This is a pre-publication draft of the paper published in the proceedings of the Road Transport Information and Control Conference (RTIC 2010). 25-27 May 2010, doi: 10.1049/cp.2010.0387.

Comparison of signalized junction control strategies that use localization probe data

Dr Simon Box*, Dr Ben Waterson*

*University of Southampton

Keywords: Localization, Signal, Control, Agents

Abstract

This paper presents a simulation test bed and a 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. Also presented are results from tests carried out using the simulation test bed to evaluate localization strategies. The tested strategies use a hierarchical structure of auctioning agents. Results from tests on an isolated signalized junction indicate that the performance of the auctioning agent algorithms compare favourably with the MOVA algorithm using inductive loop data. Results are also presented for tests on a twin junction where strategies are synchronized. These show a significant improvement in performance through synchronization.

1. Introduction

Recently a number of large European Commission funded projects (CVIS [6], 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 communication (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 [12], microwave emitter/detectors [15] 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 [14] for isolated junctions and SCOOT [5], 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 [11]. Probe data allow an analysis of the system that tracks each vehicle individually and can provide a higher resolution of position data.

Research that examines the use of V2I communications and localization systems in signalized junction control is already under way. The iBus project [4] uses localization systems on London buses to give them priority at signalized junctions.

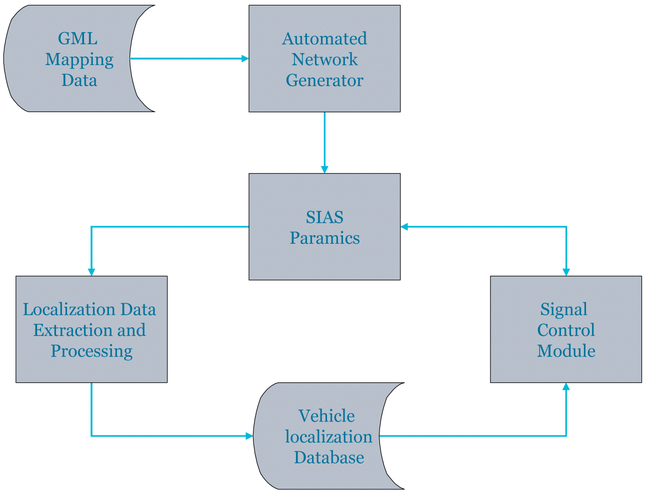
GML Mapping Data
Automated Network Generator
SIAS Paramics
Localization Data Extraction and Processing
Signal Control Module
Vehicle localization Database

Figure 1: Block diagram showing the simulation software architecture.

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

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

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

also contains results from tests on some simple control algorithms that are based on and *auctioning agents*. The performance of these algorithms is compared directly with the MOVA algorithm using simulated inductive loop data.

2. Simulator 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 then 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 the network and trace over it using the mouse to add nodes and links and define the nature of the network. To construct a simple un-signalized T-junction using this method requires minimum of 267 mouse click operations. 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.

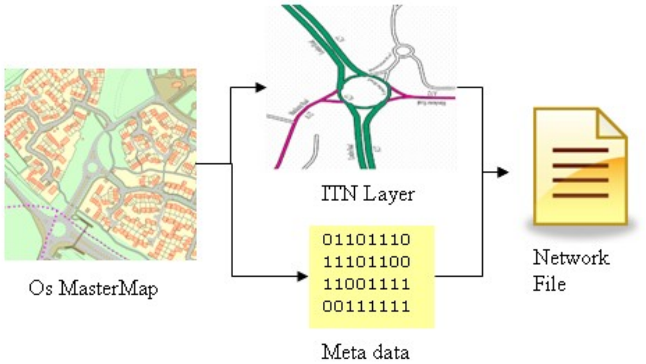

Figure 2: Network generator module structure.

The Network Generator Module (Figure 2) 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 project.

The user of the 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 access the MasterMap GML data and extract the integrated transport network layer (ITN), which provides the detailed geometry of the road network. And corresponding meta data layers, which provide details on the types of road, number of lanes, one way sections and other information. The module then uses these data to automatically generate a Paramics network file, which can be opened in Paramics for further editing and for simulations.

This approach 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.

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 which allows us to travel virtually through the example network and check details such as geometry, turn priorities and sensor positions.

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 [7], Global positioning system (GPS) [13], inertial measurement systems (IMU) [10], laser range-finding systems (LIDAR) [8], and computer vision systems [15]. 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 [9] and Bayesian recursive filtering techniques, such as the Kalman filter [3]. 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.

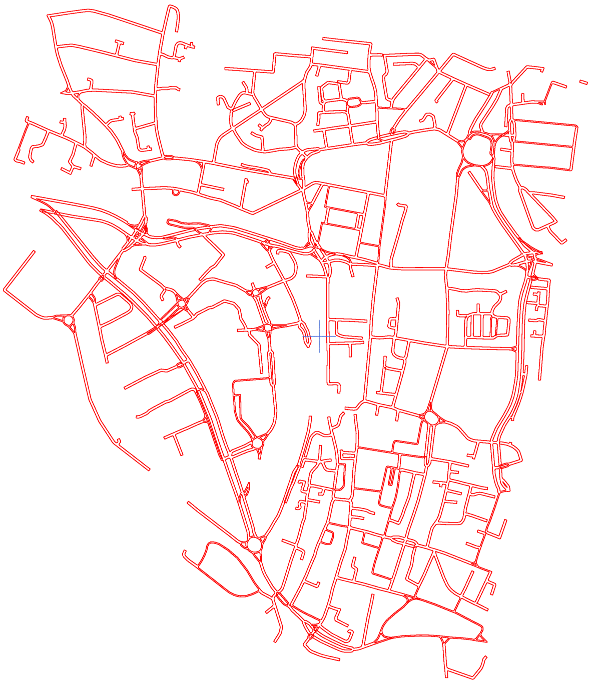

Figure 3: Large Paramics network of Southampton city centre

The performance of localization systems is a function of positioning accuracy, frequency of position measurements and reliability (e.g. latency). Figure 4 from [1] 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.

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 (x_P) are made stochastic by the addition of Gaussian noise (1).

$$x_S = x_P + \varepsilon \text{; where } \varepsilon \sim N(\mathbf{0}, \boldsymbol{\Sigma}) \text{ .} \qquad (1)$$

Here x is a two dimensional vector describing the vehicle's position, ε is sampled from a zero mean, two dimensional Gaussian distribution with covariance Σ . The covariance matrix is chosen to representative of the performance of a given localization system, such as one of the examples given in Figure 4. Thus the performance of signal control systems can be tested for different localization systems and different levels of localization performance.

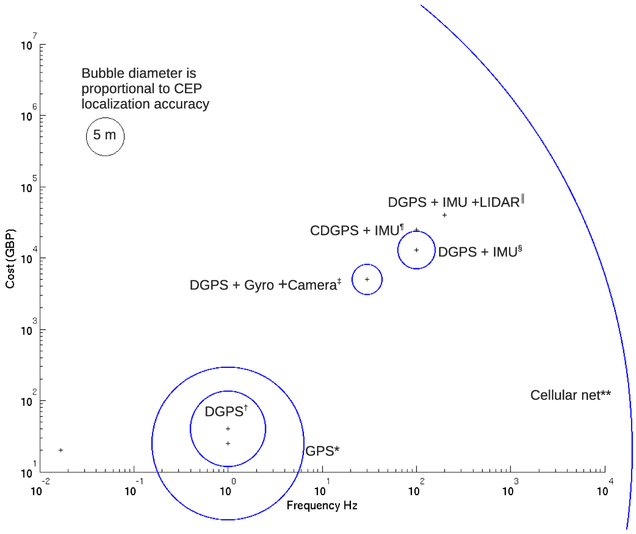

Figure 4: Comparison in the performance and cost of some example localization systems, several of the systems use more than one localization sensor in tandem

Similarly the signal control module, which receives the vehicle position data can either interpret the position data deterministically using, x_S or stochastically taking the probability of position P (x_S) to be Gaussian and centred on x_S (2).

$$P(x_S) = N(x_S | \mathbf{0}, \boldsymbol{\Sigma}) \text{ .} \qquad (2)$$

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 [2]. 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 [2], the structure is shown in Figure 5.

The lowest level agents, *stage agents*, receive the vehicle position data relating to vehicles whose 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 junctions need to coordinate signal timings the junction agent will communicate with a zone agent above them in the hierarchy before assigning priority.

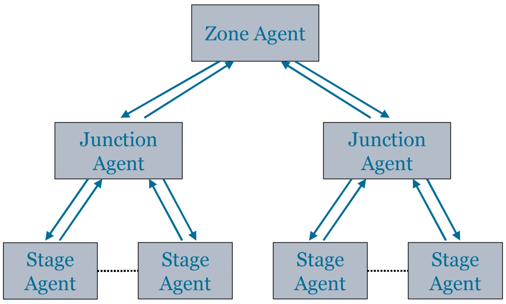

Figure 5: Structure of the agent hierarchy tree

5.1. Prototype Control Algorithms

In this paper we present some results from tests of two prototype stage agent algorithms. The approach at this early stage of the research has been to begin by testing 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 in the set of vehicles that are stationary; this count is used as the bid (B).

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

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.

$$B=\sum_{i\in N} 1-\alpha V_i-\beta X_i . \qquad (4)$$

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) called the *auctioning rate*. Both of the stage agent 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.

The behaviour of the *zone agent* is discussed in the context of an example in the next section.

6. Testing prototype algorithms

6.1. Set up

Simulated tests were carried out on two junctions, the simple isolated junction T-junction illustrated in is the Figure 6 and the twin junction shown in Figure 7. The T-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 and stage 3 gives priority to vehicles from the South. The twin junction in Figure 7 has two signalised intersections, each of which has the same staging as the T-junction.

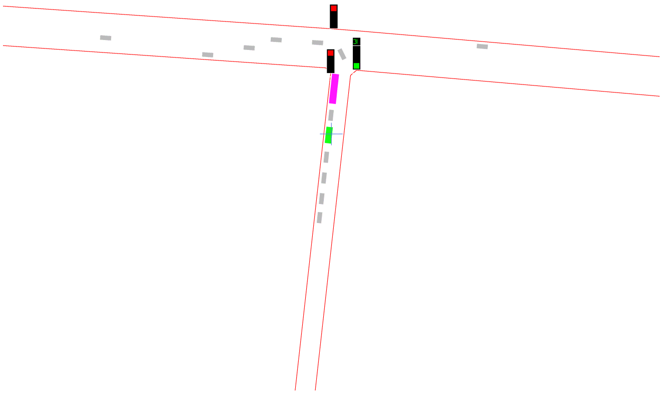

Figure 6: Paramics T-junction model used in the tests

Localization Accuracy In the simulation tests presented 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 T-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 T-junction shown in Figure 6 using the facility included in Paramics. A test was carried out where the junction was controlled by the MOVA algorithm [14]. The loop detectors used have separate sensing loops for each lane, so like the more accurate probe localization system, MOVA has lane resolution.

Here we present results comparing the performance of MOVA compared with Bidding Algorithm 1 and Bidding Algorithm 2 using both 10m and 2m localization accuracy.

The coefficients used in BA2 where α = 0▹01 and β = 0▹001. In all tests the auctioning rate was 10s.

Each test covered a simulated time of four hours, during which the level of demand was constant. The Demand matrix is shown in Table 1.

	West	*East*	*South*
West	-	750	200
East	950	-	50
South	160	160	-

Table 1: Matrix of demands (Vehicles per 5 minutes) across the junction shown in Figure 6.

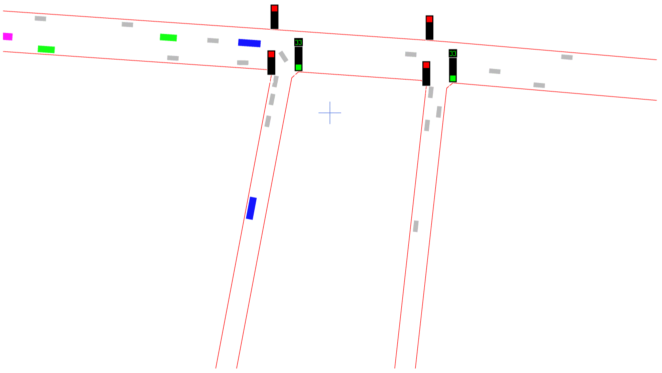

Figure 7: Twin junction Paramics model used in the tests

6.2. Results

Some averaged statistics for delay, vehicle speed and queuing time are shown in Table 2. Figure 8 shows plots of cumulative average delay (averaged over 1 minute intervals) for each of the four hour tests. This shows that the performance of each of the algorithms is stable over time allowing them to be ranked in order of performance. This shows that all the presented algorithms outperform MOVA based on the metrics used. It shows that the slightly more complicated BA2 outperforms BA1 and in general tests using the higher accuracy localization data perform better.

	Mean Delay (s)	*Mean Speed (mph)*	*Mean queue time (s)*
MOVA	49.0	15.26	12.94
BA1(10m)	48.57	15.11	12.21
BA1(2m)	47.11	15.90	10.4
BA2(10m)	47.65	15.68	12.96
BA2 (2m)	45.29	16.52	9.597

Table 2: Averaged statistics for several different control strategies.

Synchronized Agents The twin junction shown in Figure 7 features two signalized intersections close together. In this situation it makes sense to coordinate the actions of the two intersections by invoking the “zone agent” from the agent hierarchy shown in Figure 7. In this case the zone agent begins by comparing the winning bids from the two junction agents. The junction with the highest winning bid is chosen to lead the synchronization process. The other junction has its bids weighted to encourage synchronization.

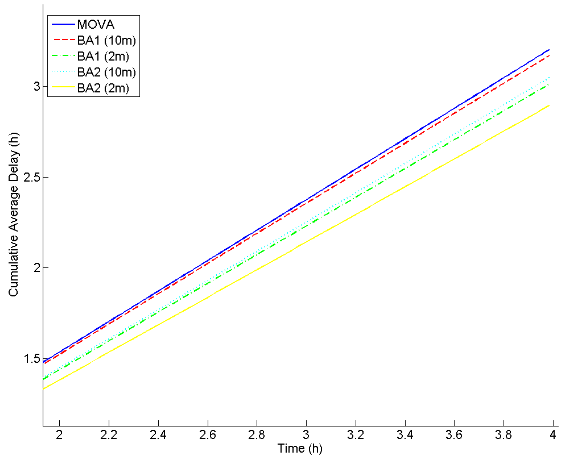

Figure 8: Cumulative average delay vs time for several control strategies.

For example if stage 1 of the Eastern junction is the highest overall bid then that stage will be selected for a green light. The bid for stage 1 on the Western junction will then be adjusted as follows.

$$B_{W1} = B_{W1} + CB_{E1} \ . \qquad (5)$$

Where B_{W1} is the bid for stage 1 on the Western junction, B_{E1} is the bid for stage 1 on the Eastern junction and C is a weighting coefficient, which is tuneable. This will increase the probability that stage 1 is selected by the Western junction agent thus synchronizing the flow through the two junctions. Similar rules are applied to all stage combinations.

Figure 9 shows plots of cumulative average delay for the twin junction being operated by stage agents using BA2 and the higher localization accuracy (2m). In one test the junction agents make their decision independently and in the other they are coordinated by a zone agent as described above.

This shows that, as expected, synchronization significantly improves performance.

6.3. Discussion

The results from the tests on an isolated T-junction have shown that the richness of the information contained within individual vehicle localization probe data means that even a very simple control algorithm that just accurately counts the number of stationary vehicles on each stage approach can produce better performance than the current standard control method, using the MOVA algorithm. Adding a little more complexity to the bid calculation to produce BA2 enabled the performance to be improved further. Also both algorithms performed significantly better when the localization data were accurate enough for the stage agents to resolve which lane each vehicle was in.

All the algorithms presented outperformed MOVA on the metrics used but it must be noted that these algorithms, while an interesting demonstration, are too simple to be deployed in the real world, where additional considerations need to be met. For example the metrics used do not account for the scenario of a single vehicle being held at a red light for a very long time, something that is undesirable in the real world but possible under the methods used here. Furthermore there are other interesting metric such as $C0_2$ emissions which it is desirable to minimize but aren't considered here.

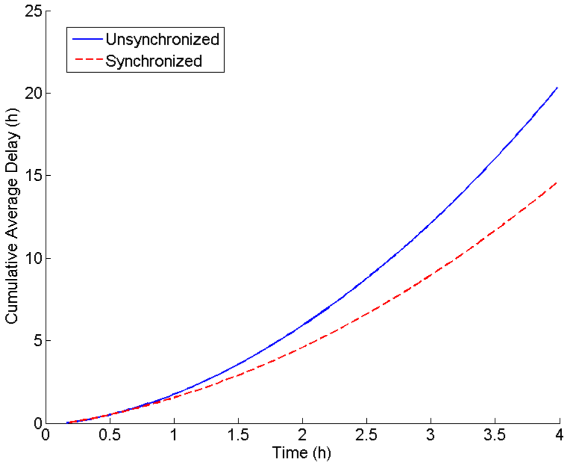

Figure 9: Cumulative average delay vs time for synchronised and unsynchronised control strategies.

The tests on the twin junction highlight the importance of using zone agents at the top of the agent hierarchy to coordinate junctions that are closely connected. We have shown that the simple auctioning process developed can be easily adapted for zone agent control by allowing the zone agent to weight the bids in certain junction agents.
Incidentally the upward curve seen in the plots of cumulative average delay in Figure 9 are due to the fact that in these tests the flow conditions were saturated due to high demand and queues are growing.

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 algorithms is at an early stage and we have presented results from some tests that implement two simple stage agent algorithms. The performance of these algorithms on a simulated isolated junction compare favourably with the performance of the MOVA algorithm despite their simplicity. 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 worth considering that lane resolution is currently not achievable with GPS only localization systems. Further tests have shown that the auctioning agent approach can be extended to coordinate control between junctions and test results show that the benefits of doing this are significant.

References

[1] S. Box and B. J. Waterson. Technologies for road vehicle localization – a technical review. In review at Transportation Research Part C: Emerging Technologies, 2009.
[2] 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.
[3] M. Grewal, L. Weill, and A. Andrews. Global Positioning Systems, Inertial Navigation and Integration. John Wiley & Sons, 2001.
[4] 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.
[5] P. Hunt, R. Bretherton, D. Robertson, and M. Royal. Scoot on-line traffic signal optimisation technique. Traffic Engineering and Control, 23:190–192, 1982.
[6] P. Kompfner. Cvis – cooperative for mobility. Online at http://www.cvisproject.org/download/cvis brochure May2008 Final.pdf, 2008.
[7] T. Kos, M. Grgic, and G. Sisul. Mobile user positioning in gsm/umts cellular networks. elmar, 0:185–188, 2006.
[8] 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.
[9] 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.
[10] OXTS. Rt3000 inertial and gps navigation system. OXTS brochure available online at: www.oxts.co.uk, 2009.
[11] G. Rose. Mobile phones as traffic probes: Practices, prospects and issues. Transport Reviews, 26(3):275 – 291, 2006.
[12] 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.
[13] Trimble. Lassen iq gps module. Trimble datasheet, available online at: http://www.trimble.com, 2007.
[14] G. Vincent and J. Peirce. 'mova': Traffic responsive, self-optimising signal control for isolated intersections. TRRL Research Report, RR170, 1988.
[15] 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.

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