SIGNAL CONTROL USING VEHICLE LOCALIZATION PROBE DATA

Dr Simon Box

Research Fellow

University of Southampton

Dr Benedict Waterson

Lecturer

University of Southampton

Abstract

This paper presents a simulation test bed and methodology for evaluating urban signalized junction control algorithms that use localization probe data from all vehicles in the local area. The simulator is based on SIAS Paramics micro-simulation software with bespoke software modules built on top for automatic network generation, localization data processing and signal control. Localization algorithms tested use a hierarchical structure of *auctioning agents*. Early tests of control algorithms on an isolated signalized junction indicate performance that compares favourably with the MOVA algorithm using inductive loop data.

1 Introduction

Recently a number of large European Commission funded projects (CVIS [Kompfner, 2008], SafeSpot[1], Coopers[2]) have focused on the development of technologies and standards for Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communications systems. This has led to common European protocols being set for this type of communications (IEEE 802.11 (WAVE) and IEEE 802.11p). Some of the most important data that may be communicated between vehicles and infrastructure are localization data, that is dynamic estimates of the vehicle's position. Localization technologies that can provide these data such as Global Positioning System (GPS) receivers are already commonplace in many vehicles, in use for navigation.

Urban signalized junction control is a task that requires sensors to monitor the state of the network, a processing system to analyse these data and make control decisions and traffic lights to implement the control. Currently sensors that are commonly used in signalized junction control are inductive loops [Sreedevi, 2005], microwave emitter/detectors [Wood et al., 2006] and traffic monitoring cameras. Examples of automated control algorithms that are currently in use to process data from these sensors and set signal timings are MOVA [Vincent and Peirce, 1988] for isolated junctions and SCOOT Hunt et al. [1982], which can coordinate multiple connected junctions.

All the sensors currently used in urban signal control collect *census* data, that is counts of vehicles passing a specific point in space. The type of data that can be collected using on board vehicle localization sensors is *probe* data and this different type of data can present a fundamentally different view of the state of the network [Rose, 2006]. Probe data allow an analysis of the system that tracks each vehicle individually and can provide a higher resolution of position data.

[1] http://www.safespot-eu.org/

[2] http://www.coopers-ip.eu/

Research that examines the use of V2I communications and localization systems in signalized junction control is already under way. The iBus project [Hounsell et al., 2008] uses localization systems on London buses to give them priority at signalized junctions.

In this paper we present a computational simulation system that can model the hypothetical scenario of urban signalized junction control using localization probe data from all vehicles in the local area. The resulting simulator constitutes a test bed for the development of new algorithms for signalized junction control based on localization probe data. This paper also contains results from tests on two basic control algorithms that are simulated on an isolated junction. The performance of these algorithms is compared directly with the MOVA algorithm sing simulated inductive loop data.

2 Simulator Architecture

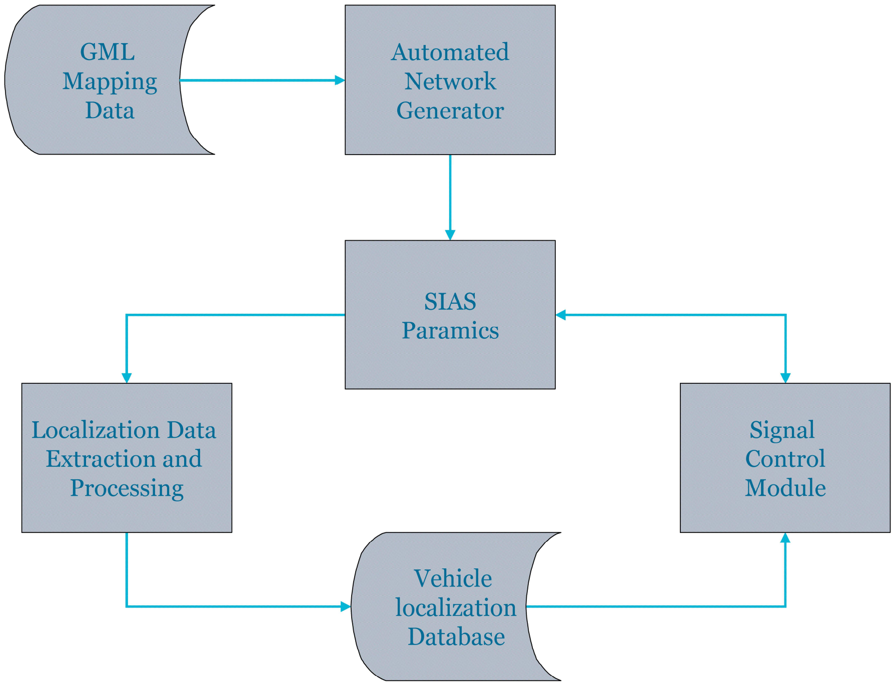

Figure 1: Block diagram showing the simulation software architecture

Figure 1 shows the architecture of the simulation test bed developed in this research. At the centre of this is a module for simulating vehicle movements and interactions through signalized junctions at the individual vehicle level (microscopic). The approach used in this research was to employ an existing commercially available microscopic traffic simulator (SIAS Paramics) to fill the roll of this module. The main advantage of this approach is expedience, allowing us to develop a test bed for control strategies relatively quickly. A further advantage is that Paramics generates rich graphical visualizations, which is a useful tool for sanity checking and observing the progression of simulations. A disadvantage of this approach is that Paramics is a “black box” in our simulator where we are not aware of all the processes occurring between the input and output of data. The calibration setting for Paramics used in this research are the default setting in Paramics version 2007.1. Care must be taken with this approach that control strategies developed in the test bed are not too highly tuned to behaviour in Paramics that may not be representative of the real world. Furthermore any strategies developed will require real world validation in order to confirm their efficacy.

As can be seen in Figure 1 The simulation test bed has three additional bespoke modules that are built around Paramics. The Network Generator module is used to automatically encode the structure of road networks in Paramics using a database of containing mapping data. The Localization Data Extraction and Processing module interrogates the Paramics simulation to obtain localization data for all the vehicles in the simulation. The processed data are then stored in a database. The Signal Control module can extract relevant localization data from the database and use these to make decisions about signal control which are implemented directly in the Paramics simulation. Prototype algorithms for signal control can be ported in and out of this module for testing. The Paramics module, the localization module and the signal control module are all synchronized to allow real time simulation of signal control using localization probe data.

The following three sections of this paper describe each of the three bespoke modules in more detail.

3 Network Generator Module

It is anticipated that this research will require complex and perhaps large road networks to be represented in Paramics. While it is not necessary for these networks to be models of existing road infrastructure it is important that they are representative. In practice this is best achieved by modelling real examples of road infrastructure as accurately as possible.

The Paramics software provides a graphical user interface for the user to build models of networks. This interface requires the user to input a scaled image of of the network and trace over it using the mouse to add nodes and links and define the nature of the network. Figure 2 shows a Paramics visualization of a simple un-signalized T-junction. The minimum number of mouse clicks required to generate this model is 267; the user time required is somewhere between 20 minutes and 1 hour depending on experience. Therefore building large networks in Paramics can be time consuming.

The Network Generator Module was built to overcome this problem by generating network models automatically. The input data that describe the network are Geographic Mark up Language (GML) data from the Ordinance Survey (OS) MasterMap[3] project. The integrated transport network layer in the MasterMap provides the detailed structure of the road network and corresponding metadata layers provide details on the types of road, number of lanes, one way sections and other information.

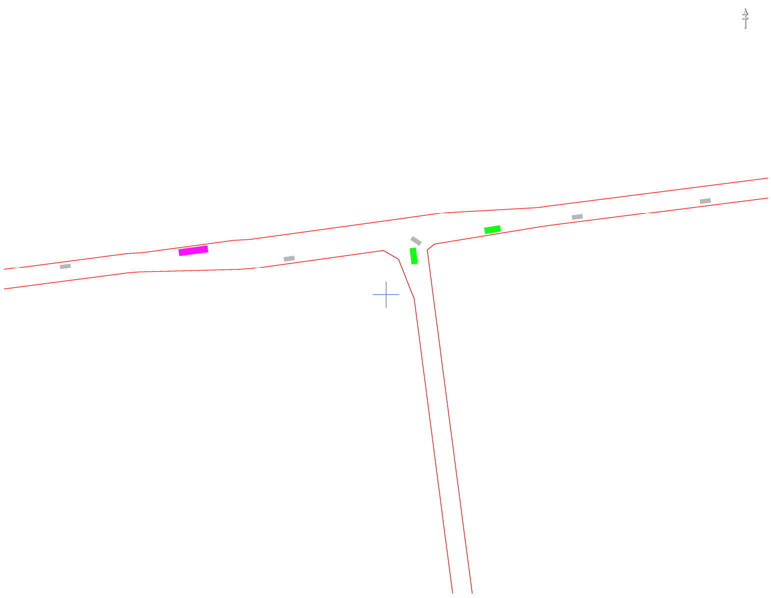

Figure 2: Paramics model of an un-signalized T-junction, showing vehicles in motion

The user of the software module is required to enter four numbers , which are the latitude and longitude of the North-East and South-West corners of a square coving the area of interest. The module will then extract the relevant GML data for that area and automatically

[3] http://www.ordnancesurvey.co.uk/oswebsite/products/osmastermap/

program the corresponding Paramics network. This enables potentially very large networks to be built in a few seconds. Figure 3 shows a large Paramics network of the centre of Southampton that was generated in this way. This figure demonstrates what is possible using the Network Generator however it is unlikely that a network of this size will be used in this research. Figure 4 shows a model of the road network in the Highfield area of Southampton which contains a chain of five linked signalized junctions. This is representative of the type of networks that will be used for simulations in this research in the future.

Figure 3: Large Parmics network of Southampton city centre

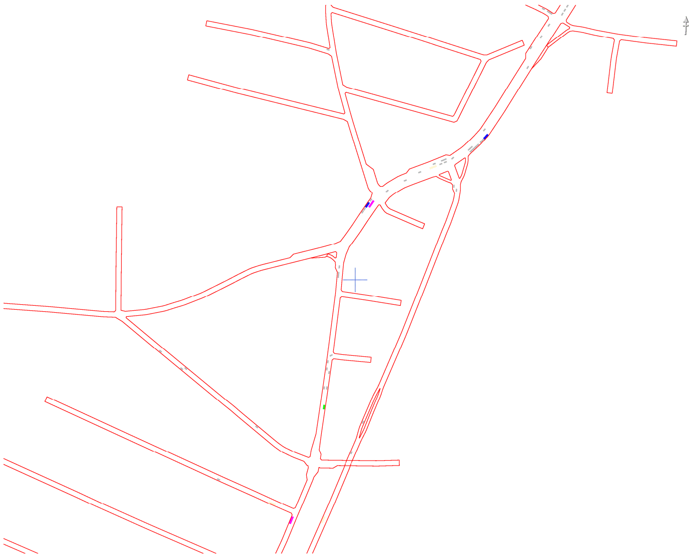

Figure 4: Paramics network of linked signalised junctions in the Highfield area

The automatic network generation process is not perfect and it still requires the user to check the generated network for errors and correct them where necessary and to add additional data that are not available in GML such as signal timings or the positions of inductive loops. Part of the ongoing research includes working on ways to add these data automatically using other sources.

To perform a final check of the network model we use the Google Street View tool (Figure 5) which allows us to travel virtually through the example network and check details such as geometry, turn priorities and sensor positions.

Figure 5: Image from Google Street View

4 Localization Data Module

There are a number of on-board vehicle technologies that can provide dynamic data on vehicle position. These include mobile telephone, or cellular network localization [Kos et al., 2006], Global positioning system (GPS) [Trimble, 2007], inertial measurement systems (IMU) [OXTS, 2009], laser range-finding systems (LIDAR) [Levinson et al., 2007], and computer vision systems [Wang et al., 2007]. In addition to these hardware technologies other software technologies can be employed to improve localization estimates. These include map matching software, which constrains the vehicle's position to the road network [Li and Leung, 2003] and Bayesian recursive filtering techniques, such as the Kalman filter [Grewal et al., 2001]. The latter allow data from more than one sensor and data from other sources such as dynamic data and vehicle control data to be fused to provide a probabilistic estimate of position.

The performance of localization systems is a function of positioning accuracy, frequency of position measurements and reliability (e.g. latency). Figure 6 from Box and Waterson [2009] shows a performance comparison between a number of different localization systems. The diameter of the circles is proportional to the positioning accuracy and the circle's centres are plotted on logarithmic axes of cost versus frequency. This shows a clear relationship between cost and performance and also indicates the level of performance provided by some current localization systems.

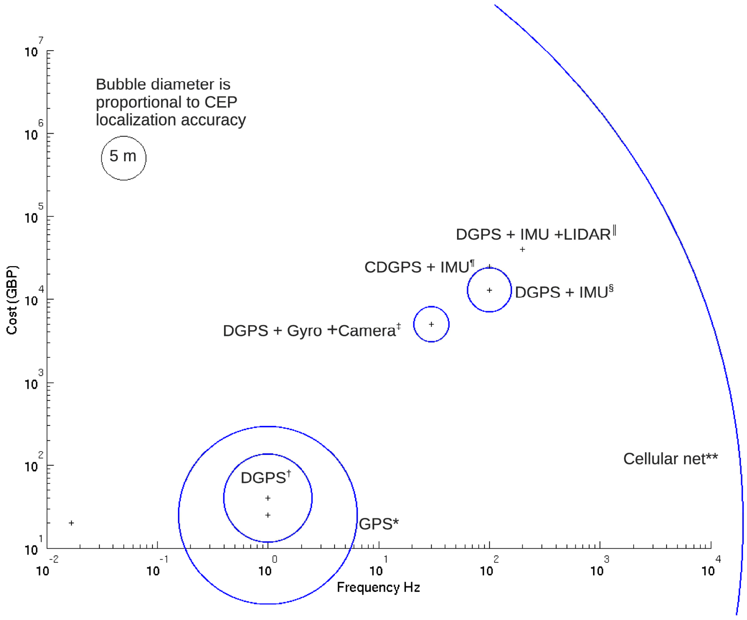

Figure 6: Comparison in the performance and cost of some example localization systems, several of the systems use more than one localization sensor in tandem

4.1 Stochastic Simulation

As shown in Figure 1, the localization module samples vehicle position data from Paramics. These data are perfectly accurate at the time of sampling. This is unrepresentative of the data that would be obtained from a real localization system. Therefore the localization module must process these data to make them more realistic. To this end the position data obtained from Paramics ($\mathbf{x}_P$) are made stochastic by the addition of Gaussian noise (equation (1)).

$$\mathbf{x}_S = \mathbf{x}_P + \boldsymbol{\varepsilon}; \text{ where } \boldsymbol{\varepsilon} \sim \boldsymbol{N}(\mathbf{0}, \boldsymbol{\Sigma}) \tag{1}$$

where $\mathbf{x}$ is a two dimensional vector describing the vehicle's position, $\boldsymbol{\varepsilon}$ is sampled from a zero mean two dimensional Gaussian distribution with covariance $\boldsymbol{\Sigma}$. The covariance matrix $\boldsymbol{\Sigma}$ is chosen to representative of the performance of a given localization system, such as one of the examples given above. Thus the performance of signal control systems can be tested for different localization systems and different levels of localization performance.

Similarly the signal control module, which receives the vehicle position data can either interpret the position data deterministically using $\mathbf{x}_S$, or stochastically taking the probability of position P ($\mathbf{x}_S$) to be Gaussian and centred on $\mathbf{x}_S$ (equation (2)).

$$P(\mathbf{x}_S) = \boldsymbol{N}(\mathbf{x}_S \mid \mathbf{0}, \boldsymbol{\Sigma}) \tag{2}$$

5 Signal Control Module

A signalized junction controller that uses localization probe data from all vehicles in the local area may have to process significant amounts of data in order to set signal timings. Previous research on signal control strategies, where a large amount of loop data needs to be processed, has demonstrated the advantage of a hierarchical agent structure [Choy et al., 2003]. Here individual software agents process small amounts of the raw data, which they then pass on in a significantly refined form to another agent above them in the hierarchy. In this research we have adopted an agent hierarchy very similar to the one presented in Choy et al. [2003], the structure is shown in Figure 7.

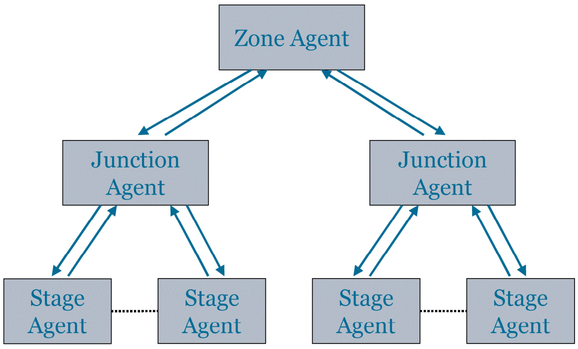

Figure 7: Structure of the agent hierarchy tree

The lowest level agents, *stage agents* receive the vehicle position data relating to the vehicles who's approach relates to a single signal stage only. These data are refined by the stage agents into a simplified form which constitutes a *bid* for priority. These bids are received by the *junction agent*, which will then assign priority to the stage with the winning bid. In a situation where a number of closely connected signalized junction need to coordinate signal timings the junction agent will communicate with a *zone agent* above them in the hierarchy before assigning priority.

5.1 Prototype Control Algorithms

In this paper we present some results from tests of two prototype control algorithms on an isolated junction, so for these purposes the *zone agent* level of the hierarchy can be disregarded.

The approach at this early stage of the research has been to begin by testing two very simple algorithms.

Bidding Algorithm 1 (BA1) Each stage agent has a set (N) of vehicles (i) to consider. These are pre-selected on the basis of their position revealing that they are on the approach to the stage. The stage agent simply counts the number of vehicles i in the set of vehicles N that are stationary; this count is used as the bid (B).

$$B = \sum_{i \in N} \begin{cases} 1 & \text{if } V_i = 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Bidding Algorithm 2 (BA2) In this case the bid is calculated as a linear function of the number of vehicles in the set N , the speed of each vehicle V_i and the distance of each vehicle from the junction X_i .

$$B = \sum_{i \in N} 1 - \alpha V_i - \beta X_i \tag{4}$$

where α and β are coefficients that can be tuned.

Having received bids from all stage agents the Junction agent simply needs to select the stage with the highest bid and assign priority. To avoid changing the stages too rapidly the junction agent performs these *auctions* only at fixed a fixed time interval (δt).

Both of the algorithms tested use vehicle speed in the calculation. Vehicle speed is taken as the derivative of vehicle position over the previous two time steps in the database.

6 Testing prototype algorithms

6.1 Set up

The junction on which simulated test were carried out is the simple T-junction shown in Figure 8. This junction has three signal stages: stage 1 gives priority to vehicles approaching from the West and East arms of the junction, stage 2 is a right turn filter giving priority to right turning vehicles coming from the West arm and stage 3 gives priority to vehicles from the South arm.

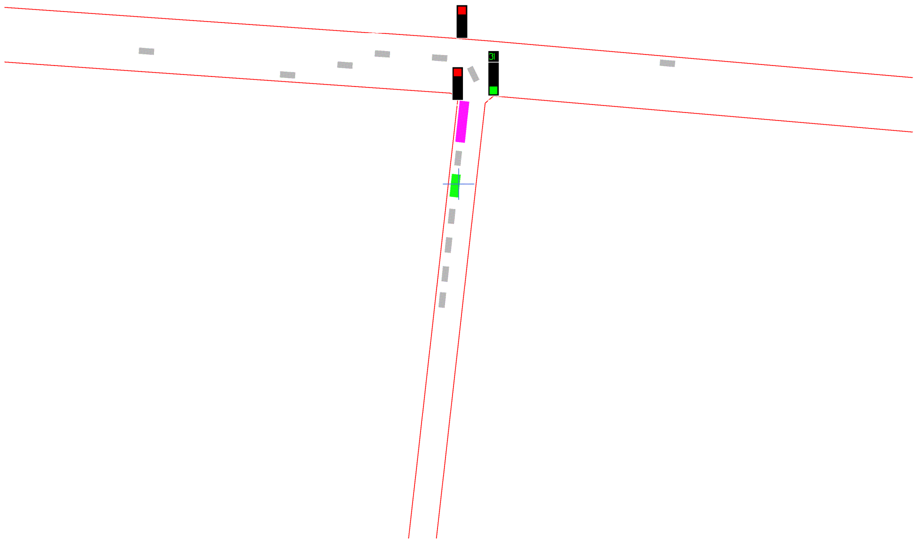

Figure 8: Paramics T-junction model used in the tests

Localization Accuracy In the tests described here two levels of localization accuracy were used. Both levels used a sampling rate of 1 Hz but in one case the positioning accuracy was 10m (1σ) an in the other it was 2m (1σ). No latency was simulated in these tests. There is an important difference between these two levels of accuracy, which is that with 2m accuracy the position of the vehicle can be resolved down to the *lane* level, where as with 10m accuracy the position of the vehicle can only be resolved to the level of *road*. This is important for the control algorithm when considering vehicles approaching the junction from the West. With lane resolution the algorithm can assign the vehicles in the right hand lane to stage 2 and those in the left to stage 1. Without lane resolution the control algorithm must assign all vehicles on the Eastbound arm to stage 2.

Baseline To provide a baseline for the tests loop detectors have been included in the simulated junction shown in Figure 8 using the facility included in Paramics. A test was carried out where the junction was controlled by the MOVA algorithm [Vincent and Peirce, 1988]. The loop detectors used have separate sensing loops for each lane, so like the more accurate probe localization system, MOVA has lane resolution.

Test #	Algorithm	δt (s)	Accuracy (m)	α	β
1	MOVA	-	-	-	-
2	BA1	30	2	-	-
3	BA1	10	2	-	-
4	BA1	10	10	-	-
5	BA2	30	2	0.01	0.001
6	BA2	30	2	0.02	0
7	BA2	30	10	0.01	0.001
8	BA2	30	10	0.02	0
9	BA2	10	2	0.02	0
10	BA2	10	10	0.02	0

Table 1: Itinerary of the tests performed with their parameter settings

Itinerary of tests A list of the tests performed is given in Table 1. Each test covered a simulated time of four hours, during which the level of demand was constant. The Demand matrix is shown in Table 2. The two bidding algorithms (BA1 and BA2) were tested using the two levels of localization accuracy and also two rates of auctioning (δt = 30 s and δt = 10 s).

	West	East	South
West	-	12.5	3.3
East	15.8	-	0.8
South	2.7	2.7	-

Table 2: Matrix of demands (Vehicles per minute) across the junction shown in Figure 8

6.2 Results

Figures 9 to 11 present statistics for delay, queuing time and vehicle speed averaged across all vehicles for the duration of the test. Test number two used the first bidding algorithm and an auctioning rate of δt = 30 s and a localization accuracy of 2m. The results show that despite the simplicity of this approach the performance of the algorithm equates very closely with MOVA. The performance of BA1 can be improved by reducing δt to 10 s as shown in test 3. However when the localization accuracy is reduced to 10m so that lane resolution is lost (test 4) then the performance of BA1 is reduced to below that of MOVA.

The second bidding algorithm (BA2) can produce further improved performance when using the lower auctioning rate (δt = 30 s), as shown in tests 5 and 6. Here two different settings for the coefficients were used, one with order of magnitude tuning (α = 0.01, β = 0.001) and one where β was set to 0, effectively eliminating the distance term from equation (4). Tests 5 and 6 were repeated in tests 7 and 8 with the localization accuracy lowered to 10m. This has a marked effect, with performance again falling below the MOVA baseline. In the last two tests (9 and 10) BA2 was tested using the faster auctioning rate of δt = 10 s in test 9, 2m positioning accuracy was used and in test 10, 10m positioning accuracy was used. Both these tests produced good performance indicating that the lower positioning accuracy is mitigated to some extent by the higher auctioning rate. In fact test 10 is the only test using 10m accuracy that outperforms the MOVA baseline.

One thing to note in tests 9 and 10 is that while their statistics for average delay and average speed are similar, the average queuing time is lower in test 10 than in test 9. This is because in test 10 more vehicles are stopped at the lights albeit for a shorter time. This can be an important consideration because stop-start driving is known to produce more CO_2 emissions than constant speed driving. This is also a reminder that the statistics that have been presented here to compare strategies are not the only statistics that may need to be considered when determining the performance of a control algorithm.

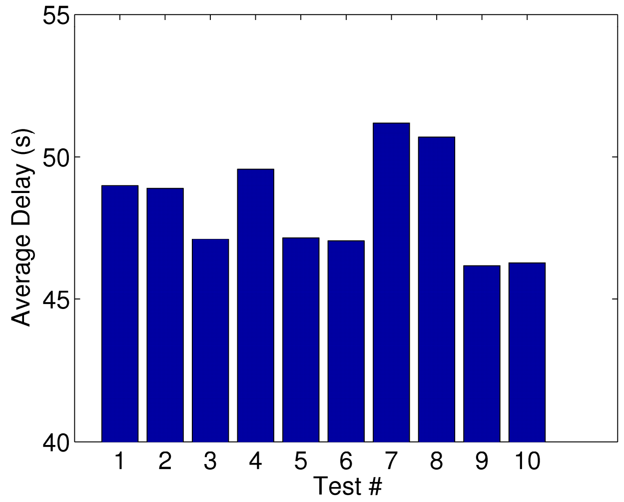

Figure 9: Average delay across all vehicles for the duration of the test

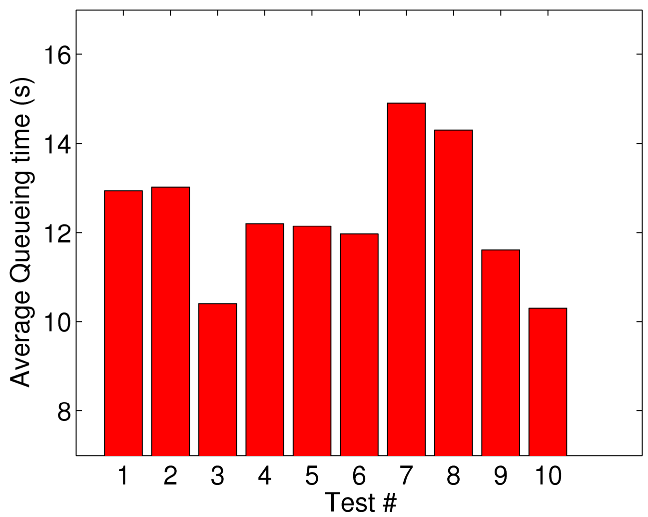

Figure 10: Average queuing time across all vehicles for the duration of the test

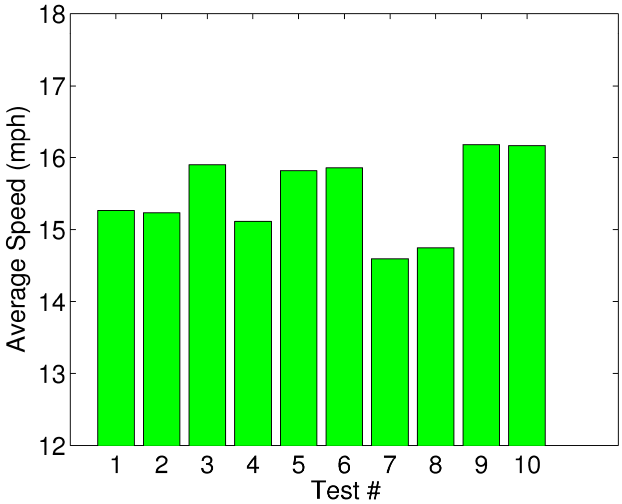

Figure 11: Average speed across all vehicles for the duration of the test

7 Conclusions

In this paper we have presented a methodology for testing urban signalized junction control algorithms that make use of localization probe data. Analysis of possible algorithms is at an early stage and we have presented results from some tests that implement two very basic algorithms. The performance of these algorithms on a simulated isolated junction compare favourably with the performance of the MOVA algorithm despite their simplicity. This indicates that the additional information contained in localization probe data is useful and further improvements in performance are likely as algorithm development continues.

An important result to note is the dependence of algorithm performance on the accuracy of the localization data. The algorithms tested performed significantly better when lane resolution was achievable. It is also worth considering that lane resolution is currently not achievable with GPS only localization systems.

References

S. Box and B. J. Waterson. Technologies for road vehicle localization – a technical review. *In review at Transportation Research Part C: Emerging Technologies, 2009*.

M. C. Choy, D. Srinivasan, and R. Cheu. Cooperative, hybrid agent architecture for real-time traffic signal control. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on,* 33(5):597–607, Sept. 2003. ISSN 1083-4427. doi: 10.1109/TSMCA.2003.817394.

M. Grewal, L. Weill, and A. Andrews. *Global Positioning Systems, Inertial Navigation and Integration.* John Wiley & Sons, 2001.

N. Hounsell, B. Shrestha, J. Head, S. Palmer, and T. Bowen. The way ahead for london's bus priority at traffic signals. *IET Intelligent Transport Systems*, 2(3):193–200, 2008.

P. Hunt, R. Bretherton, D. Robertson, and M. Royal. Scoot on-line traffic signal optimisation technique. *Traffic Engineering and Control*, 23:190–192, 1982.

P. Kompfner. *Cvis – cooperative for mobility*. Online at http://www.cvisproject.org/download/cvis brochure May2008 Final.pdf, 2008.

T. Kos, M. Grgic, and G. Sisul. Mobile user positioning in gsm/umts cellular networks. *elmar*, 0:185–188, 2006.

J. Levinson, M. Montemerlo, and S. Thrun. Map-based precision vehicle localization in urban environments. In Proceedings of Robotics*: Science and Systems*, Atlanta, GA, USA, June

2007.

W. Li and H. Leung. Constrained unscented kalman filter based fusion of gps/ins/digital map for vehicle localization. *In Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, volume 2, pages 1362–1367 vol.2, 2003.

OXTS. *Rt3000 inertial and gps navigation system.* OXTS brochure available online at: www.oxts.co.uk, 2009.

G. Rose. Mobile phones as traffic probes: Practices, prospects and issues. *Transport Reviews,* 26(3):275 – 291, 2006.

I. Sreedevi. *Its decision – services and technologies – loop detectors*. Available online at: http://www.calccit.org/itsdecision/serv and tech/Traffic Surveillance/road-based/inroad/loop_summary.html, 2005.

Trimble. Lassen iq gps module. *Trimble datasheet*, available online at: http://www.trimble.com, 2007.

G. Vincent and J. Peirce. 'mova': Traffic responsive, self-optimising signal control for isolated intersections. *TRRL Research Report*, RR170, 1988.

C. Wang, Z. Hu, S. Kusuhara, and K. Uchimura. Vehicle localization with global probability density function for road navigation*. Intelligent Vehicles Symposium, 2007 IEEE*, pages 1033–1038, June 2007. ISSN 1931-0587.

K. Wood, M. Crabtree, and S. Gutteridge. Pedestrian and vehicular detectors for traffic management and control*. TRL Report*, 2006.