https://doi.org/10.55343/CogSust.65

Road traffic queue length estimation with artificial intelligence (AI) methods

Csanád Ferencz, Máté Zöldy

0000-0002-0177-1646, 0000-0003-1271-840X

Department of Automotive Technologies, Faculty of Transportation Engineering and Vehicle Engineering,
Budapest University of Technology and Economics
Budapest, Hungary
csanadferencz@edu.bme.hu

Abstract

Sustainable traffic monitoring has always been a significant problem for engineers, queue length being one of the most important metrics required for the performance assessment of signalised intersections. The present study's authors propose a novel approach to estimating cycle-by-cycle queue lengths at a given signalised intersection. Focusing on examining shock wave phenomena and the traffic model, this study first elucidates the definitions and assumptions it employs. Subsequently, it delves into creating the queuing model alongside utilising a machine-learning (ML) based Kalman Filter (KF) algorithm for estimation. The information in the output files is visualised on distinct graphs, along with the velocities at various time intervals derived from virtual simulations involving a queue of 12 vehicles. This graphical representation is a conclusive validation, demonstrating a strong correlation between the simulation and the estimation achieved through the KF approach. The method presented yielded dependable and resilient estimates for the simulated queue lengths, even in the presence of noisy measurements.

Keywords

autonomous vehicles, artificial intelligence, sustainable mobility, traffic simulation, road network modelling

1. Introduction

Traffic signals are the most essential components of urban traffic networks, determining whether passing traffic at a junction should go or stop (Lee et al., 2015). These signal transition operations may lead to periodic changes in the queuing process, thus affecting urban traffic flow reliability (Schrank et al., 2015). Hence, when considering signal control optimisation and evaluating performance at signalised intersections, the significance of an accurate and resilient queue length estimation method cannot be overstated (Goodall et al., 2013). Because supplementary performance metrics like stops, travel time, and vehicle delay can be inferred by leveraging queue length data, these performance indicators will serve as valuable tools for traffic management, thus improving the level of service for the entire urban traffic network (Wang et al., 2019).

Road traffic queue length estimation is a crucial measurement of traffic signal control at any given urban intersection (Cao and Zöldy, 2020). However, conventional queue length estimation methods mainly rely on fixed detectors and use point detector data (Viti et al., 2010). These methods often struggle to handle scenarios with long queues that extend beyond the location of the point sensor (Sharma et al., 2007). Additionally, their focus tends to be on average queue lengths rather than considering the spatial and temporal distributions of queues (Fuliang et al., 2017). However, recent efforts have been substantial in developing precise and dependable techniques for queue length estimation at signalised intersections (Wu and Liu, 2011). The current study introduces a novel approach for estimating cycle-by-cycle queue lengths at a specific signalised intersection. This approach is particularly suited for addressing traffic challenges on a regional scale, thereby contributing to the sustainable development of intelligent transportation systems.

Following a concise examination of existing literature concerning queue length estimation at signalised intersections, with a specific emphasis on scrutinising shock wave phenomena and traffic models, this study outlines the definitions and assumptions employed. Subsequently, the creation of the queuing model is detailed, and the methodology for queue length estimation is introduced. This involves a machine-learning (ML) approach, specifically employing the Kalman Filter (KF)

https://doi.org/10.55343/CogSust.65

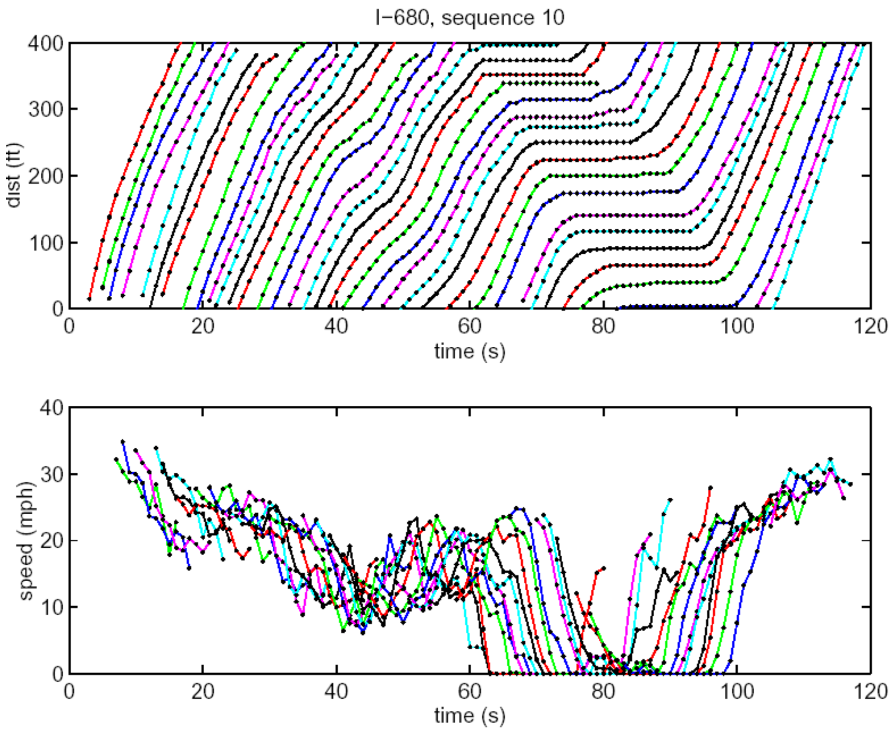
algorithm. Finally, the proposed method's result validation is also presented based on parameter estimation and traffic simulation.

2. Literature review

Real-time traffic control is vital in managing transportation, spanning urban and rural areas (Zöldy and Baranyi, 2021). Consequently, continuous measurement of every traffic parameter has become pivotal. Typically, estimation models are employed to skillfully reconstruct and foresee current and future traffic conditions based on collected information (Ferencz and Zöldy, 2021). However, the logistical and financial impracticality of placing sensors extensively across roadways renders it unfeasible (Wu and Yang, 2013). The number of measurement devices at road sections is limited to employ the minimal number of fixed sensors to curtail costs while maximising data collection (Yao and Tang, 2019). Nevertheless, this reduced sensor count yields fewer data points, amplifying uncertainty within the estimation model (Horváth and Tettamanti, 2021). Addressing this uncertainty effectively entails the application of robust methodologies, exemplified by the approach employed in this current study.

Traffic queue length estimation approaches for signalised intersections are commonly classified into two main categories: methods rooted in the input-output cumulative plot analysis and methods grounded in the principles of the shock wave theory (Liu et al., 2019). The former approach primarily infers queue lengths by examining the patterns of vehicle arrivals and departures at intersections. However, this study will investigate the latter type of estimation method in depth (Cetin, 2012).

In order to analyse shock wave phenomena, the so-called road traffic fundamental diagram theory can be used. Shock waves form in traffic (Tettamanti, 2021), when there is a sudden reduction or increase in the capacity of a roadway (Figure 1), for instance, when traffic stops at a red light (Varga and Tettamanti, 2023), in the case of an accident, or even lane reduction on a multi-lane road (Figure 2).



s

Figure 1 Shockwave profile model on freeway (Tettamanti, 2021)

https://doi.org/10.55343/CogSust.65

Applying the methods based on the shock wave theory, one can profile the stopping and discharging shockwaves based on detector data, consequently providing spatial and temporal information on the queues (Ibrahim et al., 2019). This data will encompass both the maximum and residual queue lengths (Cai et al., 2014).

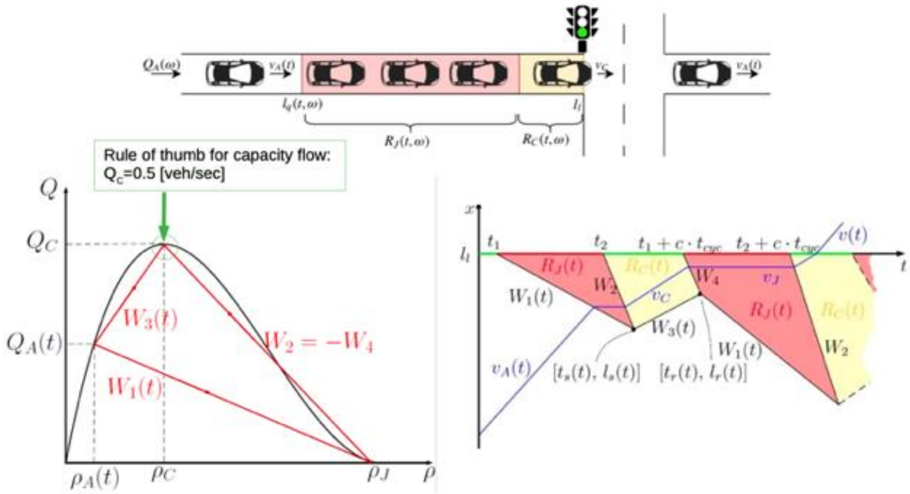


Figure 2. Special case fundamental diagram for queuing at traffic light (Tettamanti, 2021)

3. Problem definition

The main target of this study is to create a shockwave profile model as a queueing model at a signalised intersection. In order to do this, the main task is divided into four subtasks, each one presented in the corresponding section of the paper, as follows:

- Creation of a queueing model at a signalised intersection,
- Creation of SUMO network with traffic lights,
- Creation of logs of vehicle trajectories or queue lengths and parameter estimation with machine learning-based technique of the queueing model,
- Result validation with SUMO.

For these tasks in terms of software, the SUMO v1.8.0 microscopic road traffic simulator was used for network creation and simulation, respectively. MATLAB_R2019b was applied for the creation of the queueing model and for the machine learning parameter estimation.

4. Queuing model creation at a signalised intersection

Traffic simulations supported the present study in the SUMO (Simulation of Urban Mobility) road traffic simulation software. At the same time, the short-term TraCI protocol for “Traffic Control Interface” gives access to a running road traffic simulation, allowing the retrieving of simulated objects' values and manipulating their behaviour “online”.

In a general context, a stop-and-go wave, commonly referred to as a shockwave profile model, can be described as a phenomenon where vehicles slow down or halt sequentially. It is possible to leverage vehicle trajectory data to gauge these shockwaves' propagation speed within a microscopic framework. In the case of this present study, however, a particular queueing model was implemented, similar to the shockwave model (Tettamanti et al., 2019).

Commencing from a stationary stance, a vehicle will undergo maximum acceleration if there are no vehicles ahead. As its velocity increases, the acceleration will diminish until it reaches zero. Conversely, when a vehicle approaches an obstacle like a red light or another vehicle, it will decelerate until its speed reaches 0 [m/time-step], which defines how many vehicles are waiting in a queue approaching a red traffic light (Akçelik, 2001). After the initialisation and setting of the visualisation scheme, we define the main loop of the MATLAB queuing model (see *Algorithm 1*), a sequence going from the start 0, utilising ten simulation steps per second (car following model), to 240 time-steps.

https://doi.org/10.55343/CogSust.65

Algorithm 1

```
while i <= 120*10 + 1
 traci.simulationStep();
 vehicles= traci.vehicle.getIDList();
 for ii=1:length(vehicles)
  traci.vehicle.setSpeedMode(cell2mat(vehicles(ii)),0);
  leader=traci.vehicle.getLeader(cell2mat(vehicles(ii)));
D(ii,i)=traci.vehicle.getDistance(cell2mat(vehicles(ii)));
 Sp(ii,i)=traci.vehicle.getSpeed(cell2mat(vehicles(ii)));
  if((~isempty(leader)) )
   l = traci.vehicle.getPosition(leader);
dist=traci.vehicle.getDrivingDistance2D(cell2mat(vehicles(ii)),l(1), l(2));
   if(dist <= 10) %stop
    speed = traci.vehicle.getSpeed(leader);
elseif((dist>=50)&&(traci.vehicle.getSpeed(cell2mat(vehicles(ii)))== 0))
    speed = 10; else
    speed = 10; end
  else
   speed = 10; end
  nextTLS = traci.vehicle.getNextTLS(cell2mat(vehicles(ii)));
  if(~isempty(nextTLS))
   distanceToNextTLS = nextTLS{1,1}{1,3};
   stateOfNextTLS = nextTLS{1,1}{1,4};
   if ((distanceToNextTLS <=4) && (stateOfNextTLS == 'r'))
    traci.vehicle.setSpeed(cell2mat(vehicles(ii)),0);
   elseif((distanceToNextTLS <=10) && (distanceToNextTLS >4))
    traci.vehicle.setSpeed(cell2mat(vehicles(ii)),speed/2);
   elseif((distanceToNextTLS <=4) && (stateOfNextTLS == 'g'))
    traci.vehicle.setSpeed(cell2mat(vehicles(ii)),speed); else
    traci.vehicle.setSpeed(cell2mat(vehicles(ii)),speed); end
 end
 i=i+1;
end
```

After that comes the definition of the queuing model at a signalised intersection in a MATLAB file, basically the model's main control algorithm. This is associated and interconnected with the TraCI program file containing the error handling (in which the remote port is also defined) and creating an output file containing queue parameters, step length.

Here in this sequence, firstly we define the TraCI simulation step, create a vehicle ID list, and then within a sequence going all the way from 1 to the arbitrary length of the vehicles (total number of 12 cars), we set the vehicle speeds (0 [m/time-step] for queuing, 5 [m/time-step] for approaching status and 10 [m/time-step] for moving status), starting with and based on the leading vehicle, as well as based on the traffic light signals and distances to these (<=4 [m], >4 [m], or <=10 [m]), thus obtaining a more-or-less similar, customised shockwave profile queuing model.

The TraCI program file encompasses 15 distinct SUMO object folders (such as vehicles and routes), each containing 'get' and 'set' functions linked to the respective object. The typical framework for 'get' or 'set' operations also involves the domain (object name) and the methods for retrieving ('get') or altering ('set') the attributes of the targeted object. Upon launching the main control script file, the basic TraCI program becomes accessible. Another important thing is to check whether we have the same port number in the TraCI program and control script files. Otherwise, the MATLAB code will fail (SUMO User Documentation, 2021).

5. Simple SUMO network with traffic lights

From the SUMO network's perspective, we can gain additional information about the observed system if proper modelling is also involved (see Figure 3 below). When it comes to connecting the nodes with edges (see *Algorithm 2*), it has to be mentioned that edges are directed. Each vehicle enters an edge at the node given as *from* and ends at the node given as *to*.

Algorithm 2

```
<additionals>
  <route id="route2" edges="4to1 1to2 2to3 3to4 4to5 5to6 6to7 7to4 4to1"/>
  <rerouter id="rerouter_0" edges="4to1">
          <interval end="1e9">
            <routeProbReroute id="route2"/>
  </rerouter>
</additionals>
```

After that, we will specify the number of vehicles, their ID, distinguishing colours, start or departure positions, the driving route, acceleration, deceleration, and speed of these cars, respectively. We will set a delay value of 100 [ms] as well.

https://doi.org/10.55343/CogSust.65

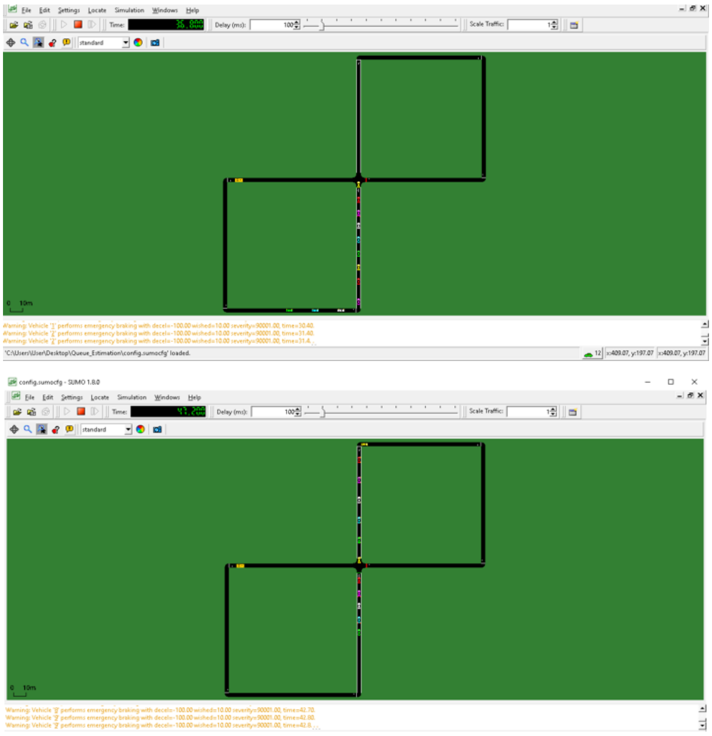


Figure 3 SUMO network with traffic lights at two different time-steps (Own work, 2023)

Once we have all the nodes and edges of the traffic network, we can generate our SUMO road traffic network (Road Traffic Control Laboratory BME, 2021). In *Algorithm 3*, we define all the routes of this network.

Algorithm 3

```
<routes>
    <vType id="type1" accel="10" decel="10" sigma="0.5" length="5" maxSpeed="20"/>
    <route id="route1" edges="1to2 2to3 3to4 4to5 5to6 6to7 7to4 4to1"/>
    <vehicle id="0" type="type1" route="route1" depart="0" color="1,1,0"/>
    <vehicle id="1" type="type1" route="route1" depart="2.0" color="1,0,0"/>
    <vehicle id="2" type="type1" route="route1" depart="4.0" color="1,0,1"/>
    <vehicle id="3" type="type1" route="route1" depart="6.0" color="1,1,1"/>
    <vehicle id="4" type="type1" route="route1" depart="8.0" color="0,1,1"/>
    <vehicle id="5" type="type1" route="route1" depart="10.0" color="0,1,0"/>
    <vehicle id="6" type="type1" route="route1" depart="12.0" color="1,1,0"/>
    <vehicle id="7" type="type1" route="route1" depart="14.0" color="1,0,0"/>
    <vehicle id="8" type="type1" route="route1" depart="16.0" color="1,0,1"/>
    <vehicle id="9" type="type1" route="route1" depart="18.0" color="1,1,1"/>
    <vehicle id="10" type="type1" route="route1" depart="20.0" color="0,1,1"/>
    <vehicle id="11" type="type1" route="route1" depart="22.0" color="0,1,0"/>
</routes>
```

In *Algorithm 4* below, the SUMO network for the simulation and configuration of the intersection are defined and built: the nodes (junctions), edges (streets connecting the junctions), connections and routes, input and output files, as well as the number of vehicles, position and settings of the traffic lights (every 12-simulation step changing red and green light, yellow is omitted for simplicity reasons). In the *Algorithm 4* configuration file, the *Algorithm 2* and *Algorithm 3* scripts are glued together, obtaining a final configuration file for the SUMO simulation (Road Traffic Control Laboratory BME, 2021).

https://doi.org/10.55343/CogSust.65

Algorithm 4

```
<configuration>
  <input>
   <net-file value="network.net.xml"/>
    <route-files value="routes.rou.xml"/>
  </input>
  <processing>
   <time-to-teleport value="-1"/>
  </processing>
  <report>
   <xml-validation value="never"/>
   <duration-log.disable value="true"/>
   <no-step-log value="true"/>
  </report>
  <time>
   <begin value="0"/>
   <end value="10"/>
    </time>
  <gui-settings-filevalue="gui settings.cfg"/>
  <additional-files value="add.xml"/>
</configuration>
```

Finally, we will output all the necessary parameters of the queueing model to validate the obtained results with the machine learning-based parameter estimation described in the following section. These parameters are the vehicle IDs and their position and speed for different time steps. If a given vehicle's speed is 0 [m/time-step], we will consider it waiting in a queue at a traffic light. Thus, it can be determined for how long and how many vehicles are at a given signalised intersection queueing.

6. Queue length estimation with ML technique

The present section introduces practical estimation tools for road traffic measurements, specifically for a signalised intersection queueing scenario. The presented method is a Kalman Filter (KF)-based queue length estimation technique, contributing to a better measurement by filtering the raw data.

The KF algorithm consists of two main stages: prediction and correction. Accordingly, there are two groups of equations in the algorithm: the prediction equations, which produce a state estimate and the extrapolation of estimate error covariance for the next step – the a priori estimation – and the correction equations, which recalculate the state estimate and estimate error covariance based on the updated measurement values – a posteriori estimation (Yin et al., 2018).

To employ the Kalman Filter for estimating the inner state of processes based solely on a sequence of noisy observations (Terra et al., 2014), it is necessary to formulate the process by defining the matrices for each time step k (Ishihara et al., 2006). The KF model postulates that the true state at time k develops from the state at $k-1$ as outlined in (1):

$$x_k = F_k x_{k-1} + B_k u_k + w_k \qquad (1)$$

where,

F_k – state transition model applied to the preceding state x_{k-1},

B_k – control-input model employed on the control vector u_k,

w_k – process noise, hypothesised to originate from a zero-mean multivariate normal distribution $\mathcal{N}$ with covariance Q_k, as presented in (2):

$$w_k \sim \mathcal{N}(0, Q_k) \qquad (2)$$

https://doi.org/10.55343/CogSust.65

At time step k, an observation (measurement) z_k of the true state is acquired, as described by (3):

$$z_k = H_k x_k + v_k \qquad (3)$$

where,

H_k – observation model that maps the true state space into the observed space,

v_k – observation noise assumed to be zero-mean Gaussian white noise, with the observation noise covariance R_k: $v_k \sim \mathcal{N}(0, R_k)$.

The initial state, as well as the noise vectors at each step $\{x_0, w_1, \dots, w_k, v_1, \dots, v_k\}$ are all considered to be mutually independent. The filter's state is described by (4), (5), (6) and (7), while the *Optimal* Kalman gain is defined by (8).

$$x_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \qquad (4)$$

$$P_{k|k} = (I - K_k H_k)\hat{P}_{k|k-1} \qquad (5)$$

$$\hat{x}_{k|k-1} = F_k x_{k-1|k-1} + B_k u_k \qquad (6)$$

$$\hat{P}_{k|k-1} = F_k P_{k-1|k-1} F_k T + Q_k \qquad (7)$$

$$K_k = \hat{P}_{k|k-1} H_k^T S_k^{-1} \qquad (8)$$

where,

$x_{k|k}$ – *a posteriori* state estimate at time k, given observations up to and including at time k,
$P_{k|k}$, – *a posteriori* estimate covariance matrix, representing the measure of estimated accuracy of the state estimate,
$\hat{x}_{k|k-1}$ – predicted (*a priori*) state estimate,
$\hat{P}_{k|k-1}$ – predicted (*a priori*) estimate covariance,
B_k – control–input model,
F_k – state–transition model,
H_k – observation model,
Q_k – covariance of process noise.

The following steps will be defined for the operation of the KF algorithm at each time-step for the prediction phase:

1. Calculating the a priori state estimation by using measured values,
2. Calculating the a priori error covariance.

Similarly, for the correction phase, we can define the following steps:

1. Calculating y_k based on measurement,
2. Calculating the a posteriori state estimation,
3. Calculating the a posteriori error covariance,
4. Increment step index and go to step 1 of the prediction phase.

As vehicles navigate through the intersection, two distinct motion statuses have been defined in the current context: queuing and moving. The queuing status pertains to vehicles that approach with a speed near 0 or come to a complete stop behind the queue. On the other hand, the moving status relates to vehicles that are in motion at a specified speed.

https://doi.org/10.55343/CogSust.65

The rear of the queue is determined as the location of the last vehicle among all those in queuing status at a specific time step. Queue length signifies the overall count of vehicles in the queuing status for a lane at a specific time step. The maximum queue length is the maximum value among the lengths observed within a given cycle.

The Kalman Filter estimation is implemented in a separate script file. The code starts with initialising and loading the output files created at the end of the main control script containing the different distance and speed values. The Kalman Filter utilises driving distances to forecast the velocity of vehicles.

The output of the distances file contains a 12 × 1201 measurement matrix, with the help of which we can estimate the vehicle speeds. The simulation is running with 12 vehicles. The speed data file is mainly used to calculate the queue lengths from the simulation. The simulated queue lengths must be calculated to validate the estimated data. This file contains a 12 × 1201 validation matrix used to calculate the simulated queue lengths for validation.

According to the model, the queuing vehicle speed is considered 0 [m/time-step], but considering the Kalman Filter's tolerance error, we calculate with a queuing velocity of less than 2 [m/time-step].

After the initialisation phase, the Kalman Filter calculates the velocity of the vehicle based on the driven distance for each time-step. *Algorithm 5* below displays the Kalman Iteration steps, where:

- Z represents the measurement vector, where $Z = Simulated\ Data + Random\ Gaussian\ Noise$,
- Phi characterises the dynamics of the vehicle, serving as the motion equation,
- K stands for the Kalman gain, where if K is low, more weight goes to the model prediction. If K is large, more weight goes to the measurement,
- Q signifies the process noise covariance, indicating the level of uncertainty inherent in the model,
- M stands for the measurement matrix,
- R denotes the measurement noise covariance,
- Xk_buffer is used for later display, it is a 2x1201 matrix in which the first row contains the estimated driven distance values, and the second row contains the estimated speed values for each time-step.

Algorithm 5

```
% Kalman iteration
for k=1:Nsamples
    % Z is the measurement vector
    Z = Xtrue(k+1)+sigma_meas*randn;
    Z_buffer(k+1) = Z;
    % Kalman iteration
    P1 = Phi*P*Phi' + Q;
    S = M*P1*M' + R;
    % K is Kalman gain.
    K = P1*M'*inv(S);
    P = P1 - K*M*P1;
    Xk=Phi*Xk_prev + K*(Z-M*Phi*Xk_prev);
    Xk_buffer(:,k+1) = Xk;
    % For the next iteration
    Xk_prev = Xk;
end
```

This estimation is calculated for every vehicle in the model by embedding the Kalman Filter estimation in a for loop, which iterates over all vehicles. The estimated value provided by the KF is always an expected value with a standard deviation. After the iteration, the queue lengths are calculated. The calculation begins after the last vehicle enters the network. The queue lengths are calculated from the simulated and estimated speed dataset by iterating over the velocities and calculating the number of queuing vehicles at each time-step (see *Algorithm 6*):

https://doi.org/10.55343/CogSust.65

Algorithm 6

```
% Queue lengths for each iteration, if speed is lower than 2, the vehicle is assumed to be in a queue
for col = 250:1201
    queueLengthEst = 0;
    queueLengthSim = 0;
    for row = 1:12
       if(Xk_sum(row,col) < 2)
          queueLengthEst=queueLengthEst+1;
       end
       if(Sp(row,col) < 2)
          queueLengthSim=queueLengthSim+1;
       end
    end
   queueLengths(1,col) = queueLengthEst;
   queueLengths(2,col) = queueLengthSim;
end
```

7. Result validation

The results were validated with the Kalman Filter-based parameter estimation technique of the queueing model in MATLAB.

In Figure 4, the estimated driving distances for the 12 vehicles are displayed in red, along with the measured distance of driving of the last vehicle plotted in blue. The figure also shows the estimated queue lengths for each time-step displayed in red and the simulated queue lengths in blue.

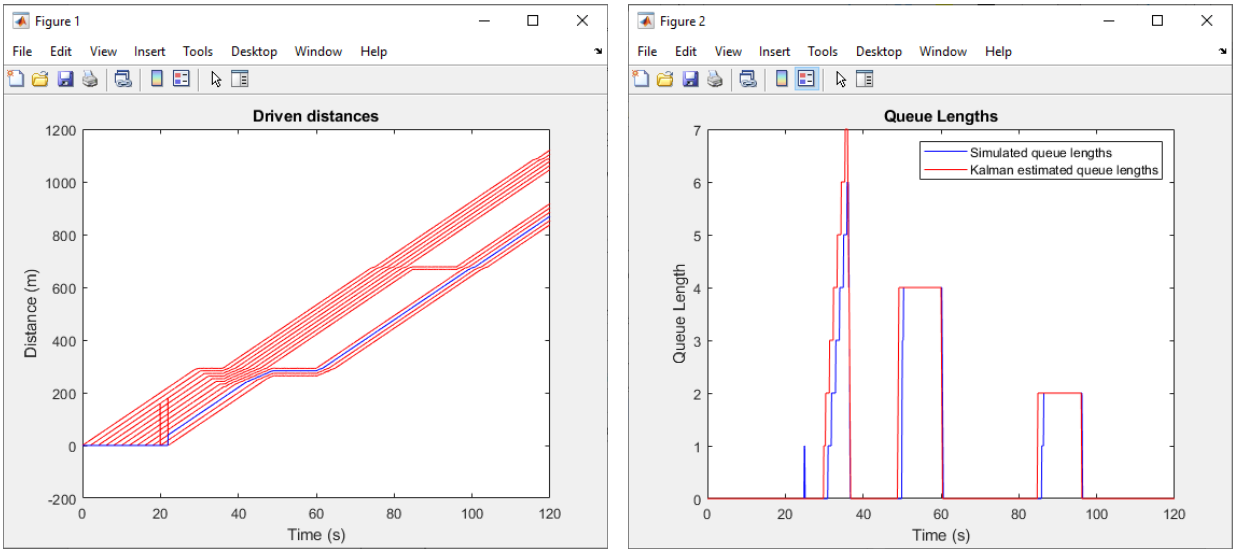


Figure 4 Estimated driving distances and simulated queue lengths (Own work, 2023)

The data in the output files are plotted on two different graphs, together with the speeds at different time-steps obtained from the SUMO simulation for all 12 vehicles, consequently achieving a clear diagram-based result validation showing a close relation between simulation and estimation.

8. Conclusions

The sustainable progress of urban mobility faces obstacles from traffic congestion, traffic accidents, and environmental pollution. Recent research in road traffic engineering, particularly concerning signalised intersections where congestion often accumulates, indicates a growing focus on real-time estimation of motor vehicle queue lengths using the shock wave theory. These approaches are frequently employed with high-resolution loop detector data or probe vehicle information. Here, a novel approach was presented, through which cycle-by-cycle queue lengths were estimated at a given signalised intersection.

Based on analysing and creating the shock wave profile model, the definitions and assumptions used in this study were described, followed by the queuing model and network creation with traffic lights, creation of logs of vehicle trajectories or queue lengths, parameter estimation with a machine-learning based Kalman Filter algorithm, as well as virtual simulation and numerical method based result validation. Despite the noisy measurements, the presented method gave robust and reliable estimation results of the simulated queue lengths.

Future research has to focus on more detailed modelling of the network traffic by considering a more comprehensive simulation of queue length evolution and more precise dynamics of the residual queue. Furthermore, it is important to

https://doi.org/10.55343/CogSust.65

consider sensible enhancements for accurately estimating the residual queue length. In this context, various shockwaves can depict the vehicle arrivals, including whether they assimilate into the queue's rear when the last vehicle is stationary or in motion. Additionally, a more comprehensive calculation of the departure shockwave can be achieved by incorporating assumptions about its stochastic characteristics.

Limitations of the proposed model and algorithms also have to be considered. Future research may enhance the suggested models' precision by utilising more extensive simulated datasets and advanced statistical methodologies (Henrickson et al., 2015).

Acknowledgement

The realisation of this present scientific research project would not have been possible without the support of the Department of Automotive Technologies of the Budapest University of Technology and Economics.

References

Akçelik, R. (2001). *HCM 2000 Back of Queue Model for Signalized Intersections*. Akçelik & Associates Pty. Ltd., Melbourne, Australia.

Cai, Q., Wang, Z., Zheng, L., Wu, B. Wang, Y. (2014). Shock wave approach for estimating queue length at signalised intersections by fusing data from point and mobile sensors. *Transportation Research Record: Journal of the Transportation Research Board*.2422(2), 79–87. DOI: https://doi.org/bkphhn

Cao, H.,Zöldy, M. (2020). An investigation of autonomous vehicle roundabout situation. *Periodica Polytechnica Transportation Engineering*. 48(3), 236–241. DOI: https://doi.org/hg4m

Cetin, M. (2012). Estimating queue dynamics at signalised intersections from probe vehicle data: Methodology based on kinematic wave model. *Transportation Research Record*. 2315(1), 164–172. DOI: https://doi.org/gjh7th

Ferencz, Cs., Zöldy, M. (2021). driving cycle modeling, simulation and validation on 1: 10 scale vehicle model platform [In Hungarian: Autonóm menetciklus-modellezés,-szimuláció és-validálás 1: 10-es méretarányú járműmodell-platformon: Autonomous]. *29th International Conference on Mechanical Engineering*. 29, 208–212.

Fuliang, L., Keshuang, T., Jiarong, Y. Keping L. (2017). Real-Time Queue Length Estimation for Signalized Intersections Using Vehicle Trajectory Data. *Transportation Research Record: Journal of the Transportation Research Board*. 2623, 49–59. DOI: https://doi.org/gdjcrs

Goodall, N. J., Smith, B. L. and Park, B. (2013). Traffic Signal Control with Connected Vehicles. *Transportation Research Record: Journal of the Transportation Research Board*. 2381, 65–72. DOI: https://doi.org/gcvj4f

Henrickson, K., Zou, Y., Wang, Y. (2015). Flexible and robust method for missing loop detector data imputation. *Transportation Research Record: Journal of the Transportation Research Board*. 2527, 29–36. DOI: https://doi.org/gqcghh

Horváth, M., Tettamanti T. (2021). Robust vehicle count estimation on urban signalised links. *Measurement*. 181, 109581. DOI: https://doi.org/kvf8

Ibrahim, A., Cicic, M., Goswami, D., Basten, T., Johansson, K. (2019). Control of Platooned Vehicles in Presence of Traffic Shock Waves, *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. 1727–1734. DOI: https://doi.org/gh95kw

Ishihara, J. Y., Terra, M. H., Campos, J. C. T. (2006). Robust Kalman Filter for Descriptor Systems. *IEEE Transactions on Automatic Control*. 51(8), 1354. DOI: https://doi.org/bm5w3g

Lee, S., Wong, S. C., Li, Y. C. (2015). Real-Time Estimation of Lane-Based Queue Lengths at Isolated Signalized Junctions. *Transportation Research Part C: Emerging Technologies*. 56, 1–17. DOI: https://doi.org/f7hr8w

Liu, H., Wu, X., Ma, W., Hu, H. (2019). Real-time queue length estimation for congested signalised intersections. *Transportation Research Part C: Emerging Technologies*. 17(4), 412–427. DOI: https://doi.org/dprtn9

Road Traffic Control Laboratory BME (2021). *Simple example code with MATLAB*. URL: http://kjit.bme.hu/index.php/en/kutatas-traffic-en/sumotraci-en

Road Traffic Control Laboratory BME (2021). *The beginning steps of SUMO TRACI COM programming with MATLAB*. URL: http://kjit.bme.hu/images/trafficlab/Research/SUMOTRACI/MATLAB_TraCI.pdf

Schrank, D., Eisele, B., Lomax, T (2015). 2014 Urban Mobility Scorecard. *Texas A&M Transportation Institute*, College Station, 2015.3.

Sharma, A., Bullock, D. M., Bonneson, J. A. (2007). Input-Output and Hybrid Techniques for Real-Time Prediction of Delay and Maximum Queue Length at Signalized Intersections. *Transportation Research Record: Journal of the Transportation Research Board*. 2035, 69–80. DOI: https://doi.org/fw429q

SUMO User Documentation (2021). *TraCI overview*. URL: https://sumo.dlr.de/docs/TraCI.html#traci_commands

Terra, M. H., Cerri, J. P., Ishihara, J. Y. (2014). Optimal Robust Linear Quadratic Regulator for Systems Subject to Uncertainties. *IEEE Transactions on Automatic Control*. 59(9), 2586–2591. DOI: https://doi.org/kvf9

Tettamanti, T. (2021). Fundamental Diagram of Traffic Flow. Master Program Course Autonomous Vehicle Control Engineer. Budapest University of Technology and Economics, Faculty of Transportation Engineering and Vehicle Engineering, Budapest. INTFIN Course Material supported by EMMI 32708-2/2017. (Downloaded: 2 May 2023 12:43)

Tettamanti, T., Luspay, T., Varga, I. (2019). *Road Traffic Modeling and Simulation*. Akadémiai Kiadó, Budapest.

Varga, B., Tettamanti, T. (2023). Jam Propagation Analysis With Mesoscopic Traffic Simulation. IEEE Transactions on Intelligent Transportation Systems. pp. 1-12. DOI: https://doi.org/kwgg

Viti, F., Van Zuylen, H. J. (2010). Probabilistic Models for Queues at Fixed Control Signals. *Transportation Research Part B: Methodological*. 44(1), 120–135. DOI: https://doi.org/dv7t7s

Wang, S., Huang, W., Lo, H. (2019). Traffic parameters estimation for signalised intersections based on combined shockwave analysis and Bayesian network. *Transportation Research Part C: Emerging Technologies*. 104, 22–37. DOI: https://doi.org/gf5t9q

Wu, A., Yang, X. (2013). Real-time queue length estimation of signalised intersections based on RFID data. *Procedia – Social and Behavioral Sciences*. 96, 1477–1484. DOI: https://doi.org/kvgb

Wu, X., Liu, H. (2011). A Shock Wave Profile Model for Traffic Flow on Congested Urban Arterials. *Transportation Research Part B: Methodological*, 45, No. 10, 1768–1786. DOI: https://doi.org/dpjqxq

https://doi.org/10.55343/CogSust.65

Yao, J., Tang, K. (2019). Cycle-based queue length estimation considering spillover conditions based on low-resolution point detector data. *Transportation Research Part C: Emerging Technologies*. 109, 1–18. DOI: https://doi.org/kvgc

Yin, J., Sun, J., Tang, K. (2018). A Kalman Filter-Based Queue Length Estimation Method with Low-Penetration Mobile Sensor Data at Signalized Intersections. *Transportation Research Record: Journal of the Transportation Research Board*. DOI: https://doi.org/gffh6j

Zöldy, M., Baranyi, P. (2021). Cognitive Mobility – CogMob. *12th IEEE International Conference on Cognitive Infocommunications – CogInfoCom 2021*. Proceedings IEEE, 921–925. URL: https://www.researchgate.net/publication/361435417_Cognitive_Mobility_-_CogMob