

## ***UTNSim*: A NEW TRAFFIC SIMULATOR BASED ON THE LWR-IM MESOSCOPIC TRAFFIC MODEL**

K. M. NG<sup>1,\*</sup>, M. B. I. REAZ<sup>2</sup>, M. H. F. RAHIMAN<sup>1</sup>

<sup>1</sup>Faculty of Electrical Engineering, Universiti Teknologi MARA,  
40450, Shah Alam, Selangor DE, Malaysia

<sup>2</sup>Department of Electrical, Electronic and Systems Engineering,  
Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia,  
UKM Bangi, 43600, Malaysia

\*Corresponding Author: kokmun77@yahoo.com.sg

### **Abstract**

Transportation and traffic simulation tools are used to describe traffic networks and simulate performances of traffic systems. The advent of traffic models in the past a few decades has become the “backbone” for these traffic simulation tools. This paper aims to presents a new traffic simulator *UTNSim* (Urban Traffic Network Simulator) which is based on a mesoscopic model called LWR-IM (Lighthill-Witham-Richards - Integrated Model). However, the structure and implementation algorithm of the *UTNSim* are explained in this paper. Therefore, as a preliminary evaluation of this traffic simulator tool; a real traffic situation in the city of Shah Alam, Malaysia is specified using the *UTNSim*. Queues simulated are compared with real queues by mean of the two-sampled t-test and mean absolute error. The obtained results reveal a valid traffic simulator tool that could simulate traffic performances that closely resemble the real traffic situation. This is proven as the results from the t-test reveal that there was no significance difference between simulated and real data.

Keywords: Mesoscopic model; Traffic simulator; *UTNSim*; LWR-IM traffic model.

### **1. Introduction**

Studies on traffic and transportation systems can be accomplished by mean of manual traffic counting [1, 2] and model-based traffic simulations [3-5]. Model-based traffic simulations provide an avenue to the mathematical modelling of urban road traffic systems (e.g., urban traffic network, freeway junctions, arterial

<b>Nomenclatures</b>	
$Q_s$	Queue at begin of green, <i>veh</i> (units of vehicle)
$Q_r$	Residue queue, <i>veh</i>
$Q_i$	Initial queue, <i>veh</i>
$q(t)$	Flow, <i>veh/s</i>
$S(t)$	Speed, <i>m/s</i>
<b>Greek Symbols</b>	
$\rho(t)$	Density, <i>veh/m</i>
$\phi_n$	Signal phase, $n= A, B, C$ and $D$
<b>Abbreviations</b>	
LWR-IM	Lighthill-Witham-Richards-Integrated Model
MAE	Mean Absolute Error
TD	Today Delay
TNDE	Traffic Network Diagram Editor
UTN	Urban Traffic Network
UTNSim	Urban Traffic Network Simulator
VTD	Vehicle to Downstream

routes, roundabouts, etc.). This is accomplished through the use of software tools that aid the planning and design of the traffic systems. Simulation tools in road traffic systems is important as it can implement models too complicated for analytical or numerical treatment, can be applied for experimental studies and provide predictions of present and future traffic scenarios. Predictions from these simulation tools could serve in model-based control of the urban networks [3, 5].

Traffic simulation tools based its functionalities on three main categories of traffic models namely macroscopic, microscopic and mesoscopic models [6]. Macroscopic model enables description of traffic flow behaviour using average aggregate traffic stream characteristics, such as flow rate, speed and traffic density. Microscopic model describes the movement of each individual vehicle in the traffic stream which derives its behaviour based on the car following model [7] and cellular automaton [8]. Mesoscopic model integrates the properties of both microscopic (individual vehicles) and macroscopic models. However, the movements of individual vehicles and their interactions are based on macroscopic parameters [6].

Traffic simulators derived their functions from traffic models. Macroscopic traffic simulators includes *FREQ 12* [9], *TRANSYT* [10] and *Aurora* [11]. Traffic simulators such as *VISSIM* [12], *Paramics* [13], *SimTraffic* [14], *AIMSUN* [15] and *Integration* [16] are examples of microscopic simulators. Some examples of mesoscopic simulators are *DynaMIT* [17] and *DynusT* [18].

Dion et al. [6] outlined the advantages and disadvantages of these three types of traffic simulator. According to [6], Microscopic simulators are able to simulate and animate the behaviour and interaction of individual vehicles; thus, providing a more detail representation of the traffic situation. However, this could attributes to modeling complexities especially for larger networks and significant computation time required for executing simulations. On the other hand, macroscopic simulators require fewer input datas; is suitable to model large networks and is computationally more efficient than microscopic simulators.

Mesoscopic simulators provide a tradeoff between macroscopic and microscopic modeling. It could provide better representation of individual vehicles compared to macroscopic models [6].

The development of traffic models for traffic simulation is still ongoing [19-22]. Recently, the authors proposed a mesoscopic model namely LWR-IM (Lighthill-Witham-Richards - Integrated Model) which is suitable for the modeling of urban traffic network (UTN) and arterial routes [23, 24]. The LWR-IM which integrates the macroscopic attributes of the Lighthill-Witham-Richards [25, 26] and the microscopic attributes of the the Rakha vehicle dynamics model [27] aims to provide modeling of UTN with efficient execution time and simpler analyzes of traffic performances such as queues and delays based on the input-output method.

As a subsequent work, this paper presents the design and development of a traffic simulator namely *UTNSim* that is based on the LWR-IM mesoscopic model. The traffic simulator enables description of the traffic dynamics and control in an UTN. Simulation execution provides estimate for traffic performances such as delays and queues. The simulator is used to specify the real traffic situation and its ability to simulate queues is validated with real-data observed. Statistical analysis and validation is conducted using the two-sampled t-test and mean absolute error (MAE) [28].

The next section outlines briefly the underlying theory of the LWR-IM; the structure and algorithm of the traffic simulator and the simulation and validation methods are presented in Section 3. Simulation results and discussion are shown in Section 4. Subsequently, the paper concludes in Section 5.

## 2. The LWR-IM Traffic Model

The authors proposed the LWR-IM in [23, 24]. The LWR-IM model which is shown in Fig. 1 integrates the Lighthill-Witham-Richards (LWR) model [25, 26] and the Rakha vehicle dynamics model [27] to describe platoon dynamics in UTN.

The LWR was described using variable continuous timed Petri net with variable speeds (VCPN) which represent the movement of vehicles in an arterial. The VCPN describes the arterial of a certain length  $Y$  (see Fig. 1) which was divided into different segments. Each segment of the arterial has varying speed  $S(t)$ , flow  $q(t)$  and density  $\rho(t)$ . On the other hand, the Rakha model was integrated to provide useful estimates of the trajectory of the first vehicle such as its behaviour of speed versus time and distance versus time as shown in Fig. 1. These predictions provide useful estimates such as free flow speed for each segment of the arterial as well as the platoon arrival time at the downstream stop line. Platoon dynamics in the form of platoon arrival curve is simulated at the downstream of the arterial. Hence, the integrated model provides a mesoscopic representation of the platoon dynamics. One of the advantages of this method is that platoon characteristics such as the arrival time of the first and last vehicle and the platoon length could be estimated without analyzing the behaviours of individual vehicles which was usually performed in microscopic models. Therefore, this resulted in lesser computational effort. A more comprehensive explanation of the LWR-IM can be found in [23, 24].

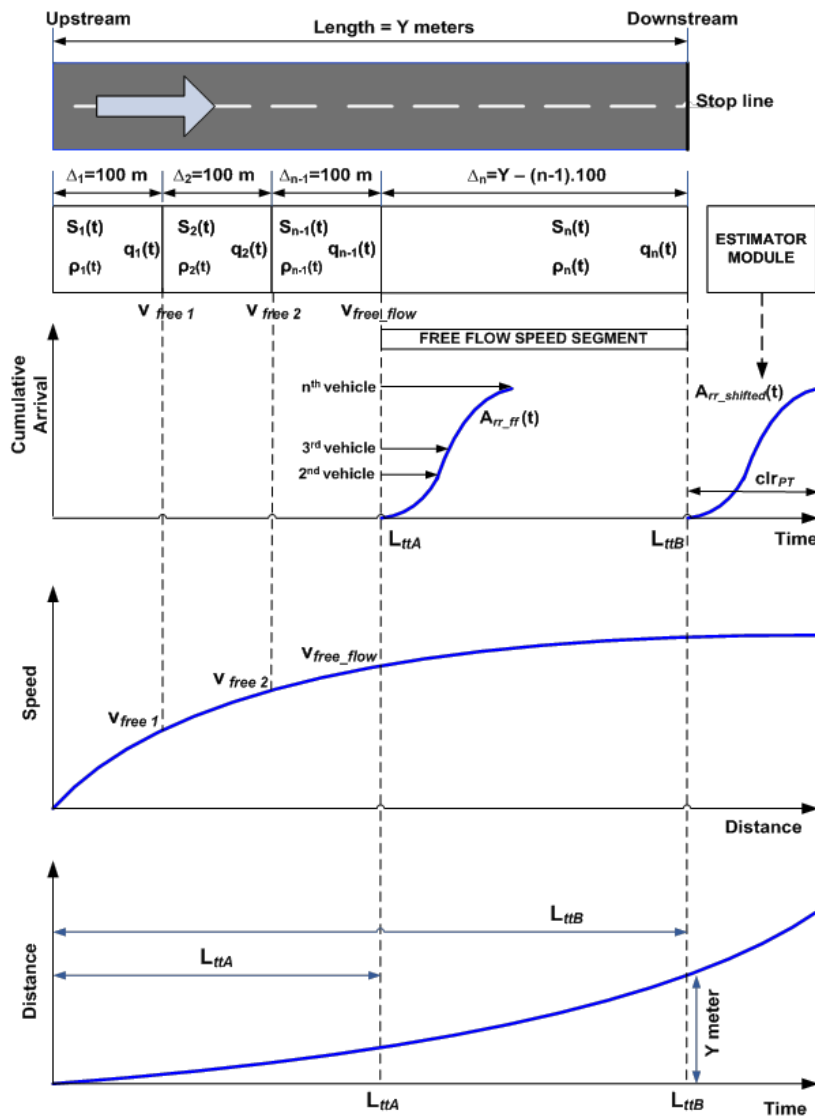


Fig. 1. Lighthill-Witham-Richards - Integrated Model (LWR-IM).

In addition, the arrival curve could be analyzed using the input-output method. This is less tedious compared to analyzing the LWR using Shockwave analysis [29]. Figure 2 shows the input-output analysis method. A typical arrival profile  $A(t)$  at downstream stop line obtained from the LWR-IM is shown in Fig. 2(a). The departure profile  $D(t)$  can be plotted by the estimator (in Fig. 1) by utilizing the phase change data, e.g. the offset between upstream and downstream green, the saturation flow as well as the fixed-time green period and cycle time of the downstream signal. Hence, the departure profile undertakes a linear behaviour as shown in Figs. 2(a) and (b). The residue queue  $Q_r$  is obtained at the ending period of the green signal.  $Q_s$  denotes the queue at begin green whereas  $Q_m$  depicts the position of the back of the queue at its

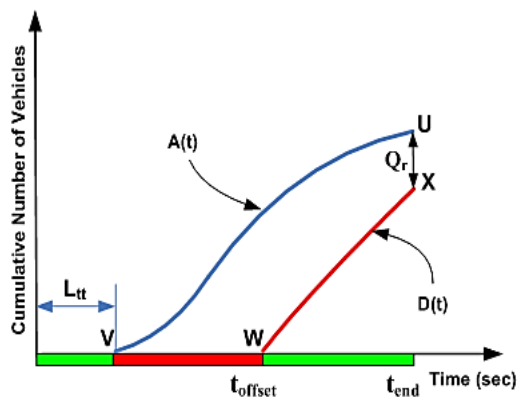
peak during the cycle. Figure 2(a) shows analysis of arrival profile without an initial queue at the downstream stop line. However, as traffic performance is evaluated from a cycle to cycle basis,  $Q_r$  from the previous cycle is considered as initial queue  $Q_i$  for analysis in the subsequent cycle. This is shown in Fig. 2(b) where the subsequent arrival profile  $A(t)$  joined the initial queue  $Q_i$ .

Equations (1) and (2) are used to calculate  $Q_s$  and  $Q_r$  respectively. The  $t_{offset}$  is offset time between signals and  $t_{end}$  is the ending time of the approach intersection green period. The enclosed area between the arrival and departure profile i.e. UVWX and UVWXY in Figs. 2(a) and 2(b) respectively is the total delay encountered by the stopping vehicles. Based on the total delay, the average delay experienced by each vehicle is given by Eq. (3).

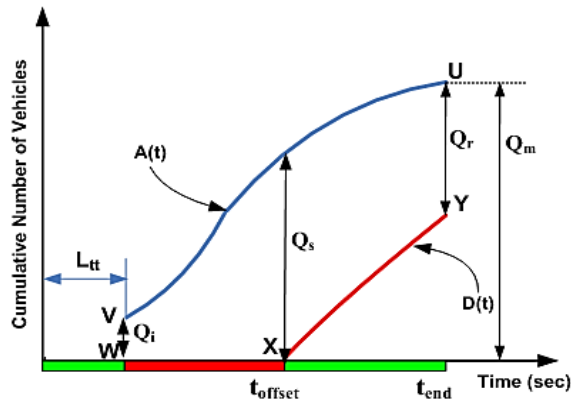
$$Q_s = A(t_{offset}) - D(t_{offset}) \tag{1}$$

$$Q_r = A(t_{end}) - D(t_{end}) \tag{2}$$

$$\text{Average delay} = \frac{\text{Total delay}}{\text{Total number of vehicles}} \tag{3}$$



(a) Analysis without initial queue



(b) Analysis with an initial queue  $Q_i$

Fig. 2. Input-output analytical techniques.

### 3. Method and Experimental Procedure

#### 3.1. The structure of the UTNSim

The structure of the *UTNSim* traffic simulator is illustrated in Fig. 3. It comprises of a traffic network diagram editor (TNDE) and its underlying data structures which is developed using the Java language. The diagram editor enables construction and specification of an urban traffic network (UTN). The underlying data structures, i.e., the directed graph stores the attributes and specific information related to the traffic network. In addition, the TNDE also constitutes a Java-MATLAB interface that facilitates an interface between the TNDE and MATLAB. Implementation and performance simulation of the UTN are executed in the MATLAB environment. The MATLAB module invokes the TNDE and accesses data structures in the TNDE via the Java-MATLAB interface. It performs performance evaluation on the network and simulates performance of the traffic intersection such as queues and average delays.

The TNDE enables visual construction and specification of a UTN. Some classifications are given to the UTN nodes and links. Figure 4 shows the different node and link types. An entry node comprises of an outgoing link known as the entry link. On the other hand, an exit node constitutes an incoming link known as the exit link. A signalized node connects both incoming and outgoing links. A link that connects two signalized nodes is classified as an intermediate link.

The directed graph data structure in the TNDE captures the attributes of the UTN. Arcs of the directed graph represent roads or links in the UTN and nodes represent entry/exit points or signalized intersections of the UTN. Figure 5 shows the attributes in these objects. For instance, an arc object contains the arc type, name of the arterial or link, arterial or link length, free flow speed of the link, entry volume, number of lanes, saturation flow and the initial queue or occupancy of the link as well as its source and destination nodes. A node contains the node number and node type. If it is a signalized node, it contains a set of node-triplets and their associated starting time, green periods and turning rates. Entry and exit nodes are only associated with a node number and node type whereas signalized nodes are further detailed with the related node-triplets set.

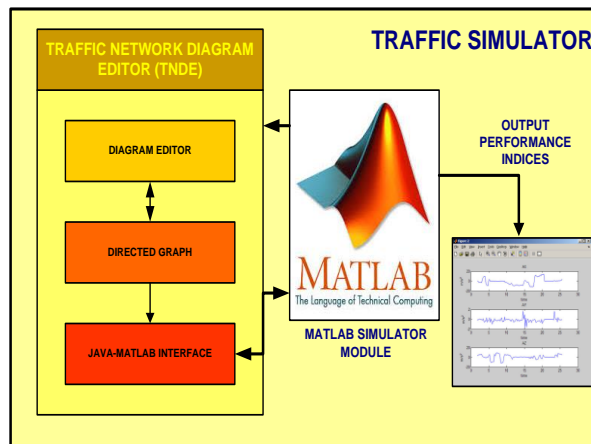


Fig. 3. Block diagram of *UTNSim*.

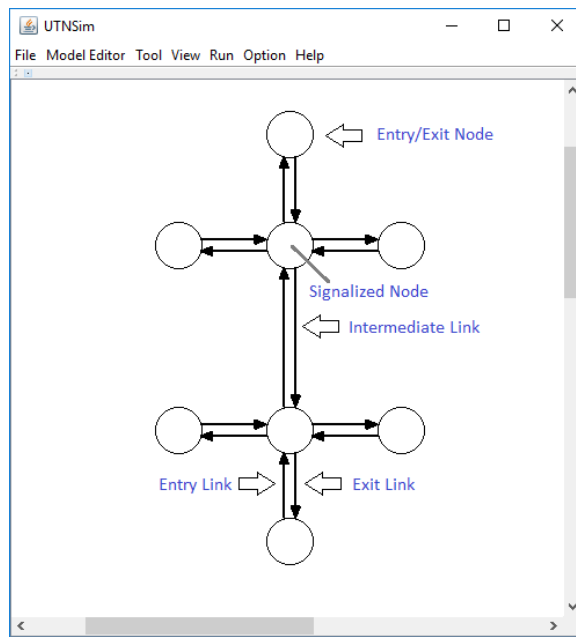


Fig. 4. A typical UTN developed in *UTNSim*.

<u>ARC</u>	<u>NODE</u>
<ul style="list-style-type: none"> <li>• Name</li> <li>• Type</li> <li>• Link length</li> <li>• Free flow speed</li> <li>• Entry Volume</li> <li>• Saturation flow rate</li> <li>• Initial queue</li> <li>• Source and destination node</li> </ul>	<ul style="list-style-type: none"> <li>• Node number</li> <li>• Type</li> <li>• Node-triplets list and their associated starting time, green period and turning rates</li> </ul>

Fig. 5. Attributes of an arc and node objects.

In this research, a class I real-world fixed-time traffic system in the city of Shah Alam, Malaysia was employed for the simulation studies. The two-intersection traffic system is shown in Fig. 6.

Persiaran Kuala Selangor (class I arterial) connects both intersections. The length of Persiaran Kuala Selangor is approximately 890 m. Traffic dynamics along Persiaran Kuala Selangor will be simulated as vehicles moved from the intersection I towards intersection II. Figure 6 also shows the fixed-time signal phases at both intersections which operates in the sequence of phase A ( $\phi_A$ ), followed by phase B ( $\phi_B$ ), then phase C ( $\phi_C$ ) and finally phase D ( $\phi_D$ ). The green periods of these signal phases are shown in Table 1. Intersection I assumes a cycle time of 260 s whereas the cycle time for intersection II is 340 s.

This real-world fixed-time system will be used to illustrate the functionalities of the TNDE. Figure 7 shows the UTN representation of the real-world traffic in Fig. 6. Node 2 represents the signalized intersection II whereas node 1 represents signalized intersection I. Node 3, 4, 5, 6, 7 and 8 are entry/exit nodes. For instance, nodes 7 and 8 depict entry and exit points of vehicles entering Persiaran Kuala Selangor from Lebuhraya Shah Alam or leaving Persiaran Kuala Selangor into Lebuhraya Shah Alam. The list of nodes and arcs in the UTN are listed in Table 2.

Intermediate arcs (1,2) and (2,1) describe and specify the traffic along Persiaran Kuala Selangor that connects both intersections. These arcs which hold attributes listed in Fig. 5 can be updated using the *Link Update* facility situated in the *Model Editor* menu. On the other hand, the *Orientation* sub-menu in the *Model Editor* specifies the route passages at the signalized nodes, e.g. the route of vehicles entering and leaving a particular signalized node. For example, in the signalized intersection shown in Fig. 8; vehicles may enter from the west (W) side and move towards the east (E) side of the signalized intersection. Thus, this route is defined as west-east (WE). Vehicles may enter from the south (S) side and move towards the north (N); this route is notated as south-north (SN). To specify these routes at a signalized node, one can simply select the signalized node that needs to be updated and subsequently invoke the *Signal Control* dialog in the *Model Editor* menu.

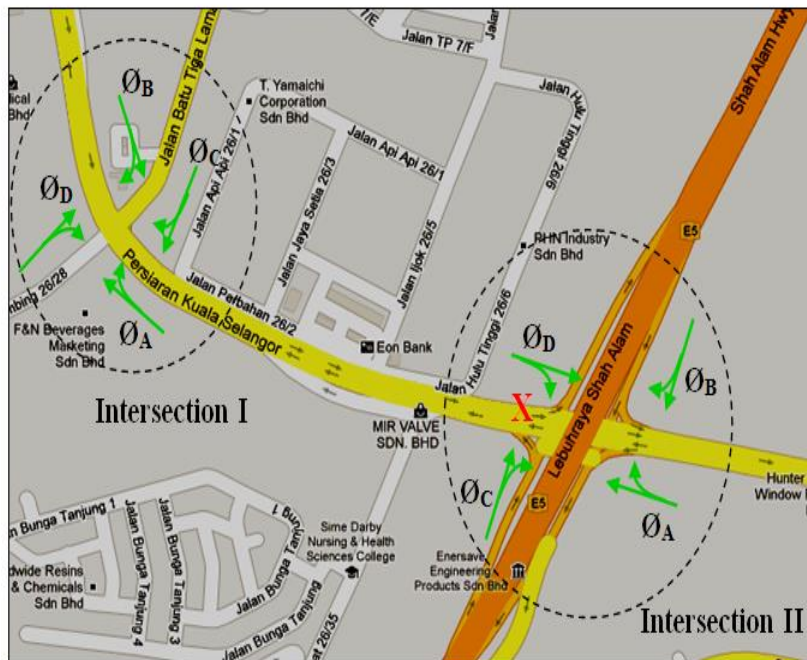


Fig. 6. Real-world fixed-time traffic systems in Shah Alam, Malaysia [30].

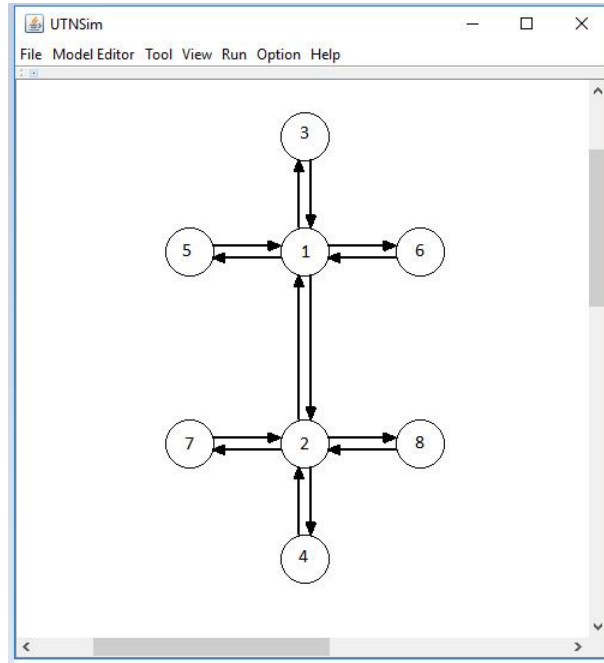
Table 1. Green time periods for Intersection I and II.

Intersection	$\varnothing_A$	$\varnothing_B$	$\varnothing_C$	$\varnothing_D$
I	90 s	95 s	35 s	40 s
II	65 s	60 s	155 s	60 s

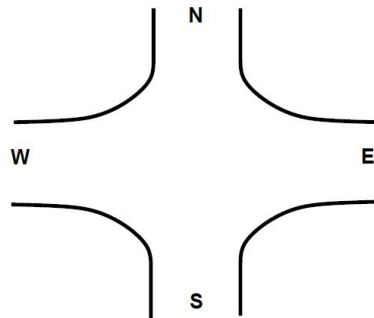


**Table 2. List of nodes and arcs.**

Type	Nodes and Arcs
Entry/exit nodes	3, 4, 5, 6, 7 and 8
Signalized nodes	1 and 2
Entry links	(4,2) (3,1) (5,1) (6,1) (7,2) (8,2)
Exit links	(2,4) (1,3) (2,7) (1,5) (1,6) (2,8)
Intermediate links	(2,1) and (1,2)



**Fig. 7. UTN model constructed in the UTNSim.**



**Fig. 8. Vehicle routes at a signalized intersection.**

Whenever a route is specified, a node-triplet object is created and stored in the node-triplets list within a signalized node object. Each node-triplet object can be further associated with the signal phase that regulates it. It is also described with

the start time (e.g. the time the signal phase turns green to allow vehicle movements), the duration of the green period and the turning rates of vehicles towards downstream links. As an example, Table 3 shows a list of the node-triplet objects that depict vehicle routes at signalized node 1 (Intersection I). These node-triplets objects are notated as SN, SE, NS, NW, EW, EN, WE and WS. In Table 3, node-triplet (2, 1, 3) depicts a route where vehicles move from node 2, through node 1 and exiting at node 3. It is notated as SN as vehicles move from the south side of signalized intersection I (node 1) towards its north side (node 3). Thus, the other node-triplets could be similarly explained.

Table 3 also shows that each node-triplet is associated with a signal phase and the start time values of these signal phases. As shown in Table 1, intersection I is controlled by a four phase signal stage namely  $\phi_A$ ,  $\phi_B$ ,  $\phi_C$ , and  $\phi_D$ . Each node-triplet (or vehicle route) is regulated by these phases and their respective starting time as shown in Table 3. The start time of a particular phase is the relative time to the system clock starting at  $T = 0$  when a simulation is executed. In addition, each node-triplet is also associated with a turning rate located at the far-right column in Table 3. Turning rate depicts the percentages of the vehicles going straight, turning right or turning left. The signalized node 2 (Intersection II) is also oriented and specified in a similar manner.

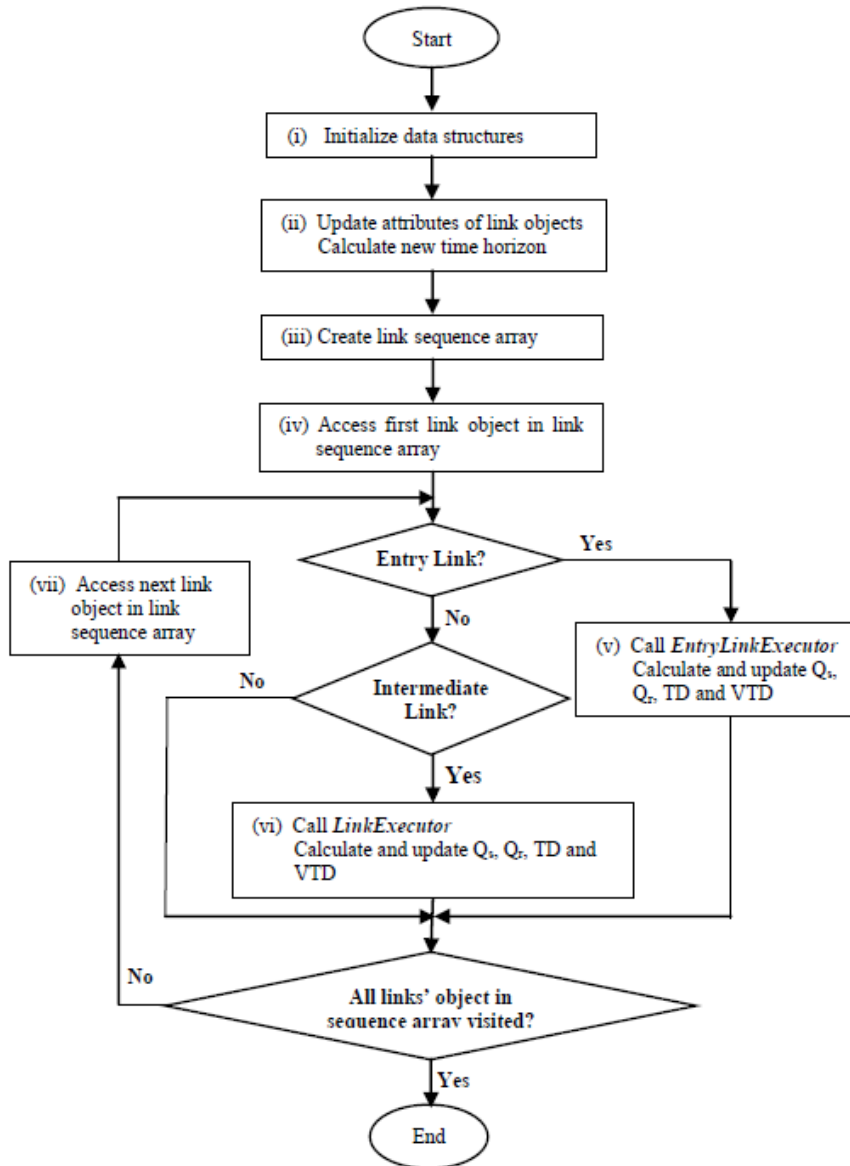
**Table 3. List of node-triplets and associated parameters.**

Node-triplet	Notation	Signal Phase	Start Time (sec)	Duration (sec)	Turning rate (%)
(2, 1, 3)	SN	$\phi_A$	0	90	95
(2, 1, 6)	SE	$\phi_A$	0	90	5
(3, 1, 2)	NS	$\phi_B$	90	95	95
(3, 1, 5)	NW	$\phi_B$	90	95	5
(6, 1, 5)	EW	$\phi_C$	185	35	10
(6, 1, 3)	EN	$\phi_C$	185	35	90
(5, 1, 6)	WE	$\phi_D$	220	40	10
(5, 1, 2)	WS	$\phi_D$	220	40	90

The above functions in the TNDE provide sufficient facilities to specify a UTN. This is not only limited to a two-intersection UTN but a wider area of UTN that comprises of many intersections. Upon complete construction and specification of a UTN, the user should compile the diagram by selecting the *Run* menu. This action will render all node-triplet objects and link objects of the UTN into two separate array list in the Java-MATLAB interface. Hence, the information of the UTN is made accessible to the MATLAB environment.

### 3.2. Algorithm of *UTNSim*

The algorithm of the MATLAB module of the *UTNSim* shown in Fig. 3 that implements performance simulation and evaluation on a particular UTN is further described by the flow chart in Fig. 9. The flow chart generalizes the algorithm functions of the MATLAB. The Java-MATLAB interface that contains all the node-triplets and link objects in the TNDE is accessed by the MATLAB module to perform the performance evaluation on the UTN which produces the estimated delays and queuing performance in the UTN.



**Fig. 9. MATLAB simulator module.**

The following is the step-by-step description of the flow chart in Fig. 9:

- (i) The module accesses Java objects such as node-triplets and link objects in the Java-MATLAB interface to initialize the MATLAB data structures for the UTN. The data structures include MATLAB link objects and sparse matrices that store attributes of the entire network.
- (ii) The MATLAB update attributes of its own link objects which replicate the information of the link objects in the TNDE. Subsequently, a time horizon is calculated by the module.

- (iii) Based on the time horizon, the MATLAB module analyzes the node-triplets and its associated parameters such as the phase start time to create a link sequence array that depicts the sequence of links execution according to their respective phase start time.
- (iv) A typical execution of the UTN starts by accessing the first link object in the link sequence array.
- (v) If the link object is an entry link, the function *EntryLinkExecutor* is invoked to calculate and update queue at begin green  $Q_s$ , residue queue  $Q_r$ , total delay (TD) and vehicles to downstream (VTD) released to its destination links based on the demands and timing plan associated with this entry link object.
- (vi) If the link object is an intermediate link, the function *LinkExecutor* is invoked to calculate and update  $Q_s$ ,  $Q_r$ , TD and VTD released to its destination links based on the demands and timing plan associated to this intermediate link object.
- (vii) If there are link objects in the link sequence array not visited, the algorithm searches for the next link object in the array. Steps (v) and (vi) are repeated until all link objects are visited.

The *LinkExecutor* mentioned in the flow chart in Fig. 9 was developed in [23] to predict queues and delays in intermediate links based on a set of rules. These set of decision rules describe the dynamics and arrival of platoons based on the LWR-IM which has been previously published in Ng and Reaz [23]. Subsequently, the functions in the *LinkExecutor* implements input-output analysis which was previously outlined in Eqs. (1), (2) and (3) to determine respectively the queues at begin green  $Q_s$ , residue queues  $Q_r$ , total delays (TD) and the number of vehicles released to downstream links (VTD). On the other hand, *EntryLinkExecutor* estimates queues and delay at entry links based on uniform arrivals, e.g., the Webster model [2].

The algorithm in Fig. 9 can be further explained by a simple UTN that comprises of two intersections in Fig. 10. Both intersections are signalized nodes 2 and 3 which are connected to each other via intermediate links and connected to other entry and exit nodes via the respective entry and exit links. Each entry and intermediate links are associated with a starting time that represents the starting time of the signal phase that regulates traffic at the link. The link sequence array mentioned in Fig. 9 represents the order of links implementation. As mentioned previously, it is sorted according to the phase start time of a particular link. By iterating through the link sequence array; each link is visited and the respective queues, delays and vehicles released to its downstream links are determined. Evaluation of entry links directly utilize the entry volume  $V_{s,d}$  and an existing queue  $Iq_{s,d}$  whereas intermediate links contain existing queue  $Iq_{s,d}$  and a series of vehicle entry volumes (i.e.,  $V_{k,s,d}$ ,  $V_{k+1,s,d}$ ,  $V_{k+2,s,d}$  .....) released from its upstream counterpart with their respective upstream start time (i.e.  $US_{k,s,d}$ ,  $US_{k+1,s,d}$ ,  $US_{k+2,s,d}$  .....). The subscript  $k$  represents the index sequence of the entry volume  $V$  and the upstream start time  $US$ ;  $s$  is the source node and  $d$  denotes the destination node of a particular arc (link).

For instance, entry link (1,2) in Fig. 10 holds an initial queue  $Iq_{1,2}$  and an entry volume  $V_{1,2}$ . The *EntryLinkExecutor* function utilizes these inputs to estimate queues and delay at entry links based on uniform arrivals [2]. The intermediate link (2,3) holds  $Iq_{2,3}$  and receives entry volume  $V_{1,2,3}$  and  $V_{2,2,3}$  which was released

from upstream at time  $US_{1,2,3}$  and  $US_{2,2,3}$  respectively. On the other hand, link (3,2) with  $Iq_{3,2}$  receives  $V_{1,3,2}$ ,  $V_{2,3,2}$  and  $V_{3,3,2}$  which was released from upstream at time  $US_{1,3,2}$ ,  $US_{2,3,2}$  and  $US_{3,3,2}$  respectively. The *LinkExecutor* function utilizes these entry volumes and their upstream start times as well as the link existing queues to determine at each intermediate link, the residue queues, delays as well as the number of vehicles released to their downstream links based on the LWR-IM arrival profile and the input-output method explained previously. Upon accessing all elements in the link sequence array, all the links would have been completely iterated. Hence, each link object contains a series of queue and delay values which are simulated by the MATLAB module.

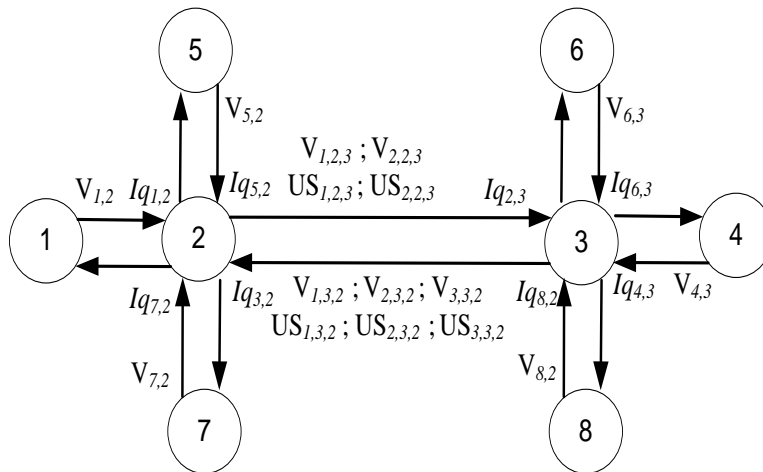


Fig. 10. A simple UTN.

### 3.3. Simulation and validation setup

Real traffic inputs moving towards intersection II from intersection I was recorded via video recording over a period of one week (from 23 November to 29 November 2015). Traffic demands were recorded during peak hours i.e. 8.00 to 9.00 am, 12.00 to 1.00 p.m. and 6.00 to 7.00 p.m. for the week days as well as weekends. Traffic situations recorded during peak hours will be able to encapsulate both under-saturated and oversaturated situations. In addition, traffic demands during week days could be used to validate the simulator for heavy traffics whereas traffic demands during the weekends could be used to validate the ability of the simulator to simulate moderate and lower traffic demands. The number of vehicles entering each lane of Persiaran Kuala Selangor from upstream intersection I is observed for 14 consecutive signal timing cycles of Intersection II (approximately an hour). As Persiaran Kuala Selangor consists of 3 lanes; the volume of vehicles per lane per hour is obtained by dividing the total volume by 3. The queues per lane such as the queue at begin of green  $Q_s$  and the residue queue  $Q_r$  at downstream location X are also observed and tabulated. Altogether, there are 21 sets of hourly traffic volumes (i.e. 8.00 to 9.00 a.m., 12.00 to 1.00 p.m. and 6.00 to 7.00 pm) collected from 23 November to 29 November 2015. Hence, altogether 21 sets of real traffic data are used to simulate 21 sets of  $Q_s$  and  $Q_r$  at Intersection II.

Next, simulated queues are validated with real queues observed at Intersection II. A two-sample t-test is employed to evaluate any significant differences between simulated and real queues. The procedure of the t-test is as follows:

- Step 1: A standard level of significance (LOS),  $LOS = 0.05$  is chosen for this test.
- Step 2: Perform the simulations and tabulates simulated queue in each cycle of the simulation. Calculate the mean absolute errors (MAEs) between simulated queues and real queues observed.
- Step 3: Conduct a two-sample t-test between simulated and real queues. Compute the p-value to ascertain the type of hypothesis achieved by the model.
- Step 4: If the p-value is more than the pre-defined LOS, the null hypothesis is accepted. This indicates that the model is valid as there is no significant difference between simulated and real indices (i.e. queues). A p-value lesser than the pre-defined LOS suggest an alternative hypothesis which indicates an invalid model.
- Step 5: The model can be concluded as valid for estimating queues if the null hypothesis is accepted. Otherwise, the model is invalid for that purpose. In addition to the t-test, the mean absolute error (MAE) is also calculated for each data set to further validate the findings obtained from the t-test.

#### 4. Results and Discussion

Performance criteria such as queue at begin of green  $Q_s$  and residue queue  $Q_r$  are simulated for location X (the downstream stop line) of intersection II of Fig. 6. The vehicles stopped at X during a red phase before turning right or going straight. Real traffic inputs observed at the intersections over a span of one week are used as inputs to perform the simulations. Subsequently, simulated queues are validated statistically with real data observed at the intersections to study the accuracy of the *UTNSim* in simulating queues that resemble the actual traffic situation.

Simulations were conducted on 21 sets of real data recorded via video recording over a period of one week (from 23 November to 29 November 2015) at Intersection I and Intersection II of Persiaran Kuala Selangor. For brevity, data collected from 8.00 a.m. to 9.00 a.m. on 24 November 2015 (Day 2) will be used to illustrate the in-depth analysis of one data set as shown in Table 4. Subsequently, the results for all the 21 data sets will be summarized in Tables 5 and 6.

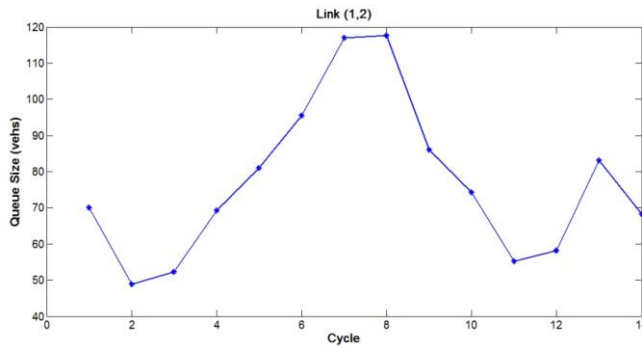
As an example, the input traffic volumes between 8.00 a.m. to 9.00 a.m. on 24 November 2015 (refer to Table 4) was fed into the UTN. Vehicles in Platoon A were released from  $\emptyset_B$  of Intersection I whereas Platoon B vehicles originated from upstream  $\emptyset_D$ . These volumes were specified into node 3 and node 5 respectively in Fig. 7.

The *UTNSim* implements the UTN based on the LWR-IM and analyse the arrival profiles at location X using the input-output analysis. Queue at begin of green  $Q_s$  and residue queue  $Q_r$  were simulated for Intersection II over a span of 14 signal cycles (approximately 1 hour). The simulated values of  $Q_s$  and  $Q_r$  could be easily obtained from the simulator. Figures 11(a) and (b) show the graph plot of both  $Q_s$  and  $Q_r$  respectively at location X for each cycle of implementation. These simulated values

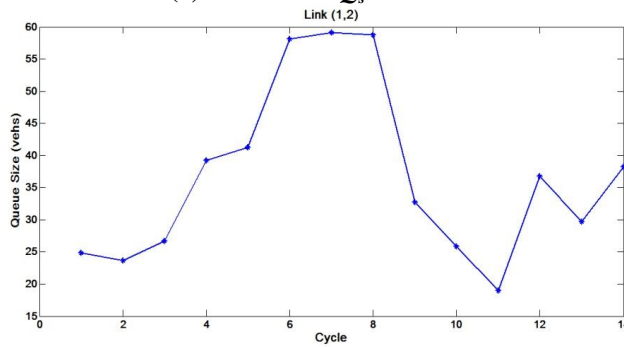
were compared with real values observed as shown in Table 5. Subsequently simulated  $Q_s$  and  $Q_r$  in Table 5 were analysed using the t-test and MAE.

**Table 4. Traffic volumes between 8.00 to 9.00 a.m. on 24 November 2015 at location X.**

Cycle	Platoon A (veh)	Platoon B (veh)
1	28	2
2	27	4
3	28	5
4	30	14
5	28	5
6	29	11
7	33	5
8	26	5
9	24	7
10	27	5
11	41	5
12	40	11
13	45	6
14	32	5
<b>Total</b>	<b>438</b>	<b>90</b>



(a) Simulated  $Q_s$  at location X.



(b) Simulated  $Q_r$  at location X.

**Fig. 11. Simulated  $Q_s$  and  $Q_r$  at location X.**

Table 5 shows the tabulation of real and simulated  $Q_s$  and  $Q_r$  over a span of 14 signal cycles at intersection II based on traffic data collected on 24 November 2015 (8.00 a.m. to 9.00 a.m.). Simulated  $Q_s$  at cycle 4 and 5 closely agrees with an absolute difference of less than 1 vehicle. Simulated  $Q_s$  are higher than real  $Q_s$  at cycle 2, 3, 6, 7, 8, 10, 11 and 14. Simulated  $Q_s$  are lower than real data at cycle 1, 9, 12 and 13. The highest discrepancy between observed and simulated queues is observed at cycle 7 whereas the lowest discrepancy is 0.03 vehicle which is observed at cycle 5. The results of  $Q_r$  are also shown in Table 5. The highest discrepancy of 13.16 vehicles is detected at cycle 6 whereas the lowest discrepancy of 1.57 vehicles is observed at cycle 5.

**Table 5. Simulated and Real Queues of at location X of Persiaran Kuala Selangor on 24 November 2015 (8.00 - 9.00 AM).**

Cycle	Queue at begin of green $Q_s$ (veh)			Residue queue $Q_r$ (veh)		
	Real	Simulated	Absolute Difference	Real	Simulated	Absolute Difference
1	76.83	70.00	6.83	28.00	24.88	3.12
2	43.17	48.88	5.71	25.00	23.68	1.32
3	52.00	55.28	3.28	22.67	26.66	3.99
4	68.67	69.26	0.59	41.83	39.26	2.57
5	81.00	81.03	0.03	42.83	41.26	1.57
6	87.33	95.43	8.10	45.00	58.16	13.16
7	105.67	117.00	11.33	62.67	59.16	3.51
8	110.00	117.60	7.60	63.33	58.75	4.58
9	88.17	86.00	2.17	29.33	32.75	3.42
10	70.67	74.23	3.56	23.67	25.86	2.19
11	53.17	55.16	1.99	20.67	18.96	1.71
12	60.67	58.23	2.44	39.63	36.75	2.88
13	84.62	83.16	1.46	27.33	29.75	2.42
14	66.55	68.23	1.68	31.55	38.23	6.68

The real and simulated data in Table 5 is further analyzed statistically using the two-sampled t-test and the mean absolute error (MAE). Table 6 shows the result of the two-sampled t-test performed on both real and simulated  $Q_s$  and  $Q_r$  in Table 5. The p-values of 0.78 and 0.88 for  $Q_s$  and  $Q_r$  respectively indicate that the null hypothesis can be accepted as these p-values are more than  $LOS = 0.05$ . This indicates that simulated and real  $Q_s$  and  $Q_r$  are not significantly different at  $LOS = 0.05$ . Thus, the acceptance of the null hypothesis concluded by the two-sampled t-test reveal the ability of the *UTNSim* to simulate queues that closely resemble real situation. This close resemblance is further supported by low MAE between real and simulated data. The MAE for real and simulated  $Q_s$  and  $Q_r$  are 4.06 and 3.79 vehicles respectively as shown in Table 6.

**Table 6. Results of t-test and mean absolute error (MAE) for data in Table 5.**

Type of queue	Statistical Analysis	
	t-test (p-value)	MAE (veh)
Queue at begin green $Q_s$	0.78	4.06
Residue queue $Q_r$	0.88	3.79



Similar statistical analysis was conducted for the other 20 sets of data. Table 7 shows the results of the two-sampled t-test and MAEs on all 21 simulated and real data of  $Q_s$  at Intersection II whereas Table 8 shows the results for  $Q_r$ . The traffic demands for morning peak hours during weekdays (Day 1 to Day 5) ranged from 412 to 528 veh/h/ln. Traffic demands observed between 1.00 p.m. to 2.00 p.m. during the weekdays ranged from 393 to 571 veh/h/ln. Traffic demands in the evening peak hours ranged from 423 to 664 veh/h/ln. Traffic demands during weekends (Day 6 and Day 7) were generally lesser compared to traffic demands during the weekdays. This could be observed during the morning and afternoon peak hours with demands lesser than 300 veh/h/ln.

**Table 7. Results of two-sampled t-test and MAE of  $Q_s$  at location X of Persiaran Kuala Selangor.**

Day	8.00 to 9.00 a.m.			1.00 to 2.00 p.m.			6.00 to 7.00 p.m.		
	Demand (veh/h/ln)	t-test	MAE (veh)	Demand (veh/h/ln)	t-test	MAE (veh)	Demand (veh/h/ln)	t-test	MAE (veh)
1	417	0.65	5.71	393	0.48	1.35	556	0.98	4.58
2	528	0.76	4.65	537	0.68	1.89	664	0.76	3.96
3	412	0.67	5.90	491	0.78	2.59	591	0.91	2.76
4	425	0.76	4.20	541	0.68	3.42	657	0.84	3.36
5	447	0.86	5.22	571	0.79	2.02	425	0.39	3.54
6	300	0.76	2.66	279	0.39	2.66	459	0.70	2.68
7	228	0.68	2.82	283	0.49	3.21	423	0.74	3.75

**Table 8. Results of two-sampled t-test and MAE of  $Q_r$  at location X of Persiaran Kuala Selangor.**

Day	8.00 to 9.00 a.m.			1.00 to 2.00 p.m.			6.00 to 7.00 p.m.		
	Demand (veh/h/ln)	t-test	MAE (veh)	Demand (veh/h/ln)	t-test	MAE (veh)	Demand (veh/h/ln)	t-test	MAE (veh)
1	417	0.84	4.55	393	0.36	0.97	556	0.95	3.56
2	668	0.95	3.74	537	0.38	2.59	664	0.96	2.10
3	412	0.95	3.28	491	0.27	0.70	591	0.96	2.85
4	425	0.90	3.67	541	0.23	1.64	657	0.58	2.00
5	447	0.92	2.89	571	0.43	1.77	425	0.41	2.86
6	300	0.94	1.10	279	0.66	2.17	459	0.16	2.06
7	228	0.57	1.20	283	0.77	0.67	423	0.87	1.67

All p-values of the two-sampled t-test conducted on the 21 sets of simulated and real  $Q_s$  and  $Q_r$  in Tables 7 and 8 are more than the LOS = 0.05. Hence, the null hypothesis cannot be rejected. Therefore, it can be concluded that simulated results are not significantly different from real data. The MAEs for real and simulated  $Q_s$  ranges from 1.35 to 5.90 vehicles. On the other hand, MAEs for  $Q_r$  ranges from 0.67 to 4.55 vehicles. These results show that at every signal cycle at location X of Intersection II; the difference between real and simulated queues is less than 5.90 vehicles for  $Q_s$  and less than 4.55 vehicles for  $Q_r$ . Thus, the results from the t-test and MAE have ascertained the validity of the *UTNSim* in simulating performance indices that could resemble the real-world.

The application of t-test was implemented previously in [31] to compare mean headways and mean speed from field data using microscopic simulation. In [32], t-test was applied to compare vehicle travel time. However, there were instances where the compared data failed the t-test and model calibration was recommended

[32]. Therefore, as the t-test for all the 21 sets of simulated and real data for both  $Q_s$  and  $Q_r$  were above the LOS; the LWR-IM was indeed a reliable and valid model. This was further supported by previous studies where the LWR-IM was proven to be closely fitted to the Highway Capacity Manual 2000, the cell transmission model and the platoon dispersion model [23, 33].

## 5. Conclusions

The structure and functions of the *UTNSim* have been briefly introduced and concisely explained by the authors. As a preliminary evaluation of the *UTNSim*, this paper also aims to provide a validation study on the *UTNSim* using a real traffic situation in the city of Shah Alam, Malaysia. The preliminary validation experiment conducted using this real world traffic system has proven the validity of the *UTNSim* in simulating queues that resemble real situation. This was proven when the t-test conducted on all real and simulated data sets concludes that the null hypothesis cannot be rejected. Hence, simulated and real queues were not significantly different from each other.

## Acknowledgement

The authors wish to express their appreciation to Ministry of Higher Education (MOHE) Malaysia and Universiti Teknologi MARA, Shah Alam, Selangor for funding this research under the Research Acculturation Grant Scheme (RAGS) (grant no.: RAGS/1/2015/TK0/UITM/03/1)

## References

1. Yaldi, G.; Nur, I.M.; and Momon, A. (2016). Defining suitable parking controls to minimize negative impacts of road traffic: a case study in Padang City. *International Journal on Advanced Science, Engineering and Information Technology*, 6(5), 600-606.
2. Roess, R.P.; Prassas, E.S.; and McShane, W.R. (2011). *Traffic Engineering* (4<sup>th</sup> Ed.). USA: Pearson.
3. Schutter, B.D.; Hellendoorn, H.; Hegyi, A.; Van den Berg, M; and Zegeye, S.K. (2010). Model-based control of intelligent traffic networks. *Intelligent Systems, Control and Automation: Science and Engineering*, 40(1), 277-310.
4. Basile, F.; Chiacchio, P; and Teta, D. (2012). A hybrid model for real-time simulation of urban traffic. *Control Engineering Practice*, 20, 123-137.
5. Lai, G. R.; Soh, A.C.; Sarkan, H.M.D.; Abdul Rahman, A.Z.; and Hassam, M.K. (2015). Controlling traffic flow in multilane-isolated intersection using ANFIS approach techniques. *Journal of Engineering Science and Technology*, 10(8), 1009-1034.
6. Dion, F.; Sivakumaran, K.; and Ban, X. (2012). Evaluation of traffic simulation model use in the development of Corridor System Management Plans (CSMPs). *Final Project Report*, California PATH.
7. Newell, G.F. (2002). *A simplified car-following theory: a lower order model*. Institute of Transportation Studies, University of California, Berkeley.
8. Nagel, K. (1993). Particle hopping models and traffic flow theory. In: *Physical Review*, E53, 4655-4672.

9. May, D.; and Leiman, L. (2005). *Freeway analysis manual: Parts 1 and 2*. California Partners for Advanced Transit and Highways, University of California, Berkeley.
10. University of Florida (2017). McTrans: Moving technology website. Retrieved May 1, 2017 from <http://mctrans.ce.ufl.edu/featured/transyt-7f/>
11. Aurora Software, Inc. (2014). Aurora Software website. Retrieved May 1, 2017 from <http://http://www.aurorasoftware.com/>
12. PTV Group (n.d.). PTV Vissim. Retrieved May 1, 2017 from <http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/>
13. SYSTRA Limited (2016). Paramics microsimulation. Retrieved May 1, 2017 from <http://www.sias.com/2013/sp/spparamicshome.htm>
14. MMXV Trafficware (n.d.). Trafficware website. Retrieved May 1, 2017 from <http://www.trafficware.com/>
15. Aimsun (n.d.). TSS-Transport simulation system website. Retrieved May 1, 2017 from <https://www.aimsun.com/>
16. Rakha, H (2014), INTEGRATION Rel. 2.40 for Windows - User's Guide, Volume I: Fundamental Model Features, M.Aerde and Associates, Ltd., Blacksburg, Virginia.
17. ITS Lab (n.d.). Intelligent transportation systems lab. MIT website. Retrieved May 1, 2017 <https://its.mit.edu/software/dynamit>
18. DynusT Lab (2017). DynusT website. Retrieved May 1, 2017 <http://dynust.net/>
19. Maranica, N.; and Boel, R. (2012). Platoon based model for urban traffic control. *2012 American Control Conference*, 6563-6568.
20. Nordiana, M.; Rosli, H. M.; Norhidayah, A.H.; Nor Zurairahetty, M.Y.; and Mariyana, A.A.K. (2014). Exploring the pattern of platoon dispersion caused by traffic signal. *Jurnal Teknologi*, 71(3), 7-13.
21. Akcelik, R. (2014). A new lane-based model for platoon patterns at closely-spaced signalized intersections. *26<sup>th</sup> ARRB Conference*, Sydney, Australia.
22. Akcelik, R. (2015). Modelling signal platoon patterns by approach lane use and movement class. *WIT Transactions on the Built Environment*, 146, 531-532.
23. Ng, K.M.; and Reaz, M.B.I. (2015). An integrated approach for platoon-based simulation and its feasibility assessment. *PLoS ONE*, 10(3), 1-25.
24. Ng, K.M.; and Reaz, M.B.I. (2016). Platoon interactions and real-world traffic simulation and validation based on the LWR-IM. *PLoS ONE*, 11(1), 1 -17.
25. Lighthill, M.J.; and Whitham, G.B. (1955). On kinematic waves I: Flood movement in long rivers. II: A theory of traffic flow on long crowded roads. *Proceedings Royal Society (London)*, 229A, 281-345.
26. Richards, P.I. (1956). Shock waves on the highway. *Operation Research*, 4, 42-51.
27. Rakha, H.; Lucic, I.; Demarchi, S.; Setti, J.; and Aerde, V.M. (2001). Vehicle dynamics model for predicting maximum truck acceleration Levels. *Journal of Transportation Engineering*, 127(5), 418-425.

28. Ni, D.; Leonard II, J.D.; Guin, A.; and Williams, B.M. (2004). Systematic approach for validating traffic simulation models. *Journal of the Transportation Research Board*, 1876, 20-31.
29. Ng, K.M.; and Reaz, M.B.I. (2015). A comparative study of the LWR-IM traffic model and Shockwave analysis. *2015 IEEE Conference on Systems, Process and Control (ICSPC 2015)*, 39-43.
30. Google Map (n.d.). Persiaran Kuala Selangor. Retrieved Sept 19, 2017 from <https://www.google.com.my/maps/place/Persiaran+Kuala+Selangor,+40400+Shah+Alam,+Selangor/@3.0378263,101.5514662,17z/data=!4m5!3m4!1s0x31cdb2cd5f55e77d:0x569c177fd025eaa3!8m2!3d3.0353139!4d101.5593948>
31. Rao, L.; and Goldsman, D. (1998). Development and application of a validation framework for traffic simulation models. *Proceedings of the 1998 Winter Simulation Conference*, 1079-1086.
32. Park, B.; and Schneeberger, J.D. (2006). Microscopic simulation model: Calibration and validation case study of VISSIM simulation model for a coordinated actuated signal system. *Transportation Research Record 1856*, 185-192.
33. Ng, K.M.; Reaz, M.B.I.; Ali, M.A.M.; and Razak, N.A. (2016). A comparative assessment of the LWR-IM model using linear regression. *2016 International Conference on Advances in Electrical, Electronic and System Engineering*, 626-630.