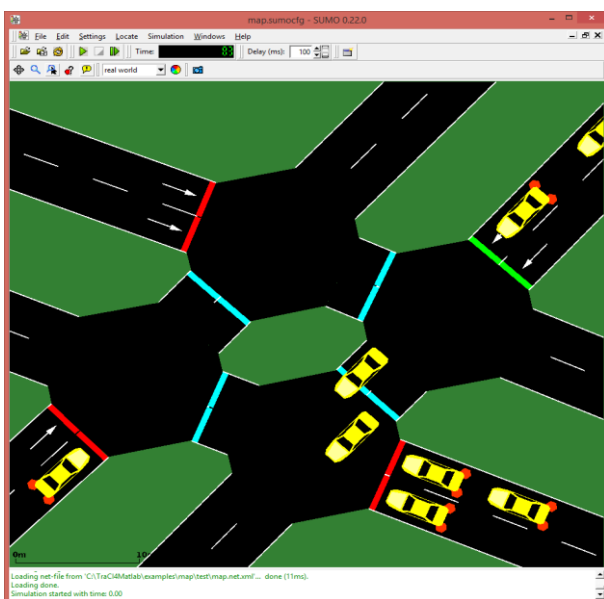


Fig.15. SUMO simulation with smooth vehicle flow

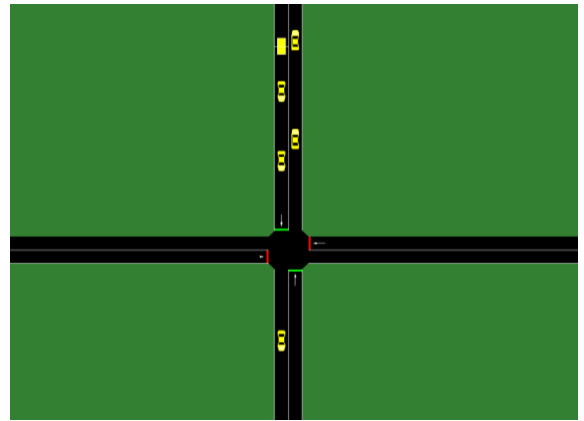


Fig.16. SUMO simulation initiated in MATLAB

Name	Value	Min	Max
MinExpectedNum...	0	0	0
NSGREEN	'GrGr'		
NSYELLOW	'yryr'		
NSlaneoccupancy	<1x200 double>	0	11
PROGRAM	<1x14 cell>		
WEGREEN	'rGrG'		
WEYELLOW	'ryry'		
WElaneoccupancy	<1x200 double>	0	4
ans	<1x2 cell>		
i	200	200	200
indloopSubsResults	<1x1 containers.Map>		
myRYGDefinition	<1x1 traci.trafficlight...>		
no	0	0	0
programPointer	14	14	14
steps	<1x200 double>	1	200
testVehicle	'1'		

Fig.17. Data in SUMO is exported into MATLAB's workspace when simulation is running

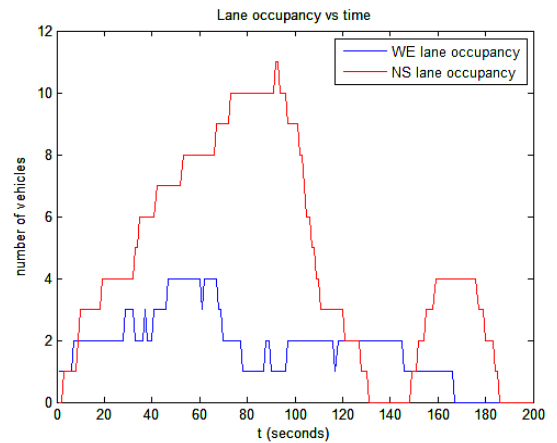


Fig.18. Result generated in MATLAB based on simulation results in SUMO

So the next procedure is to edit this file. The process of editing is more or less the same as modifying the road connection. Traffic light junction name has to be found before modification. Once the name has been identified, open the traffic light file and search for the desired traffic light. As shown in Fig.11, every traffic light in SUMO simulation is having its own *tlLogic* id. The *phase duration* indicates the time needed for traffic light before switching to next state, while *state* refers to whether the traffic light is in green, yellow or red condition. Traffic light *id* is assigned to every

junction, meaning that one id will control the junction's traffic lights that allow vehicles to flow in. In the traffic light state, "O" refers to traffic light is turned off, "G" is green light, "y" is yellow and "r" is red light. Modification is made based on Fig.13.

Modifying the road connection file and traffic light file may use up a lot of time, depending on the size of the map. Larger map means more road connections and more traffic lights to be edited. It is preferable that simulation is taken place under a controlled area of map. After all the modification, the separated files need to be combined together again before generating vehicles flow file using the command "...\\sumo-0.22.0\\bin\\netconvert" --edge-files <name2>.edg.xml --node-files <name2>.nod.xml --connection-files <name2>.con.xml --tlogic-files <name2>.tll.xml -o <name>.net.xml". Result after such modification is shown in Fig.14 while vehicles are simulated as in Fig.15. From the figures, it can be seen that vehicles will no longer stacked in the middle of the junction as in Fig.10, solving the unnecessary congestion which is not happening in real world, making the simulation environment closer to real world. In Fig.14, traffic lights in the middle of the junction have turned into light blue, indicating that they have been disabled.

5. RESULTS AND DISCUSSION

In section 3, it can be seen that SUMO is suitable for simulation as compared to MATLAB. However, SUMO does not offer user the feature to compute and implement algorithm into it. On the other hand, MATLAB has been globally used in algorithm computation is suitable for VANET algorithm development. Hence, in order to have a smooth simulation and close to real world simulation, SUMO's configuration files need to be edited accordingly as mentioned in Fig.7 to be able to link with MATLAB. The advantages and disadvantages for using SUMO and MATLAB are summarized in Table.1.

Table.1. Pros and Cons between Matlab and SUMO

Characteristics	Mobility Simulator	
	MATLAB	SUMO
VANET Design Process	Takes a lot of time, need to calculate and draw the road	Fast design, map can be obtained through Open Street Map
VANET Visualization	Messy and laggy network, different from real world road network	Real world road network and smooth vehicles flow
Vehicle's Characteristics	Need to define vehicle's characteristics in detail	Has all information related to vehicles such as speed, emission, etc.
Algorithm Development	Allow complex algorithm to be developed	Need third party software to compute research algorithm
Results Analysis	Easy data collection and result analysis	Difficult to collect vehicles information for analysis

In section 4, it is shown that a lot of works are needed to modify the road connection and traffic light files. It is time

consuming and can be difficult to modify the files line by line, searching for desired junction, modifying the road for smoother traffic flow, yet the results shown in Fig.14 and Fig.15 are satisfying. It might not be significant in current stage as only simulation results are shown, however in the later stage where algorithm is going to be implemented into SUMO, unnecessary disturbance showed in Fig.10 can be a big issue in determining the efficiency of algorithm developed.

TraCI4Matlab which can be used to link SUMO and MATLAB has been introduced in section 3. Combining all the advantages and compensating the disadvantages of MATLAB and SUMO, TraCI4Matlab makes the development of VANET simulation an easier task. In fact, SUMO has become one of the preferred open-source platforms for researchers to perform microscopic road traffic simulation thanks to TraCI, which offers a high level of flexibility, allowing a client to retrieve and modify the objects in the simulation.

The Fig.16 shows the simulation in SUMO which is initiated from MATLAB. This simulation is different from ordinary SUMO simulation, as connection port between MATLAB and SUMO has been established. When the simulation is running, number of vehicles at the intersection is being counted and recorded into workspace, as in Fig.17. The Fig.18 shows the result plotted in MATLAB using data saved in the workspace.

6. CONCLUSION

SUMO is proved to be a good simulation tool for vehicular ad hoc network (VANET) while MATLAB is good for future algorithm development. To develop a good simulation platform, modification on SUMO files are necessary. This paper provides a direction for user to create a left-hand traffic (LHT) simulation file that is suitable to be used in Malaysia. Most of the hard work in creating LHT simulation is to modify the road connection and traffic light files. Section 4 focuses on the files to be edited and showed the results after modification. While section 5 showed the results of using TraCI4Matlab to establish connection between SUMO and MATLAB. It is recommended that user make a good decision in deciding the map for simulation. Smaller map is easier to modify, while maintaining the same map for the whole VANET development can prevent a start over on the SUMO files modification process.

ACKNOWLEDGMENT

The authors would like to acknowledge the Ministry of Higher Education (KPT) for supporting this research under Fundamental Research Grant Scheme (FRGS), grant no. FRG0365-ICT-1/2014, Exploratory Research Grant Scheme (ERGS), grant no. ERG0046-ICT-1/2013 and MyPhD scholarship support under MyBrain15 Program.

REFERENCES

- [1] O. Olarte, "Human Error Accounts for 90% of Road Accidents", Available: <http://www.alertdriving.com/home/fleet-alert-magazine/international/human-error-accounts-90-road-accidents>, Accessed on 2016.

- [2] M.Y. Choong, R.K.Y. Chin, K.B. Yeo and K.T.K. Teo, "Trajectory Clustering for Behavioral Pattern Learning in Transportation Surveillance", *Proceedings of 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology*, pp. 119-123, 2014.
- [3] M.Y. Choong, R.K.Y. Chin, K.B. Yeo and K.T.K. Teo, "Trajectory Pattern Mining via Clustering based on Similarity Function for Transportation Surveillance", *International Journal of Simulation: Systems, Science and Technology*, Vol. 17, No. 34, pp. 191-197, 2016.
- [4] M.Y. Choong, L. Angeline, R.K.Y. Chin, K.B. Yeo and K.T.K. Teo, "Vehicle Trajectory Clustering for Traffic Intersection Surveillance", *Proceedings of IEEE International Conference on Consumer Electronics*, pp. 783-787, 2016.
- [5] H.S.E. Chuo, M.K. Tan, B.L. Chua, R.K.Y. Chin and K.T.K. Teo, "Computation of Cell Transmission Model for Congestion and Recovery Traffic Flow", *Proceedings of IEEE International Conference on Consumer Electronics*, pp. 667-671, 2016.
- [6] K.T.K. Teo, K.B. Yeo, S.E. Tan, Z.W. Siew and K.G. Lim, "Design and Development of Portable Fuzzy Logic based Traffic Optimizer", *Proceedings of IEEE International Conference on Consumer Electronics*, pp. 783-776, 2013.
- [7] K.T.K. Teo, K.B. Yeo, Y.K. Chin, H.S.E. Chuo and M.K. Tan, "Agent-based Traffic Flow Optimization at Multiple Signalized Intersections", *Proceedings of 8th Asia International Conference on Mathematical Modelling and Computer Simulation*, pp. 459-463, 2014.
- [8] C.H. Lee, K.G. Lim, B.L. Chua, R.K.Y. Chin and K.T.K. Teo, "Progressing Toward Urban Topology and Mobility Trace for Vehicular Ad Hoc Network (VANET)", *Proceedings of IEEE Conference on Open Systems*, pp. 343-346, 2016.
- [9] V. Kumar, S. Mishra and N. Chand, "Applications of VANETs: Present and Future", *Computer Science and Communications*, Vol. 5, No. 1, pp. 12-15, 2013.
- [10] C.H. Lee, K.G. Lim, B.L. Chua, R.K.Y. Chin and K.T.K. Teo, "Performance Evaluation of IEEE 802.11 for Vehicular Communication", *Proceedings of IEEE International Conference on Consumer Electronics*, pp. 743-747, 2016.
- [11] C. Sommer, I. Dietrich and F. Dressler, "Realistic Simulation of Network Protocols in VANET Scenarios", *Proceedings of International Conference on Mobile Networking for Vehicular Environments*, pp. 117-119, 2007.
- [12] H. Hartenstein and K.P. Laberteaux, "A Tutorial Survey on Vehicular Ad Hoc Network", *IEEE Communications Magazine*, Vol. 46, No. 6, pp. 164-171, 2008.
- [13] M.K. Patel, "Comparative Study of Vehicular Ad-hoc Network Mobility Models and Simulators", *International Journal of Computer Applications*, Vol. 47, No. 6, pp. 38-43, 2012.
- [14] P.S.I. Ltd, Available at: <http://www.paramics-online.com/>, Accessed on 2016.
- [15] M. Tsis-Corsim, Available at: <http://mctrans.ce.ufl.edu/featured/TSIS/>, Accessed on 2012.
- [16] I.O.T. Systems, Available at: http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/, Accessed on 2016.
- [17] K.G. Lim, C.H. Lee, R.K.Y. Chin, K.B. Yeo and K.T.K. Teo, "Simulators for Vehicular Ad Hoc Network (VANET) Development", *Proceedings of IEEE International Conference on Consumer Electronics*, pp. 913-917, 2016.
- [18] R.L. Bertini, R. Lindgren and S. Tantiyanugulchai, "Applications of Paramics Simulation at a Diamond Interchange", Research Report, Portland State University, 2002.
- [19] P.V. Martin Fellendorf, "Validation of the Microscopic Traffic Flow Model VISSIM in Different Real-World Situations", *Proceedings of Annual Meeting on Transportation Research*, pp. 171-175, 2001.
- [20] W. Lu, W. Feng and L. Huang, "Integrated Simulation Platform of VISSIM, VS++ and MATLAB", *Proceedings of 5th International Conference on Transportation Engineering*, pp. 221-224, 2015.
- [21] L.E. Owen, Y. Zhang, L. Rao and G. McHale, "Street and Traffic Simulation: Traffic Flow Simulation using CORSIM", *Proceedings of 32nd Conference on Winter Simulation*, pp. 443-447, 2000.
- [22] C. Somme, I. Dietrich and F. Dressler, "Realistic Simulation of Network Protocols in VANET Scenarios", *Proceedings of International Conference on Mobile Networking for Vehicular Environments*, pp. 1117-1119, 2007.
- [23] D. Krajzewicz, J. Erdmann, M. Behrisch and L. Bieker, "Recent Development and Applications of SUMO-Simulation of Urban Mobility", *International Journal on Advances in Systems and Measurements*, Vol. 5, No. 3-4, pp. 128-138, 2012.
- [24] T. MathWorks, Available at: <http://www.mathworks.com/matlabcentral/fileexchange/41832-vanet-node-roadside-unit-scenario-simulations>, Accessed on 2013.
- [25] TraCI4Matlab: User's Manual, Available at: <https://in.mathworks.com/matlabcentral/fileexchange/44805-traci4matlab?requestedDomain=www.mathworks.com>
- [26] M.K. Tan, H.S.E. Chuo, R.K.Y. Chin, K.B. Yeo and K.T.K. Teo, "Optimization of Urban Traffic Network Signalization using Genetic Algorithm", *Proceedings of IEEE Conference on Open Systems*, pp. 943-946, 2016.
- [27] M.K. Tan, H.S.E. Chuo, R.K.Y. Chin, K.B. Yeo and K.T.K. Teo, "Genetic Algorithm based Signal Optimizer for Oversaturated Urban Signalized Intersection", *Proceedings of IEEE International Conference on Consumer Electronics*, pp. 945-949, 2016.
- [28] K.T.K. Teo, R.K.Y. Chin, S.E. Tan, C.H. Lee and K.G. Lim, "Exploration of Genetic Algorithm in Network Coding for Wireless Sensor Networks", *International Journal of Simulation: Systems, Science and Technology*, Vol. 15, No. 6, pp. 83-89, 2014.
- [29] K.T.K. Teo, R.K.Y. Chin, S.E. Tan, C.H. Lee and K.G. Lim, "Performance Analysis of Enhanced Genetic Algorithm based Network Coding in Wireless Networks", *Proceedings of 8th Asia International Conference on Mathematical Modelling and Computer Simulation*, pp. 49-53, 2014.