Passage-Detector-Based Traffic Queue Estimation in Intelligent Transportation Systems: A Computational Study of Competing Algorithms

Simon Moskowitz
Emmanuel Fernández-Gaucherand*

and

Michael Whalen

Department of Systems and Industrial Engineering
University of Arizona
Tucson, Arizona 85721

Transmitted by L. Duckstein

ABSTRACT

We present here the results of a comprehensive computational study of competing algorithms for traffic queue estimation. These estimators can be used in conjunction with adaptive feedback traffic light control schemes in Intelligent Transportation Systems. Dynamic queue length estimators proposed by Baras, Levine and Lin are shown to be significantly superior to other methods proposed in the literature. © Elsevier Science Inc., 1997

1. INTRODUCTION

Nobody enjoys being detained in traffic, whether as a passenger or a driver. However, it is an evil we have to deal with on a daily basis, particularly in the United States where most people also prefer the freedom

*Corresponding author.
of transportation afforded by private vehicles. The first widespread attempt
to address this problem was by means of traffic lights. Increased commuter
and pedestrian safety through the reduction of car accidents was the
primary reason for the institution of time-varying traffic signals. However, it
is only until relatively recently that computer and sensor electronics have
become powerful and inexpensive enough to attempt decentralized, real-time
adaptive flow control of a large urban network, one of the main objectives
behind Intelligent Transportation Systems (ITS).

The goal of minimizing overall stopping time may be realized through
controlling the timing of the coordinated traffic lights (red vs. green time) at
the principal road intersections of a city, based on measured information
about the state of the traffic conditions, e.g. size of queues and arrival rates.
Many groups, including one at the University of Arizona (UA), are currently
developing and evaluating schemes and algorithms for such systems [1].
Most of these schemes require an accurate estimate, based on sensor
information, of the number of cars queued at the stoplines of key intersec-
tions. Not only do they need to estimate accurately how many cars are there
now, which could be provided (albeit expensively) perfectly through sensors,
but they need to accurately estimate how many cars will be there in the
future so that the traffic lights may be adjusted accordingly.

This paper examines and compares existing one-step-ahead predictors.
Here one-step-ahead means estimating what the queue length will be after
the next time interval, given information available up to the present time.
In Section 2, two one-step prediction algorithms are examined, one is based
on recursive estimation for discrete-time point processes [2, 3, 4], and the
other is a novel simple algorithm based on heuristics and approximations. In
Section 3, two test intersections used for the computational evaluation of
these algorithms are detailed. Section 4 shows how the basic algorithms are
adapted to the intersections. Parameter selection for the algorithms and
intersections is then detailed in Section 5. The outcome from several detailed
simulation experiments is presented in Section 6. Finally, conclusions are
drawn from the results and future actions are suggested.

2. ONE-STEP-AHEAD PREDICTORS

It appears from the available literature on prediction of traffic queues [2,
5–9], that only Baras, Levine, and Lin [2] have attempted to estimate
dynamically the queue length at an intersection. The most that anyone else
seems to have done is to estimate the residual queue remaining at the end of
the green phase of a steady-state system, i.e., via a constant estimate of the
mean queue length [5–9]. It has even been suggested that a constant
estimate of average queue length would have a better length versus time curve fit than any other estimate [9]. Possible reasons for the apparent lack of subsequent use and extension of the results in [2] may be: a) dry-out of federal funding for this type of research in the late '70s to mid '80s, and b) considerable computational complexity of the algorithms for the available field-deployable computers of the time. However, dramatic advances in software and hardware, as well as the current and ever increasing importance of this field amply justify the reconsideration of techniques as in [2]. As will be demonstrated these algorithms are not only implementable, but also very efficient. A computational study is presented here, where one of the Baras-Levine-Lin models is juxtaposed with a constant estimate and the QuickQ algorithm—a novel simple counting technique* [10].

**Baras-Levine-Lin Models**

Baras, Levine, and Lin derived theoretically three discrete-time, point process driven, minimum square error, single-step models [2], based on more general recursive estimation techniques† [3]. All three assume a single isolated intersection of two one-way single-lane streets. Model A obtains its point-process data from a vehicle passage detector (e.g. inductive loop) upstream from the stopline. The detector is assumed to provide one pulse for each vehicle passing over it. Model B additionally acquires velocity data from the detector. Model C places an additional detector (of the type in Model B) at the stopline. A variant of Model A is implemented for this study; since it is the simplest computationally, it would require the least amount of hardware if employed in reality and, as will be shown, it seems to perform well enough. The following equations comprise Baras-Levine-Lin Model A [2]:

\[
\hat{x}_i(t | t) = \begin{cases} 
1 - \lambda(i, t) & \text{if } n(t) = 0 \\
\frac{\sum_{i=0}^{N} \{1 - \lambda(i, t)\} \hat{x}_i(t | t - 1)}{\sum_{i=0}^{N} \lambda(i, t) \hat{x}_i(t | t - 1)}, & \text{if } n(t) = 1,
\end{cases}
\]

\[ \hat{z}(t + 1 | t) = \begin{cases} 
M^T(t) \hat{x}(t | t) + \frac{\{Q^T(t) - M^T(t)\} \hat{x}(t | t - 1)}{\sum_{i=0}^{N} (1 - \lambda(i, t)) \hat{x}_i(t | t - 1)}, & \text{if } n(t) = 0 \\
M^T(t) \hat{x}(t | t), & \text{if } n(t) = 1,
\end{cases} \]  
(2)

\[ M^T_{ii}(t) = \mu(i), \]
\[ M^T_{i+1,i}(t) = 1 - \mu(i), \forall i = 0, 1, \ldots, N, \]  
(3)

\[ M^T_{ij}(t) = 0, \text{ elsewhere}, \]

\[ Q^T_{ii}(t) = \{1 - \lambda(i)\}\{1 - \mu(i)\} + \lambda(i)\mu(i), \]

\[ Q^T_{i,i-1}(t) = \lambda(i - 1)\{1 - \mu(i - 1)\}, \]

\[ Q^T_{i,i+1}(t) = \{1 - \lambda(i + 1)\}\mu(i + 1), \]  
(4)

\[ Q^T_{ij}(t) = 0, \text{ elsewhere}, \]

\[ \hat{Z}(t + 1 | t) = E[\hat{z}(t + 1 | t)] = \sum_{i=0}^{N} \hat{x}_i(t + 1 | t), \]  
(5)

where,

- \(n(t)\): the observed point-process signal from the detector, equal to the number of vehicles (at most one) that passed over the detector during time interval \(t\).
- \(N\): the maximum queue capacity (the number of vehicles that can fit between the stopline and the detector).
- \(\hat{x}_i(t | s)\): the probability that the queue contains \(i\) cars at time \(t\), given the information available up to time \(s\), where \(i\) varies from 0 to \(N\).
- \(\lambda(i, t)\): the rate at which vehicles arrive at the detector during time interval \(t\), given that the queue length is \(i\).
- \(\mu(i, t)\): the rate at which vehicles depart from the queue during time interval \(t\), given that the queue length is \(i\).
- \(Q^T_{ij}(t)\): the \((i, j)\) element of the transpose of the standard Markov chain transition matrix; i.e., the probability that the queue at time \(t + 1\) contains \(i\) vehicles given that the queue at time \(t\) contains \(j\) vehicles.
Traffic Queue Estimation

- $M^T(t)$: the transpose of the departure rate matrix.
- $X(t)$: the actual queue length at time $t$—the quantity we wish to estimate.
- $\hat{X}(t + 1 | t)$: the estimate of the queue length at time $t + 1$ given the information available up to time $t$.

QuickQ

The idea behind this estimator is very simple: count the number of arrivals, as given by $n(t)$, and assume cars leave at a rate dependent on the state of the light. Thus, we have:

$$\hat{X}(t + 1 | t) = \max\{ \hat{X}(t | t - 1) - \mu(t), 0 \} + n(t),$$  \hspace{1cm} (6)

$$\mu(t) = \begin{cases} 
\mu_{\text{green}}, & \text{if Light}(t) = \text{Green}; \\
\mu_{\text{arrow}}, & \text{if Light}(t) = \text{Green Turn Arrow}; \\
\mu_{\text{red}}, & \text{otherwise.}
\end{cases}$$  \hspace{1cm} (7)

The maximum ensures that predictions of departures don’t exceed the number of cars present.

3. TEST INTERSECTIONS

The ITS group at the UA has modeled some heavily traveled roads in Tucson, Arizona, on NETSIM, a microscopic traffic network simulation package [11]. The data for queue predictor computational evaluation was drawn from their model. Figure 1 shows the location of the two intersections chosen. Campbell Ave. approaching Broadway Blvd. from the north (Figure 2) has three through lanes, a right-hand turn pocket, and a double left-hand turn pocket. The intersection was selected because it has all three types of lanes, is centrally located, and is fairly busy. Speedway Blvd. approaching Campbell Ave. from the east (Figure 3) was selected as a second test intersection for comparison. This intersection is similar to the first, except that it has only one left turn lane and higher traffic volume.

4. ALGORITHM MODIFICATIONS

The Baras-Levine-Lin Model A algorithm described earlier assumes single-lane, one-way streets. In order to apply this algorithm to a more complex intersection, like Broadway Blvd. and Campbell Ave., some modifications are required.
FIG. 1. Tucson, Arizona NETSIM network.

FIG. 2. Intersection of Campbell and Broadway.
First a small correctional addition is needed. When the queue estimate was a full queue with probability 1 \( \pi_N(t | t - 1) = 1 \) and another car passes over the detector \( n(t) = 1 \), the next estimate should also be a full queue with probability 1 \( \hat{\pi}_N(t | t) = 1 \). Verifying the denominator of (1) is not zero before revising the estimate alleviates this problem.

A multi-lane passage detector will often have multiple cars going over it at the same time. Therefore, in order to generate one pulse per passing car in the point process driving the algorithm the actual time interval over which the data is taken was subdivided for computational purposes into as many intervals as there are lanes. For simplicity, the first \( n \) intervals were each assigned one arrival if there are \( n \) arrivals during the detector time interval. NETSIM reports the total number of cars that have passed over the detector in one second periods. Thus, for example, for Broadway and Campbell where the passage detector covers three lanes, if the NETSIM detector data reports two cars passed in the last second, the algorithm will receive one car in the first third of a second, one car in the second third of a second, and zero cars in the last third of a second.
The cars at the stopline were divided into two queues. The right-hand turn lane and the three through lanes were designated Queue 1. The left-hand turn lane(s) were designated Queue 2. This separation produces greater accuracy and provides more information for those attempting traffic control.

In the years since Baras, Levine, and Lin derived their estimator, devices which detect an empty queue have been installed at several intersections. Data from these "empty queue detectors" is used to reset the estimate to an empty queue with probability 1 in both algorithms. This resetting improves the algorithm by correcting the estimate before it diverges greatly from reality. The algorithm that incorporates the information from the empty queue detectors is referred to in the sequel as Baras-Levine-Lin A0.

5. PARAMETER SELECTION

Baras-Levine-Line Model A0

Estimation of the departure rate and arrival rate parameters, $\mu$ and $\lambda$, respectively, as a function of queue length is very difficult. Furthermore, since queue length dependence can be expected to be weak (although variable) in general, $\mu$ and $\lambda$ will be treated as functions only of the state of the downstream light for simplicity, as is done in [2]. This is not to say that these parameters do not vary also with intersection location, lane structure, traffic density, and the stochastic nature of traffic flow.

Since the model is primarily arrival point-process driven, it is not surprising that the model is practically insensitive to $\lambda(k,t)$, as was discovered experimentally. However, for Queue 1 (through and right lanes), $\mu_{\text{green}}$ is of utmost importance and $\mu_{\text{red}}$ is secondary, but still significant. The same is true for $\mu_{\text{arrow}}$ (green left-arrow) and $\mu_{\text{green}}$ respectively, for Queue 2 (left lanes). $\mu_{\text{green}}$ and $\mu_{\text{red}}$ were varied while $\lambda$'s were held constant for experiment 1 of Queue 1 at Broadway and Campbell. This produced a concave surface (ensuring the existence of a global minimum) when using sample mean absolute error (S.M.A.E.) for measuring the "goodness of fit" of Baras-Levine-Lin A0 to the actual queue length, where:

$$S.M.A.E.(\hat{\lambda}) = \frac{\sum_{t=0}^{T-1} \hat{\lambda}(t+1,t) - Z(t+1)}{T - 1}.$$  \hspace{1cm} (8)

The optimal parameters were found to have only small variations for different experiments (different random number seeds) at the same intersec-
TABLE 1
PARAMETERS USED FOR BARAS-LEVINE-LIN A0

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Broadway &amp; Campbell</th>
<th>Speedway &amp; Campbell</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Queue 1</td>
<td>Queue 2</td>
</tr>
<tr>
<td>$\lambda_{\text{green}}$</td>
<td>0.6</td>
<td>0.4</td>
</tr>
<tr>
<td>$\lambda_{\text{arrow}}$</td>
<td>N/A</td>
<td>0.1</td>
</tr>
<tr>
<td>$\lambda_{\text{red}}$</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>$\mu_{\text{green}}$</td>
<td>1.5</td>
<td>0.2</td>
</tr>
<tr>
<td>$\mu_{\text{arrow}}$</td>
<td>N/A</td>
<td>0.6</td>
</tr>
<tr>
<td>$\mu_{\text{red}}$</td>
<td>0.005</td>
<td>0</td>
</tr>
</tbody>
</table>

This procedure was repeated for Queue 2 and then for Speedway and Campbell. The use of the word *optimal* here means that use of these parameters provides the (approximate) lowest S.M.A.E. in experiments using Baras-Levine-Lin A0. The parameters may not be identical to the actual arrival and departure rates, but the values selected provide an optimal tuning for the algorithms used. The parameters selected for algorithm comparison are shown in Table 1.

**QuickQ**

For the QuickQ algorithm $\mu_{\text{green}}$ for Queue 1 (see Figure 4) and $\mu_{\text{arrow}}$ and $\mu_{\text{green}}$ for Queue 2 were found to be the important parameters. The parameters used in the algorithm comparison were obtained again by minimizing sample mean absolute error for several experiments (Table 2). Again, the parameters may not correspond with either real rates or the Baras-Levine-Lin Model A0 parameters. However, if either algorithm were to be used in reality, a tuning procedure of this type could be followed using historical data.

6. SIMULATION RESULTS

NETSIM was run ten times, each time with a different random number seed. Then the algorithms were run on the detector and traffic light data were generated for each test intersection. The estimates were compared with the actual queue lengths from NETSIM.
Figure 5 is a sample plot of an estimated queue length versus time, showing good fit (run 1, Queue 1, Broadway and Campbell). "Constant" refers to the perfect average queue length estimator. Figure 6 shows the worst fit run. Figures 7, 8, and 9 show the mean absolute error for each run. Figures 10, 11, 12 show the overall mean absolute error. Figures 13 and 14

<table>
<thead>
<tr>
<th>TABLE 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARAMETERS USED FOR QUICKQ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Broadway &amp; Campbell</th>
<th>Speedway &amp; Campbell</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Queue 1</td>
<td>Queue 2</td>
</tr>
<tr>
<td>$\mu_{\text{green}}$</td>
<td>1.06</td>
<td>0.16</td>
</tr>
<tr>
<td>$\mu_{\text{arrow}}$</td>
<td>N/A</td>
<td>0.76</td>
</tr>
<tr>
<td>$\mu_{\text{red}}$</td>
<td>0.06</td>
<td>0</td>
</tr>
</tbody>
</table>
1-Step Predictors
Queue 1, Broadway & Campbell, Run 1

FIG. 5. Sample plot of estimated queue length v. time, good fit.

1-Step Predictors
Queue 1, Speedway & Campbell, Run 9

FIG. 6. Estimated queue length v. time, worst fit run.
Mean Absolute Error
Queue 1 @ Broadway & Campbell

Fig. 7. Mean absolute error, Queue 1, Broadway and Campbell.

Mean Absolute Error
Queue 1 @ Speedway & Campbell

Fig. 8. Mean absolute error, Queue 1, Speedway and Campbell.
Mean Absolute Error
Queue 2 @ Broadway & Campbell

Fig. 9. Mean absolute error, Queue 2, Broadway and Campbell.

Overall Mean Absolute Error
Queue 1 @ Broadway & Campbell

Fig. 10. Overall mean absolute error, Queue 1, Broadway and Campbell.
Overall Mean Absolute Error
Queue 1 @ Speedway & Campbell

![Bar chart showing mean absolute error for different algorithms: Constant, Baras A0, Baras A, QuickQ.]

Fig. 11. Overall mean absolute error, Queue 1, Speedway and Campbell.

Overall Mean Absolute Error
Queue 2 @ Broadway & Campbell

![Bar chart showing mean absolute error for different algorithms: Constant, Baras A0, Baras A, QuickQ.]

Fig. 12. Overall mean absolute error, Queue 2, Broadway and Campbell.
show the average queue length for each run. Figures 15, 16, and 17 show the mean relative error for each run. Figures 18, 19, and 20 show the mean overall relative error. (Run 9 for Queue 1 at Speedway and Campbell is an outlier, i.e., statistically negligible, and is therefore not used in overall calculations. Its occurrence is due to randomness generation capabilities of NETSIM.) The mean relative error is equal to the mean absolute error divided by the average queue length.

7. CONCLUSION

As illustrated through ample experimentation in this study, queue lengths can be efficiently predicted dynamically with a great degree of accuracy. For Queue 1, the Baras-Levine-Lin A algorithm performs nearly the same with or without the empty queue detector. For Queue 2 at Broadway and Campbell, Baras-Levine-Lin A performs much worse than the constant estimate but Baras-Levine-Lin A0 performs significantly better than the constant estimate. This almost certainly has to do with this queue being often empty, averaging less than two cars, and always being less than six
Fig. 14. Average queue length, Speedway and Campbell.

Fig. 15. Mean relative error, Queue 1 at Broadway and Campbell.
Mean Relative Error
Queue 1 @ Speedway & Campbell

FIG. 16. Mean relative error, Queue 1 at Speedway and Campbell.

cars long. Estimating short queues of left-turners is difficult and not very worthwhile, thus Queue 2 at Speedway and Campbell was not examined.

Overall mean absolute error seems to be the best comparison of effectiveness. For Queue 1, Baras-Levine-Lin A0 gets within 20–30% of the true queue length 95% of the time, regardless of the intersection. QuickQ is about 5% worse at Broadway and Campbell and 20% worse (with a higher variance) at Speedway and Campbell. Therefore Baras-Levine-Lin A0 is more robust than QuickQ. Not surprisingly, and as evidenced by, e.g., Figures 7–9 and 18–20, the Baras-Levine-Lin A0 algorithm performs significantly better than the QuickQ algorithm for higher traffic intensities, the first one being explicitly derived to track traffic variations dynamically. However, as Figure 7 clearly shows, any estimator can at times be very poor. This seems to be caused by occurrence of the experimental event of more than a dozen right-hand turns in a row, probably a rare event in reality.

The Baras-Levine-Lin A0 also provides (conditional) probability distribution data, which QuickQ (and the constant mean queue length estimator) do not. This additional information may prove helpful for implementing feedback traffic control strategies where the state is taken precisely as this distribution data, see [12].
Mean Relative Error
Queue 2 @ Broadway & Campbell

FIG. 17. Mean relative error, Queue 2 at Broadway and Campbell.

Overall Mean Relative Error
Queue 1 @ Broadway & Campbell

FIG. 18. Overall mean relative error, Queue 1 at Broadway and Campbell.
Fig. 19. Overall mean relative error, Queue 1 at Speedway and Campbell.

Fig. 20. Overall mean relative error, Queue 2 at Broadway and Campbell.
In terms of computational complexity, the constant estimate is $O(0)$, QuickQ is $O(1)$, and Baras-Levine-Lin A0 is $O(N^2)$, where $N$ is the maximum queue capacity. With the speed and storage capacity of today's field-deployable computers, the Baras-Levine-Lin A0 algorithm is both implementable and a viable alternative for those situations where the increased accuracy in the estimates provided by it may be crucial.

Looking back again at Figure 7 suggests two possible improvements for the QuickQ algorithm. First, a maximum queue capacity should prove useful. Second, a small delay before cars are assumed to depart after the downstream light turns green might be useful.

Examining the intersection layouts (Figures 2 and 3), one realizes that cars leaving Queue 1 to turn left will often pass over the Queue 2 arrival detector. This information can be used to augment the knowledge about Queue 1.

It is easy to envision algorithms which could dynamically adjust the parameters as well, leading to a truer adaptive/dynamic estimator. However, the complexity for adaptive schemes of this nature may be significantly high; see [13]. The queue length dependence of the departure rates in Baras-Levine-Lin A0 could also be studied, either through empirical computer trials or by direct observation of a real intersection.

Finally, a multi-step predictor could also be derived, e.g., along the lines of Baras-Levine-Lin A. Here the additional information in the Baras B and C models might be useful, since the prediction would be several seconds into the future.

The research of E. Fernández-Gaucherand was partially supported by The University of Arizona Foundation and The Office of the Vice President for Research, by The National Science Foundation under grant NSF-INT 9201430, and by The Engineering Foundation under grant RI-A-93-10. Email: <emmanuel@sie.arizona.edu>. The authors wish to thank the referees for many valuable comments, and thank Assistant Professor K. L. Head from our department for sharing references [5]–[7] with us.

REFERENCES


