Speed Advisory System Using Real-Time Actuated Traffic Light Phase Length Prediction

Mikhail Burov

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2019-184
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-184.html

December 24, 2019

Speed Advisory System Using Real-Time Actuated Traffic Light Phase Length Prediction

by Mikhail Burov

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor Murat Arcak

Research Advisor

12/17/2019

(Date)

****

Pravin Varaiya

Professor Pravin Varaiya

Second Reader

12/23/2019

(Date)

Speed Advisory System Using Real-Time Actuated Traffic Light Phase Length Prediction

Mikhail Burov

1 Abstract

A speed advisory system (SAS) for connected vehicles (CVs) on urban streets is based on the estimation of green (or red) light duration at signalized intersections. A particular challenge is to predict the signal phases of semi- and fully-actuated signals.

In this paper, we introduce an algorithm predicting whether a given CV will be able to make it through the next intersection with an actuated signal on green or not, based on available traffic measurement data. Our mechanism processes traffic data collected from advanced detectors on incoming links and assigns "PASS"/"WAIT" labels to vehicles according to their estimated ability to go through the intersection within the current phase. Additional computations provide an estimate for the duration of the current green phase that can be used by SAS to minimize fuel consumption.

Simulation results show 95% prediction accuracy, which yields up to 30% reduction in fuel consumption when used in SAS. Traffic progression quality also benefits from our mechanism demonstrating an improvement of 20% at peak for medium traffic demand, reducing delays and idling at intersections.

2 Introduction

Vehicles equipped with the Speed Advisory System [1] use traffic light information and environment data to obtain the optimal speed trajectories to minimize idling at traffic lights. Such trajectories, in turn, serve to minimize fuel consumption. One of the main parameters that is required by the SAS is the estimated remaining time until the end of phase, more specifically, how much time left in a "green phase"/when the nearest "green phase" starts. In the case of static traffic light (constant phase length) this information is easily obtained directly from the traffic light (TL) via Signal Phase and Timing (SPaT) messages or any other signal system. However, if the intersection is equipped with an actuated TL, it becomes impossible to find that parameter due to dependency on the demand and traffic condition in general. Thus, it may vary from "minimal duration" to "maximal duration" - specific variables characterizing the TL. In the latter case, it is essential to obtain some other information, which will be helpful in building optimal speed trajectories.

The software presented in [2] encourages drivers to use their smartphone cameras to detect the traffic light color at an upcoming intersection and estimate the remaining time within the current phase. Identifying the color itself turned out to be quite imprecise, showing error rates from 7.8% to 12.4%. The phase length estimation, based on the five previous green-red/red-green transitions is also inefficient. According to the algorithm, the best prediction is just slightly better than estimating the current phase length to be the same as the previous phase length.

A follow-on study [3] addresses the problem of finding optimal speed trajectories in order to minimize fuel consumption. Although the paper states that the problem of dynamic traffic light is resolved via "1) a set of logical rules that calculates a reference velocity for timely arrival at green lights combined with 2) a model predictive controller that tracks this target velocity", only pre-timed signals were considered in the algorithm, leaving behind issues with adaptive phase durations.

The studies [4] and [5] also focus on pre-timed traffic lights. The duration of cycles and phases are estimated based on the speed measurements of "floating cars".

References [2] and [6] use noisy measurements of signal phase to process SPaT estimation. The study processes many samples of GPS position and speed from 4300 buses within a period of one month in order to estimate phase duration, cycle length and cycle start time. Due to the small percentage of usable data, the accuracy is limited (6s error for 36s phase duration).

The later study [7] estimates the wait time spent by the bus in queue. The research presents significantly better results in the SPaT estimate.

All of the noisy measurement-based algorithms are implemented only for pre-timed traffic lights. In addition to that, collecting and processing noisy measurements take a lot of time and computational resources, which seems to be inefficient, since most of the signal data is available from Transportation Authority.

The study [8] makes probabilistic SPaT predictions based on the traffic intersection data. For every second within the cycle, the algorithm tries to predict whether the phase k is **G**(green), **R**(red) or **M**(uncertain) with some level of confidence. Unfortunately, the predictions may be uninformative. Moreover, since the algorithm uses some "level of confidence", it does not provide firm guarantees.

A statistical approach is pursued in [9], where the algorithm relies on the historical data from several intersections in Munich. The algorithm presented in the study uses a Kalman Filter to predict future probability distributions. Although a high level of accuracy was achieved (95 percent), the practical applicability is limited by the fact that availability level is only 71 percent in average.

Another study [10] suggests using the previous phase measurements and the real time information that locates the current time within the current phase in order to predict the times for all future phase transitions. One of the approaches is to compute the "conditional expectation based prediction", which estimates $\bar{d} = \mathbb{E}[d|d > t]$ - expected phase duration and $\bar{r} = 1 - \bar{d}$ - residual phase duration (d - length of the phase). Another algorithm, presented in the paper, is "confidence based prediction". It uses the empirical probability density function (PDF) to give a prediction with a given confidence bound. By solving the simple conditional equation $1 - F(t \mid d > \bar{t}) = \alpha$ (where $F(t) = P(d \leq t)$) with respect to t, the estimated phase duration is obtained with confidence level α. Those methods greatly improve the prediction of the residual time for that phase as well as for a subsequent phase; however, as stated in the paper, such conditional algorithms "pose a challenge to the design of speed profiles that reduce fuel consumption". Since both of the methods return predictions as functions of time, SAS-equipped vehicles, computing the optimal trajectories, might use conflicting estimations of residual time at every time-step. For example, assuming the current time within the phase is t_1, the prediction is t^1_{est}; the next time step $t_2 = t_1 + t_{step}$ the prediction might be $t^2_{est} \neq t^1_{est}$, which gives a different estimated remaining time and, thus, causes jerky motion. Now assume only one such prediction is considered (the later ones do not influence the decision). Assume also, that according to this prediction, a vehicle is required to stop. However, if we considered the following estimate, we would be guided through the intersection without stopping. Such mistakes would be common and would bring the efficiency of the algorithm to a very low level.

Several papers study predicting vehicle flow at arterial roads and using the prediction for adaptive signal control [11] [12][13].

Another approach in finding a speed trajectory that minimizes fuel consumption is solving an optimization problem, where phase duration estimation is addressed as a constraint. In [15] this constraint has the form of $c^i_p \geq c^i_r + F^{-1}(\eta)$, where c^i_p is the vehicle passing time in the signal-cycling clock of intersection i, c^i_r is minimal red-phase duration of the i^{th} intersection, $F^{-1}()$ is the inverse of the CDF function $F()$ of random variable α representing stochastic time of delay, caused by signal uncertainties or vehicle waiting queue, and η is the required reliability level. This algorithm relies on the assumption that CDF is continuous and bijective. Moreover, the distribution function of α is non-parametric in general and may vary depending on arbitrary conditions (time of the day, day of the month, weather, etc.). Thus, determining the true distribution of α might be a challenging problem. Although the paper presents impressive results in terms of fuel consumption (50% - 57%), the algorithm can be used only on the secondary road with no actuation capability (Effective Red implies that green phase on the perpendicular direction is the actuated one). We believe that this approach will not be efficient on the primary actuated road and may be improved, allowing even better results.

Most of the algorithms discussed above try to estimate or predict the phase length/residual time of the phase using historical data and statistical methods. In contrast, our approach relies on the real-time data and assigns "PASS" / "WAIT" labels to vehicles depending on their capability of crossing the intersection within the nearest green phase.

3 Technical Part

3.1 Overview

Our primary objective is to determine whether or not a certain car is going to pass the upcoming intersection during the current "green phase" rather than to estimate the phase duration (which, in fact, can be done as a secondary objective). Using the data from advanced detectors and the upcoming actuated TL itself, the algorithm determines for each car individually whether the TL state and upstream traffic create a sufficient condition to let the vehicle through.

Only after obtaining the labeling ("PASS"/ "WAIT"), we move on to computing the residual phase time if necessary.

It is important to mention that only traffic lights with fixed cycle length and one actuated axis (from this point on any "green phase" / "red phase" is referred to the actuated axis TL color) are considered. In that case, knowing the time within the cycle allows us to compute the precise remaining time until the next cycle (i.e. next green phase, assuming that every cycle starts with the green phase). Thus, the predictions are unnecessary while the current phase is not green, since the residual time is known. In other words, we are making a prediction only when the current phase is green.

All the computations and simulations were conducted in an open source simulator SUMO (Simulation of Urban MObility).

3.2 Simple and Complex Network Architecture

The work consists of two parts. In the first part, we analyze a simple symmetric one-TL-intersection (Fig. 1) with East $\leftrightarrow$ West actuated axis in order to test an idealistic set-up and obtain the best possible result as a benchmark for further explorations. The second part is a simulation of a complex system (Montgomery County network) (Fig. 2) consisting of 9 actuated traffic lights with different schedules. Each intersection corresponding to one of those TLs has its own geometry and surroundings. That set-up allows us to test the algorithm in more realistic conditions and show that even with a lot of unknown information it performs better than many known procedures.

Every incoming link on the actuated axis is equipped with three types of detectors: stop-bar detector and two advanced detectors ("actuator" and "counter"). An actuator is responsible for prolonging the green phase when a vehicle is detected. It is placed 40-100 meters (100m in simple case simulation and 40-60m for complex network) before the intersection. A counter collects speeds and crossing times of all vehicles passing it. Based on that data, the algorithm makes a prediction and transmits it to a vehicle. Counters are located far enough (50+ m) from actuators.

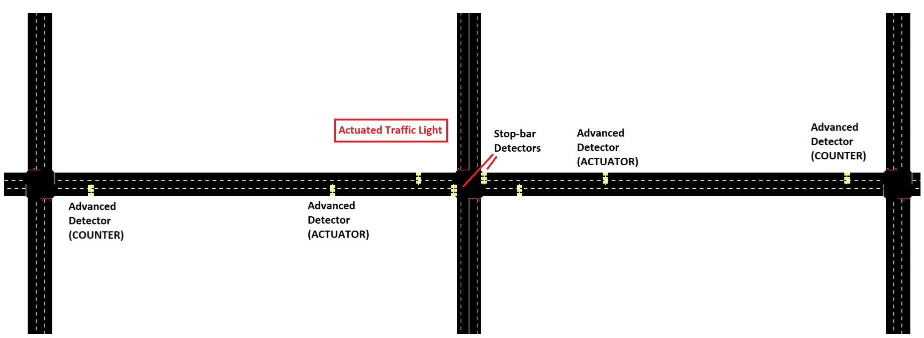


Figure 1: Simple Network.

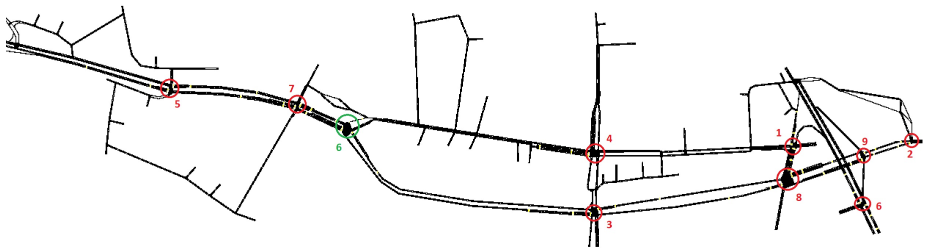


Figure 2: Complex Network. Actuated traffic lights are highlighted RED, the only static traffic light - GREEN

3.3 Traffic Light Properties

Here we introduce the TL parameters used by the algorithm. As stated earlier, the cycle length is fixed and equals to $cycLen_j$ (90 and 90-120 seconds for simple and complex cases respectively). The cycle consists of 4 phases in the simple case (Table 1) and up to 8 phases in the complex set-up. As it will be seen later, the number of phases is not important for the algorithm, since it distinguishes "green" and "NOT green" phases only.

Moreover, we assume that the TL is time-gap actuated, i.e., a vehicle can activate it only if the previous actuation was at most $minGap$ seconds ago (3 seconds for our simulations).

Furthermore, the algorithm requires the knowledge of the travel time from the actuator to the corresponding intersection j at speed limit: $T^j_{a-i} = \lceil \frac{D^j_k}{sl^k} \rceil$, where D^j_k - distance from the actuator on the incoming link k to the intersection j and sl^k - speed limit at that link. Since the duration of the phase must be at least $minDuration^j$, the actuation must be enabled only after the time passes some specific threshold $T^j_{th} = minDuration^j - T^j_{a-i}$. In other words, only after the time within the phase exceeds T^j_{th} seconds, the actuation gets enabled and the first vehicle must arrive within $minGap$ in order to prolong the phase. If such vehicle exists, the next car has $minGap$ seconds to trigger the TL again, etc. The process stops when either no such vehicle is found or the $maxDuration^j$ is reached.

Phase	TL States	Minimal Duration(s)	Maximal Duration(s)
0	rrrGGGrrrGGG	39	48
1	rrryyyrrryyy	6	6
2	GGGrrrGGGrrr	30	39
3	yyyrrryyyrrr	6	6

Table 1: Traffic light states: groups of three from left to right - North $\rightarrow$ South, West $\rightarrow$ East, South $\rightarrow$ North, East $\rightarrow$ West; r - red, G - green, y - yellow

3.3.1 Performance of a counter

A counter is an advanced detector serving to obtain the necessary information about the downstream traffic. When a vehicle passes a counter, it stores the vehicle's speed and time of crossing (real time of the day) (Table 2).

Speed	Time
v_1	t_1
v_2	t_2
...	...
v_k	t_k

Table 2: Data storage

At the end of every cycle, the data is erased. It is necessary because the vehicles from the previous cycle no longer have an impact on the traffic light actuation and their information is irrelevant. This happens because T_{th} seconds from the beginning of the cycle is sufficient to travel from the counter to the intersection in moderate traffic. Failure to do so implies congestion, which initiates the switch from the SAS to the car-following model. Also, such data storage system does not require a lot of memory and significantly reduces further computational time.

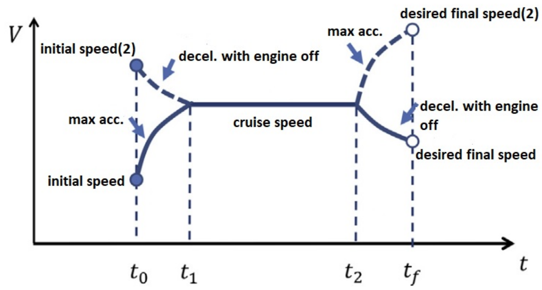
3.4 Speed Advisory System summary

In order to create an efficient predictive algorithm, it is essential to understand what variables and parameters are required by the Speed Advisory System (SAS). Here we are going to use a slightly simplified version of SAS proposed in [1], so we believe it is worth summarizing the main points of the system. According to the research [1], the optimal (in terms of fuel consumption) speed trajectory consists of bang-singular-bang segments: accelerate with maximal acceleration/decelerate with engine off - keep constant speed - decelerate with engine off/ accelerate with maximal acceleration (Fig. 1 (a)). The singular segment is present only at very low speeds, so most of the time the optimal trajectory is bang-bang shaped. This is both hard to implement in real life and very uncomfortable for the drivers, so the paper suggests using a suboptimal speed trajectory: bang-singular (accelerate with maximal acceleration/ decelerate with engine off or with minimal deceleration - keep a constant speed) (Fig. 3 (b)-(c)).

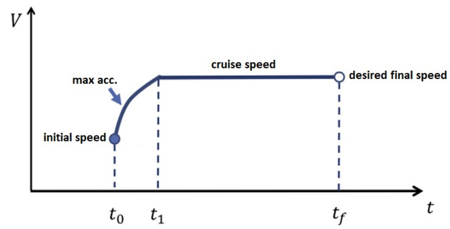
As it can be seen from the graphs, acceleration and deceleration are not considered to be constant. However, in our work, we assume piece-wise constant acceleration/deceleration due to two reasons:

1. Simulation of such complicated dynamics would be unnecessarily difficult, time and resource consuming to perform in SUMO.

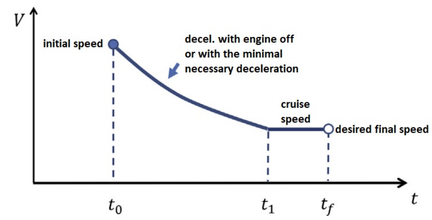
2. According to the original dynamics, for speed under $30^{m}/_{s}$ the engine-off deceleration slightly varies from $0.1460^{m}/_{s^2}$ to $0.1480^{m}/_{s^2}$. Rounding it up to $0.15^{m}/_{s^2}$ would make negligible difference compared to other uncertainties and assumptions. Regarding the acceleration, it is also almost constant and, taking into account the fact that model embedded in SUMO has constant acceleration, we decided to set it to some constant value a_{max} ($2.5^{m}/_{s^2}$ in our model).



(a) A schematic of bang-singular-bang strategy.



(b) A schematic of accelerate-then-cruise case.



(c) A schematic of glide-then-cruise case.

Figure 3: Schematics of optimal (a) and suboptimal (b)-(c) speed trajectories.

Thus, the suboptimal trajectory used in our simulation is one of the following:

1. **accelerate** with a constant maximal acceptable acceleration to a certain desired speed (not acceding the speed limit) and cruise,
2. **decelerate** with an engine off ($\approx 0.15 m/s^2$) to a certain desired speed and cruise,
3. **apply** a necessary constant **breaking** to meet boundary conditions.

3.5 Algorithm

3.5.1 Estimating possible actuation time

The moment a vehicle crosses a counter the algorithm takes action to estimate the time from the beginning of the current phase to the time when that particular vehicle reaches an actuator. The system requires several parameters to perform:

- *Data, obtained via counters* - speeds and times of downstream vehicles - stored at the intersection environment.
- *Current phase* - number from 0 to $n-1$ (if n phases) - TL parameter.

- *Time when the current phase j started* - T^j_{start} - can be received directly from the TL environment or computed based on the statistical data.
- *Traffic light properties* - actuation gap, $minDuration^j$ and $maxDuration^j$ of the current phase, actuation threshold time T^j_{th}, etc. - can be obtained directly from the TL environment.

First step is to compute the required time to travel from a counter to an actuator - applying maximal comfortable acceleration until a vehicle reaches the speed limit and travel the rest of the distance at the speed limit.

$$T^i_{travel} = \frac{sl_k - v}{a} + \frac{d_k - \frac{sl_k^2 - v^2}{2a}}{sl_k}$$

Where:

sl_k: speed limit on the link k

v: vehicle's speed

d: distance between the counter and the actuator on the link k

a: desired acceleration

Second step is to compute the time within the current phase, when the vehicle crossed the counter: $T^j_{phase} = T_{current} - T^j_{start}$

Last step is to compute the the estimated arrival time within the cycle: $T^i_{est} = T^j_{phase} + T^i_{travel}$. Once computed, T^i_{est} gets stored for one cycle for further computation (same as counter data).

NOTE: The discussed computations are expected to be done by the *infrastructure*, more specifically, by the computer installed at the intersection.

3.5.2 "PASS" or "WAIT" procedure

The algorithm assigns labels "PASS" or "WAIT" according to the passing capability of a vehicle. "PASS" label implies that a vehicle is expected to be able to cross the intersection within the current green phase. "WAIT" label, in turn, suggests that the remaining time is insufficient for a vehicle to cross the intersection and advises it to wait for the next green phase. After obtaining the estimated time T_{est}, the algorithm tries to determine whether or not the vehicle is going to make it through the intersection within the current phase. Fig. 4 shows the logic behind that procedure: go through the T_{est} values until finding the one within the interval $[T^j_{th}; T^j_{th} + minGap]$. Then continue going through the remaining T_{est} values checking if the $minGap$ is broken or $maxDuration^j$ is exceeded. If either of those occur, the vehicle receives a "WAIT" label. Otherwise, after reaching the T^i_{est} that belongs to that particular vehicle, it receives a "PASS" label.

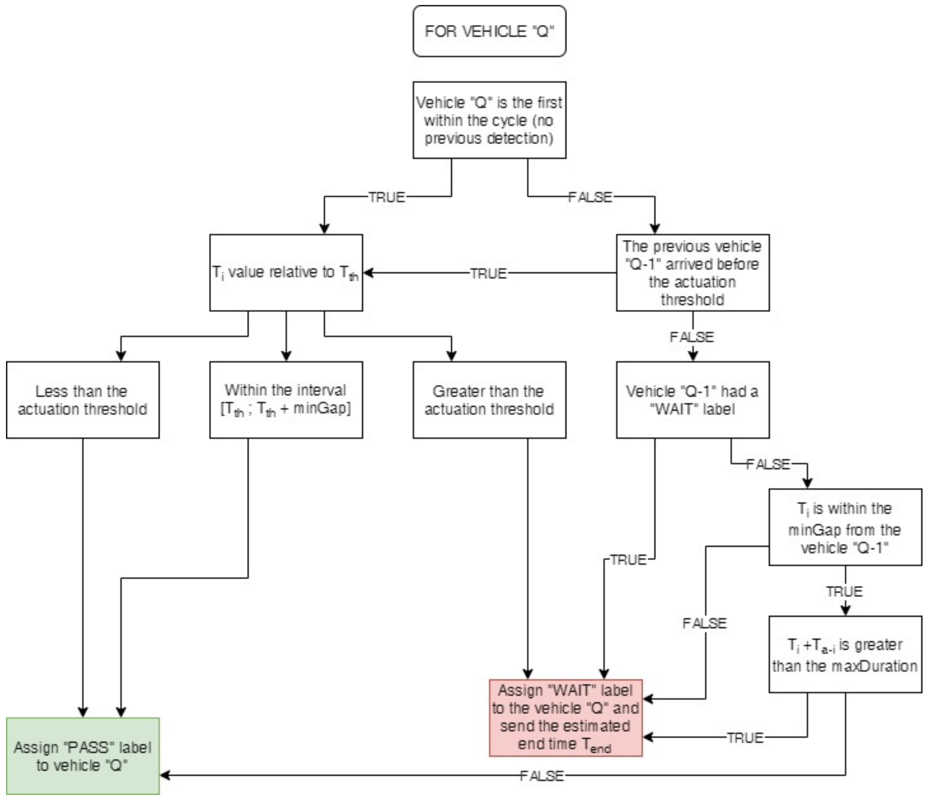


Figure 4: Algorithm Diagram.

If no vehicle with T_{est} within $[T^j_{th}; T^j_{th} + minGap]$ is found and all of the T_{est}s are less than T^j_{th}, the car receives a "PASS" label. If all T_{est} values are greater than $T^j_{th} + minGap$, then the vehicle receives a "WAIT" label.

In this approach, $T^j_{end} = max(T^*_{est} + T^j_{a-i}, minDuration^j)$, where T^*_{est} is the estimated arrival time for the last vehicle with "PASS" label if any.

NOTE: T_{end} is not necessary for computing suboptimal speed trajectories, but might be important for further development of the algorithm, giving vehicles on the "secondary road" (not actuated direction) an opportunity to construct their desired trajectories. Furthermore, T_{end} will be used to compare accuracy with another algorithm.

NOTE: Those computations can be executed by either the *infrastructure* or *vehicles*. Choosing the first option, we face a challenge of transmitting the information to vehicles, because with heavy traffic it is difficult to distinguish cars and find a right recipient for a particular message. On the other hand, by forcing *vehicles* to conduct those calculations, we risk to make a mistake in matching received data with the current vehicle's state due to delays and differences in timing. Further research is needed to address these issues.

3.5.3 Combining predictions and SAS

At this stage, the optimal speed trajectory can be computed. Fig. 5 shows the logic behind decision making inside the algorithm. The basic idea is that vehicles labeled "PASS" should go as fast as possible, since besides fuel consumption we also want to minimize the travel time. On the other hand, being labeled "WAIT" is fundamentally the same as being not labeled at all (crossing a counter during a non-green phase). In both cases an estimate of the residual time until the beginning of the next green phase: $T_{rem} = cycLen_j - T_{incycle} = T_{e-c} - T_{curr}$, (where $T_{incycle}$ - time within the current cycle; T_{e-c} - estimated real time when the current cycle ends; T_{curr} - current real time) is known.

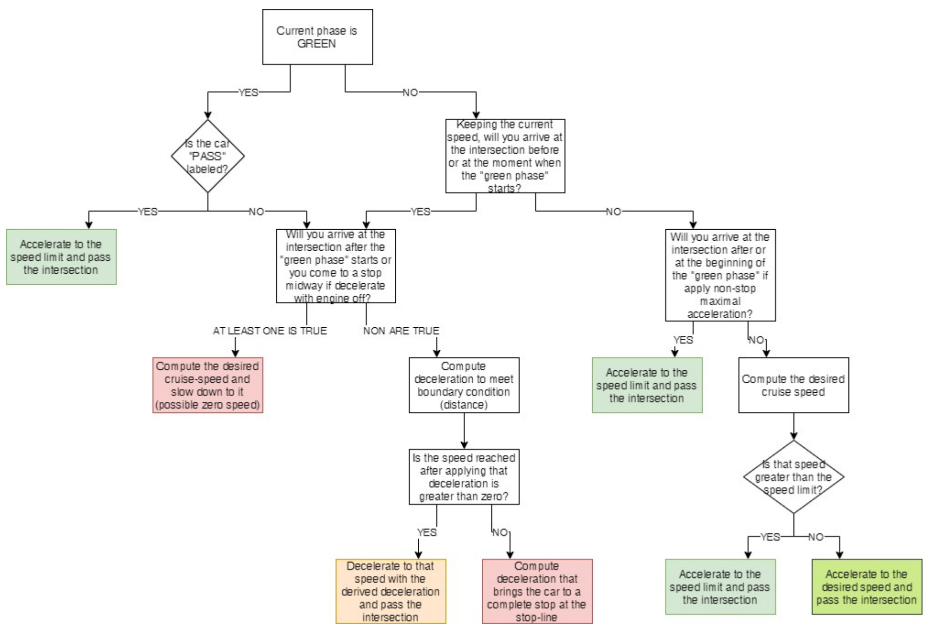


Figure 5: Speed Advisory System Diagram.

NOTE: The discussed part of the algorithm is expected to be performed by each *vehicle* itself.

4 Results

The essential purpose of the algorithm is to provide required information for optimal speed trajectory computation and, thus, fuel consumption reduction. Correct prediction of vehicle's passing capability and accurate phase residual time estimation for Speed Advisory System are the two main objectives of our algorithm. In order to test the efficiency in addressing those concerns, we simulate every vehicle with and without active SAS and compare their intersection-passing-cycle-numbers. Moreover, by computing fuel consumption in those two cases and calculating gas savings we can evaluate the effectiveness of the introduced algorithm. Vehicles without any driver's assistance system and following the Krauss car-following model will be referred as "ordinary" vehicles.

4.1 Simple case

4.1.1 Simulations

To cover a wider range of possible scenarios we ran simulations with three different traffic demands: low demand - $\frac{1}{40}(\frac{veh}{sec})$; medium demand - $\frac{1}{10}(\frac{veh}{sec})$ and high demand - $\frac{1}{3}(\frac{veh}{sec})$. Moreover, for every demand, different penetration rates of SAS-equipped vehicles were tested: 0%, 20%, 30%, 60%, 90% and 100%. Based on the received data, fuel savings for both ordinary and SAS-equipped vehicles were calculated. Such approach allows us to not only see the changes in fuel consumption, but also evaluate the impact vehicles with driver assistance and other traffic participants have on each other.

In addition, we check phase utilization and progression quality of the actuated traffic light. That step is necessary to estimate the influence of vehicles with driver's assistance on traffic state and throughput at the intersection. We want to make sure that introduction of SAS-equipped vehicles does not significantly slow down the traffic flow.

4.1.2 Accuracy of "PASS"-"WAIT" algorithm

Speed Advisory System provides a set of possible suboptimal speed trajectories and our task is to choose the one corresponding to minimal travel time. Thus, the algorithm should not prevent a vehicle from passing the intersection if it has a chance to do so by following some comfortable trajectory. Failure to guide the vehicle through the intersection within the same cycle it would go through without using SAS is considered to be a "mismatch", i.e. error. The results of the conducted simulations are compiled in table 3.

The algorithm showed 100% accuracy in free traffic, telling vehicles to pass the intersection within the earliest possible cycle.

For medium demand, we observed some mismatches, however they are related to ordinary vehicles passing the intersection on yellow, which is forbidden in Speed Advisory System. By softening this constraint, we could achieve a higher accuracy via letting those cars through.

Moreover, the congested traffic brings more uncertainty into simulation, increasing the number of errors. However, 65% of them are related to road segment before the SAS activates, which means that those mismatches are not representative. On the other hand, even if we consider them as "real" mismatches caused by inefficiency of the algorithm, the accuracy is still very high - more than 98%.

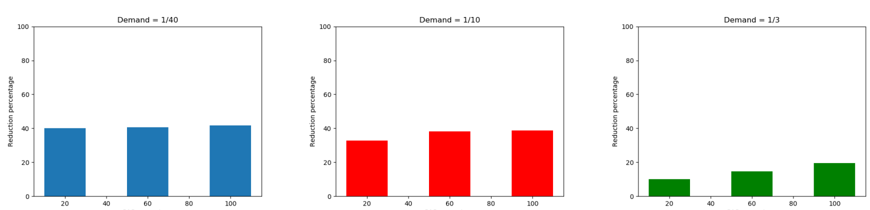
SAS % / Demand ($\frac{veh}{sec}$)		20%	60%	100%	
Low demand($^1/_{40}$)		58	176	285	# Simulated Cars
		0	0	0	# Mismatch
Medium demand($^1/_{10}$)		176	565	960	# Simulated Cars
		1	1	3	# Mismatch
High demand($^1/_3$)		400	1251	2054	# Simulated Cars
		3	20	27	# Mismatch

Table 3: Cycle mismatches for various traffic demands.

4.1.3 Fuel Consumption

Introduction of prediction-based-SAS-equipped vehicles shows a significant reduction in fuel consumption for low and medium traffic demands: 35% - 40% (Fig. 6(a)-(b)). According to Fig. 6(a), the improvement is uniform for all penetration rates of SAS-vehicles in free traffic. That result correlates with a simple logic - the number of cars on the road is insufficient to prevent driver's assistance from following the suboptimal trajectory. However, a slightly smaller reduction for the 20% penetration scenario in medium demand occurs. That outcome can be explained by the fact, that the amount of ordinary vehicles (80% of the entire traffic) in $\frac{1}{10}$ demand is significant enough to influence controlled vehicles, but still not enough to drastically deviate them from their desired speed patterns.

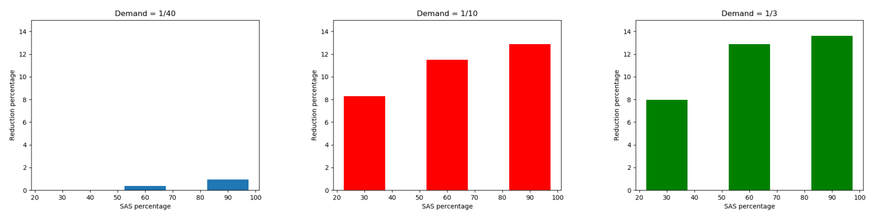
On the other hand, according to Fig. 6(c), congested traffic neutralizes most of the impact of the Speed Advisory System. Since after reaching the end of the queue, vehicles switch from driver's assistance to the car-following model, no fuel is preserved and the resulting savings are much lower than in previous scenarios. The reason of that small reduction is related to the road segment prior to the congestion, where vehicles manage to save up some fuel.



(a) Low demand. (b) Medium demand. (c) High demand.

Figure 6: Fuel consumption reduction for SAS-equipped vehicles in mixed traffic.

Moreover, analyzing ordinary vehicles' fuel consumption in such mixed traffic, we came to the conclusion that having SAS-equipped vehicles on the road causes traffic participants with no driver's assistance system to save up fuel as well. That happens because following a certain suboptimal pace, controlled vehicles make others slow down to the similar speed profile, which unintentionally reduces their fuel consumption. In free traffic, the change is negligibly small - less than 1% (Fig. 7(a)). However, in mildly and highly congested traffic, the reduction is quite significant - ranging from 8.3% to 12.5% and from 8% to 13.2% respectively (Fig. 7(a)-(b)).



(a) Low demand. (b) Medium demand. (c) High demand.

Figure 7: Fuel consumption reduction for ordinary vehicles in mixed traffic.

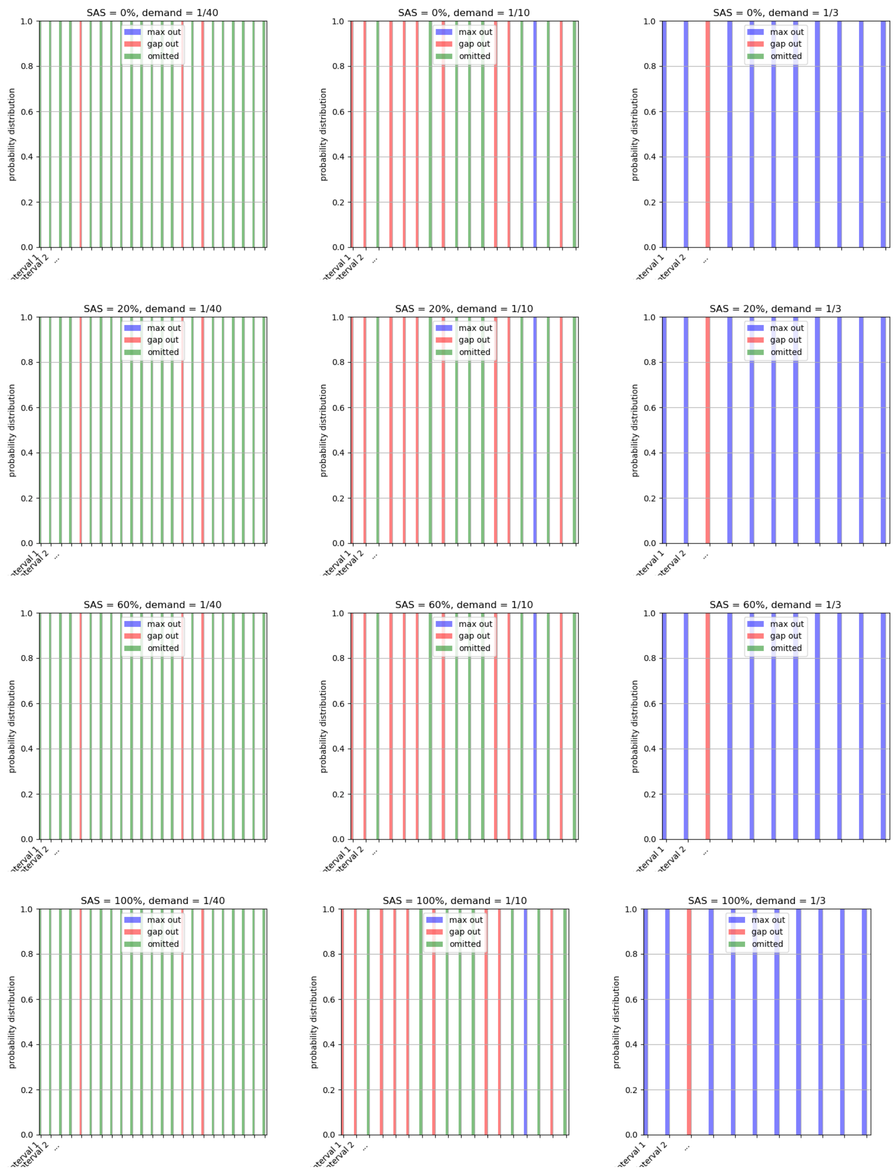
4.1.4 Phase Utilization

In addition to main objectives, we are also interested in some specific performance measures. First of them is phase utilization, which can be effectively characterized by phase termination metric [14]. There are four possible reasons for phase termination, however we focus on three of them ($force - off$ is not considered). An actuated phase can be *omitted* when there is no actuation during the cycle; it can $gap - out$, when the TL was actuated at least once and then the actuation gap was broken and it can $max - out$ when the phase duration reaches it's maximal allowed length. Max-outs indicate that a phase is exceeding capacity, while gap-outs and omits indicate that there is capacity to spare. Our goal is to understand how the introduction of Speed Advisory System influences the capacity utilization, more specifically whether it forces the traffic to exceed the capacity or not.

According to the simulation results (Fig. 6), driver's assistance system leaves the phase utilization unchanged for all levels of traffic demands. Thus, we can conclude that even if the Speed Advisory System does not help utilize the capacity more effectively, it at least does not bring more imbalance into the traffic states.

4.1.5 Progression Quality

Another important performance measure is progression quality. It is relevant to understanding the delay performance of a coordinated movement. In other words, progression quality corresponds to the waiting time in a queue at the intersection due to the arrival-departure patterns. In order to characterize that measure, the proportion of vehicles arriving on green (or simply "percent on



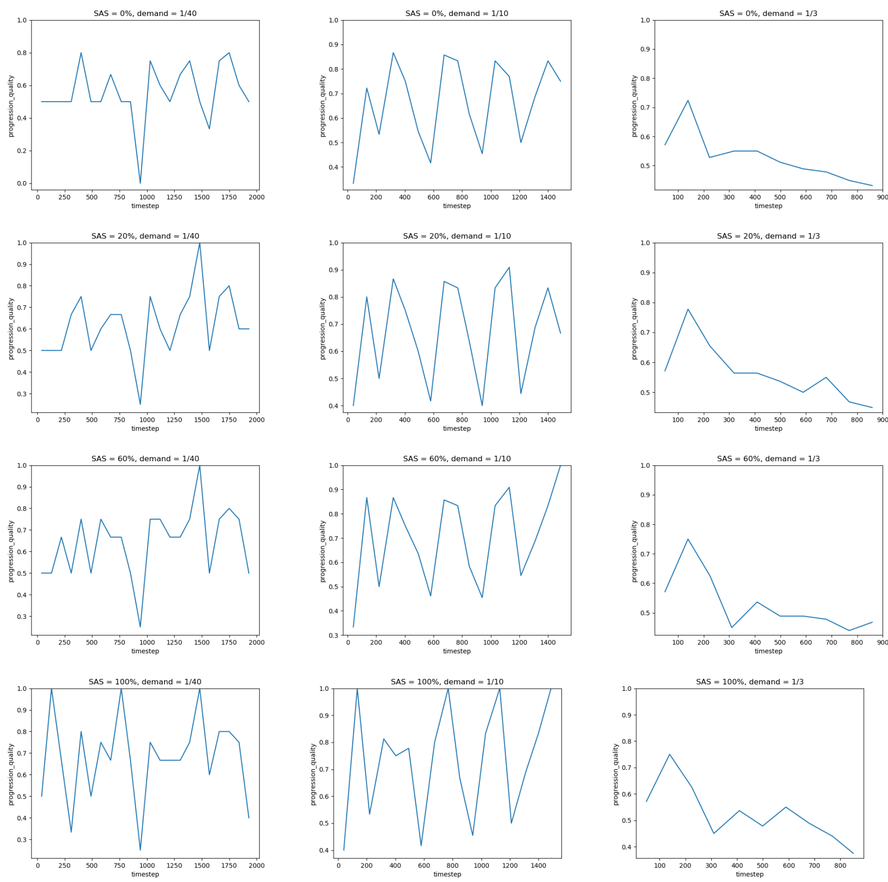
(a) Low demand. (b) Medium demand. (c) High demand.

Figure 8: Phase termination reasons for various demands and SAS percentages.

green", POG) P can be computed.

$$P = \frac{N_g}{N_r + N_g} = \frac{N_g}{N}$$

where N_g - number of vehicle arriving at the intersection during green, N_r - number of vehicles arriving at the intersection during red/yellow and N - total number of arriving vehicles for the entire cycle. The higher the number P is, the less delay vehicles would experience passing the intersection. Thus, the progression quality measure is potentially able to clearly indicate the impact of the prediction-based-SAS introduction on the traffic states.



(a) Low demand. (b) Medium demand. (c) High demand.

Figure 9: Percent on green (POG) for progression quality measure.

According to the results (Fig. 7(a)), in free traffic it is possible to obtain a better progression quality by simply equipping just 20% of vehicles with the driver's assistance system. In that case, we were able to reach a peak of 100% POG, which corresponds to zero delay. Furthermore, making all vehicles use prediction-based-SAS, three peaks of 100% POG were obtained in addition to growth of average and minimal (compare 0 with 0% SAS and 0.3 with 100% SAS) POG. That demonstrates the significant improvement in the progression quality.

In case of medium traffic demand (Fig. 7(b)), the algorithm shows noticeably better results only when all the vehicles on the road are equipped with Speed Advisory System - three peaks of 100% POG compared to zero peaks with no SAS introduced. Moreover, with 60% penetration of SAS-equipped vehicles, even though the outcome is not significantly more efficient, we still can comment on the growth of average POG percentage. For any other portion of controlled vehicles, the progression quality is at least the same as with no SAS-equipped cars at all, which is a valid result.

Finally, congested traffic (Fig. 7(c)), as expected, has a relatively low progression quality and can hardly be improved due to constant indissoluble or slowly dissoluble queues. In these conditions Speed Advisory System is active for a short period of time before the vehicle reaches the end of the queue and switches to car-following model. Thus, the impact is negligible.

4.2 Complex Case

4.2.1 Simulations

Due to the complex structure of the network, we managed to obtain different traffic demands on different intersections within a single setting. Thus, unlike the simple case, the testing has no need to be split into separate simulations with changing congestion levels. However, the variance of SAS-equipped vehicle penetration is preserved, though reduced to 0%, 50%, 100%. Moreover, in order to analyze robustness of the algorithm, some assumptions about traffic participants were reconsidered. Now, three possible options for vehicles' accelerations are available: predetermined and fixed acceleration ($2.5\frac{m}{s^2}$), random but known acceleration ($uniform(2, 3.5)$) and random unknown acceleration ($uniform(2, 3.5)$). In the latter case the algorithm assumes that all vehicles have an average acceleration ($2.75\frac{m}{s^2}$) and makes computations based on that assumption.

In addition, we compare phase residual time prediction procedure with the one used in [15] - denoted "algorithm B" further in the paper. Thus, we simulate the discussed scenarios with our algorithm and then with algorithm B to see the improvement in prediction accuracy.

4.2.2 Accuracy

Similar to the simple case, first we want to analyze the accuracy of "PASS" algorithm. Table 4 shows the precision percentages for 9 intersections. Green rows indicate a free traffic, yellow - moderate demand and red - congestion. According to the data, the algorithm performs with a precision of at least 99% in low demand for all possible acceleration scenarios. Having a little number of vehicles on the road implies minimized interference rate, which, in turn, insures precise calculations.

Medium and high demands cause more issues in correct prediction of passing capability, but still show impressive results: at least 89% accuracy in worst case and $\approx$ 95% in average. Vehicles interacting with each other are forced to slow down, accelerate, stop, etc. causing rare miscalculations. Moreover, intersection's geometry can have an impact on accuracy. Intersection 6 has relatively short incoming links, i.e. vehicles travel from counters to actuators within a very short amount of time. Since the algorithm is a discrete time mechanism, that travel time is rounded up or down, which results in mistakes.

SAS % / Intersection #	50%	100%	
Intersection 1	100%	99.4%	Fixed known acc
	100%	99.4%	Random known acc
	100%	99.4%	Random unknown acc
Intersection 2	99.6%	99.8%	Fixed known acc
	99.6%	99.5%	Random known acc
	99.3%	99.6%	Random unknown acc
Intersection 3	98.7%	99.1%	Fixed known acc
	99%	99.2%	Random known acc
	99%	99.2%	Random unknown acc
Intersection 4	96.4%	98.7%	Fixed known acc
	98.3%	97.5%	Random known acc
	98.1%	97.8%	Random unknown acc
Intersection 5	98.5%	98.6%	Fixed known acc
	97.5%	97.5%	Random known acc
	97.7%	97.8%	Random unknown acc
Intersection 6	89.3%	91.3%	Fixed known acc
	91.1%	91.3%	Random known acc
	91.2%	91.3%	Random unknown acc
Intersection 7	93.4%	92.5%	Fixed known acc
	93.2%	94.9%	Random known acc
	94.6%	94.2%	Random unknown acc
Intersection 8	97.9%	99.7%	Fixed known acc
	99.4%	99.7%	Random known acc
	99.5%	99.7%	Random unknown acc
Intersection 9	98%	98.1%	Fixed known acc
	99.3%	99.2%	Random known acc
	98.5%	99.3%	Random unknown acc

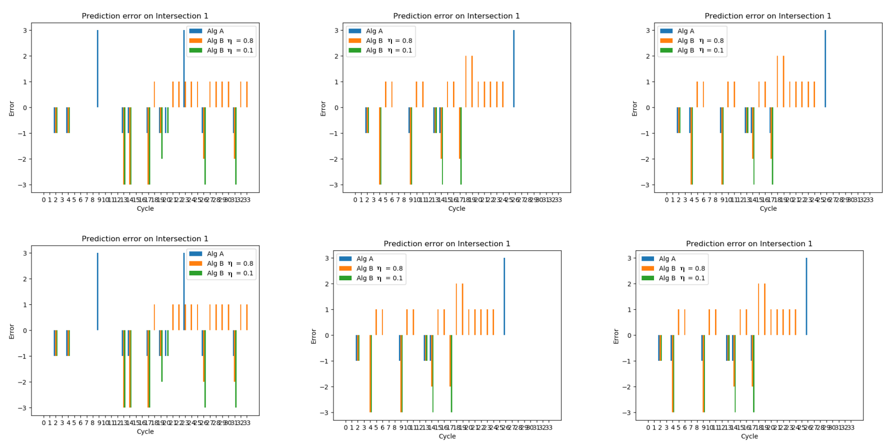
Table 4: "PASS" algorithm prediction accuracy

The second important outcome is obtained after the comparison with the Algorithm B. Since the essential goals of both procedures are different (predicting passing capability in our case and estimating remaining time in case of algorithm B) and mechanism presented in this paper does not explicitly compute the remaining time, we have to artificially extract the estimation from compiled vehicle data.

Note: every vehicle j also receives an estimated phase termination time T^j_{end}. For "PASS" labeled vehicles such time does not necessarily equal to the actual termination time, because upstream traffic can potentially extend the green phase, however, for "WAIT" labeled vehicles T^j_{end} serves as a good final estimate for the phase duration.

Thus, using the T^j_{end} from the last "PASS" labeled (or any "WAIT" labeled) vehicle gives us a valid comparison material for accuracy check. Figures 10-12 represent the absolute values of estimation deviations (errors) (in seconds) of Algorithm B and our Algorithm A from the actual registered data. Positive and negative values of errors correspond to overestimation and underestimation of the phase duration respectively, i.e. the algorithm's prediction is greater or smaller than the actual phase length.

Note: absolute errors of less than 3 seconds can be considered to be insignificant, due to several assumptions and time discretization.

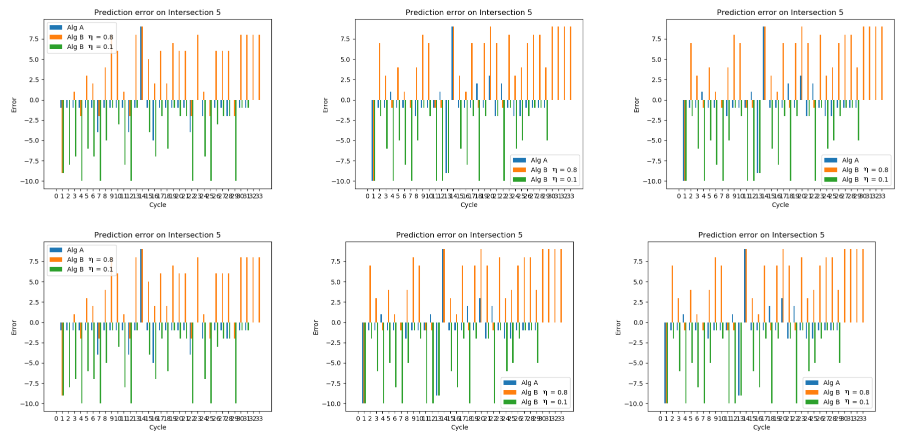


(a) Known fixed acc. (b) Known random acc. (c) Unknown random acc.

Figure 10: Absolute prediction error on intersection 1 for Algorithm A and Algorithm B with η=0.8 and η = 0.1

Figure 10 corresponds to the intersection 1 with low traffic demand. According to the outcome, all three methods show high accuracy levels for any simulated scenario. Having a few vehicles on the road implies low probability of triggering the traffic light actuation, which means that historical phase duration is almost always at minimal duration. Thus, CDF is almost constant and Algorithm B's prediction is relatively accurate. Although, most of the errors can be view as insignificant, our algorithm managed to predict the phase duration precisely more often than the Algorithm B.

Medium traffic demand is represented by intersection 5 (fig. 11). According to the histogram, our algorithm performs much more precisely than Algorithm B. Most of the errors do not exceed 2 seconds and occur less frequently. In

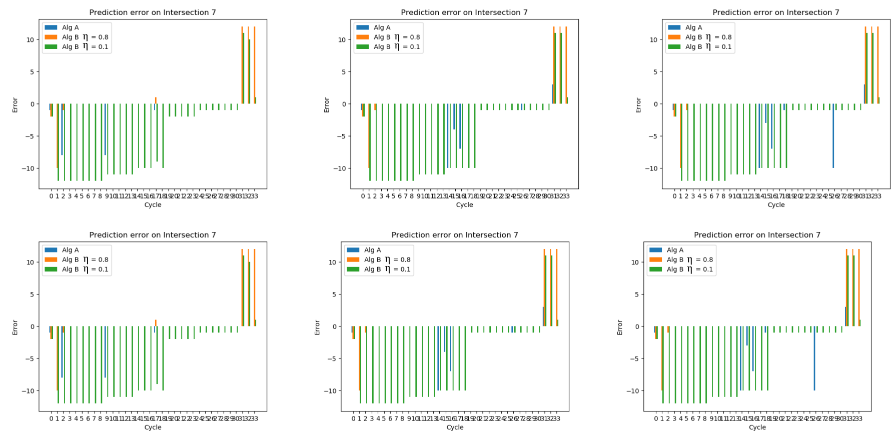


(a) Known fixed acc. (b) Known random acc. (c) Unknown random acc.

Figure 11: Absolute prediction error on intersection 5 for Algorithm A and Algorithm B with η=0.8 and $\eta = 0.1$

the cases with random accelerations, appearance of greater errors is observed. That is due to the fact that the algorithm relies on additional assumptions and guesses, which make it less accurate. However, as it can be seen, those errors are rare and do not have a significant influence on traffic flow. Regarding algorithm B, setting the reliability level η to 0.8 results in heavy overestimation for many cycles (up to 8 seconds errors). On the other hand, with $\eta = 0.1$ we obtain a serious underestimation of the phase length. The reason is that in moderate traffic, phase duration varies from cycle to cycle with possibly high deviation. One additional data point does not make a big difference for the CDF function, thus, estimation for cycle k will most likely be very similar to the one for cycle $k-1$ even if their actual durations differ dramatically.

Congested traffic data is compiled into the figure 12. Intersection 7 has a very high demand and, as it will be shown later, its green phase is usually terminated due to max-out, i.e. reaching maximal allowed phase length. Initial historical data for algorithm B corresponded to medium demand. Thus, the setting with $\eta = 0.1$ took some time to receive enough new data points to shift CDF and output accurate estimation. During that time we observe extremely poor performance with drastic underestimations for more than 10 seconds. Algorithm A and Algorithm B with reliability level at 0.8, on the other hand, show impressive results with 100% accuracy for most cycles. Our algorithm allowed several significant miscalculations, however they are rare and related to the randomness of vehicles' accelerations. The last several cycles correspond to congestion dissolution and thus to phase duration reduction. Algorithm B for both reliability levels cannot accommodate to sudden change of pattern and continues to predict maximal duration, which is not correct anymore. Algorithm A, in comparison, manages to give a correct prediction independent of



(a) Known fixed acc. (b) Known random acc. (c) Unknown random acc.

Figure 12: Absolute prediction error on intersection 8 for Algorithm A and Algorithm B with η=0.8 and $\eta = 0.1$

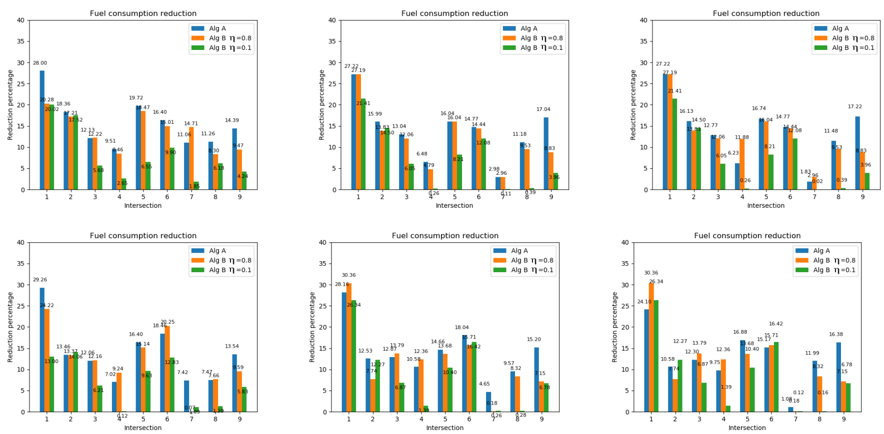
change in traffic state. That is an important outcome: statistical approach with "reliability level" concept shows insufficient results in the case of changing traffic demands, while real-time algorithm processes only the current data, independent of previous cycles and deals with the changes well.

4.2.3 Fuel Consumption

One of the main purposes of Speed Advisory System is fuel consumption reduction. By following suboptimal trajectories, vehicles manage to reduce idling at intersections and increase energy efficiency. Derivation of such trajectories depend on the phase length prediction. We already compared the accuracy of those estimations, and now it is important to calculate the resulting fuel consumption. Figure 13 presents a comparison between the two algorithms - A and B (with $\eta = 0.8$ and $\eta = 0.1$). According to the histograms, in average our algorithm performs better than Algorithm B for both tested reliability levels. Setting η to 0.1 results in much smaller fuel consumption reduction for all intersections and traffic demands. Moreover, choosing the reliability level to be 0.8 for Algorithm B gives a similar pattern as applying Algorithm A. Switching between scenarios with different acceleration settings and SAS-vehicle penetrations does not give a significantly distinct picture. Some intersections benefit more from Algorithm B in rare cases in terms of fuel consumption, however the difference is small.

Next step is to analyze our algorithm's performance independent of any other procedure. As it can be seen, we managed to achieve good result for low demand traffic situation - up to 29% energy savings. Moreover, medium demand and congested scenarios also benefit from the introduction of prediction-based-SAS

- up to 18% and 7% fuel savings respectively. Those results correlate with the simple case simulations. Speed Advisory System is most effective in terms of fuel consumption in free and moderate traffic, because, as stated earlier, vehicles actually follow the given advice due to low interaction rates. Congestion, on the other hand, forces traffic participants to switch from SAS to car-following model and loose most of the impact driver's assistance could have on energy savings.



(a) Known fixed acc. (b) Known random acc. (c) Unknown random acc.

Figure 13: Fuel consumption reduction for Algorithm A and Algorithm B with η=0.8 and $\eta = 0.1$

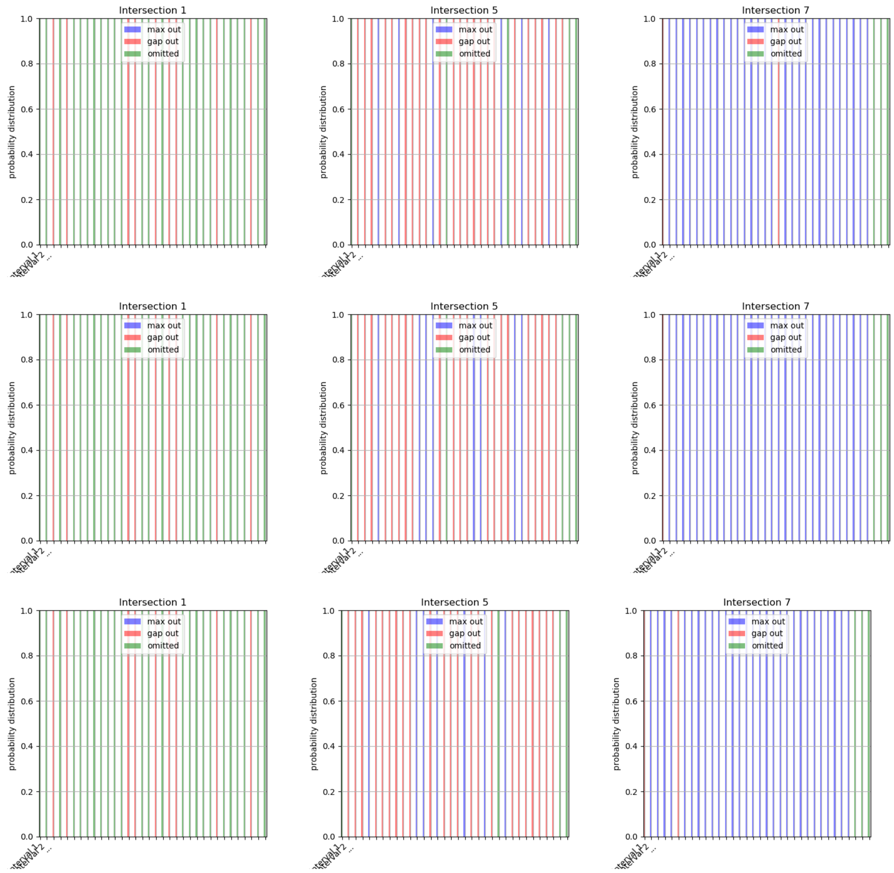
4.2.4 Phase utilization

The picture is similar for all three acceleration scenarios (known fixed, known random and unknown random), so we present the results for only known fixed acceleration setup. The data from intersections 1, 5 and 7 is compiled into figure 14. The first row corresponds to no cars with driver's assistance system on the road, the second and third ones - 50% and 100% of SAS-equipped vehicle penetration levels respectively.

Similarly to the simple case, phase utilization is barely affected by the introduction of speed advisory system. Although the difference is insignificant, it is important to highlight that driver's assistance system keeps the utilization at at least same level as using no SAS at all.

4.2.5 Progression Quality

Progression quality comparison is presented in figure 15. Each graph contains outcomes for 0%, 50% and 100% SAS-vehicle penetration. All three tested



(a) Intersection 1 - Low demand (b) Intersection 5 - Medium demand (c) Intersection 7 - High demand

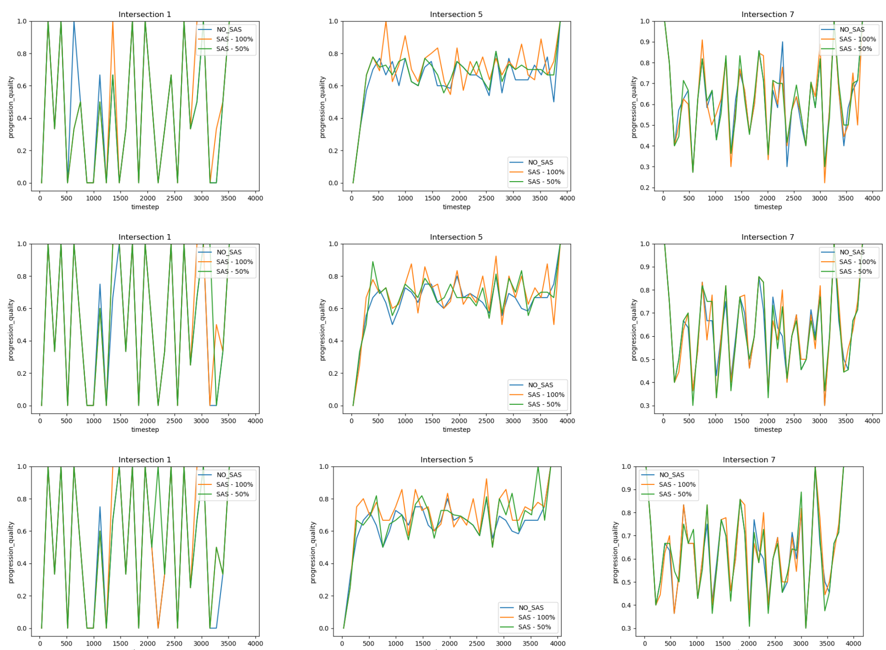
Figure 14: Phase utilization comparison for different SAS-vehicles penetration levels.

acceleration options are compiled by rows in the following order (starting from the top): known fixed, known random and unknown random accelerations.

According to the results (fig. 15(a)), progression quality for low traffic demand slightly benefits from the introduction of SAS-equipped vehicles. Being able to achieve 100% POG more often implies that during some cycles we got rid of delays and slightly improved the traffic state. The important detail worth of mentioning is that reaching 50% controlled cars penetration on the road makes almost the same impact as equipping all vehicles with Speed Advisory System.

In case of moderate traffic (fig. 15(b)), the algorithm managed to achieve more impressive results. Both 50% and 100% SAS penetration rates resulted in noticeable improvements in progression quality by at least 10% and 8% in average respectively. Moreover, in the case of known fixed accelerations the maximal metric's growth for 1 cycle reached 20% for 100% penetration rate and in the case of unknown random acceleration - 33% for 50% penetration rate. Those results indicate that real-time prediction algorithm was able to provide accurate enough information to significantly improve traffic conditions at intersections in medium demand scenario.

Similarly to low demand, congested traffic (fig. 15(c)) is not affected much by the Speed Advisory System in terms of progression quality. On average, the impact on the metric does not exceed 2%, which correlates with the simple case results. The only meaningful conclusion we come to is that the system performance does not suffer from implementation of prediction-based algorithm and driver's assistance system.



(a) Intersection 1 - Low demand

(b) Intersection 5 - Medium demand

(c) Intersection 7 - High demand

Figure 15: Phase utilization comparison for different SAS-vehicle penetration levels.

5 Conclusion

The thesis describes a real-time prediction algorithm that ties together two systems: Speed Advisory and actuated traffic light. The primary objective of the algorithm is to determine a passing capability of all “green vehicles”[1] within the current cycle. The obtained information can be used in driver’s assistance system for building suboptimal speed trajectories to minimize fuel consumption. In addition, the mechanism is able to estimate the current green phase duration for any further computations.

Conducted simulations showed impressive results: at least 89% prediction accuracy in worst case and over 95% correct estimations in average. The algorithm outperformed a statistic-based approach for both reliability levels of 0.8 and 0.1 for all traffic demands, SAS-vehicles penetrations and acceleration scenarios. Substitution of the CDF-approach with the one presented in this thesis can potentially allow to implement the algorithm [15] on primary actuated road.

Significant fuel consumption reduction for both SAS-equipped (up to 30%, 20% and 7% for low, medium and high demands respectively) and ordinary (up to 14%) vehicles is another important outcome. Introduction of real-time prediction algorithm forced disconnected vehicles to save up fuel by indirectly changing their speed profiles. In addition, progression quality and phase utilization managed to stay untouched or even be improved by the implementation of the mechanism for most of the scenarios.

Several issues need to be addressed in further studies. The accuracy of algorithm might suffer from the fact, that we do not consider queues and congestions. In that case, vehicles are forced to slow down and switch to car-following model even if they are following a suboptimal speed trajectory. That causes idling and increase of fuel consumption compared to suboptimal scenario. Moreover, the algorithm has been tested only in SUMO, which does not fully reflect real world conditions. Thus, we are not able to predict the behavior and effectiveness of the algorithm once implemented in reality.

The algorithm might be viewed as a step towards a potentially high-impact system: a comprehensive intersection infrastructure that incorporates various road sensors, driver’s assistance compatible software, hazardous behavior prevention and safety regulation.

6 References

[1] N. Wana, A. Vahidi, A. Luckow, ”Optimal speed advisory for connected vehicles in arterial roads and the impact on mixed traffic”

[2] E. Koukoumidis, L.-S. Peh, and M. R. Martonosi, “Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory,” in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, ACM, 2011, pp. 127–140.

[3] B. Asadi and A. Vahidi, ”Predictive Cruise Control: Utilizing Upcoming Traffic Signal Information for Improving Fuel Economy and Reducing Trip Time”, *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, May 2011.

[1] vehicles that crossed a counter when the actuated TL phase was green

[4] C. Wang and S. Jiang, “Traffic signal phases’ estimation by floating car data,” in *12th International Conference on ITS Telecommunications(ITST), 2012*, IEEE, 2012, pp. 568–573.

[5] V. Protschky, C. Ruhhammer, and S. Feit, “Learning traffic light parameters with floating car data,” in *18th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2015, pp. 2438–2443.

[6] S. A. Fayazi, A. Vahidi, G. Mahler, and A. Winckler, “Traffic signal phase and timing estimation from low-frequency transit bus data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp.19–28, 2015.

[7] S. A. Fayazi and A. Vahidi, “Crowdsourcing phase and timing of pre-timed traffic signals in the presence of queues: algorithms and back-end system architecture,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 870–881, 2016.

[8] V. Protschky, S. Feit, and C. Linnhoff-Popien, “Extensive traffic light prediction under real-world conditions,” in *2014 IEEE 80th Vehicular Technology Conference (VTC Fall)*, 2014, pp. 1–5.

[9] V. Protschky, K. Wiesner and S. Feit, ”Adaptive Traffic Light Prediction via Kalman Filtering” in *2014 IEEE Intelligent Vehicles Symposium (IV)*, June 8-11, 2014.

[10] S. Ibrahim, D. Kalathil, R. O. Sanchez and P. Varaiya, ”Estimating Phase Duration for SPaT Messages” for *IEEE Transactions on Intelligent Transportation Systems*.

[11] J. Sun and L. Zhang, “Vehicle actuation based short-term traffic flow prediction model for signalized intersections,” *Journal of Central South University*, vol. 19, no. 1, pp. 287–298, 2012.

[12] S. Chen and D. J. Sun, “An improved adaptive signal control method for isolated signalized intersection based on dynamic programming,” *IEEE Intelligent Transportation Systems Magazine*, vol. 8, no. 4, pp. 4–14,2016.

[13] S. Coogan, C. Flores, and P. Varaiya, “Traffic predictive control from low-rank structure,” *Transportation Research Part B: Methodological*, vol. 97, pp. 1–22, 2017

[14] Day, C. M., D. M. Bullock, H. Li, S. M. Remias, A. M. Hainen, R. S. Freije, A. L. Stevens, J. R. Sturdevant, and T. M. Brennan, ”Performance Measures for Traffic Signal Systems: An Outcome-Oriented Approach”. *Purdue University, West Lafayette, Indiana*, 2014. doi: 10.5703/1288284315333.

[15] C. Sun, X, Shen, and S. Moura, ”Robust Optimal Eco-driving Control with Uncertain Traffic Signal Timing”. arXiv:1802.07192v1 [math.OC] 18 Feb 2018.